SmartSlog knowledge patterns: initial experimental performance evaluation

Pavel Vanag, Dmitry Korzun

Petrozavodsk State University
Department of Computer Science

This demo is supported by grant KA179 of Karelia ENPI - joint program of the European Union, Russian Federation and the Republic of Finland

11th FRUCT conference
April 23–27, Saint-Petersburg, Russia
Smart Spaces and Smart-M3

- Publish-subscribe system
- Application consists of several KPs
- Smart Space consists of SIBs (which maintain space content in RDF triples)
- KPs communicate throw SSAP protocol
KP developer can think in abstract ontology terms with SmartSlog ADK

- ADK stands for Application Development Kit
- Ontology describes with OWL (mapped to code: ANSI C or C#)
- SmartSlog uses KPI_Low library as low-level interface
SmartSlog advantages

- Simplifying KP code using high-level OWL terms
  - SIB uses low-level RDF triples
  - KP uses high-level abstractions

- Speed development of huge amount of KPs
  - Multilingual support
  - Cross-platform code generation

- Target devices could be low-performance
  - Subset of ANSI C version
  - Modest code schemes

- Space search
  - Knowledge patterns...
Knowledge Patterns: filtering

- KP storage – ”local space”
- Local objects are linked with Object Properties
Knowledge Patterns: filtering

- Knowledge Patterns is an abstract object graph (K-graph)
Knowledge Patterns: filtering

- The result object would be placed to SIB
Knowledge Patterns: searching, K-graph

- The same pattern could be used for searching objects in the "global" Smart Space
- Pattern would be mapped to RDF triples
- So Knowledge pattern would be used for searching triples

Summary:

- Filtering is used for transferring/delivering necessary parts of objects to/from the smart space
- Searching is used to deliver (search) new objects, existing in SS
Patterns search: the most complex operation

Here is a scheme how pattern based search works...
K-graph: worst-case model

Size parameters for K-graph:

1. $s_{wg}$ – number of datatype properties that every object has (graph weight)
2. $s_{wd}$ – number of object properties that every object has (graph width)
3. $s_{hg}$ – longest path from a fixed node to other nodes (graph height)
Performance KP

We developed special KP for our experiment scenario:

- Generates ontology with defined parameters
- Sends ontology
- Generates pattern with defined parameters
- Time measuring
Parameters of experiments

Lets consider RDF-triples store:

\[ N \] – the number of triples stored in the smart space

\[ N_{\text{ind}} \] – individuals

It requires:

- \( N_{\text{rdf}} \) RDF triples with facts about individual
- \( N_{\text{ont}} \) RDF-scheme triples with high-level ontology declarations (constant)

\[ N = N_{\text{ont}} + N_{\text{ind}} N_{\text{rdf}} \]

\[ N_{\text{rdf}} = 1 + s_{\text{wg}} + s_{\text{wd}} \]

\[ N_{\text{ind}} = (s_{\text{wg}}^{s_{\text{hd}}} - 1) / (s_{\text{wd}} - 1) \]
Experiments

We vary $s_{wg}, s_{wd}$ from 1 to 10 and $s_{hg}$ from 1 to 5.
Evaluation model

We measure the time

\[ T(s_{wg}, s_{hg}, s_{wd}) = b_0 \exp (b_1 s_{wg} + b_2 s_{hg} + b_3 s_{wd}) . \]

Applying multiple non-linear regression analysis

\[ b_0 \approx 11.582, \ b_1 \approx 0.034, \ b_2 \approx 5.538, \ b_3 \approx 0.388 \]

Performance-impact proportion

\[ s_{hg} : s_{wd} : s_{wg} \approx 1 : 10 : 10^2 . \]
Conclusion and Plan

Early measurements showed the basic trends

Complexity grows with size of Knowledge Patterns

Helps developer to decide the size limit of Knowledge Patterns

We plan...

- to continue this research applying other benchmarks and models
  - Measurements on every step
  - Reduce connections impact

- further focus on typical scenarios of real-life Smart-M3 applications
  - Patterns based algorithms
  - Subscriptions measurements
References

- SmartSlog developers wiki:
  http://oss.fruct.org/wiki/SmartSlog/

- Open source code:
  http://sourceforge.net/projects/smartslog/

Thank you!