New Lemon API proposal

Karol Stanisławek
IT-FIO

Outline

- Current status
  - limitations and performance
- Proposed solution
  - architecture
  - server and client implementations
  - new API
  - performance tests
- Conclusions

Current status (I)

Current implementation:

- Lemon-cli and TCL,PERL,PHP interfaces using SOAP as transport protocol
- Server (*fmonSever, OraMon*) written in C language with Oracle and flat-file back-end implementation. Each contains multithreaded SOAP server
- Libraries written in C language (MR_API) are using gSOAP implementation

Limitations:

- slow – large overhead of the current SOAP layer and its way of implementation (linked lists)

Current status (II)

Limitations (cont.):

- not scalable – both fmonServer and OraMon were not optimized for querying and data serving
- limited to about 18 000 samples; larger requests crash both *OraMon* and *fmonServer*
- authorization mechanisms are absent
- *OraMon (fmonServer)* server implements data insertion and data access interfaces, that leads to competition between clients who want to store (“monitors”) and receive data.
  - performance reduction
  - decreased scalability and reliability
- lemon-cli displays measurement data only, without any additional metadata (metadata is not present in current MR_API impl.)

Proposed solution (I)

- Separation of functionality
  - *fmonServer*/*OraMon* should only serve for data insertion purposes
  - separated data server(s)
- Using raw XML as transport protocol to organize, describe and possibly reuse data

XML example

```
<lemonxml>
   <metric id="20002">
        <meta>
            <column name="LoadAvg" type="FLOAT" length="21"/>
        </meta>
        <data node="lxb0001">
            <r ts="1121424900"> <d>2.44000006E+000</d> </r>
        </data>
   </metric>

</lemonxml>
```

Proposed solution (II)

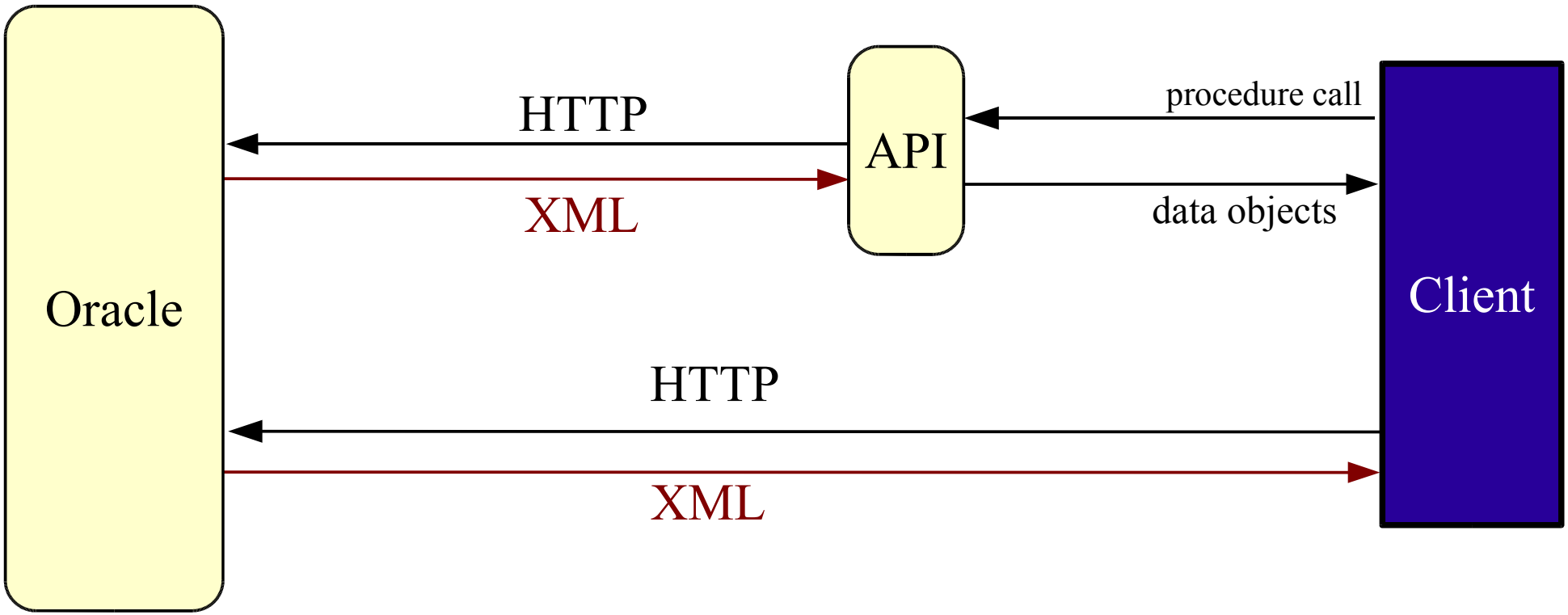
- Independent from implementation technology
- simple implementation and use
- access to service through HTTP will hide implementation and infrastructure details

  an example:
  http://pcitfio23.cern.ch:8080/lxg/?metrics=9011,9012
  &nodes=^lxb000[1-9]&end=1121427211&i=10h

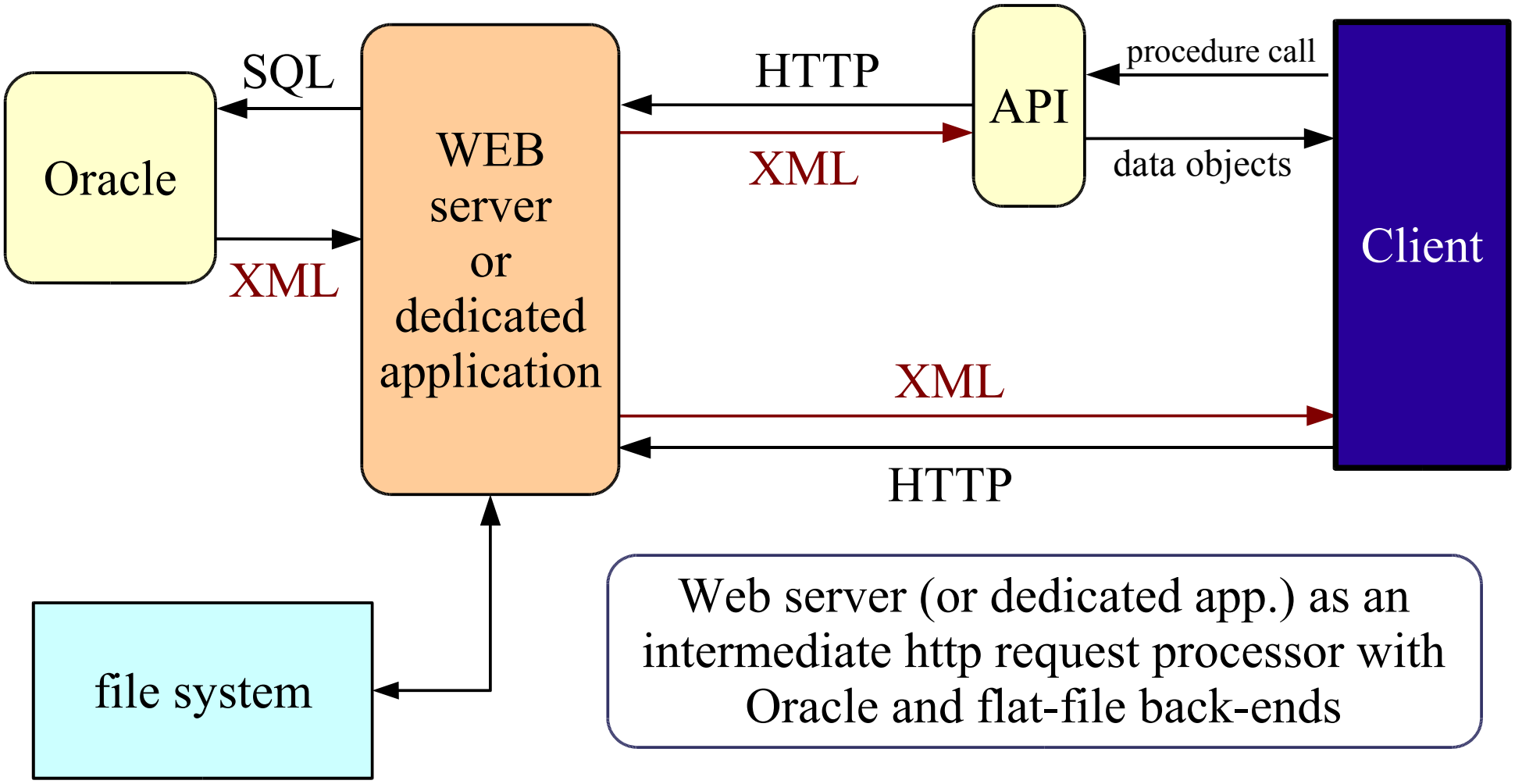
- Authorization mechanism based on public-private key schema (could be extended to **X509**.)

Server architecture (I)

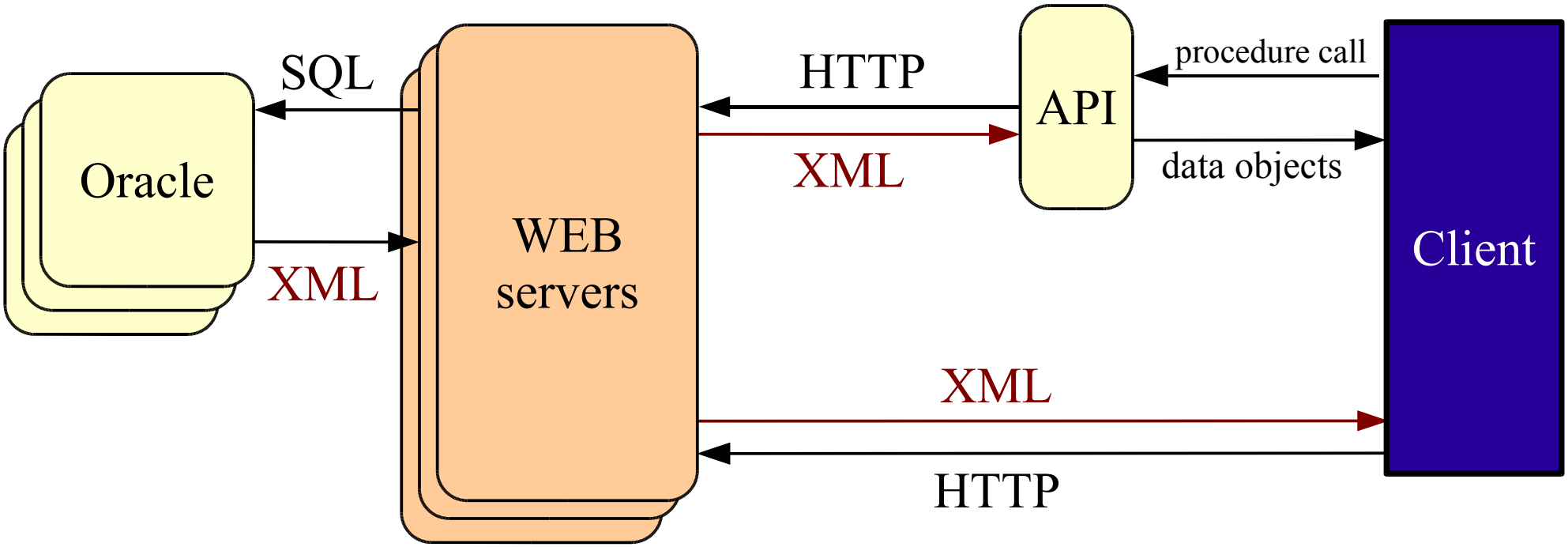
Oracle OC4J web server processing
http requests and serving XML files



Server architecture (II)



Server architecture (III)



possible infrastructure enhancements:
scalable service architecture with
fail-over and redundancy layer

Client API

Plans:

- common API based on XML schema
- libraries written in C/C++, Perl, Python and Java.
- *lemon-cli* adapted for new API
- raw XML download is possible if needed
- RSS feeds (probably)

API example (Java):

```
LemonAPI lemon= new LemonAPI();
// initialization
lemon.setSources(new String[]{"http://ccs003d.cern.ch/xml",".."});
lemon.setNodes(new String[]{"lxb0001","lxb0002"});
lemon.setMetrics(new String[]{"9101","9102"});
LemonData data = lemon.getLatestData();    //DB query and data download
String val = data.getValue("lxb0001","9101");     // local query
.....
```

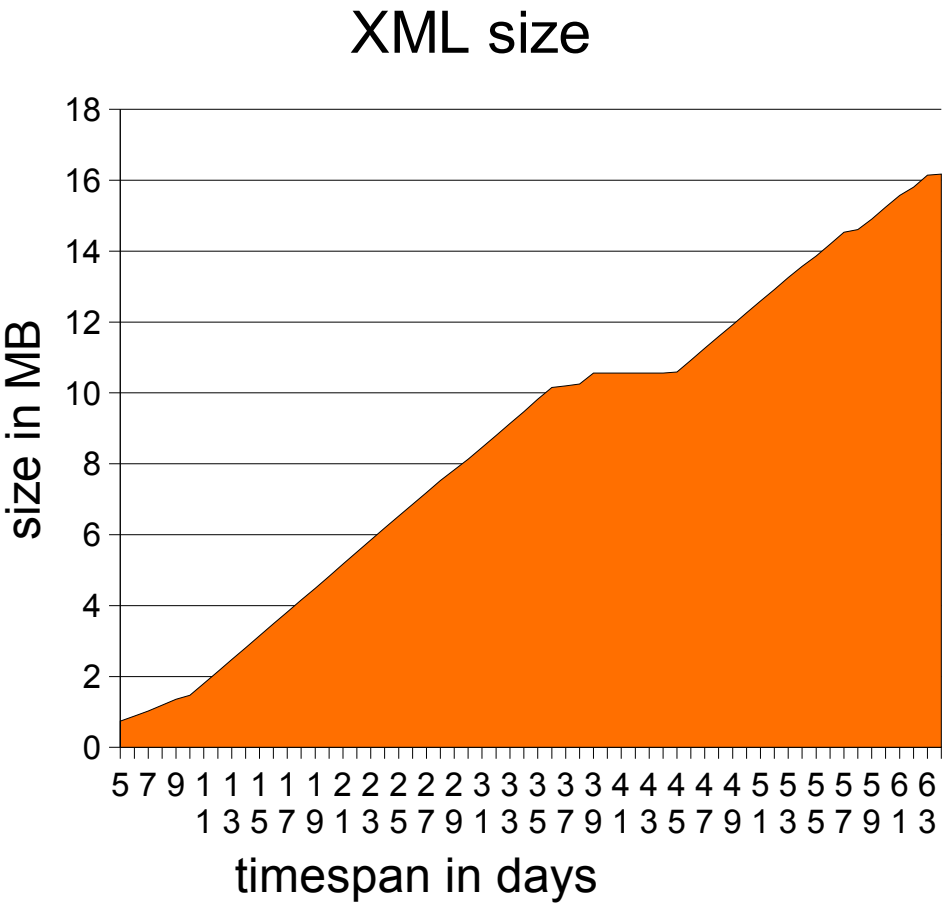
Prototype implementation

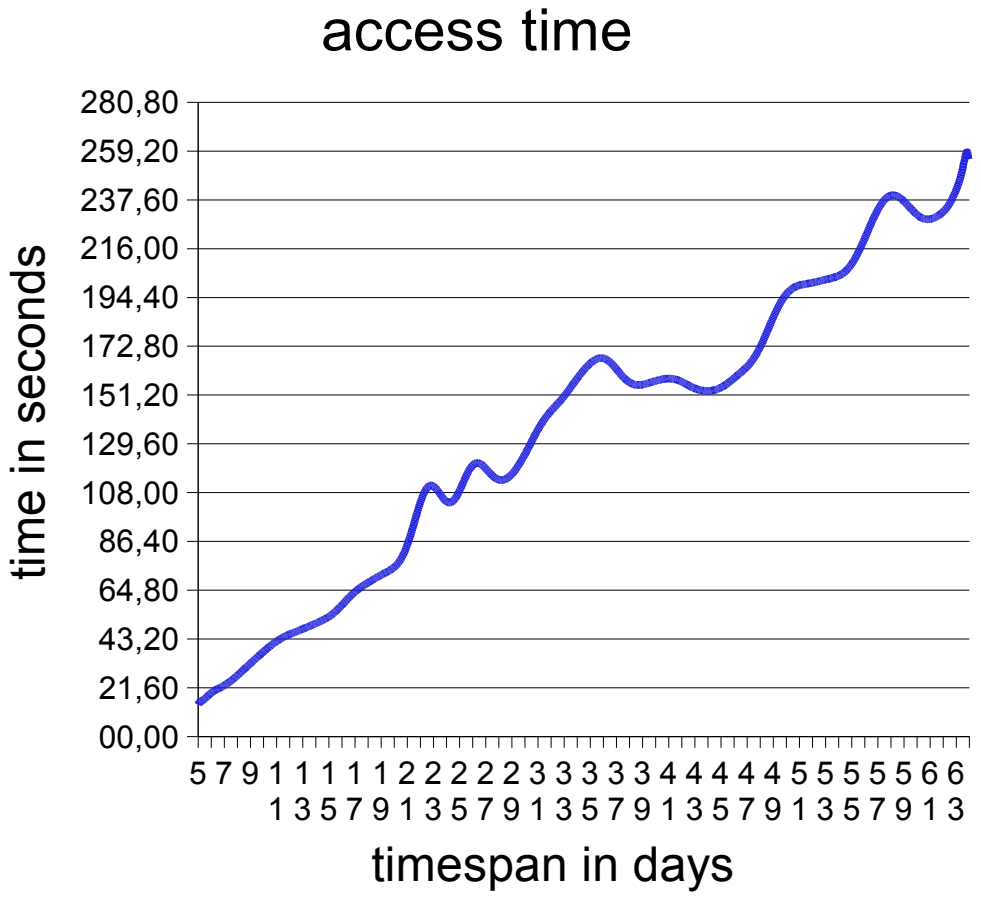
- Prototype was build using **SQL/XML**, **Spring** framework and run under Apache Tomcat application server.
- final solution will be based on PL/SQL, SQL/XML languages and **Apache/PHP** or **OC4J** server
- *Oracle* and *flat files* back-ends will be supported

Test setup:

- Oracle server (lemon3): dual P III 1GHz with 2GB of RAM
- Tomcat running on: P IV 1.6GHz with 512MB of RAM (workstation)

Data-size tests (I)

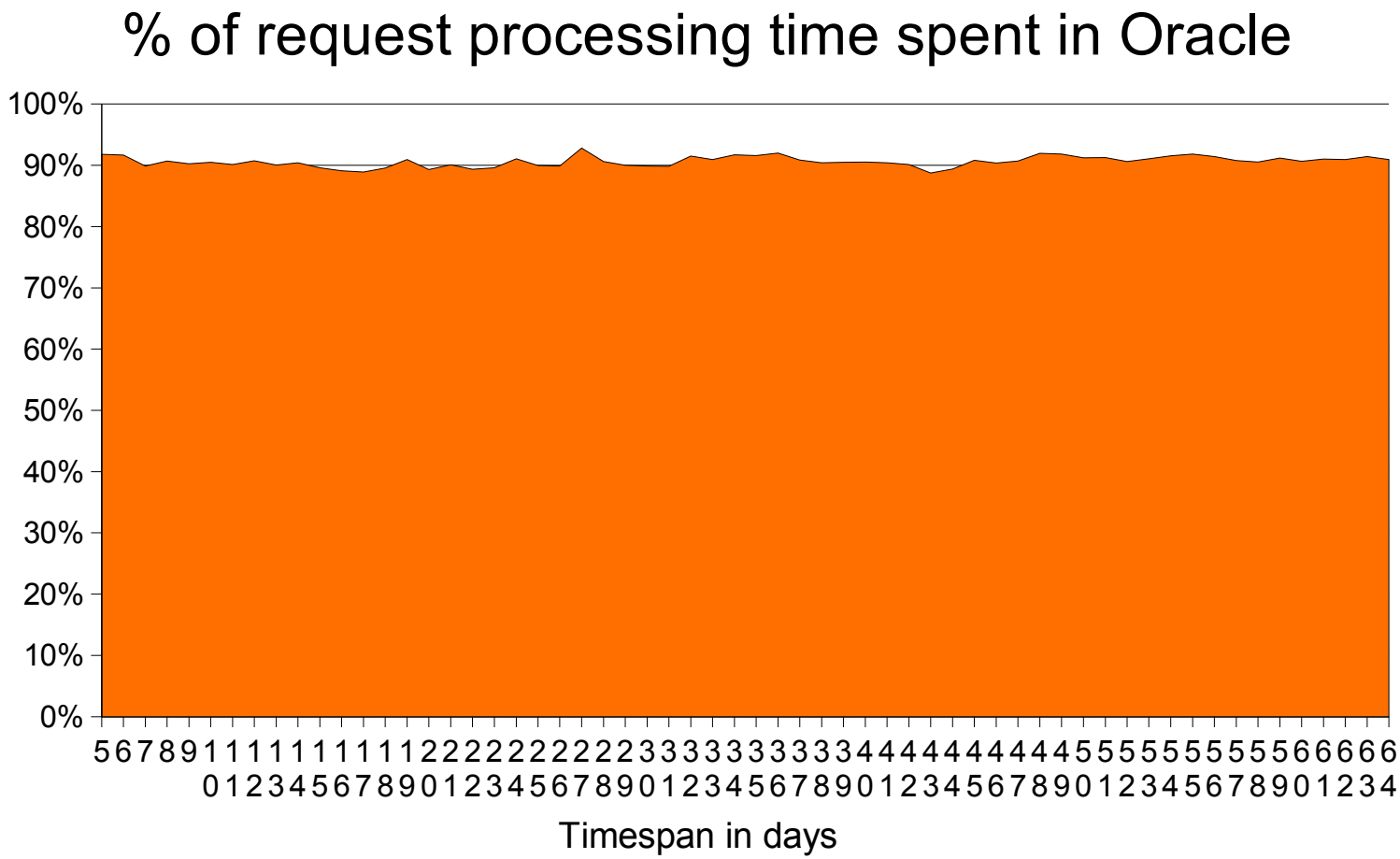




1 client, data for 9 hosts ,1 metric, time span 5..64 days
= 13k...166k samples

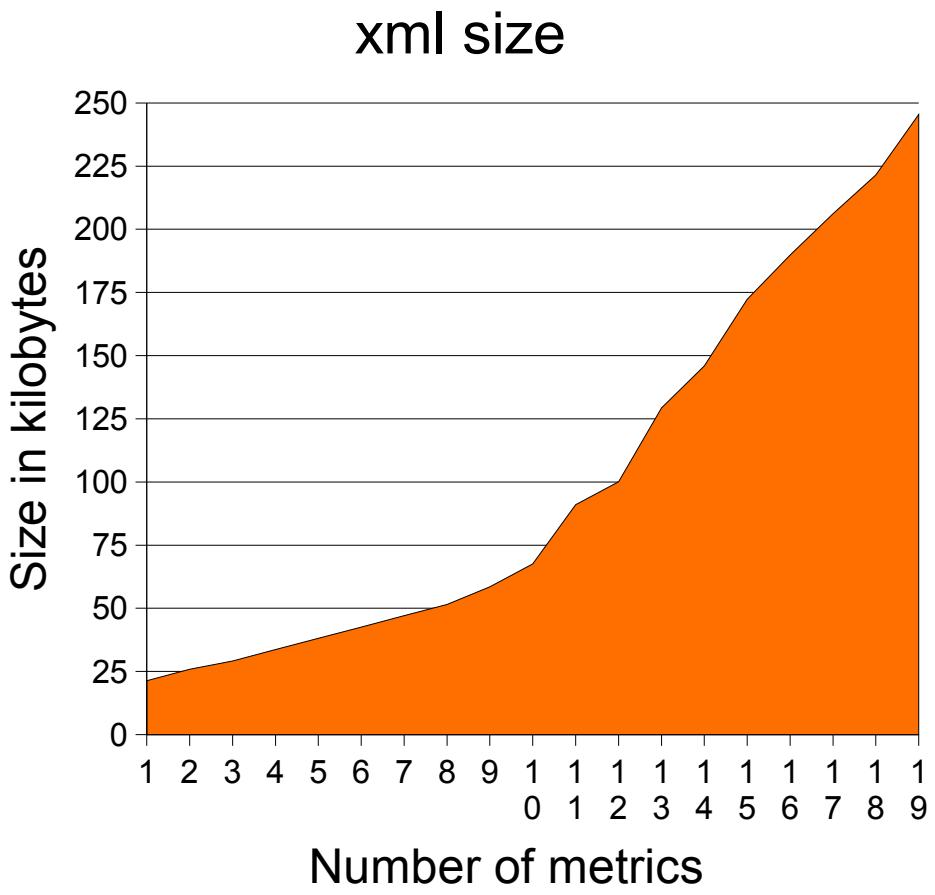
- access time is linearly-dependent on data size

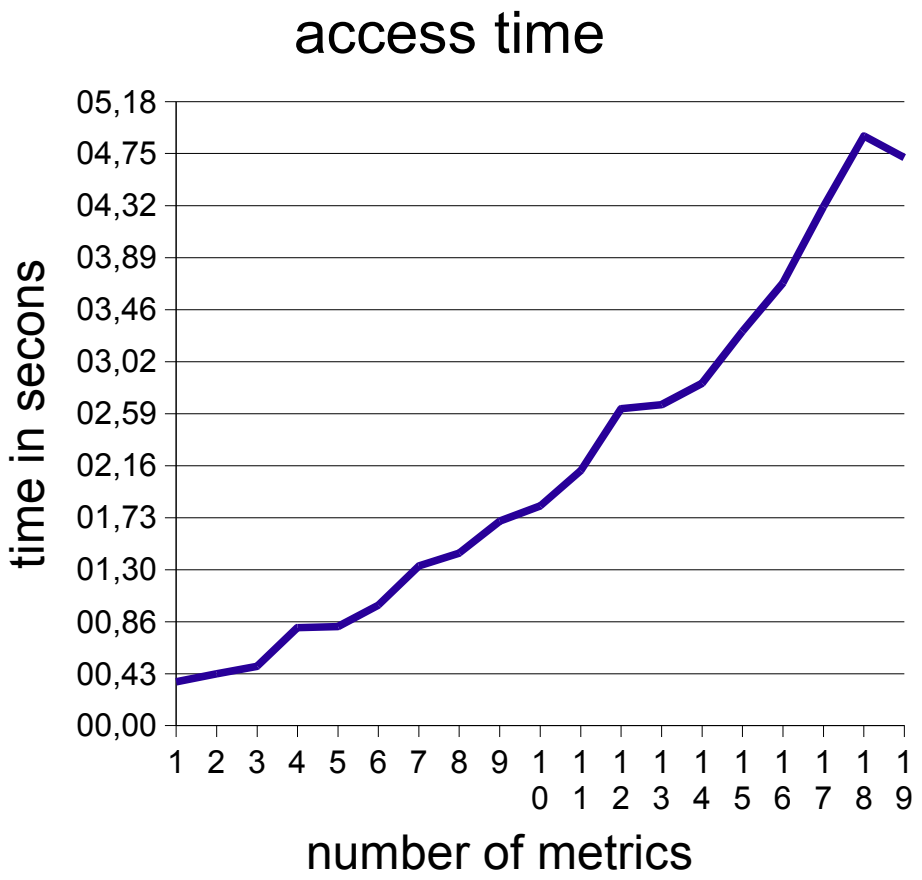
Data-size tests (II)



About 90% of XML creation time is used by Oracle server to create XML file. More web layer optimization is not too much important.

Multiple metrics tests

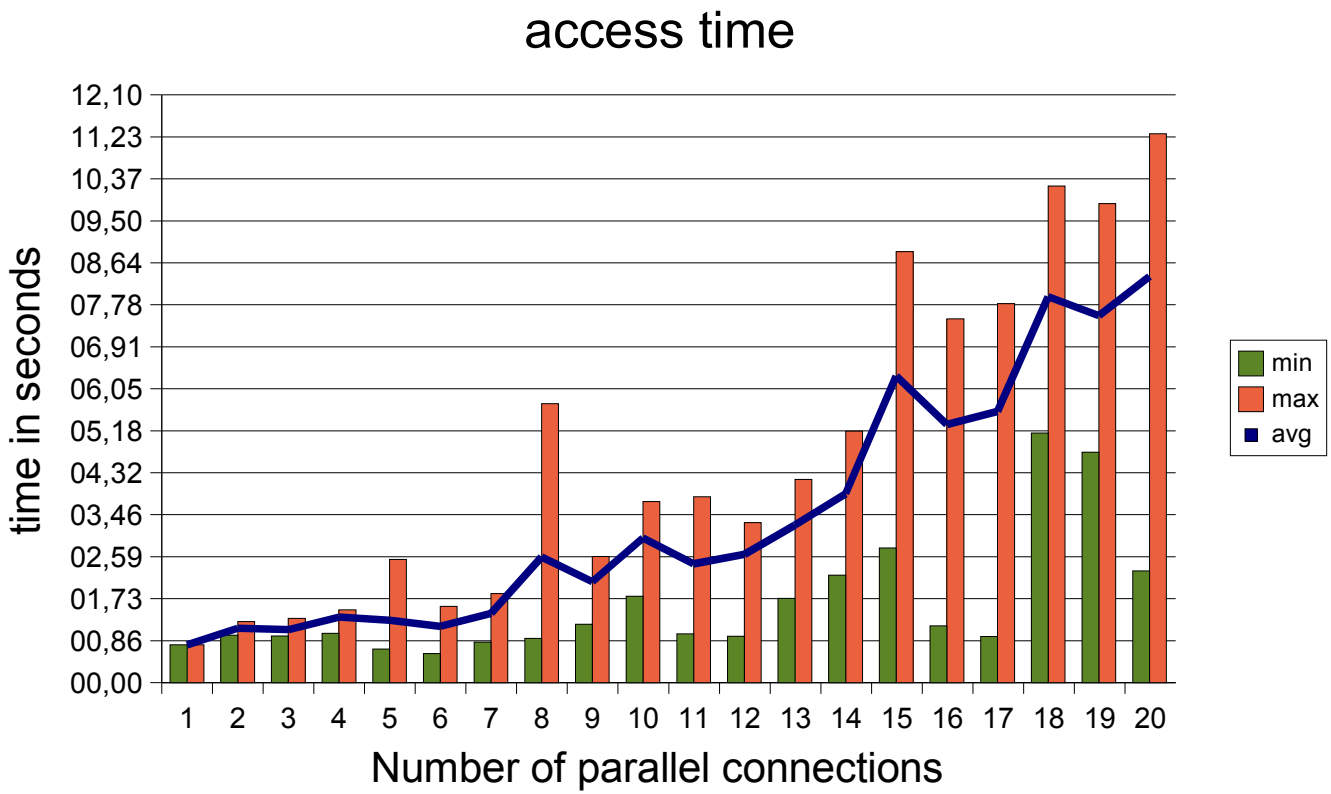




1 client, data for 2 hosts, timespan 1 day, 1 ~ 19 metrics

- access time is linearly-dependent on number of metrics

Concurrent access test



1..20 clients, data for 711 hosts, timespan 2h, 1 metric - different for each client, average XML size = 43kB

- our prototype scales well, concerning running it on stock hardware (workstation PC)

Summary

- Current implementation is not suitable for future use and needs redesigning
- We propose new, more flexible, scalable, secure and robust schema
- Using XML allows us to have multiple implementations and different consumers of Lemon data
- By providing API in major programming languages we will allow fast development of client applications

Questions

?