Содержание

Contents

Предисловие

Ежегодный международный научный семинар «Передовые методы информационных и коммуникационных технологий» (Advances in Methods of Information and Communication Technology, AMICT) является непосредственным продолжением ежегодного международного семинара «Неделя финской информатики в Петрозаводском государственном университете» (Finnish Data Processing Week at the Petrozavodsk State University, FDPW). Впервые семинар FDPW проводился в 1994 году как неделя лекций и докладов приглашенных лекторов из Хельсинкского университета. Однако рамки лекционной недели вскоре расширились до научно-исследовательского семинара, на котором представлялись доклады из различных научных и образовательных организаций, центров, институтов и предприятий промышленности Финляндии и Республики Карелия, а также из ряда регионов Российской Федерации. Это расширение привело к смене названия, которая фактически произошла в конце 90-х годов. Интересующийся читатель найдет более детальное описание истории семинара на странице `http://www.cs.karelia.ru/fdpw/index.php.en`.

Одиннадцатый том сборника Трудов ежегодного международного научного семинара «Передовые методы информационных и коммуникационных технологий» содержит избранные статьи на основе докладов, которые были представлены на очередном семинаре, проходившем 19–20.05. 2009 под названием AMICT’2009.

Программа семинара состояла из 19 докладов участников от университета Хельсинки, Петрозаводского государственного университета, Санкт-Петербургского государственного университета, университета Куопио и Института прикладных математических исследований Карельского научного центра Российской академии наук.

На основе представленных на семинаре докладов были подготовлены и включены в состав данного сборника 12 статей и 7 тезисов студенческих проектов для платформы Maemo.

Статья Ю. Кангашарью, М. Му и Г. Д. Колусси посвящена исследованию метрик качества обслуживания на уровне приложений на примере трафика Интернет.

И. Л. Братчиков и И. С. Лебедев рассматривают проблему автоматической обработки спецификаций естественного языка для генерации программного кода. Их результаты учитывают вычисление объектов программы и связь между ними.

Тина Никландер представляет краткий обзор методик кэширования в многоядерной системе. Решения разделены на две категории: архитектурного и программного уровня.

Статья Александра Норты представляет среду для современных электронных деловых транзакций. Здесь же приведен обзор исследований, которые требуют особого внимания при реализации таких сред.

В статье М. Форселла, В. Леппанена и М. Пенттонена обсуждаются проблемы проектирования параллельного процессора. Они изучают альтернативное проектирование многоядерного процессора, что позволяет эффективно выполнять параллельные программы.

В статье Р. С. Горичевой и Е. В. Морозова рассматривается проблема регенеративной имитации очередей с конечным буфером. Такие модели важны для сетевых систем, поскольку они позволяют получить надежную доверительную оценку производительности в стационарном состоянии.

Л. Лангохр использует вероятностные меры подобия и k-medoid кластеризацию для нахождения групп узлов и представителей этих групп в сети.

Г. Лежён в своей статье описывает расширение системы PULS для обработки французских текстов, в дополнение к поддерживаемому в настоящее время английскому. Тем самым разрабатываются новые механизмы извлечения информации, поддерживающей несколько языков.

О. В. Лукашенко и Е. В. Морозов представляют обзор самоподобных процессов в моделировании коммуникационной сети. Они также обсуждают методы обнаружения нормальности сетевого трафика.

Статья М. Первилы представляет систему измерения ошибок в большой компьютерной сети с целью анализа наиболее распространенных отказов.

А. С. Румянцев и А. Л. Чухарев представляют построение модели процесса обработки ошибок пакета в средах NanoLOC методом хаотической карты для изучения тяжелых хвостов и зависимости в беспроводных средах.

Г. С. Сиговцев и И. О. Семенов используют когнитивные карты для построения оглавлений адаптивных образовательных ресурсов. Они также представляют несколько эволюционных алгоритмов для решения проблемы.

Мы хотим выразить нашу искреннюю признательность ректору Петрозаводского государственного университета, профессору А. В. Воронину, президенту ПетрГУ, профессору В. Н. Васильеву и проректорам Н. С. Рузановой и Н. В. Доршаковой, которые как организационно, так и финансово поддержали семинар. Мы благодарны заведующему кафедрой информатики Хельсинкского университета профессору Х. Тойвонену за поддержку публикации данного сборника трудов.

Мы также хотим поблагодарить доцентов Д. Ж. Корзуна и Т. Никландер, которые активно участвовали в организации семинара в Петрозаводске, а также взяли на себя организацию технической подготовки сборника.

Техническое редактирование сборника было тщательно выполнено М. Первила (Хельсинкский университет) и Д. Б. Чистяковым (Петрозаводский государственный университет). Всем этим лицам мы выражаем нашу искреннюю благодарность.

Мы также благодарим за значительную поддержку, традиционно оказанную нам международными отделами наших университетов-организаторов в лице П. Тауриайнен и Л. Ю. Куликовской.

Доц. Юрий Богоявленский, заведующий кафедрой информатики
и математического обеспечения
Петрозаводского государственного университета

Проф. Юсси Кангашарью, кафедра информатики
Хельсинкского университета

Preface

The Annual International Seminar «Advances in Methods of Information and Communication Technology» AMICT is a direct continuation of the «Finnish Data Processing Week at the University of Petrozavodsk». FDPW was originally started in 1994 as a guest lecturers' week, where the presentations came from the University of Helsinki. However, the frame of FDPW very soon developed to a research-oriented seminar with presentations from various Finnish, Karelian, and Russian computer science departments, research institutions, and even from industry. An interested reader finds a more detailed description of the FDPW history on the www page http://www.cs.karelia.ru/fdpw/index.php.en.

The tenth volume of the Proceedings of the Annual Finnish Data Processing Week / Advances in Methods of Information and Communication Technology contains selected presentations from the seminar organized 19.–20.5. 2009 under the title AMICT 2009.

The seminar program consisted of 15 presentations with participants from the University of Helsinki, the State University of Petrozavodsk, the University of Saint Petersburg, University of Kuopio, and the Karelian Research Center of the Russian Academy of Science.

Out of the seminar presentations and posters, these proceedings contain 12 papers and 5 abstracts for student presentations from their projects on the Maemo platform.

The paper of J. Kangasharju, M. Mu, and G. D. Colussi covered the topic of investigating application-level quality of service metric on Internet traffic.

I. L. Brathikov and I. S. Lebedev consider the problem of automatic processing of natural language specifications for program code generation. Their results allow for the calculation of the objects of the program and the communications between them.

Tiina Niklander presents an overview of caching techniques in multicore systems. The solutions are divided into two categories on the architecture and software levels.

Alexander Norta's article presents a framework for advanced electronic business transactions. The paper also maps out areas of research that need specific attention in realizing electronic business transaction frameworks.

The paper of M. Forsell, V. Leppänen, and M. Penttonen discusses challenges in parallel processor design. They study alternate multi-core processor designs that can efficiently run parallel programs.

The paper of R. S. Goricheva and E. V. Morozov considers the problem of regenerative simulation of finite buffer queuing systems. Such models are important in telecommunications as they allow for reliable confidence estimation of the steady-state network performance.

L. Langohr uses probabilistic similarity measures and k-medoids clustering to find groups of nodes and representatives for these groups in a network.

G. Lejeune's article extends the PULS system to handle French language texts in addition to the currently supported English, in an attempt to develop new mechanisms for extending information extraction systems to support multiple languages.

O. V. Lukashenko and E. V. Morozov present a survey of self-similar and long-range dependent process in communication network modeling. They also discuss methods to detect normality of network traffic.

M. Pervilä's paper presents a system for measuring faults in a large computer network, in order to analyze common-mode failures.

A. S. Rumyantsev and A. L. Chukharev build a chaotic map for NanoLOC environments, in order to study heavy tails and long-range dependence in wireless environments.

G. S. Sigovtsev and I. O. Semenov use cognitive maps in order to build table of contents of adaptive educational resources. They also present several evolutionary algorithms for solving the problem.

We want to express our sincere gratitude to the Rector of the University of Petrozavodsk, professor Anatoly V. Voronin, the President of the University, professor Victor Vasiljev, and to the vice-rectors Natalia S. Ruzanova, and Natalia V. Dorshakova, who all have provided both organizational and financial support to the seminars. We are grateful to professor Hannu Toivonen, the head of the Department of Computer Science of the University of Helsinki, for the support to the publication of these proceedings.

We also want to thank Dr. Dmitry Korzun, who undertook the burden of the event organization in Petrozavodsk, and his Helsinki partner Dr. Tiina Niklander.

The technical editing of the proceedings was carefully done by Mikko Pervilä (University of Helsinki) and Dmitrii Chistykov (Petrozavodsk State University). The book was printed by the We would like to thank both of them for their work.

We also gratefully acknowledge the remarkable support we traditionally have had from the International Offices of our Universities personified by Päivi Tauriainen and L. Y. Kulikovskaya.

Dr. Yury Bogoyavlenskiy, Head of the Department of Computer Science
of the University of Petrozavodsk

Prof. Jussi Kangasharju, Department of Computer Science
of the University of Helsinki

Application-Level Fairness

Prof. Jussi Kangasharju[†], Mu Mu[‡], Gian Donato Colussi[†]

[†]Department of Computer Science, University of Helsinki
P.O.Box 68, FI-00014 University of Helsinki, Finland

[‡]Computing Department,Lancaster University
Lancaster LA1 4WA, United Kingdom

E-mail: jakangas@cs.helsinki.fi, m.mu@lancaster.ac.uk,
Dado.Colussi@helsinki.fi

Abstract

A fair bandwidth allocation is typically considered sufficient evidence of network flows and applications being treated fairly. While this is sufficient for applications which depend only on bandwidth, as our work in this paper shows, other applications can suffer greatly. In reality, the utility of an application to the user depends on more than bandwidth allocation. In this paper, we define application utilities as functions of bandwidth, delay, jitter, and data loss. We propose to use these utilities for determining application-level fairness. We discuss profiles of different applications and derive utility functions for several real-world applications. We analyze utility functions with an analytical model and our results show that a fair bandwidth sharing by no means implies that the applications are treated fairly. Our results indicate a clear need for future research in this topic.

Contents

1 Introduction

As argued by Shenker, network performance should be evaluated in terms of the degree to which the network satisfies the requirements of each user's applications instead of network-centric quantities like utilization [1]. If application requirements are met, the application performs well and the user is satisfied. The question is how to model application requirements and satisfaction of users.

Restricting modeling to bandwidth is common to most approaches to fairness. We believe that such restriction to a single resource also restricts the applicability of those approaches. In this paper, we model the requirements of applications as multidimensional utility functions. User satisfaction is driven by many factors and we concentrate our work on the factors influenced by the network: bandwidth, delay, jitter, and data loss. Although these factors do not fully capture application utility regarding network, we believe they are sufficient for modeling the essential properties of network applications. We will show in this paper that our model captures the interactions between different applications in a way which is impossible to obtain by considering only bandwidth as resource.

This paper investigates the use of application level utility metrics as indicators of fairness in the network. We show that bandwidth alone is not sufficient to determine whether different kinds of applications are treated fairly. We also present a survey and classification of network applications and derive representative utility functions for the classes. Our work helps us understand the true nature of networked applications, their interaction in times of congestion, and the efficiency of using the network as a resource.

1.1 Motivation and Related Work

A particular challenge in a packet switched network like the Internet, is congestion which emerges in routers when the resource demand of flows grows beyond the offered capacity. Congestion control protocols aim at a fair distribution of network resources. Studies on fairness have typically focused on bandwidth distribution, e.g., [2–6]. Our objective is to maximize the *aggregate of application utilities*. We argue that fairness reduces to bandwidth distribution only when all flows can operate under optimal conditions, i.e., with minimum delay and no data loss. We support this argument by deriving utility functions for different application classes and show how they depend on the mentioned parameters. A fair bandwidth distribution can result in a poor aggregate utility due to loss and delay in a congested network environment.

Our work is motivated by the following observation. Traffic flows from applications crossing a congested router are *affected* by the resource shortage causing the congestion, most notably in form of increased queuing delay and packet loss. The *impact* of congestion varies per application, depending on the service required by the application. Users *experience* the impact of congestion as they interact with their applications. We believe that the correct way to study fairness is a cross-cutting approach, which includes all the three aspects mentioned above: network level effects, their impact on the application, and the overall effect on the user experience. In this paper, we concentrate on how network level effects impact application level performance.

Our work differs from previous approaches in that instead of focusing on sharing of single resources (i.e., bandwidth), or trying to optimize a set of network parameters (e.g., as in QoS), our goal is to *define fairness at the application level*. We present a model which captures how users experience network performance and show that traditional metrics are not sufficient for capturing what the users see.

2 Application Classes

Application requirements can be translated into network parameters of bandwidth (x), delay (τ), jitter (ϕ), and loss (p), by taking into account the actual characteristics of the application type. In this paper, we model applications as four-dimensional utility functions of these core parameters.

The essential observation is that there are some *application dependent bounds for delay and data loss*. Below a threshold value, no differences in quality can be perceived and above another threshold value the application is unusable. Between the threshold values the perceived quality degrades depending on the application. Section 3 presents threshold values for several applications.

2.1 Utility Functions

We now present our utility functions for the four different parameters—bandwidth, delay, jitter, and data loss. An interesting observation regarding the effect of bandwidth and the other parameters is that increase in bandwidth has a *positive* effect on application performance, while an increase in the other parameters has a *negative* effect on performance. Most existing work takes into account bandwidth and thus neglects the negative effects of the other parameters. For this reason, we also use the term *damaging parameters* for delay, jitter, and loss, and call the result damage utility.

The approach is similar to the E-Model developed by ITU [8]. However, the E-Model is specific to telephone applications and our work is a generalization of the E-Model into any

Bandwidth Utility

For determining the bandwidth utility, we divide applications into two types: elastic and real-time. Elastic applications are able to fine-tune their transmission rate and have no minimum or maximum bandwidth. Typically elastic applications use TCP. Real-time applications, on the other hand, usually have a restricted set of transmission rates which depend on the data being transmitted.

We define the bandwidth utility for elastic traffic as

$$u_x^e(x) = C_{el}\log(1+x) \tag{2.1}$$

where x is the bandwidth share. Similar function has been used in [2]. The factor C_{el} normalizes the utility to interval $[0,1]$. For an elastic application, we can determine a bandwidth which satisfies the user and thus gives utility 1. If we denote this bandwidth as x_s, then $C_{el} = 1/\log(1+x_s)$. If the application gets more bandwidth than x_s, then its utility is mathematically greater than 1. We explicitly allow such values since they serve as an indicator that the application is receiving more resources than needed. If an elastic application has $u_x^e > 1$, we could take away some of its bandwidth, without damaging it. We return to selection of x_s in Section 4.

Because the transmission rates of real-time applications are limited, we define a discrete utility curve with a maximum transmission rate. We define the real-time bandwidth utility as

$$u_x^{rt}(x) = C_{rt}\log(1+\gamma x_i) \qquad x_i \le x < x_{i+1} \tag{2.2}$$

where x is the available bandwidth, x_i and x_{i+1} are the application-specific transmission rates, and γ is a *priority factor* (typically $\gamma > 1$), because we assume that real-time traffic is likely to give a greater user satisfaction than elastic traffic for a given bandwidth share. The equation gives the same utility for real-time traffic at rate x_i than for elastic traffic at rate γx_i. The value of γ depends on the particular real-time application. For a real-time application with c different transmission rates, the normalization coefficient C_{rt} is given by $C_{rt} = 1/\log(1 + \gamma x_c)$.

Utility Model for Damaging Parameters

For the damaging parameters—delay, jitter, and loss—we define the following *damage utility function*:

$$u_{\tau,\phi,p}(z) = \begin{cases} 1 & \text{if } z \leq th_{min} \\ F(z) & \text{if } th_{min} < z < th_{max} \\ u_{min} & \text{if } z \geq th_{max} \end{cases} \tag{2.3}$$

where z is the actual parameter value, th_{min} and th_{max} are threshold values, $0 \leq F(z) \leq 1, z \in [th_{min}, th_{max}]$ is an application-specific function which defines how the utility decreases between the thresholds and u_{min} is the utility achieved after th_{max}. For some applications u_{min} is 0, but for others it could be some other low constant value.[1] Due to the way we model the impairments caused by the damaging parameters in Section 2.2, we have chosen to limit the values of their utility functions to $[0, 1]$.

Different applications are likely to have their own $F(z)$ and u_{min}, *however*, the general form of the damage function is justified by existing studies. In other words, below a certain value, utility is not affected; above another value utility remains at a constant level; between the two values, utility decreases in an application-specific manner. In Section 3 we present threshold values th_{min} and th_{max} for different applications which have been determined experimentally. Some studies [7] even include a definition of $F(z)$. In this paper, we do not consider the details of $F(z)$ or u_{min}, but assume that defining them for each larger application class is feasible in practice.

Figure 1 shows examples of different damage functions.

[1] For example, a telephone conversation with high delay becomes a one-way conversation, with low, but non-zero utility.

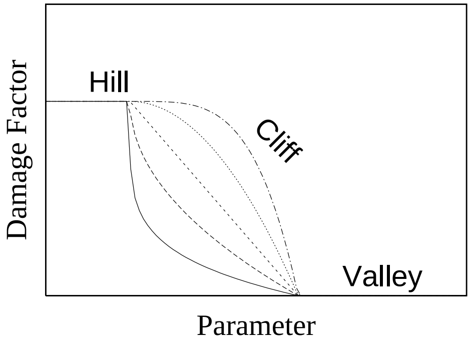


Figure 1: *Example of damage function df.*

2.2 Aggregate Utility

So far, we have discussed individual utility functions for the parameters. When aggregated with a *combination function* U, they indicate the overall utility of an application. We consider it desirable for U to have the following intuitive properties: 1) If all damaging utilities are 1, then $U = u_x(x)$; 2) If any damaging utility is less than one, then $U < u_x(x)$; 3) If any damaging utility is 0, then $U = 0$.

In this paper we use a product combination function:

$$U_1(x, \tau, \phi, p) = u_x(x) \cdot u_\tau(\tau) \cdot u_\phi(\phi) \cdot u_p(p) \tag{2.4}$$

where τ is delay, ϕ is jitter, p is packet loss rate and x is bandwidth share. If an application is insensitive to some parameters, the corresponding utility is set to 1. A product combination function is also used in [8], where the psychological impairments of damaging parameters on application performance are considered.

Table 1: *Threshold values for different applications reported by different sources. The values for th_{min} indicate satisfactory quality. The value for th_{max} indicates a level where the users in the cited study considered the application to be unusable. Sources: [9–21]*

Application	**Bandwidth**		**Delay**		**Jitter**		**Loss**	
	th_{min}	th_{max}	th_{min}	th_{max}	th_{min}	th_{max}	th_{min}	th_{max}
VoIP	Codec	64 kbit/s	100 ms	150 ms	40 ms	75 ms	1%	3%
Video streaming	Depends on codec			10 s	(Buffer)		10^{-4}%	1%
Video phone	16 kbit/s	384 kbit/s	150 ms	400 ms	50 ms	80 ms		1%
IPTV	Depends on codec		200 ms	150–500 ms	100 ms	500 ms	10^{-4}%	1%
Web browsing	Elastic		2 s	4 s	N/A		N/A	
Unreal Tournament	Decides framerate		40 ms	60 ms	N/A			
Quake3	Decides framerate		50 ms	200 ms			N/A	
Xbox Halo Client	Decides framerate		50 ms	150 ms			1.5%	3.5%
Bulk data transfer	Elastic		N/A		N/A		N/A	

3 Real Applications

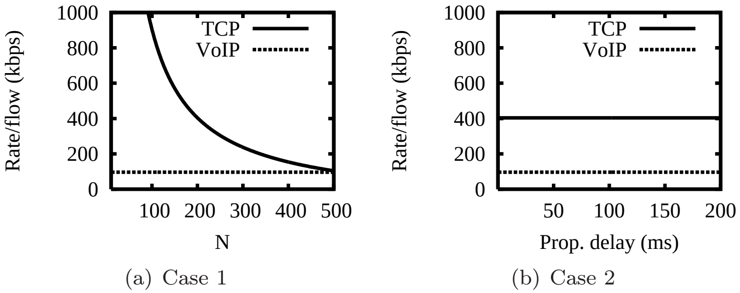
We now present threshold values for several applications which have been discovered in user studies reported in the literature. We summarize these in Table 1. The list is not meant to be exhaustive; instead, our goal is to illustrate that the utility parameters already exist for the most common Internet applications. We show threshold values th_{min} and th_{max} and indicate the work on which the numbers are based. An empty entry means that no values were reported and N/A means that the application was found to be independent of that parameter.

The values in Table 1 come from several different studies and may in some cases (e.g., IPTV delay) appear inconsistent. If different studies reported different values, we chose the stricter numbers. The values for loss assume the use of loss concealment algorithms (where possible) and depend also on the codec. Available bandwidth for the games (Unreal Tournament, Quake3, and Halo) determines the framerate of the game; the reported studies found that at 30 frames/s the game was good and 7 frames/s the quality was too poor to play. Knowing the amount of data in a frame, it is possible translate the framerates into bandwidth (game-specific).

We have reported values for jitter, but do not expect it to be a major factor in practice. This is because jitter can be absorbed by buffering with negligible effect on user level performance [22, 23]. Also, not many studies consider jitter, exceptions being [21] which states that jitter at high delays in first-person-shooter games is highly damaging, and [24] which states that high jitter affects video quality like loss; neither provides any threshold numbers. For these reasons, we do not consider jitter in our evaluation below, but have reported the values in Table 1 for the sake of completeness.

Bandwidth values for bulk data transfers are also an area with little existing data. These values are important, since they are needed to determine the value of C_{el} in equation (2.1), which is used to "calibrate" the bandwidth utility. Large-scale user studies would be required to determine appropriate values, but they are beyond the scope of this paper. Furthermore, we speculate that the user's satisfaction would also be affected by the conditions under which the download happens, as already observed on studies of web downloads [16–18]

A further point to note about bandwidth and its effect on utilities is the



(a) Case 1 (b) Case 2

Figure 2: *Bandwidth shares of applications*

saturation of a user's downstream link. Suppose the user's own network connection is the bandwidth bottleneck for an application and suppose that this bandwidth is not sufficient to yield a high utility for that user. Then, the best bandwidth utility that can be achieved is governed by the downstream link, but this maximum utility will be less than 1. However, there is no way to improve it. This observation applies equally to both elastic and real-time applications.

Some values, e.g., delay for VoIP, are widely applicable, since they come from extensive user studies. Other values, e.g., bandwidth, may depend heavily on the actual codec being used. Furthermore, individual user preferences could introduce some variation in the values. Although user preferences are a factor, we strongly believe that deriving general utility functions for larger application classes is feasible. We believe that all the different applications on the current Internet can be grouped into about 10 different classes and that future applications would not need more than 15–20 classes. This would provide enough flexibility to distinguish between the requirements of different applications, and still be feasible to derive generally applicable utility functions for all classes.

4 Analysis and Evaluation

-3mm We now present an analytical evaluation which shows that fair bandwidth sharing is not sufficient to capture fairness in terms of application utilities. We simulate a network and calculate the individual utilities for each of the four parameters and combine them with the product combination function from equation (2.4). We use the threshold values from Table 1. and a linearly decreasing $F(z)$ for delay and loss.

We consider a bottleneck link with a bandwidth capacity C where all flows experience the same propagation delay t_p. We assume N_V VoIP flows and N_T TCP flows that share the bottleneck link. We assume the VoIP flows take 96 kbit/s share each and let the TCP flows take the rest. Assuming the bottleneck implements RED, we can find an equilibrium with the TCP response function [25]. At equilibrium, RED determines the packet loss rate p by using the average queue length q, which determines the queuing delay t_q. Then, all flows crossing the bottleneck link will see an end-to-end delay $t_p + t_q$ and packet loss rate p. By varying C, t_p, N_V and N_T we can study the system equilibrium properties and the application utilities at equilibrium. We have chosen to use RED because it can be modeled analytically with relative ease. RED marks (or drops) packets as follows. When the average queue length is below min_{th}, no packet is marked. When it is between min_{th} and max_{th}, marking probability varies linearly between 0 and max_p, and when it is above max_{th}, all packets are marked. We set the RED parameters as follows: $max_p = 0.1$, $min_{th} = 0$ (needed by our analytical model to find the equilibrium), and max_{th} was varied across a large range to cover many different scenarios. We covered the range equivalent to 0–3000 packets and found the optimum to be around 300 packets, which is used in the results reported here.

We analyze two cases. In the first case, we vary the number of flows in the system and keep propagation delay fixed. In the second case, we also vary propagation delay. In both cases, bandwidth is fairly shared, and most existing work would consider the flows being treated fairly. By examining the utilities, we see that the flows are indeed not treated fairly.

We fixed the value of x_s (see Section 2.1) as follows to determine at what bandwidth do the TCP flows achieve utility 1. Recall from Section 3 that a VoIP flow is not affected by delays under 100 ms or packet loss rates below 1%. We used these values in the TCP response function [25] and determined that under such network conditions, a TCP flow would receive 170 kbit/s bandwidth. We chose this point as the "anchor point", because any change in either delay or loss would change the utility of VoIP from 1, by either reducing it or giving more "breathing room". Thus, it is fair to consider that TCP also achieves utility 1 at this same anchor point.

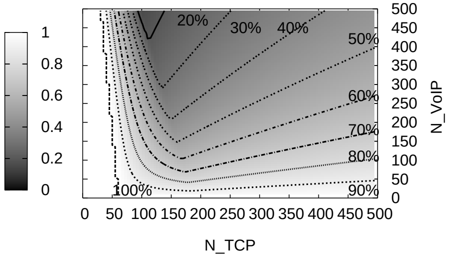
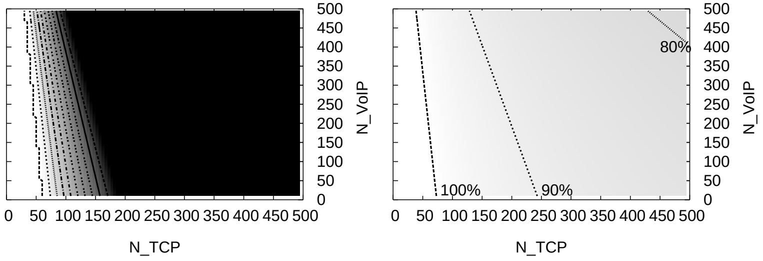


Figure 3: *Average utility as function of the number of flows in the system for 80 ms propagation delay.*



(a) VoIP utility (b) TCP utility

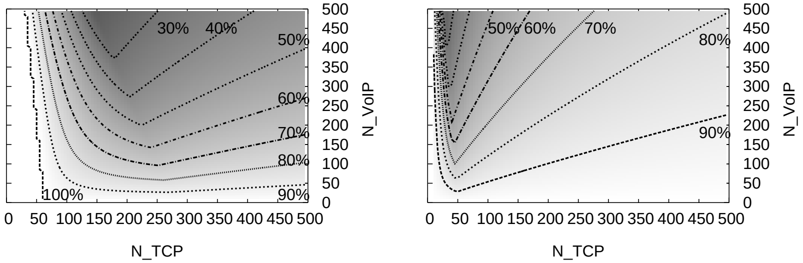
Figure 4: *Utilities as function of the number of flows in the system for 80 ms propagation delay.*

Figure 2 shows the bandwidth shares of the different applications under the assumption that VoIP flows get what they need and TCP shares the rest.

Case 1: Varying Number of Flows

Let bottleneck capacity be 100 Mbit/s and propagation delay 80 ms. We vary the number of TCP and VoIP flows independently from 10 to 500. In every case, the VoIP flows always get the bandwidth they require, with the rest shared by the TCP flows. In other words, bandwidth is fairly shared.

Figures 3 and 4 show the utilities of the flows. We show three graphs, average system utility in Figure 3, utility of VoIP in Figure 4(a), and utility of TCP in Figure 4(b). X-axis shows the number of TCP flows and y-axis shows the number of VoIP flows. The color determines the utility where lighter colors represent larger utilities. We also show the contours for several different levels of utility in all the figures.



(a) 20 ms delay (b) 140 ms delay

Figure 5: *Utilities as function of the propagation delay.*

The average utility in Figure 3 shows that when the number of TCP flows is small, the system is at its optimal utility, regardless of the number of VoIP flows. As the number of TCP flows increases, utility of VoIP flows drops rapidly (Figure 4(a)) which leads to a drop in average utility. If the number of VoIP flows is small, then the system is dominated by TCP flows, thus the utility remains high. An interesting observation is the sharp corner on most of the contour lines (in particular 30% and 40% lines). This represents the number of TCP flows where the utility of VoIP flows drops to zero and to the right of that corner, only TCP flows contribute to the average utility.

Figures 4(a) and 4(b) show the utilities for VoIP and TCP individually. The utility of VoIP flows drops rapidly as the number of TCP flows goes beyond 50 and drops to zero when the number of TCP flows has exceeded 150–250, depending on the case. TCP flows, on the other hand, do not suffer much under any circumstances and remain above 80% utility throughout, as shown in Figure 4(b).

Case 2: Varying Delay

In the second case, we pick a point from Figure 4 and vary the propagation delay on the link. We varied the propagation delay from 10 ms to 300 ms and report the results for 20 ms and 140 ms. In all the cases, bandwidth was fairly shared. The results are shown in Figure 5. Figure 5(a) shows a very similar situation as in Figure 3 with 80 ms propagation delay. Both flows are able to get decent utilities in some parts of the graph, but as the number of TCP flows increases, VoIP begins to deteriorate. The corners show when VoIP utility drops to 0. The individual utilities were similar to Figure 4.

The case of 140 ms propagation delay in Figure 5(b) is more interesting. Recall that the utility of a VoIP application drops to 0 at 150 ms one-way delay. Given a propagation delay of 140 ms, even a small amount of queueing delay caused by the TCP flows is sufficient to drop VoIP's utility to 0. This is demonstrated by the sharp corners on the contours which, as before, indicate the point where VoIP utility becomes 0.

The reason for VoIP's low utility is different in different scenarios. When propagation delay is small, the loss of utility is due to loss, and when delay is large, the delay is the main cause of loss of utility.

Figures 4 and 5 clearly show that a fair bandwidth allocation by no means implies that the application-level performances are in any way fair. In other words, even though bandwidth is fairly shared, the mechanisms which ensure this fair sharing are such that the damaging parameters can be pushed out of their feasible ranges, thus rendering applications useless. Our results clearly point out that fair bandwidth sharing is *not sufficient* to call the system fair and they underline a pressing need for further research into the topic.

5 Utilities in Practice: Example

Ideally, network-level components, e.g., routers, would determine the utilities for different flows and manage them appropriately. Although such an in-network control is likely to yield much more effective results, we recognize the inherent difficulties in terms of a practical implementation.

As a complementary solution, we now present how utilities can be used at the application level to improve user satisfaction. As our example shows, individual applications can also adapt their own behavior to optimize their own performance.

Consider a VoIP stream with encoding rate r_1 which is also the transmission rate. Let p_1 be the loss fraction that this stream sees. The rate r_1 determines the bandwidth utility, which we denote with u_{x_1} and the loss causes a damage utility of u_{p_1}.[2] The aggregate utility is given by $U_1(u_{x_1}, u_{p_1})$

The VoIP application could decide to change to a lower encoding rate $r_2 < r_1$ and add forward error correction (FEC) to protect the stream against loss. Let's assume this would bring the transmission rate to r_1, i.e., stream with lower encoding rate plus FEC would have the same transmission rate as the original stream without FEC. However, although the network level loss would still be the same, FEC would bring the *application level loss* to $p_2 < p_1$, i.e., the application would see less loss. The application would then see an overall utility of $U_2(u_{x_2}, u_{p_2}))$ and if $U_2 > U_1$, the application should switch over to the lower rate encoding with FEC transmission.

All the information needed to evaluate the situation are already available to the application. The application knows its transmission rate and can, through feedback, know the loss rate. Furthermore, the application can be programmed with information about user satisfaction at different encoding rates and the effectiveness of FEC schemes. The computation about whether a change is warranted is then straight forward.

[2] For simplicity, we assume that delay and jitter do not cause any damage.

6 Conclusion

We have argued against the traditional concept of achieving fairness between network applications through a fair sharing of network bandwidth. Instead, we have claimed that to maximize the satisfaction of users of network applications, we must take into consideration four factors: bandwidth, delay, jitter, and loss. With these four factors, we are able to characterize network applications and determine how network resources should be distributed to maximize the aggregate utility. We have shown that the parameters required by our utility model already exist for most Internet applications. Finally, we have presented an analytical evaluation of our utility functions, which clearly demonstrates that fair bandwidth sharing is not sufficient to ensure that all applications are fairly treated. As our results show, fair bandwidth sharing does not guarantee that the damaging parameters are within a useful range.

In our future work we plan to perform an in-depth study of the effects of utilities on user-perceived quality. We will extend our model and augment the results with simulations and experiments in real-world networks.

Acknowledgement

M. Mu's research is partly funded by EU Network of Excellence CONTENT.

Bibliography

[1] S. Shenker, "Fundamental Design Issues for the Future Internet," *IEEE/ACM Journal on Selected Areas in Communications*, vol. 13, no. 7, pp. 1176–1188, Sept. 1995.

[2] F. Kelly, "Charging and rate control for elastic traffic," in *European Transactions on Telecommunications*, vol. 8, Jan. 1997, pp. 33–37.

[3] S. H. Low and D. E. Lapsley, "Optimization flow control — I: basic algorithm and convergence," *IEEE/ACM Transactions on Networking*, vol. 7, no. 6, pp. 861–874, 1999.

[4] L. Massoulié and J. Roberts, "Bandwidth sharing: objectives and algorithms," in *IEEE Infocom*, Mar. 1999, pp. 1395–1403.

[5] M. Vojnovic, J.-Y. L. Boudec, and C. Boutremans, "Global fairness of additive-increase and multiplicative-decrease with heterogeneous round-trip times," in *IEEE Infocom*, Mar. 2000, pp. 1303–1312.

[6] S. Deb and R. Srikant, "Congestion Control for Fair Resource Allocation in Networks With Multicast Flows," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 274–285, 2004.

[7] C. Boutremans and J. Y. Le Boudec, "Adaptive delay aware error control for internet telephony," in *IP Telephony Workshop*, 2001.

[8] ITU, "G.107: The E-Model: A computational model for use in transmission planning," ITU," ITU Recommendation, 2003.

[9] ——, "G.114: One-way transmission time," ITU," ITU Recommendation, 1996.

[10] N. Kitawaki and K. Itoh, "Pure Delay Effects on Speech Quality in Telecommunications," *IEEE/ACM Journal on Selected Areas in Communications*, vol. 9, no. 4, pp. 586–593, May 1991.

[11] S. Vegesna, *IP Quality of Service*. Cisco Press, 2001.

[12] P. P. Francis-Cobley and A. D. Coward, "Voice over IP versus voice over frame relay," *International Journal of Network Management*, vol. 14, no. 4, pp. 223–230, 2004.

[13] ITU, "G.1010: End-user multimedia QoS categories-11/01," ITU," ITU Recommendation, 2001.

[14] T. Rahrer, R. Fiandra, and S. Wright, "Triple-play services quality of experience (QoE) requirements and mechanisms," DSL Forum," Draft working text, Feb. 2006.

[15] T. Szigeti and C. Hattingh, *End-to-End QoS Network Design: Quality of Service in LANs, WANs, and VPNs*. Cisco Press, 2004.

[16] J. Ramsay, A. Barbesi, and J. Preece, "Psychological investigation of long retrieval times on the world wide web," *Interacting with Computers*, vol. 10, no. 1, pp. 77–86, Mar. 1998.

[17] F. F.-H. Nah, "A study on tolerable waiting time: how long are Web users willing to wait?" *Behaviour & Information Technology*, vol. 23, no. 3, pp. 153–154, May 2004.

[18] A. Bouch, A. Kuchinsky, and N. Bhatti, "Quality is in the eye of the beholder: Meeting users' requirements for Internet quality of service," in *CHI* 2000, pp. 297–304.

[19] P. Quax, et al., "Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game," in *NetGames workshop*, Aug. 2004, pp. 152–156.

[20] S. Zander and G. Armitage, "Empirically measuring the QoS sensitivity of interactive online game players," in *Australian Telecommunications Networks and Applications Conference*, Dec. 2004, pp. 511–518.

[21] G. Armitage and L. Stewart, "Limitations of using real-world, public servers to estimate jitter tolerance of first person shooter games," in *Advances in Computer Entertainment Technology*, June 2004.

[22] C. Boutremans and J. Y. Le Boudec, "Adaptive joint playout buffer and FEC adjustment for internet telephony," in *IEEE Infocom*, 2003.

[23] R. Ramjee, J. F. Kurose, D. F. Towsley, and H. Schulzrinne, "Adaptive playout mechanisms for packetized audio applications in wide-area networks," in *Proceedings of IEEE Infocom*, 1994, pp. 680–688.

[24] M. Claypool and J. Tanner, “The effects of jitter on the perceptual quality of video,” in *ACM Multimedia*, Orlando, FL, Oct. 1999, pp. 115–118.

[25] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, “Modeling TCP throughput: a simple model and its empirical validation,” in *ACM SIGCOMM*, Sept. 1998, pp. 303–314.

The usage of text specifications for program code generation

Prof. Igor L. Brathikov, Dr. Ilya S. Lebedev

Saint-Petersburg state university

Lenin St., 7, St.Peterburg, 199034, Russia

E-mail: braigor@yandex.ru, i.lebedev@econ.pu.ru

Abstract

One of the first problems standing in the way of construction of information systems consists in processing the text of a technical project on working out a result suitable for the analysis. After that we have a set of natural language specifications. This paper deals with the problem of the automatic processing of such text information.

Contents

1 Introduction

Software engineering is quite an intensive process on which the quality of systematization and the efficiency of development depend. The development of modern software assumes the presence of a large quantity of text documentation (requirement specification, technical requirements) which should be converted into texts of specifications depending on the project methodology[1]. These texts are necessary for the elimination of ambiguity in the interpretation of Software parameters, the validation of structure, the construction of schemes and diagrams of links of objects, the solution of questions of testing. The representation of texts specifications also is carried out according to rules (IDO, IDF-diagrams). However specifications represent the arranged sentences in a natural language. For example:

1. The Application should provide the user description test before Camera Test execution.

2. During the Camera Test the Application should prompt the user to capture the image.

3. The user should be able to confirm or reject the result of the Camera Test in response to the test question.

The first scheme is as follows:

Precond: if Camera Test was not executed
User:: Param: Display
App:: Param: Test description, Camera Test
Action: Provide Param: Test description
Postcond: Display - Test description

2 The concept of processing

Here we have a problem of automatic processing of such text information. For this purpose we can use the NL processor based on adapted model NL by Professor V.A.Tuzov. The processing of information by this processor includes three levels: morphological, syntactic and semantic.

The processor can analyze the construction of a natural language not only to external tags, but also on the basis of the information on links between construction words. Under the sentence from item 1 of the abovestated text fhe information in Table 1 will be received.

From such a representation of information it is not only possible to use the structure, but also the system of semantic-morphological markers on the basis of which the rules of further processing of the received structures can be created. This processor is used in information retrieval systems for the compilation of summaries of engineering specifications, help Software systems, and speech synthesis systems. We can state that the processor has been proved quite successfully for Russian language systems, producing from 65 to 95 of adequate structures, depending on subjects and the complexity of the text. Setting processing rules of structures, it is possible to transform the text of each specification to the following form:

Table 1: *Sentence from item 1.*

word	structure	type
008	shall provide	[action]
002	before	[when?]
003	Test	[what?]
005	Camera	[what?]
005	execution	[what?]
006	The Application	[what?]
009	the user	[whom?]
010	description	[what?]
012	test	[what?]

Condition - Object - Operation

Therefore the following basic elements of structure can be selected:

1. The specification object is the subject of the specification sentences. (For Russian it is a Noun in the Nominative case together with the noun phrase). For the analysis in the structure of the specification the object is any noun.

2. Operation is a verb with the verb phrase. The verb group is defined by the structure of the sentence.

3. The condition is selected by identifier words and links (if, then, before, during). Having created such a system of descriptions, it becomes possible to transform our sentences of specifications to the following form:

Condition: before Camera Test execution

Action: The Application shall provide the user with the test description

Object: Application

Such a representation gives us information for the analysis of the scheme block described by the specification. Now it is possible to convert the specification text to the code template. And the form of the template is completely dependent on a programming paradigm. For the object-oriented approach it is necessary to select the specification object. Each object has a set of properties and methods.

Let's consider the selected object of the specification sentence. As well as any object, "Application" has properties and methods. On the one hand, properties and methods can be considered as modular procedures with standard sets of parameters in various programming tools (for example, VC ++, BCBuilder ++). On the other hand, this object can be constructed while "reading" specifications. In this case it is possible to select the interconnected parameters. For example, the analysis of parameters of the object "Application" under the specifications allows to receive the following description:

OBJECT: Application
PARAMETR-CONDITION1 : Before Camera Test execution;
PARAMETR-CONDITION 2 : During the Camera Test;
ACTION1: shall provide the user with the test description
ACTION2: shall prompt the user to capture the image

Having designated operations as procedures and parameters variables, it is possible to generate the following class:

```
class App
{
bool par1; // PARAMETR-CONDITION1 : Before Camera Test execution;
bool par2; // PARAMETR-CONDITION1 : During Camera Test execution;
obj action1(); // ACTION1: shall provide the user with the test description
obj action2(); // ACTION2: shall prompt the user to capture the image
};
```

Usually parameters and methods strongly depend on a subject domain and the development experience. Considering structures of noun phrases and superimposing identification rules, it is possible to make distinctions between such objects, as Test (on the camera) and Test (on working capacity) and to determine parameters in methods.

```
class App
{
bool par1; // PARAMETR-CONDITION1 : Before Camera Test execution;
bool par2; // PARAMETR-CONDITION1 : During Camera Test execution;
obj action1(bool par1);//ACTION1: shall provide the user with the test description
```

```
obj action2(bool par2); // ACTION2: shall prompt the user to capture the image
};
```

Another point is the definition of links between objects. The presented structures of specifications allow receiving representation of links automatically. For example, selecting the main parts of each specification in operation:

Application – user – the description (the test)
Application – user – hint

The objects set by conditions define the external relations between objects of specifications. On the basis of a state of their properties, a solution on the choice of a given specification, i.e. on its fulfilment is made.

```
IF (par1==true)
App.action1(par1);
```

Such approach inherently uses the mechanism of events given by resources of visual programming. There is a form, have created the object "button" on it and on event of pressing "button" its code is generated. In a resulted example the operations are described by the specifications containing the information objects, their methods and events. The specification sentence sets some function with its entry output parameters in the form of objects and their properties and a condition of execution of this function. Usually texts of specifications consist of arranged sentences. Each specification describes one logically completed function. One variant of an automatically created frame of an application code depending on those conditions is given below.

```
App.initialization();
While(App.paramn== true)
Switch(App.condition)
{
case 1: App.action1(par1); break;
case 2: App.action2(par2); break;
...
case N: App.actionN(parN); break;
default :
};
```

Table 2: *Conditions and actions.*

word	structure	type
001	If	[condition]
003	has pressed	[action]
002	the user	[who?]
004	the button	[what?]
007	to display	[action]
008	the message	[what?]

In this example the initialization block of the application is executed first. Then the application is included into a cycle of the current state analysis where methods become dependent on the current conditions.

3 Calculation of the structure

The following question is conversion of the elementary notation to a code. Let the text contains the following sentence:

If the user has pressed the button to display the message.

We calculate its structure in Table 2

Then it is necessary to calculate the condition and action:

Condition: If the user has pressed the button

Action: to display the map

Object: "mssage"

Now we should define the information required for the code generation. For this purpose we select objects in the given specification on the basis of noun phrases:

the Message
the User
the Screen
the Button

Similarly to CASE-means the information on these objects and their typification should be stored in such a system. A dictionary of the subject domain of the application which in the most minimum case enumerates objects synonyms and a set of object types is necessary. Usually try to avoid synonymous names of objects in a specifications text, such as in Table 3.

Thus taking such a text the system should generate a code of the description of object in the heading unit as:

Table 3: *Object identification*

Type	Text synonym
TypeMessage *OBJMessage	Message,signal...
TypeUser *OBJUser	User...
TypeConsol *OBJConsol	Screen,Console,Terminal...
TypeButton *OBJButton	Button...

Table 4: *Function identification*

function	Text synonym
TipeCondApp Press(TipeButton *, TipeObj *)	To press, activate etc.
TipeCondApp Print(Tipe Message *, TipeObj *)	to output, print out etc.

```
TipeMessage *OBJMessage1;
TipeUser *OBJUser1;
TipeConsol *OBJConsol1;
```

Functions described by verb operations can be considered similarly. In a resulting example we can see verbs from the dictionary of the subject domain. It must have synonymous concepts as well. Some examples you can see in Table 4. Operations should be identified by functions, as one of the options, returning the result treated as a state of the application.

Like CASE-means where the button is pressed at an application creation on the form and the programmer should write a code on the occurring event, given functions also can be opened for further filling by a code and the possibility of their detailing. However in case of sample formulation of notations and sufficient contents the complete calculation of Input data of such functions is theoretically probable. After detection of such identifiers we build in file identifiers of functions with an empty body which should be filled:

```
TipeCondApp Press(TipeButton *OBJButton, TipeUser *OBJUser )
{
};
TipeCondApp Print(TipeMessage *OBJMessage, TipeConsol *OBJConsol)
{
};
```

Let us come back to the structure *condition - operation*. We identify a condition by the *IF* statement, the verb operation "has pressed" by the *function Press()* with the arguments the user and the button. The operation "output" is defined with the *function Print()* with the arguments the message and the screen. Then it is possible to generate the following code in the *function Main ()*:

```
void Main(void)
{
...
IF(Press(OBJButton, OBJUser ))
 Print(OBJMessage, OBJConsol);
...
};
```

4 Conclusion

In conclusion it is important to note that the basic development of programming technologies is set by methods of the automatic processing of information and knowledge. Using the described approach it is possible to calculate objects of the program and communications between them.

Bibliography

[1] V. A. Tuzov, *Computational Semantics of Russian.* Saint-Peterburg State University, St.Petersburg, 2004. p.400

Caching in Multi-Processor Systems

Dr. Tiina Niklander

Department of Computer Science, University of Helsinki

P.O.Box 68, FI-00014 University of Helsinki, Finland

E-mail: `tiina.niklander@cs.helsinki.fi`

Abstract

This paper gives a short introduction to the current research of caching in multicore systems. The approaches modify either the architecture or the software. On the architecture level, the focus is on non-uniform cache architectures (NUCA) where the core's access time to the particular cache item depends on the item's location in the cache. In these architectures, there are still unclear issues about an optimum cache architecture. Current research focuses on methods that combine the private cache and shared cache approaches together. The software-based approaches focus on the interaction between the running application and the cache system. The key question here is, how the application can control or help the cache mechanism to give the best support to this particular application.

Contents

1 Introduction

The improvement in computer performance is currently achieved by adding more processors to the system. One single die today has plenty of components and is able to host multiple cores or processors. The novelty of this approach partially shows in the names like multicore, chip multiprocessor, etc. The cores can share some units on the chip or be totally separated and just located on the same chip. They can be similar general-purpose processing units or each have a different purpose, like multimedia cells, integrated GPU, etc. These on-chip-processors can also share communication ports and any elements available on the chip, including caches on all levels.

With increased number of cores on the chip, the cache organization within the chip becomes more important. Several alternatives for private and shared cache organizations have been proposed and evaluated in the literature. Common organization is that the L1 caches are private and the L3 cache is shared, but the interesting issue is the L2 cache. There multiple alternatives have been proposed.

Sharing the cache by all cores makes the cache coherence issue easier, since there is just one copy of the memory content in the cache. However, increasing the cache size means that some parts of the cache are further away creating different access latencies for different cache lines have different latencies in the core access. This eventually leads to the non-uniform cache architecture (NUCA). An addition issue with multiple cores sharing one cache is the cache access competition. The cores may occasionally need to wait for their access turn, while another core is accessing the shared cache.

Splitting the cache area to private caches reduces the access latencies and their variation. It also removes the access competition, but introduces the problem of cache coherence. Now multiple private caches may contain the same cache line and the write by any core must invalidate the line on other caches.

The issue of shared versus private cache organization is not just black and white. With several cores on the chip, it is possible to organize the cache is such a way that a pair of cores share cache, but other core pairs are not allowed to access this cache.

Current trend is to try to find a way to combine shared and private structures in the same cache architecture. The cache could then dynamically adapt to different needs for shared and private areas. An interesting approach is the cooperative caching, where a directory structure is added to handle the coherence issues and also to avoid necessary memory access, if the data is on another core's cache already.

2 Cache organization on multicore chip

The cache is an important part of today's computer architecture. There is not just one cache on the way from core to the memory. Common organization has three cache levels. Cache's main purpose is to keep the needed data as close to the processor as possible, because the speed of memory access is very slow compared with the processor speed. Also the off-chip bandwidth is limited. In single processor the cache is always private. With multiple cores and processors each cache level can be private or shared.

The L1 cache is closest to the cores and it is private for each core. The L1 cache speed must match the speed of the core itself. This restricts the size of the cache. The L1 cache access time must be the same for all cache lines.

The L2 is commonly also on the die. It is large and occupies a big part of the die's area. The organization of L2 cache has several alternatives. To make private L2 caches, it can be architecturally split to the processors and each part is located in the floor-plan close to its owner. This way each processor will be able to access all of its allocated cache lines with similar latencies. To form a shared L2 cache we need to use NUCA, which allows different cache lines to experience different latencies, depending on their distance from the core that issues the access request [6]. This means that the parts of the shared cache close to a particular processor are faster to access and should be preferred, while other parts suit better to the processors closer to them.

It seems to be still an open issue how the cache should be organized when considering the issue of private cache versus shared cache organization. Neither one is the optimum for all cases. The key issue is that in shared cache there is only copy, but there might be access delays. In private case the access time is more or less static, but the coherence of the multiple copies becomes an issue. A table 1 from [8] collects the benefits and drawbacks of these approaches in a static case. A study about L3 level cache [8] shows that the selection between shared and private caches depends on the workload and the access bandwidth to the L3 cache lines. During heavy load the private cache is better while in general the shared cache seems to offer better access.

The balancing between private or shared cache can be done dynamically. The cache banks can be organized from the private or shared view-point and either be collected together or distributed close to the cores. Even in the physically distributed banks it is possible to give all cores access to all cache banks. For the performance of each core, it is important to locate the requested cache lines as close to them as possible to reduce the access latency.

The cache can be split to clusters [6] and each cache line can be originally placed to the cluster closest to the processor asking for it. Since the cache is shared there is just one copy of each cache line and a new processor may be requesting an item already in the cache, but in a distant cluster. The requested item once it has been located can then be migrated gradually closer to the new requesting processor. The issue of where to place the cache item frequently requested by multiple processors is discussed in the paper [6]. Their goal is to avoid unnecessary migrations, so called ping-pong effect, with these items.

It is also possible to start with assuming that the processor closest to each cluster has some ownership to the cache or at least to part of it. This approach has been discussed in [3]. The cache is split for private and shared part. The split to these parts is adaptively adjusted to minimize the total miss rate. Each core search first its private cache and if not found there, then the shared cache of all other cores is searched parallel. To reduce access time, the found item is moved to the local private cache and one item is evicted to its place in the shared cache. The article does not consider consistency issues or competition of cache items by multiple cores. The architecture requires the usage of sharing engine to manage the cache sharing, especially to control the replacements of the cache items and to calculate the optimum split for each core's cache.

The private caches can be also made shared using a cooperative caching approach [1]. In the cooperative caching local active data is kept in the private cache of the particular core and the globally active date used by multiple cores, or needed by core whose own L2 private cache is full, is stored in any of the remote ≪private ≫L2 caches to be used by any core. The cooperative cache architecture has a global directory to locate the remotely stored data items from other core's caches. This way an off-chip cache miss can be avoided, but the data still need cache-to-cache transfer to be available for the requesting core. The original cooperative caching has been modified to support partitioning [2] and to allow distributed control for the shared data [4].

Table 1: *Benefits and drawbacks of cache sharing [8]*

Characteristic	Shared caching	Private caching
Area	Reduce	Increase
Hardware duplication	Reduce	Increase
Block replication	Reduce	Increase
Cache pool availability to one core	Increase	Reduce
Coherence traffic and interconnect contention	Reduce	Increase
Miss rate	Reduce	Increase
Utilization	Increase	Reduce
Coherence hardware cost	Reduce	Increase
Inter-core communication	Reduce	Increase
Ease of powering off subcomponents	Increase	Reduce
True and false sharing costs	Reduce	Increase
Variation of impact on threads' performance	Increase	Reduce
Hit time latency	Increase	Reduce
Implementation complexity	Increase	Reduce
Performance isolation	Reduce	Increase
Interconnect hardware requirements	Increase	Reduce
Next upper level cache coherence costs	Increase	Reduce
Prefetcher complexity	Increase	Reduce

3 Software-based improvements

It is clear that the improvements in the caching mechanism are helpful only to a certain point. The research shows clearly that different applications would benefit from different improvements or different cache behavior. Since the caching mechanism can only guess how this particular application works, it cannot always select the best functionality for a particular application. This makes it necessary to do improvements directly in the application software also.

To improve cache usage the compiler can insert data prefetching instructions to the program code. The prefetch-instruction just copies the data from memory to cache just-in-time so that the actual instruction can be executed faster. For prefetching both instructions and data to the cache for a single-threaded application in a multicore environment, it is also possible to add a special ≪helper≫thread to perform in advance the memory accesses which are most likely to cause cache-miss.

The prefetching from memory to cache in its best will remove all potential cache-misses and will allow the execution of the application thread to proceed smoother and faster. However, it requires extra computing resources, since there are in execution either extra instructions in the program itself on the same core or the additional, helper thread on a separate core. For the overall system performance point of view, the application with a helper thread now needs two cores for its execution, while without the helper thread it could be executed on one core only. This approach is feasible in a system which has extra resources to spare for this purpose.

The prefetching can be also harmful, if the prefetched memory locations are needed later than the cache content they replaced [7]. The code with added prefetch instructions that works well in the uniprocessor environment may not do so in the multicore system. In the multicore system with shared cache, the prefetches may replace data that is needed by some other program before this program needs the prefetched data. To avoid this behavior the system could [7] be based on the history information to pin the needed data to cache and to suppress the prefetching for a certain duration.

While the prefetching improves the performance of a single thread, a thread-based control for moving shared data elements from cache level to another could improve the performance of a tightly-synchronized multithreaded application. Giving threads the control possibility to evict cache elements from the private cache levels to a shared cache level for faster access by other threads running concurrently on other cores can improve the performance. This Software Controlled Eviction [5] needs modifications to the program code. A new eviction-command needs to be added to the application code. With this command a thread can give hint to the cache mechanism that this data element could be moved to a shared cache for the other threads. These eviction hints can be added by the compiler for example after the last data modification in iteration.

4 Conclusion

The cache management in multicore systems is still under a very active research. The focus of the research is on the last level of on-chip cache independent of whether it is L2 or L3. A lot of alternative mechanisms have been created to improve the cache performance. Common to all of them is added functionality to the cache management. The architecture-based approaches do not consider the functionality of the application. They try to focus on minimizing the cache-miss and in the NUCA also the cache-access delays. The software-based approaches try to find simple and efficient mechanisms for the running applications to control and improve the cache usage. They assume that the software itself knows best the actual cache-needs of a particular thread. They require the compilers to generate the needed changes to the executable instructions.

Bibliography

[1] Chang, J. and Sohi, G. S. *Cooperative Caching for Chip Multiprocessors.* In Proceedings of the 33rd Annual international Symposium on Computer Architecture. IEEE Computer Society, 2006. pp 264–276.

[2] Chang, J. and Sohi, G. S. *Cooperative cache partitioning for chip multiprocessors.* In Proceedings of the 21st Annual international Conference on Supercomputing. ACM, 2007.

[3] Dybdahl, H. and Stenstrom, P. *An Adaptive Shared/Private NUCA Cache Partitioning Scheme for Chip Multiprocessors.* In Proceedings of the 2007 IEEE 13th international Symposium on High Performance Computer Architecture. IEEE Computer Society, 2007.

[4] Herrero, E., GonzAlez, J., and Canal, R. *Distributed cooperative caching.* In Proceedings of the 17th international Conference on Parallel Architectures and Compilation Techniques. ACM, 2008. pp. 134–143.

[5] Fide, S. and Jenks, S. *Proactive Use of Shared L3 Caches to Enhance Cache Communications in Multi-Core Processors* IEEE Computer Architecture Letters 7, 2 (July-Dec. 2008), pp. 57–60.

[6] Kandemir, M., Li, F., Irwin, M. J., and Son, S. W. *A novel migration-based NUCA design for chip multiprocessors.* In Proceedings of the 2008 ACM/IEEE Conference on Supercomputing. IEEE Press, 2008.

[7] Kandemir, M. Zhang, Y. and Ozturk, O. *Adaptive prefetching for shared cache based chip multiprocessors* In Proceedings of Design, Automation and Test in Europe. EDAA, 2009. pp. 773–778.

[8] Sibai, F. N. *On the performance benefits of sharing and privatizing second and third-level cache memories in homogeneous multi-core architectures.* Microprocessors and Microsystems 32, 7 (Oct. 2008), pp. 405–412.

Exploring a Framework for Advanced Electronic Business Transactions

Dr. Alexander Norta

Department of Computer Science, University of Helsinki

P.O. Box 68, FI-00014 University of Helsinki, Finland

E-mail: alexander.norta@cs.helsinki.fi

Abstract

With the emergence of service-oriented computing technology, companies embrace new ways of carrying out business transactions electronically. Since the parties involved in an electronic business transaction (eBT) manage a heterogeneous information-systems infrastructure within their organizational domains, the collaboration complexity is considerable and safeguarding an inter-organizational collaboration with an eBT is difficult, but of high significance. This paper describes a conceptual framework that pays attention to the complexities of an eBT and its differentiating characteristics that go further than traditional database transactions. Since the eBT is a framework that comprises separate levels, pre-existing transaction concepts are explored for populating the respective levels. To show the feasibility of the described eBT framework, industry initiatives that are aspiring to become business-transaction standards, are checked for eBT compatible characteristics. Since realizing an eBT framework raises many tricky issues, the paper maps out important research areas that require scientific attention. Essentially, it is required to investigate how the business semantics influences the nature of an eBT throughout its lifecycle.

Contents

1 Introduction

The emergence of electronic business promises for companies a sustainable market advantage that comprises an integration and coordination of information flow and product flow between heterogeneous information-system infrastructures. Such information flow that bridges different organisations, includes the linking of business elements into an integrated whole. For such electronic business collaboration (eBC), a loose coupling of those information systems is a requirement as a tight coupling of information systems results in too many agreement details and too much shared context has to be revealed that to the business counterpart.

In eBC, the registration of business transactions is of major legal importance for organisations. A business transaction [1] is a consistent change in the state of a business relationship that is driven by a well-defined business function. Each party in a business transaction holds its own business transaction. For eBC, a transaction concept is important to ensure reliability. To facilitate a loose coupling and highly dynamic establishment of business collaboration, the service oriented computing (SOC) paradigm is increasingly important. Services are self-describing, logical manifestations of physical resources that are grouped as a process, i.e., as a set of actions [50] that an organisation is prepared to execute and expose to the web.

With the complexity involved in eBC, no single transaction model is able to meet all requirements. Instead it is necessary to inter-organizationally establish transaction frameworks in a way that does not force companies into disclosing an undesirable amount of business internals. In this paper a conceptual model of an electronic business transaction (eBT) is put forward, based on an investigation of features that also incorporates business aspects and in which collaborating organizations can safeguard their business internals. Note that an eBT needs to safeguard the legally binding contractual relationships between collaborating parties that dictate responsibilities and the consequences of behaviour. The importance of business semantics in an electronic collaboration also has consequences for the nature of the atomicity, consistency, isolation, and durability properties of an eBT.

The remainder of this document is structured as follows. Section 2 discuses the challenges and specific characteristics of business transactions, followed by the description of an electronic business transaction framework that takes the previously mentioned characteristics into account. Section 3 shows which pre-existing transaction concepts may be used for the earlier presented electronic business-transaction framework. Section 4 discusses industry initiatives for electronic business transactions and investigates to which extent they realize the framework contained in Section 2. Finally, Section 6 concludes this document and maps out future research direction.

2 A Framework for e-Business Transactions

The complexity of transactions that span multiple organisations rises in loosely coupled distributed computer networks that are enabled by SOC [40]. Here, business processes that use database systems must be inter-organizationally integrated. The transactional parts of a business process are referred to as a business transaction. By linking business processes inter-organizationally with electronic means, collaborating parties hope to safe time and money during the setup, enactment, and post-enactment of supply chains.

For eBC that takes place in a highly dynamic environment, applying conventional transaction mechanisms is not sufficient, as data from resources that are at the back-end of web services, would need to be locked in order to assure atomicity and isolation. However, locking data for isolation in long running transactions for eBC is unrealistic as this might block resources that are consequently not available for others. For example, locking tables for selling a product blocks other potential customers, results in lower turn-over, and prevents other organisations from participating in the business process.

The synchronization of business processes between organisations must be part of a wider business coordination protocol that defines the publicly agreed business interactions between business parties, which is based on web services. Additionally, well founded possibilities are missing to compose eBTs out of several transaction models [51] to support long running transactions for heterogeneous systems that are integrated in a loosely coupled fashion.

2.1 Features of electronic business transactions

An eBT is a consistent change in the state of the business that is carried out with electronic means and that is driven by a well-defined business function. An eBT is automated, complex, long running and may involve multiple internal and external parties. Additionally, an eBT requires commitments to the transaction that needs to be negotiated by the participating organisations [36]. Further features of an eBT are: support for the formation of contracts, shipping and logistics, tracking, varied payment instruments and exception handling. Compared to traditional database transactions, eBTs have several distinguishing characteristics. Firstly, they extend the scope of traditional transaction processing as they may encompass classical transactions which they combine with non-transactional processes. Secondly, they group both classical transactions as well as non-transactional processes together into a unit of work that reflects the semantics and behavior of their underlying business task. Thirdly, they are governed by unconventional types of atomicity, e.g., payment atomicity, goods atomicity, delivery atomicity, contract atomicity. In [52] these unconventional atomicities are described in further detail.

Unconventional behavioral features [43] of an eBT are specified as follows: Generic characteristics tackle issues like who is involved in the transaction, what is being transacted, the destination of payment and delivery, the transaction time frame of permissible operations. Examples for special purpose characteristics are links to other transactions, receipts and acknowledgments, identification of money transferred outside national boundaries. Furthermore, advanced characteristics are the ability to support reversible and repaired transactions, the ability to reconcile and link transactions with other transactions, to specify contractual agreements, liabilities and dispute resolution policies, transactions that guarantee the integrity of information, confidentiality and non-repudiation; the ability for transactions to be monitored logged and recovered.

2.2 Managing electronic business transactions

The integrated heterogeneous systems of an eBC need to be loosely coupled because of different reliability requirements that exist within long running eBTs. Different reliability requirements result from the properties of an eBT such as the phase the transaction is in and the level in which the transaction is taking place.

In [41] a phased model is introduced that distinguishes between pre-transaction, main transaction and post-transaction phases in a collaborative business process. In [24] the need for a three level process framework is identified as companies are not willing to directly connect their legacy system. Enabling interoperability between systems of different organizations is not just a matter of coupling systems, as this introduces interoperability problems, like semantic differences, autonomy, non-disclosures, company secrets, etc. Presenting the backend systems applications as services, i.e., wrapping them as web services, is a valid solution for most of the problems mentioned.

Given the complex features of eBT as described in Section 2.2, the conceptual model of Figure 1 represents an integration of the separate solution concepts [24, 41] that permits a manageable separation of concerns. In Figure 1, the organizational domains of a service consumer and a service provider are bridged by an external level where companies inter-organizationally harmonize their business collaboration and, hence, their business transaction. Along a time line, the external-level phases of an eBT are visualized that need to be coordinated with the eBT phases on the conceptual level within an organization. Finally, the conceptual-level coordinates the legacy system of the internal level, e.g., ERP systems, workflow systems, database systems, and so on. The latter give technical feedback to the higher level to inform about the success or failure of a transaction. Likewise, the conceptual level releases coordination information to the external level for aligning an eBT with the domain of the collaborating counterpart.

To control an eBT as depicted in Figure 1, spheres of control are a useful vehicle [40] (Norta 2007) to demarcate process parts that are provided by a collaborating party. The theory of spheres of control [4] originates from the domain of traditional database transactions. So called workflow spheres [34] expand the transaction theory into the dynamic world of complex business processes. Those concepts are applied in [28] for analyzing atomicity criteria dependencies and atomicity spheres without relating the workflow concepts of highly dynamic inter-organization processes. In the work of [44] a substantial emphasis is put on the characteristic atomicity properties of e-business. These unconventional atomicities for spheres in electronic business transactions (eBT) are explored and related [41] to each other along the categories system-level atomicity,

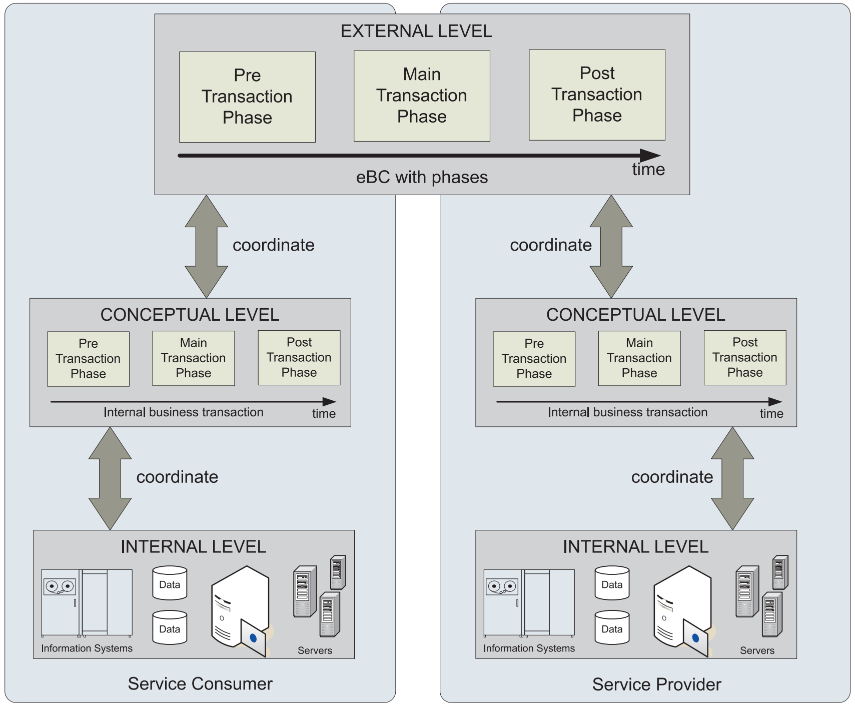


Figure 1: *The levels and phases of the eBT framework*

business-interaction atomicity, and operational-level atomicity. The unconventional atomicities need to be part of a transaction model that pays attention to the business realities that form the context of eBC. In [40] the concept called eSourcing uses spheres for enabling inter-organizational business collaboration.

Based on an elaborate historic survey [48] that covers the development of business transactions up to the latest developments in the domain of SOC, the following Section 3 investigates what existing transaction concepts are suitable for populating which level of the eBT framework depicted in Figure 1. Furthermore, Section 4 uses the same historic transaction survey to explore if the eBT framework is realized in industry efforts that aspire to become a standard for electronic business transactions.

3 Populating eBT Levels with Transactions

For the collaboration levels of Figure 1, pre-existing transaction concepts are available and assessed in this paper for eBT management on different levels of concern. The remainder of this section is structured as follows. First, Section 3.1 discusses transaction management for the internal level of an eBT, followed by Section 3.2 that comprises advanced transactions for the conceptual and external level. These advanced transaction concepts must be harmonized on the external level into transaction frameworks, which Section 3.3 covers.

3.1 Internal-Level Transactions

To traditionally manage data in a sound way, transactions must fulfil the following requirements. Atomicity states a transaction executes completely or not at all, consistency means a transaction preserves the internal consistency of the database, isolation infers a transaction executes as it were running alone with no other transactions, and finally, durability demands the results of a transaction are not be lost in a failure. These ACID properties of flat transactions are instrumental for handling exceptions in transaction management that are shielded from applications running on top of databases.

Flat transactions still dominate the database world because of their simple structures and they can be easily implemented Hence, ACID properties are suitable for the legacy systems of the internal level of an eBT as depicted in Figure 1. However, from a technical point of view, web-service composition faces the transactional challenges of relaxed atomicity, i.e., a situation where intermediate results may be kept without rollback despite the failure to complete the overall execution of a composite service.

3.2 Advanced Transactions for the Conceptual and External Level of an eBT

Advanced transaction models are extensions to flat transactions that release one or more ACID constraints to meet with specific requirements. Two strategies have been adopted for extension purpose to achieve different structures inside a transaction, namely the modularization of a complex transaction with hierarchies, and the decomposition of a long-lasting transaction into shorter sub-transactions. In the sequel the first type is referred to as distributed and nested transactions and the latter as chained transactions and Sagas.

A supporting concept for advanced transactions is the mechanism of savepoints [2] that enables a transaction rollback to an intermediate state for recovery. Savepoints are important for supporting recoveries in distributed and nested transactions. Furthermore, the use of checkpoints in transaction logs is instrumental in chained transactions to indicate a point until which a rollback does not result in an inconsistent database state. Details are explained in the sequel of this chapter.

Distributed and Nested Transactions

A *distributed transaction* is needed if an organization must integrate in a bottom-up way several database systems that reside on different servers in different geographic locations. As depicted in Figure 2, distributed transactions consist of sub-transactions that may access multiple local database systems and comprise two types of transactions [11], namely local transactions and global ones. Local transactions are executed under the control of the local database management system (DBMS), while the multi-database system (MDBS) is in charge of global transactions. Hence, local and global integrity constraints must be aligned. Also the atomicity and isolation is managed globally when the whole transaction is aborted if any sub-transaction fails. The most influential example of distributed transactions is the X/Open Distributed Transaction Processing (X/Open DTP) software architecture [52] that is a standard for the two-phase commit (2PC) protocol. In combination with ACID transactions, a multiphase protocol like 2PC is used to ensure sound database-state changes.

A *nested transaction* is a generalization of savepoints [23], which is suitable for complex-structured applications and adopts a top-down method to decompose a complex transaction into sub-transactions or child transactions according to their functionalities [38]. In nested transactions, parts of a transaction may fail without aborting the entire transaction. Sub-transactions are composed in a hierarchical manner and only the leaf sub-transactions perform database operations while others function as coordinators. As Figure 2 shows, a sub-transaction is atomic and when it aborts, the parent may trigger another sub-transaction as an alternative without necessarily violating the database consistency.

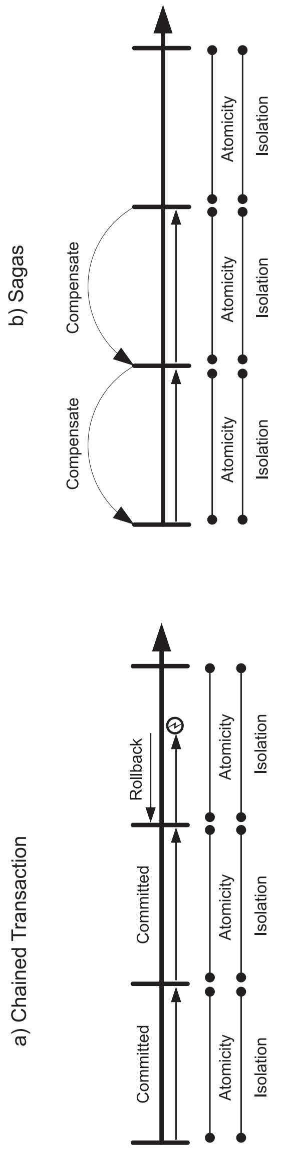


Table 1: *Simplified Informal Models of Distributed and Nested Transactions*

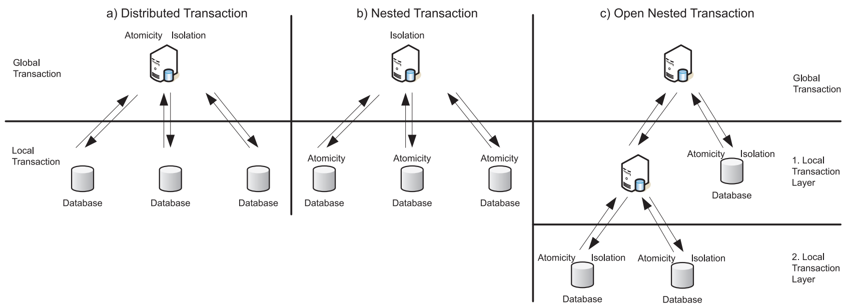


Figure 2: *Simplified Informal Models of a Chained Transaction and Saga*

Multilevel transactions are a variation of a nested transaction that are also called layered transactions [51] and their generalization is called open nested transactions [33]. In multilevel transactions a transaction tree has its layers corresponding to those of the underlying system architecture. Here, a pre-commit concept allows an early commitment of a sub-transaction before the root transaction actually commits, which requires a sub-transaction to semantically undo the committed one.

Multilevel transactions evolve to *open nested transactions* if the structure of the transaction tree is no longer restricted to layering, i.e., leaves in different layers are allowed (see Figure 2). Open nested transactions relax the ACID properties compared to achieve a higher level of concurrency. The latter guarantee global level isolation, which means in open nested transactions, the intermediate results of committed sub-transactions in nested transactions are invisible to other concurrently executing ones.

Chained Transactions and Sagass

Differently to a nested transaction and its extensions, a chained transaction is appropriate for a time-consuming application with long-lasting transaction processes. As depicted in Figure 3, a chained transaction is a variation of savepoints [23], where long running transactions are decomposed into small, sequentially executing sub-transactions that roughly correspond to savepoint intervals.

The difference is that each sub-transaction is atomic, while each interval between every two save points is part of an atomic transaction. In the chain, a sub-transaction triggers the next upon commit until the whole chained transactions commit. When encountering a failure, the previously committed sub-transactions have already durably changed the database so that only the results of the currently executing sub-transaction are lost. This way the rollback only returns the system to the beginning of the most recently-executing sub-transaction.

Since the atomicity and isolation properties are relaxed in a chained transaction, this leads to aborting problems of the whole chain in the middle of execution as all the committed sub-transactions cannot be undone, other concurrent transactions see intermediate results generated during the execution of the chain.

Sagas [17, 22] adopt the idea of chained transactions of including a compensation mechanism to roll back. As shows Figure 3, Sagas divide a long lasting transaction into sequentially executed atomic sub-transactions with ACID properties and each sub-transaction, except the last one, has its own compensating sub-transaction. When any failure arises, the committed sub-transactions are undone by compensating sub-transactions. Unlike chained transactions, Sagas can return the whole transaction back to the very beginning with compensations.

3.3 Transaction Frameworks

In some transaction frameworks several transaction models are integrated, which is a prerequisite for external-level business-transaction harmonization. This section briefly introduces the two relevant frameworks, namely ACTA and BTF.

The comprehensive framework called ACTA [16] unifies existing models to capture the semantics and reason about the concurrency and recovery properties of complex transactions. More elaborate extensions to this ACTA model are described in [17–19]. In the ACTA framework, interactions among transactions are expressed in terms of effects, i.e., effects of transactions on other transactions and effects of transactions on objects they access. ACTA capture the effects of transactions on objects by two objects sets and the concept of delegation. Every transaction is associated with several other objects that are contained in a view set or access set. The view set contains all the objects potentially accessible to the transaction while the access set contains the objects that have already been accessed by the transaction. Based on ACTA, the ASSET transaction model [3] uses primitives at a programming language level based on ACTA building blocks such as history, delegation, dependency, conflict set, etc.

A more applicable approach than ACTA for the eBT framework of Figure 1, is explored in the XTC (eXecution of Transactional Contracted electronic services) project [49] that aims at laying a generic foundation to the transactional support for business processes in a service-oriented environment. Hence, a Business Transaction Framework (BTF) is developed to support contract-driven, inter-organizational business processes. A BTF is a transaction hierarchy composed of so-called Abstract Transaction Construct (ATC) that comprises existing transaction models, which are stored in a library. The architecture of the BTF is a multi-level, multi-phase design [31]. Three phases exist along the BTF life-cycle, namely, definition phase, composition phase and execution phase. During the definition phase, the ATCs are abstracted from the classic and adopt the transaction models based on a taxonomy, which covers and classifies the existing work in transaction management domain. After the design of an ATC library, these contained constructs are used to build a transaction plan for a complex process within the composition phase. The abstract plans resulting from the composition phase [31, 47] are instantiated to form real business transactions for the execution phase.

4 Industry Initiatives for e-Business Transactions

For service-oriented eBC applications, supporting technologies and standards are needed to guarantee consistency and reliability in eBTs. While no transaction mechanism is widely accepted as a standard, there are three possible candidates, which realize the eBT framework of Section 2.2 to different degrees.

4.1 Business Transaction Protocol

The XML-based business transaction protocol BTP [15] is not exclusively designed for web services. BTP is instrumental for representing and seamlessly managing complex, multi-step business-to-business (B2B) transactions over the Internet to ensure consistent outcomes of parties that use applications disparate in time, location and administration, and that participate in long running business transactions [35].

In a BTP compliant web-service environment, a transaction manager confirms or cancels the backend system a web service encapsulates. Hence, a direct communication exists between the transaction manager and the backend system, which contradicts the web-service philosophy. Opening up backend systems to play the role of participant within the transaction for external parties introduces security issues and bypasses the purpose of web services.

Every phase of a transaction within BTP [20] stands on its own and may be implemented in any way by a BTP compliant web service or application. To reflect the differences with the traditional 2PC protocol, the commands used in BTP are different and extended. Additionally, using business logic in BTP, the application also determines which participants to commit as a consensus group and which to cancel.

4.2 Web Services Transactions

The combined Web Services Transactions specifications that are indicated by WS-Tx, consist of WS-Coordination (WS-C) [12], WS-AtomicTransaction (WS-AT) [13], and WS-BusinessActivity (WS-BA) [14]. The specifications are aimed at the reliable and consistent execution of web based business transactions using different interconnected web services.

While in BTP the coordination of an eBC is interwoven with transaction management, WS-Coordination (WS-C) defines a framework that solely focuses on outcome determination and processing. This way WS-C provides a generic coordination infrastructure for web services, making it possible to plug in specific coordination protocols [21, 37]. Currently, the WS-Transaction specifications (WS-AT and WS-BA) are the first and only protocol specifications based on WS-Coordination.

The WS-AtomicTransaction (WS-AT) specification is focused on the existing transaction systems and protocols with strict ACID requirements. These systems are heterogeneous and coupling them together within one organization is the first step towards interoperability. The following protocols are specified in WS-AT: Completion, TwoPhase Commit (2PC) with two variants, Volatile 2PC, and Durable 2PC. Details of these protocols [13] can be found in the WS-AT specification.

While the WS-AT specification resembles very much traditional 2PC ACID transactions with its problems, WS-BA is designed to support long running business transactions and uses atomic transactions to preserve the autonomy of participating organizations whilst at the same time providing mechanisms to reach overall agreement. The WS-BA specification defines two types for a coordinator, namely the atomic outcome type and mixed outcome type. The first type requires the coordinator to drive all participants to the same final state. The latter type allows a coordinator to choose which participants need to commit or compensate. The behaviour of the coordinator is determined by the application driving the activity. Besides the two coordination types, the following two coordination protocols [14] are specified: BusinessAgreementWithParticipantCompletion and BusinessAgreementWithCoordinator-Completion. The reader is referred to the specification for details on these protocols.

4.3 Web Services Composite Application Framework

The purpose of WS-CAF [6] is to develop an interoperable, easy to use framework for composite web services applications. WS-CAF is composed of a series of specifications consisting of WS Context [7], WS Coordination Framework [8] and WS Transaction Management [9]. Each specification covers a certain level of the overall architecture required to build reliable business applications that span multiple systems and use web service technology.

In contrast to BTP and WS-Tx, the WS-CTX specification (Bunting et al. 2003b) defines a generic context management mechanism for sharing common system data (i.e., context) across multiple web services [39]. Compared to WS-CTX, WS-Coordination combines both context and coordination, while in BTP, context, coordination as well as transaction management are combined. WS-CTX [7] makes it possible to connect multiple web services into one activity or scope for correlating them with specific context information that is not managed by a coordinator.

The second layer of WS-CAF is the Web Service Coordination Framework WS-CF which provides a coordination service that is plugged into WS-CTX. It manages and coordinates multiple web services that are grouped in one or more activities to perform some task together. The WS-CF architecture has three main components. The Coordinator at which Participants can register so that they receive the context and outcome of an activity, and the Coordination Service, which defines the behaviour for a specific coordination model.

On top of the coordination framework, Web Service Transaction Management WS-TXM [9] specifies three different transaction protocols. These protocols can be used to reach an agreement of outcome among the participants of a transaction in a consistent way. To do this, the transaction protocols can use context information and coordination protocols.

WS-TXM defines three protocols that are plugged into WS-CF and can be used with a coordinator to negotiate a set of actions for all participants that need to be executed, based on the outcome of a series of related web services executions [6]. The web service executions are linked together in scopes by the overall context and can be nested and executed concurrently [37]. WS-TXM binds the scope of an activity to the lifetime of a transaction.

Three specific transaction models are defined in WS-TXM that can be used for different situations: ACID Transaction resembles the traditional ACID transactions that is applies to web services, enabling tightly-coupled network-based transactions, which are most suitable, just like WS-AT, to achieve interoperability between existing transaction systems within one organization. Long Running Action (LRA) is designed to cover transactions that have a long duration. In LRA, an activity is seen as a set of business interactions for which compensation is possible comparable to Sagas. Business Process Transaction Model tries to integrate different heterogeneous transaction systems, e.g., using ACID transactions and messaging, from different business domains into one overall business-to-business transaction [37].

4.4 Comparison of Industry Initiatives

In a comparison of BTP and WS-Tx [37], both specifications address the problems of running transactions with web services while the differences in critical areas are present, e.g., transaction interoperability. The main problem of BTP is that it needs to leverage the ACID transactions that underlie the strongly coupled internal information infrastructures instead of replacing them with new models to design transactions for loosely coupled web services.

In [30] all three specifications are compared with the conclusion that a need exists for one open standard to realize the interoperability both in web services and business areas, possibly by integrating the existing ones within the WS-CAF framework. According to [29], BTP is the most promising standard candidate for transaction management in combination with agent technology.

With the exception of BTP, the WS-Tx and the WS-CAF initiative show that a development is pursued towards an eBT with ACID properties on a lower level and advanced business transactions on a higher level. Still, the latter two industry initiatives lack the ability to create elaborate transaction frameworks for inter-organizationally harmonizing heterogeneous transactions of collaborating business domains.

5 Related Work

In the area of workflow-oriented research, inter-organizational business transaction concepts were developed. Hence, the two most relevant research projects are described below. In [26], a two-layer transaction model, known as the WIDE transaction model, is presented. The model combines the concept of savepoints with Sagas [22] so that more flexibility is offered in compensation paths in case of exceptions. The bottom layer consists of local transactions with a nested structure that conform to the ACID properties [10]. The upper layer is based on Sagas that roll back the completed sub-transactions using the compensation mechanism, thus, relaxing the requirement of atomicity [45]. The semantics of the upper layer is formalized using simple set and graph theory [25]. The local transaction layer is designed to model low-level, short-living business processes, whilst the global transaction models high-level and long-living business processes.

The flexible approach of WIDE is adopted in the CrossFlow project [46] and developed into the more comprehensive X-transaction model. The X-transaction model is a three-level, compensation-based transaction model to support cross-organizational workflow management, namely the outsourcing level, the contract level and the internal level, each with a different visibility to the consumer or the provider organization. The X-transaction model views an entire workflow process as a transaction. For intra-organizational processes, X-steps are divided into smaller I-steps that adhere to ACID properties. Each I-step has a compensating step in case of failure

6 Conclusion

This paper specifies a conceptual framework for managing the complex issues of electronic business transactions where a separation of concerns is achieved by establishing different coordinated transactional levels within and across organizational domains. That way it is not required to directly link the respective legacy systems of business parties who simultaneously are able to protect their competitive advantage. The paper discusses which pre-existing transaction concepts are employable for which level of

an eBT. Furthermore, industrial initiatives are investigated for the extent to which they realize the electronic business transaction framework.

For future research activities about electronic business transactions, several domains need to be investigated. For specifying the requirements of a suitable transactional model, the requirements of business transactions in business process must be explored. The semantics of business collaborations must be clarified as a foundation for developing an automated composition of several transaction concepts for a heterogeneous system environment. For such automated rules-based composition, existing transaction models must be analyzed and formalized into abstract transactional primitives. Such formalization also needs to cover unconventional atomicities for electronic business transactions. Hence, the automated composition needs to integrate traditional technical transactions from the domain of database management with inter-organizational electronic business transactions that are long-lived open and nested, that have unconventional atomicities and are suitable for a service-oriented collaboration environment. If exceptional situations occur in an electronic business transaction, it must be ensured that a semantic rollback of already carried out real-world collaboration activities is supported. Besides atomicities, it also needs to be investigated how isolation, consistency, and durability are affected in an electronic business transaction when the semantics of a business collaboration is taken into account.

Bibliography

[1] Little M., and T. J. Freund, (2003), *A comparison of web services transaction protocols.* `http://www-106.ibm.com/developerworks/webservices/library/ws-comproto/`, 2003.

[2] Astrahan M. M., and M. W. Blasgen, D. D. Chamberlin, Kapali P. Eswaran, J. Gray, P. P. Griffiths, W. F. King III, R. A. Lorie, P. R. McJones, J. W. Mehl, G. R. Putzolu, I. L. Traiger, B. W. Wade, and V. Watson, (1976). *System r: Relational approach to database management.* ACM Trans. Database Syst., 1(2):97-137.

[3] Biliris, A. and S. Dar, N. H. Gehani, H. V. Jagadish, and K. Ramamritham, (1994). *Asset: A system for supporting extended transactions.*

In Richard T. Snodgrass and Marianne Winslett, editors, SIGMOD Conference, pages 44-54. ACM Press.

[4] Bjork, V. and C.T. Davis, (1978) *Data processing spheres of control-data.*

[5] Box, D. and D. Ehnebuske, and G. Kakivaya et al., (2003) *Simple Object Access Protocol (SOAP) 1.1.* `http://www.w3.org/TR/SOAP/`.

[6] Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, (2003a). *Web Services Composite Application Framework (WSCAF),* July 2003.

[7] Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, (2003b). *Web Service Context (WS-CTX)*, July 2003.

[8] Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and K. Swenson, (2003c). *Web Service Coordination Framework (WS-CF)*, July 2003.

[9] Bunting, D. and M. Chapman, O. Hurley, M. Little, J. Mischkinsky, E. Newcomer, J. Webber, and Keith Swenson, (2003d). *Web Services Transaction Management (WS-TXM)*, July 2003.

[10] Boertjes, E. and P.W.P.J. Grefen, J. Vonk, and P.M.G. Apers, (1998). *An architecture for nested transaction support on standard database systems.* In G. Quirchmayr, E. Schweighofer, and T.J.M. Bench-Capon, editors, (1998), DEXA, volume 1460 of Lecture Notes in Computer Science, pages 448-459. Springer.

[11] Breitbart, Y. and H. Garcia-Molina, and A. Silberschatz, (1992). *Overview of multidatabase transaction management.* VLDB J., 1(2):181-239.

[12] Cabrera, L. F. and G. Copeland, M. Feingold, R. W. Freund, T. Freund, J. Johnson, S. Joyce, C. Kaler, J. Klein, D. Langworthy, M. Little, A. Nadalin, E. Newcomer, D. Orchard, I. Robinson, J. Shewchuk, and T. Storey, (2005a). *Web service coordination (WS-coordination)*, August 2005.

[13] Cabrera, L. F. and G. Copeland, M. Feingold, R. W. Freund, T. Freund, S. Joyce, J. Johnson, C. Kaler, J. Klein, D. Langworthy, M. Little, A. Nadalin, E. Newcomer, D. Orchard, I. Robinson, T. Storey, and S. Thatte, (2005b). *Web Services Atomic Transaction (WSAtomicTransaction)*, August 2005.

[14] Cabrera, L. F. and G. Copeland, R. W. Freund, T. Freund, S. Joyce, J. Klein, D. Langworthy, M. Little, F. Leymann, E. Newcomer, D. Orchard, I. Robinson, T. Storey, and S. Thatte, (2005c). *Web Services Business Activity Framework (WS-BusinessActivity)*, August 2005.

[15] Ceponkus, A. and S. Dalal, T. Fletcher, P. Furniss, A. Green, and B. Pope, (2002). *Business transaction protocol version 1.0*, June 2002.

[16] Chrysanthis, P. K. and K. Ramamritham, (1990). *Acta: A framework for specifying and reasoning about transaction structure and behavior.* In Hector Garcia-Molina, H. V. Jagadish: Proceedings of the 1990 ACM SIGMOD International Conference on Management of Data, Atlantic City, NJ, May 23-25, 1990. ACM Press 1990, pages 194-203.

[17] Chrysanthis, P. K. and K. Ramamritham (1992). *Acta: The saga continues. In Database Transaction Models for Advanced Applications*, pages 349-397. Morgan Kaufmann, 1992.

[18] Chrysanthis, P. K. and K. Ramamritham. *Delegation in acta to control sharing in extended transactions.* IEEE Data Eng. Bull., 16(2):16-19, 1993.

[19] Chrysanthis, P. K. and K. Ramamritham. *Synthesis of extended transaction models using acta.* ACM Trans. Database Syst., 19(3):450-491, 1994.

[20] Dalal, S. and S. Temel, M. Little, M. Potts, and J. Webber, (2003). *Coordinating business transactions on the web.* Internet Computing, IEEE, 7(1):30-39, January / February 2003.

[21] Freund, T. and T. Story, (2002). *Transactions in the world of web services, part 1.* Web Article, 8.

[22] Garcia-Molina, H. and K. Salem, (1987). Sagas. In U. Dayal and I.L. Traiger, editors. SIGMOD Conference, pages 249-259. ACM Press.

[23] Gray, J. and A. Reuter, (1993). *Transaction Processing: Concepts and Techniques.* Morgan Kaufmann.

[24] Grefen, P.W.P.J. and H. Ludwig, and S. Angelov, (2002). *A framework for e-services: a three-level approach towards process and data management.* In 02-07, C. T. R., editor, University of Twente. (also appeared as: IBM Research Report RC 22378; IBM Research Division, 2002).

[25] Grefen, P.W.P.J. and J. Vonk, and P.M.G. Apers, (2001). *Global transaction support for workflow management systems: from formal specification to practical implementation.* VLDB J., 10(4):316-333.

[26] Grefen, P.W.P.J. and J. Vonk, E. Boertjes, and P. M. G. Apers, (1997). *Two-layer transaction management for workflow management applications.* In Abdelkader Hameurlain and A. Min Tjoa, editors, DEXA, volume 1308 of Lecture Notes in Computer Science, pages 430-439. Springer.

[27] Hanemann, A. editor, (2005). *IBM PhD Student Symposium at the 3rd International Conference on Service Oriented Computing (ICSOC 2005)* Amsterdam, The Netherlands, December 12, 2005, Proceedings, IBM Research Report RC23826. IBM Research Division.

[28] Heuvel, W.J. van den and S. Artyshchev, (2002). *Developing a three-dimensional transaction model for supporting atomicity spheres.* Proceedings of NetObjectDays 2002 vol. 2.

[29] Jin, T. and S. Goschnick, (2003). *Utilizing web services in an agent-based transaction model (abt).* In International workshop on Web Services and Agent-based Engineering.

[30] Kratz, B. (2004). *Protocols for long running business transactions.* Technical report, Infolab, Tilburg University.

[31] Kratz B., (2005). *A business aware transaction framework for service-oriented environments.* In Hanemann (2005).

[32] Kratz, B. and T.Wang, P.W.P.J. Grefen, and J. Vonk, (2005). *Flexible business-transaction composition in service-oriented environments.* Technical report, Eindhoven University of Technology.

[33] Lewis, P.M. and A. J. Bernstein, and M. Kifer, (2001). *Databases and Transaction Processing: An Application-Oriented Approach.* Addison-Wesley.

[34] Leymann, F. and D. Roller, (2000). *Production Workflow: Concepts and Techniques.* Prentice-Hall, N.J.

[35] Little, M. (2002). Business transaction protocol: Transactions for a new age. Web Services Journal, 2(11):50-55.

[36] Little, M. and T. J. Freund, (2003). *A comparison of web services transaction protocols.* `http://www-106.ibm.com/developerworks/webservices/library/ws-comproto/`.

[37] Little, M. and J. Webber, (2003). *Introducing WS-CAF - more than just transactions.* Web Services Journal, 3(12):52-55, December2003.

[38] Moss, J.E.B. (1981). *Nested Transactions: An Approach to Reliable Distributed Computing.* PhD thesis, M.I.T., April 1981. Also available as Technical Report MIT/LCS/TR-260.

[39] Newcomer, E. (2004). *Context, coordinators, and transactions: the importance of WS-CAF.*

[40] Norta, A. (2007), *Exploring Dynamic Inter-Organizational Business Process Collaboration*, PhD Thesis, Technology University Eindhoven, Department of Information Systems.

[41] Papazoglou, M.P. (2003). *Web Services and Business Transactions.* World Wide Web: Internet and Web Information Systems, 6:49-91.

[42] Papazoglou, M.P. and D. Georgakopoulos, (2003). *Service-oriented computing.* Communications of the ACM, 46(10):25-28.

[43] Papazoglou, M.P. and P. Ribbers, A. Tsalgatidou (2000), *Integrated value chains and their implications from a business and technology standpoint*, Decision Support Systems (29)4, pp. 323-342

[44] Tygar, J.D. (1996). *Atomicity in electronic commerce.* In Proceedings of the 15th Annual ACM Symposium on Principles of Distributed Computing (PODC'96), pages 8-26, New York.

[45] Vonk, J. and P.W.P.J. Grefen, E. Boertjes, and P.M.G. Apers, (1999). *Distributed global transaction support for workflow management applications.* In T.J.M. Bench-Capon, G. Soda, and A. Min Tjoa, editors, DEXA, volume 1677 of Lecture Notes in Computer Science, pages 942-951. Springer.

[46] Vonk, J. and P.W. P. J. Grefen, (2003). *Cross-organizational transaction support for e-services in virtual enterprises.* Distributed and Parallel Databases, 14(2):137-172.

[47] Wang, T. (2005). *Towards a transaction framework for contract-driven, service-oriented business processes.* In Hanemann 2005.

[48] Wang, T. and P. Grefen, (2005). *A historic survey of transaction management from flat to grid transactions.*, Beta Working Paper (Rep. No. WP-138), 22 pp.

[49] Wang, T. and P. Grefen, and J. Vonk, (2006). *Abstract transaction construct: Building a transaction framework for contract-driven, service-oriented business processes.* In A. Dan and W. Lamersdorf, editors, Service-Oriented Computing, ICSOC 2006, 4th International Conference, Volume 4294 of Lecture Notes in Computer Science, pages 352.364, Chicago, USA. LNCS Springer.

[50] Webber, J. and S. Parastatidis, (2003). *Demystifying service-oriented architecture.* Web Services Journal, 3(11):40-44.

[51] Weikum, G. and H.-J. Schek, (1992). *Concepts and applications of multilevel transactions and open nested transactions.* In Elmagarmid, A., editor, Database transaction models for advanced applications, chapter 13, pages 515-553. Morgan Kaufmann Publishers, San Mateo, California, USA.

[52] X/Open Company Ltd., (1996). *Distributed transaction processing: Reference model, version 3.* Yu, E. et al., (1998). From Organization Models to System Requirements: a "Cooperating Agents" Approach. In M.P. Papazoglou and G. Schlageter, editors, Cooperative Information Systems: Trends and Directions. Academic Press.

Challenges of parallel processor design

M. Forsell[†], V. Leppänen[‡] and Prof. M. Penttonen[§]

[†]VTT, Platform Architectures
[‡]Department of Information Technology, University of Turku
[§]Department of Computer Science, University of Kuopio

[§]PL 1627, 70211 Kuopio, Finland

Email: [†]`Martti.Forsell@vtt.fi`, [‡]`Ville.Leppanen@it.utu.fi`,
[§]`Martti.Penttonen@uku.fi`

Abstract

While processor speeds have grown, also the gap between the speed of processing and the speed of accessing data has grown. Therefore the speed of the processor cannot be used. As a reaction, processor industry has started to build more processor cores on a chip, but there is no easy way to utilize multiple processors. In this work we study alternative multicore processor designs that could efficiently run parallel programs.

Contents

1 What Moore said?

Very soon after the success story of microelectronics had started, G. Moore published a forecast [13] that was later to be called the "Moore's law". By development from year 1959, when a circuit consisted of one electronic component, to year 1965, when 50 components could be packed on a circuit, he bravely forecast that in 1975 one could perhaps pack economically as much as 65000 component. In other words, in 16 years the packing density would grow 2^{16}-fold. In 2007 the packing density was almost 5 billions, or about 2^{32}-fold. Hence, in 48 years the packing density did not grow 2^{48} fold but still the “law” can be stated in a milder form: “packing density doubles every 18 months”.

In recent years the “Moore’s law” got more popular formulations like “the power of pc’s doubles every 18 months” or alike. Similar “laws” have been presented for the growth of the bandwidth of data communication. Can we trust on such “laws”? Even if we can always hope for the revolutionary inventions by scientists, such as quantum computation, we must be aware of the physical constraints of our world. An electronic component cannot become smaller than an atom (or an elementary particle). To transport information from one place to another needs some time. At very high packing density and high clock rate, heat production becomes a problem. Electrical wires on circuits cannot be radically thinner than what we have got now, or quantum effects start to appear.

The current packing density already has lead the processor industry to a problematic situation: How to get optimal computational power from the chip? How to get data at right time at right place so that the computation is not delayed by latencies? Overheads of the memories and the time of moving data over physical distances imply latencies that are about a hundred times more than the time required by the instruction itself. This has lead to complicated caching, which should be called art rather than science. As computation depends on data, also compilation of programs has become art, when one tries to guess the branch, how the computation continues, in order to start fetching data as early as possible. It has been possible to build more and more components on a chip, but it is difficult to speed up a single computation. The computer industry has kind of “raised up hands” by starting to build multiple processor “cores” on one chip without clearly seeing, how to use them.

In principle multicore chips solve the “von Neumann bottleneck” problem, as all data need not be processed at the same processor core. The big new problem is that can we utilize multiple cores. If there are two or four cores, we can give detailed instructions, what to do in each core. But if the number of cores grows, can the programmer use them optimally? Anyway, programming is difficult enough without new factors to optimize. If the multiple cores are not used in a systematic and efficient way, one can easily end up with a program that is slower than the sequential unicore program.

2 Theoretical basis of general purpose parallel computing

The advantage of high processing speed is lost, if the data to be processed is not present. All data cannot always be present and due to distance and hardware overheads, fetching data takes enormous time in comparison with processing speed. Without losing the generality of the computation, we cannot assume that at successive steps of computation only “local” data is used. In the worst case, each sequential step of the computation could need data from the “remotest” part of the memory. One can perhaps predict, what data the next few instructions need and prefetch them to a cache, but the bigger the latency gap grows, the harder is the prediction. This is a hard fact: it is useless to just speedup a processor, if we cannot guarantee that it can do something useful.

Parallel processing offers a solution. If there are many independent threads of computation available, instead of waiting for data the processor can move (even literally) to process another thread. While other threads are processed, the waited data hopefully comes, and the processing of the waiting thread can be continued.

However, some questions rise:

1. Can we find enough independent parallel threads in the computation so that latency time can be meaningfully used?

2. Can the data communication machinery of the computer serve all parallel threads?

3. Can the swapping between threads be implemented so that efficiency is not lost?

4. Can all this be done so that programming does not become more difficult than before?

The first question is algorithmic and the second one concerns the hardware architecture. The last two questions are not theoretically as fundamental, but the idea of parallel threads lives or dies depending on how successful we are at these questions. In short, the condition of a successful model of computation is that algorithm design should be possible at a high level of abstraction, and still algorithms can be automatically compiled to programs that run efficiently on hardware.

The theory of parallel algorithms, in particular the PRAM (Parallel Random Access Machines), see [9] answers positively to the first question. A lot of parallelism can be found in computational tasks. It also answers quite positively to the last question. Even programming for the PRAM model is different from programming for the sequential RAM model, it is not more difficult as soon as we can get rid of the fixed idea of sequential programming.

Questions 2 and 3 are more difficult to answer. Based on the analysis [2] of the situation in nineties, the PRAM model was deemed to be unrealistic. PRAM assumes that an unbounded number of processors can run synchronously from instruction to instruction, even if some instructions require data from an arbitrary memory location, and many processors may be writing to the same memory location. Parallel processors cannot get rid of the latencies of the physical world, but as it was hinted above, parallelism offers a way to use the waiting time meaningfully. The idea of *parallel slackness* was proposed by Valiant [14]. If the parallel algorithm uses sp virtual processors, where s is the slackness factor and p is the number of physical processors, each virtual processor can use only fraction $1/s$ of the physical processor time, i.e. the computation of the virtual processor proceeds every s'th step of time. If s is not smaller than the latency l, the computation of the virtual processor proceeds at full speed with respect to the amount of physical processors. In other words, the slackness factor can be seen to decrease the clockrate of a virtual processor executing a thread by the factor s compared to the clockrate of the physical processor (and thus making the processor-memory speed gap to vanish).

In principle, the slackness solves the latency problem, but it implies a demanding data communication problem. All the time all p processors may want to read data anywhere in the computer, and each read instruction lives for the slackness time s in the routing machinery. Hence, there may be ps data packets in the internal network of the computer. As processors have nonzero volume, the distances grow at least with the qubic root of the number of processors, i.e. $s \geq l \in \sqrt[3]{p}$. In practice, the topology and many other properties of the network determine, how much slackness is needed, but the bandwidth requirement ps is unavoidable. In 2-dimensional sparse torus [12], for example, p processors are connected by a $p \times p$ mesh (or torus), where the processors are on the diagonal. In this structure the diameter of the network is p, the latency is $2p$ and latency $s = 2p$ can be used, because the bandwidth of the network is $2p^2$.

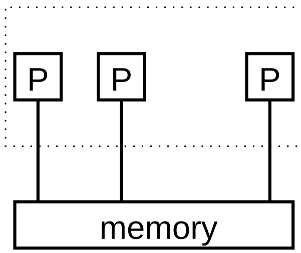


Figure 1: *Parallel Random Access Machine*

3 Parallel processor designs

In order to keep the programmer's model of computation as simple as possible, we want to see the parallel computer as in Figure 1

A vector compaction program (eliminating zero elements) for this machine would look like

```
proc compact(A)
for i=0..n-1 pardo
  if A[i]=0 then C[i]=0 else C[i]=1
E=prefix-sum(C)
for i=0..n-1 pardo
  if A[i]<>0 then B[E[i]]=A[i]
return B
```

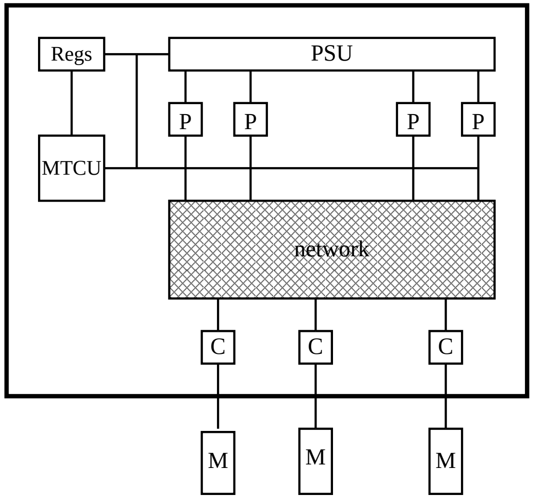


Figure 2: *Structure of the Paraleap processor*

In this program, function `prefix-sum(C)` computes vector E where E[i]=C[0]+C[1]+ . . . +C[i]. If we assume that `prefix-sum` can be computed in time $O(1)$, also `compact` can be computed in time $O(1)$.

In Figure 1, there is a dotted box around the processors, hinting that those processors should be built on a chip. We are now taking a closer look, how could such a multiprocessor chip be built.

3.1 Paraleap — PRAM on Chip

Vishkin's team [15] has built a processor, whose simplified schematic is shown in Figure 3.1. Master Thread Control Unit (MTCU) has a central role in scheduling the threads. Whenever there are independent parallel threads waiting for being processed, they ask their turn from MTCU. Whenever a thread is completed at a processor P, it is reported to MTCS. In the current version of the Paraleap, the number of processors, or more exactly Thread Control Units (TCU), is 64. However, TCU's are not fully indendent but clusters of 16 TCU's share some functional units.

Prefix Sum Unit (PSU) is another important component in Paraleap. It can be used in programs for such purposes as in the vector compaction program. However, the most important usage is the scheduling of threads to TCU's. When TCU's run and complete threads in parallel, PSU calculates, which is the next thread to be started.

Processors are connected to the shared memory by the internal network, which is a mesh of trees in Paraleap. There are 8 caches between the network and the main memory. Processors share a set of Registers (Reg).

Paraleap has been implemented on Field Programmable Gate Array (FPGA) of 75 MHz clock rate, and implementation on an 800 MHz ASIC (Application Specific Integrated Circuit) is going on. Vishkin et al. [15] claim that on a test set of 8 progams (such as binary tree search, vector compaction, matrix multiplication, or convolution), Paraleap is 1.9 to 9 times faster than a 2.6 GHz AMD Opteron processor. They expect that in near future the number of TCU's on a chip could grow to 1024.

3.2 Eclipse

In an on-going Finnish effort [4], a full architectural realization (so called Eclipse framework) of a strong PRAM model on a distributed memory chip multiprocessor (CMP) is being investigated. The main idea has been to provide enough communication bandwidth to be able to solve the routing problem described in Section 2 with a high probability, to use interleaved multithreading to hide the latency of the distributed memory system, and to synchronize the execution of emulated PRAM steps by an efficient wave synchronization technique. The communication solutions applied so far have been output buffered acyclic sparse mesh and multi mesh networks. See Figure 3.2.

Additional performance boost is sought by integrating instruction-level parallelism seamlessly with multithreading by chaining of functional units, which allows exploitation of virtual instruction-level parallelism even if executed threads are strictly sequential. Concurrent memory access to a single memory location is implemented with step caches that reduce the number of references per location to one per processor core. The data of pending multioperations are kept in the scratchpads, and active memory units are used to implement multiprefix operations [5].

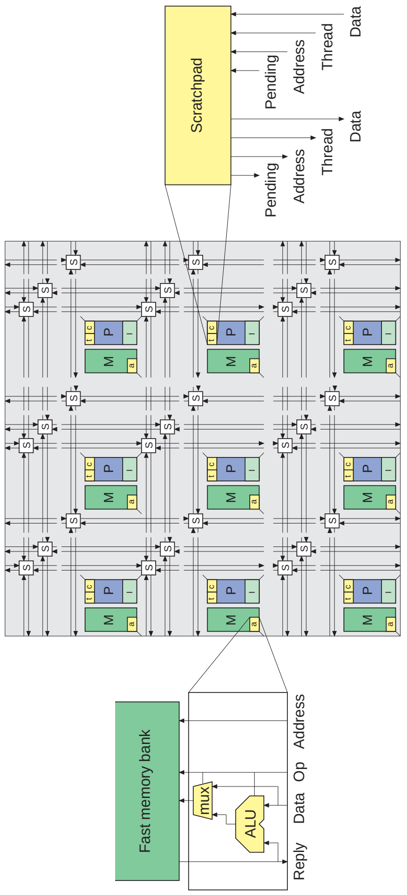


Figure 3: *High-level block diagram of the Eclipse architecture (P=processor, M=data memory, I=instruction memory, a=active memory, c=step cache, and t=scratchpad memory).*

According to recent investigations [7], the silicon area and power consumption of such a CMP is roughly comparable to those of contemporary multi-core offerings with the same number of cores from Intel. Interestingly also the performance/area and performance/power figures of the strongest PRAM variants have turned out to be the best. CMP has been investigated by software simulations. Implementation on FPGA is planned.

3.3 Moving threads

MOTH project studies the realization of a new kind of approach for mapping the computing of an application to MP-SOC architectures [6] (some preliminary ideas appear in [10, 11]). Instead of moving data read and write requests, we move extremely lightweight threads between the processor cores. Each processor core is coupled with memory module and parts of each memory module together form a virtual shared memory abstraction. Applications are written using a high-level language based on shared memory. As a consequence of moving threads instead of data we avoid all kinds of cache coherence problems.

In our architecture, the challenge of having efficient implementation of an application reduces to mapping the used data so that the need to move threads is balanced with respect to the bandwidth of the communication lines. This method also eliminates the need for separate reply network and introduces a natural way to exploit locality without sacrificing the synchronicity of the PRAM model.

In the moving threads approach, a multicore system consists of P processor cores that are connected to each other with some sparse network [10], e.g. with a butterly, a sparse mesh, a mesh of tree, etc. In traditional approaches, the messages correspond to read or write requests and replies, whereas in the moving threads approach, a message moves a thread consisting of a program counter, an id number, and a small set of registers. The messages in the moving threads approach are a little bit longer, but respectively there is no need for a network deliver the replies of read requests.

A cache-based access to the memory system is provided via each processor core. However, each core sees only a unique fraction of the overall memory space, and thus there are no cache coherence problems and when a thread makes a reference out of the scope of the core's memory area, the referencing thread must be moved to the core that can access that part of the main memory. Besides a cache to access the data memory, each core also has another cache for program instructions.

Each of the cores has $\Theta(s)$ threads to execute, and the threads are independent of each other – i.e. the core can take any of them and advance its execution. By taking an instruction cyclically from each thread, the core can wait for memory access taking a long time (and even tolerate the delays caused by moving the threads). The key to hide the memory (as well as network and other) delayes is that the average number of threads s per core must be higher than the expected delay of executing a single instruction from any thread.

The less there is need to move the threads, the smaller can the slackness factor be. Thus, although the moving threads approach does not require it, it might be wise to allow careful design of the allocation of actual data used in the program, and thus allow the programmer to balance the work-loads and to minimize the movement of data. We can e.g. assume that the program's address space is statically distributed into the memories accessible via cache modules attached to each core. The advantage of this is that the programmer can have influence on the physical allocation of data – and consequently on the physical allocation of the work of each thread on the processor-storage modules.

For the creation and termination of threads in the programming language level, we take the approach of supporting only implicit termination as well as creation of threads. We do not consider Java-like explicit declaration of threads as first-class objects as a feasible solution. In practice, we have a parallel loop-like construction which creates threads with logical id-numbers in the interval [low,high] and each threads is running the same program block. The code in the program block can of course depend on the logical processor id-number. The id-numbers are program controlled, but the runtime system expects them to be unique at anytime during the program execution. We also consider supporting nested thread creations. Each thread faces an implict termination at the end of the program block (which was defined in the thread creation statement).

4 Conclusions

In this section we discuss, what is common and what is different in the presented three parallel processor designs. The first difference is in the degree of existence and experience. Paraleap is moving from FPGA stage to

ASIC stage and some programming experience has already been collected. Eclipse has not yet reached the hardware stage, and moving threads design is still quite sketchy. They all share the view that processor must support parallel processing, otherwise the latency gap prevents efficiency.

Paraleap requires long, independent parallel threads to run efficiently. Programmer must be aware of that and therefore programming is more difficult than just writing PRAM algorithms. Eclipse and moving threads processor are synchronous and allow more fine-grained parallel processing, which is easier for the programmer. This is theoretical reasoning. Currently the difficulty of Paraleap programming is dominated by undeveloped programming environment, which is less developed for Eclipse and nonexistent for moving threads.

In all designs, a part of the chip surface is dedicated for the internal network. Paraleap uses a mesh of trees for this purpose, while we believe in sparse mesh networks, due to scalability. There is no proof yet, which is better for this purpose.

In sequencial programming caches are a central tool to fight against latency, while cache coherence is an unsolvable problem. In parallel processing caches are just one useful trick. In Paraleap and in Eclipse, caches are between the internal network and the memory. Due to slackness, coherence is not a problem. In moving threads processor, cache must be next to each core, but due to them moving thread, coherence is not a problem.

Each of the designs have their own pluses and minuses. In Paraleap the prefix sum unit has a very central role in in thread scheduling. It is sequential and said to be fast enough. Is it scalable? Eclipse is apt for instruction level optimization and it provides multioperations used in some PRAM models. But how will it work as hardware? An interesting property of moving threads processor is that it needs only one-way traffic, but packets containing register environment are bigger. How do they balance?

A lot of research is needed to clarify these design questions. Parallel processor design is taking its first steps, while there is an enormous investment in sequencial processing. A funny proof of the need of rethinking is the CUDA project [3]: A graphics processor, when used as a parallel processor, is much more efficient than the main processor of the computer. What if we designed parallel processors for computing?

Bibliography

[1] P. Bach, M. Braun, A. Formella, J. Friedrich, T. Grun, C. Lichtenau. “Building the 4 processor SB-PRAM prototype.” In *Proc. of the 30th Hawaii International Conference on System Sciences: Advanced Technology Track - Vol. 5*, 1997.

[2] D.E. Culler, R.M. Karp, D.A. Patterson, A. Sahay, K.E. Schauser, E. Santos, R. Subramonian, T. von Eicken. “LogP: Towards a realistic model of parallel computation.” In *Principles Practice of Parallel Programming, pp 1-12, 1993.*

[3] `http://www.nvidia.com/cuda`

[4] M. Forsell. “A Scalable High-Performance Computing Solution for Network on Chips.” *IEEE Micro 22(5)* (September-October 2002), pp. 46-55.

[5] M. Forsell. “Realizing Multioperations for Step Cached MP-SOCs.” In *Proc. SOC’06*, November 14-16, 2006, Tampere, pp. 77-82.

[6] M. Forsell and V. Leppänen. “Moving Threads: A Non-Conventional Approach for Mapping Computation to MP-SOC.” In *Proceedings of the 2007 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA’07)*, pages 232-238, Jun 2007.

[7] M. Forsell, J. Roivainen. “Performance, Area and Power Trade-Offs in Mesh-based Emulated Shared Memory CMP Architectures.” In *Proc. PDPTA’08*, Jul 14-17, 2008.

[8] S. Fortune, J. Wyllie. “Parallelism in Random Access Machines”. In *Proc. 10th ACM Symposium on Theory of Computing*, pp 114-118, 1978.

[9] J. J’aj’a. *An Introduction to Parallel Algorithms.* Addison Wesley, 1992.

[10] V. Leppänen: *Studies on the Realization of PRAM*, PhD thesis, University of Turku, Department of Computer Science, TUCS Dissertation 3, November, 1996.

[11] V. Leppänen. “Balanced PRAM Simulations via Moving Threads and Hashing.” *Journal of Universal Computer Science*, 4:8, 675–689, 1998.

[12] V. Leppänen, M. Penttonen. *Sparse Optical Torus.* AMICT’07 proceedings.

[13] G.E. Moore. “Cramming more components onto integrated circuits”. *Electronics* 38(8), 1965.

[14] L.G. Valiant. “General Purpose Parallel Architectures”. In *Handbook of Theoretical Computer Science, Ed. Jan van Leeuwen, Elsevier and MIT Press*, volume 1. Elsevier Science, 1990.

[15] X. Wen, U. Vishkin. “FPGA-based prototype of an PRAM-On-Chip processor.” *Computer Frontiers 2008*, May 5-7, 2008.

Regenerative simulation of finite buffer queueing systems

Ruslana S. Goricheva[†], Prof. Evsey V. Morozov[‡]

Department of Application Mathematics and Cybernetics
Pushkinskaya St., 11, Petrozavodsk, Republic of Karelia, 185000, Russia

Petrozavodsk State University
Lenin St., 33, Petrozavodsk, Republic of Karelia, 185910, Russia

E-mail: [†]Lana@mail.ru,[‡]emorozov@karelia.ru

Abstract

Renewal and regenerative processes play an important role in the modeling and simulation of the modern telecommunication systems. It is especially important to note that the regenerative approach can be used for the reliable confidence estimation of the steady-state network performance. We apply regenerative approach to construct confidence intervals mainly for the steady-state queue size and loss probability in the finite buffer systems describing typical node of a telecommunication network. Infinite buffer systems are also considered in brief. Moreover, non-standard regeneration instants are considered.

Contents

1 Introduction

In this note, we discuss in brief the regenerative simulation which is one of the most effective methods in the simulation and estimation of the steady-state parameters of the telecommunication networks. Indeed, the regenerative simulation produces asymptotically valid confidence intervals for the point estimators both for continuous - time and discrete-time regenerative processes.

The purpose of this note is to demonstrate how the regenerative simulation can be applied for a reliable confidence estimation of the steady-state queue size and the loss probability in a finite buffer systems with a heavy-tailed service time distribution. We also consider the properties of the empirical variance and conditions for the applicability of the regenerative central limit theorem. Infinite buffer systems are also simulated.

To illustrate the regenerative method, we describe a regenerative structure of a general queueing system with a finite buffer. In particular, we discuss the non-standard regeneration instants arising in the overflow. An important feature of the models is that we assume Pareto service time (or input) distribution having the so-called heavy tail. It is well-known that such distributions are adequate to describe the traffics in the modern telecommunication systems and, on the other hand, are very difficult to be analyzed by statistical methods. Simulation results related to the output of the served customers and to overflow stream, respectively, are given. We also present the calculation of the failure/loss probability based on the non-standard (hidden) regenerations.

2 Regenerative simulation

In this section, we introduce in brief the method of regenerative simulation and the weakest known condition under which the regenerative method can be applied for the confidence estimation. First, we give a basic definition. A process $X = \{X_t, t \in T\}$, where $T = [0, \infty)$ for continuous time (or $T = \{0, 1, \dots\}$ for discrete time) is called *regenerative process* if there exists an infinite sequence of the instants $0 = \beta_0 < \beta_1 < \beta_2 < \cdots$ (called regeneration points) such that the segments $G_n = (X_t,\, \beta_{n-1} \le t < \beta_n)$ (called regeneration cycles) are independent identically distributed (i.i.d.) with the generic element Y. Of course, the cycle periods $\beta_n - \beta_{n-1}$, $n \ge 1$, are also i.i.d. and we denote by β generic regeneration period (assuming that $\beta := \beta_1$ is distributed as any period $\beta_n - \beta_{n-1}$. (In fact, it means that we consider the so-called zero-delayed case, when the first customer arrives at an empty system at instant $t = 0$.)

Throughout the paper we assume that the process X is positive recurrent (that is $\mathsf{E}\beta < \infty$) and (generic) regeneration period β is aperiodic. Then the weak limit $X_n \Rightarrow X$ as $n \to \infty$ exists such that $\mathsf{P}(X < \infty) = 1$. Moreover, if f is a measurable function, then the following statement holds (for discrete-time case):

$$\lim_{n\to\infty} \frac{1}{n} \sum_{i=0}^{n} f(X_i) = \frac{\mathsf{E}[\sum_{i=0}^{\beta} f(X_i)]}{\mathsf{E}\beta} := r. \qquad (1)$$

(An evident analog for continuous time also exists.) To estimate the unknown parameter r (steady-state performance measure), we group the data belonging to the same regenerative cycle to obtain the i.i.d. enlarged variables

$$Y_p := \sum_{n=\beta_{p-1}}^{\beta_p - 1} f(X_n), \qquad p \geq 0.$$

If condition

$$0 < \mathsf{E}(Y - r\beta)^2 < \infty \qquad (2)$$

holds, then we can apply the *Regenerative Central Limit Theorem* [2]:

$$n^{1/2}\overline{\alpha}_n[r_n - r] \Rightarrow \sigma N(0,1) \qquad n \to \infty,$$

where the estimator

- $r_n := \frac{\overline{Y}_n}{\overline{\alpha}_n}$,
- N(0,1) is a standard normal variable,
- $\overline{\alpha}_n$ is the sample mean cycle period,
- $\overline{Y}_n = \frac{1}{n}\sum_i^n Y_i$,
- $\sigma^2 = \mathsf{E}(Y - r\beta)^2$,

- n is the number of completed regeneration cycles used during simulation.

Thus, we obtain the following $100(1-\gamma)\%$ asymptotic confidence interval for the unknown parameter r:

$$[r_n - \frac{z_\gamma s(n)}{\overline{\alpha}_n\sqrt{n}}, r_n + \frac{z_\gamma s(n)}{\overline{\alpha}_n\sqrt{n}}],$$

where

$$z_\gamma := \Phi^{-1}(\frac{1-\gamma}{2}),$$

Φ is the standard normal distribution and $s^2(n)$ is the empirical variance, which converges with probability 1 to the variance,

$$s^2(n) \to \sigma^2, \ n \to \infty.$$

3 Simulation results

In this section, we present some simulation results based on the regenerative method. We consider first a standard single-server system $M/G/1/m$, where buffer size m may be finite or infinite. Under finite buffer, if an arriving customer meets a full buffer, then a failure (or loss) happens. Such customers generate the overflow output (or failure flow). The served costumers produce an output flow.

3.1 M/Pareto/1/m system

Consider a queueing single-server system with Poisson arrivals with a rate $\lambda > 0$, with arrival instants $\{t_n\}$ and with the Pareto service time distribution with parameter $\alpha > 2$, that is

$$\mathsf{P}(S \geq x) = x^{-\alpha}, \ \ x \geq 1,$$

where, as usual, it is assumed that the generic service time $S \geq 1$ with probability 1. For the infinite buffer ($m = \infty$), the stability condition

$$\rho := \lambda \mathsf{E}S = \frac{\alpha\lambda}{\alpha - 1} < 1$$

is assumed to be held.

Let $\nu(t)$ be the right-continuous queue-size (number of customers) at the system at instant t and $\nu_n = \nu(t_n^-)$ be the queue-size at the arrival instants $n \geq 1$. It is easy to see that classical regenerations of the queue size (and other typical processes) appear when arriving customer meets an empty system. Thus, they can be defined in the following way:

$$\beta_{n+1} = \min\{k > \beta_n : \nu_k = 0\}, \qquad n \geq 0,$$

where, recall $\beta_0 = 0$ and first customer arrives in an empty system at instant $t = 0$. (Note that the model can be easily extended to the non-zero initial conditions.) It is also well-known that under assumption $\rho < 1$ (in infinite buffers system) the mean regeneration period is finite, $\mathsf{E}\beta < \infty$, and β is aperiodic. Note that the under assumption $\alpha > 2$ service time has finite second moment, $\mathsf{E}S^2 < \infty$, and it implies the finiteness of the 2nd moment of regeneration period, $\mathsf{E}\beta^2 < \infty$. Moreover, it also implies $\mathsf{E}Y^2 < \infty$, see [6]. In other words, under our assumptions, the confidence estimation based on the regenerative simulation can be applied. In the experiments, we vary values of the parameters to construct a number of confidence intervals for the desired parameter r. In the Table 1 we present the values of the parameters for $M/Pareto/1$ system with different buffer sizes (presented in Table 2).

Table 1: *Parameters of $M/Pareto/1/m$ system.*

λ	α	ρ
0.02	2.9	0.0305
0.35	2.8	0.5444
0.46	2.5	0.7666
0.55	2.5	0.9166
0.75	2.4	1.2857

Numerical results on confidence estimation are presented in Table 2, where

$$\Delta := \Delta(n) = \frac{z_\gamma s(n)}{\overline{\alpha}_n \sqrt{n}}, \quad \gamma = 0.05.$$

Now we study a dependence of the confidence interval on the number of arrivals.

Table 2: *Confidence intervals for r in $M/Pareto/1/m$ system.*

ρ	m	n	r_n	Δ	$r_n - \Delta$	$r_n + \Delta$
0.0305	∞	19394	0.0420	0.00037	0.0417	0.0424
0.5444	∞	9101	1.7579	0.50910	1.2488	2.2670
0.7666	∞	4580	1.7911	4.9576	0	6.7487
0.7666	5	5693	1.1849	0.1104	1.0745	1.2952
0.9166	∞	1639	13.2554	1001.2128	0	1014.4682
0.9166	5	3979	3.9913	1.04226	2.9491	5.0336
0.9166	10	2559	7.1875	7.18753	0	25.3867
1.2857	5	1198	3.0285	0.5756	2.4528	3.6041
1.2857	10	180	6.3497	8.5228	0	14.8723
1.2857	20	6	13.6798	5.4124	8.2674	19.0922

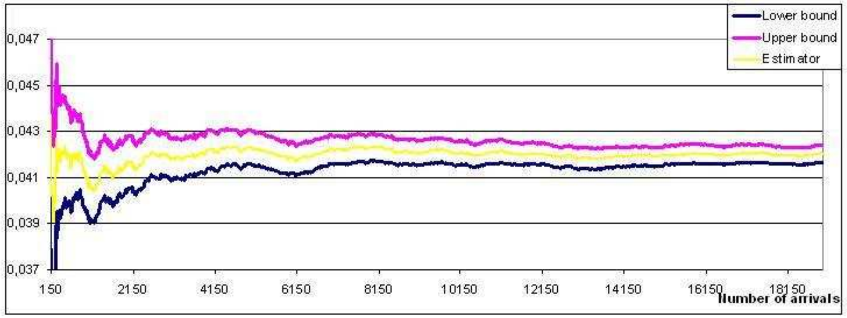


Figure 1: *Dependence the confidence interval on the number of arrivals in $M/Pareto/1/\infty$ with $\rho = 0.0305$.*

3.2 Pareto/M/1/m system

We also consider a dual finite buffer $Pareto/M/1/m$ system, where now interarrival times have Pareto distribution with parameter $\alpha > 2$ and service times are exponential with rate $\lambda > 0$.

By the memory less property of the service time, the unfinished service time (at any instant) has the same exponential distribution. This allows us to introduce another regenerations, called k-regenerations, when an

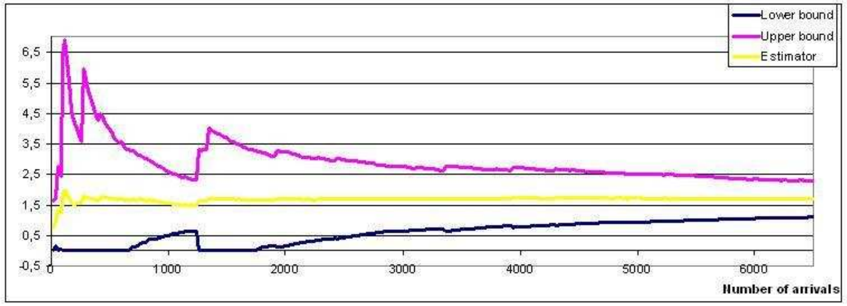


Figure 2: *Dependence the confidence interval on the number of arrivals in $M/Pareto/1/5$ with $\rho = 0.5444$.*

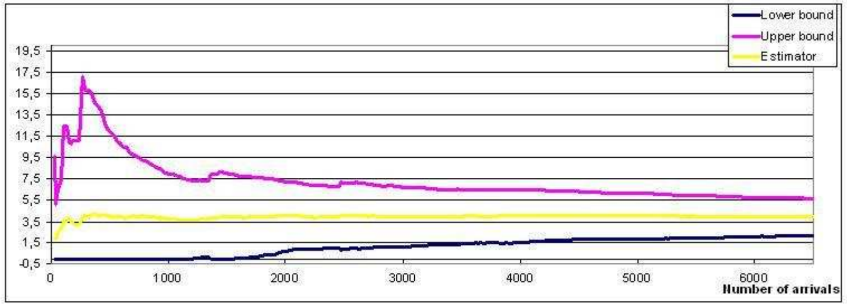


Figure 3: *Dependence the confidence interval on the number of arrivals in $M/Pareto/1/5$ with $\rho = 0.9166$.*

arriving customer meets a fixed k customers in the system ($k \le m$). In this case, the distribution of the system process is completely defined by the number k and the remaining service time having, however, the known exponential distribution.

In particular, for $k = 0$ we obtain classical 0-regenerations. It is worth to mention that such instants are not connected with the events in the overflow caused by the lost customers, but nevertheless these instants can be used in regenerative simulation and estimation of the loss probability (and other parameters of the overflow). By this reason they may be called *hidden regenerations.*

Our goal again is to estimate the unknown parameter r, the mean number of customers in the system (mean queue size). Varying parameters α and μ (as in Table 3) we obtain different confidence intervals for r.

Table 3: *Parameters in $Pareto/M/1/m$ system.*

α	μ	ρ
2.1	3.500	0.1496
2.4	0.950	0.6140
2.4	0.770	0.7575
2.5	0.601	0.9983

Table 4: *Confidence intervals for r in $Pareto/M/1/m$.*

ρ	n	k	r_n	Δ	$r_n - \Delta$	$r_n + \Delta$
0.1496	19796	1	1.0670	0.00911	1.0579	1.0761
0.6140	10979	1	2.5984	0.48581	2.1126	3.0842
0.6140	2204	3	2.5997	1.13686	1.4628	3.7365
0.7575	6874	1	4.2750	3.06209	1.2129	7.3371
0.7575	4501	2	4.2752	3.77476	0.5005	8.0499
0.7575	1997	4	4.2772	5.75791	0	10.0350
0.9983	1902	1	9.0477	17.16564	0	26.2133
0.9983	1789	6	9.0470	18.12796	0	27.1750
0.9983	5203	10	9.0577	11.12543	0	20.1832

Numerical simulation results are presented in Table 4.

Figure 4 shows the comparison of confidence intervals for two different types of regeneration. Note that the classical regenerations give better estimation.

3.3 Estimation the loss probability

Consider a $M/Pareto/1/m$ finite buffer system with Poisson arrivals with rate $\lambda > 0$ and Pareto service time distribution with parameter $\alpha > 2$. We keep notation ν_n for the queue size embedded at the arrival instants and recall that classical regenerations $0 = \beta_0 < \beta_1 < \cdots$ denotes the arrival instants into an empty system. Meeting the full buffer, an arriving customer is lost. Such losses form a failure/loss flow, and our purpose

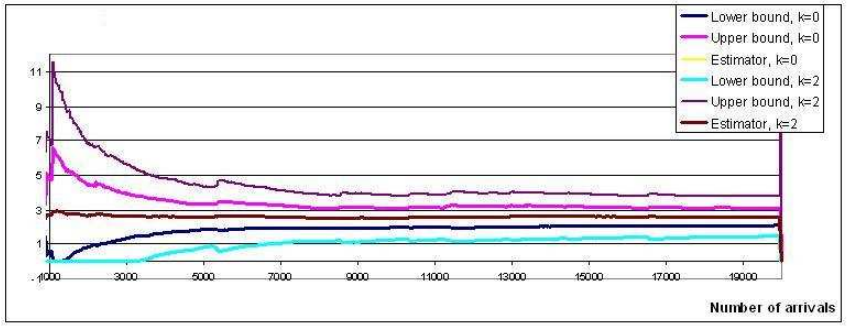


Figure 4: *Dependence the confidence interval on the number of arrivals in $Pareto/M/1/10$ with $\rho = 0.6140$ for 0 (classical)- and 2-regenerations.*

is to use regenerative simulation to estimate the fraction of the lost customers during a fixed time interval to obtain a reliable estimate of the loss probability. The following variable denotes the number of failures within discrete-time time interval $[0, n]$:

$$\xi_n = \sum_1^n I_i,$$

where indicator function $I_i = 1$ if the ith customer is lost. Because the regeneration cycles $G_n = \{\nu_i, \beta_{n-1} \le i < \beta_n\}$ are i.i.d., then the the parts of the overflow belonging to intervals $[\beta_{n-1}, \beta_n)$, $n \ge 1$ are i.i.d. random elements. Thus, the sequence $I = \{I_i,\ i \ge 1\}$ constitutes a regenerative process. Our system regenerates whenever an arrival happens in an empty system. It means that no event in the overflow at a regeneration instant. In other words, regeneration instants can not be detected by the observation the overflow only. From this point of view, such instants are hidden, as it mentioned above. It follows that the following limit (with probability 1)

$$\lim_{n\to\infty} \frac{\xi_n}{n} = \frac{\mathsf{E}\{\sum_{i=1}^{\beta} I_i\}}{\mathsf{E}\beta} := r,$$

exists and is the desired loss probability. Confidence estimation of r using regenerative simulation is presented in Table 6. Parameters of the system are presented in Table 5.

Table 5: *Parameters in a $M/Pareto/1/m$ system.*

λ	α	ρ	m
0.11	2.4	0.1885	3
0.40	2.6	0.6500	6
0.40	2.6	0.6500	10
0.65	2.9	0.9921	6
0.65	2.9	0.9921	10

Table 6 contains more details and numerical simulation results.

Table 6: *Confidence intervals for the loss probability r in $M/Pareto/1/m$.*

ρ	m	n	ξ	r_n	Δ	$r_n - \Delta$	$r_n + \Delta$
0.1885	3	16251	523	0.00289	0.00002	0.00287	0.0029
0.6500	6	7195	879	0.01766	0.00063	0.01703	0.01829
0.6500	10	6958	124	0.00249	0.00012	0.00237	0.00261
0.9921	6	2873	4873	0.15907	0.01409	0.14498	0.17316
0.9921	10	1816	3007	0.09836	0.02361	0.07475	0.12197

4 Conclusion

The purpose of this note is to show how regenerative simulation can be applied to estimate correctly some steady-state performance measures in the single-server queueing systems. Regenerative approach allows to apply classical statistical analysis to study regenerative systems because of the independence of regenerative cycles. It is especially important that we describe and use the so-called *k-regenerations* for the estimation of the mean queue size. For the exponential service time we can use different types of regenerations to improve/accelerate estimation. In particular, it may save simulation time to obtain confidence interval with a given precision in an acceptable time. We also present the non-standard regenerations and estimation the loss probability in the $M/Pareto/1/m$ systems.

We expect that k-regenerations can be very useful tool for the estimation of a steady-state performance measure of the overflow including the loss probability.

Bibliography

[1] P. Bratley, and B. L. Fox, *A guide to simulation.* 2nd edt, New York: Springer-Verlag, 1987.

[2] P. W. Glynn, and D. L. Iglehart, *Simulation medhods for queues: an oweview.* Queueing systems, vol. 3, York: Springer-Verlag, 1988. pp. 221–237.

[3] R. Serfozo, *Basics of applied stochastic processes.* Berlin: Springer-Verlag, 2009.

[4] A. M. Law, and W. D. Kelton, *Simulation modeling and analysis.* 2nd edt, New York: McGraw- Hill, 1991.

[5] A. Karr, *Point processes and their statistical inference.* 2nd edt, New York: Marcell Dekker, 1991.

[6] E. Morozov, Weak regeneration in modeling of queueing processes Queueing Systems, 46, 295-315, 2004.

[7] L. Scruben, *Confidence interval estimation using standardized time series.* Operation Research, 31, New York: Springer-Verlag, 1983. pp. 1090–1108.

Finding Representatives in a Heterogeneous Network

Laura Langohr

Department of Computer Science, University of Helsinki

P.O.Box 86, FI-00014 University of Helsinki, Finland

E-mail: `laura.langohr@cs.helsinki.fi`

Abstract

We introduce the problem of identifying representative nodes in probabilistic graphs. We define a probabilistic similarity measure for nodes, and then apply k-medoids, a clustering method, to find groups of nodes. Finally, a representative is output from each cluster. An exemplary experiment and its promising result are shown.

Contents

1 Introduction

Information contained in large networks is difficult to view and handle by users. The problem is obvious for networks of hundreds of nodes, but the problems start already with dozens of nodes.

In this paper, we propose identification of a few representative nodes as one approach to help users make sense of large networks. As an example scenario of the approach, consider link discovery. Given a large number of predicted links, it would be useful to present only a small number of representative ones to the user. Or, representatives could be used to abstract a large set of nodes, e.g., all nodes fulfilling some user-specified criteria of relevance, into a smaller but representative sample.

Our motivation for this problem comes from genetics, where current high-throughput techniques allow simultaneous analysis of very large sets of genes or proteins. Often, these wet lab techniques identify numerous genes (or proteins, or something else) as potentially interesting, e.g., by the statistical significance of their expression, or association with a phenotype (e.g., disease). Finding representative genes among the potentially interesting ones would be useful in several ways. First, it can be used to remove redundancy, when several genes are closely related and showing all of them adds no value. Second, representatives might be helpful in identifying complementary or alternative components in biological mechanisms.

The network in our application is Biomine [1], an integrated network database currently consisting of about 1 million biological concepts and about 10 million links between them. Concepts include genes, proteins, biological processes, cellular components, molecular functions, phenotypes, articles, etc.; weighted links mostly describe their known relationships. The data originates from well known public databases such as Entrez(`http://www.ncbi.nlm.nih.gov/Entrez/`), GO(`http://www.geneontology.org/`), and OMIM(`http://www.ncbi.nlm.nih.gov/omim/`).

The problem thus is to identify few representative nodes among a set of them, in a given weighted network. The solutions proposed in this paper are based on defining a probabilistic similarity measure for nodes, then using clustering to group nodes, and finally selecting a representative from each cluster.

In this framework, two design decisions need to be made: how to measure similarities or distances of nodes in a probabilistic network (Section 2), and which clustering method to use on the nodes (Section 3). A first experimental result is reported in Section 4, and we conclude in Section 5 with some notes about the result and future work.

2 Similarities in probabilistic graphs

Probabilistic graphs offer a simple yet powerful framework for modeling relationships in weighted networks. A probabilistic graph is simply a weighted graph $G = (V, E)$ where the weight associated with an edge $e \in E$ is probability $p(e)$ (or can be transformed to a probability). The interpretation is that edge e exists with probability $p(e)$, and conversely e does not exist, or is not true, with probability $1-p(e)$. Edges are assumed mutually independent.

The probabilistic interpretation of edge weights $p(e)$ gives natural measures for indirect relationships between nodes. In this paper we call these similarity measures, as is conventional in the context of clustering.

Probability of a path Given a path P consisting of edges $e_1, \ldots, e_k$, the probability $p(P)$ of the path is the product $p(e_1) \cdot \cdots \cdot p(e_k)$. This corresponds to the probability that the path exists, i.e., that all of its edges exist.

Probability of the best path Given two nodes $u, v \in V$, a measure of their connectedness or similarity is the probability of the best path connecting them:

$$s(u, v) = \max_{P \text{ is a path from } u \text{ to } v} p(P).$$

Obviously, this is not necessarily the path with the least number of edges. This similarity function $s(\cdot)$ is our choice for finding representatives.

3 Clustering and representatives in graphs

Our approach to finding representatives in networks is to cluster the given nodes, using the similarity measure defined above, and then select one representative from each cluster (Algorithm 1). The aim is to have representatives that are similar to the nodes they represent (i.e., to other members of the cluster), and also to have diverse representatives (from different clusters). We experiment with the k-medoids method, a well-known and widely used method which can be applied to our problem of finding representatives, and directly produces representatives.

k-medoids k-medoids is similar to the better known k-means method, but better suited for clustering nodes in a graph. Given k, the number of clusters to be constructed, the k-medoids method iteratively chooses cluster centers (medoids) and assigns all nodes to the cluster identified

Algorithm 1 Find representative nodes

```
Require: Set $S$ of nodes, graph $G$, number $k$ of representatives
Ensure: $k$ representative nodes from $S$
  Find $k$ clusters of nodes in $S$ using similarities $s(\cdot)$ in graph $G$
  For each of the $k$ clusters, output its most central node (the node with
    the maximum similarity to other nodes in the cluster)
```

by the nearest medoid. The difference to the k-means clustering method is that instead of using the mean value of the objects within a cluster as cluster center, k-medoids uses the best object as a cluster center. This is a practical necessity when working with graphs, since there is no well defined mean for a set of nodes. The k-medoids method also immediately gives the representatives. See, e.g., [2, 3] for more information about the methods.

For very large graphs, a straight forward implementation of k-medoids is not necessarily the most efficient. In our applications we use the Biomine database and tools to facilitate faster clustering. Given a set S of nodes, i.e., biological entities, to be clustered, and k, the number of clusters to be constructed, the method proceeds as follows. First, the Biomine system is queried for a graph G of at most 1000 nodes cross-connecting nodes in S as strongly as possible. The pairwise similarities between nodes are then calculated as the best path probabilities in G.

The Biomine system uses a heuristic to obtain G, details are omitted here. As the Biomine network consists of a million nodes, querying it for a graph exceeds by far the computational complexity of running k-medoids on the extracted graph. For brevity, we here omit discussion of the computational complexity of k-medoids.

To start the actual clustering, k nodes from S are chosen randomly as initial medoids. Each remaining node in S is then clustered to the most similar medoid. If the pairwise similarity between a node and all medoids equals zero, the node will be considered an outlier and is not assigned to any medoid in this iteration. Then, a new medoid is calculated for each cluster. The node that has a maximal product of similarities between each other node in the cluster and itself is chosen as the new medoid. The last two steps are then repeated until the clustering converges or the maximum number of iterations is reached.

4 Experiment

As an example, k-medoids was run with $k = 3$ and a set of nine genes. The genes belong to three known groups, each group of three genes being associated to the same phenotype. The three OMIM phenotypes used in the example are a pigmentation phenotype (MIM:227220), lactase persistence (MIM:223100), and Alzheimer disease (MIM: 104300).

The algorithm converged in this case after two iterations. The result of the example run is shown in Figure 1. Looking at the quality of clustering, only one gene (EntrezGene:1627) was assigned to another cluster than it should with respect to the OMIM phenotypes. Apart from this gene, the clustering produced the expected partitioning: each gene was assigned to a cluster close to its corresponding phenotype. The three representatives (medoids) are genes assigned to different phenotypes. Hence, the medoids can be considered representative for the nine genes.

5 Conclusion

We have described the problem of finding representative nodes in large probabilistic graphs. We based our definition of node similarity on a

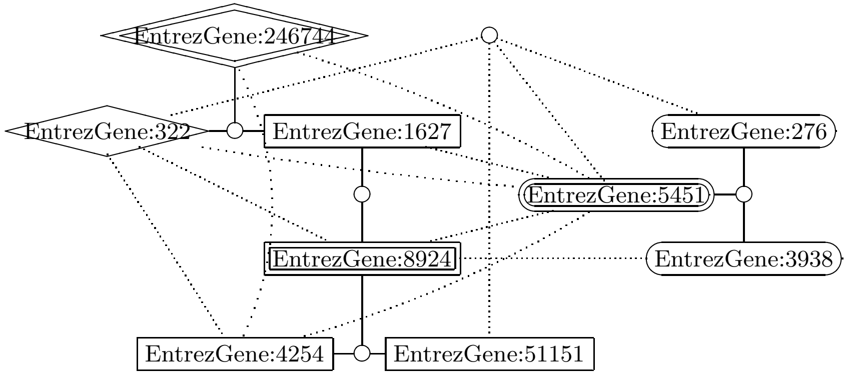


Figure 1: *Clusters (diamonds, boxes, ellipses) and representatives (double borders) of nine given nodes, and some connecting nodes (circles) on best paths between them. Lines represent edges between two nodes, dotted lines represent best paths with several nodes.*

simple probabilistic interpretation of edge weights. We then gave a k-medoids-based method for identifying representatives.

An exemplary run was described and the result shows that k-medoids clustered the genes quite well and three representatives were found. Hence, using k-medoids in a heterogeneous biological network has the potential of being useful for finding representative nodes.

However, no statistical evaluation of the method has been carried out yet. A statistical evaluation with published data is intended, as well as validations in real applications. In addition it is planed to apply hierarchical clustering to the network and to compare both methods to each other. A complete conclusion will be drawn after knowing the results of all these intended evaluations.

Based on the simple method introduced here, and the exemplary experimental result, k-medoids seems to be capable of identifying a set of representatives.

Acknowledgments

This work has been supported by the Algorithmic Data Analysis (Algodan) Centre of Excellence of the Academy of Finland and by the European Commission under the 7th Framework Programme FP7-ICT-2007-C FET-Open, contract no. BISON-211898.

Bibliography

[1] P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi, and H. Toivonen, *Link discovery in graphs derived from biological databases.* 3rd International Workshop on Data Integration in the Life Sciences 2006 (DILS'06), Hinxton, UK, 2006. pp. 35–49.

[2] J. Han, and M. Kamber, *Data Mining. Concepts and Techniques.* 2nd ed., Morgan Kaufmann, San Francisco, CA, 2006.

[3] L. Kaufman, and P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley Inc., New York, NY, 1990.

Structure patterns in Information Extraction: a multilingual solution?

Gael Lejeune

Department of Computer Science, University of Helsinki
GREYC, University of Caen

P.O.Box 68, FI-00014 University of Helsinki, Finland

E-mail: `lejeune@cs.helsinki.fi`

Abstract

IE systems nowadays work very well, but they are mostly monolingual and difficult to convert to other languages. We maybe have then to stop thinking only with traditional pattern-based approaches. Our project, PULS, makes epidemic surveillance through analysis of On-Line News in collaboration with MedISys, developed at the European Commission's Joint Research Centre (EC-JRC). PULS had only an English pattern-based system and we worked on a pilot study on French to prepare a multilingual extension. We will present here why we chose to ignore classical approaches and how we can use it with a mainly language-independent based only on discourse properties of press articlestructure. Our results show a precision of 87% and a recall of 93%. And we have good reasons to think that this approach will also be efficient for other languages.

Contents

1 Introduction

In a domain like epidemic surveillance, having an IE system limited to only one language is insufficient. For instance, for countries like France or Togo it will be difficult to find press articles about diseases written in English. Therefore some crucial information might take time to be shown by the system. If we wait for a better Machine Translation (as a recent article on Euro- surveillance proposed [1] it might take a long time as well. Much effort as been devoted to multilingual analysis. How should one extend a system to another language? As we will have different steps to follow to extract information, our goal will be to have as few language-dependent steps as possible. The purpose of this article is to show a method whose aim is modest but may also be simple and efficient.

2 Classical works in NLP area

The concept of text as a ≪bag of words ≫seems to be declining, it's obviously because from now on linguistic approaches in NLP are quite powerful. These approaches are mainly based on academical subdivisions of linguistics:

- Lexical normalization (aiming to morphemes)
- Morphological analysis (identifying ≪words ≫)
- Syntactic analysis (combining words)
- Semantic analysis (meaning representation, sentence level)
- Discourse analysis (combination of sentences or documents)

Improvements have been made to some of these tasks, using predicate/argument constructions and ontology-driven analysis. Also, Part of Speech (POS) taggers are quite efficient now. However, in the current view, a multilingual parser is a sum of language-dependent parsers. Building these resources is a long and hard job and even automatic learning always needs human fixing. Furthermore as it requires many steps and many tools, many different errors can occur during the process. In this ≪standard vision ≫the granularity used is the word (or the lexical item) and almost nothing is said about two other fields of linguistics:

- Stylistics
- Pragmatics

These parts are maybe supposed to be somewhat useless for our purpose or to have a lack of models. But there are some useful works that might help Computer Scientists.

For instance, pragmatics tell us that language is governed by effectiveness rules [2] or said differently by pertinence rules [3]. Human beings use speech acts principles to limit the cognitive cost of the exchange. For instance, as a journalist has to prove to his reader that his article is worthy of reading he will be very careful about his title and first sentences. He will also try to focus on an important and easy understandable fact. It means that in such a special type of text we have possibilities to guess what the main information is just by refering to those rules. There is also an important probability that there is only one interesting piece of information in each article, if a piece of information is worthy it should need its own article.

Studies with text as different granularity levels exist for different text types [4]. For press articles which are the main part of our corpus, many models have been used. The structure model that our approach is based on was elaborated by Nadine Lucas [5] It is conveyed by the ≪5W rule≫saying that answers to ≪Who,What, Where, When Why ≫have to be given in the very beginning of documents. It works in both French and English and we can guess that it would be a good candidate for a multilingual rule. This rule says that the main information is to be found in the top of the document. As in our epidemic surveillance system PUlS the corpus only contains press articles, we have tried to apply this rules to our task. As an experiment our simplified goal was to identify:

- What: disease
- Where: country
- Who: cases (people affected by diseases)
- When: date (in this version we extract only the document's date)

3 PULS French System

PULS's aim is to monitor as many language as possible to help epidemiologists in their task. The French system which we present here is therefore intended to be a real seedwork for monitoring new languages with as few human labor as possible:

First the documents is divided into two parts,

- HEAD: title and two first sentences
- BODY: rest of the document

3.1 Disease

If a disease from the database (150 items) is found in both parts then the document is considered possibly relevant. Then a small blacklist is used to filter out less relevant documents (cf Table 1). It is very important to control redundancy and to give to the user really relevant documents. If more than one disease are matched, the document is also considered less relevant referring to the pertinence rule.

3.2 Location

Any location from the database (400 items) found in the ≪Head≫ part is considered possibly the good one. If there is no location in the head, we search locations that appears at least twice in the body. If there is still no location matched we consider, according to the pertinence rule, that the event is happening in the default country of the news source which we find in our source database (30 items).

If more than one country is matched, this algorithm is applied:

- The relevant location appears more than 2 times in the head and twice as many as any other location
- If it's undecided, the same algorithm is applied to the whole text
- Finally if it's still undecided, the document is marked as less relevant assuming that the pertinence rule suggests that if there is an important fact in a document you must talk mostly about this fact and therefore about only one location.

3.3 Cases (descriptor)

To find cases (descriptors) we apply this rule: Cases in a relevant document are specified as the first numeric information in the first half of the text that is not related to money, distance or time. We use a stoplist which includes names of months, currency names and date nouns (20 items). When no descriptor is matched it's mostly because the number of cases is in letters therefore we extract the first phrase containing the disease name.

Table 1: *Relevance scoring*

Type of event	Explanations	To be extracted
Highly relevant	new information	Yes
Quite relevant	important update	Yes
Less relevant	review article	Yes
Very low relevance	historical, not current	No
Not relevant	non-specific event	No
Not relevant	wrong event	No

4 Results

The corpus is provided by Medical Information System, MedISys, which gather reports concerning Public Health. We worked with a sample of approximately 1200 files (from December 3 to 14 2008), from which 210 documents were manually tagged as relevant according to our scale (Table 1 :score 1 to 3). it is important to say here that we are very careful about which documents we consider relevant for our purpose.

> **Indian** **swine flu** death toll hits **100**
> MUMBAI A total of **100 people** have died from **swine flu** in **India** since the first fatality was recorded one month ago, the government said.
> The health ministry announced in a statement late on Monday that the most recent victims were four people in western Maharashtra state, which has seen 55 deaths from the **A(H1N1)** virus. **Indian** 's first confirmed case of **swine flu** was in May. The first death was on August 3. Fears about contracting **swine flu** led to huge queues forming outside government hospitals while the rising death toll led to the temporary closure of schools and cinemas in and around Mumbai and Pune.

Example on English: **disease** **country** **cases**

Table 2: *Results*

Manually tagged	Extracted	Ignored	Results
Relevant documents	196	14	Recall 93%
Non relevant documents	28	962	Precision 87.5%

Our results (Table 2) are difficult to compare to other systems since we have only language-dependent systems to compare with. However we are already very close to the English version of PULS [6] which is pattern-based.

5 Conclusion

The promising scores we got from the above experiment has convinced us that there are still improvements to get from the existing models. Our next step will be to test our system on other Romance languages (for instance Italian and Spanish) then to other Indo-European ones. If we can keep the idea and the simplicity of it in a number of language families we would be able to say that we can monitor an important part of the epidemic data in the world.

Acknowledgements

AFFRST, Content factory, Algodan and European's Commission JRC for their support. Nadine Lucas for basic algorithm and useful ideas. Roman Yangarber and Antoine Doucet for advices and reviews. Charlotte Lecluze and Calliopi Sachtouri for previous work on news reports.

Bibliography

[1] JP.Linge, R.Steinberger, T.P Weber, R.Yangarber, E van der Goot, D H Al Khudhairy, N I Stilianakis *Eurosurveillance Vol 14.* Issue 13, 02 April 2009

[2] Reboul and Moeschler *La pragmatique aujourd'hui. Une nouvelle science de la com- munication.* Paris: Le Seuil 1998(Points)

[3] Sperber and Wilson *Relevance: Communication and cognition.* Blackwell Press 1998.

[4] N. Kando *Text structure analysis based on human recognition: cases of newspaper articles and English newspaper articles.* Bulletin of the National Center for Science Information Systems, 8 (1996), pp. 107-129.

[5] N.Lucas (2004) *The enunciative structure of news dispatches: A contrastive rhetorical approach A contrastive rhetorical approach, in C. Ilie, ed., Language, culture, rhetoric: Cultural and rhetorical perspectives on communication* ASLA, Stockholm, 2004, pp. 154-164.

[6] R.Steinberger, F.Fuart, E.Van der Goot, C.Best, P. Von Etter, R.Yangarber *Text mining from the Web for medical intelligence in Mining massive data sest for security.* Amsterdam, the Netherlands OIS Press 2008.

Gaussian Processes in Communication Networks

Oleg V. Lukashenko, Prof. Evsey V. Morozov

Karelian Research Center RAS
11, Pushkinskaya Street, Petrozavodsk, Karelia, Russia, 185640

Petrozavodsk State University
Lenin St., 33, Petrozavodsk, Republic of Karelia, 185910, Russia

E-mail: lukashen@snw.ru, emorozov@karelia.ru

Abstract

We give a brief survey of the main notions and basic features of the self-similar and long-range dependent processes arising in the modeling of modern broadband communication networks. We concentrate on the mathematical properties of the Gaussian models. Moreover, the methods to detect normality of the network traffic are also discussed.

Contents

1 Introduction

Telecommunication systems have drastically changed in the past 20 years mainly because they allow to transfer data, video, voice in the same support. The traffic in telecommunications networks does not behave as the classical models of telephony based on Poisson processes. Instead, self-similarity and long-range dependence of the Internet traffic were revealed.

Self-similarity means an invariance of the distribution under suitable time and space scaling while long-range dependence means a slow decay of the autocorrelation function.

Gaussian models became popular tool to describe long-range dependence when this property of the network traffic has been revealed. The best understood Gaussian process is the fractional Brownian motion (fBm) is now the most studied in the Internet traffic modeling. This is theoretically justified that the sum of a large number of the so-called on-off inputs, with on-times or/and off-times having heavy-tailed distributions with infinite variance, converges to a fBm under appropriate time scaling.

Below we present main notions and basic features of the Gaussian self-similar and long-range dependent processes arising in the modeling of modern broadband communication networks. We concentrate on the mathematical properties of the models. However, Gaussian approximation cannot be accepted a priory. (There are many situations in which Gaussianity does not apply.) That is why the Gaussian approximation in the context of communication networking should be quantitatively justified. By this reason, we also give a brief survey of some methods which allow to detect whether the one-dimensional marginal distribution of the traffic increments satisfies normal approximation. These methods are based on statistical tests which are also presented.

2 Gaussian sources

Denote by $(A(t)$ the amount of data (traffic) arrived in a communication node within time interval $[0, t]$, $t > 0$.

It is assumed that the traffic has the following properties.

1. Stationarity. It means that the cumulative process $A(t)$, $t \geq 0$ has stationary increments, i.e.

$$A(s,t) = A(0, t-s) = A(t) - A(s), \quad \text{for all } \ s < t.$$

At the same time, Gaussian process possesses this property.

2. High aggregation level. In the modern communication networks, each node is usually fed by a superposition of a large number of the individual (independent) streams. It is known that under appropriate scaling such a superposition converges to a Gaussian process.

It shows that the Gaussian models describe adequately a wide class of the modern telecommunication networks.

Assume that $\mathsf{E}A(s,t) = \mu\,(t-s)$, where $\mu > 0$ is a constant, and that variance $VarA(s,t)$ depends on the length of the interval, that is $VarA(s,t) = \upsilon(t-s)$ for some function υ.

A Gaussian source $A(\cdot)$ is the Gaussian process with stationary increments, if for all $s < t$

$$A(s,t) =_d N(\mu\,(t-s), \upsilon(t-s)),$$

where $=_d$ denotes the equality in distribution. Define covariance function

$$C(t,\epsilon) = Cov(A(0,\epsilon), A(t,t+\epsilon)) = \frac{1}{2}(\upsilon(t+\epsilon) - 2\upsilon(t) + \upsilon(t-\epsilon)).$$

We call a traffic source long-range dependent(lrd) if the covariances $C(k,1)$ are nonsummable:

$$\sum_{k=1}^{\infty} C(k,1) = \infty,$$

and short-range dependent(srd), otherwise.

We give two important examples of Gaussian sources.

1. Fractional Brownian motion(fBm). A fractional Brownian motion has variance function $\upsilon(t) = t^{2H}$, for $H \in (0,1)$. The most known network traffic model based on fractional Brownian motion is presented in [4].

1. Integrated Ornstein-Uhlenbeck(iOU) process. An Integrated Ornstein-Uhlenbeck process has variance function $\upsilon(t) = t - 1 + e^{-t}$.

One can show that fBM is lrt process while iOU is srd process. A rather complete introduction to the Gaussian processes with application to communication networks is [3].

3 Statistical justification

Assume that the number of arrivals $\{x_i\}$ in the unit time intervals are the identically distributed random variables with unknown distribution F. Our *null hypothesis* is as follows:

$$H_0: \quad F = \Phi_{\mu,\sigma},$$

where $\Phi_{\mu,\sigma}$ is the distribution of normal variable $N(\mu,\sigma)$. The main difficulty to verify the hypothesis is that the observations $\{x_i\}$ are not independent but rather strongly correlated. To resolve the problem, we can use the well-known normal quantile (N-Q) plots for the testing the Gaussian approximation. A N-Q plot presents the pairs

$$\{a_i, x_{(i)}\}, \quad i = 1, ..., n,$$

where n is the sample size, $x_{(1)} < \cdots < x_{(n)}$ are the order statistics, i.e. increasingly ordered observations, and the points a_i are the plotting positions. The plotting positions are always assumed to be ordered, $a_1 < \cdots < a_n$, and, by the symmetry of the normal distribution, $\sum_{i=1}^n a_i = 0$.

N-Q plot is a variant of more general Q-Q plot, which is used to test whether two samples are taken from the same distribution. The empirical distribution function

$$F_n(x) = \frac{1}{n}\sum_{i=1}^{n} I_{(x_i \le x)}$$

has inverse function

$$F_n^{-1}(t) = \inf\{x \in R : F_n(x) \ge t\},$$

and if $\frac{i-1}{n} < t < \frac{i}{n}$, then $F_n^{-1}(t) = x_{(i)}$. For any two samples x_i, y_i we obtain a plot

$$\{x_{(i)}, y_{(i)}\}, \quad i = 1, \ldots, n.$$

A linear shape of the plot suggests that the samples are taken from the same underlying distribution F.

Consider construction of the plotting positions a_i. If $1 \le i \le n$, then $\frac{i-1}{n} < \frac{i}{n+1} < \frac{i}{n}$. And a_i can be defined as

$$a_i = \Phi^{-1}\left(\frac{i}{n+1}\right), \quad i = 1, ..., n,$$

where $\Phi = \Phi_{0,1}$. Again, by the symmetry of the normal distribution, $\sum_{i=1}^n a_i = 0$.

Another approach to choice of a_i is based on the Wasserstein metric, which is defined as

$$\int_0^1 \left(F_n^{-1}(t) - \mu - \sigma\Phi^{-1}(t)\right)^2 dt. \tag{3.1}$$

It is necessary to minimize this expression with respect to parameters μ and σ. In other words, we should find such $\hat{\mu}$ and $\hat{\sigma}$ which give minimum to (3.1). Using the properties of the empirical distribution function and the standard normal random variable, one can obtain by direct calculation that

$$\begin{aligned}\hat{\mu} &= \int_0^1 F_n^{-1}(t)dt \\ &= \int_{-\infty}^{\infty} y dF_n(y) \\ &= \frac{1}{n}\sum_{i=1}^{n} x_i = \overline{x},\end{aligned}$$

$$\begin{aligned}\hat{\sigma} &= \int_0^1 F_n^{-1}(y)\Phi^{-1}(y)dy \\ &= \sum_{i=1}^{n}\int_{(i-1)/n}^{i/n} F_n^{-1}(y)\Phi^{-1}(y)dy = \sum_{i=1}^{n} b_i x_i.\end{aligned}$$

Denote by ϕ the density of $N(0,1)$ and consider the b_i in more detail. Namely,

$$\begin{aligned}b_i &= \int_{(i-1)/n}^{i/n} \Phi^{-1}(t)dt \\ &= \int_{\Phi^{-1}((i-1)/n)}^{\Phi^{-1}(i/n)} y dF(y) \\ &= \frac{1}{\sqrt{2\pi}}\int_{\Phi^{-1}((i-1)/n)}^{\Phi^{-1}(i/n)} y e^{y^2/2} dy \\ &= \phi\left(\Phi^{-1}\left(\frac{i-1}{n}\right)\right) - \phi\left(\Phi^{-1}\left(\frac{i}{n}\right)\right).\end{aligned}$$

Define $\phi_i = \phi\left(\Phi^{-1}\left(\frac{i}{n}\right)\right)$, then $\phi_0 = \phi_n = 0$. Plotting positions a_i then are defined as

$$a_i = \frac{\phi_{i-1} - \phi_i}{\sum_{i=1}^{n}(\phi_{i-1} - \phi_i)^2}, \quad i = 1, ..., n.$$

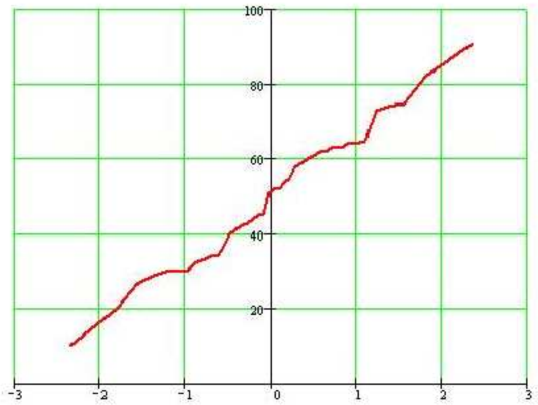


Figure 1: *N-Q plot*

Note that the equality $\sum_{i=1}^{n} a_i = 0$ is obvious.

The N-Q plot being essentially a visual tool, and there is need for more quantitative tests. An example of the N-Q plot is presented on Figure 1. (Recall that a observation is the amount of arrivals in the unit of time.)

An often-used tool is the so-called linear correlation coefficient between two samples $x = (x_{(1)}, ..., x_{(n)})$ and $a = (a_1, ..., a_n)$ which is defined as

$$r := r(x,a) = \frac{\sum_{i=1}^{n}(x_{(i)} - \overline{x})(a_i - \overline{a})}{\sqrt{\sum_{i=1}^{n}(x_{(i)} - \overline{x})^2 \sum_{i=1}^{n}(a_i - \overline{a})^2}},$$

where $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\overline{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$. It is well known that $|r| \leq 1$. If the points $(x_{(i)}, a_i)$ lie close to the straight line, then the correlation coefficient $r(x,a)$ will be close to 1.

More precisely testing the Gaussian approximation of aggregate traffic is discussed in [5].

4 Conclusion

The presented results can be summarized as follows. We consider in brief the basics of the Gaussian sources and also their applicability to describe the behavior of a wide class of communication networks. Moreover, we give the definition of long-range dependent process and two generic examples of the Gaussian sources. Also we present an elementary N-Q plot test to verify the hypothesis of the normality of network traffic. It is shown that the Gaussianity is a reasonable assumption for the network traffic modeling.

Bibliography

[1] W.E.Leland, M.S.Taqqu, W.Willinger, and D.V.Wilson. *On the self-similar nature of ethernet traffic (extended version).*IEEE/ACM Transactions of Networking, 2(1):1-15,1994.

[2] W. Willinger, M. Taqqu, W. Leland and D.Wilson. *Self-similarity in high-speed packet traffic: analysis and modeling of Ethernet traffic measurements, Statistical Sciences* **10**, No 1, 67-85, (1995).

[3] *Mandjes M.* Large Deviations of Gaussian Queues. – Chichester: Wiley, 2007.

[4] *Norros I.* A storage model with self-similar input, Queueing Systems, 1994. Vol. 16. P. 387–396.

[5] *Kilpi J., Norros I.* Testing the Gaussian approximation of aggregate traffic, Proceedings of the second ACM SIGCOMM Workshop, Marseille, France, 49-61, 2002.

Automatic Discovery of Diverse and Dynamic Network Services

Mikko Pervilä

Department of Computer Science, University of Helsinki

P.O.Box 68, FI-00014 University of Helsinki, Finland

E-mail: `pervila@cs.helsinki.fi`

Abstract

Common-mode failures have thus far been considerably overlooked in reliability studies. By their definition, common-mode failures correlate heavily, breaking the assumption of probabilistic independence. This article describes how monitoring a live, large-scale network infrastructure provides a realistic environment for analyzing common-mode failures, while also introducing new problems not immediately obvious in analytical or simulation studies. Administrators perform changes to a system whenever new requirements, research projects, or unplanned failures manifest. To distinguish changes from failures, the process from network discovery to service monitoring must be streamlined and enabled to handle extremely diverse installations.

Contents

1 Introduction

Common-mode failures (CMFs) or *M-plex* faults have been known to computer scientists since the 19th century, from the days of Babbage and his Calculating Engine [1]. The standard method of tolerating faults is by adding redundancy. Redundancy may be applied in either time, hardware, or software by performing more calculations than required. What makes CMFs so interesting is that by their definition, these faults manifest as failures in more than one repetition. This property breaks many previous failure models.

Despite our knowledge of CMFs, assumptions of independent and identically distributed failures continue to surface in academia. Simultaneously, reports from hardware enthusiasts [2–4] show an alarming number of faults affecting entire series of products. It seems logical that global processes have not made CMFs any scarcer than in the early days of computation, when computer machinery was uniquely built and maintained by local engineers.

As the methodology and bias of web site reporters is sometimes questionable, an initiative has begun to collect a scientific and open database of hardware failures [5]. A recent line of work by Schroeder et al. [6, 7] explores the field of empirical system reliability [8], within which this article belongs. Studies in the field are categorized by their methodology: measuring real-world systems' failure properties and occurrence, and then adapting new models to fit the data. These methods are hardly new, but the magnitude and dynamism of the observed systems are.

Current distributed systems and peer-to-peer networks are becoming increasingly heterogeneous or *diverse*. Diversity has been proposed as a possible cure for software-based CMFs [1]. Instead of repeating the calculations on (almost) identical components, the principle of design diversity advocates introducing heterogeneity to the system on purpose.

To complement previous studies, here the viewpoint is precisely on the diversity of the test subjects. In order to capture the states of a real-world environment, we have built a measurement framework on top of the Nagios sentinel service[1]. In this initial phase the goal is to seek the ratio of CMFs to independent failures. In later phases the measurement framework will be continously adapted to better distinguish possible causes of encountered failures. The hypothesis is that common-mode failures are as common as independent failures. Furthermore, we begin to formulate a conjecture that diversity is beneficial to the fault tolerance of distributed systems, but perhaps harmful for their maintainability.

Common-mode failures and the problem statement are formulated in Section 2. Section 3 describes a framework designed to measure CMFs in a real-world environment, along with the framework's subsystems under construction. Section 4 describes new difficulties encountered along with possible solutions. Section 5 concludes the contributions of this article.

[1] Available from `http://www.nagios.com`

2 Common-mode failures

Fault-tolerance in distributed systems and peer-to-peer networks is normally attained by adding redundancy to the system. Avižienis and Kelly [1] divide redundancy into the three domains of *time* or repetitions, *space* or hardware, and *program* or software. The textbook definition by Tanenbaum and van Steen [10] differs with regards to the domains: their definition divides *information*, *time*, and *physical* redundancies. In both definitions, redundancy can be applied in different measures to multiple domains, meaning that a general $XT/YH/ZS$ fault-tolerant system could perform $X-1$ redundant calculations in time, on $Y-1$ hardware components, with $Z-1$ program codes.

Regardless of what the actual domains are, the definition of common-mode failures remains the same. A single common-mode failure can manifest in multiple redundant repetitions or replicas. This ability can negate the benefits of redudancy altogether, provided that the replicas are similar. Similarity in time manifests naturally as the extra calculations are executed by the same program code. Similarity in software or hardware means that the redundant replicas are either copies of the same software, or hardware devices carrying similar (potential) defects. Schroeder et al. have defined similarity in this contect more formally as the *autocorrelation* [6] of failure events. Roughly, a single component failure with a high autocorrelation factor sharply increases the probability of similar components failing in the near future. This probability then diminishes with time.

Presently, common-mode hardware and software failures are reported by the media as failures of entire model series of hardware or software components [2–4]. These reports show that autocorrelated failures do occur, but they are still rare enough to warrant attention from the media. On the other hand, ≪classical ≫or *independent* failures have become so well-known that they receive attention only in the most severe of cases. Thus, the research questions that we have set to figure out are as follows

1. What is the proportion of CMFs to independent failures?
2. What types of CMFs are most common?

The first question stems from the hypothesis that current real-world systems are more vulnerable to CMFs than predicted. The rationale behind this is that a homogeneous environment seems to be easier to maintain than a highly diverse or heterogeneous one. Additionally, subjective factors seem to gravitate real-world systems towards homogeneity. It is common for system administrators to possess intuitive arguments like ≪machines from vendor A are better than those of vendor B ≫or ≪systems from vendor C should be avoided because they are so unreliable≫.

The second question explores the possibility that CMFs may result more naturally either in software or hardware environments. CMFs in software may be the result of vulnerabilities or flaws in the program code and thereof resulting updates or upgrades. In hardware devices, CMFs may be the result of common manufacturing, shipping, handling, or administration practices.

Ultimately, the purpose of the ongoing research is to prove the conjecture that diversity is beneficial to the fault-tolerance of distributed systems and peer-to-peer networks. In this context, diversity is defined as the intentional introduction of heterogeneity to a given environment. The main drawback is that diversity is notoriously difficult to formally model or express in simulations. Hence, we turn our attention to a real-world environment, and describe its benchmarking in the next section.

3 The Nagios sentinel service

The Nagios measurement framework or is a robust and mature combination of *checks* to be run against *services* and the core logic that handles check intervals and interleaving. As a *sentinel service*, Nagios is able to function as the operative component of self-healing systems [11]. For this work, the initial interest is on Nagios' error-detection and -reporting capabilities only. However, all experience gathered thus far seems to support the idea that administrators commonly perform similar investigative and restorative tasks. Assuming proper authentication and authorization, some of these tasks could well be delegated to an automated system.

In contrast to laboratory models, the object we are monitoring is the Computer Science Department's environment of infrastructure and research services. This means that the set of services is highly heterogeneous, employing both long-lasting and critical services, like e-mail and NFS, but also experimental and transient projects used by a few researchers only.

Along with those network services which are more or less always available, we are also interested in transient services that are online for the duration of a single project only. Conversely to network services, hardware components usually have no monitors to be read remotely. For monitoring components like CPU temperatures, PSU voltages, or fan rotation speeds, we employ a distributed daemon running in the local operating systems. The following subsections describe a new tool designed for the discovery of transient network services and the distribution of checks for services without a network component. Finally, service dependencies are explained as a concept orthogonal to but closely resembling CMFs.

3.1 Mapping with Nmap

The basic idea of using network mapping with Nagios is simple. Any local subnets are scanned through with a well-behaving network discovery tool, e.g., Nmap[2], and the output is then processed into Nagios' configuration files. Any changes occurring in the environment cause Nagios' configuration to slowly become outdated or stale. After the configuration becomes too stale, another scan is performed and the processed configuration files replaced. Currently, we try to scan every three to six months. At the end of each cycle, around 20-50 services have simply disappeared or been reassigned to different hosts.

Automatic discovery of network services and their configuration on Nagios is an old concept, and it has been implemented in multiple addons. Conversely, Nagios' core FAQ list makes a remark that actively

[2] Available from `http://nmap.org`

discourages automatic discovery of services [12]. The reason why we have both embraced automatic discovery despite the official view and built yet another tool for this purpose is two-fold.

First, along with services, network administrators themselves come and go. In a recent anecdote, a long-term administrator rediscovered a physical server he had set up a year ago for a specific task which had but recently become active. Specially with robust installations, it is not uncommon for a service to become interesting only after it has experienced a (partial) failure. While doing automatic discovery, we have found services which nobody had missed – yet. Our rationale is that it is better to discover too many services and then reduce the monitored set to a reasonable size, than to try and fail to assemble a complete list.

Second, we did perform an extensive search for existing open source utilities built for the matter at hand. Specifically the Nmap2Nagios and Nmap2Nagios-ng projects were very useful at first. Unfortunately, Nmap2Nagios has not been developed in seven years. Nmap2Nagios-ng[3], the successor, has been updated more recently, but the program comments were also rewritten in German. For small changes, the language barrier was surmountable, but when more extensive changes became necessary it became easier to just do a complete rewrite.

For the above reasons, we have now released the Nmap3Nagios tool. For more information, please visit the project web page[4]. Nmap3Nagios is intended to become the first tool in a new toolkit designed primarily to monitor diverse environments. The next utility under development is a visualization aid that transforms Nagios' archived log files into input presentable by the Timeline widget[5].

3.2 Monitoring local devices

In difference to the already released tool, the implementation of local device monitoring is still undergoing testing. Nagios provides at least three major alternatives for doing local checks. In order of increasing complexity, the alternatives are the check-by-ssh plugin and the addons NRPE and NSCA. For more information, please see the manual entry on distributed monitoring [13]. Trying to anticipate requirements that will appear later on, we have chosen NSCA as the distributed component.

[3]-ng probably stands for "next generation".

[4]Nmap3Nagios Home: `http://www.cs.helsinki.fi/u/pervila/Nmap3Nagios/`

[5]Timeline widget: `http://www.simile-widgets.org/timeline/`

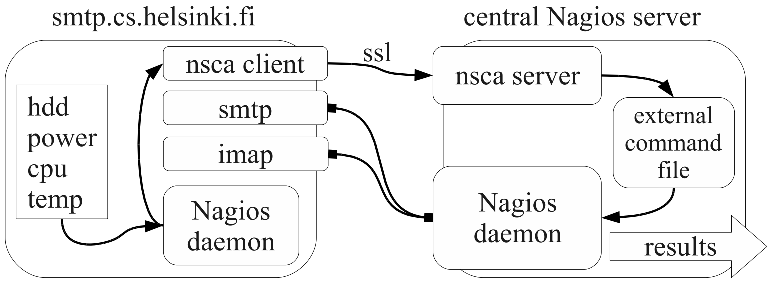


Figure 1: *Distributed monitoring of local components. Conversely with network services, local components are checked by a locally-running Nagios instance. Output is encrypted and sent through the network to a corresponding NSCA daemon, which communicates with the core Nagios process through the external command file.*

Authorization issues are perhaps the essential difficulty of distributed monitoring. Accessing local devices administration interfaces does require quite low-level privileges on our current operating systems. This means that the distributed component must be carefully given the right privileges and preferrably none but them. These requirements might be conflicting when, for example, monitoring local hard drives S.M.A.R.T. reports is only possible with root-level access to the device, but that access also enables writing to any file system block.

The benefit of using NSCA is that it does not require input from the central Nagios server. Communication flows from the clients to the central server, which makes restricting unwanted access significantly easier. In addition, all communication is encrypted with standard, SSL-based cryptography libraries. Figure 1 visualizes the communication process.

A common enough question concerns why local monitoring performed by an elaborate distributed setup are preferrable over the standard solution of using SNMP. The reason for this is that not all devices have SNMP components; using local checks enables the monitoring daemon to run any program code available. This is a considerable advantage, for some server equipment can only be monitored through proprietary software utilities.

3.3 Service dependencies

A vaguely related and quite similar concept to CMFs can be found in Nagios' software and hardware dependencies. Intuitively, it is easy to understand service dependencies through the example of switches. Server environments are usually built in hierarchical star-shaped topologies where multiple host computers are connected to a hardware switch, and the switches are then interconnected through a spanning-tree protocol (STP). If the Nagios monitoring daemon now recides on a host connected to switch A and some services are served by a host connected to switch B, each service is defined as *dependent* on both[6] switch B and switch A. This means that failures encountered in either switch will result in the dependent services' state being marked as *unknown*, not as having failed. Simply put, nothing more can then be known about the services, because another failure makes more thorough checking impossible.

Without this feature, it would be quite impossible for Nagios to distinguish CMFs from independent failures causing communication loss to a set of services. The main difference to CMFs is exactly this uncertainity: as the service state can not be known, we have chosen to err on the side of caution and ignore failures possibly masked by dependency failures. As infrastructure malfunctions are luckily quite rare, there remains a possibility of investigating these events case-by-case, perhaps adding discovered causes to the results manually.

With only 19 switches in the department network, our dependency graph is still of a manageable size. Keeping scalability in mind, this is an area where further research will most certainly be necessary. Tools that can form a topology from network addresses do exist, and will be needed if the network size grows. However, there are still more pressing issues in the work queue.

4 Ongoing problems

Hardly any research project finishes without discovering additional problems to be solved. Some of the issues described in this section were accurately predicted at the very beginning of the project. For example, the user interface issues described below have been well-known for quite a while, and multiple addon packages have tried to incorporate enhancements. Others, like the reconfiguration problem of booting multiple operating systems, exist on the boundary of what is feasible to monitor using Nagios. Finally, some problems were simply excluded by limiting the observed set of services. Monitoring the department's set of laptops was one of these, for even though a distributed monitoring setup using VPN might have been possible, the potential for false positives was judged too high.

[6] Nota bene: service-to-host dependencies are implicitly handled by the core logic.

4.1 Planning and communication

Perhaps the major problem in the measurement setup stems, unsurprisingly, from the human component. It is easy to understand that any administrator loathes redundancy in her workload, for it is vexing to upkeep the same configuration in different places. There are limits to how much paperwork any worker can be expected to reasonably fulfill. It is possible that with issue tracking, e-mail correspondence, the monitoring setup, and finally, making the actual changes, we are approaching the limit of an administrator's patience.

We have tried to reduce the amount of configuration tasks by taking the burden of maintaining Nagios from the administrators. This means that one of the researchers has worked part-time for the benefit of the IT team and implemented the monitoring framework on behalf of them. This solution seems to have worked reasonably well. The benefit is that by integrating closely with the IT staff, the measurements are done with a much better background knowledge of IT processes.

On the other hand, the integration is not perfect. There is always a small delay between noticing a failure report and then investigating what the root cause is. In many of the cases, the cause is a failure-preventive task executed by one of the administrators. These effects are difficult to completely classify as either service failures or just glitches with the measurement system. The current solution is to mark all reports consistently as failures, on the basis of the assumption that both independent and CMFs are then similarly measured. Any additional information received from the IT staff is also recorded case-by-case. The final output data will contain both the measurements and background reports investigated.

As a more long-lasting solution, we are constantly evaluating Nagios user interface addons. The idea in this is that a more tempting GUI would make it easier to receive input directly from the administrators.

4.2 User interface issues

Not all service failures do ever get repaired. For some events, the fix may be performed by moving the service to another host, merging the service with another existing service, creating a new service, or simply removing the old service altogether. In these cases, the failure is usually acknowledged and/or further notifications surpressed. If manual pruning is not regularly performed, the number of these obsolete services can make distinguishing new failures difficult.

Unfortunately, the core Nagios web GUI does not contain any reconfiguration features. An extensive number of GUI addons have been released, and we have evaluated almost every one of them[7]. For a partially complete listing, see [4]. With the exception of the Centreon[8] project, most of the addons have a very similar *binding* problem. Typically, each addon incorporates the core Nagios' reporting features and adds generative configuration facilities. This means that a new GUI can be used to define templates and individual services, and once their configuration is complete, the entire configuration is output and Nagios' core daemon restarted. The binding problem revolves around this duality of reporting and configuration, as the reporting interface has been left as is. This means that when a failure has been reported, an administrator can view the problem report from the reporting interface, but must look for and reconfigure the service from a separate interface. This flaw seems trivial to begin with, but causes a severe hindrance in practice.

Very recently, two projects with novel approaches for the GUI have been announced. One of them is the NINJA project[9] written by a group of dedicated software developers. Another is the upcoming Nagios XI[10] release offered by a team closely related to Nagios' core. As of writing, these two are the projects that we have least experience with, but are working on testing both as soon as possible. The main hindrance to evaluating GUI addons is their configuration complexity. Centreon is a prime example of this, as the software is primarily sold as a support service.

[7] As of writing, many of the open source projects have long since been abandoned.

[8] Centreon: `http://en.doc.centreon.com/Main_Page`

[9] NINJA: `http://www.op5.org/community/projects/ninja`

[10] Nagios XI: `http://library.nagios.com/library/products/nagiosxi/`

4.3 Delicate instrument

For the analysis of results, Nagios is a demanding system. The reason for this is that Nagios' check plugins can be very delicate. Even with reasonable definitions of failures, e.g., at least five-minute downtimes, Nagios can collect a surprising amount of data. Failures normally invisible to both users and administrators surface regularly, specially those that occur during the hours of least usage. For example, backup schedules and automated software upgrades may give cause for a service to be taken offline for a few minutes in the small hours of the night.

Typically, users are blissfully ignorant of these service windows, while Nagios remains ever vigilant. The downtime is a problem only in the sense that failure severity depends only indirectly on daytime, while Nagios approach is somewhat based on it. Also, some of our researchers and students keep working hours which might be classified as odd. Therefore, we have eschewed silencing failure notifications based on classical working hours, at the cost of suffering a somewhat more chatty monitoring setup.

4.4 Reconfiguration

Currently, the most difficult problem to solve remains the automatical reconfiguration of monitored services at run-time. More specifically, the problem occurs in at least two very different cases. It is encountered in workstations with multiple operating systems (OSs), and it is also encountered with network devices able to reconfigure pathways when necessary.

For multiple OSs, reconfiguration must be done when a different OS is rebooted. Each of the department's workstations runs a local distribution of Linux as the default operating system. Additionally, a significant number of workstations is also installed with Microsoft Windows XP. In our classroom configuration, any student is able to change the OS at the start of her session. This freedom of choice requires the distributed component of Nagios to be configured twice: once for each OS. Our prototype versions of NSCA are based on the default Linux configuration. For Windows, the system is somewhat more complex as the NSCA component is not exactly the same program. We are trying to make the change transparent to the central Nagios service, so that reconfiguration would be necessary only at the client end. This would enable the central Nagios to reuse the same service and host names, bypassing reloads of the central configuration.

Using a spanning-tree protocol (STP) for network switches is more or less the *de facto* standard for any larger ethernet segment. STP provides a major benefit as the switches are able to renegotiate alternative pathways should an interconnected link fail. But STP also causes difficulties for any services depending on the switch pathways, i.e., all services not on the same host as the central Nagios itself.

As of writing, no solution for STP reconfiguration has been found. In theory, the reconfiguration step should be based on an informed topology change, which would seem to require getting information from the switches. This means that configuring the initial network dependencies and STP reconfiguration should optimally be processed by the same utility. The proprietary nature of the hardware devices make this less than trivial, but the problem is somewhat mitigated by its rarity. Currently, the network dependencies are based on the default interconnections, and changes to the topology are resolved manually.

5 Conclusion

Despite the number of problems encountered, and that this article describes a measurement project in its early phases of implementation, the research questions presented seem feasible to answer. The failure monitoring service that we have devised is clearly not a fire-and-forget solution, but one that requires constant nurturing. This effect seems to be in line with the predicted difficulty of administering truly diverse environments, however, and not a fault of the Nagios sentinel service.

At this point, common-mode failures seem very much real and not just an academic exercise. Later publications will present measurement results attained, along with the actual data for further analysis. Even though many of the details are in flux, and not many conclusions have yet been made, research in this field seems both inspiring and necessary for the well-being of computer systems and networks to come.

Acknowledgments

The IT staff of both the Department of Computer Science and the Helsinki Institute of Information Technology (HIIT) for their patience.

Bibliography

[1] A. Avižienis and J. P. J. Kelly, *Fault Tolerance by Design Diversity: Concepts and Experiments* Computer, vol. 17, no. 8, IEEE, Aug.1984, pp. 67–80.

[2] W. Gruener, *Nvidia GPU Failures Caused By Material Problem, Sources Claim* Tom's Hardware US, 26.8.2008, `http://www.tomshardware.co.uk/Nvidia-GPU-failure,news-29026.html`

[3] *Malware suspected of 'Black Screen' issue* BBC News, 1.12.2009, `http://news.bbc.co.uk/2/hi/technology/8388253.stm`

[4] A. Moses, *IBM GXP Problems: The Truth Exposed and Possible Fixes* TweakTown, web site, 12.2.2002, `http://www.tweaktown.com/articles/180/ibm_gxp_problems_the_truth_exposed_and_possible_fixes/index.html`

[5] B. Schroeder, G. Gibson *The Computer Failure Data Repository (CFDR): collecting, sharing and analyzing failure data* Proc. 2006 ACM/IEEE conference on Supercomputing, ACM, New York, NY, USA, 2006, p. 154.

[6] B. Schroeder, G. A. Gibson, *Understanding disk failure rates: What does an MTTF of 1,000,000 hours mean to you?* ACM. Trans. Storage (TOS), vol. 3, no. 3, ACM, New York, NY, USA, 2007, pp. 8.

[7] Schroeder et al., *DRAM errors in the wild: a large-scale field study* Proc. SIGMETRICS '09, ACM, New York, NY, USA, 2009, pp. 193–204.

[8] B. Schroeder, *Empirical System Realibility* web site, 2009, `http://www.cs.cmu.edu/~bianca/project_pages/project_reliability.html`

[9] Marius, *10 Nagios Web Frontends* MDLog:/sysadmin, The Journal Of A Linux Sysadmin, web site, 2008, `http://www.ducea.com/2008/01/16/10-nagios-web-frontends/`

[10] A. S. Tanenbaum, M. van Steen, *Distributed Systems, Principles and Paradigms* Prentice-Hall Inc., Upper Saddle River, New Jersey, 2002.

[11] M. A. T. Pervilä, *Using Nagios to monitor faults in a self-healing environment* Seminar on self-healing systems, 2007, `http://wiki.nagios.org/index.php/Miscellaneous`.

[12] H. van der Kooij, *Autodiscovery* Nagios FAQ Database, 2007, `http://support.nagios.com/knowledgebase/faqs/index.php?option=com_content&view=article&id=52&catid=35&faq_id=347&expand=false&showdesc=true`.

[13] E. Galstad, *Distributed Monitoring* Nagios Core Version 3.x Documentation, web site, 16.6.2009, `http://nagios.sourceforge.net/docs/3_0/distributed.html`.

Chaotic map model of packet error processes in NanoLOC

Alexander S. Rumyantsev[†], Aleksei L. Chukharev[‡]

Karelian Research Centre of the Russian Academy of Sciences

Pushkinskaya St., 11, Petrozavodsk, Republic of Karelia, 185000, Russia

E-mail: [†]`ar.psoid@gmail.com` [‡]`alex3185@yandex.ru`

Abstract

Recent researches have highlighted the presence of heavy tails and long range dependence in wireless environment [1]. Works are dedicated to developing new approaches in existent protocols to achieve best Quality of Service and to minimize the effect of heavy-tails on system performance [2]. Our work is dedicated to using heavy-tailed approach in wireless networks based on the NanoLOC technology. A chaotic map is built based on the measurements taken from NanoLOC environment. (A connected work for IEEE 802.11 environment is [3]) Chaotic maps achieve modeling accuracy and are among the best models for heavy-tailed processes, despite their parsimony. Their parameters have clear intuitive meaning and provide flexibility to manage the resulting process.

Contents

1 Introduction

Heavy-tailed distributions nowadays play an important role in network analysis. An increasing number of papers discussing their presence in this or that network environment are submitted. Presence of heavy-tails in physical, biological, social processes and their evidence in contemporary computing systems has for several years focused the attention of scientists in applied mathematics, especially, probability theory.

Among possible consequences of presence of heavy-tailed data in the system may be long range dependence of the process (revealing itself in divergence of autocorrelations) which lead to undesired buffer overflows and other nasty consequences [4], that have clear impact on network design. And as a number of sources increase, the aggregated process may become self-similar.

Two main approaches arise when speaking about suitable models for long range dependent data. These are *Fractional Brownian Motion* models and *dynamical system* models, e.g. chaotic maps. The former one is powerful and parsimonious enough (i.e. it needs only one essential information, Hurst parameter), but unfortunately has some drawbacks, e.g. it needs to be used «offline»for sample generation. While the latter one is a fast, flexible and parsimonious way to generate «long range dependent like»data, that mimics a real heavy-tailed ON-OFF source and suits many of the real network data properties [5]. Moreover, chaotic maps as models for network sources provide an easy way to generate sources with various distributions from exponential to Pareto-like heavy tailed with infinite mean, and even give a possibility to model a non-stationary process «online».

We have chosen chaotic map models to fulfill the gap in the wireless NanoLOC technology based network analysis. Our goal was to provide a simple yet effective model that could estimate the properties of wireless packet transfer error process. An experiment was held, and it's results are discussed below.

2 Intermittency map as ON-OFF model

2.1 Definition

Intermittency map is a private case of dynamical system models based on one-step map iteration: $x_{n+1} = f(x_n)$. The intermittency map's orbit

$\{x_n\}_{n=0}^{\infty}$ follow the iteration law of the form [6]

$$x_{n+1} = \begin{cases} \varepsilon_1 + x_n + \frac{1-\varepsilon_1-d}{d^{m_1}} x_n^{m_1}, & x_n \in (0,d) \\ -\varepsilon_2 + x_n - \frac{d-\varepsilon_2}{(1-d)^{m_2}}(1-x_n)^{m_2}, & x_n \in [d,1). \end{cases} \tag{2.1}$$

Parameter values are bounded: $1 \leqslant m_{1,2} \leqslant 2$, $0 \leqslant \varepsilon_{1,2} \ll d$. When $m_1 = m_2 = 1$ and $\varepsilon_1 = \varepsilon_2 = 0$ one gets a Bernoulli shift map that generates a random variable uniformly distributed in the $[0,1]$ region:

$$x_{n+1} = \begin{cases} \frac{x_n}{d}, & x_n \in (0,d) \\ \frac{x_n-d}{1-d}, & x_n \in [d,1). \end{cases}$$

One may connect an indicator variable series $\{y_n\}$ with the iterations $\{x_n\}$ by the rule

$$y_{n+1} = \begin{cases} 0, & x_n \in (0,d) \\ 1, & x_n \in [d,1). \end{cases}$$

As a result, one gets an ON-OFF process $\{y_n\}$ associated with an intermittency map (2.1).

The ON and OFF periods, containing consequential ON and OFF times, are connected through the map, hence not independent. In order to remove this dependence one may use a Random Wall map [7], which is an intermittency map with the following condition: when the iterate $f(x_n)$ crosses the border value d (e.g. $x_n \in (0,d)$, $f(x_n) \in [d,1)$), the next iterate x_{n+1} is defined as an uniformly distributed random variable from the appropriate region ($[d,1)$ in the example).

2.2 Properties of the process

The phenomena of intermittency manifests itself as an alternation between two sufficiently different states: a «burst»and a «laminar phase». During the laminar phase the trajectory of the system varies very slowly, while during the period of bursts the system sharply changes. Under sufficient conditions this phenomena leads to long range dependence in the sense of autocorrelation divergence. One of the possible causes of long range dependence is the low speed of trajectory changing in the laminar phase.

In case of intermittency map, the slow varying of orbit $\{x_n\}$ leads to long periods of either ON or OFF sequences $\{y_n\}$. It's necessary to notice some asymptotic relations within this context. Let the orbit $\{x_n\}$ enter the current OFF state at the point x_0. The rightmost point of transition to ON state is $x = d$, hence the map should pass through the $[x_0, d]$ region. As the increment of trajectory is $x_{n+1} - x_n = \varepsilon_1 + c_1 x_n^{m_1}$, the number of iterations L spent in the current state may be approximated by integral [6]

$$L = \int_{x_0}^{d} \frac{dx}{\varepsilon_1 + c_1 x_n^{m_1}}.$$

The lower and upper bounds for the length of OFF periods are given by two special cases, $m_1 = 1$ and $m_1 = 2$:

$$L_1 = \frac{1}{c_1} \ln\left(\frac{c_1 d + \varepsilon_1}{c_1 x_0 + \varepsilon_1}\right); \tag{2.2}$$

$$L_2 = \frac{1}{\sqrt{c_1 \varepsilon_1}} \left(\arctan d \sqrt{\frac{c_1}{\varepsilon_1}} - \arctan x_0 \sqrt{\frac{c_1}{\varepsilon_1}}\right). \tag{2.3}$$

Similar equations arise in the ON state. For $m \in (1, 2)$ the maximum length of consequential ON or OFF states is in-between the two values, L_1 and L_2 defined by the formulas (2.2), (2.3). As we see, ε sets the maximum value of consequent ON's and OFF's generated by the intermittency map. When $\varepsilon_1 \to 0$ we have $L_2 \to \infty$, hence, an intermittency map is able to generate ON and OFF periods of arbitrary length.

Setting $\varepsilon_{1,2} = 0$, one could also use the following asymptotic relation to determine the probability ON or OFF region has length l [6]:

$$P(l) \sim l^{m/(1-m)}, \quad l \to \infty.$$

Hence the tail asymptotic for the length of ON or OFF period is

$$\overline{F}(l) = P(L > l) = \int_{l}^{\infty} P(l)\, dl \sim l^{1/(1-m)}, \quad l \to \infty. \tag{2.4}$$

As one could mention, when $m \to 1$ the tail has an exponential rate of decay, while for $m = 2$ we see a heavy-tailed Pareto-like distribution with infinite expectation, $\overline{F}(l) \sim l^{-1}$. Hence, depending on the value m, the map is able to generate the ON's and OFF's, which lengths may be distributed exponentially, or may have heavy-tailed distribution with given moment properties.

When $m = \max\{m_1, m_2\} \in (1.5, 2)$, the autocorrelation of the ON-OFF process $\{y_n\}$ diverge [7]. The Hurst parameter is predicted by the following relation (provided by $\varepsilon_{1,2} = 0$):

$$H = \frac{3m - 4}{2m - 2}.$$

For $m \leqslant 1.5$ the value of Hurst parameter is $H = 0.5$, determining there is no long range dependence, and the resulting process $\{x_n\}$ is a simple white noise. Otherwise, $H > 0.5$ detecting the presence of long range dependence in data.

3 Parameter estimation for Random Wall map

The task of synchronous fitting the model parameters, $m_{1,2}$, $\varepsilon_{1,2}$ and d to the best of our knowledge is still open. Therefore we have to use some ad-hoc assumptions and probably fit the desired model later using numerical techniques.

3.1 Method I: through the tail properties

One possible way to define the model coefficients is to set $\varepsilon_1 = \varepsilon_2 = 0$ and use the asymptotic given in (2.4). Denote $\{Z_n\}_1^N$ the lengths of consequent y_n's in one state (either ON or OFF). Build empirical tail probability function with the help of indicators:

$$\overline{F}(x) = P(Z > x) = \frac{\sum_{i=1}^{N} I\{Z_i > x\}}{N}.$$

Fit the $\overline{F}(x)$ with x^α, i.e. use the least square method for $\ln \overline{F}(x)$ versus $\ln x$. And use the relation given in (2.4), $\alpha = \frac{1}{1-m}$, hence

$$m = \frac{\alpha - 1}{\alpha}.$$

Here α is the slope of regression line of the points $\big(\ln Z_n, \ln \overline{F}(Z_n)\big)$. The parameter d may be set to the value $d = 0.5$.

3.2 Method II: through the length of active-silent periods

One could examine the behavior of the points $\left(\ln Z_n, \ln \overline{F}(Z_n)\right)$ and decide, are they Pareto-like (is there a visible line on the plot) or not. Also one could plot $\left(Z_n, \ln \overline{F}(Z_n)\right)$ and decide, are they Exponent-like (if there's a line on the plot). In the Pareto case one could set $m = 2$ and determine the relevant parameter ε through the relation (2.3). Noting that, in terms of the length of OFF period L

$$\max L_2 = \frac{\pi}{2\sqrt{c_1 \varepsilon_1}}$$

Setting $d = 0.5$ and approximating the value $c = \frac{1-\varepsilon_1-d}{d^2} \approx 2$, one gets the desired ε estimate

$$\varepsilon_1 \approx \frac{\pi^2}{8\left(\max\{Z_n\}\right)^2}.$$

Otherwise one may set $m = 1$ and use the (2.2) relation. Here we have similarly

$$\max L_1 = \frac{1}{c_1} \ln\left(\frac{c_1 d + \varepsilon_1}{\varepsilon_1}\right)$$

Approximating $c_1 = \frac{1-\varepsilon_1-d}{d} \approx 1$, we see

$$\max L_1 \approx \ln \frac{0.5 + \varepsilon_1}{\varepsilon_1}.$$

Hence the desired estimate:

$$\varepsilon_1 \approx \frac{0.5}{e^{\max\{Z_n\}}}$$

One may get the equations for the ON case likewise.

3.3 Method III: through the workload

If both ON and OFF periods look like exponentially distributed variables, one may set $\varepsilon_1 = \varepsilon_2 = 0$ and $m_1 = m_2 = 1$ and use the property of this Bernoulli shift map: it's iterates $\{x_n\}$ are uniformly distributed over $[0, 1]$ region, hence, they will visit the OFF region $[0, d]$ with probability d. One may count the number of ON iterates in data, N_{ON} and the number of OFF times N_{OFF} and set

$$d = \frac{N_{OFF}}{N_{ON} + N_{OFF}}.$$

4 Experiment

4.1 Plan of study

The aim of our study was to explore the suitability of Random Wall map for modelling the packet error process in the NanoLOC environment. We derived some data in the form of ON-OFF process, consisting of successes (ON's) and errors (OFF's) in transmission. We used the three methods of fitting Random Wall model parameters discussed above, and determined which one suits our data best by measuring the following two characteristics:

- Hurst parameter of the source and the model output, measured with the help of R/S statistics;
- Log-Log plot of the tails for ON and OFF periods for the source and the model output.

A brief sketch of our experiment is presented on Figure 1.

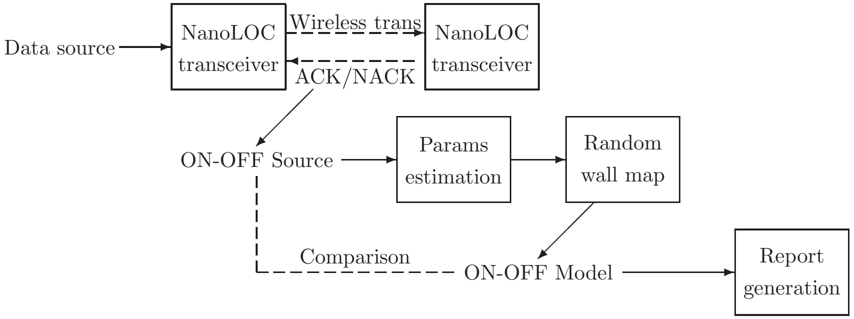


Figure 1: *Scheme of the experiment*

4.2 Experimental details

We used wireless modules designed by lab127 team. The device contains master controller unit (Atmega 644) which manages wireless transceiver. Wireless transceiver is nanoLOC (produced by Nanotron) working in the unlicensed ISM (Industry Science Medicine) radio band 2.4 GHz. The maximum bitrate of wireless transmission supported by nanoLOC is 2 megabits per second.

There were two devices distanced by about 4 meters, first one («master») connected to PC through the COM-port, and the second («slave») just awaiting for packets in the air. Devices were programmed in a following way: master sent 128 byte packets filled with random values to slave with a low level of radiation. After receiving a packet from master, slave checked the packet's CRC and in affirmative case sent ACK-packet to master with radiation level set to maximum — 100 mW (in order to reduce the possibility of ACK packet to be lost).

Hence master module could determine the success of the transmission. If master got ACK-packet from slave after transmission it put the value «1»to COM-port of a PC. Otherwise it put «0» value. As a result of experiment we had an ASCII-file containing about 600000 «0-1» values. During the experiment conditions of environment were not changing, and there were no working units that could influence radio transmission of the devices.

4.3 Results and discussion

The Hurst parameter for data source was equal to $H = 0.853597$. The estimated parameter values for the three methods mentioned above are listed in the Table 1. Log-log plots for ON and OFF periods estimated tails from three models and source data containing $\ln x$ for x-axis and $\ln \overline{F}(x)$ for y-axis are presented in Figure 2.

As a result we see that Method III was the most successful one to fit the ON periods data, while it failed to get the OFF period structure well. Nonetheless, we see thus that the ON periods for NanoLOC are distributed exponentially, $\overline{F}(x) = e^{-\lambda x}$, having value of $d \approx 0.005$ hence $\lambda \approx 2000$, they have huge, yet finite expectation $E(x) = \lambda = 2000$ and variance $D(x) = \lambda^2$. That's why they are «much alike»heavy-tailed. Method II captured well the shape of OFF periods, and had a «cut-off» at the same level as the source data for ON periods. Moreover, Method II gave the most appropriate Hurst parameter close to original, while Method I

Table 1: *Parameter estimates and Hurst parameter value*

Parameter	Method I	Method II	Method III
m_1	1.52918	1.0	1
m_2	2.0	2.0	1
ε_1	0	0.0033689	0
ε_2	0	$1.94271 \cdot 10^{-7}$	0
d	0.5	0.5	0.0050194
H	0.936014	0.766072	0.685939

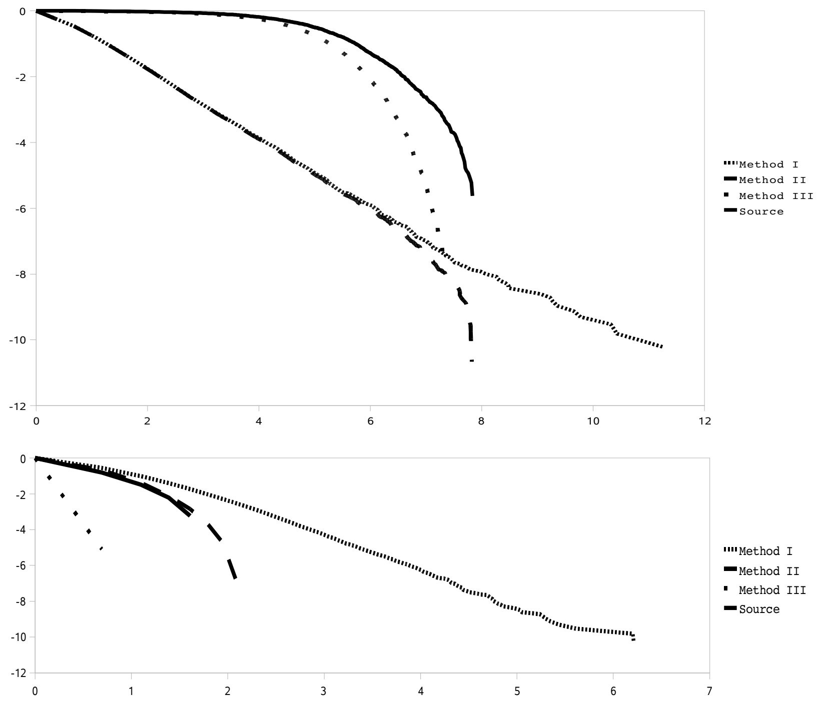


Figure 2: *The log-log plot comparison of the ON (upper) and OFF (lower) estimated tails*

overestimated it, and Method III underestimated it. After all, the overall behavior of these three methods leaves much to be desired and needs further investigation.

5 Conclusion

Three methods of ad-hoc parameter fitting for Random Wall map from the source data were introduced in this paper. They were successfully deployed in case of packet error process analysis for NanoLOC environment. Among the unexpected results we would name the absence of heavy-tailed properties in the ON periods distribution, despite the «popularity»of heavy tails across the networking environments.

As a material for the desired future work one could view these aspects:

- explore some possibilities for simultaneous mining of ε and m parameters of the map;
- investigate a way to confidence estimation of the map parameters;
- give answer to the question about stationarity of Hurst parameter and estimated tail for a given dataset.

Bibliography

[1] Z. Shao, U. Madhow *A QoS Framework for Heavy-tailed Traffic over the Wireless Internet.* Proceedings of Military Communications Conference, 2002, vol. 2, 2002. pp. 1201–1205.

[2] Z. Shao, U. Madhow *Scheduling Heavy-tailed Data Traffic over the Wireless Internet.* Proceedings of Vehicular Technology Conference, 2002., vol. 2, 2002. pp. 1158–1162.

[3] A. Körke, A. Willig, H. Karl. *Chaotic Maps as Parsimonious Bit Error Models of Wireless Channels.* Proceedings of Infocom, 2003. pp. 513–523.

[4] E. V. Morozov, M. Pagano, A. S. Rumyantsev *Heavy-tailed Distributions with Applications to Broadband Communication Systems.* Proceedings of AMICT'2007, vol. 9, University of Petrozavodsk, 2008. pp. 157–174.

[5] A. Erramilli, R. P. Singh, P. Pruthi *Modeling Packet Traffic with Chaotic Maps.* Internet draft, 1994. 33 pp.

[6] L. G. Samuel *The Application of Non-Linear Dynamics to Teletraffic Modelling.* University of London, 1999. 165 pp.

[7] R. J. Mondragón, D. K. Arrowsmith, J. M. Pitts *Chaotic Maps for Traffic Modeling and Queueing Performance Analysis.* Elsevier Preprint, 1999. 23 pp.

Modeling resources for training based on cognitive maps

Dr. Gennady S. Sigovtsev, Igor O. Semenov

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {sigovtsev, isemenov}@cs.karelia.ru

Abstract

This article is devoted to the problem of planning the table of contents of adaptive educational resources. We consider approaches to modeling this class of problems, show the application of cognitive maps to simulate the contents learning course and formulate the problem of optimal planning of content, with a limit of time and show the application of special evolutionary algorithms to solve it.

Contents

1 Introduction

Planning the table of contents (TOC) of the training resource is the standard problem to be solved by the developer training course on any discipline. It includes selection and structuring of content, and, as a rule, under the given restrictions on the volume and complexity of the material. There are the same problems in development of educational resources provided to the educational portals and sites of educational institutions, electronic libraries and collections. In most cases, TOC of such resources is the same for all users, do not take into account the intended use of the resource and the level of the user's knowledge.

Learning Resources, in which TOC or its mode, depend on some of user characteristics are called adaptive. The main point of adaptation is the ability to ≪conform≫ for purposes of training, user knowledge level or user preferences. A requirement for this is the organization of educational material in the form of hypertext with the desirable use of multimedia capabilities. This resources are adaptive hypermedia systems - adaptive hypermedia educational resources (AHMER).

2 Mathematical modeling

Mathematical formalization of the planning and content management training resource problems based on graph representation of information. For example, in [1] considers the problem of structuring content of e-learning resource, when the mathematical model of the structure represents a directed graph (tree). The nodes of the tree are the course study units, the arcs represent the ratio of hierarchical subordination of the elements of each another (i.e. relations like ≪part-whole≫). For example one study unit can require full learning of some another units. Models of this type are useful for the rational structuring content of the training resource in the form of a set of hierarchically organized units. They can be used in computer aided instruments for didactic design and context analysis by course developers.

Semantic network is a graph model and is widely used method be presented knowledge in various subject areas. The nodes of the network matched by objects and properties of subject area, its call the generic term concept. Various relationships between concepts form the arcs of the network. Among these may be hierarchical relations, and relations, which indicate the using certain concept of information related to other concepts, i.e. relations ≪previous - next≫. These relationships determine the possible variants of the concept studying, forming the content of the training resource.

Semantic network as a model for the TOC of the training resource has great expressive possibilities than a graph model ([1]), allowing express not only hierarchical ≪part-whole≫ relations between the study units, but other relations also. In addition, mathematical term graph - semantic network is a more general construction than oriented tree.

3 Cognitive maps using

Cognitive map is one kind of graph models. The basic terms of cognitive maps and semantic networks indeed very close, but in the cognitive maps, relationships between concepts are used for indicate character and measure of influence one concept to another ([5]). Educational resource domain model as a semantic network with relations ≪previous - next≫ becomes a numerical cognitive map, if this relationship will be assigned weights, characterizing the importance of knowing one concept in the study of another. Obvious that such weights can not be obtained in some formal way, but must be given by the expert or educational resource developer, reflecting their knowledge and understanding of the subject area.

In the process of designing a training course is selected from the domain model a number of concepts with their connections, forming a training course model (it is also a cognitive map, which is a sub graph of cognitive map of the domain model). A training course model should be correct cognitive map on the relation between the concept ≪previous - next≫, in the sense that the concept is included in the model course, if it contains all the concepts that precede it.

So, numerical cognitive map is a weighted directed graph. The possibilities of cognitive maps application for modeling AHMER were considered in [2], [3]. There are some numerical characteristics of cognitive maps that can be used for the analysis and design of network domain models of AHMER and for constructing algorithms for adaptation the TOC resource to the academic goals and the user knowledge level.

Relations between concepts (arcs of the graph) are presented in the form of weights $w_{ij} \in [-1;1]$, they are considered as elements of the adjacency matrix for the graph. The main task of studying maps is search indirect influence of any concept K_i on the other concept K_j, i.e. determining the ≪weight≫ of the path between these concepts, which characterized by a maximum aggregate value of the impact w_{ij}, which is denoted as V_{ij}.

Define the l-path between concepts K_i and K_j of researched map:

$$K_i \longrightarrow K_j, P_l = (i, z_1^l, z_2^l, \ldots, z_n^l, j), l = 1, \ldots, m.$$

Where m - the number of possible paths between concepts K_i and K_j. Then the influence w_{ij} of i-th concept for the j-th, is defined as:

$$V_{ij} = \max_{l=1}^{m} \Big(\prod_{p \in P} (w_{p,p+1}) \Big).$$

I.e. our task is calculate a way with maximum positive weight and a way with minimum negative weight between concepts.

If there are both positive and negative ways in the map, then special method is using - the splitting of the map. Each column and each row of the original cognitive maps of order n (with n vertices) is splitting into two columns and two rows, respectively, and the element w_{ij} of cognitive map adjacency matrix is transformed according to the following rule:

$$\textbf{IF } w_{ij} > 0 \textbf{ THEN } r_{2i-1,2j-1} = w_{ij}, r_{2i,2j} = w_{ij}$$

$$\textbf{IF } w_{ij} < 0 \textbf{ THEN } r_{2i-1,2j} = -w_{ij}, r_{2i,2j-1} = -w_{ij}$$

The remaining elements will be zero. As a result, we obtain from the matrix $W_{n \times n}$ – matrix $R_{2n \times 2n}$ with positive elements.

The splitting map has the property that the construction of it in the s-th power, its structure is preserved, i.e., positive and the negative influence concept on the concept clearly separated. Positive and negative influence concept on the concept, circulating on the path length no more than s, defined as the values of the relevant elements of the matrix $R + R^2 + R^3 + \ldots + R^s$. The obtained results are presented in a matrix consisting of positive-negative pairs of elements (v_{ij}, v_{ij}^*), that generate by the rule:

$$v_{ij} = \max\left(r_{2i-1,2j-1}, r_{2i,2j}\right)$$

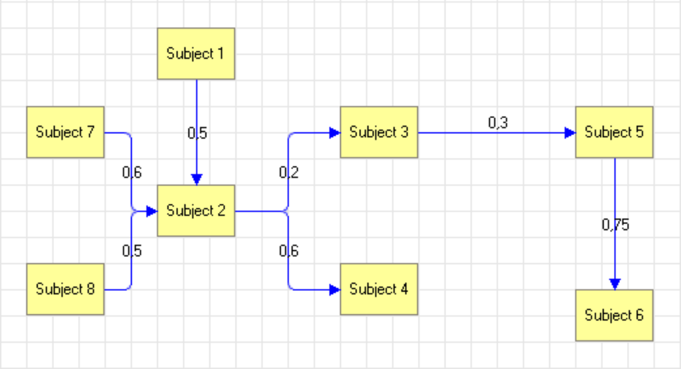


Figure 1: *Cognitive map example.*

$$v^*_{ij} = -\max\left(r_{2i-1,2j}, r_{2i,2j-1}\right)$$

Matrix consisting of such elements is called a transitively-closed cognitive matrix, its elements characterize the model:

1. The influence of one concept to another – $p_{ij} = sign(v_{ij} + v^*_{ij}) \cdot max(\mid v_{ij} \mid, \mid v^*_{ij} \mid)$;

2. The influence of the concept of the system – $\overrightarrow{P_i} = \frac{1}{n}\sum_{j=1}^{n} p_{ij}$;

3. The system influence on the concept – $\overleftarrow{P_j} = \frac{1}{n}\sum_{j=1}^{n} d_{ij}$;

These characteristics make it possible to design rational in terms of structure and content TOC of the resource, without overloading it unimportant for the course in general concepts, and organize an effective system of hyperlinks between concepts.

In the adaptation algorithm, which uses AHMER, these characteristics can be used to determine the content of the resource, give concrete choice to the user depending on the level of knowledge, fixed in his model, or dynamic changes content in the process of the user work with the resource.

Figure 1 shows an example of simple set of study units with its relationship.

4 Optimal planning problem

The task of planning the content of regular or adaptive educational resources can be given the nature of the optimization task, if input in the model the quantitative characteristic of the importance of concepts for the course. As to be maximizing the objective function in this case is the total importance of concepts included in the resource. The importance of i-th concept, v_i is equal to the sub of expert estimations of each concept for course d_i and the value of influence the concept on training resource u_i (calculated as the sum value of influence the concept on many other concepts, included in the resource).

This task is not trivial, if there are a certain set of restrictions on the concepts, making it impossible to include in the resource of all available concepts. This requires the input to the model have some quantitative characteristics of the concepts in terms of which may be formulated the constraints. The most obvious version of such characteristic – is the time spent on the study of concept – t_i (this characterization is also can be determined by expert estimation value). Then the constraint for the set of concepts included in the resource will be a requirement that total time of learning the resource does not exceed a specified value.

Thus, the formulation of the problem of content optimization training resource is:

$$\sum_{i \in M} v_i \to \max,$$

$$\sum_{i \in M} t_i \le T,$$

where $M \subset N$ – a subset of the concept numbers included in the model of educational resource. $N = \{1, \ldots, n\}$ – the set of all concept numbers of resource.

Another limitation in this problem is the requirement of the correctness of the model training resource: the concept can not be into a valid solution of the problem (i.e. can not be included in a possible model for resource) without any regard to including other concepts. So, the concept may be included in the solution only with with all other concepts, which correspond to these nodes of cognitive maps, which get out path, that's leading to the node of the concept.

By the last condition, this task does not belong to a class of linear programming. (Even if the importance of the concept does not depend on the influence value on the concept resource as a whole, i.e. the assumption that $v_i = d_i$.)

5 Genetic algorithms using

To solve this problem, we are proposing special method based on genetic algorithm. Genetic algorithm (GA) – a heuristic search algorithm used for solving optimization and modeling by successive selection, combination and variation of the unknown parameters using the mechanisms reminiscent of biological evolution ([4]).

GA can find quasi-optimal solutions of complex combinatorial-logical problems on graphs. As a fitness-function, we have chosen this function – $F = \sum \alpha(c_i)_M + \beta(d_i)_M - \gamma(e_i)_N$

The coefficients α, β, γ – are the algorithm parameters, i.e. user can change it. The set M is the set of study units, which we choose. N – contains units that we do not include. c_i – expert values for all units which we choose at each iteration. d_i – influence value on the course for all units, which we choose at each iteration. Computed at each iteration from scratch, because at each iteration we obtain different maps. e_i – penalty value – influence of not included units on included. Due to this penalty, we expect that the optimum population will not contain such units, which are dependent on many not include in final set other units.

6 Experiments

For testing and validation of described technologies, was developed a special computer system for cognitive maps analysis. The system allows:

- Creating, storing and editing of Cognitive Maps (CM) of various types;
- Calculation of different parameters of cognitive maps – the mutual influence, consonance, dissonance, etc;
- Pulse simulation for prediction the behavior of CM in the during the time;
- Optimizing cognitive maps using genetic algorithms.

Figure 1 shows the screenshot of the program.

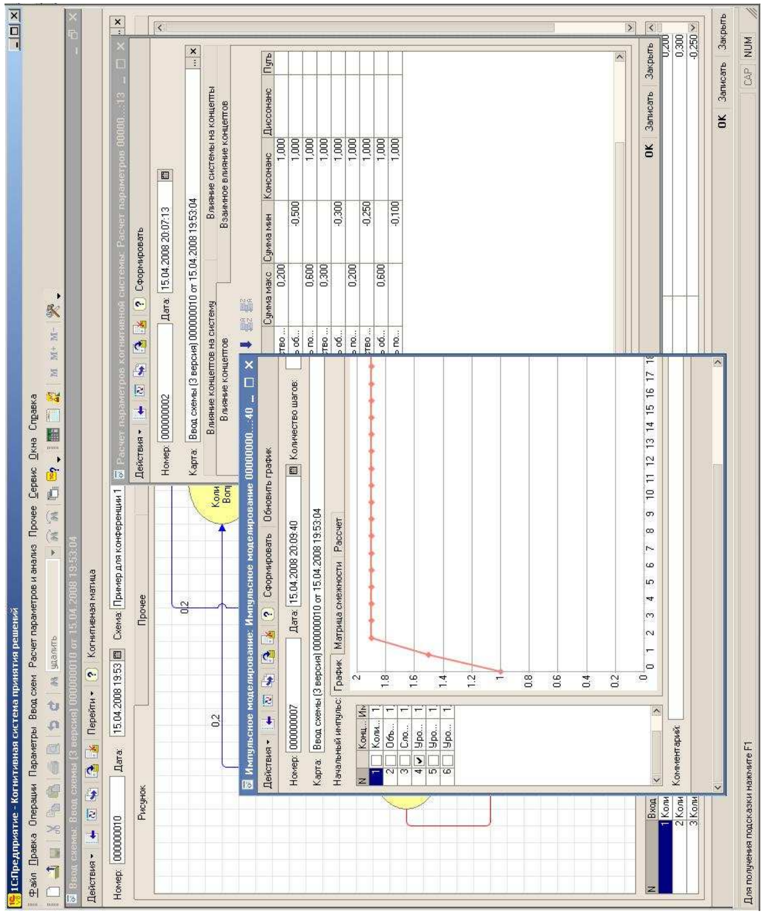

1С:Предприятие - Когнитивная система принятия решений
Файл Правка Операции Параметры Ввод схем Расчет параметров и анализ Прочее Сервис Окна Справка
Действия
Когнитивная матица
Расчет параметров когнитивной системы
Импульсное моделирование
Сформировать
Обновить график
Матрица смежности
Расчет
График
Сумма макс
Сумма мин
Консонанс
Диссонанс
Путь
Влияние системы на концепты
Взаимное влияние концептов
Влияние концептов на систему
OK
Записать
Закрыть
Для получения подсказки нажмите F1

Figure 2: Cognitive Modeling System.

As an example to show calculations of cognitive maps we consider a simple model of the training course, cognitive map which contains both positive and negative weights. We have proposed a simple model of the training course. As the vertices of cognitive maps, modeling this course, we have selected the most important, from our perspective, concepts:

- Count of course questions;
- Student training level;

- Teacher assistance level;
- Training material volume;
- Training material complexity;
- Student ability level.

Figure 3 shows the cognitive map of simple training recourse.

This map was also analyzed. Were calculated integral characteristics (mutual influence, consonance, dissonance, etc.).

This example is quite simple in terms of interpretation of cognitive map characteristics. For example, from figure 4 shows that greatest influence on the system has a volume of training material, obviously, just that the system does not affect this concept.

As an example for the demonstration of a genetic algorithms using to consider mention the cognitive map, which is simple content model training course (1). By setting different parameters of genetic algorithm, such as: the coefficients of fitness function, the number of chromosomes in the population, the restriction on time, the probability of reproduction and mutation, etc., we were closed in most cases to a single solution, which was correct (with all units in the solution which are included, all units depend on first units are included too) and optimal (maximized fitness function).

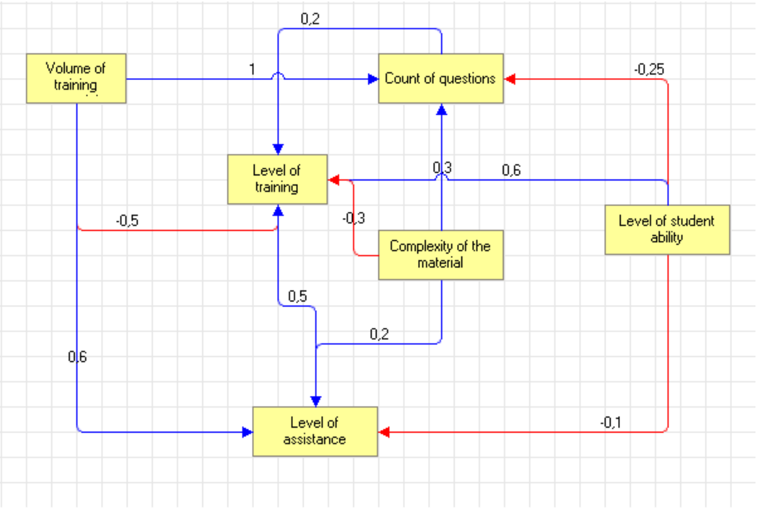
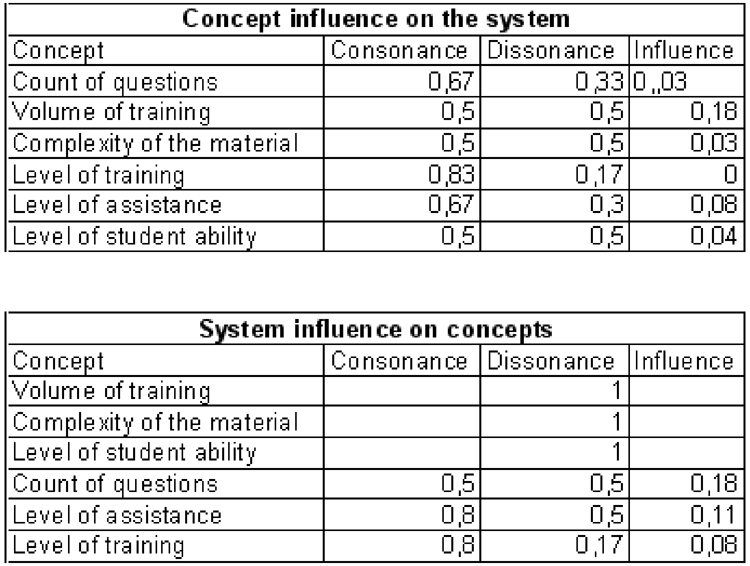


Figure 3: *Cognitive map of simple training recourse.*



Concept influence on the system			
Concept	Consonance	Dissonance	Influence
Count of questions	0,67	0,33	0,,03
Volume of training	0,5	0,5	0,18
Complexity of the material	0,5	0,5	0,03
Level of training	0,83	0,17	0
Level of assistance	0,67	0,3	0,08
Level of student ability	0,5	0,5	0,04

System influence on concepts			
Concept	Consonance	Dissonance	Influence
Volume of training		1	
Complexity of the material		1	
Level of student ability		1	
Count of questions	0,5	0,5	0,18
Level of assistance	0,8	0,5	0,11
Level of training	0,8	0,17	0,08

Figure 4: *Integral characteristics for the training course*

7 Conclusion

So, we have shown application of modeling training resource content based on cognitive maps, also, application of modeling different systems based on cognitive maps.

We have developed and implemented application of special genetic algorithms to optimize content of training resource on the importance, given the constraints of time, and, also it can be used for solving different optimization tasks in Cognitive Maps using genetic algorithms.

Computer System «Cognitive Making Decisions System» was developed and tested on different educational resources. In the future, we plan to improve our system by including module for stability analysis (using a pulsed modeling) and prediction system.

Bibliography

[1] A. V. Solovov, *Modeling the structure of electronic educational resources.* Information Technologes, vol. 2, pp. 43–48.

[2] I. O. Semenov, G. S. Sigovtsev, *Modeling resources for training based on cognitive maps.* All-Russia Conference ≪Telematika ≫'2008, vol. 2, St.Petersburg, 2008. pp. 448–449

[3] O. V. Rusanov, I. O. Semenov, G. S. Sigovtsev, *About adaptive electronic educational resources modeling.* Second All-Russia Conference ≪Informational environment of High School, XXI century ≫'2008, University of Petrozavodsk, 2008, pp. 127–130.

[4] V. M. Kureichik *Genetic algorithms.* Advanced information technologies and intelligent systems'2000, vol. 1, pp. 18–22

[5] B. Kosko *Fuzzy cognitive maps.* Int. Journal of Man-Machine Studies. 1986 vol. 24, pp. 65-75

Student Software Engineering Projects for the Maemo Platform at Petrozavodsk State University: State-of-the-Art and Perspective

Dr. Yury A. Bogoyavlensky, Dr. Dmitry G. Korzun

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {ybgv, dkorzun}@cs.karelia.ru

Abstract

This talk describes activity of Petrozavodsk State University (PetrSU) in student software engineering (SE) projects and, in particular, the branch of SE projects for the Maemo platform. We start with motivation and brief history of this activity and of its context. A general picture of SE education in PetrSU is given; curriculum, organization, scheme, etc. The Maemo activity is a result of collaboration between PetrSU and Nokia. We discuss its goals, current state, and future. Short overview of past and ongoing student SE project is presented.

The PetrSU scheme of ICT education is strongly focused on SE aspects. The introduction phase starts at school and continues for 1st and 2nd year students (Faculty of mathematics). The SE basics are carefully studied by 3rd year students in two-semester course in software engineering. It includes theoretical background in Autumn semester with team miniprojects and close-to-real projects in Spring semester. Therefore, 4th year student is ready real-life projects and she/he can be invited to participate in research and development projects at PetrSU departments, Regional Center of Information Technology, IT-park, and other ICT institutions and companies.

Some of the student SE projects in Spring semester can be supported by ICT industry. In this year (2009), we organize four such projects.

MySocials: A Client for Social Network Services with Cross-Profile Features

Kimi: A Personal Organizer in the Internet event space.

Scribo: Distributed blogs for the Maemo Platform.

MobileTrade: A Maemo Mobile Trade Client for Business Systems.

WidSets: Porting WidSets to the Maemo Platform.

These projects were very successful and we plan to continue them as real-life projects next year. Their details are published as separate abstracts in this volume.

This activity leads to modifications to ICT curricula at the Faculty of Mathematics. The Department of Computer Science conducts four new mandatory courses related to open-source and mobile programming.

1. User Interface Design with GTK/Qt (1st year students, Spring semester).
2. UNIX Programming (2nd year students, Autumn semester).
3. Maemo Programming (2nd year students, Spring semester).
4. Symbian Programming (3rd year students, Autumn semester).

As for further plans, we define the following directions for SE education improvement in PetrSU.

- More competences on open mobile programming in PetrSU SE educational courses.
- Student SE projects for the Maemo platform and other open platforms.
- Focus on the Smart Spaces concept as a new paradigm for development of distributed applications.

A Maemo Client for Social Network Services with Cross-Profile Features

Dr. Kirill A. Kulakov, Alexandr V. Borodin,
Alexandr A. Lomov, Andrey O. Bogachev,
Sergey V. Zakharov, Ilya L. Burlak

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {kulakov, aborod, lomov, bogachev, szaharov, burlak}@cs.karelia.ru

Abstract

MySocials is a mobile client to access services of social networks from a Nokia Internet tablet (Maemo platform). The client allows editing user profile data, managing contacts (friend lists), and messaging. The current version works with VKontakte, the most popular social network service in Russia, Ukraine, and Belarus.

Contents

1 Introduction

Many Internet users spend a lot of time in social networks. The latter integrate the functionality of home pages, blogs, talk boards, etc. providing comfortable hosting platform for sharing photos, videos and applications. The current trend is that social services move to open interfaces to their user databases, and client applications can interact with the service through HTTP requests (so called WebAPIs) [1].

The project "MySocial" develops a client application for such social network services with open WebAPI as VKontakte [2], MySpace [4], and Facebook [3]. VKontakte is most popular in Russia, Ukraine, and Belarus. FaceBook is very popular in Western Europe. The application is primarily oriented to mobile devices with the Maemo platform.

The client application provides cross-profile features by combining user information from different services. It allows creating user's own virtual space in a more flexible and appropriate way than known browser-based solutions. For example, a user can create groups of friends based on combined accounting data from VKontakte and MySpace, or forward messages received from VKontakte to MySpace users.

The project was started in 2009 at Petrozavodsk State University (PetrSU), PetrSU-Nokia-NSN lab on wireless and mobile technologies [6]. MySocials belongs to the family of FRUCT research projects [8]. The idea was initiated from the FRUCT Program [5], thanks to Sergey Balandin.

2 Overview of social network services

A typical social network service provides the following functionality.

Each user has own "profile" (text fields, pictures and some other data). Users can upload pictures from the service, fill additional fields about their career, Political and Religious views, Relationship status, Education, etc. Also users can be "friends" of other users, hence having access to some data of others' profiles. Social networks usually have privacy control that allows every user to choose who can view their profile or contact them.

The key communication primitives are sending and receiving messages. For example, VKontakte has private messages, wall messages, opinions, and comments. Only sender and recipient may access private messages. Wall messages is placed on the user profile page and shown as a part of user profile. Opinions are sent like private messages, but the sender hides herself (anonymous). Comments state opinions of friends.

As a rule, user has some accounts on different social network services. He/she uses the same profile data with small differences. In this case, merging contacts from different services (metacontacts, metaprofiles, and others) is be very useful. For example, the user can receive messages from different social networks into the same mailbox, view "combined" friend profile and send messages via selected service.

3 Project architecture and data flows

The application architecture is based on drivers of social network services (see Fig. 1). Each driver implements data exchange with a certain service. The local database stores data and performs transformations (sorting, searching, merging profiles, etc.).

Each module is implemented as a separate library. The internal coordination follows our original XML-RPC-based protocol (see Fig. 2). The Kernel redirects requests to the corresponding modules. This approach allows adding new drivers easily.

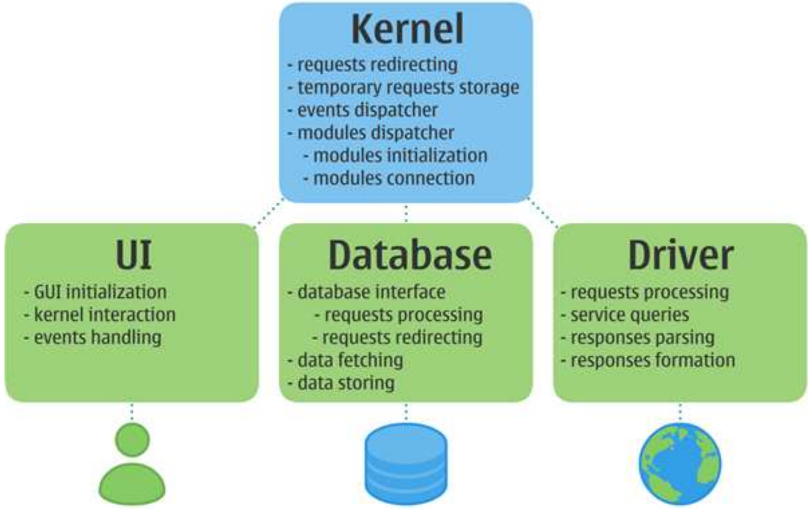


Figure 1: *The high-level architecture of a MySocial client.*

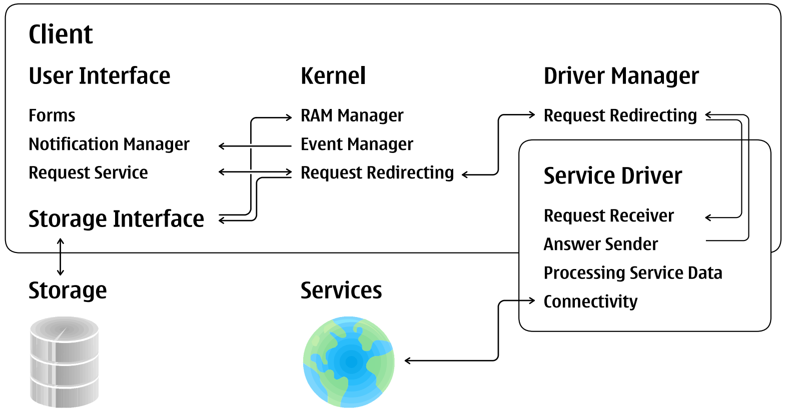


Figure 2: *Basic data flows in a Mysocial client.*

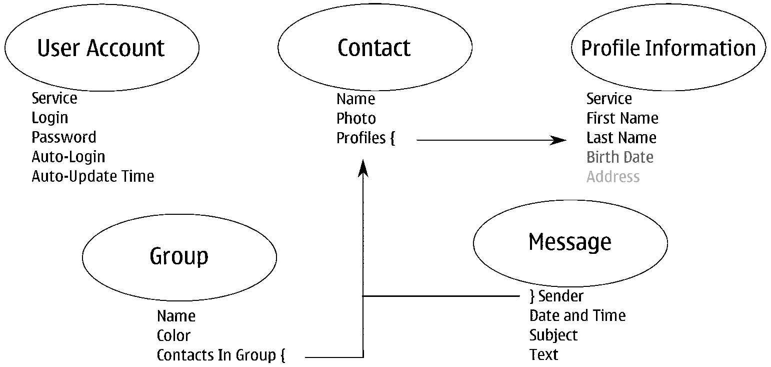


Figure 3: *Basic domain elements.*

The key data structures reflect domain elements of social networks (see Fig. 3). Each driver implements “User Account” to store username, password, auto-update time, and other related attributes. Service identifiers are provided by social network services.

Local database is implemented with RDF/OWL technology. We developed the SONET dictionary for storing user profiles locally; SONET is extension of the FOAF dictionary. OWL (Web Ontology Language) allows describing relations between domain elements (e.g., user profile fields). For example, our ontology of message is shown in Fig. 4.

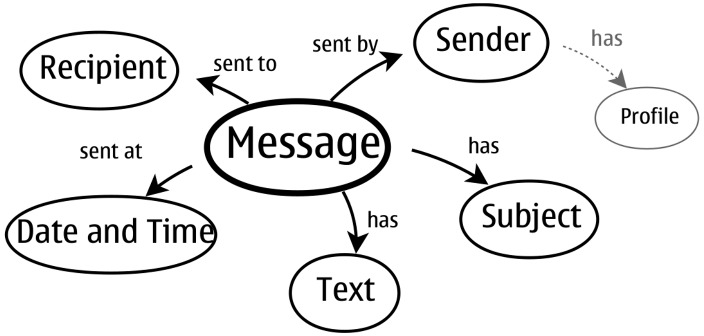


Figure 4: *The ontology of “message”.*

4 User interface

The GUI is created with the GTK+ widget toolkit. GUI objects are defined using data node types and data structure from XML descriptions.

The contact page contains a list of user’s friends as shown in Fig. 5. At this page, the user sorts friends, finds a friend, views friend’s profile, and sends messages to her/him. Also the user can merge two own profiles from different services to one metacontact.

When a user works with messages, she/he sees the message box as shown in Fig. 6. The user can find, read, write and reply. Messages from different social network services are combined into one inbox and outbox.

5 Conclusion

The project is a student team software engineering project at the Department of Computer Science of Petrozavodsk State University (PetrSU). Project started in Febrary 2009.

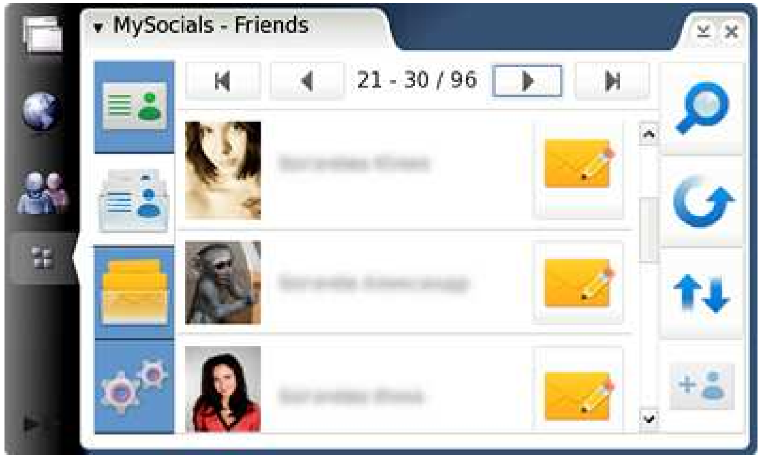
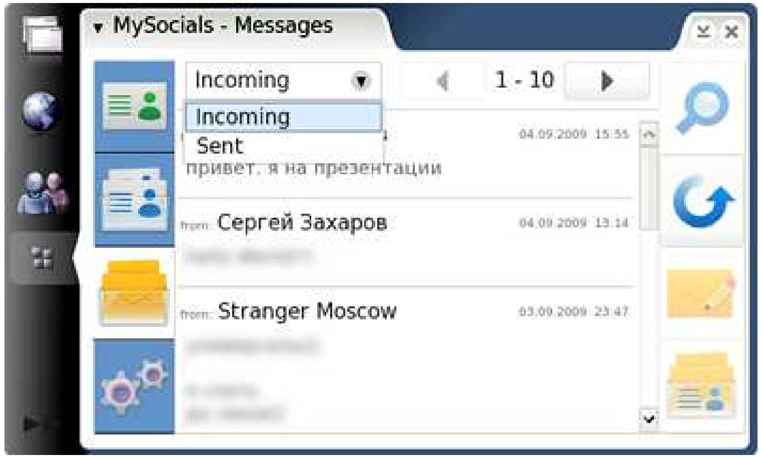


Figure 5: *The GUI form that shows a friend list.*



04.09.2009 15:55
привет. я на презентации

from: Сергей Захаров 04.09.2009 13:14

from: Stranger Moscow 03.09.2009 23:47

Figure 6: *The GUI form that shows messages.*

We have developed and implemented a prototype of client for VKontakte for Maemo 4.1 Diablo. The prototype is written in C and runnable at Nokia N8x0 tablets. It supports friend list, user profiles and messaging. More details about the project can be found in [7].

Bibliography

[1] User API: official website. `http://userapi.ru/`

[2] VKontakte: all-purpose tool for finding friends. `http://vkontakte.ru`

[3] Facebook: Connect and share with the people in your life. `http://www.facebook.com`

[4] MySpace: Official website. `http://www.myspace.com/`

[5] Finnish-Russian University Cooperation in Telecommunications (FRUCT) Program. `http://fruct.org`

[6] Open platforms for mobile devices. `http://oss.fruct.org`

[7] Project MySocials: wiki for developers (in Russian). `http://oss.fruct.org/wiki/Maemo-VKontakte`

[8] A Client for Social Network Services with Cross-Profile Features. `http://fruct.org/index.php?option=com_content&view=article&id=63%3Aclientsocialnetworks&Itemid=59`

A Maemo-based Personal Organizer in the Internet Event Space

Aleksandr S. Kolosov, Dmitriy N. Petrov,
Aleksandra I. Reiss, Evgeniy M. Tsvetkov, Ilya I. Shmarov,
Ivan A. Shterkel, Anastasiya V. Shipulova

Petrozavodsk State University

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185910, Russia

E-mail: {akolosov, dpetrov, reiss, cvetkov, shmarov, shipulov,
shterkel}@cs.karelia.ru

Abstract

This project aims at developing an application for planning and managing personal events based on different Internet services (organizer). Such events are created by a user on her own or taken from Internet (e.g., Google and VKontakte.ru services). Therefore, the organizer, in contrast to other organizers, additionally allows surfing the Internet event space with collecting interesting events for personal use.

Events are closely related to concrete persons, and the organizer maintains a contact list. Its content is typically available from the system address book, but the organizer also can exploit user profiles from social networks. Based on the contacts a user can invite and notify other persons to participate in appropriate events. The FOAF (an acronym of Friend of a Friend) ontology is used as a storage format for contact data.

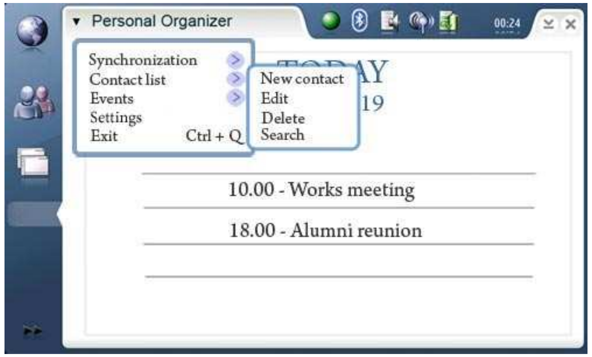
The organizer keeps all events locally. If necessary, the user synchronizes his organizer with services like Google Calendar. The current version is for the Maemo 4 platform (e.g., Nokia Internet Tablets N8x0). Moving to Maemo 5 is in progress. The screenshots below are produced on Maemo 4.

The basic functionality is the following.

- Presentation of events structurally (on day, week, etc. basis).
- Editing events. For each event the following data can be specified: date, start time, end time, name, description, location, persons invited, reminding mark, periodicity (e.g., every Monday since 01.03.2009 to 01.04.2009).
- Surfing the Internet event space. A user searches interesting events using such services like Google or VKontakte.ru. The latter provides public events that can be interesting to the user (e.g., a rock concert or a meeting of friends).
- Contact list. At this phase, the service VKontakte.ru is used to retrieve contact data of other persons (exploiting user profiles). For each record in the list the following data are kept: name, photo, home address, work address, phone numbers, fax, email, ICQ, jabber, Vkontakte ID, etc.
- Events reminder. For events with reminding marks a notification message appears on the screen with given periodicity.
- Notification of invited persons. Each invited person is notified about a coming event using the contact list data. Notifications can be sent via such services as VKontakte.ru, e-mail, etc.

Let us describe one of the basic usecase scenarios. If the user wants to create an event then she/he makes the following actions:

1. Select *Menu* → *Events* → *New event* as shown in Fig. 1.



10.00 - Works meeting

18.00 - Alumni reunion

Figure 1: *Select from menu*

2. Fill data fields for the event as shown in Fig. 2. The fields are the following.
   - Set remaindering.
   - Invite friends from the contact-list
   - Set notifications about events for certain friends
   - Send notifications
   - Save the event

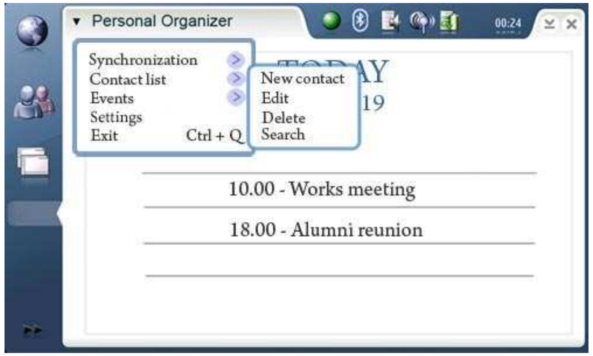
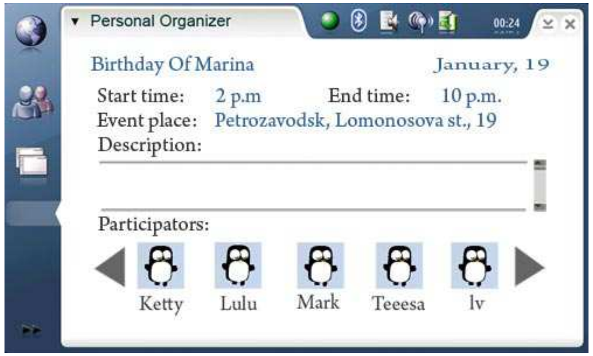


Figure 2: *Fill fields*

3. The final result is shown in Fig. 3.



Birthday Of Marina January, 19

Start time: 2 p.m End time: 10 p.m.

Event place: Petrozavodsk, Lomonosova st., 19

Description:

Participators:

Ketty Lulu Mark Teeesa Iv

Figure 3: *The result view*

A Blog Client for the Maemo Platform

Diana V. Zaiceva, Artem D. Mezhenin,
Alexandr A. Sannikov, Kirill A. Germanov,
Mikhail A. Kryshen, Dr. Dmitry G. Korzun

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: `maemo-scribo@cs.karelia.ru`

Abstract

We introduce a blog client for the Maemo platform (Maemo4), which is developed in the scope of Scribo project at the Petrozavodsk State University. We discuss the basic functions of the client, its additional features and our plans of further development. Specific attention is given to recent technologies incoming in the mobile software world; we experiment with them and exploit to implement Scribo.

Contents

1 Introduction

The Scribo project develops a Maemo application to support mobile blog users. A typical mobile device is Nokia Internet tablet from the N8x0 family. In the basic scenario, a user views and edits her profiles as well as makes blog postings. The current version of Scribo supports livejournal.com; additional blog services are in progress. In further plans, Scribo allows representing and editing information retrieved from various blog services. The information can be combined for a user to work with multiple blogs simultaneously.

The support of the following blog services are in our primary focus.

1. LiveJournal: a free blog service that includes photo storage, publishing tools, and style templates, and various configuration options.

2. Blogger is another free blog service.

3. WordPress is a publishing web platform.

2 Blogging terminology

Blog is from the "web log", "online journal or diary of events". At a given blog service, the blog is a collection of all blog discussions of a given user. Each discussion consists of regular text entries with images or multimedia.

There are the following key differences from the traditional diary. Blogs are public and they usually involve third-party readers who may publicly debate with the author. Readers make their feedback in a current blog (messages) or in their own blogs.

Account: User account at a blog service.

Profile: A structure that integrates the information from all accounts of a given user. Note that a user can have many accounts at several blog services.

Blogger: a person who has a blog (author). Collection of all blogs is called the blogosphere. A blog can be personal, group (corporate, club) or public (open).

Message: Text written by a blogger in her or other's blog. Hypertext features (images, multimedia, etc.) are typically supported by blog services.

Post: The first message, which opens the discussion on a specific topic. For simplicity, we also use term "post" to identify the whole discussion.

Comment: A message written in response to another message. That is, any discussion starts with a post; then a sequence of comments from the rest of the discussion.

Friend: A blogger that has special preferences from the user. With friends blog discussions occur frequently.

Group: Several bloggers (of the same or different services) in accordance with their interests or other criteria.

Duplication: Publishing the same message to several posts (of the same or different services) simultaneously.

3 Basic scenarios

Let a user start Scribo. The screenshots are for the Maemo5 platform.

The user logs in to a selected blog service as shown in Fig. 1.

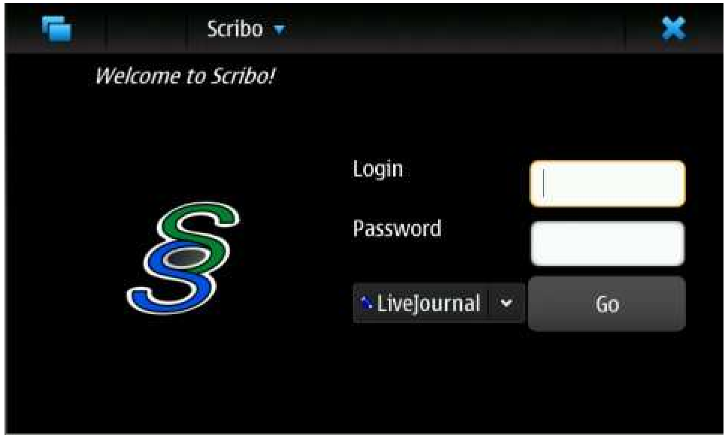
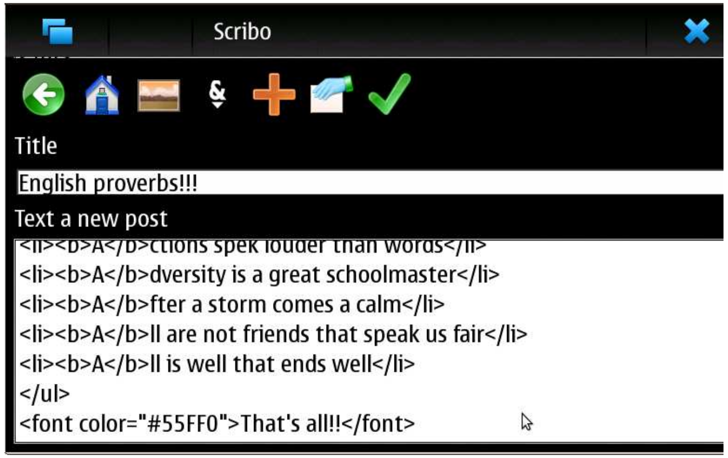


Figure 1: *Selecting a blog service and logging in.*

The user starts blogging and writes her posts with HTML tags using a simple editor, see Fig. 2. When the user needs to read her own posts then she goes to “My posts” window shown in Fig. 3. The user goes through the post list and selects that post she wants to read, see Fig. 4.



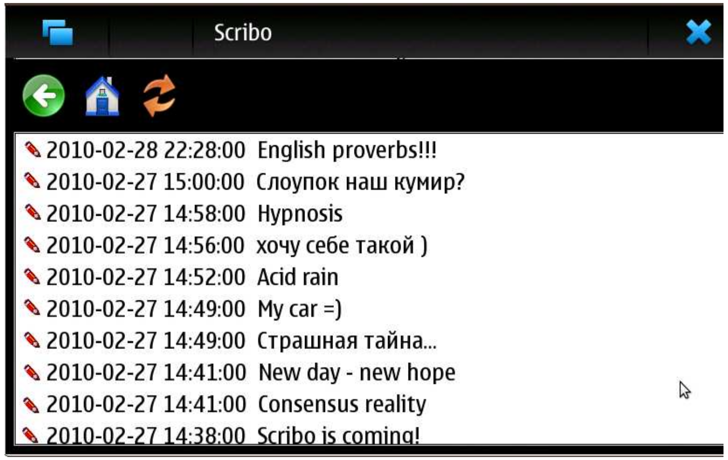
Title

English proverbs!!!

Text a new post

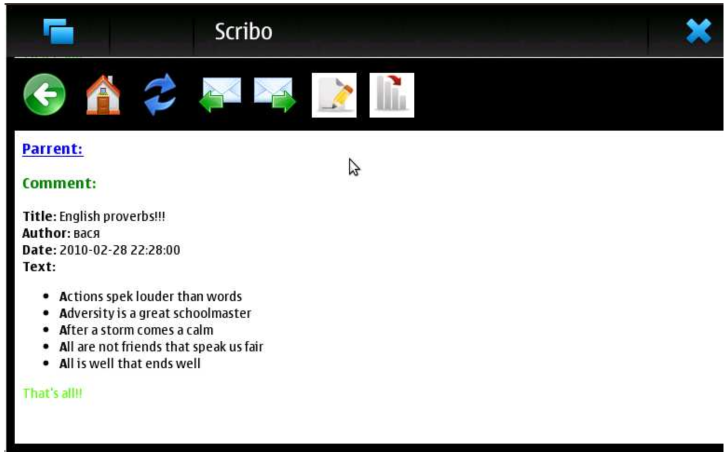
```
<li><b>A</b>ctions spek louder than words</li>
<li><b>A</b>dversity is a great schoolmaster</li>
<li><b>A</b>fter a storm comes a calm</li>
<li><b>A</b>ll are not friends that speak us fair</li>
<li><b>A</b>ll is well that ends well</li>
</ul>
<font color="#55FF0">That's all!!</font>
```

Figure 2: *Writing a post. Several basic HTML tags are supported.*



2010-02-28 22:28:00 English proverbs!!!
2010-02-27 15:00:00 Слоупок наш кумир?
2010-02-27 14:58:00 Hypnosis
2010-02-27 14:56:00 хочу себе такой)
2010-02-27 14:52:00 Acid rain
2010-02-27 14:49:00 My car =)
2010-02-27 14:49:00 Страшная тайна...
2010-02-27 14:41:00 New day - new hope
2010-02-27 14:41:00 Consensus reality
2010-02-27 14:38:00 Scribo is coming!

Figure 3: *Listing user’s posts.*



Parrent:

Comment:

Title: English proverbs!!!
Author: вася
Date: 2010-02-28 22:28:00
Text:

- Actions spek louder than words
- Adversity is a great schoolmaster
- After a storm comes a calm
- All are not friends that speak us fair
- All is well that ends well

That's all!!

Figure 4: *Reading a given post.*

Several cross-service and cross-blog features are in progress. Data retrieved from blog services allows constructing a cross-service user profile that is applicable for many services. It can combine friend relations and groups. Scribo will support a so-called distributed blog that combines posts and messages from several blogs. That is, a user works with posts as if they are written in a single service, makes duplicate posts into different services, tracks comments and participates in cross-blog discussions.

4 Technologies

The XML-RPC API implements the communication with livejournal.com. Basically, this API allows reading and writing blogger’s posts and comments.

The FOAF (an acronym of Friend of a Friend) ontology describes cross-service user profiles. FOAF is used by some blog services (e.g. livejournal.com, limited support in yandex.ru). In Scribo, we apply FOAF as a format to represent a friend list of any blog owner.

The RSS (abbreviation for Really Simple Syndication) format supports tracking posts from blogs of other users.

A Maemo Mobile Trade Client for Business Systems

Igor O. Semenov, Ingmar R. Bergmann, Denis D. Zabirohin,
Pavel Y. Andrianov, Ekaterina V. Zabolotskih,
Dr. Dmitry G. Korzun

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {isemenov, bergmann, zabirohi, andriano, zabolots, dkorzun}@cs.karelia.ru

Abstract

The MobileTrade project develops a demo C++ application for the Maemo platform to automate in a mobile way the work with a trading business system (TBS) and to exploit the distinguished mobility potential of Nokia Internet tablets, service-oriented architecture (SOA) and web services. The application allows making customer orders and sales, producing various reports, and synchronizing with TBS. MobileTrade aims at supporting various web-services in accordance with user's needs. Local data base is implemented using SQLite. User interface focuses on ergonomic and finger-touch properties.

Contents

1 Introduction

Nowadays there is a wide range of Trading Business Systems (TBS) at the software solutions market for PCs. Examples include "1C: Enterprise 8. Trade Management" (1C company, Russia) [1], SAP R\3 [2], Microsoft Dynamics [3] (former Microsoft Axapta), and Oracle E-Business Suite [4]. The mobile device extensions of these solutions, however, will focus on providing interface to the PC-like system, and likely no new business models will be available.

Typical TBS usecases assume extensive use and maintenance of a central database (CDB). A user can retrieve, process, and store business data, e.g., make orders for goods, checking goods availability, or viewing business reports. In today's business world, mobility is a typical case; those operations should be done regardless of the user location and quality of the network environment or access channel,

Recent solutions, however, are not pure mobile. Access the company TBS may be restricted within its LAN because of security reasons. Although a protected protocol like HTTPS can be used, the focus is on Internet browsers for PCs or laptops, hence a workplace is required when a user is outside. Summarizing, (i) a workplace at a third-party is insecure, (ii) deploying own secure workplaces at all points of interest is expensive, (iii) a laptop as a workplace is better than PC but it is not a pure at-hand tool compared with any mobile device (weigh and size barriers).

In this extended abstract, we present a mobile automation solution for a potentially wide range of TBSes. It exploits the features of Nokia Internet tablets (Maemo-based [5]) and Service-Oriented Architecture (SOA). We aim at combining the abilities of classical TBS (potentially a proprietary or any well know TBS system could be used as a back-end TBS) with the idea of free access of small players to the large training markets.

We focus on an approach for integrating at a tablet the TBS functionality that a user needs to work within business processes regardless of her location and poor Internet access. The MobileTrade project is not for porting known applications (e.g, PC-, laptop-, or PDA-based) to a tablet. In contrast, we propose a pure mobile solution that enhances the existing TBS features and accelerates the recent SOA-based trends in this area.

When a network is unavailable the local mirror database is used (synchronized with CBD when possible). This service is very useful for the trade agents that travel to customers and would benefit from making on-site sell agreements (e.g., PRE-selling and VAN-selling) and for individuals who want to sell or buy goods on equal rights as the large retailed companies. In the latter case, TBS implements a rendezvous service keeping seller announcements.

2 Business process and mobility

TBS includes accounting (fiscal, taxation, personnel), support for trade, logistic, and some other business processes. An example TBS is “1C: Enterprise 8. Trade Management” (1C company, Russia), which is one of the most popular in Russia [1]. Other examples can be found in [2–4]. Users are trade agents, storekeepers, managers, etc. A typical TBS maintains a central data base (CDB); users retrieve and store business data, view reports, and produce documents.

The portability of various TBS can be solved based on the Service Oriented Architecture (SOA) and web services [6]. As a result, a new network component is easily introduced for other systems. Web services use the Simple Object Access Protocol (SOAP) for communication between clients and TBS.

Let TBS include accounting module (fiscal, taxation, personnel) and support for trading, logistic, and some other business processes. The system users are trade agents, storekeepers, managers, etc. Basically in the use case scenarios, the TBS maintains a central database (CDB), and the users retrieve and store business data, view reports, and produce and consume documents.

Often the business processes require extensive traveling (outside company buildings). It would be a competitive advantage to simplify work processes if the user would have access to TBS data and could input and collect new data from clients. Nowadays it is typically collected in paper form or in text files at PDA or laptops. Hence, after returning to the office, the user or somebody else has to update the data in the CDB. This way leads to extra work, results in mistakes when reading user’s notes, and unwanted delays in the business process. There are already some mobile device oriented solutions [7], but they are restricted to a certain TBS and cannot give a flexibility that one could expect from solutions build on the SOA principles.

3 MobileTrade client

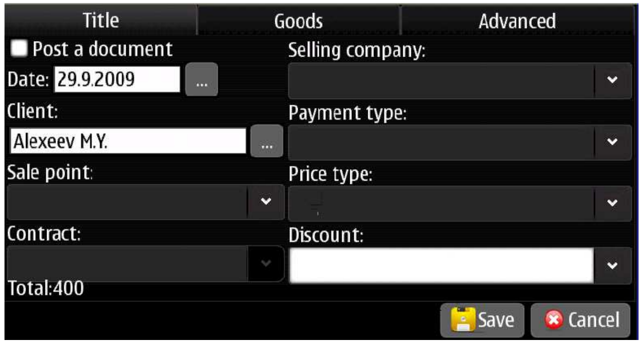
MobileTrade is a cross-platform solution. The client runs on tablet- and desktop-based devices. A web-services provides API to access the data from TBS CBD. All data that a user need for her business activity (a partial copy of CDB) are stored locally in a mobile data base (MBD).

As a result, a part of TBS functionality moves to the user side. The user retrieves and stores business data, views reports, and produces documents as if she is in the company building.

The primary user is a trade agent who visits clients (they are geographically distributed), provides them with related advertisement and other information on goods, collect clients’ orders and comments, and then forward the data to the TBS for further processing. The web-services support XML data.

Consider MobileTrade basic screenshots (N900, Maemo 5). Fig. 1 shows the order window that is used for making orders from a customer. All sales to the customer are collected and available for later use in the sales window, see Fig. 2. The user analyses the basic statistics on goods sales in the report window, see Fig. 3.

The CDB—MBD synchronization happens when needed (e.g., new data are collected from customers) and possible (e.g., Wi-Fi is available at the current location). For many functions no immediate network access is required since MBD substitutes CBD temporarily. Clearly, electricity is



☐ Post a document

Date: 29.9.2009

Client: Alexeev M.Y.

Total:400

Figure 1: *Customer orders.*

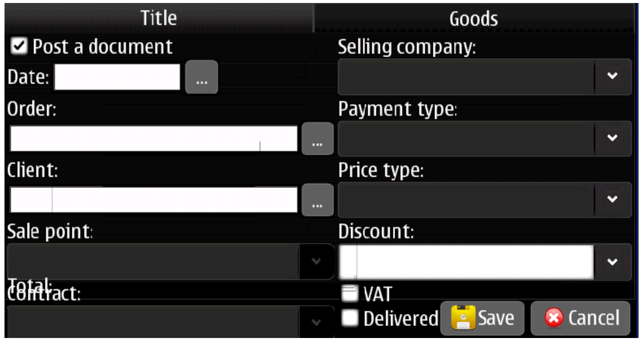
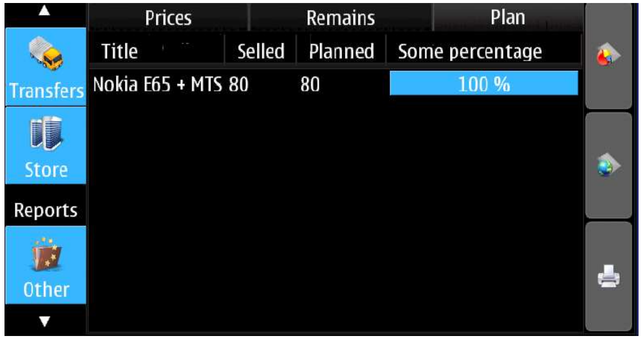


Figure 2: *Customer sales.*



Title	Selled	Planned	Some percentage
Nokia E65 + MTS 80		80	100 %

Figure 3: *Statistics reports*

available at most client locations, hence the battery is not a very restrictive factor. Therefore, our application demonstrates a new type of mobile trade clients. Note that PDA has not satisfactory capacity and features to implement such a client.

At the current phase, the project produced a demo application for the Maemo 4.x platform [5]. It acts as a proof-of-the-concept. Moving to the Maemo 5 platform is in progress. User interface is adopted to a small tablet display and mobile usability requirements. Our design is flexible and configurable to various web-services having the similar functionality.

MBD is an XML database implemented with SQLite. Web-service communications use SOAP with either an open (http) or protected (https) data transfer. Authentication is based on public/private key pair, and the OpenSSL library implements data encryption.

Although the current usecase scenario is simple, it captures the key mobility issues and allows further extensions for other types of users and functionality. Examples include making photos of clients and goods, constructing routes for visiting clients (GPS-based enhancements are also possible here), notifications from TBS about new clients, goods or special offers, immediate reading commercial codes using camera, providing information on closely located goods when a user walks in a warehouse.

Bibliography

[1] 1C: Enterprise 8. Trade Management. `http://v8.1c.ru/trade/`

[2] SAP R\3. From Wikipedia, the free encyclopedia. `http://en.wikipedia.org/wiki/SAP_R/3`

[3] Microsoft Dynamics. `http://www.microsoft.com/dynamics/`

[4] Oracle E-Business Suite. `http://www.oracle.com/us/products/applications/ebusiness/018714.htm`

[5] The Maemo community. `http://maemo.org/`

[6] Eric Newcomer. Understanding SOA with Web Services: XML, WSDL, SOAP and UDDI. Addison-Wesley, 2002.

[7] The Mobile Trading Systems Company Website. `http://www.worldtradingsystem.com/`

Porting WidSets to the Maemo Platform

Mikhail A. Kryshen, Stanislav A. Epifanov, Alexandr A. Sannikov

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {kryshen, epifanov, sannikov}@cs.karelia.ru

Abstract

The goal of this project is to provide WidSets widgets to the Maemo platform. They should be integrated nicely into Hildon user interface. There are two key problems that the project is focused on. First, creating performance-effective port of WidSets software on Maemo. Second, developing a software layer to integrate widgets into the tablet desktop environment.

The Maemo port of WidSets software uses GCJ Java-to-native code compiler, GNU Classpath (Java API implementation), and MidPath library (MIDP2 implementation). Modifications are made to these open-source libraries. The modifications are needed for compatibility with WidSets software and for integration with the Maemo platform.

Student Software Engineering Projects for the Maemo Platform at Petrozavodsk State University: Architecture and Support of Distributed

Vadim A. Ponomarev, Aleksandr S. Kolosov, Mikhail A. Kryshen

Department of Computer Science, University of Petrozavodsk

Lenin St., 33, Petrozavodsk, Republic of Karelia, 185640, Russia

E-mail: {vadim, akolosov, kryshen}@cs.karelia.ru

Abstract

Maemo is the embedded platform for Nokia Internet Tablets. Petrozavodsk State University is collaborating with Nokia on the Maemo activity conducting hands-on trainings about the Maemo platform.

This talk is about technical side of organization of such trainings. We describe the problem scope, proposed solution for home and guest trainings, and concern the issues of the installation and deployment of the Maemo development environment for many users.

Научное издание

Труды AMICT'2009

Передовые методы информационных
и коммуникационных технологий

AMICT'2009 Konferenssijulkaisu

Modernin tieto- ja viestintätekniikan menetelmätutkimusta

Proceedings of AMICT'2009

Advances in Methods of Information and Communication Technology

Том Nide Volume
11

Оригинал-макет подготовлен в системе LaTeX2e кафедрой информатики и математического обеспечения математического факультета Петрозаводского государственного университета на основе стиля, разработанного Д. Ж. Корзуном

Техническое редактирование *Д. Б. Чистяков*
chistyak@cs.karelia.ru
М. Первила
pervila@cs.helsinki.fi

Литературный редактор *Т. Н. Музалева*

ISBN 978-5-8021-1140-6
9 785802 111406

Подписано в печать 10.05.2010. Формат 60 × 84 1/16.
Бумага офсетная. Печать офсетная.
Уч.-изд. л. 10.5. Тираж 100 экз. Изд. № 109.

Государственное образовательное учреждение
высшего профессионального образования
Петрозаводский государственный университет
Отпечатано в типографии Издательства ПетрГУ
185910, Петрозаводск, пр. Ленина, 33