**Slawomir Trzaszczka (trzaszcz@man.poznan.pl)**
**Maciej Glowiak (mac@man.poznan.pl)**
**PSNC**
**21.12.2009**

# LS1 and LS2 multiThreading 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 miliseconds.

**Description:**

This document describes more real use case tests. In this test I used 6 threads. Every thread worked separately. Every thread was responsible for one type of operation (registration, de-registration, query, re-registration). This is simple algorithm of thread that was used in this test.

*Thread :*

```
public run()
 begin
   this.sleep(START_DELAY)
   while(true)
    begin
      //do operation
      this.sleep(DELAY)
    end
 end
```

Every thread has 2 parameters:
- START_DELAY – thread waits this time on the thread start
- DELAY – thread waits this time after every iteration

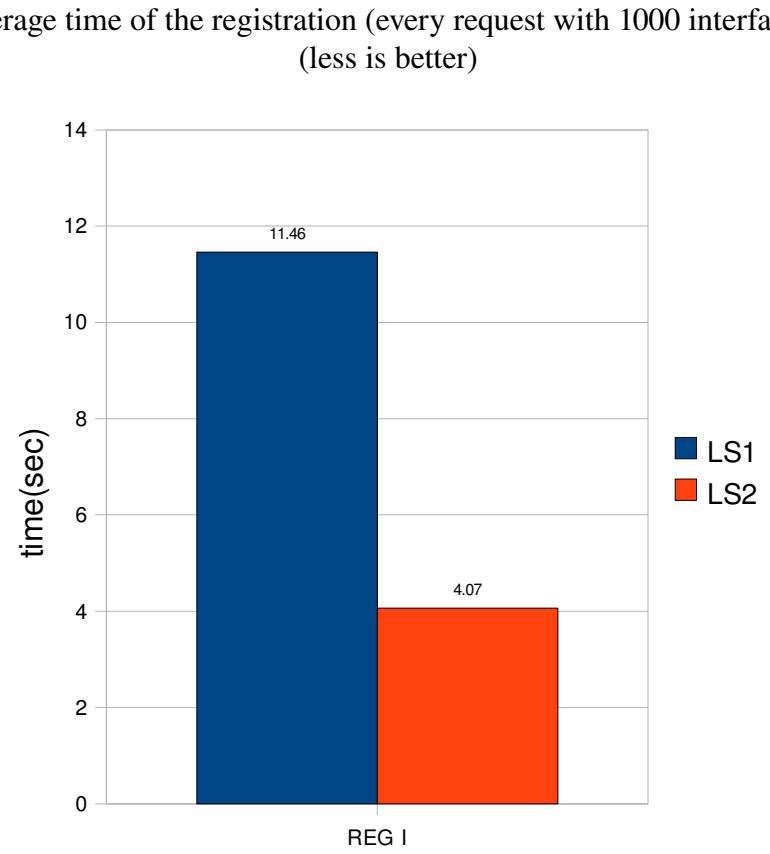
Threads in tests:

| Nr | Thread operation | START_DELAY(sec) | DELAY(sec) |
|---|---|---|---|
| 1. | Registration I | 0 | 1 |
| 2. | Registration II | 30 | 2 |
| 3. | Query I | 5 | 0.05 |
| 4. | Query II | 10 | 0.05 |
| 5. | De-registration | 40 | 0.05 |
| 6. | Re-registration | 15 | 1.5 |

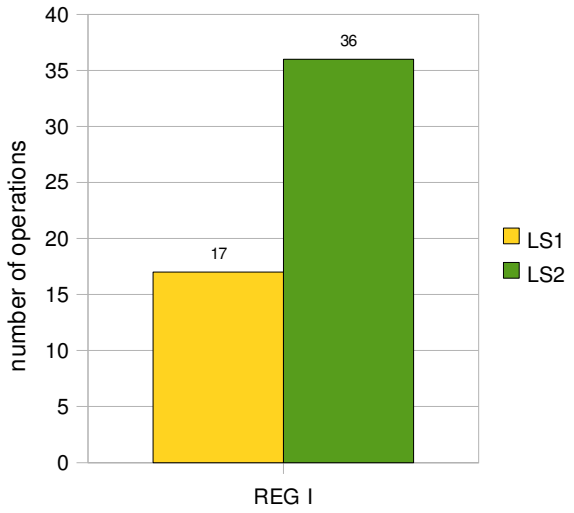
This test bases on requests that every registration request contains 1000 interfaces. These 6 threads works parallel 120 seconds.

**Results:**

1. Thread nr 1 – Registration I

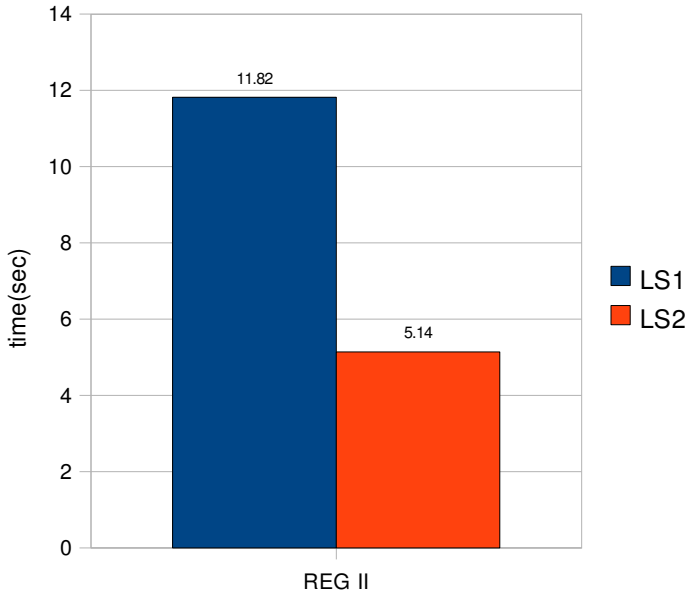


Number of executed operation during test (120 sec)
(more is better)

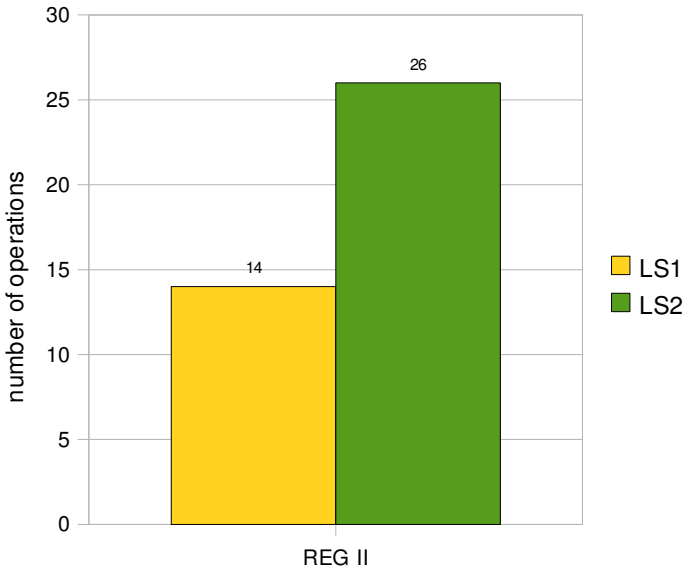



2. Thread nr 2 – Registration II

Average time of the registration (every request with 1000 interfaces)
(less is better)

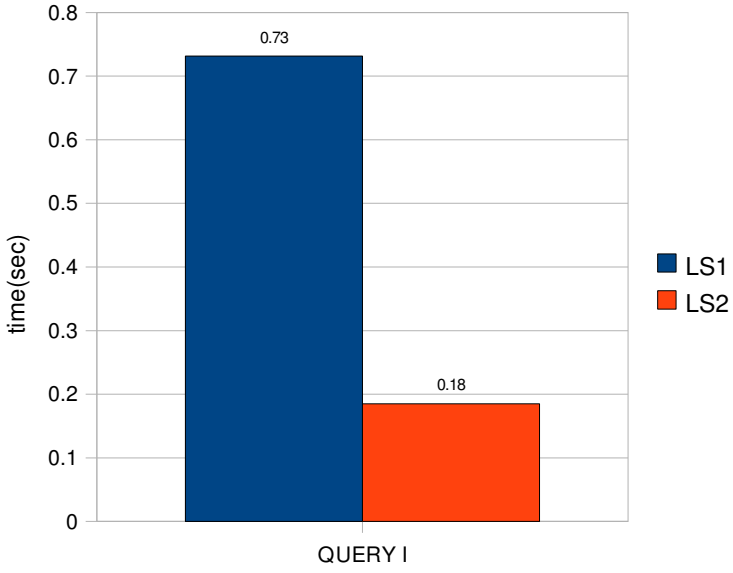


Number of executed operation during test (120 sec)
(more is better)

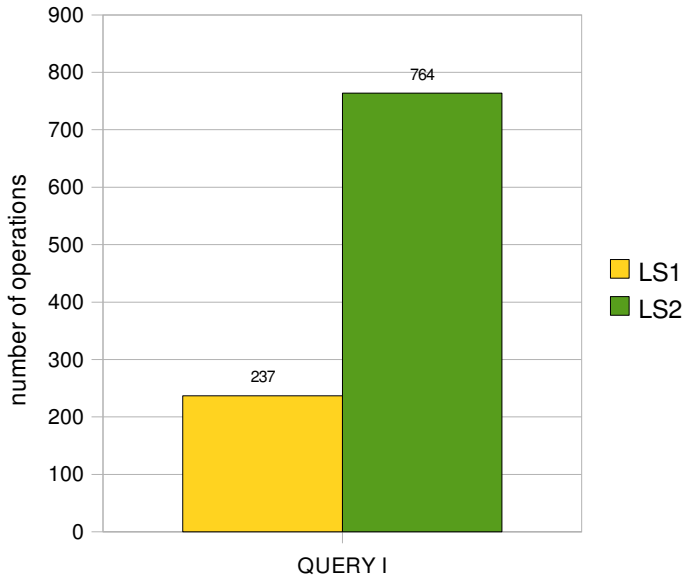



3. Thread nr 3 – Query I

Average time of the querying
(less is better)

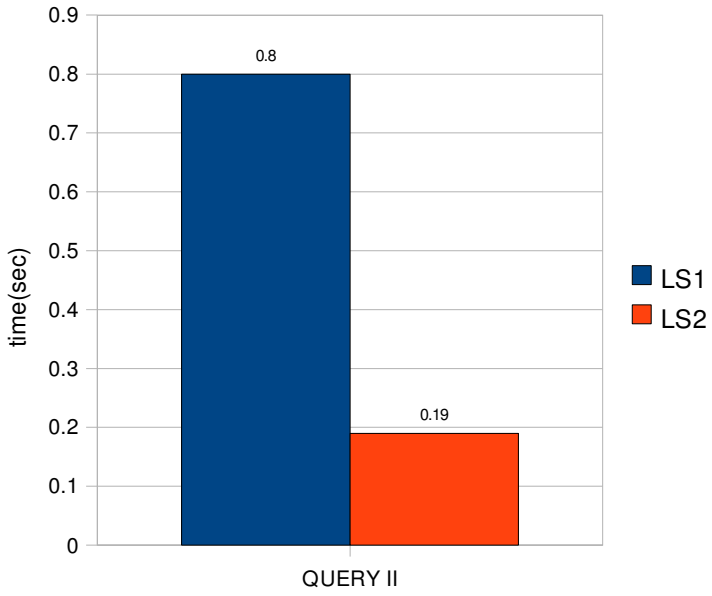


Number of executed operation during test (120 sec)
(more is better)

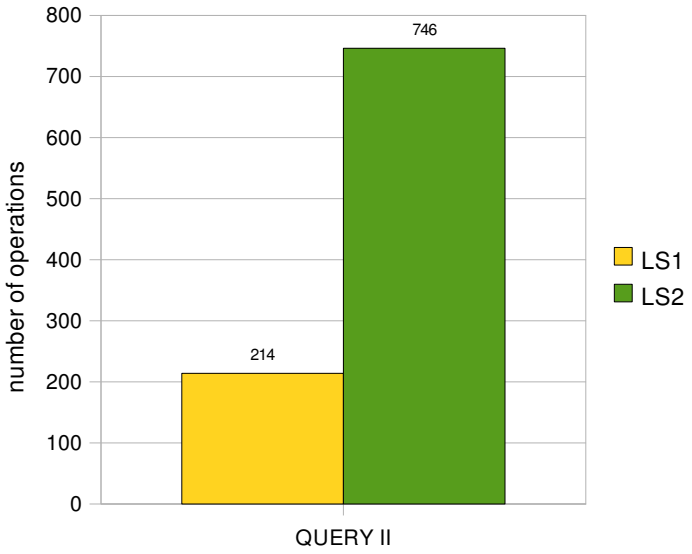



4. Thread nr 4 – Query II

Average time of the querying
(less is better)

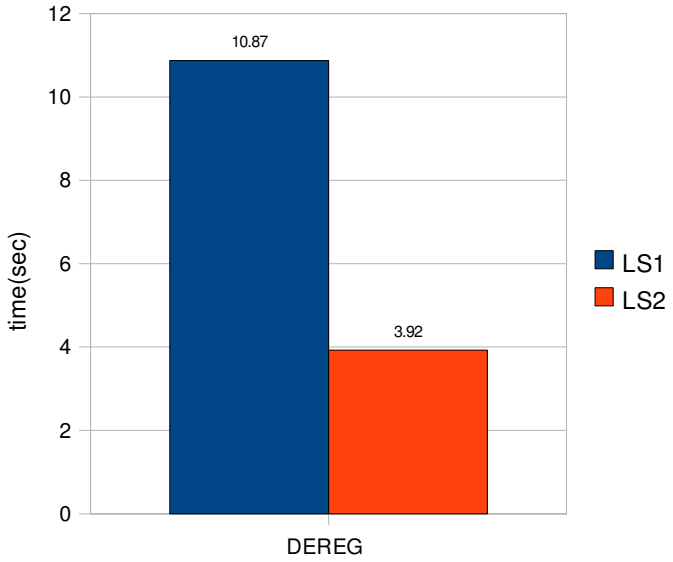


Number of executed operation during test (120 sec)
(more is better)

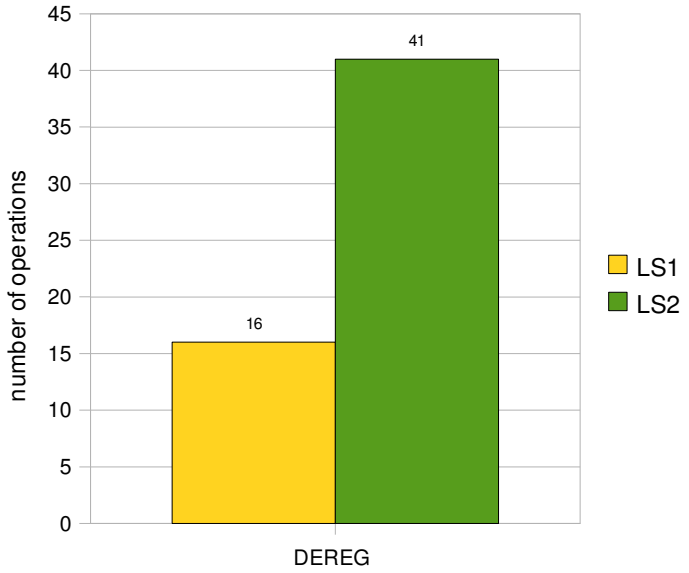



5. Thread nr 5 – De-registration

Average time of the de-registration
(less is better)

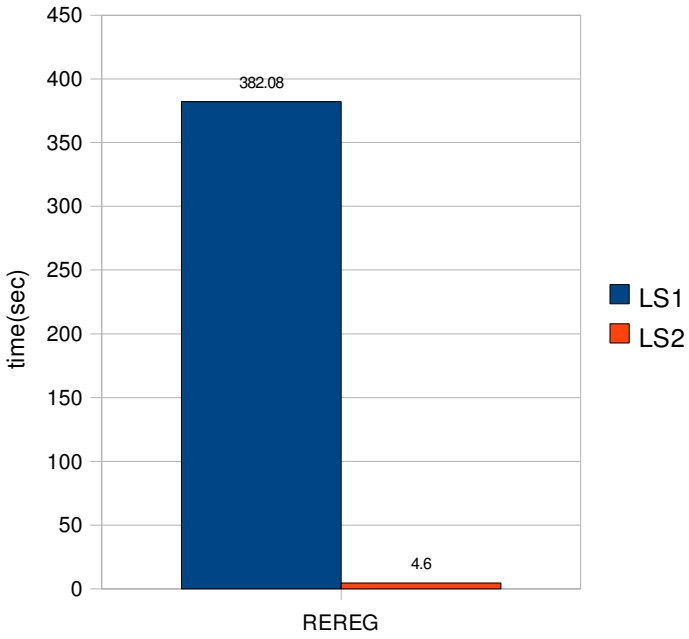


Number of executed operation during test (120 sec)
(more is better)

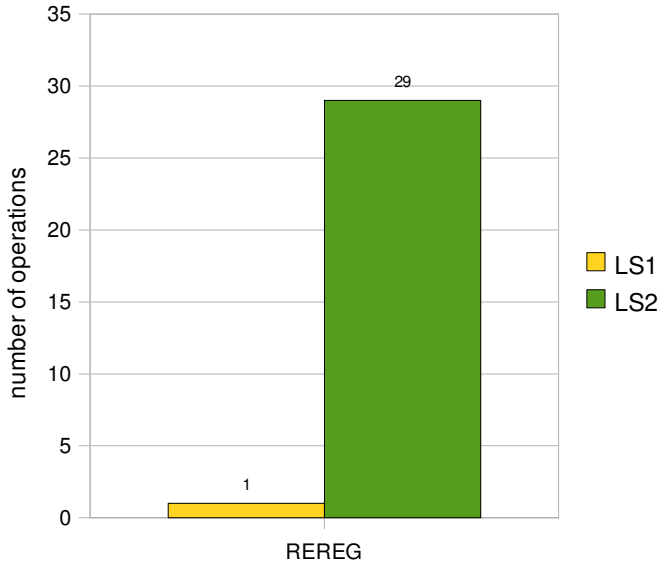



6. Thread nr 6 – Re-registration

Average time of the re-registration (every with 1000 interfaces)
(less is better)

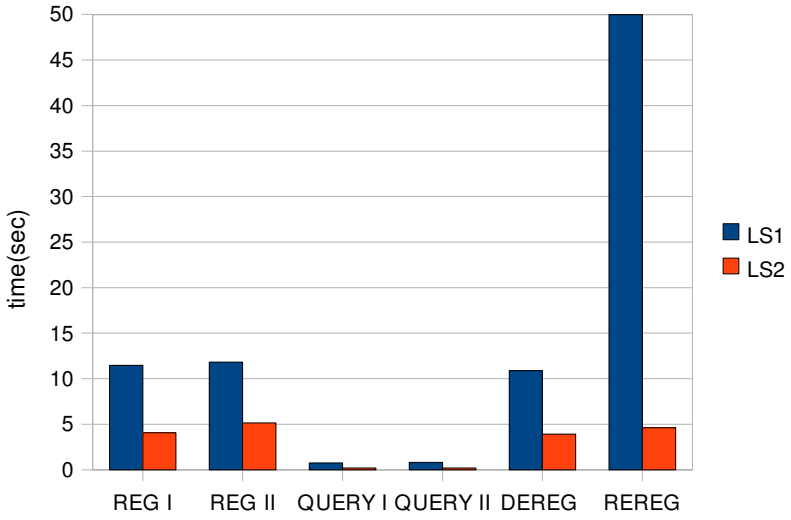


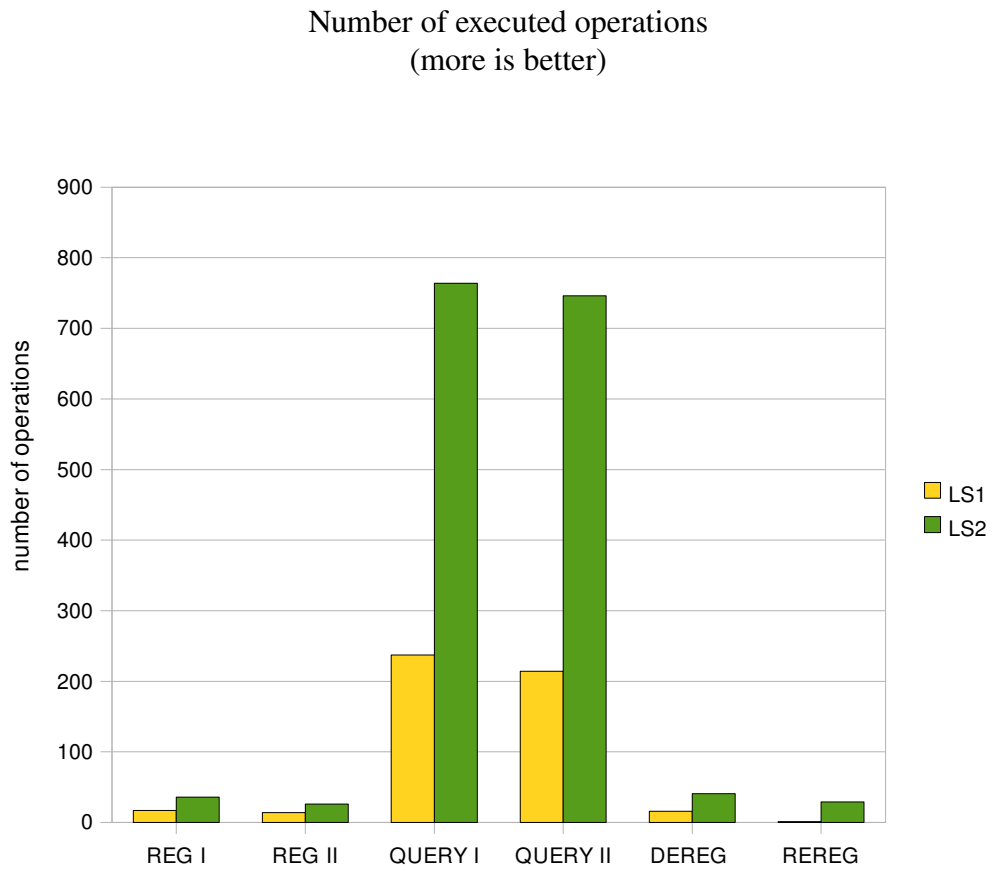
Number of executed operation during test (120 sec)
(more is better)




7. All in one

Average processing time
(less is better)






**Summary:**

There is big performance improvement between lookup service based on psBase 1 and lookup service based on psBase 2. Average processing time of the operations are in all cases less than in LS1. This cause opportunity to execute more operations in the same time.