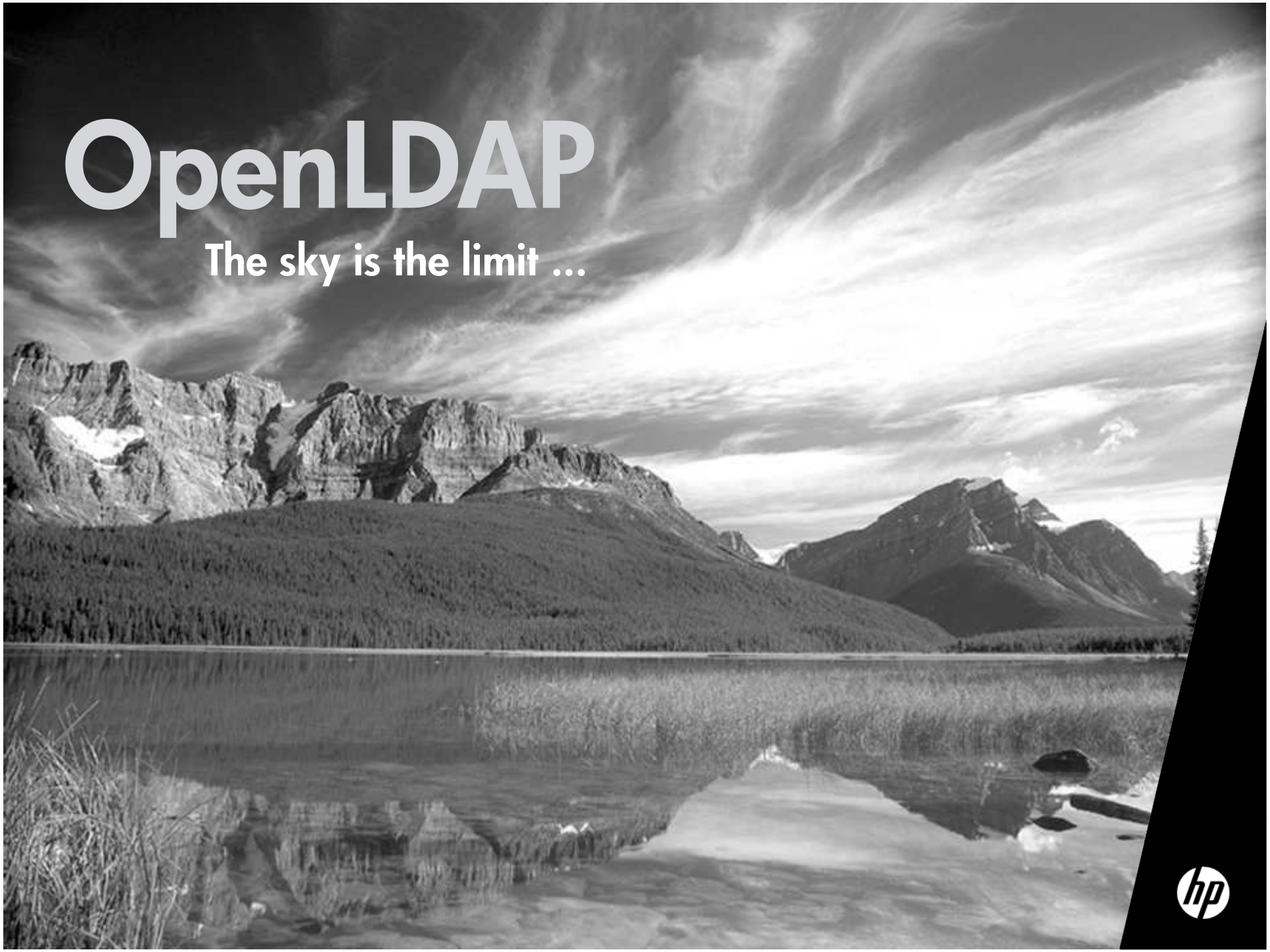
Large Scale High Performance OpenLDAP

A real production world experience

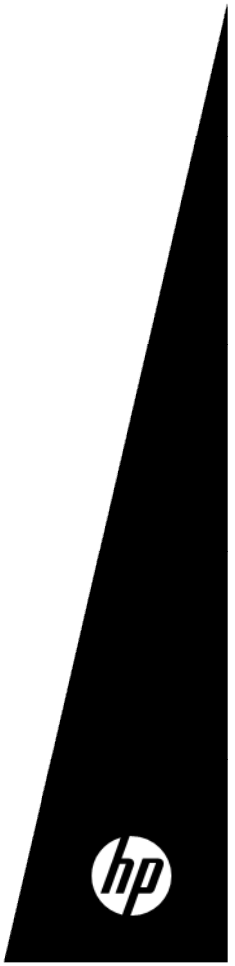
Wolfgang Hummel
Solution Architect
October 10th 2011


OpenLDAP
The sky is the limit ...

Agenda

What to talk about …

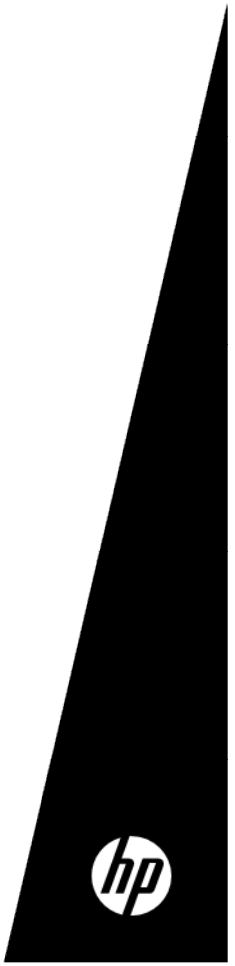
- What is Large Scale ?
- What is High Performance ?
- A typical deployment scenario
- Benchmark Setup
- Benchmark Results
- Tuning Screws



What is "Large Scale"

How much is many ?

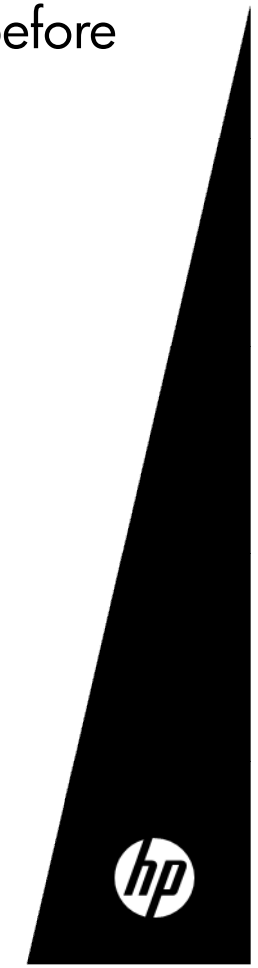
- Most typical LDAP Use Case:
  - All persons in an organization
  - One person = one entry, each with n attributes
- HP is a large organization that uses OpenLDAP.
  Corporate LDAP Directory has ~ 1 Mio. entries
- Our customers have 20 - 40 Million entries …
  So the HP LDAP experience is only of limited use here
- Only chance: Test, Test, Test …



What is "High Performance"

Is one second a long time ?

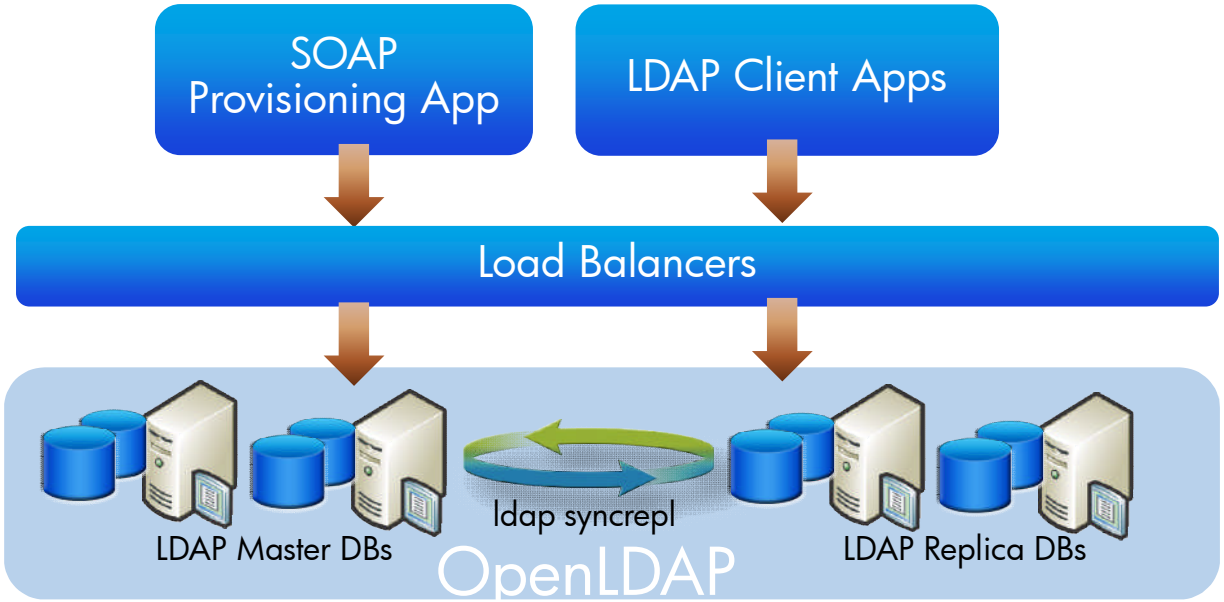
- LDAP is used for telephony services
  requires "near realtime" response times
  = < 1 second response time for all transactions that are executed before accepting a call
  = 10-15 transactions, one of them is LDAP subscriber profile lookup
- "fair share" of LDAP request is
  - < 20 ms for read
  - < 100 ms for write
- up to 10.000 read / 1.000 write requests / second
  → **yes, in parallel !**



A typical deployment scenario

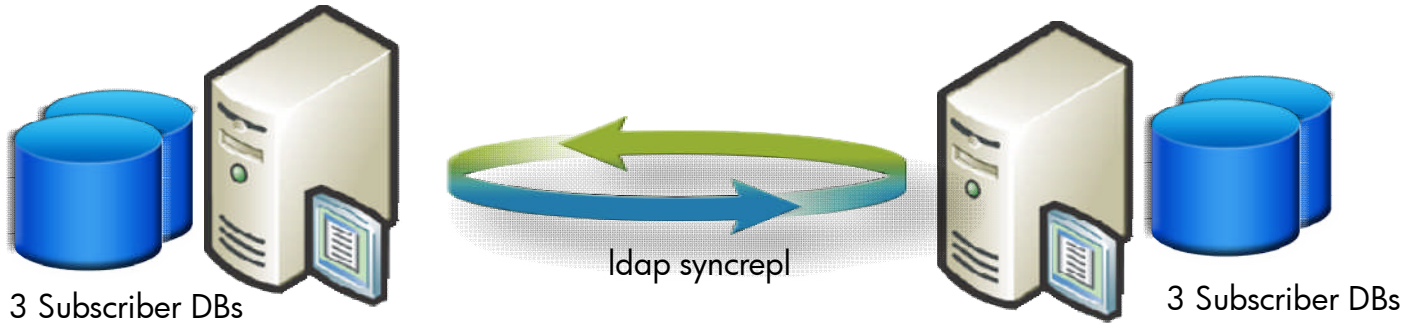
... in the real world

- 2 Masters
- 2 - n Replicas
- 2 large Subscriber DBs, 2 small DBs
- distributed over 2 physical sites



Benchmark Setup

Hardware and OS

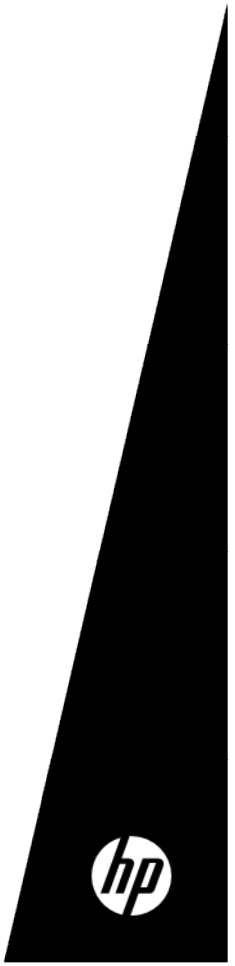


- 2 x HP DL380 G7 each with
  - 192 GB Memory
  - 4 hexcore-CPUs Intel Xeon X5680 @ 3.33GHz 12 MB level 1 cache
  - RAID Controller with 1 GB BBWC
  - 16 local disks 143 GB 15k rpm 2,5 " with
    2 x OS RAID10
    10 x LDAP RAID0
    4 x logging RAID0
- OS RedHat Enterprise Linux 5.5 64bit
- One Server is LDAP Master + load driver
- One Server is LDAP Read Cache + load driver

Benchmark Setup

Load Scenarios

- 3 DBs:
  → DB1 22 Mio. entries
  → DB2 35 Mio. entries
  → DB3 40 Mio. entries
- Working Set:
  → DB1 9 Mio.
  → DB2 13.5 Mio.
  → DB3 16.2 Mio.
- 11 attributes, 325 bytes LDIF data per entry
- Target Load:
  - 2.500 reads/sec DB1, 3.500 reads/sec DB2, 4.000 reads/sec DB3
  - 250 writes/sec DB1, 350 writes/sec DB2, 400 writes/sec DB3
    peak load with 700 writes/sec in DB3
  - For writes 70/30 ratio between modify and add



Benchmark Setup

Software & Tools

- OpenLDAP 2.4.23 with 2 syncrepl patches (contained in 2.4.24)
- custom multi-threaded perl load script
- custom monitoring script for memory consumption
- custom monitoring script to check if DBs are in sync
- nmon → all system resources
- top → memory, CPU
- OpenLDAP log files (loglevel 16640)
- load script log files (logs each request and measures response times)

Benchmark Results

Summary - 1

- All load scenarios have been achieved or over-achieved:
  → 17.000 reads / sec on a single DB (on a single server !)
  → 4.500 reads / sec combined with 700 writes / sec on a single DB
  → 10.000 reads / sec combined with 1.000 writes / sec on 3 DBs
- For read transactions the load driver was the limit
  For write transactions the disk i/o was the limit
- Latency for read and write requests is extremely low
  spread for write requests is bigger than for read requests
  → 1msec avg response time for read
  → 2msec avg response time for write
  (measured from client perspective)

Benchmark Results

Summary - 2

- Different Scenarios:
  → "All In One" LDAP Master for read & write on same server
  → LDAP Cache Read Only with 1 DB on 1 server or 3 DBs on 1 server
  → LDAP Master for Write with sync to LDAP Cache for Read
- "Side" Test Case Migration:
  bulk add of 40 Mio. Subscribers in an empty DB takes
  97 minutes = 6.872 adds / second
- "Side" Test Case Bulk Modifications:
  modify on all 3 DBs on LDAP Master with sync to LDAP Cache
  No read transactions during this time
  → 1.500 requests / sec

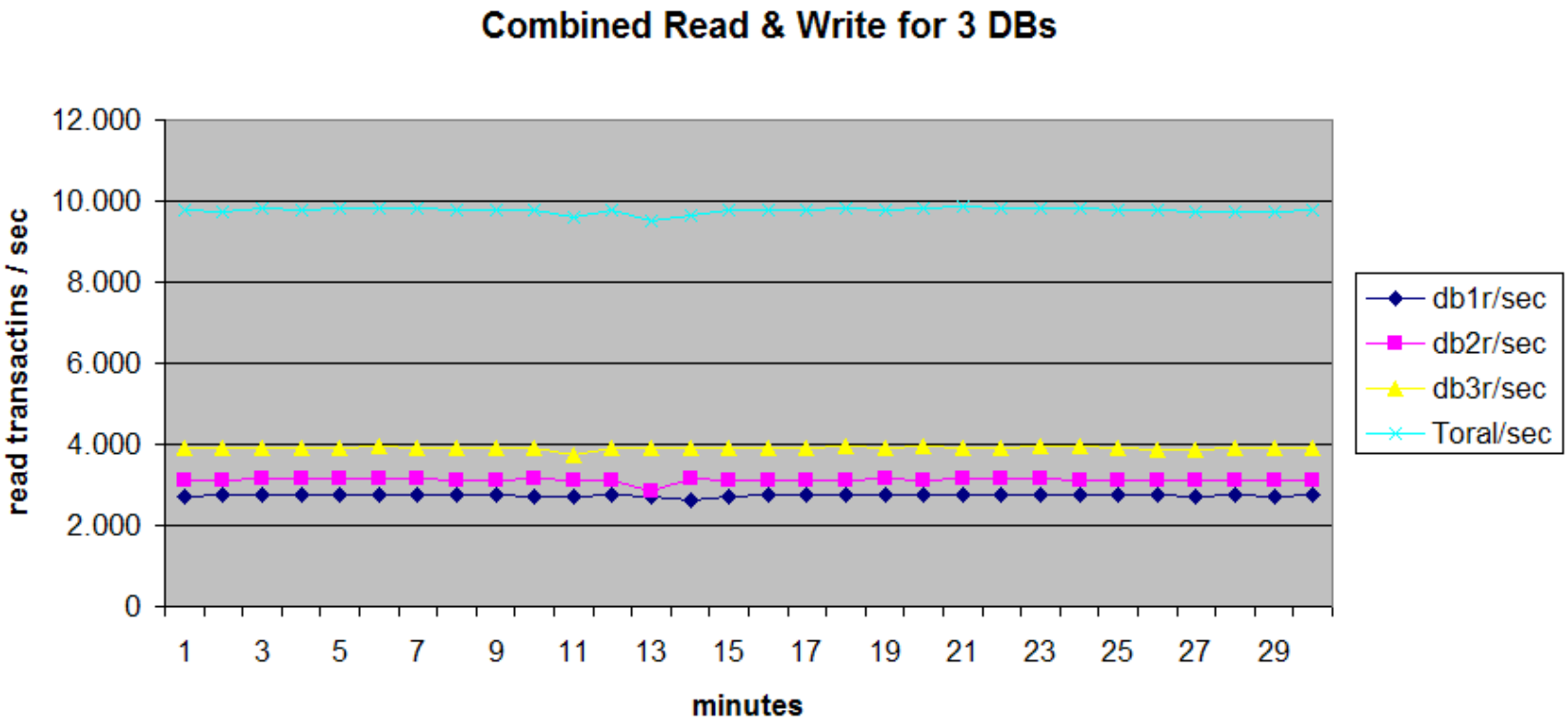
Benchmark Results

Read Cache Server with 3 DBs and **parallel Sync** from Master - 1

- *LDAP Master is load driver for read transactions on LDAP Cache*
- *LDAP Cache is load driver for write transactions on LDAP Master*
- *Read Transactions on LDAP Cache*
- *Modify & Add Transactions on LDAP Master with Sync to Cache*
- *All 3 DBs are used*
- *Result:*
  10.000 reads / sec
  1.000 writes / sec with ~ 400 adds and ~ 600 modifies

Benchmark Results

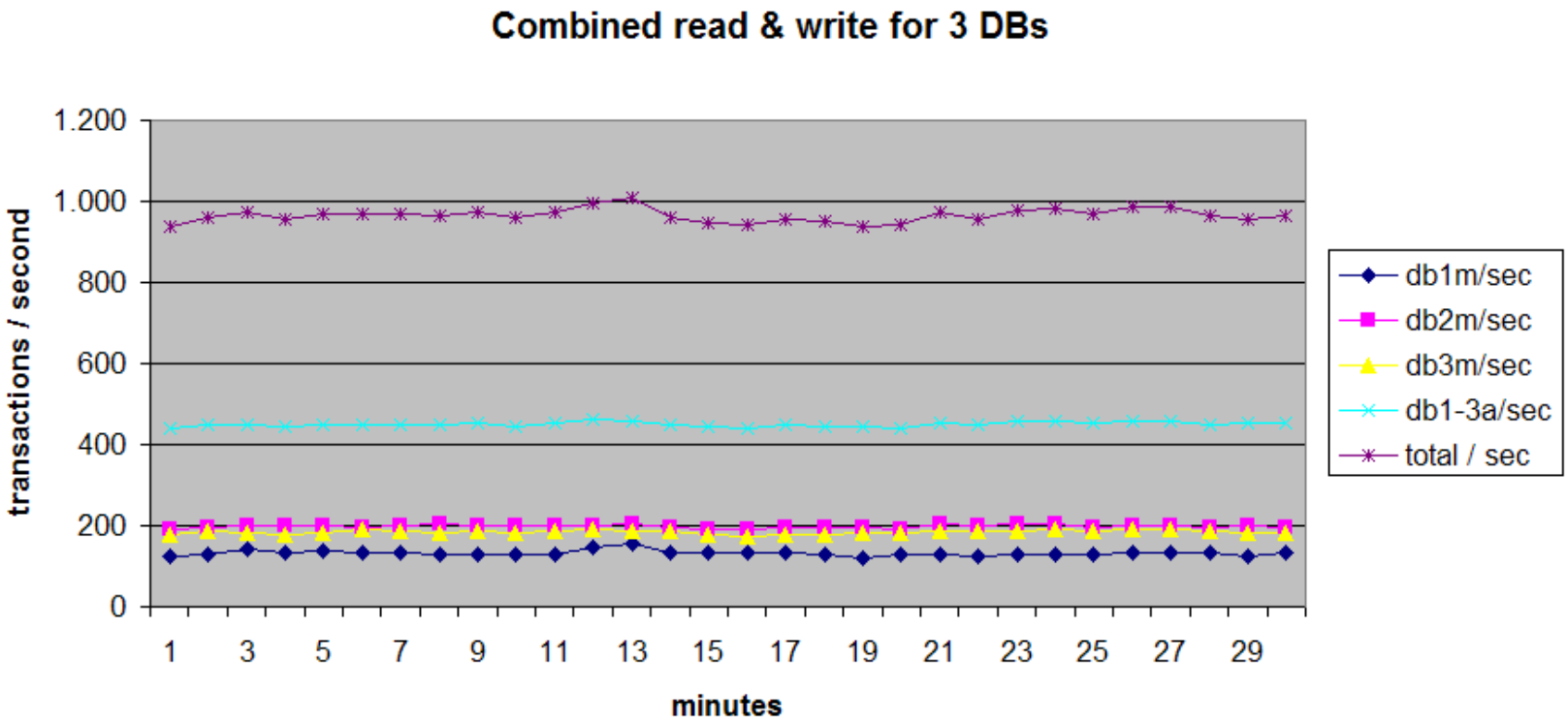
Read Cache Server with 3 DBs and **parallel Sync** from Master - 2



- *Test ran for 7 hours, here a segment of 30 minutes is shown*
- *Transaction rate was quite stable*

Benchmark Results

Read Cache Server with 3 DBs and **parallel Sync** from Master - 3



– load is very constant

– add transactions are distributed evenly over all 3 DBs

Benchmark Results

Read Cache Server with 3 DBs and **parallel Sync** from Master - 4

```
top - 19:30:09 up 4 days, 11:34,  6 users,  load average: 3.36, 3.48, 3.44
Tasks: 209 total,   2 running, 206 sleeping,   0 stopped,   1 zombie
Cpu(s): 18.7%us,  4.2%sy,  0.0%ni, 75.8%id,  0.0%wa,  0.2%hi,  1.1%si,  0.0%st
Mem:  198089124k total, 197594132k used,   494992k free,   895800k buffers
Swap: 16386292k total,   274880k used, 16111412k free, 147687432k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
23704 ldap      25   0  116g  90g  47g S 144.1 47.8 854:57.41 slapd
 6002 root      24   0 2466m  57m 1032 S  0.0  0.0   0:11.34 java
32466 root      15   0  288m 1708  864 S 19.0  0.0   1079:07 rsyslogd
 5789 named     25   0  260m 1468  744 S  0.0  0.0   0:00.01 named
 7278 root      18   0  252m 2468 2108 S  0.0  0.0   3:57.20 cmanicd
12283 root      15   0  181m  15m 2012 S  3.0  0.0  15:19.07 ldapBenchmark1w
```

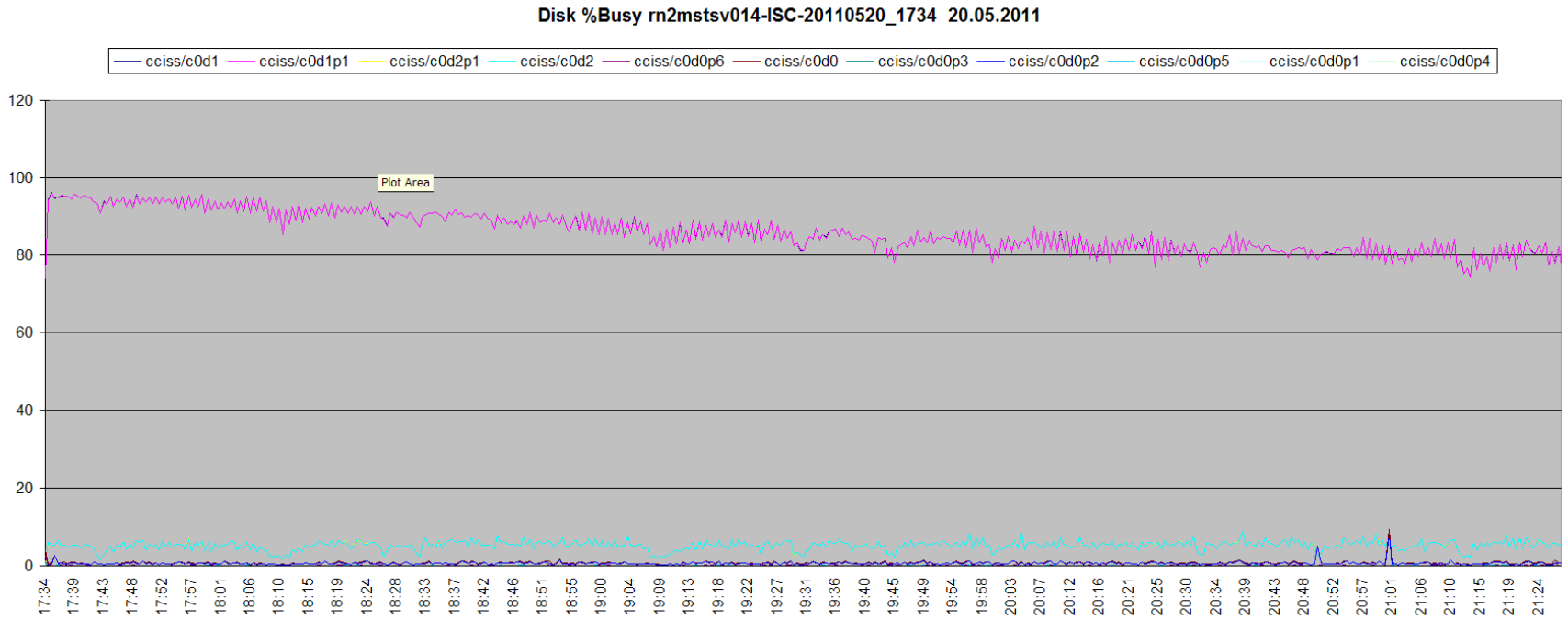
```
top - 21:30:54 up 18 days,  6:58,  5 users,  load average: 16.96, 16.95, 16.79
Tasks: 198 total,   2 running, 195 sleeping,   0 stopped,   1 zombie
Cpu(s): 65.2%us,  5.6%sy,  0.0%ni, 23.0%id,  3.9%wa,  0.2%hi,  2.1%si,  0.0%st
Mem:  198089124k total, 197276020k used,   813104k free,   360576k buffers
Swap: 16386292k total,      292k used, 16386000k free, 134856168k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 7319 ldap      25   0  127g 113g  59g S 85.2 60.1 359:31.50 slapd
25588 root      19   0  604m 117m 2016 S 271.3  0.1   1286:22 ldapBenchmark2.
25579 root      20   0  519m  95m 2016 S 211.1  0.0   1021:32 ldapBenchmark1.
25567 root      22   0  511m  88m 2016 S 180.8  0.0 887:51.18 ldapBenchmark.p
29270 named     25   0  260m 4772 1976 S  0.0  0.0   0:00.09 named
19318 root      15   0  224m 2072 1040 S 24.6  0.0 289:23.17 rsyslogd
```

– top 1 - LDAP Cache & Sync Target & Load Driver

– top 2 - LDAP Master & Load Driver

Benchmark Results

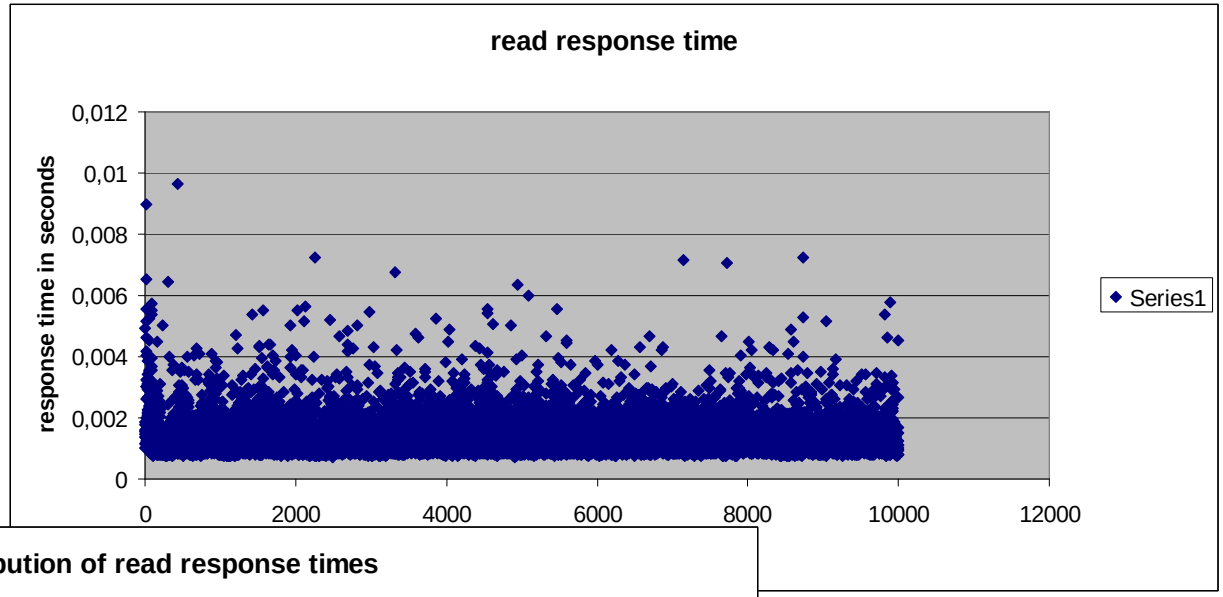
Read Cache Server with 3 DBs and **parallel Sync** from Master - 5

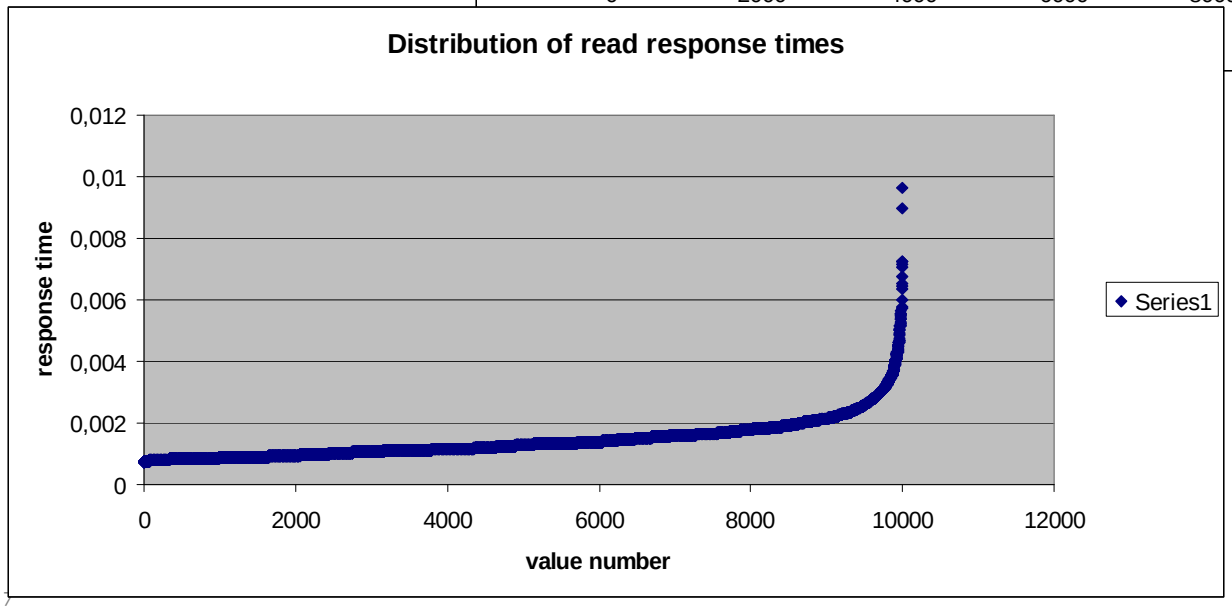


– Disk I/O on LDAP Master is very high for the LDAP DB filesystem

Benchmark Results

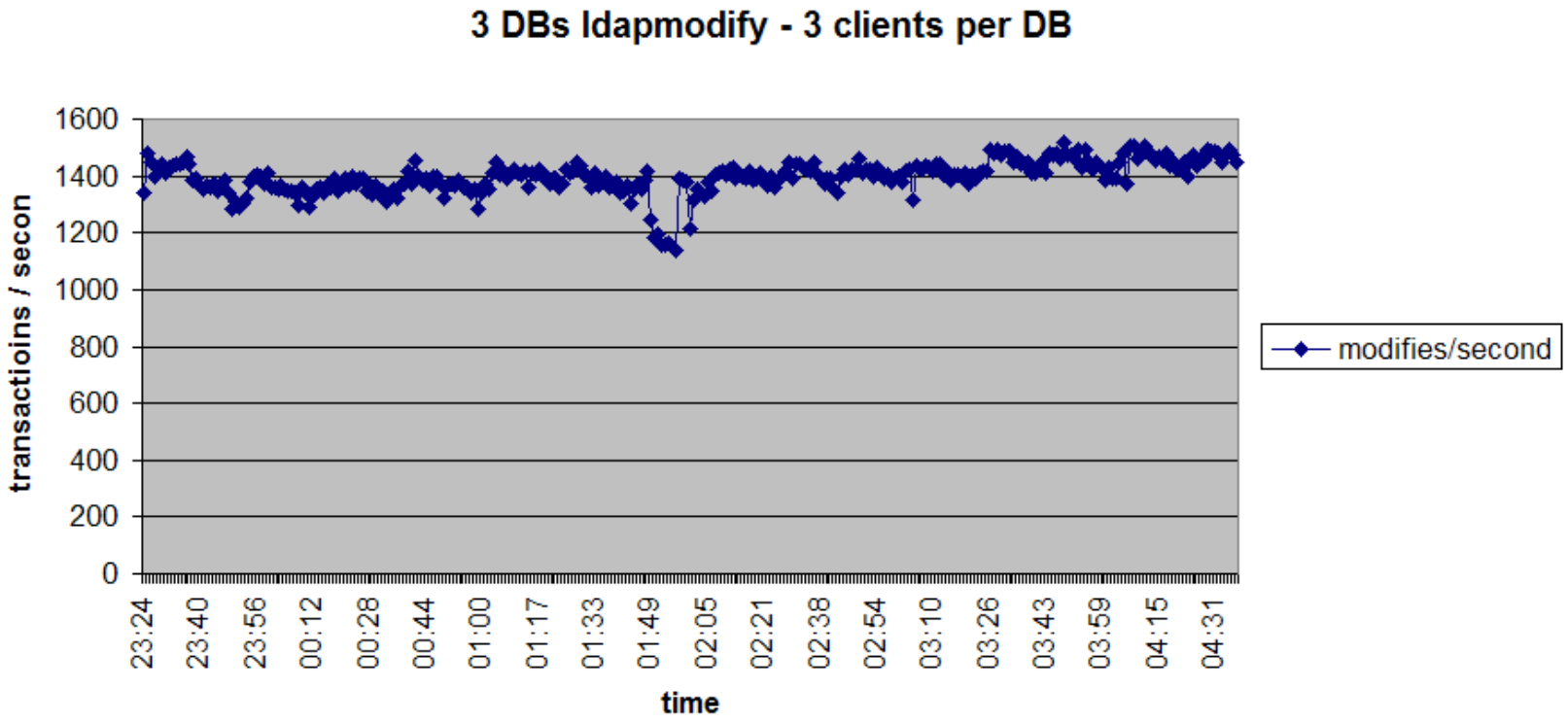
Read Cache Server with 3 DBs and **parallel Sync** from Master - 6





Benchmark Results

Bulk DB Modifications - no parallel read transactions



- ldapmodify on 3 DBs in parallel on LDAP Master with sync to LDAP Cache
- DBs stay completely in sync

Performance Screws - 1

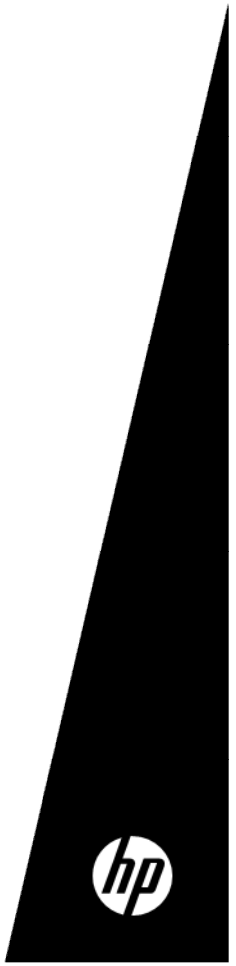
How to boost performance

- More Servers
  Specific servers for read and write load
  Zoning = frontend + backend
  Specific servers for specific DBs
  → single large DB is the most critical: Load increases steadily with number of replication sync targets, too many instances are not good
- Faster & More CPUs
  Faster CPU is better than many Cores
  → not all system components scale linear with number of CPUs

Performance Screws - 2

How to boost performance

- More Memory
  The more the better.
  OpenLDAP has 4 caches
- More & Faster Disks
  Reads should come from cache
  Writes always go to the disks
  → fast disks with battery backed write cache, RAID 0 or 10 only



Performance Walls

Natural Limits

- number of TCP connections, 64 K is the OS limit per IP Address
  (including all session in time_wait, not only active sessions)
- Loglevel and Logger
  high loglevel decreases performance heavily
  logger with single threaded components is a potential bottleneck
  rsyslog scales quite good here
- Database congestion
  too many parallel clients with write transactions reduce throughput
- "Warming up" the disk backed bdb cache after a cachesize change
  imposes a high disk i/o penalty, so better warm it up before high traffic
  hits the server

Questions ?