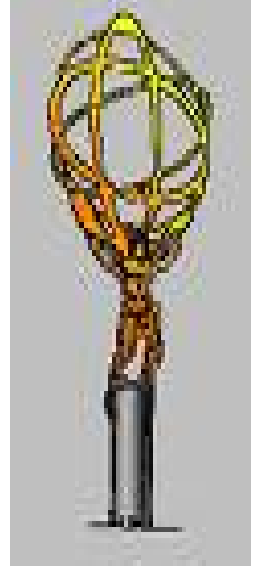


PSS

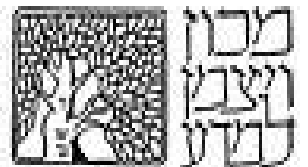
athena RECONSTRESSing: stress testing COOL reading of athena reconstruction clients



Database mini workshop, CERN 26 Jan 07

David Front

Weizmann Institute



contents

- goals
- testing means
- results
- conclusions
- resources/further testing
- related work

goals

- Stress test 100 athena reconstruction clients, each reading the same COOL data
- Reading client is either **local** to data-server site or **remote**
- Compare reading means: Oracle/squid/sqlite
- Compare frontier compression levels
- Supply input/ recommend:
will Atlas benefit from using squid?

testing means - servers/clients

- Oracle server: DB cooldev
 - Host: lxfsrk402, Intel Pentium III CPU 1133MHz, 2 CPUs, 1GB
- Frontier/squid server:
 - Host: lxb5556 Intel Xeon CPU 2.80GHz, 2 CPUs, 2GB
- Client machines: lxb machines:
 - Pentium III (Coppermine), 999 Mhz, 2 CPUs, 512 MB
 - 1 'manager' to spawn/monitor work + 4 'worker nodes'.

testing means – SW

The client

- Richard Hawkins's client, that queries data needed for one reconstruction job:
~50 COOL folders, amounting to ~10MB of payload data

'Verification client'

- Framework/scripts to run stress tests on multiple machines, monitor and present results:
 - 'Test variables' – things that changed between tests:
 - Max number of clients running at the same time: 1 – 50
 - Connection type: Oracle, squid, sqlite
 - Scheduling enhanced not to spawn a new client if load > 10
 - Added lemon graphs of involved (server/client) machines.

results

In the following graphs:

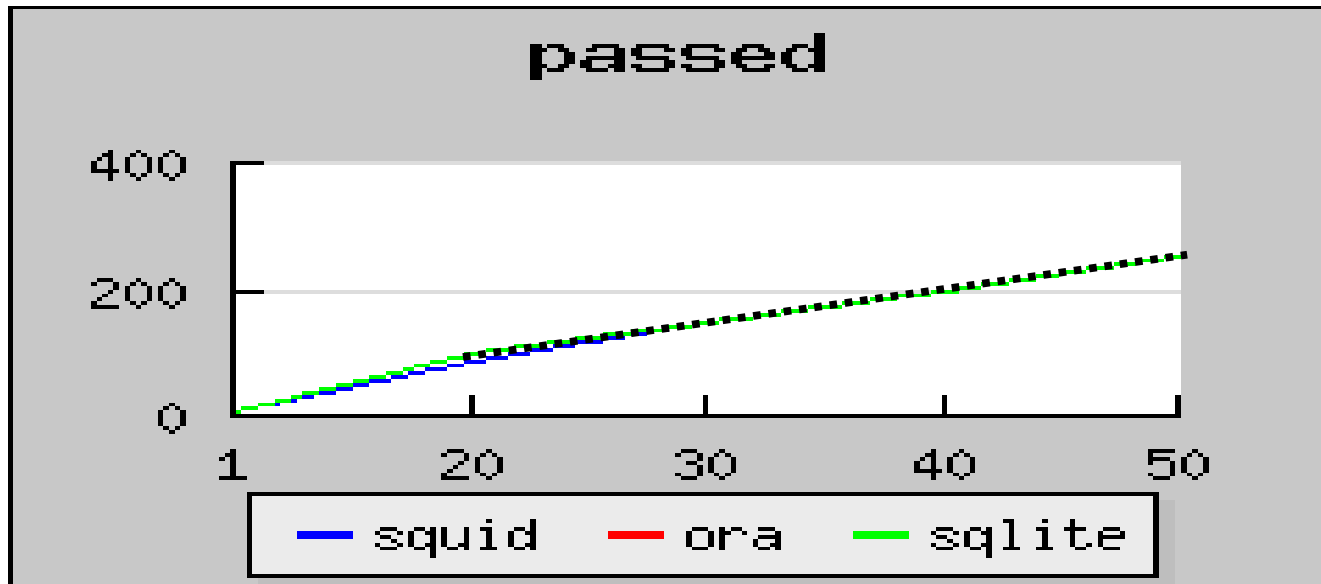
Y access is the average of measures taken by each client

X access is max number of clients that should run,

the different access modes to the data appear at different colors.

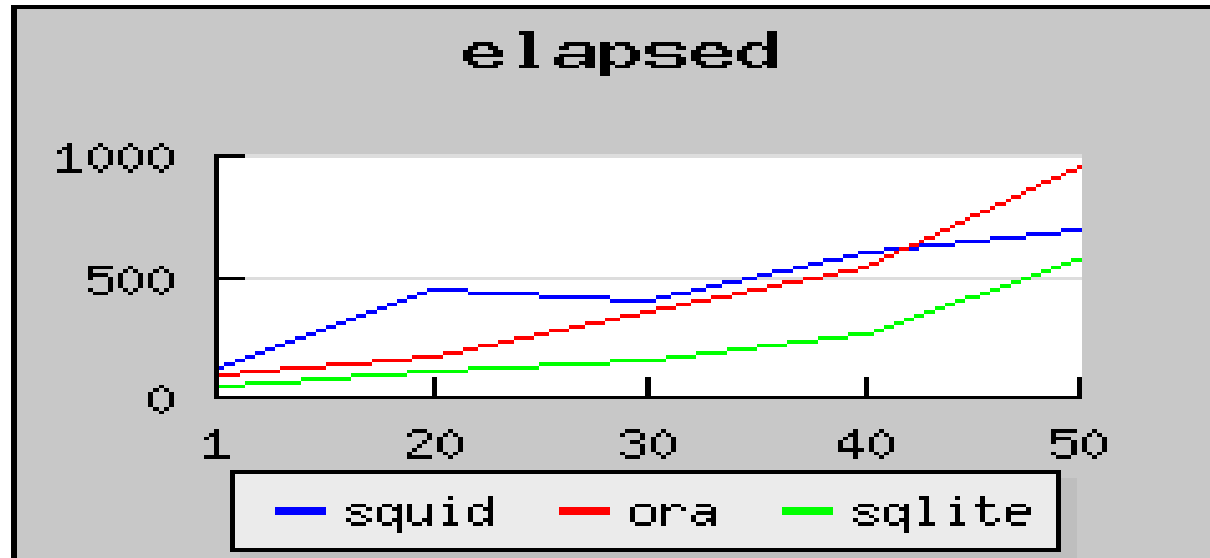
where all modes agree, we see only the color of the latest mode

amount of clients that passed



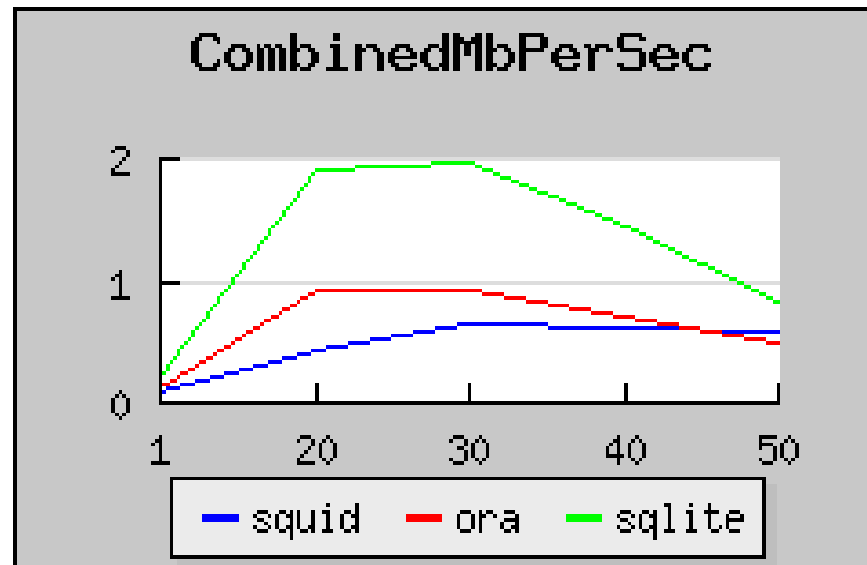
For each unit at X axis, 5 clients should run, once after the other.
Hence, expected passed are: $(1 \times 5 =) 5, 100, 150, 200, 250$
expected:

elapsed time to read



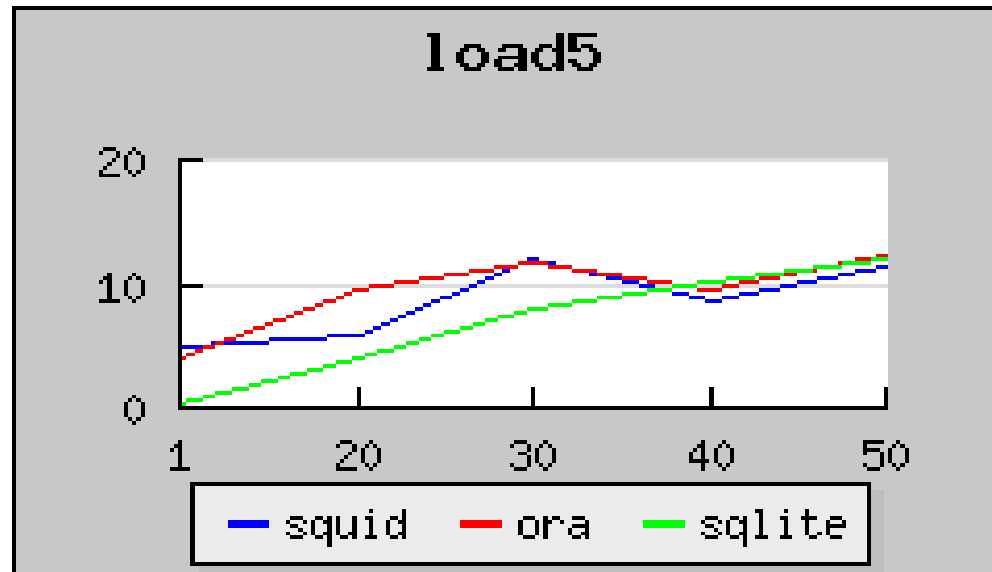
- 'Elapsed' is the average number of seconds that each client runs,
- **Elapsed tends to grow with max-num-clients up to hundreds of seconds**
- At bigger numbers of clients, squid does better than Oracle.
- sqlite is the fastest, but at 50 clients its advantage becomes smaller
- A few hundreds of seconds is acceptable, however 'remote' clients are expected to have TBD higher 'elapsed'

throughput - MB/sec



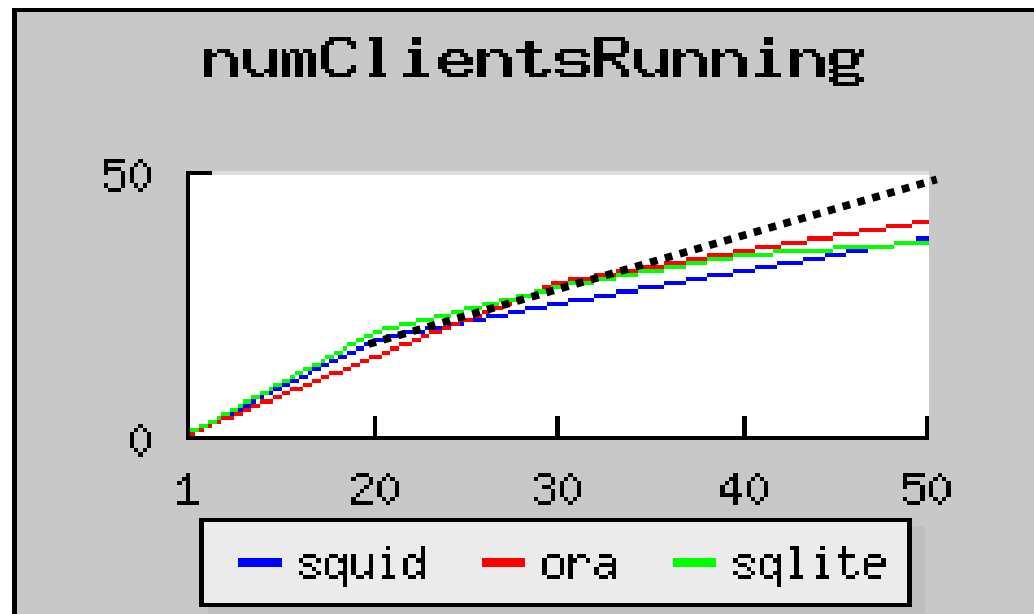
- * Combined throughput saturates at 20-40 clients.
- * The higher the number of clients – the more squid seems to out-perform oracle.
- * sqlite does better than squid and Oracle
(because it uses a copy of the DB on the file system,)
- * TBD: Sqlite slowdown at 40/50 clients may be due to client CPU or AFS access constraints

client host load



- 'Load' is the number of processes waiting for CPU.
- Client load is limited to 10 by purpose, in order to prevent overload

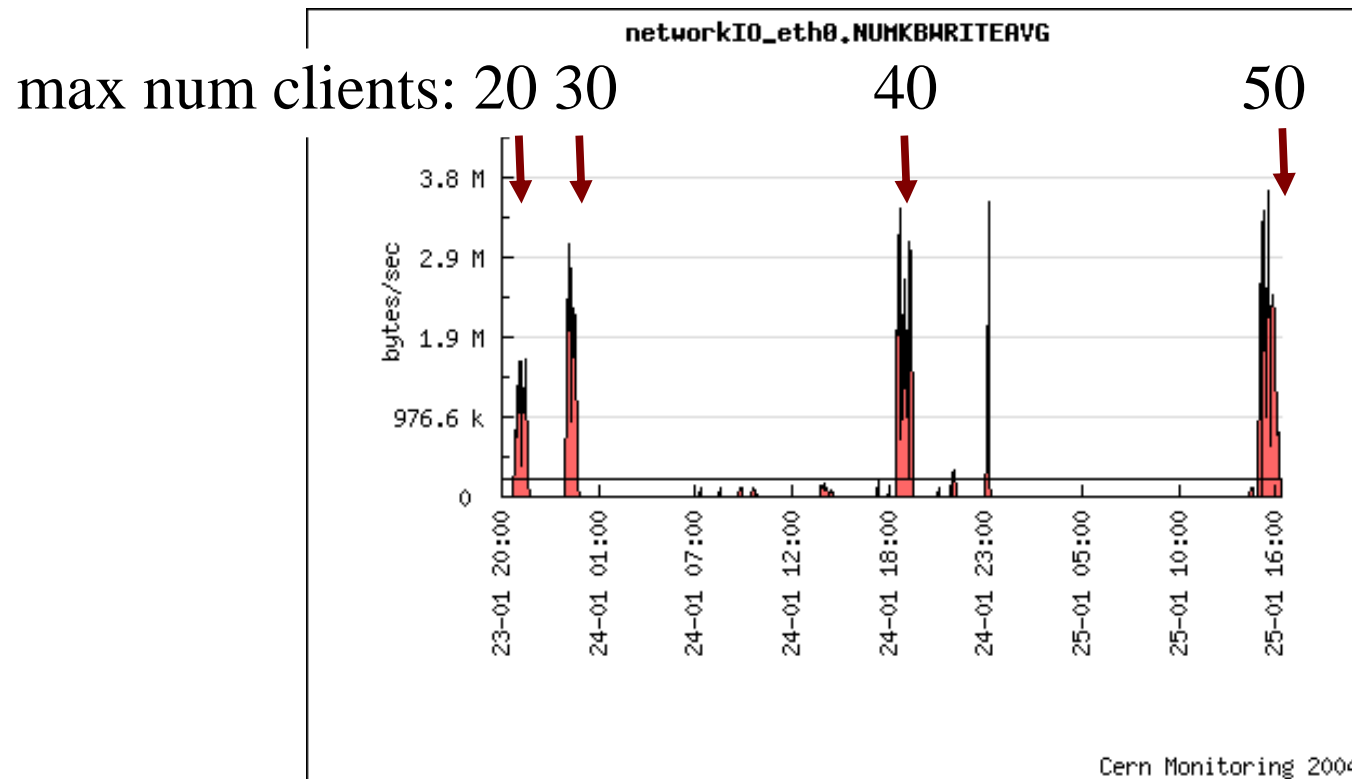
actual number of running clients



expected:

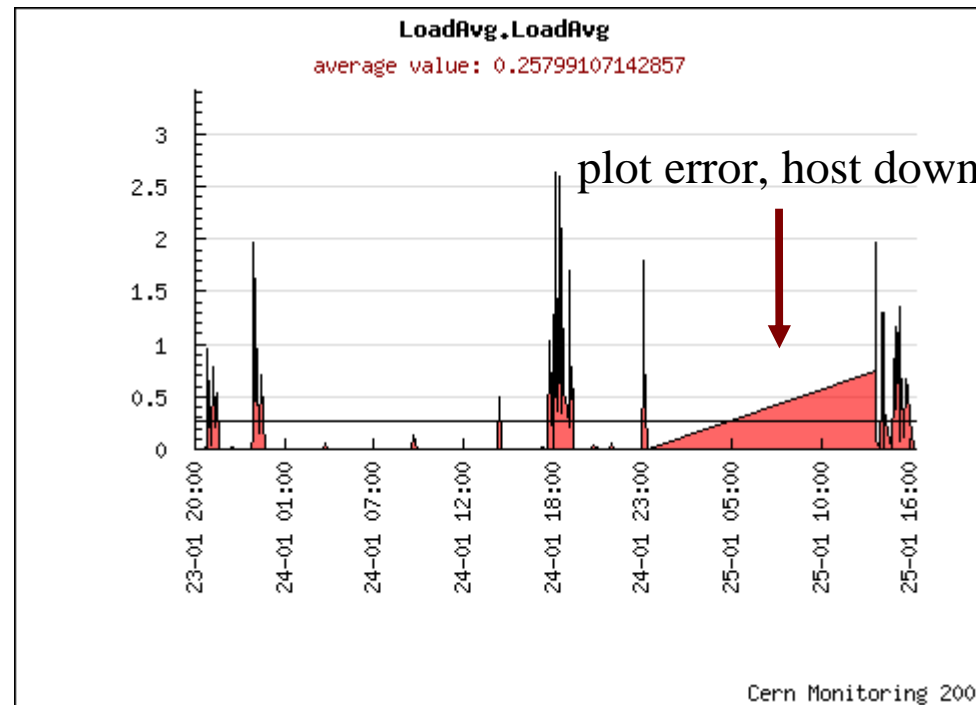
- At 40 and 50, more sleeping is done before spawning clients, to avoid overload

lemon: squid/frontier server write data



The amount of writing for 50 squid clients does not seem to be bigger than for 40 clients

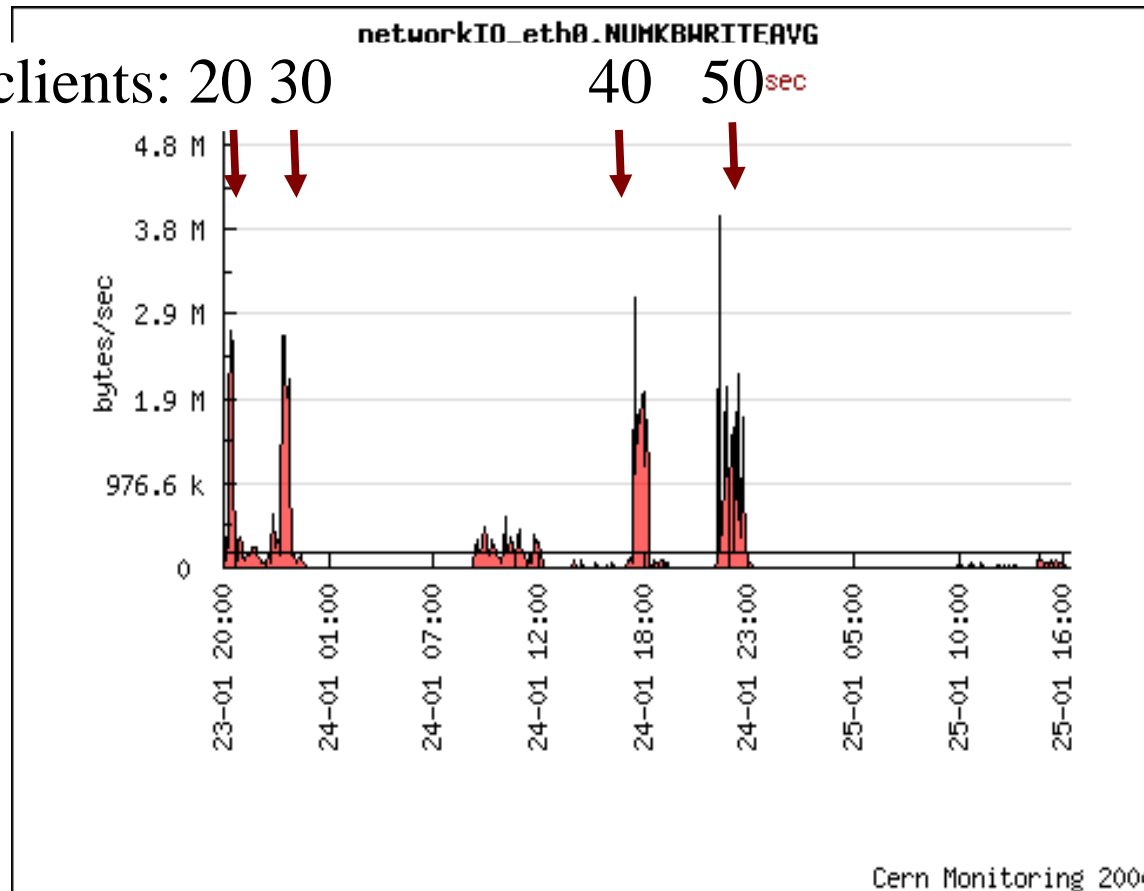
lemon: squid/frontier server load



Load of squid/frontier server is low, never above 3

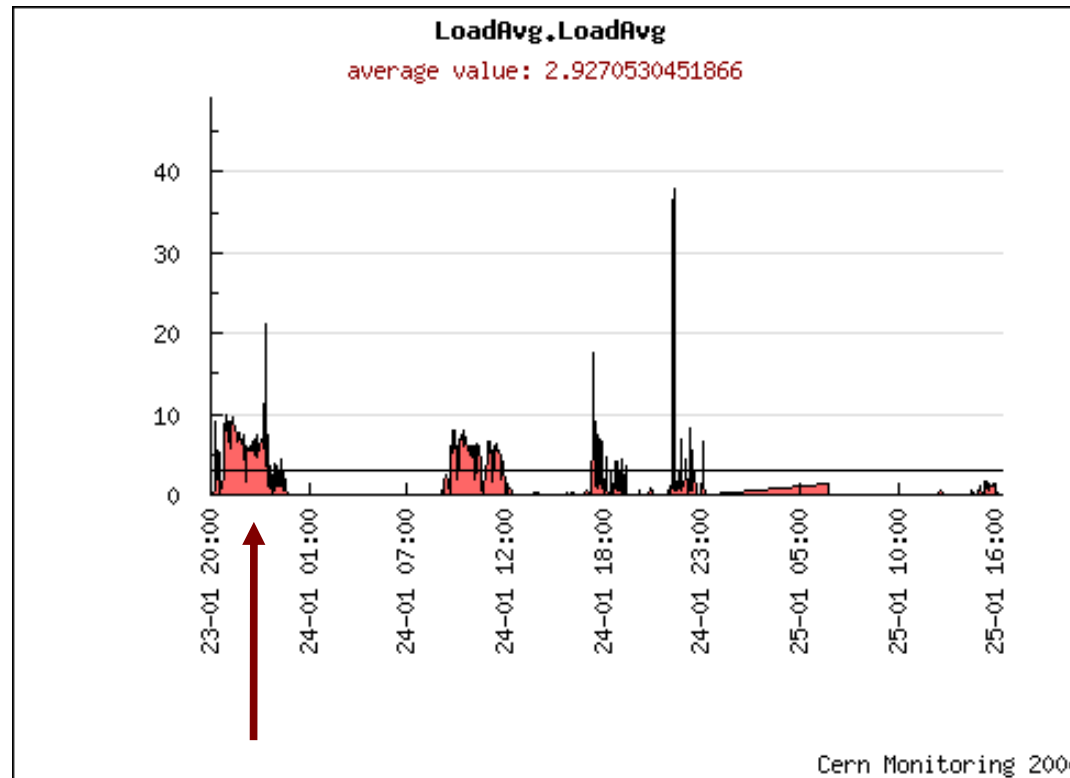
Oracle server write data

max num clients: 20 30



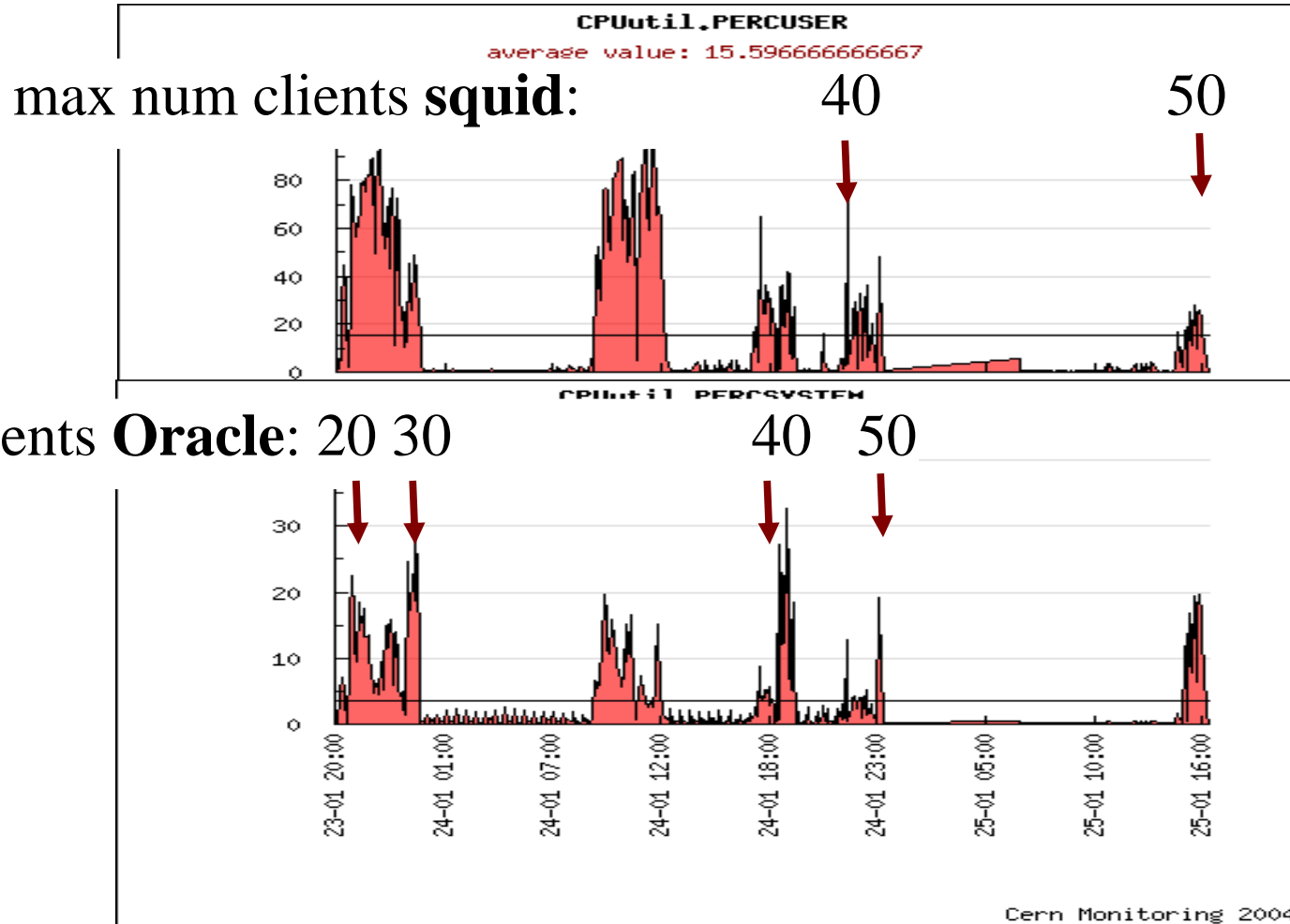
Oracle-client reading shows well,
Squid reads hardly show at Oracle server (, because of caching)

Oracle server load



- Oracle server more loaded than squid/frontier server
- At arrow, squid reading does seem to coerce visible load on Oracle server

Oracle server CPU



Oracle reads consume more Oracle CPU than squid,
but the amount of CPU consumed by squid is higher relative to its low traffic,
(maybe because frontier does not use bind variables)

results summary

- Scaling of this testing environment saturates around 40-50 clients
- 'elapsed' time tends to grow to hundreds of seconds
- squid MEM_HIT snapshot: 93%, meaning that squid caching works well
(in particular, the 'select 1 from dual' queries have been observed to be cached)
- Performance bottlenecks:
 - Clients – sleep to avoid high self load.
More clients are needed to further scale
 - Squid/frontier server – was not highly loaded - did not reach full capacity
 - Oracle server – worked harder than squid server, but not fully utilized.
TBD: Learn Oracle waits events, looking for potential performance improvements.

conclusions

- Stress test 100 athena reconstruction clients?
 - The available testing environment saturates around 50 clients
 - With actual stronger resources, I guess that 100 is feasible
 - Care should be taken to make sure that 'elapsed' is acceptable
- Compare data location: remote/local, and frontier compression level
 - Not done yet
 - I guess that working remotely, and optimizing compression level, will work in favor of squid

more conclusions

- Compare reading means: Oracle/squid/sqlite
 - squid seems to scale better than Oracle
 - sqlite is the fastest, but its elapsed grow fast around 50 clients (hitting client load limit)
- Supply input to decide if Atlas will benefit from using squid
 - For a conclusive recommendation, better do remote testing also
 - Yet, looking into CMS positive experience and assuming that no hard Atlas show-stoppers appeared at this testing, using squid for tier-2 retrieval seems to me reasonable

testing resources/plans - local

- lxb clients at CERN are scarce and 'local', yet easier to configure/control/monitor than 'remote' machines.
- Hence, it is planned to continue 'local' tests, shortly:
 - control squid compression level
 - (partially) simulate network overhead by causing local traffic between client and server to go via remote site, or simply sleep
 - ? Repeat tests with new CORAL version with enhanced frontier performance

testing resources/plans - remote

- Similar testing is planned to be done remotely:
 - Weizmann may allocate some client resources
 - Paul Millar from University of Glasgow suggested to arrange allocation of machines for testing

related work

- Richard Hansen is working on causing local traffic between client and server to go via remote site, or simply sleep.
- Submitting similar jobs via the grid simulates the actual workload better than submitting jobs from dedicated clients. Stefan Stonjek and Sasha Vanyashin are looking into that direction.

link

- Raw data related to this presentation is available at <https://webafs3.cern.ch/dfront/cool/vc133new/> at the line: athenaRead_07-01-25_15-25-13