Scalability Test of Hadoop-based Storage Element at UCSD

James Letts, Terrence Martin, Haifeng Pi,
Abhishek Rana, Frank Wuerthwein

OSG Storage Forum at Fermilab
June 30-July 1, 2009

Quick Facts about Hadoop-SE as of today @ UCSD

- **Bestman Server** bsrm-1.t2.ucsd.edu, 8-core, 8GB Mem, GUMS-authen, dynamic gridftp selector (developed by Terrence)

- **NameNode Server** (hadoop-0.19.1-8.e15) proxy-1.t2.ucsd.edu, 8-core, 16GB Mem

- **15 DataNode Client** combination of 4-8 core machines, 8-16GB Mem, 1Gb up-link, also working as WorkerNode

- **27 Gridftp Server** ~67% run dcache pool and WN, ~33% run Hadoop-DataNode

- **ALL WorkerNode** (Hadoop-fuse-0.19.1-8.e15) Fuse-mount (read via fuse, write via srmcp)

- **6 interactive machines** Fuse-mount (read and write via fuse)

- **10+ active local users** 42 TB in total, 25-35 TB used in the past week. As planned, all user data will be migrated to hadoop-SE (almost done).

- **ALL grid users** CMS VO users have write access, other VOs have pool mapping and request-based account ...

- **Daily adminstration** new release (rpm, ROCKS), balancer, new user setup, monitoring, very few problems reported by user once it is setup and validated. Less than 20 min a day for operation

Scalable Architecture

Highly distributed services across worknodes

- Hadoop data-node client
- gridftp server

Two replication of files in hadoop

- Balance between space usage, and scalability and reliability
- Block size 128 MB

In the long run, possible bottlenecks in central service, architecture and network (most of components are quite scalable)

- Bestman
- GUMS
- Hadoop name-node
- Cluster arachitecture
- WAN

Scalability Test Goals

Debug the release and system configuration

Understand the performance

- define a number of tunable parameters
- find the most possible use cases and access pattern from users

Look for weakest point in the architecture

- I/O, memory usage ...

Establish test/validation for the hadoop-SE

- management of raw monitoring data
- system analysis procedure
- observables

Gain more operation experience

Specification of the Test Jobs

Scalability Test of Bestman

grid jobs run srmls

grid jobs run srmcp with small files (1KB)

Scalability Test of Gridftp and Hadoop

grid jobs run srmcp with large size (1GB)

PhEDex loadtest

Scalability Test of Hadoop via Fuse access

local jobs run CMSSW against 1 file

local jobs run CMSSW against 10 files

Grid jobs are sent via glideinWMS

similar to the normal user data access pattern.

controlled ~1000 jobs are concurrently running for the test at the largest scale, because limited size of our hadoop system

Limitation of the current Test

The test was run on the production system, which inevitably has some limitation in how the tests to be organized.

- Test won't show the physical limit of some standalone components, because datanode, worknode, gridftp, dcache pool are running “together”
- The grid jobs are not under full control. GlideinWMS and direct condor job submissions won't give a smooth curve of job from idle to run. The results possibly dependent on the status of the CE during the period of the test
- The impact of the I/O of the worknode from other user jobs are not under control

So we can take a factor of extra 10-30% w.r.t. the ideal scalability that can be achieved due to the limitation of the current tests

But the results are more realistic ...

Some numbers from STEP09

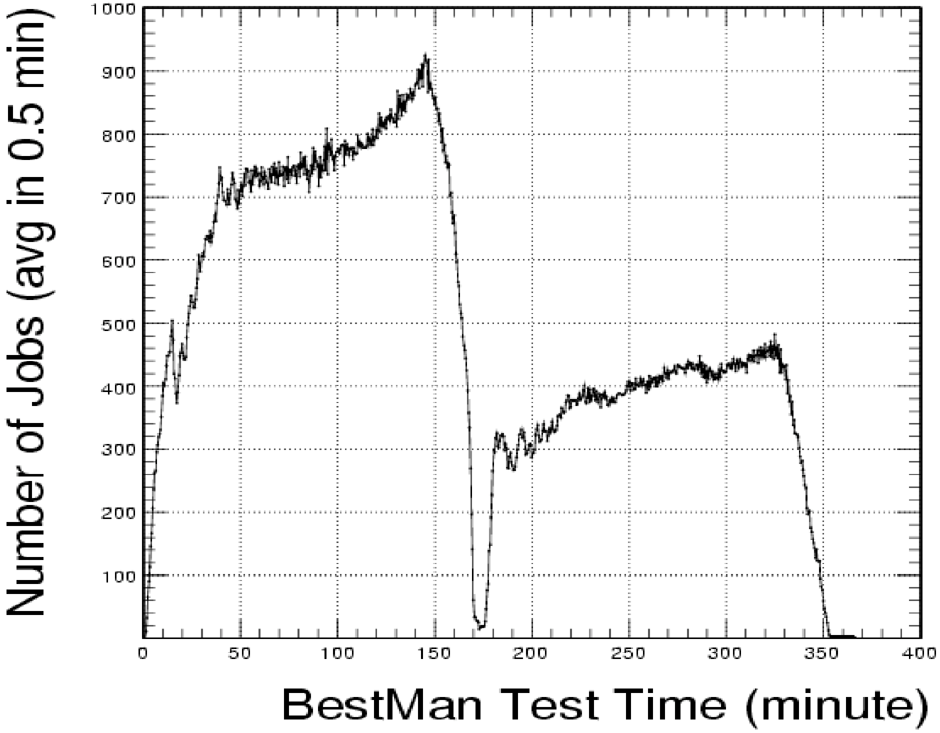
CMS has ~20-30,000 Cores in T2/3 globally across ~50 sites.

We have seen single users use up to 10k jobs at once.

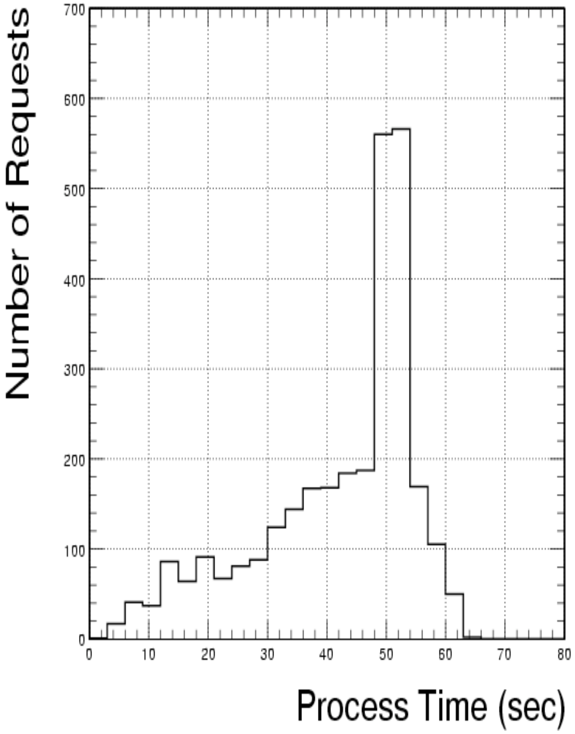
Testing at the scale of O(1000) simultaneous “stage-out” is not completely unrealistic.

srmls on Bestman

Up to 900 srmls active during 30sec period



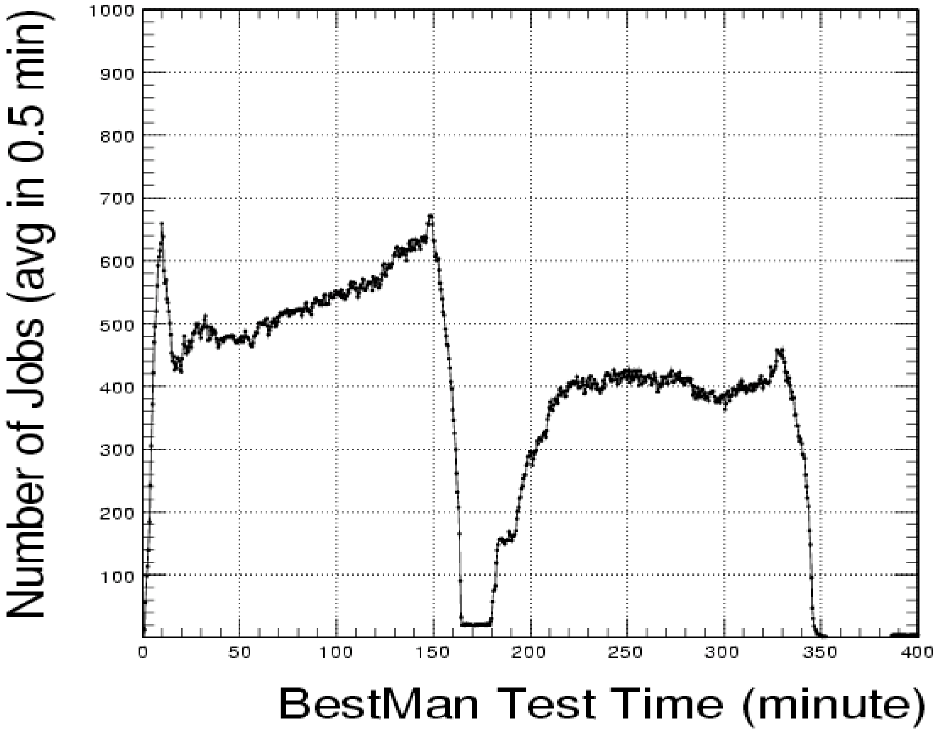
Most likely time per srmls = 50s



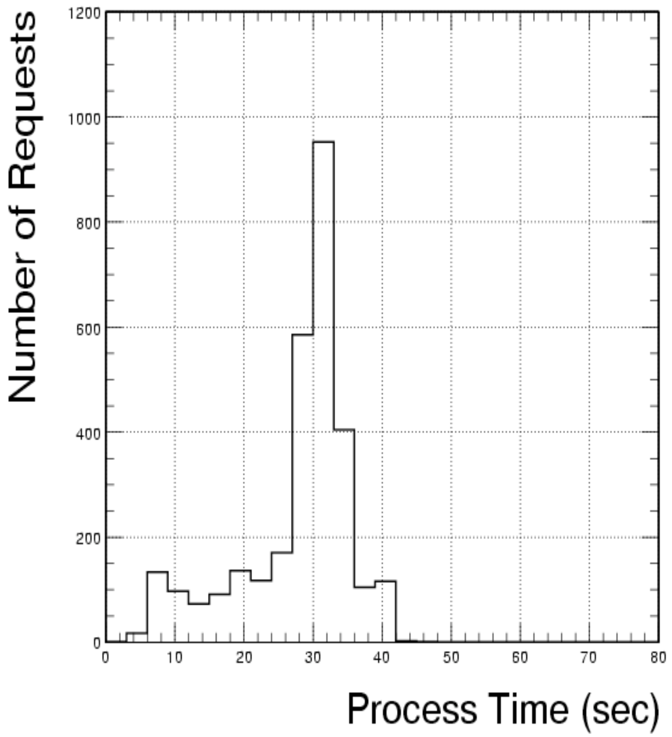
Test measured via job output file which record the start_time and end_time of the srmls command

srmcp of small files (Bestman test)

Up to 700 srmcp active during 30sec period

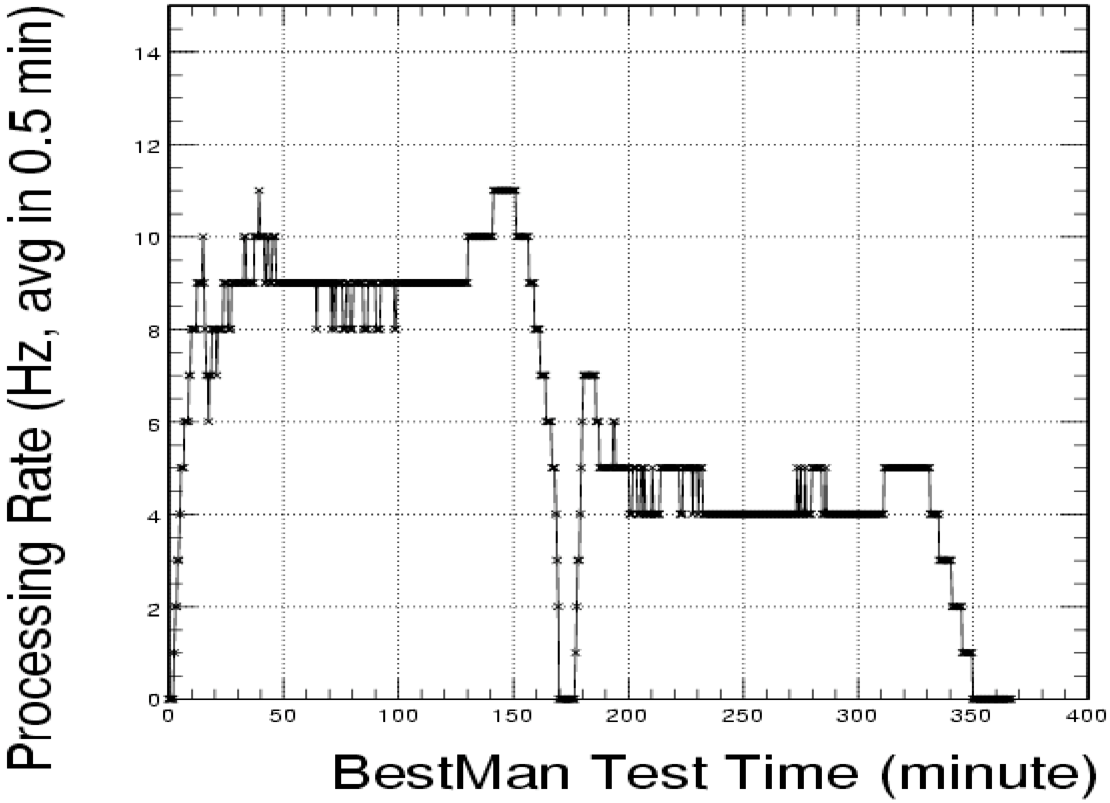


Most likely time per srmls = 30s



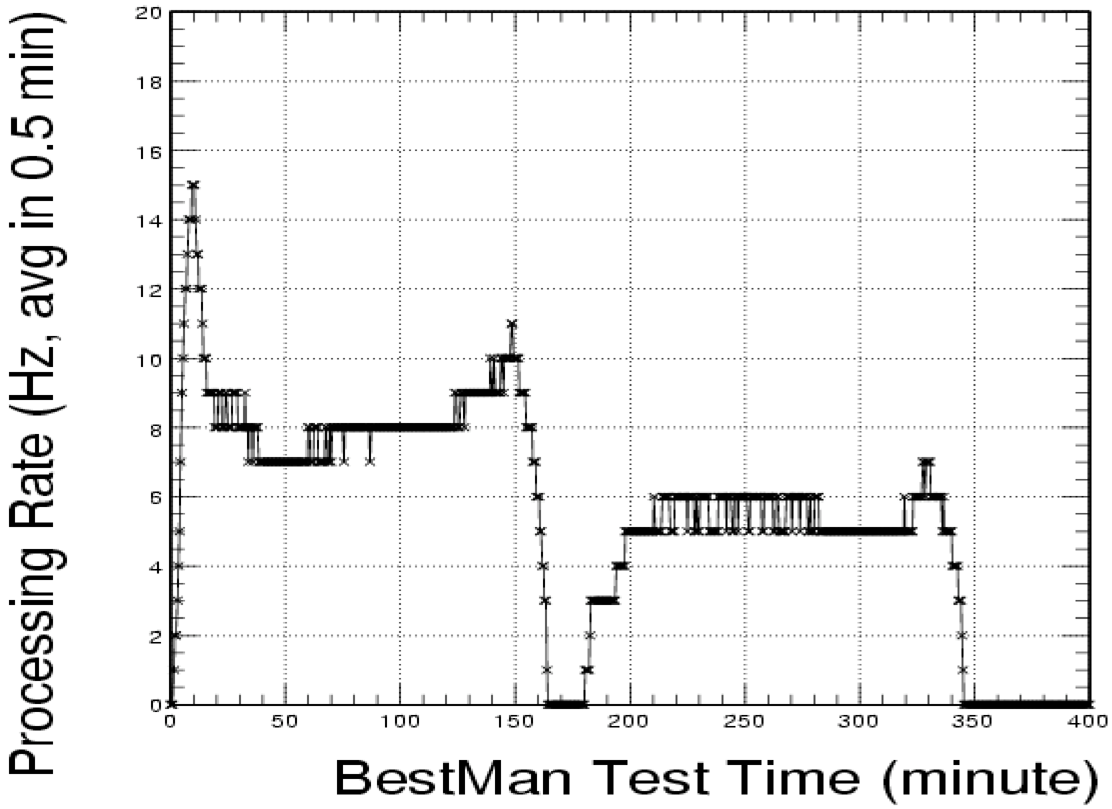
Better performance of srmcp small file (1KB) than srmls
This requires further study to understand it better.

srmls rate of Bestman



Sustained performance of around 10Hz

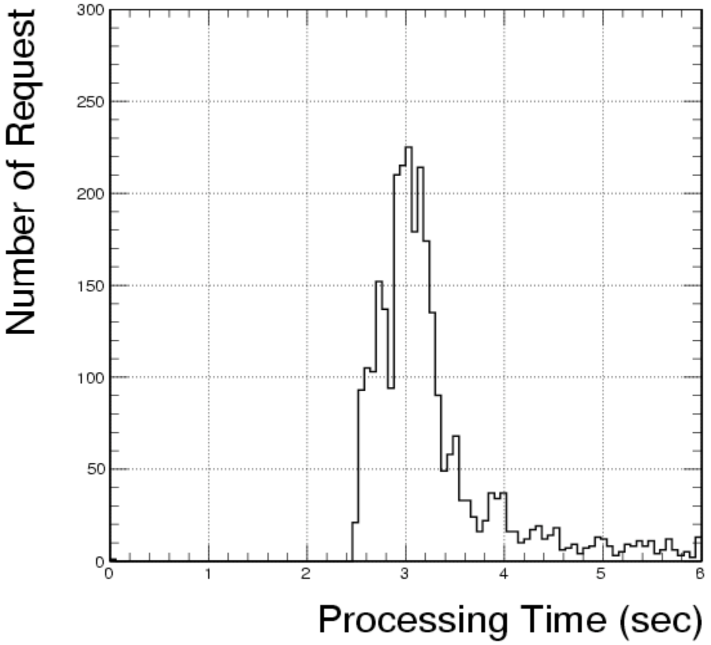
srmcp rate of small files (Bestman test)

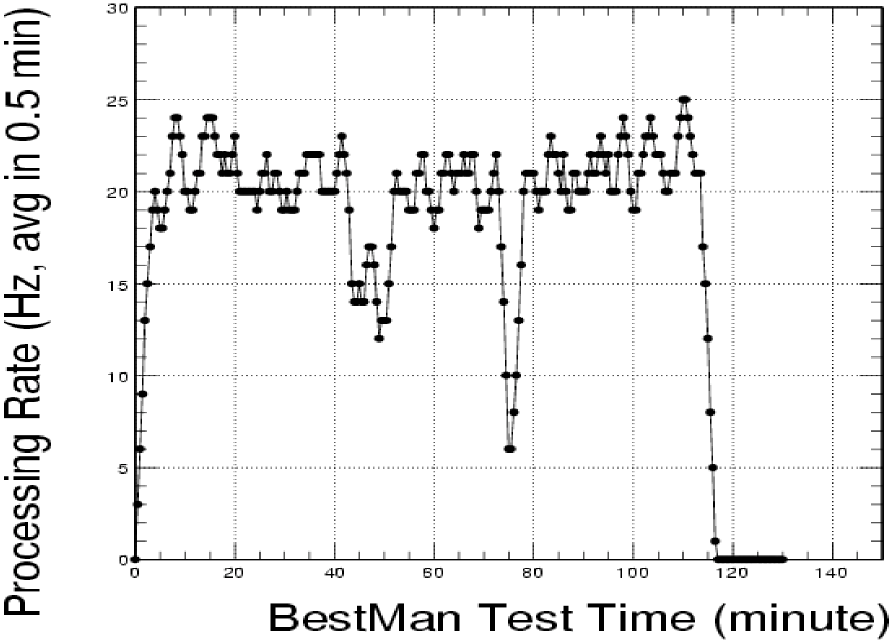


Both srmls and srmcp tests shows the highest rate of 10 Hz in the Bestman
Others have seen srmls of up to 50Hz
Differences remain to be understood.

Turn-off Delagation in srm client

Without proxy delegation, the scalability will be significantly increased.

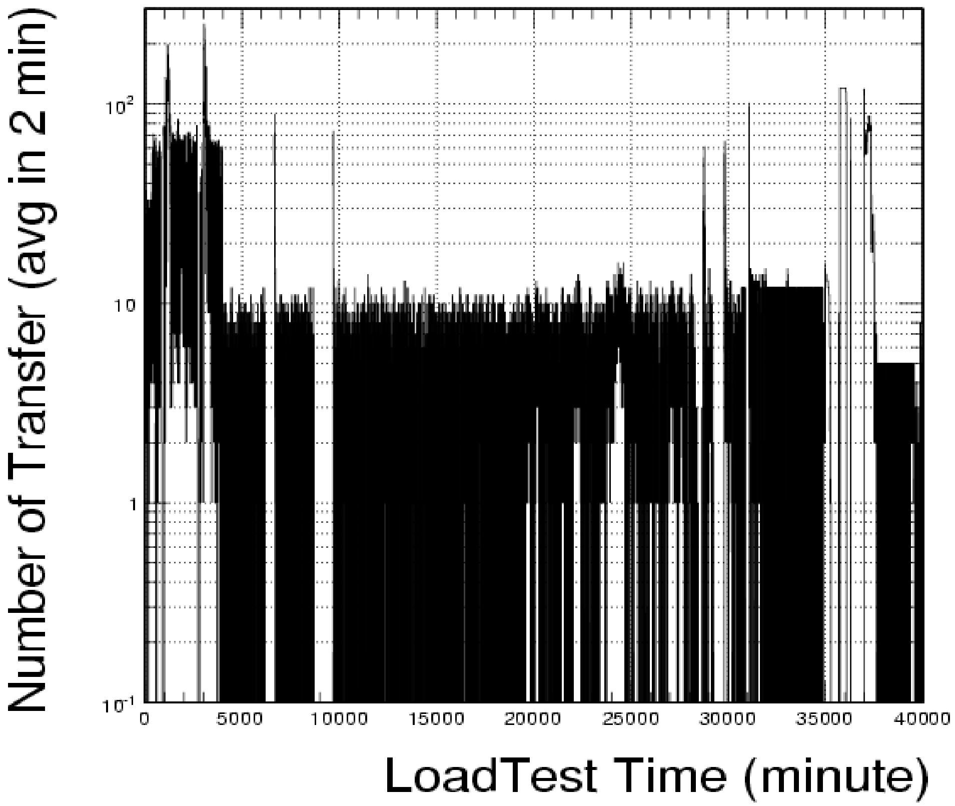


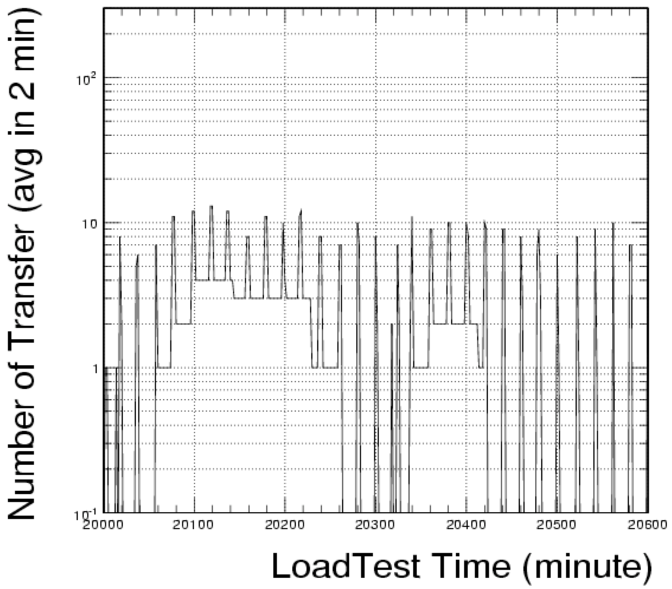


Average processing time per job decreases from 30+ sec (w/ delegation) to 3 sec (w/o delegation) with ~800 simultaneous jobs in the system

Average processing rate increases from 10 Hz (w/ delegation) to 20-25 Hz (w/o delegation) with ~800 simultaneous jobs in the system

PhEDEx Load Test between UCSD and Caltech

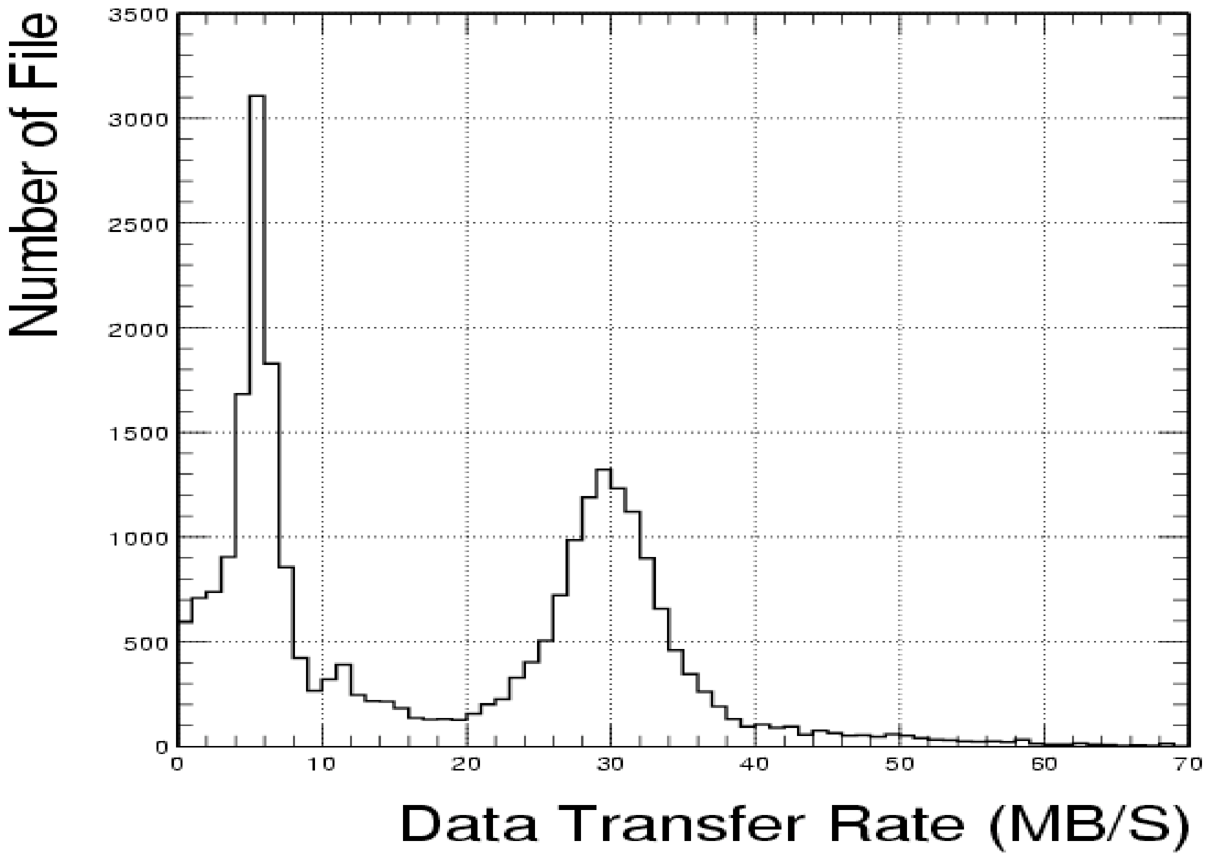




SE-SE file transfer, continuously running for months, network traffic via Cenic 10 Gb shared with others

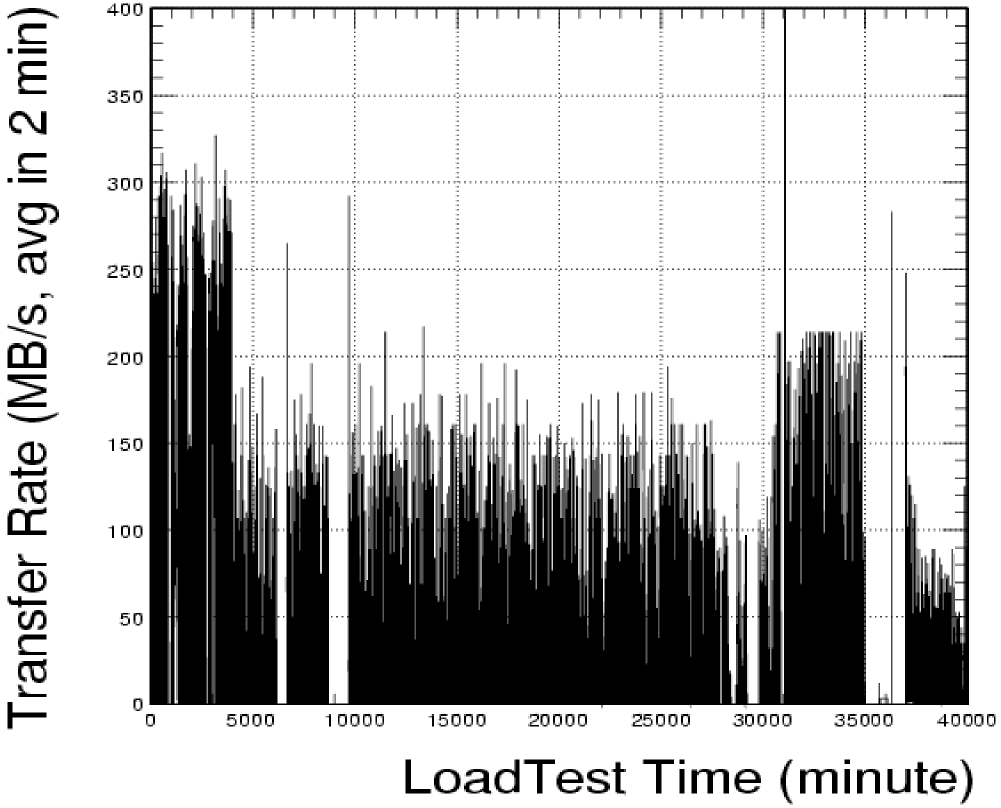
Test results are collected from gridftp log file

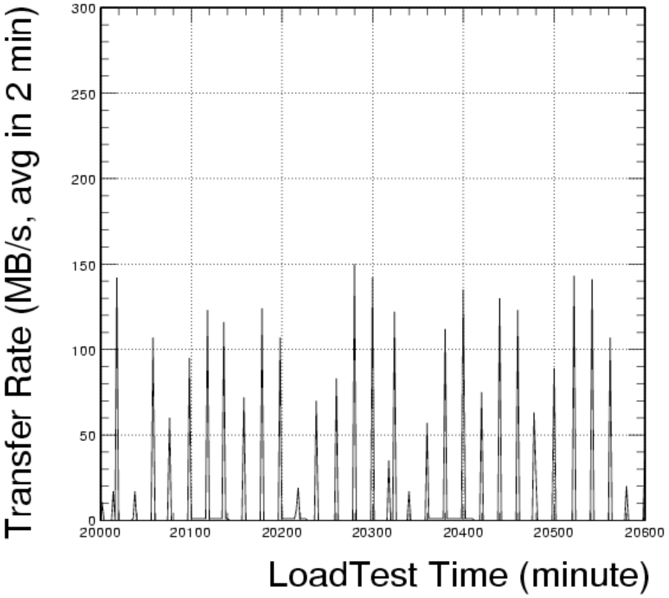
Performance of Single File Transfer in Loadtest



30 MB/s is a reasonable rate for transfering single file over the grid with 10 streams and average 10 file transfers a time
Detailed structure not yet understood.

Accumulated Rate in LoadTest

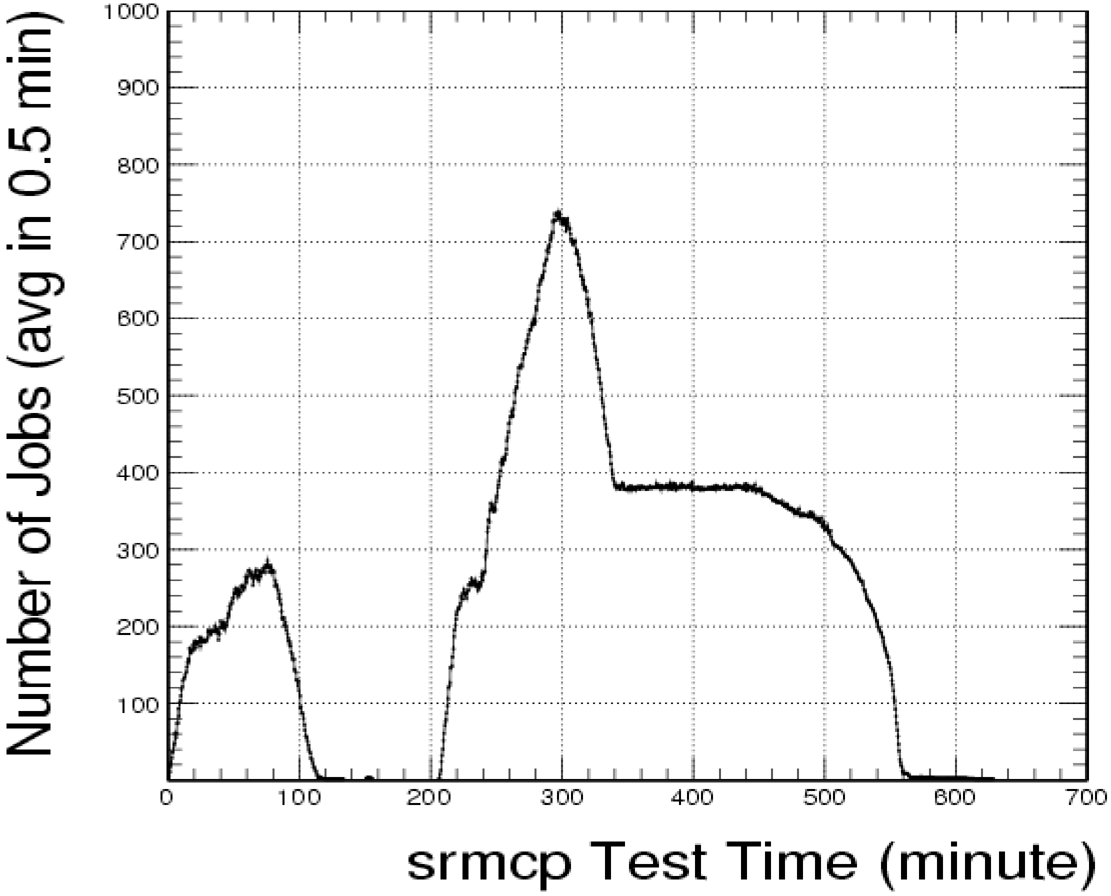




150 MB/s transfer rate from loadtest is sustained for months
Scale here is by design. We have not tried to stress the system via this test.

Large Number of File Transfers

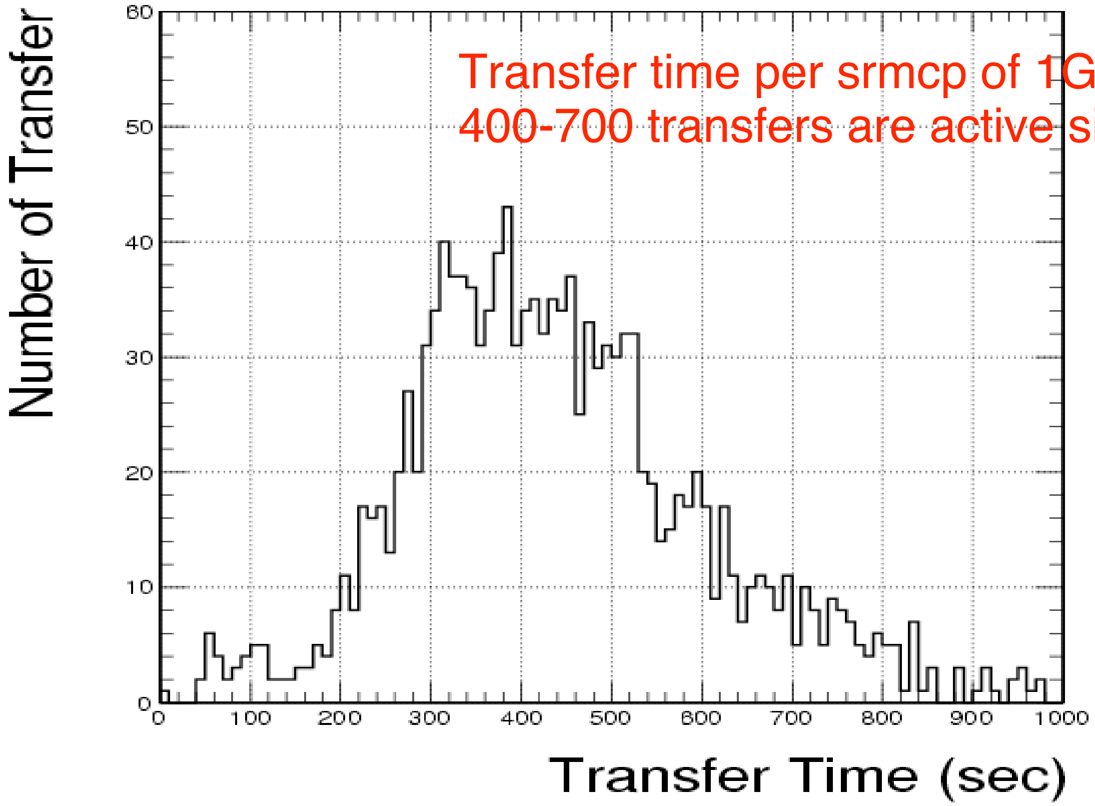
Up to 700 simultaneous srmcp of large files



Each job transfers 1 GB file with one stream

Test results are collected from job log file

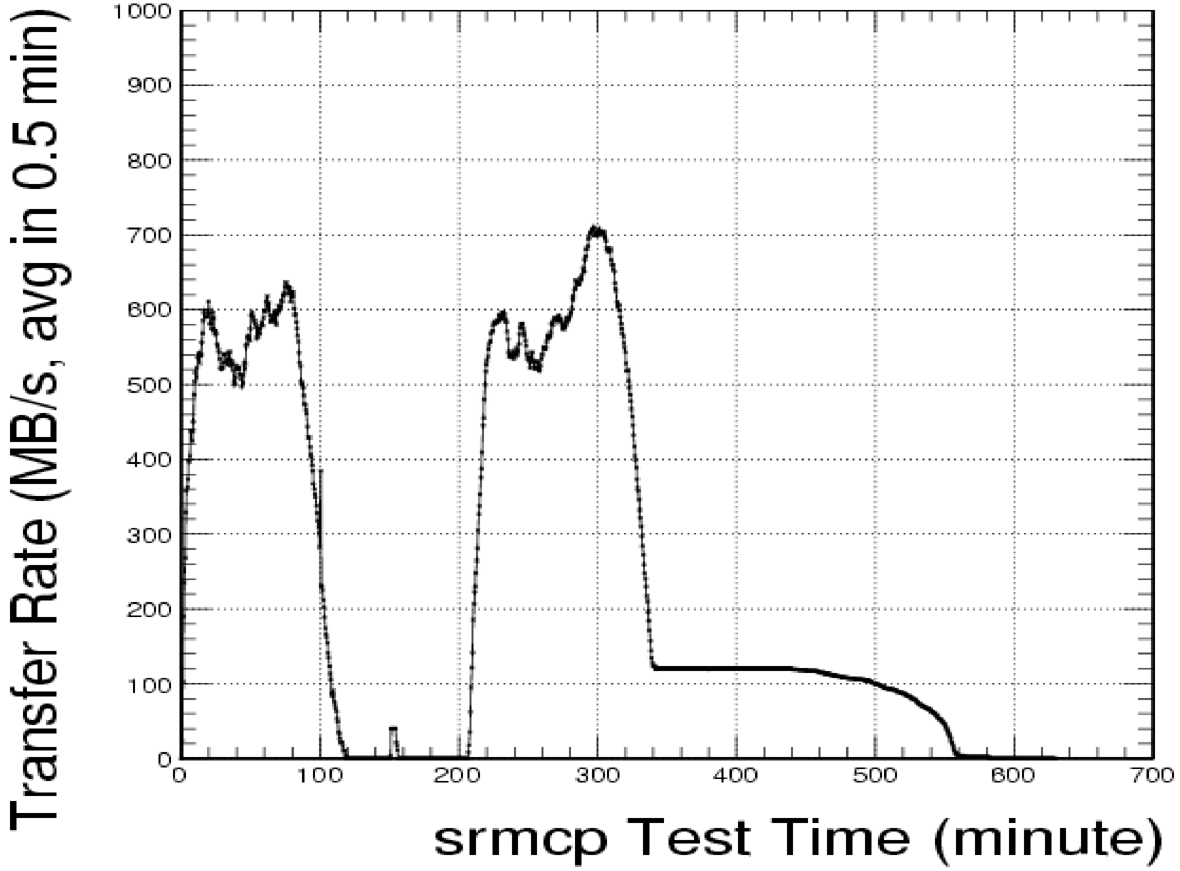
Performance of Single Job Transfer



Transfer time per srmcp of 1GB file when 400-700 transfers are active simultaneously.

The scale test is characterized by many file transfer and each transfer only taking small amount of bandwidth, in contrast to loadtest, small number of transfer, each transfer taking a decent amount of network

Accumulated Rate



Peak rate
700 MB/s

sustained for hours
550 MB/s

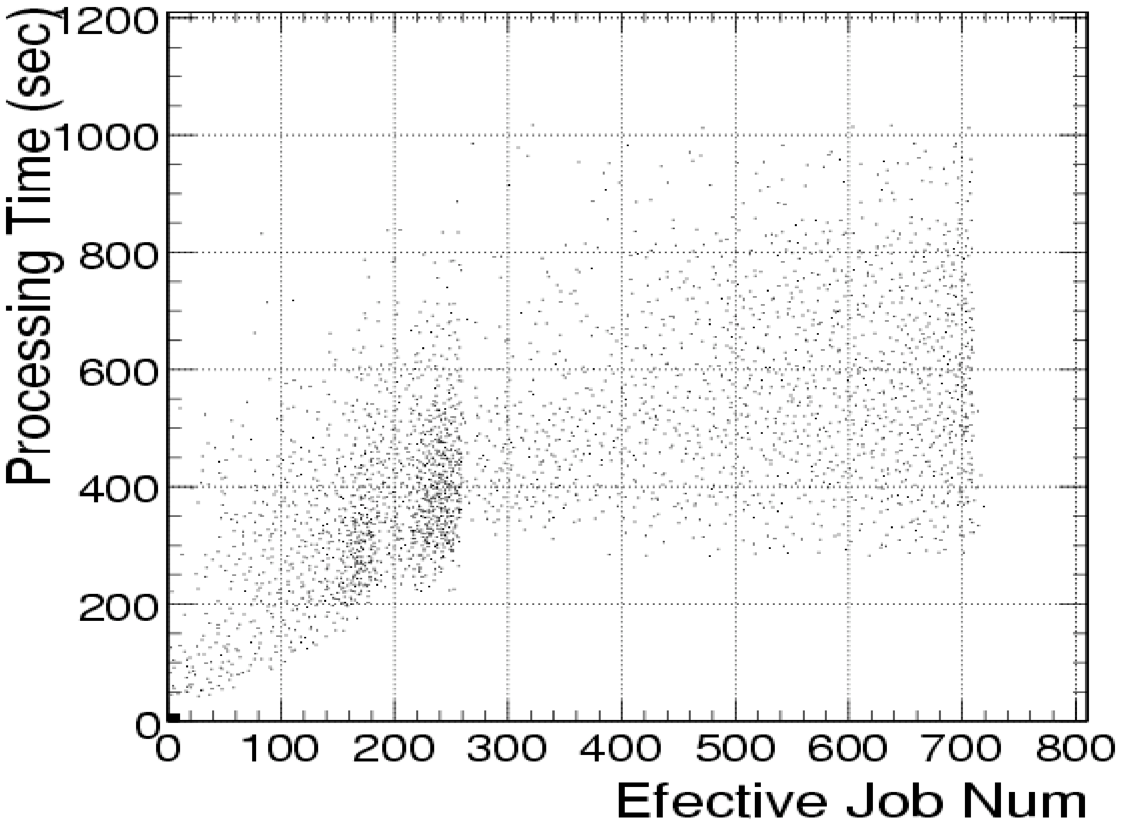
“Tails” probably explained by sites with NAT as sources for srmcp from WN.

Three Calculations to estimate the Peak Rate

(1) Limit from gftp server count: Highest theoretical performance of single gridftp transfer 90 MB/s. Max write rate 27gftp * 90MB/s = **2430 MB/s**, but at present we are limited by number of datanode to accept those data from gridftp.
(2) Limit based on LAN network: Total 15 data node, ideal max write rate 15 Gb/2 = **930 MB/s.** Factor ½ is assuming replication takes 50% of the internal network.
(3) Limit based on hardrive IO: If assuming max rate of writing to disk is ~90MB/s combined with gridftp (as measured separately), the max write rate is 90 * 15/2 = **675 MB/s**

What we see, 700 MB/s, is consistent with the estimated maximum for hardrive IO.
=> adding more datanodes into hadoop should increase total IO.

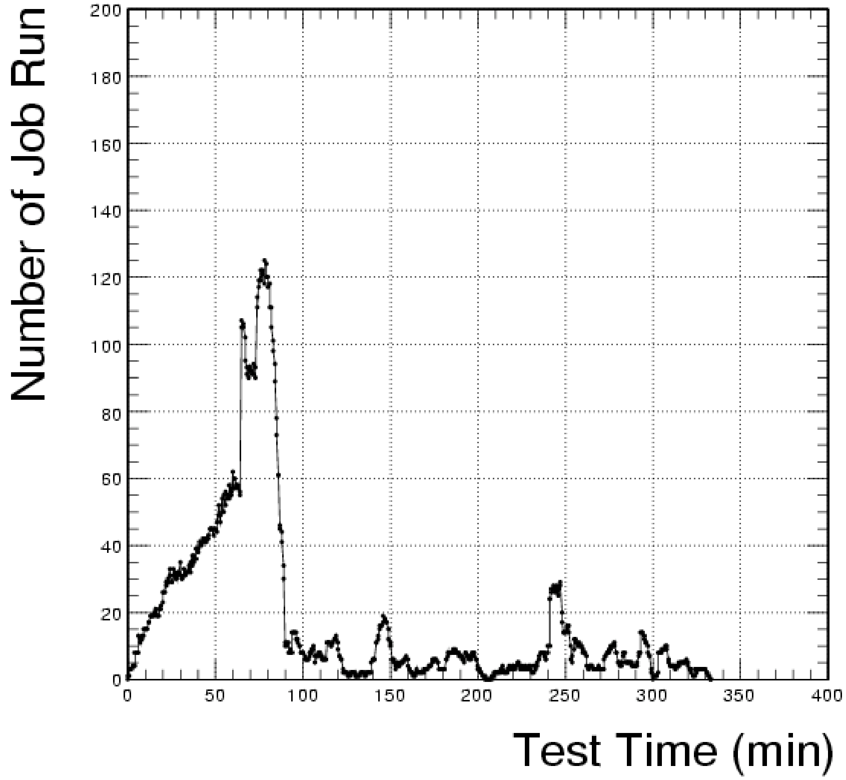
Correlation of single transfer time and number of transfers in the system



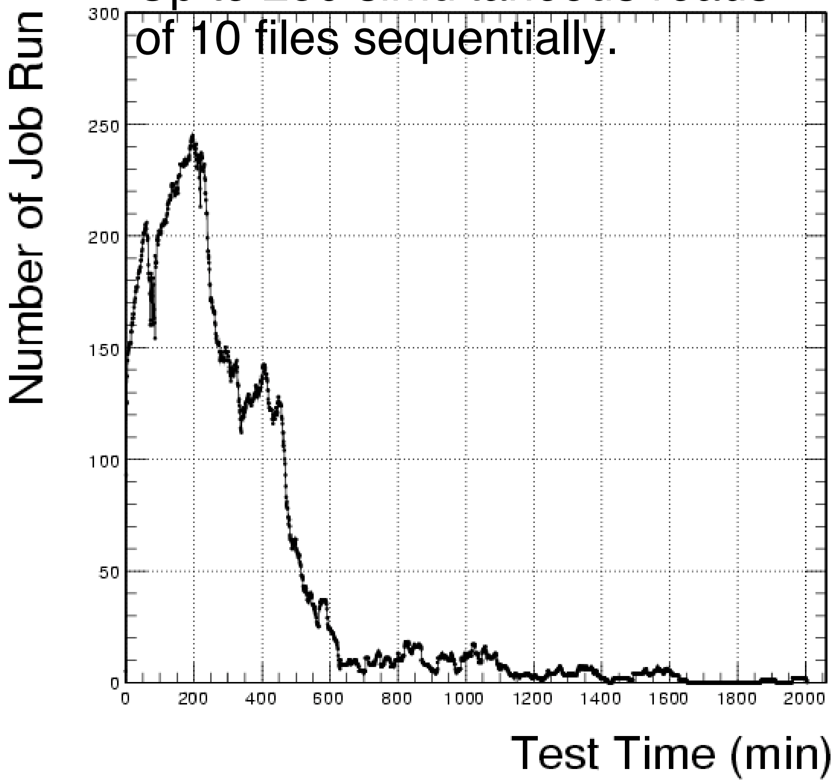
Average time per srmcp increases with # of simultaneous srmcp.
Spread in time per srmcp increases with # of simultaneous srmcp.

CMSSW Jobs accessing Data via Fuse

Up to 120 simultaneous reads of one file



Up to 250 simultaneous reads of 10 files sequentially.

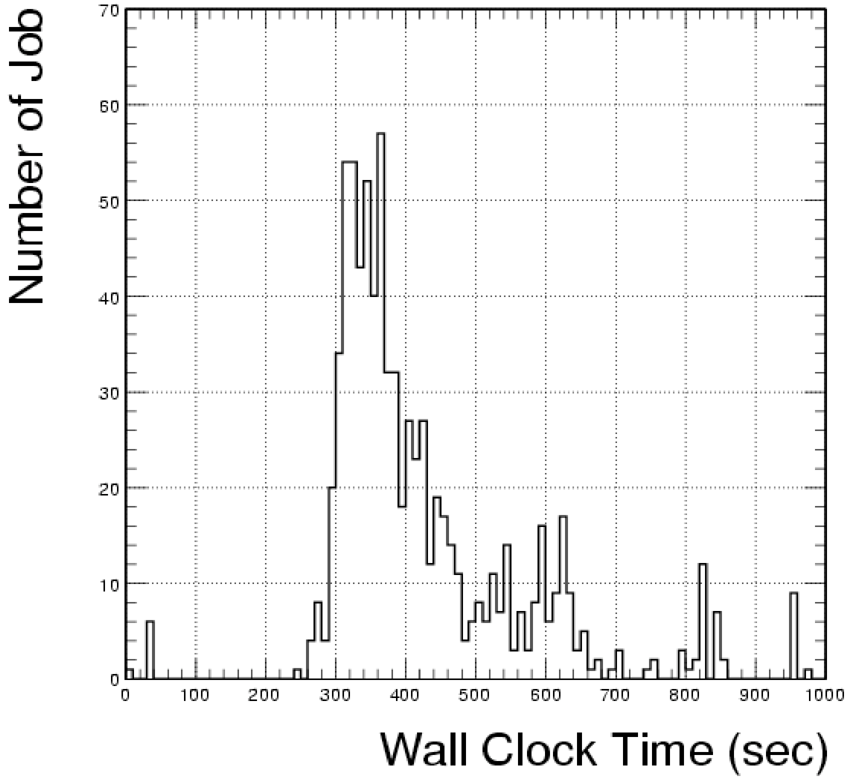


CMSSW jobs consuming large amount of data
Left: all jobs access one same file (1GB in size)
Right: all jobs access ten files (10 GB in size)

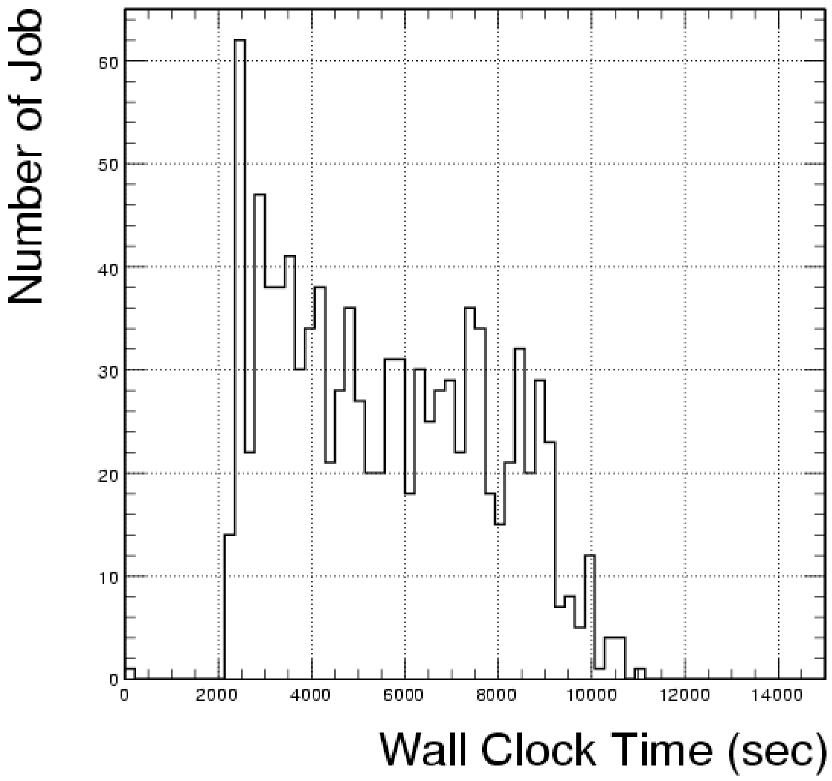
Each 1GB file has 8 blocks (16 with replications). Good distribution of all the blocks in the hadoop datanode

Average Jobs Processing Time

Read time from local disk (~250sec) is comparable to read time via FUSE



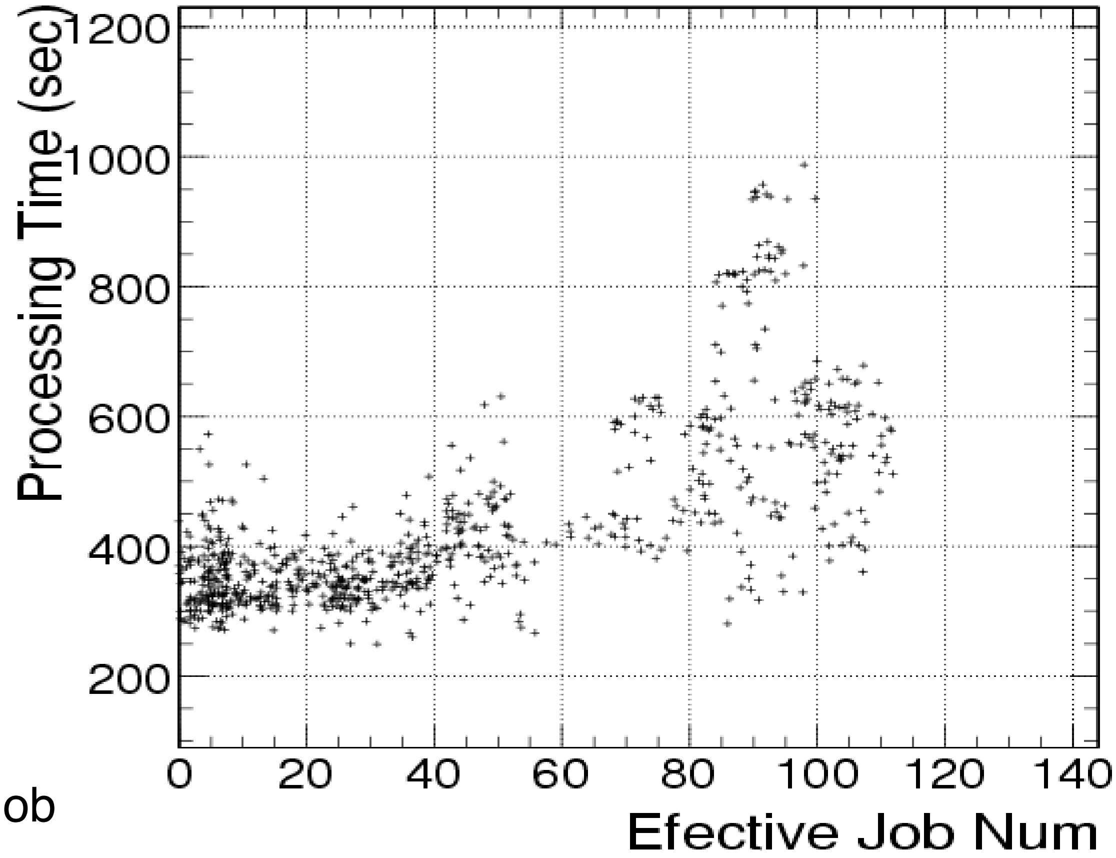
One job access 1 file



One job access 10 file

Difference in performance spread between the two tests not yet understood.

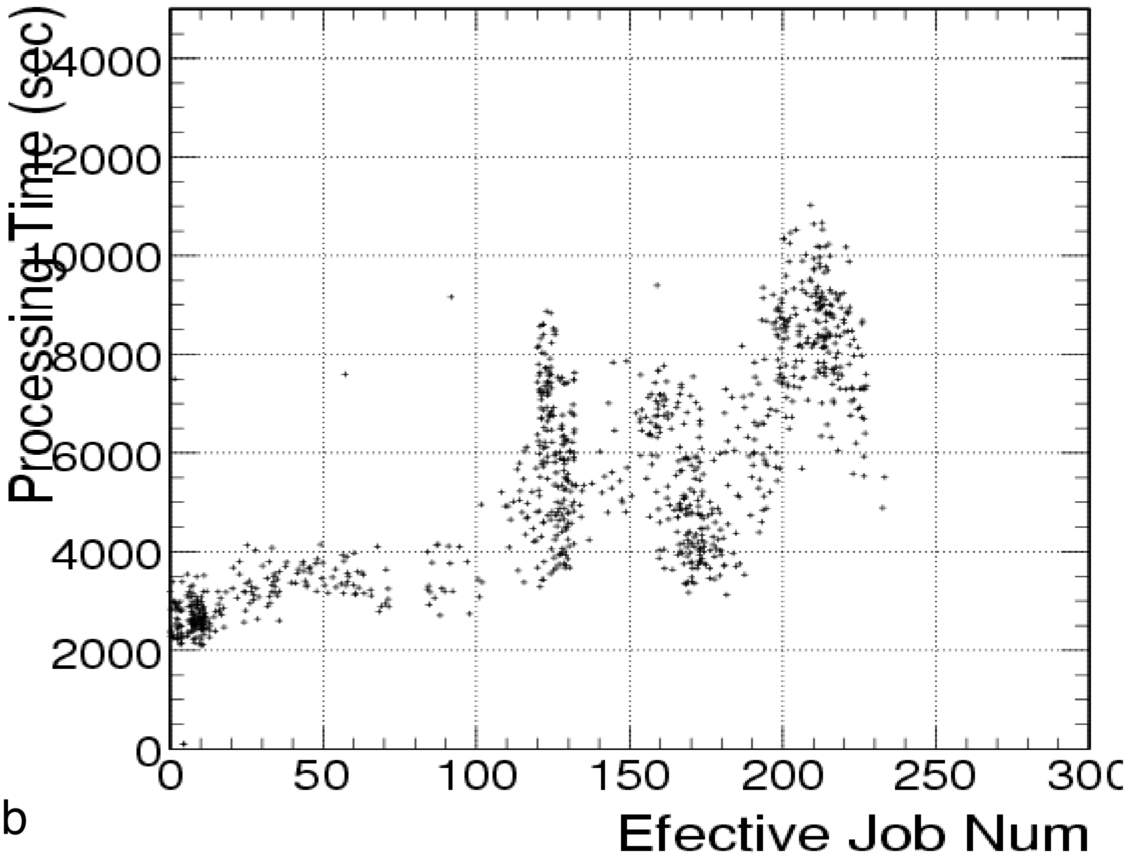
Correlation between Job processing time and number of same jobs in the system



One file read per job

Apparent increase in spread of processing times per job for more than 70-80 simultaneous jobs.

Correlation between Job processing time and number of same jobs in the system



10 file read per job

Apparent increase in spread of processing times per job for more than 100 simultaneous jobs.

Summary

Excellent scalability of Hadoop-based SE is observed

- No damaging effects were found in the hadoop-SE during the test
  - Hadoop appears reliable even under extreme conditions
- Most the results are in line with our expectation in terms of physical limits of the network, I/O of each component
- The system under heavy stress (I/O, memory, CPU ...) is still responding with reasonable performance

To-be-investigated

- Understand Bestman scalability
  - currently it has a limit of 10Hz
  - Brian sees a limit of 50Hz -> difference in tests, and maybe installation?
- Continue study why we see accumulated transfer rate 600 MB/s of our system
  - Add more storage to hadoop and verify that performance scales
- Understand how much file transfer rate is limited by the remote site.
- Find the limit of Fuse ... currently we are only running at ~200 concurrent reading
- Any limit set by hadoop system, although most of the limits we observed are set by the network or architecture ....

Appreciate Brian Bockelman, Micheal Thomas ... for the help throughout the tests and providing the bug-fix new releases!