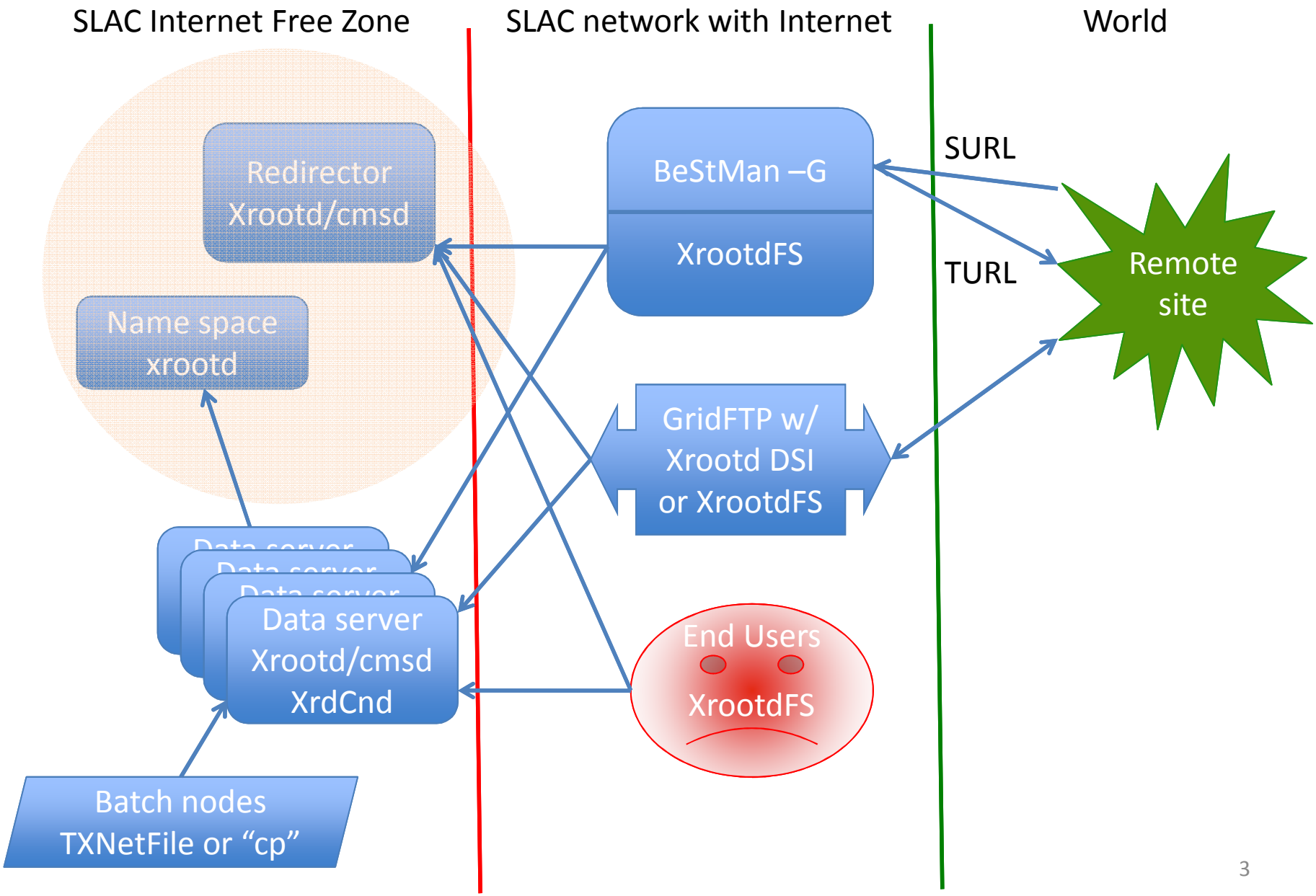
SLAC Experience on Bestman and Xrootd Storage

Wei Yang
Alex Sim

US ATLAS Tier2/Tier3 meeting at Univ. of Chicago
Aug 19-20, 2009

Over View of what SLAC has

Storage Architecture



Storage Components

- **Bestman Gateway** ← T2/T3g
- **XrootdFS** ← For users and minimum T3g
  - Usage is like NFS
  - Based on Xrootd Posix library and FUSE
  - Bestman, dq2 clients, and Unix tools need it
- **GridFTP for Xrootd** ← WT2 for a while
  - Globus GridFTP + Data Storage Interface (DSI) module for Xrootd/Posix
- **Xrootd Core** ← All Babar needed is this layer
  - redirector and data servers

Storage hardware at SLAC

Bestman Gateway and Xrootd redirector/CNS

- Dual AMD Opteron 244 (single core), 1.8Ghz, 2GB, 1Gb
- DQ2 site service, LFC and GUMS also use this type of machine.

GridFTP requires CPU power and fat network pipe

2 host, each has 2x2 AMD Opteron 2218, 2.6Ghz, 8GB, 3x1Gb

Xrootd data servers on Thumpers

- 2x2 core AMD Opteron, 16GB, 4x1Gb, 48x 0.5-1TB SATA drives
- Solaris and ZFS, ***prevent silent data corruptions.***
- Optimized for reliability and capacity, not for IO ops

- Will be Thors, 2x6 cores, 32GB, 4x1Gb, 48x 1-1.5TB SATA drives

Storage software deployment at SLAC

OSG provides bestman-xrootd as a SE

Including Bestman-Gateway, Xrootd, DSI module, XrootdFS

ATLAS production env. is not installed from OSG

Handle special need from ATLAS
SLAC often runs newer releases before they are pushed to OSG

Have a tiny Bestman-Xrootd installation from OSG

Bestman at SLAC

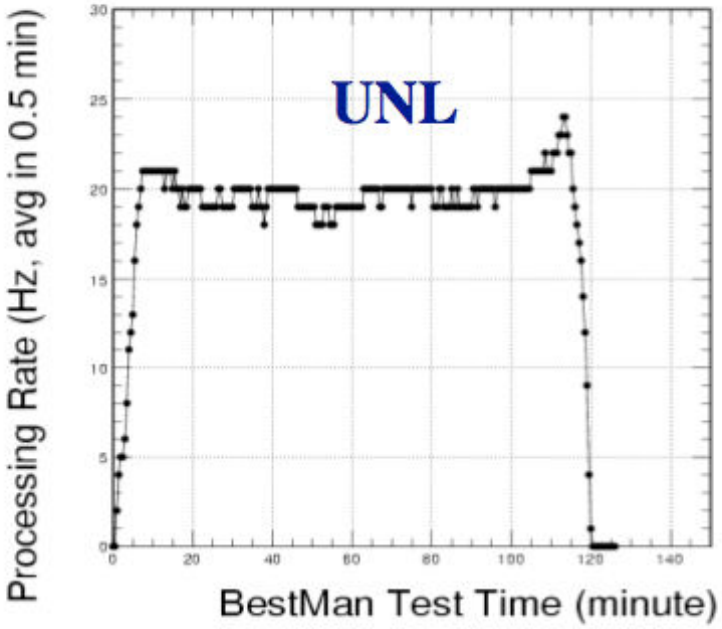
Difference between BeStMan Full mode and BeStMan Gateway mode

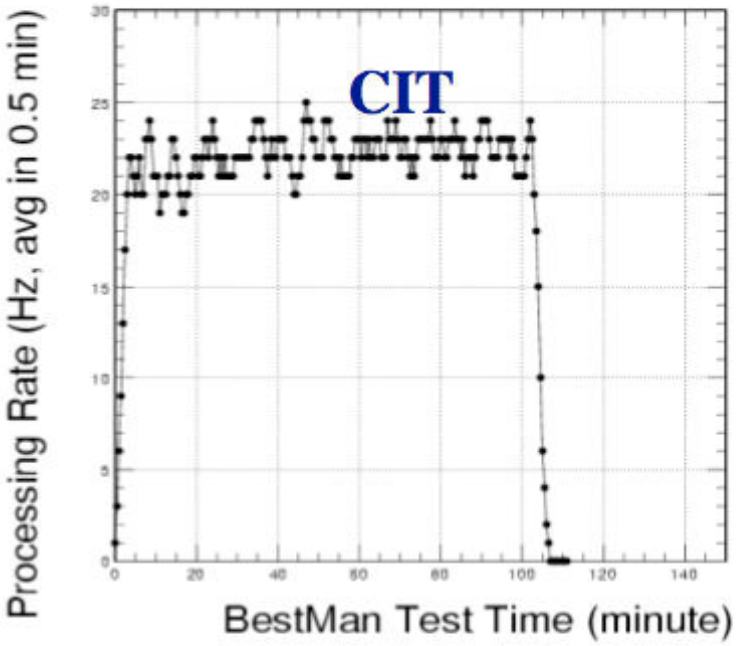
- Full implementation of SRM v2.2
- Support for dynamic space reservation
- Support for request queue management and space management
- Plug-in support for mass storage systems
- Follows the SRM v2.2 specification

- Support for essential subset of SRM v2.2
- Support for pre-defined static space tokens
- Faster performance without queue and space management
- Follows the SRM functionalities needed by ATLAS and CMS

Test on the BeStMan @ UNL and Caltech

effective processing rate of srmls





The effective processing rates of UNL, CIT and UCSD are essentially the same.

Average at 20 Hz, peak at ~25 Hz.

It also shows the latency in running the command doesn't affect the scalability of the BeStMan.

Bestman-Gateway operation at SLAC

Stable! we tuned a few parameters

Java heap size: 1300MB (on a 2GB machine)
Recently increased the # of contains thread from 5 to 25

Failure modes

When Xrootd servers are under stress

- Xrootd stat() call takes too long:
  result in HTTP time out or CONNECT time out

- Redirector can't locate a file, result in file not found
- Panda jobs (not going though SRM interface) will also suffer

Xrootd at SLAC

Accessing Xrootd data by ATLAS

DDM site services use SRM and GridFTP

Calculates ADLER32 checksum at Xrootd data server

Panda production jobs copy Xrootd data to batch nodes

- Expect that data accessing pattern is mostly sequential
- Will process all data in a data file

Panda analysis jobs read directly from Xrootd servers

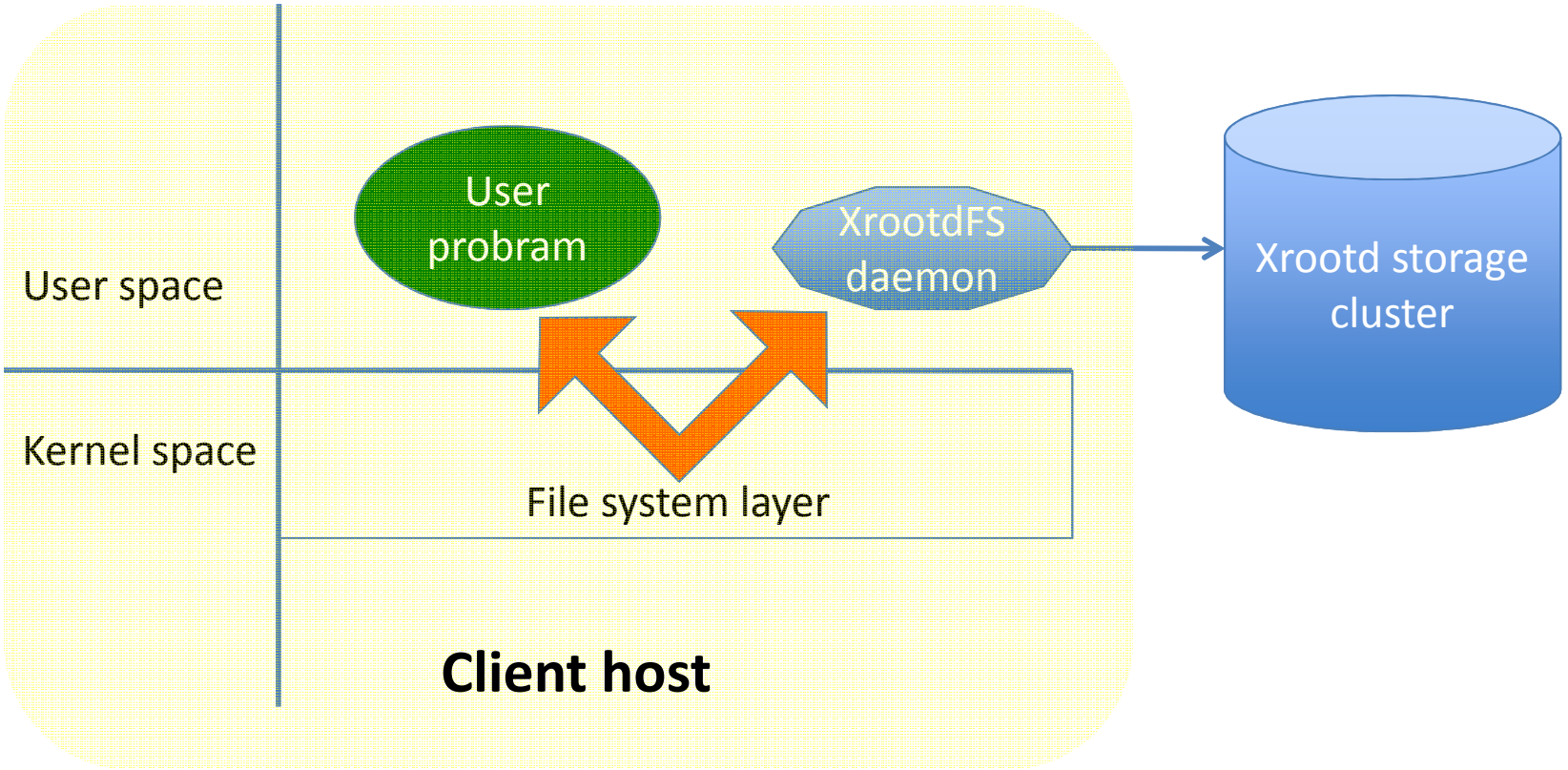
- For ROOT files, non-ROOT files are still copied to batch nodes
- Expect frequent random reads
- May only need a small fraction of data in a file

Interactive users at SLAC accessing data via XrootdFS

- Mounted on ATLAS interactive machines
- Required to run dq2 client tools in and out of Xrootd servers.

XrootdFS

- NFS like data accessing
- Can implement its own read cache
- Relatively expensive compare to direct accessing



Failure modes and Disaster Recovery

Xrootd servers are too busy

- Xrootd server is light weight, normally not the bottleneck
- ***File system and hardware are***
- Redirector can't find files
- Panda jobs, SRM, GridFTP will get "file not found"
- Simple stat() takes too long, FTS complains that SRM connection time out.

Composite Name Space failure

- Will result in SRM failure because the underline XrootdFS will fail
- Can be reconstructed from data servers
- New CNS mechanism is much better

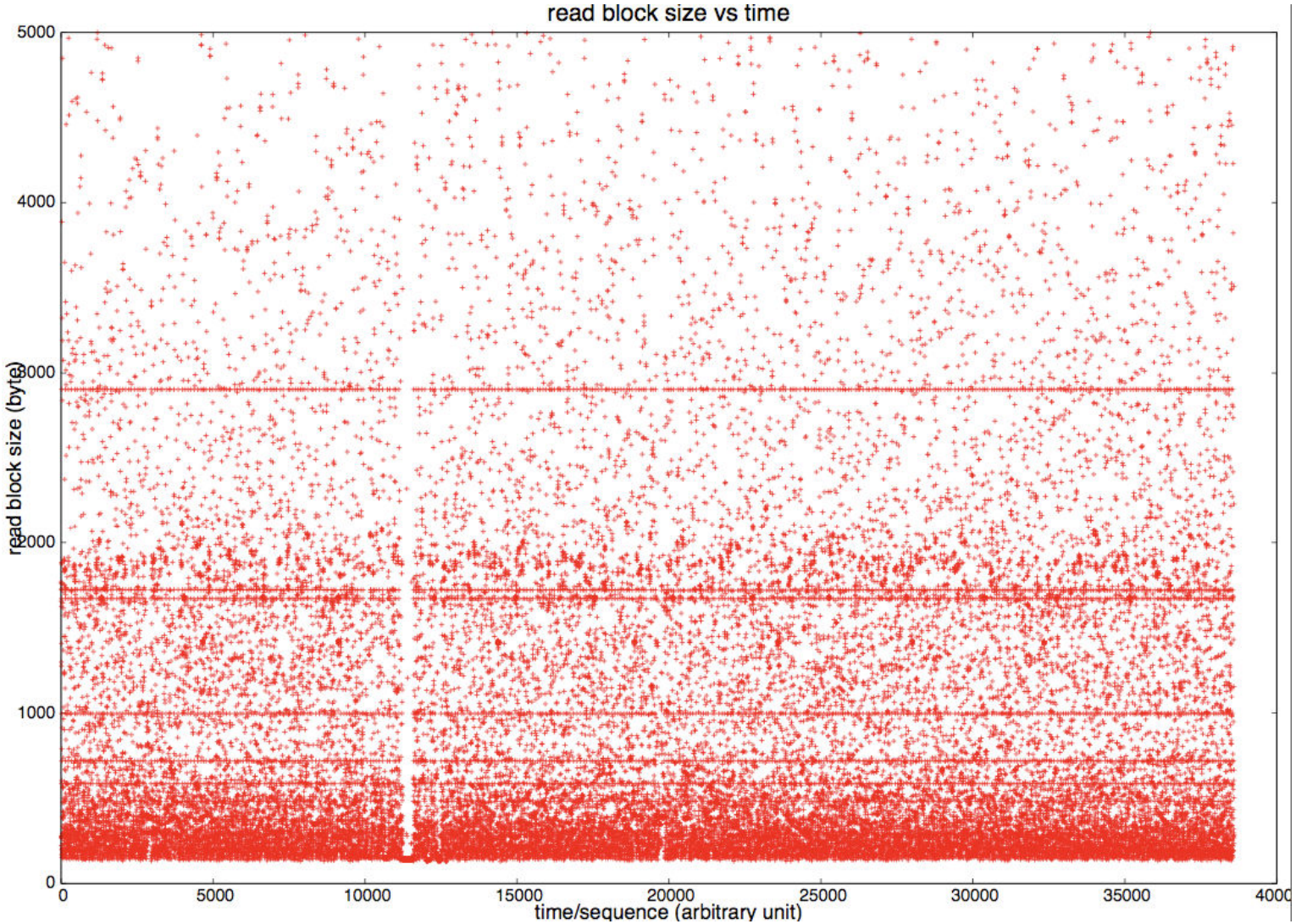
When a data server is down and data is lost

- No tape backup, need to bring data back from other sites
- Added a table to LFC database to record the data server name of each file/guid
- New feature may allow us to continue operate while bringing data back

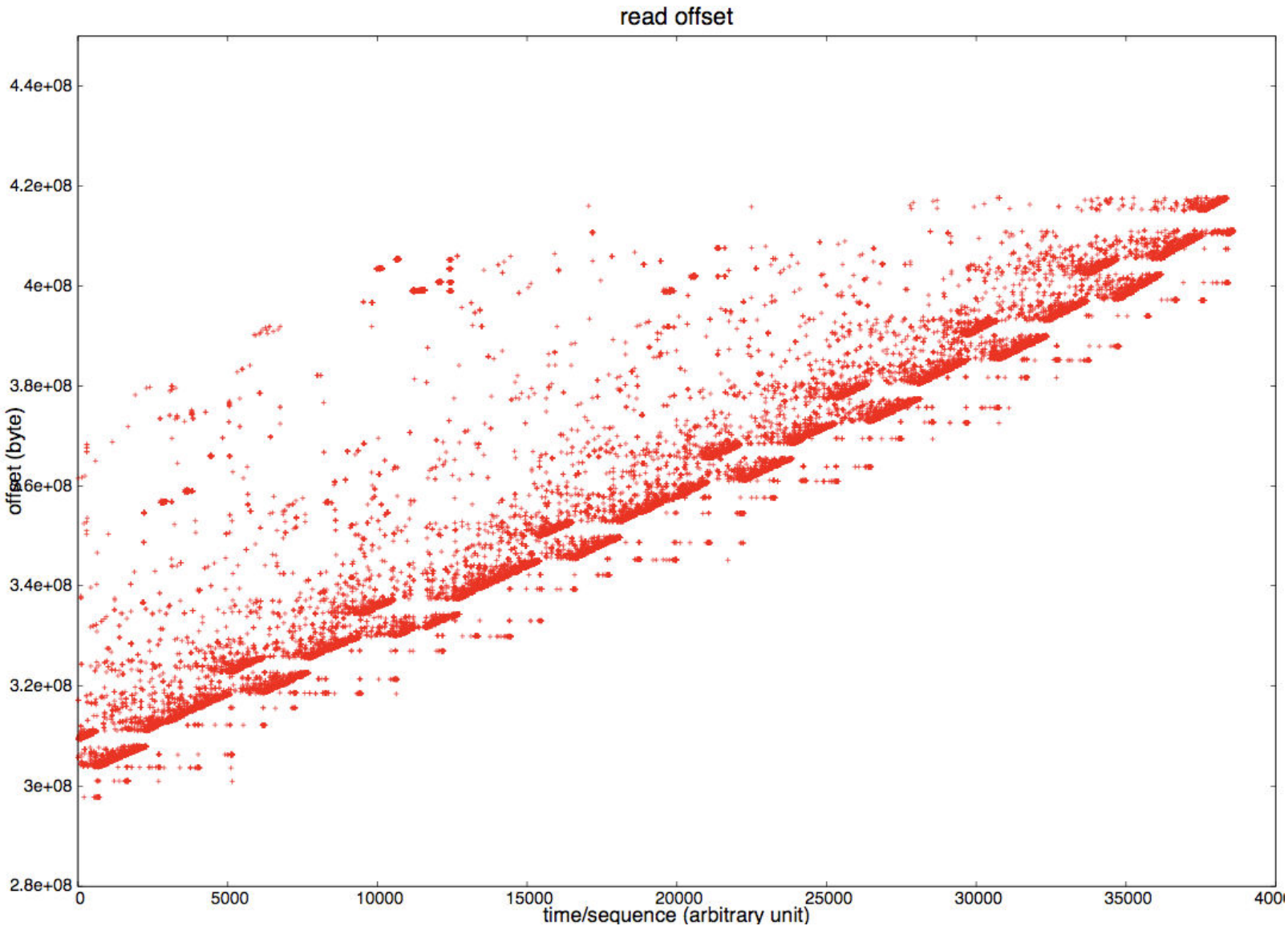
Performance, ATLAS jobs' data reading pattern

Actual read request sent over the wire by a Xrootd client

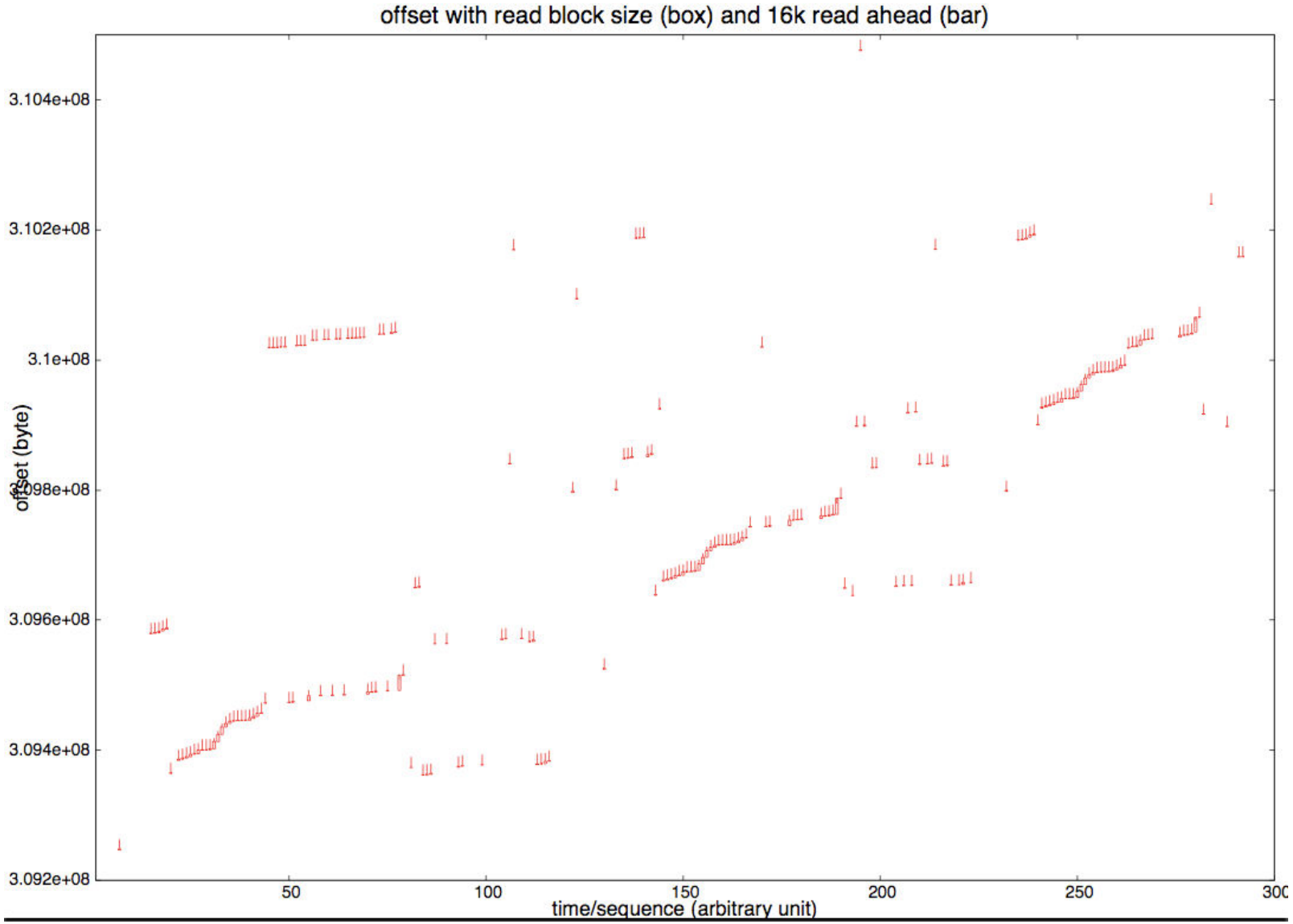
Time	Offset	Len	Offset + Len
1029	1362000529	997	1362001526
1030	1362001526	1671	1362003197
1031	1362328993	164	1362329157
1032	1362329157	726	1362329883
1033	1362329883	163	1362330046
1034	1362875860	165	1362876025
1035	1362876025	288	1362876313
1036	1362876313	175	1362876488
1037	1362876488	282	1362876770
1038	1362003197	2904	1362006101
1039	1362006101	1718	1362007819
1040	1361804637	719	1361805356
1041	1361834590	1478	1361836068
1042	1361836068	1471	1361837539
1043	1361837539	1460	1361838999
1044	1361838999	2667	1361841666
1045	1361841666	530	1361842196
1046	1361842196	3855	1361846051
1047	1361846051	149	1361846200
1048	1361846200	159	1361846359
1049	1361846359	1239	1361847598
1050	1361847598	7565	1361855163
1051	1361855163	7570	1361862733
1052	1361862733	7676	1361870409
1053	1361870409	3911	1361874320
1054	1361874320	2278	1361876598
1055	1361876598	661	1361877259
1056	1361877259	182	1361877441
1057	1361877441	186	1361877627
1058	1361877627	184	1361877811
1059	1361877811	1974	1361879785
1060	1361879785	2267	1361882052



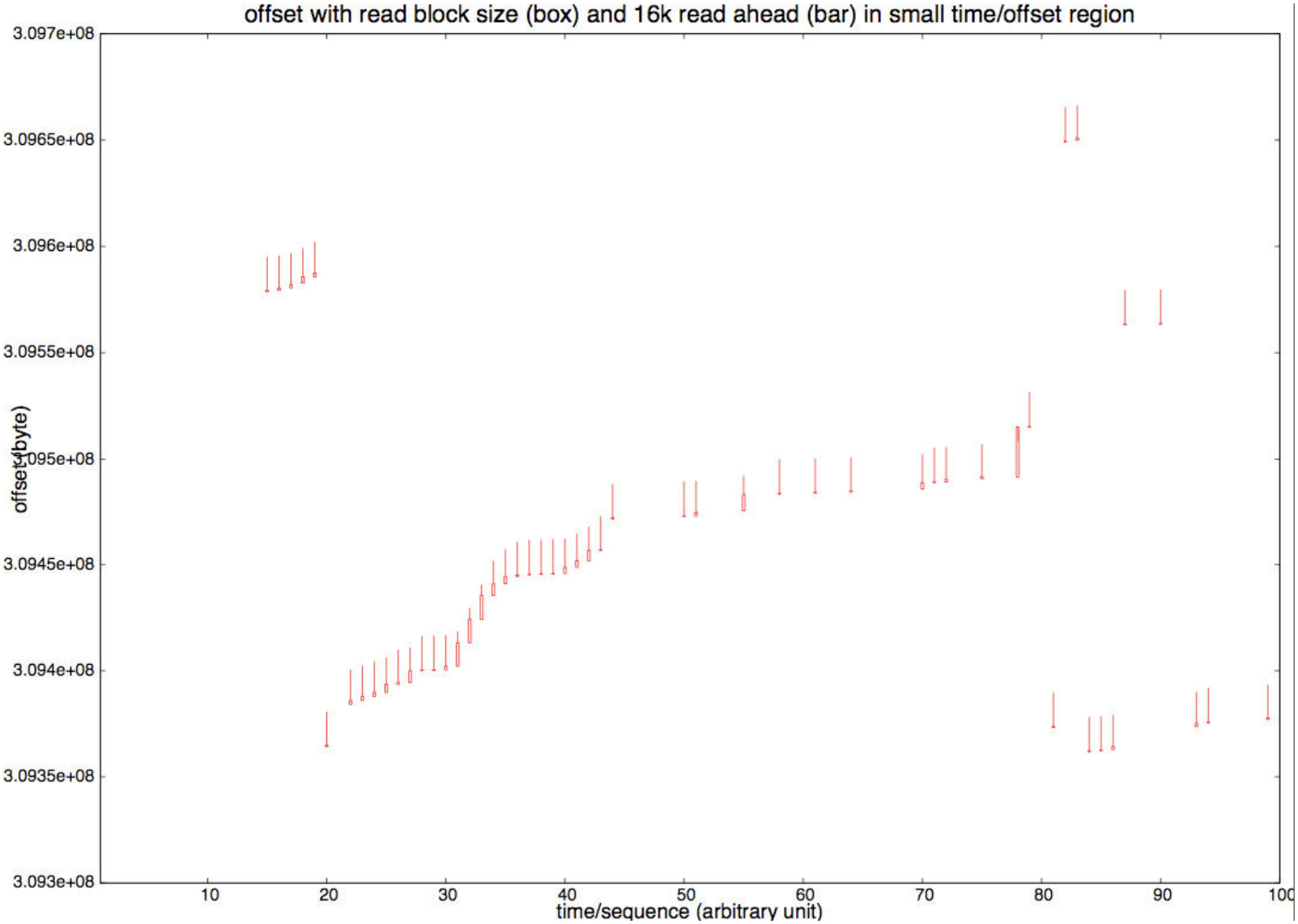
Small block reading dominates in ATLAS jobs



Reading are from head to tail in general, with large jumps



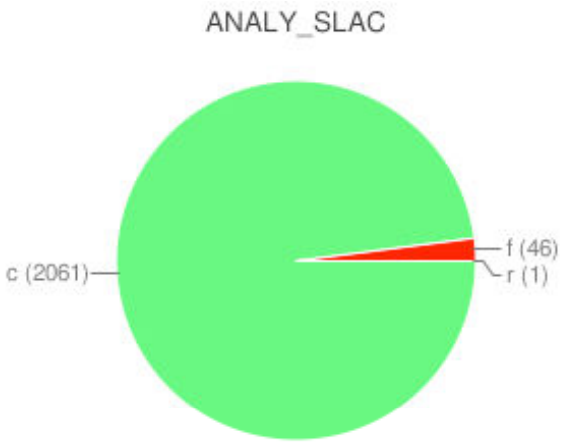
Mixture of sequential and random reads, most are small reads

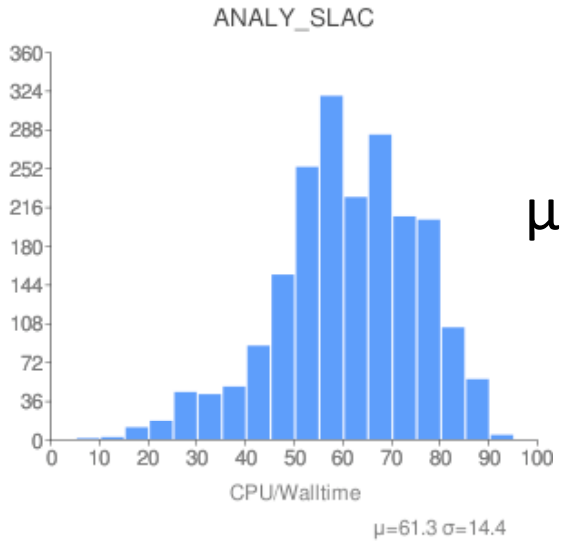


A small (~8K) read ahead may improve the reading performance

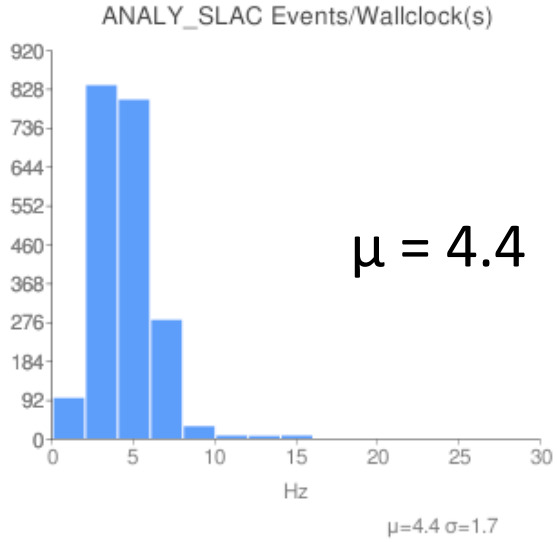
Performance: HammerCloud test (558)

No read ahead: Extra long CPU time, low through put

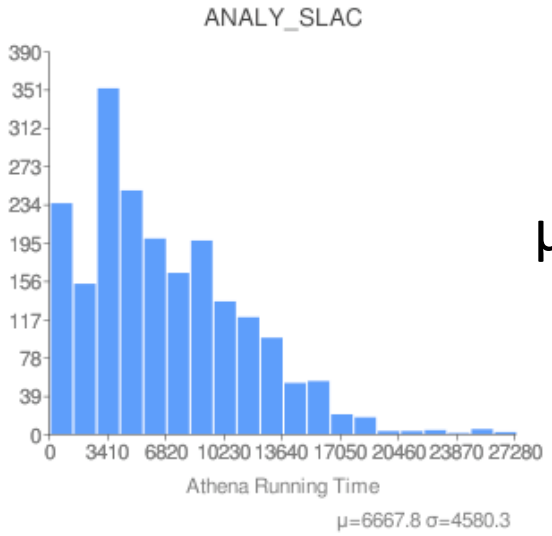




μ = 61.3



μ = 4.4



μ = 6667

Performance, XrdClient

Xrootd client read ahead cache

- Cache size needs to be big enough to host a ROOT file's base tree
- XNet.ReadCacheSize: 60000000 in $HOME/.rootrc

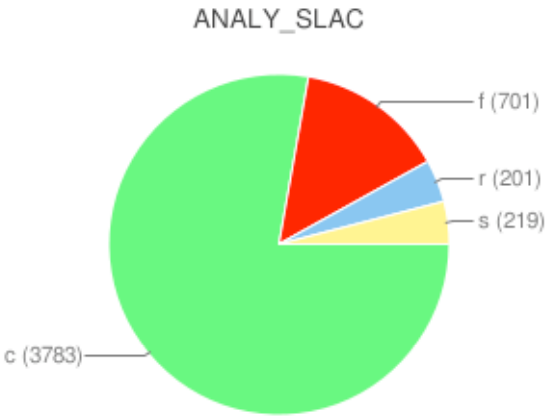
- Complicate read ahead algorithm
- Read ahead size should be big enough to avoid been tuned off
- XNet.ReadAheadSize: 10000000 in $HOME/.rootrc ← **don't follow!**
- Read ahead may cause repeated data transfer. Developer contacted
- Experimenting → Inflate small reads to ~ 8K and turn on/off read ahead

A rare but fatal bug exists in Xrootd client used by all ATLAS releases

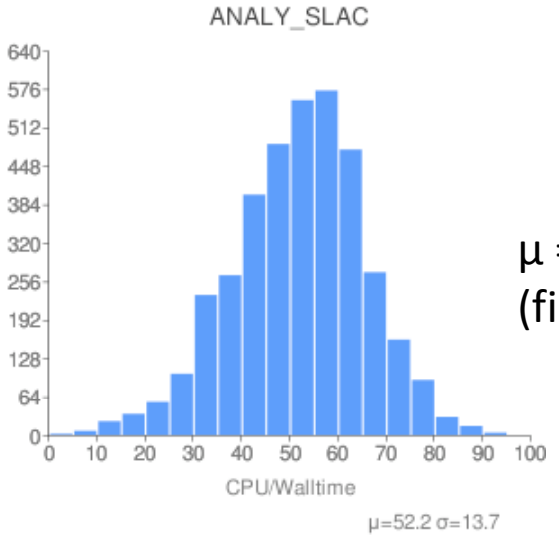
- Problem fixed in the latest Xrootd and ROOT releases
- ATLAS release 14.5.0**+** : use libXrdClient.so from ROOT 5.24
- No solution for pre-14.5.0 releases

Performance, HammerCloud (564)

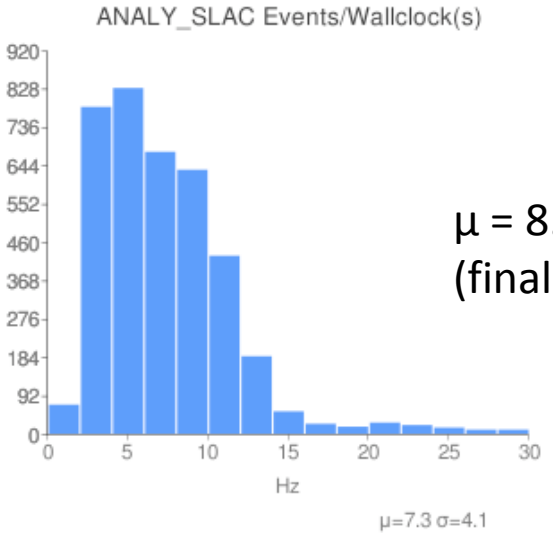
Improved, there may still be room to improve



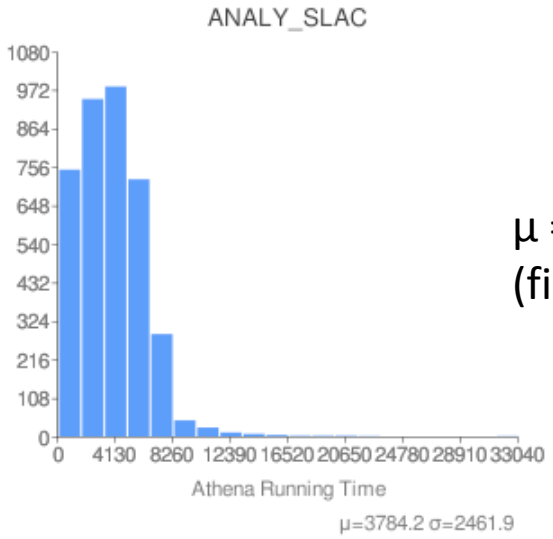
c/f = 6978/766 (final)



μ = 56.6 (final)



μ = 8.4 (final)



μ = 3493 (final)

Wish List

Bestman-G framework allowing plug-in module for storage/file systems

- Bestman-Gateway configuration file is messy, some info are static
- A module developed by storage systems team can better handle the underline FS/SS
- A reference implementation for Posix file system or Ext3 is very useful

Solid State Disk support by Xrootd

- Current storage configuration is optimized for capacity and reliability, at the expense of performance
- SSD as a cache is not forbiddingly expensive
- SSD as a file system cache might not be optimized for HEP data analysis

Other interesting topics

- **Improved CNS, in progress**
- **Use other T1/T2s as Mass Storage System (not the ALICE mode)**
- **Metric to determine if the underline IO system is busy**
- **Monitoring**