A scalable distributed M3 platform on a low-power cluster

Anders Berg, Petteri Karvinen, Stefan Grönroos, Frank Wickström, Shohreh Hosseinzadeh, Natalia Díaz Rodríguez, Johan Lilius

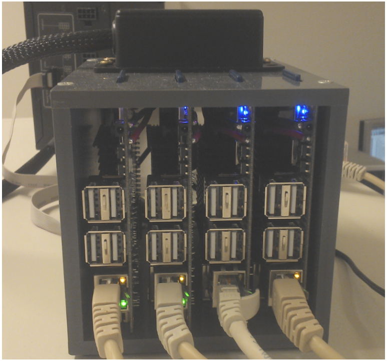
Åbo Akademi University

November 2013

Introduction

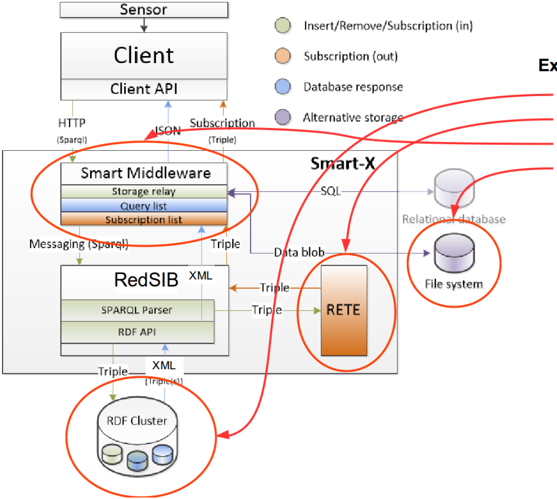
- Requirement: Small Smart-M3-based box
- In-home Smart-M3 box collects and processes
  - Sensor data (heartbeat, blood pressure, presence sensors, etc.)
  - Video streams (for example from wearable camera)
- Needs to process a large number of input data streams and handle a large number of subscriptions efficiently

The Prototype Box



- 4 ODROID-X ARM development boards
- Cluster connected by 100 Mbps ethernet
- ODROID-X Node:
  - Exynos 4 SoC
  - Quad-core ARM Cortex-A9 @ 1.6 GHz
  - 1 GB RAM
  - eMMC storage (64 GB)
- Power consumption (cluster):
  - Idle: 8 W
  - Full load: 24 W
- Prototype has a fan

Architecture overview



Extensions to S-M3:

- Clustered RDF Store
- RETE Engine
- Middleware layer
- Binary data storage

RETE algorithm using CLIPS

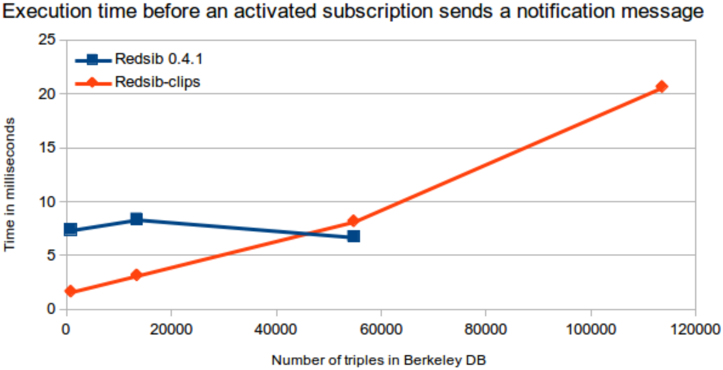
- The Publish/Subscribe paradigm one of the most useful features of the Smart-M3 platform
- The current solutions have problems regarding scalability
- We have looked in to the use of the RETE-algorithm [1] as a scalable solution
- We have implemented RDF triple publish/subscribe functionality in RedSib using the CLIPS [2] rule engine

[1] http://en.wikipedia.org/wiki/Rete_algorithm

[2] http://en.wikipedia.org/wiki/CLIPS

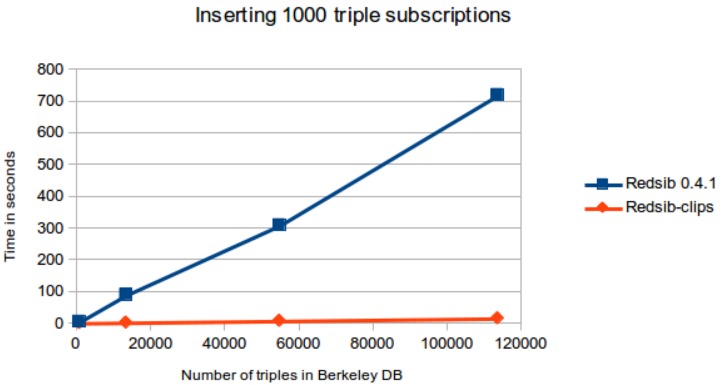
CLIPS implementation performance

- Note: Very preliminary results
- Benchmarks performed on a desktop PC
  - Not yet tested on ARM platform, but relative performance vs RedSib 0.4 expected to be similar



CLIPS implementation performance

- Benchmarks performed on a desktop PC

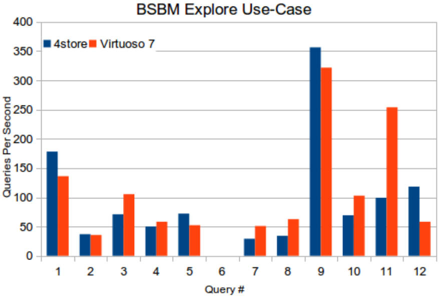


4store and Smart-M3

- We needed a distributed RDF-store to test our Smart-M3 box
- 4store is the only open-source, distributed, native RDF-store written in C
- We have written a librdf storage module for 4store, similar to the existing Virtuoso module

RDF-store Benchmarks

- Preliminary benchmarks using the BSBM[3] benchmarking tool
- Virtuoso 7 vs 4store on a Laptop (queries only)
- Performance appears quite similar between the stores on average.



[3] http://wifo5-03.informatik.uni-mannheim.de/bizer/berlinsparqlbenchmark/

RDF-store cluster benchmarks

- 1.6M triples: Performance similar for 1,2,4 nodes.
- Benchmark results at over 1.6M triples are still inconclusive.
- Some impressions:
    - Performance on one node seems to be dramatically reduced at 7M triples
        - 4 nodes at this number of triples still perform well.
    - Unstable results for one node already at 3.4M triples, while 2-4 nodes perform well.
    - Likely a consequence of the much larger amount of combined RAM in the whole cluster.
- DB also designed with 64-bit architectures in mind, while the ARM is 32-bit.

Context-aware Role-based Access Control

- Why?
  - Existing security model has only coarse-grained access control
  - We would like fine-grained control, down to the triple level
- We are planning an access control scheme using *access control ontologies* and *rules*
- Access control would be based on users' roles and context information
- Users may define their own rules for access to their data, which have higher priority than general rules defined by administrators.

Middleware

- We are developing a middleware between the Smart-M3 and users and sensors
- Modular design!
- The different parts of Smart Space will be accessible through a REST web framework
- The middleware manages Subscriptions in a **IF T**his **T**hen **T**hat [4] manner
  - IF stepmeter value over 7000(This) Then post message on Twitter(That)

[4] www.ifttt.com

Future Work

- Work still in progress on all parts of the platform.
- Further performance evaluation/benchmarking needs to be performed.

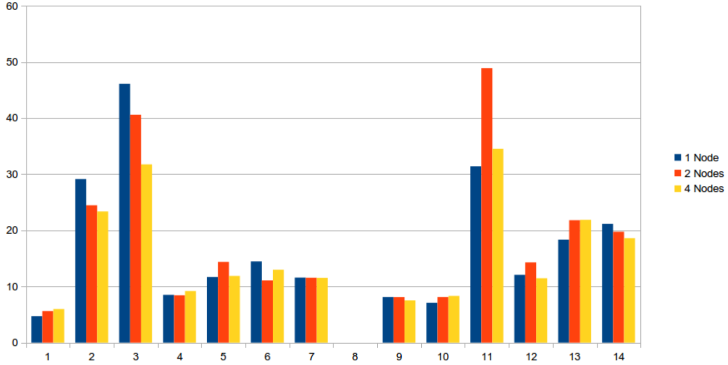
The End

Thank you for listening!

Questions?

RDF-store cluster benchmarks

- 4-store cluster performance with 1.6M triples.
- No benefit from clustering at this number of triples.


1 Node
2 Nodes
4 Nodes