

LS1 and LS2 performance comparison

Hardware:

cpu: Intel(R) Core(TM)2 Duo CPU E8400 @ 3.00GHz

memory : 3,6 GB

system: Ubuntu 8.10

Software:

server: tomcat 5.5

java: 1.6.0_14

eXist: 7866-20080610

lookupService: LS1(psBase1) and LS2(psBase2)

Measurement:

Measurement of the single request is difference between time before send request and time after response is received. Units of measurement are milliseconds.

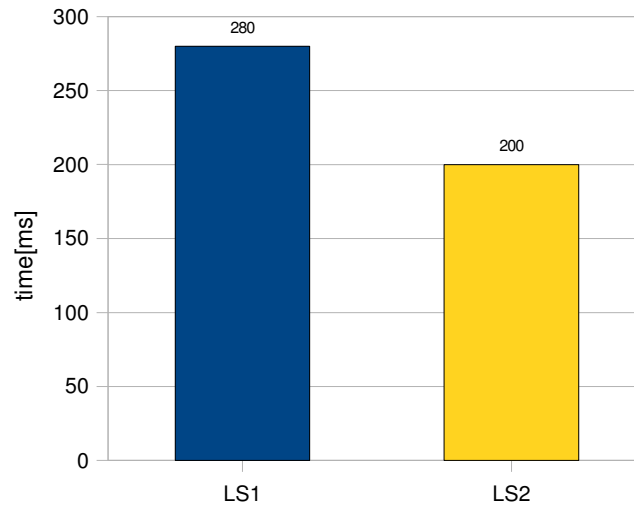
Description:

This document describes 3 kind of tests: registration, querying, deregistration. All this tests were executed using services which every contains 100, 500 or 1000 interfaces. LS2 in tests uses new database model (multiple files) and this change has a big influence on the efficient (especially de-registration).

1. Scenario – Services with 100 interfaces

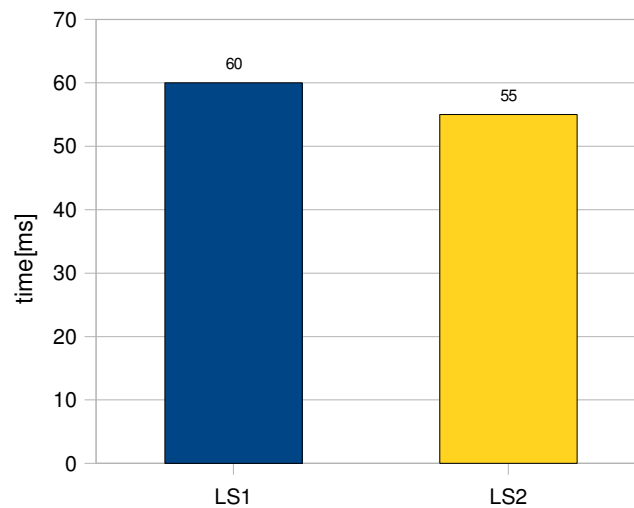
1.1) Registration (every service with 100 interfaces) - 150 registration requests

Avg processing time for registration request
(less is better)



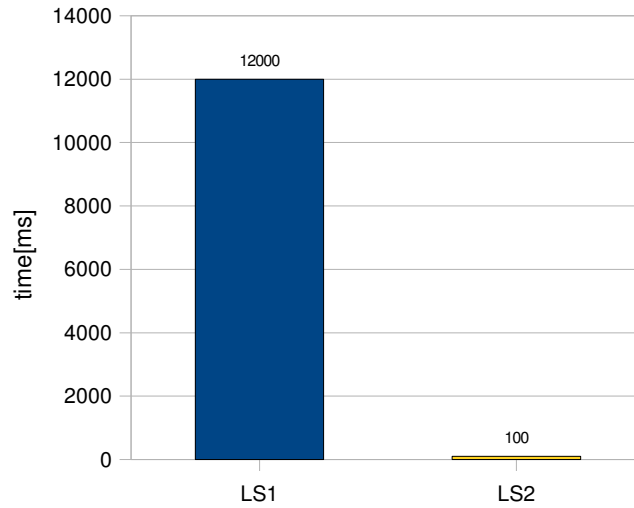
1.2) Querying (database with 150 services which every contains 100 interfaces) - 30 query requests

Avg processing time for query request
(less is better)



1.3) De-registrations (database with 150 services which every contains 100 interfaces) - 30 deregistration requests

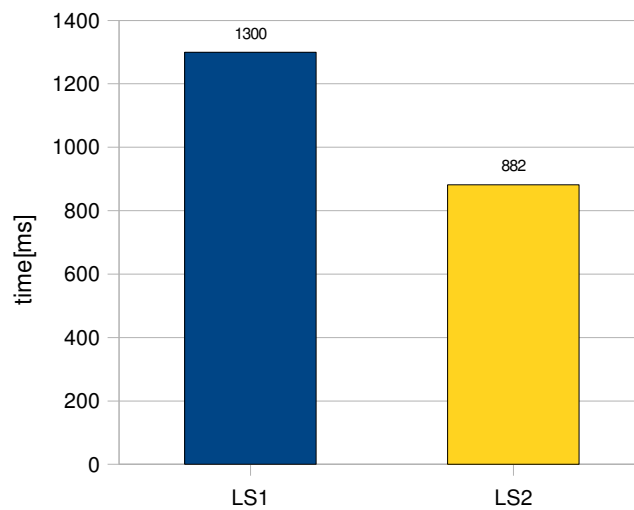
Avg processing time for de-registration request
(less is better)



2. Scenario – Services with 500 interfaces

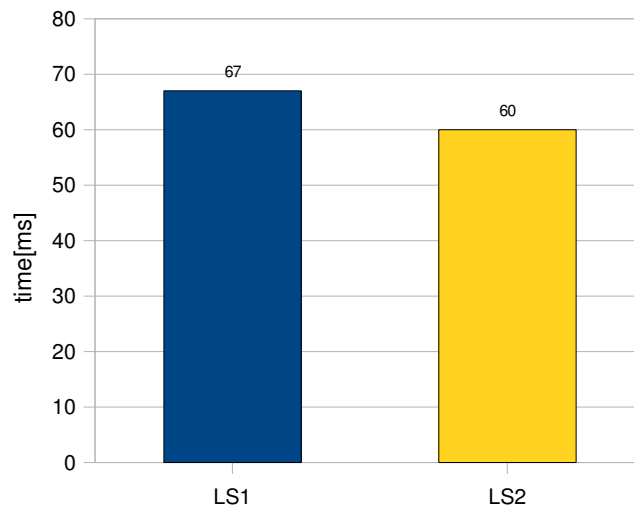
2.1) Registration (every service with 500 interfaces) - 150 registration requests

Avg processing time for registration request
(less is better)



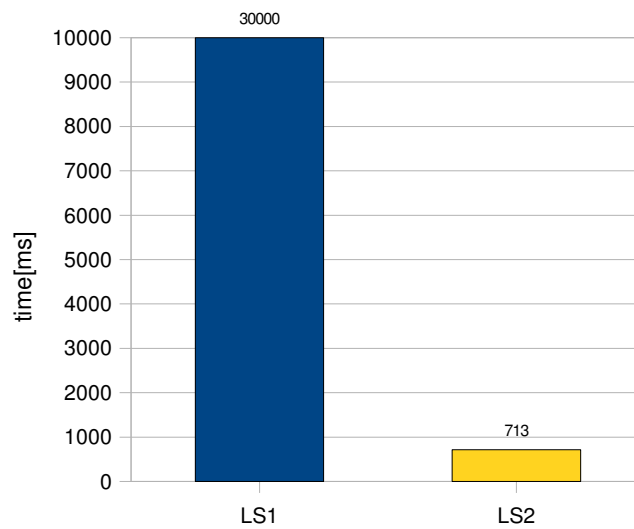
2.2) Querying (database with 150 services which every contains 500 interfaces) - 30 query requests

Avg processing time for query request
(less is better)



2.3) De-registrations (database with 150 services which every contains 500 interfaces) - 30 deregistration requests

Avg processing time for de-registration request
(less is better)

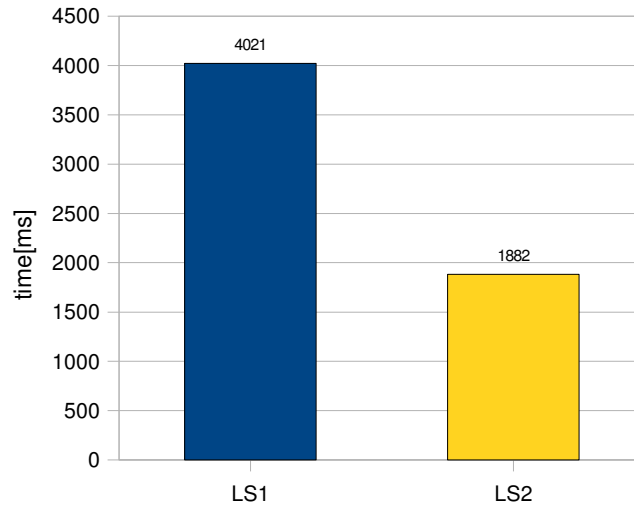


Time of deregistration in LS1 in this scenario was greater then 5 minutes.

3. Scenario – Services with 1000 interfaces

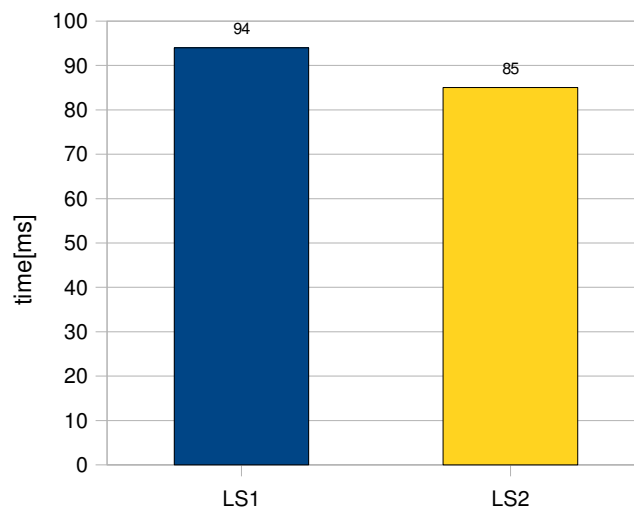
3.1) Registration (every service with 1000 interfaces) - 100 registration requests

Avg processing time for registration request
(less is better)



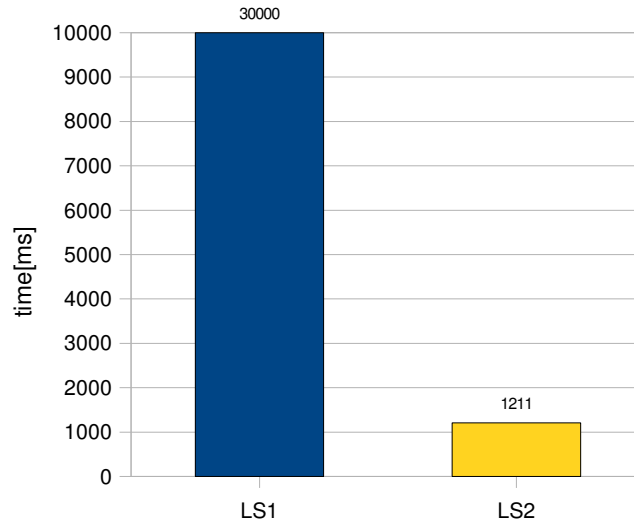
3.2) Querying (database with 100 services which every contains 1000 interfaces) - 30 query requests

Avg processing time for query request
(less is better)



3.3) De-registrations (database with 100 services which every contains 1000 interfaces) - 30 de-registration requests

Avg processing time for de-registration request
(less is better)



Time of deregistration in LS1 in this scenario was greater then 5 minutes.

Summary

LS1 has big problem with processing operations on big part of data - deregistration of the service which contains more than 300 interfaces lasts few minutes. It is insufficient value for lookup service. We are still working on performance improvement in LS2. The bottleneck of those operations (registration, de-registration, querying) are operations on the ExistDB. In LS2 we are using multiple files model, where every registration causes creation of new file(1 file per registration). It is a big difference with model used in LS1 where data are stored in one file. This change improves performance especially deregistration from eXist's database.