

What's in Store?

Tuning grid storage resources for LHC data analysis

Wahid Bhimji
and the GridPP Storage Crew

University of Edinburgh

CHEP
19th October 2010



GridPP
UK Computing for Particle Physics



Outline

- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 Application tuning
 - ROOT I/O
- 3 Filesystem / protocol tuning
 - Rfio buffer sizes and readaheads
- 4 Alternative technologies
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination
- 6 Summary / Future plans

Outline

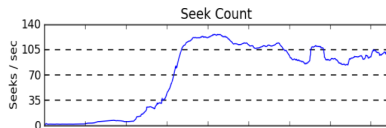
- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 Application tuning
 - ROOT I/O
- 3 Filesystem / protocol tuning
 - Rfio buffer sizes and readaheads
- 4 Alternative technologies
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination
- 6 Summary / Future plans

Problems?

- LHC data analysis is demanding on “Tier 2” storage.
- Large data volumes and “random” file access causes various bottlenecks.
- Regularly see < 50% cpu eff.
- But also can get > 90% eff.
- The difference =

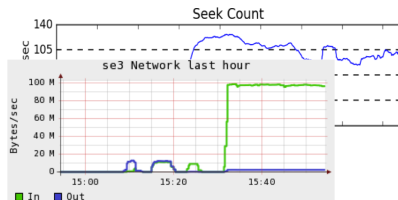
Problems?

- LHC data analysis is demanding on “Tier 2” storage.
- Large data volumes and “random” file access causes various bottlenecks.
- Regularly see < 50% cpu eff.
- But also can get > 90% eff.
- The difference =



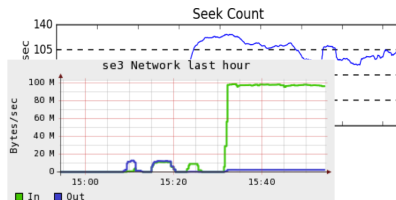
Problems?

- LHC data analysis is demanding on “Tier 2” storage.
- Large data volumes and “random” file access causes various bottlenecks.
- Regularly see < 50% cpu eff.
- But also can get > 90% eff.
- The difference =



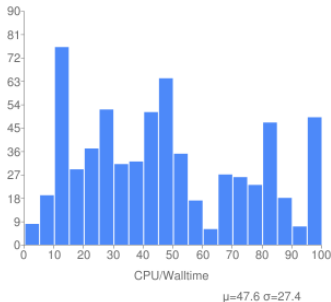
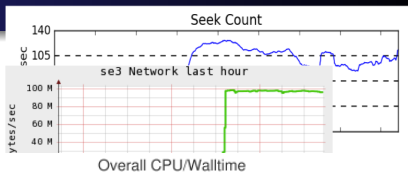
Problems?

- LHC data analysis is demanding on “Tier 2” storage.
- Large data volumes and “random” file access causes various bottlenecks.
- Regularly see < 50% cpu eff.
- But also can get > 90% eff.
- The difference =



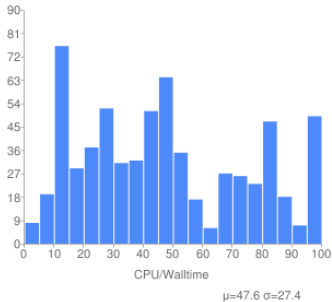
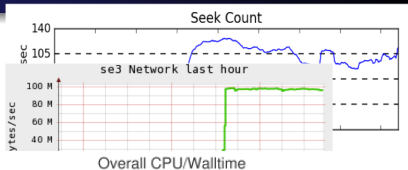
Problems?

- LHC data analysis is demanding on “Tier 2” storage.
- Large data volumes and “random” file access causes various bottlenecks.
- Regularly see < 50% cpu eff.
- But also can get > 90% eff.
- The difference =



Problems?

- LHC data analysis is demanding on “Tier 2” storage.
- Large data volumes and “random” file access causes various bottlenecks.
- Regularly see < 50% cpu eff.
- But also can get > 90% eff.
- The difference =



Solutions?

Many ways to tackle this:

Application

Some things that can be done in ROOT layer centrally by experiment. Others need coordination/ education of many physicists.

Filesystems/Protocols

Many alternatives. Each can require specific tuning.

Hardware

Eg. SSDs - requires more cash
Worth it? see Sam Skipsey's talk

Coordination

Currently a strong interplay between the above - so sites need to be aware of experiment changes and feedback experiences.

Solutions?

Many ways to tackle this:

Application

Some things that can be done in ROOT layer centrally by experiment. Others need coordination/ education of many physicists.

Filesystems/Protocols

Many alternatives. Each can require specific tuning.

Hardware

Eg. SSDs - requires more cash
Worth it? see Sam Skipsey's talk

Coordination

Currently a strong interplay between the above - so sites need to be aware of experiment changes and feedback experiences.

Solutions?

Many ways to tackle this:

Application

Some things that can be done in ROOT layer centrally by experiment. Others need coordination/ education of many physicists.

Filesystems/Protocols

Many alternatives. Each can require specific tuning.

Hardware

Eg. SSDs - requires more cash
Worth it? see Sam Skipsey's talk

Coordination

Currently a strong interplay between the above - so sites need to be aware of experiment changes and feedback experiences.

Solutions?

Many ways to tackle this:

Application

Some things that can be done in ROOT layer centrally by experiment. Others need coordination/ education of many physicists.

Filesystems/Protocols

Many alternatives. Each can require specific tuning.

Hardware

Eg. SSDs - requires more cash
Worth it? see Sam Skipsey's talk

Coordination

Currently a strong interplay between the above - so sites need to be aware of experiment changes and feedback experiences.

Solutions?

Many ways to tackle this:

Application

Some things that can be done in ROOT layer centrally by experiment. Others need coordination/ education of many physicists.

Filesystems/Protocols

Many alternatives. Each can require specific tuning.

Hardware

Eg. SSDs - requires more cash
Worth it? see Sam Skipsey's talk

Coordination

Currently a strong interplay between the above - so sites need to be aware of experiment changes and feedback experiences.

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - File copy: as most ATLAS jobs copy data to WN to run.
 - ROOT reading events: Portable test -TTreePerfStats.
 - Full ATLAS athena jobs "D3PDMaker": Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- "[Hammercloud](#)" for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - **File copy**: as most ATLAS jobs copy data to WN to run.
 - **ROOT reading events**: Portable test -TTreePerfStats.
 - **Full ATLAS athena jobs “D3PDMaker”**: Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- **“Hammercloud”** for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - **File copy**: as most ATLAS jobs copy data to WN to run.
 - **ROOT reading events**: Portable test -TTreePerfStats.
 - **Full ATLAS athena jobs "D3PDMaker"**: Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- **"Hammercloud"** for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - **File copy**: as most ATLAS jobs copy data to WN to run.
 - **ROOT reading events**: Portable test -TTreePerfStats.
 - **Full ATLAS athena jobs "D3PDMaker"**: Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- **"Hammercloud"** for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - **File copy**: as most ATLAS jobs copy data to WN to run.
 - **ROOT reading events**: Portable test -TTreePerfStats.
 - **Full ATLAS athena jobs “D3PDMaker”**: Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- **“Hammercloud”** for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - **File copy**: as most ATLAS jobs copy data to WN to run.
 - **ROOT reading events**: Portable test -TTreePerfStats.
 - **Full ATLAS athena jobs “D3PDMaker”**: Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- **“Hammercloud”** for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Tests performed

- Real ATLAS data (mostly merged AODs \approx 2-3GB /file)
- Simple framework developed for performing range of tests. Eg:
 - **File copy**: as most ATLAS jobs copy data to WN to run.
 - **ROOT reading events**: Portable test -TTreePerfStats.
 - **Full ATLAS athena jobs “D3PDMaker”**: Commonly used - writes data out in different format, not much analysis, I/O Heavy but also largely event sequential.
- **“Hammercloud”** for stress tests - initially developed on ATLAS; automates repeated submission of jobs and collects statistics.
- And ... some really real site experiences!

Test sites

Production T2s and Testbeds (See paper for full details)

Glasgow: Large T2 Site. (\approx 2000 Cores and \approx 1PB Disk)
DPM for Production, DPM-Fuse testbed

Edinburgh: Smaller T2.
DPM, StoRM-GPFS for Production, and testbeds for HDFS, Ceph

QMUL: Large T2 Site. StoRM-Lustre

Test sites

Production T2s and Testbeds (See paper for full details)

Glasgow: Large T2 Site. (≈ 2000 Cores and ≈ 1 PB Disk)
DPM for Production, DPM-Fuse testbed

Edinburgh: Smaller T2.
DPM, StoRM-GPFS for Production, and testbeds for HDFS, Ceph

QMUL: Large T2 Site. StoRM-Lustre

- Using production sites: Advantage - already tuned, realistic environment. Disadvantage - contending with other work.
- Tests run several times - same hardware where possible.
- Event rates / efficiencies not precise - but some *big* differences.
- Too many permutations to list all results here.

Test sites

Production T2s and Testbeds (See paper for full details)

Glasgow: Large T2 Site. (≈ 2000 Cores and ≈ 1 PB Disk)
DPM for Production, DPM-Fuse testbed

Edinburgh: Smaller T2.
DPM, StoRM-GPFS for Production, and testbeds for HDFS, Ceph

QMUL: Large T2 Site. StoRM-Lustre

- Using production sites: Advantage - already tuned, realistic environment. Disadvantage - contending with other work.
- Tests run several times - same hardware where possible.
- Event rates / efficiencies not precise - but some *big* differences.
- Too many permutations to list all results here.

Test sites

Production T2s and Testbeds (See paper for full details)

Glasgow: Large T2 Site. (≈ 2000 Cores and ≈ 1 PB Disk)
DPM for Production, DPM-Fuse testbed

Edinburgh: Smaller T2.
DPM, StoRM-GPFS for Production, and testbeds for HDFS, Ceph

QMUL: Large T2 Site. StoRM-Lustre

- Using production sites: Advantage - already tuned, realistic environment. Disadvantage - contending with other work.
- Tests run several times - same hardware where possible.
- Event rates / efficiencies not precise - but some *big* differences.
- Too many permutations to list all results here.

Outline

- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 **Application tuning**
 - **ROOT I/O**
- 3 Filesystem / protocol tuning
 - Rfio buffer sizes and readaheads
- 4 Alternative technologies
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination
- 6 Summary / Future plans

Application tuning - ROOT I/O

- Several recent improvements in ATLAS use of native ROOT features - e.g. Basket ordering and TTree cache.
- See talk by Ilija Vukotic (at same time as this(!)) for more details.
- Both significantly reduce random-like file access and number of requests to disk, so improving performance.
- Further changes on the way (soon!) in ATLAS when it uses ROOT 5.26 with “Optimise baskets” and “autoflush”.
- Changes such as these require retuning: e.g. read ahead buffers, choices for filesystems or hardware purchases.

Application tuning - ROOT I/O

- Several recent improvements in ATLAS use of native ROOT features - e.g. Basket ordering and TTree cache.
- See talk by Ilija Vukotic (at same time as this(!)) for more details.
- Both significantly reduce random-like file access and number of requests to disk, so improving performance.
- Further changes on the way (soon!) in ATLAS when it uses ROOT 5.26 with “Optimise baskets” and “autoflush”.
- Changes such as these require retuning: e.g. read ahead buffers, choices for filesystems or hardware purchases.

Application tuning - ROOT I/O

- Several recent improvements in ATLAS use of native ROOT features - e.g. Basket ordering and TTree cache.
- See talk by Ilija Vukotic (at same time as this(!)) for more details.
- Both significantly reduce random-like file access and number of requests to disk, so improving performance.
- Further changes on the way (soon!) in ATLAS when it uses ROOT 5.26 with “Optimise baskets” and “autoflush”.
- Changes such as these require retuning: e.g. read ahead buffers, choices for filesystems or hardware purchases.

Application tuning - ROOT I/O

- Several recent improvements in ATLAS use of native ROOT features - e.g. Basket ordering and TTree cache.
- See talk by Ilija Vukotic (at same time as this(!)) for more details.
- Both significantly reduce random-like file access and number of requests to disk, so improving performance.
- Further changes on the way (soon!) in ATLAS when it uses ROOT 5.26 with “Optimise baskets” and “autoflush”.
- Changes such as these require retuning: e.g. read ahead buffers, choices for filesystems or hardware purchases.

Application tuning - ROOT I/O

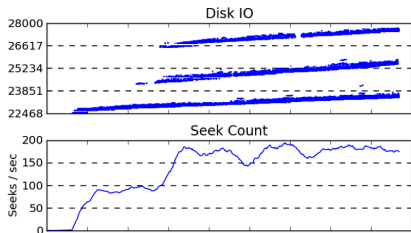
- Several recent improvements in ATLAS use of native ROOT features - e.g. Basket ordering and TTree cache.
- See talk by Ilija Vukotic (at same time as this(!)) for more details.
- Both significantly reduce random-like file access and number of requests to disk, so improving performance.
- Further changes on the way (soon!) in ATLAS when it uses ROOT 5.26 with “Optimise baskets” and “autoflush”.
- Changes such as these require retuning: e.g. read ahead buffers, choices for filesystems or hardware purchases.

Basket ordering - effects for local jobs

- Six ATLAS athena D3PDMaker jobs running on different 2GB files accessing one disk partition.
- Unordered files show large scatter in reads and hit seek limits.
- Reading new files is significantly more linear and so seek counts reduced.

Basket ordering - effects for local jobs

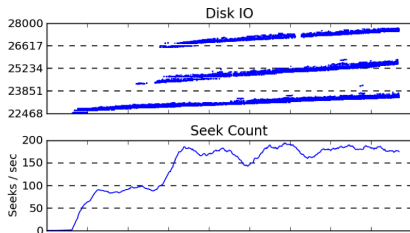
- Six ATLAS athena D3PDMaker jobs running on different 2GB files accessing one disk partition.
- Unordered files show large scatter in reads and hit seek limits.
- Reading new files is significantly more linear and so seek counts reduced.



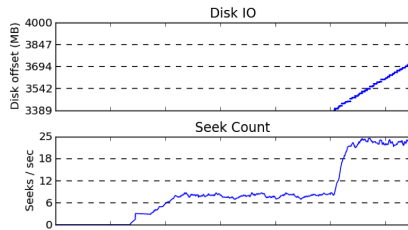
Unordered ATLAS AOD files

Basket ordering - effects for local jobs

- Six ATLAS athena D3PDMaker jobs running on different 2GB files accessing one disk partition.
- Unordered files show large scatter in reads and hit seek limits.
- Reading new files is significantly more linear and so seek counts reduced.



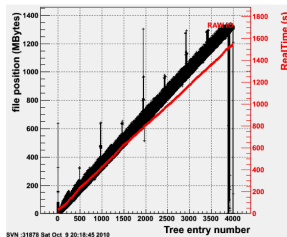
Unordered ATLAS AOD files



Reordered AOD files.

Basket ordering - remote reading example

- ROOT test on these AODs, output from TTreePerfStats.
- GPFS running on same site as DPM.
- Ordering makes a much bigger impact.



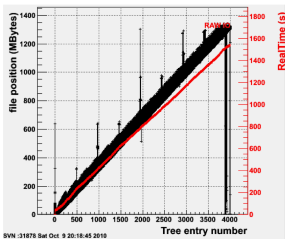
Unordered - DPM (Rfio)

Disk Time \approx 1500s

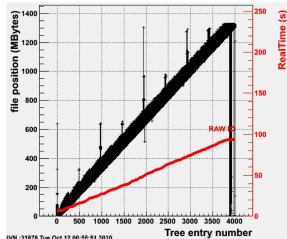
Wall Clock \approx 1700s

Basket ordering - remote reading example

- ROOT test on these AODs, output from TTreePerfStats.
- GPFS running on same site as DPM.
- Ordering makes a much bigger impact.



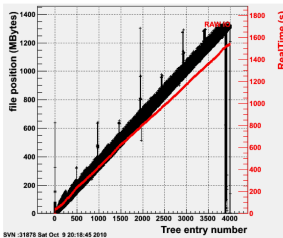
Unordered - DPM (Rfio)
Disk Time \approx 1500s
Wall Clock \approx 1700s



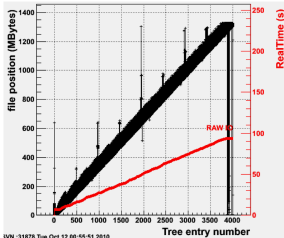
Unordered - GPFS
Disk time \approx 100s
Wall Clock \approx 230s

Basket ordering - remote reading example

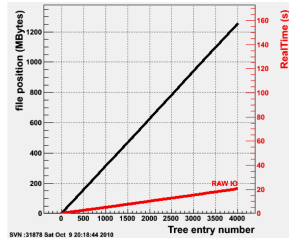
- ROOT test on these AODs, output from TTreePerfStats.
- GPFS running on same site as DPM.
- Ordering makes a a much bigger impact.



Unordered - DPM (Rfio)
Disk Time \approx 1500s
Wall Clock \approx 1700s



Unordered - GPFS
Disk time \approx 100s
Wall Clock \approx 230s



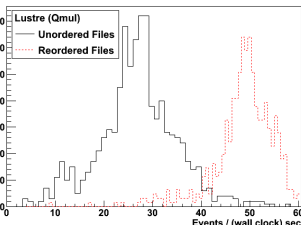
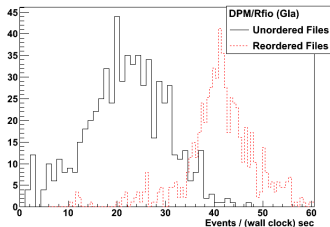
Ordered - DPM (Rfio)
Disk time \approx 20s
Wall Clock \approx 160s

Basket ordering - Rfio / Lustre

- Athena on unordered and reordered files.
- Stress (hammercloud) test (filtered results to select similar conditions: CPU, no. running jobs etc.)
- Compared at time of change when sites had plenty of both types of file.
- Do not compare event rates with other Athena tests here - its a different analysis.

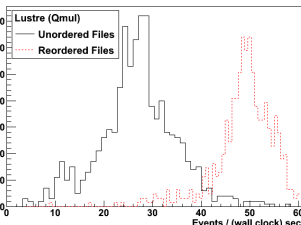
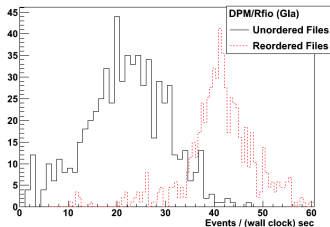
Basket ordering - Rfio / Lustre

- Athena on unordered and reordered files.
- Stress (hammercloud) test (filtered results to select similar conditions: CPU, no. running jobs etc.)
- Compared at time of change when sites had plenty of both types of file.
- Do not compare event rates with other Athena tests here - its a different analysis.



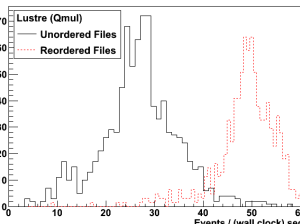
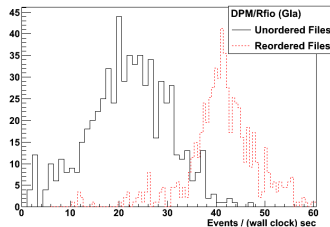
Basket ordering - Rfio / Lustre

- Athena on unordered and reordered files.
- Stress (hammercloud) test (filtered results to select similar conditions: CPU, no. running jobs etc.)
- Compared at time of change when sites had plenty of both types of file.
- Do not compare event rates with other Athena tests here - its a different analysis.



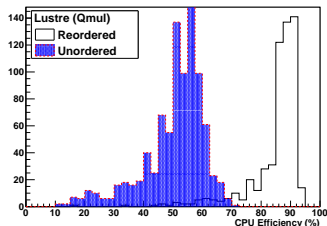
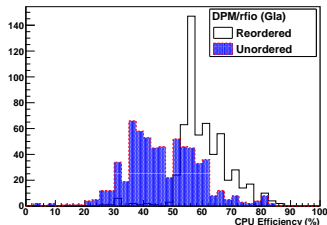
Basket ordering - Rfio / Lustre

- Athena on unordered and reordered files.
- Stress (hammercloud) test (filtered results to select similar conditions: CPU, no. running jobs etc.)
- Compared at time of change when sites had plenty of both types of file.
- Do not compare event rates with other Athena tests here - its a different analysis.



Basket ordering - Rfio / Lustre

- Athena on unordered and reordered files.
- Stress (hammercloud) test (filtered results to select similar conditions: CPU, no. running jobs etc.)
- Compared at time of change when sites had plenty of both types of file.
- Do not compare event rates with other Athena tests here - its a different analysis.

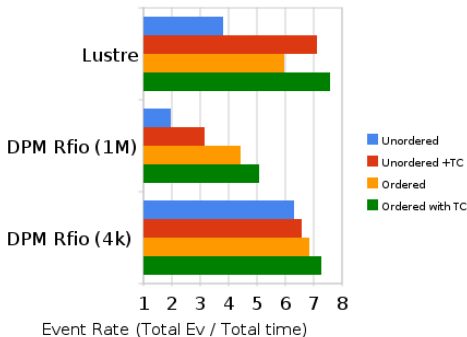


TTreeCache

- Using TTreeCache with DPM for files > 2GB required some fixes to DPM (in since 1.7.4-7) and ROOT (backported to 5.26c used in Atlas Releases > 15.9.0).
- ATLAS D3PD, single job on same file each time (merged AOD - real data).

TTreeCache

- Using TTreeCache with DPM for files > 2GB required some fixes to DPM (in since 1.7.4-7) and ROOT (backported to 5.26c used in Atlas Releases > 15.9.0).
- ATLAS D3PD, single job on same file each time (merged AOD - real data).



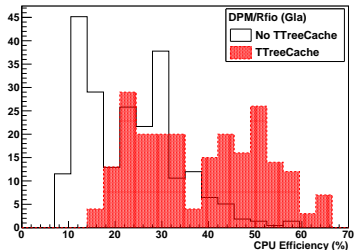
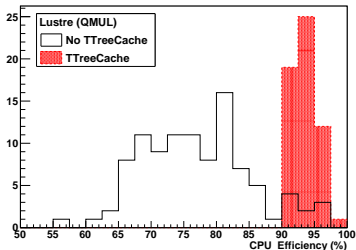
Further improvements from TTreeCache even after the reordering.

TTreeCache

- Running on the whole dataset.
- Around 300 jobs all run simultaneously

TTreeCache

- Running on the whole dataset.
- Around 300 jobs all run simultaneously



Outline

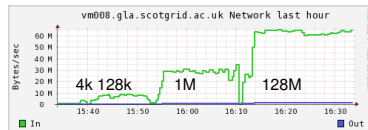
- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 Application tuning
 - ROOT I/O
- 3 Filesystem / protocol tuning**
 - Rfio buffer sizes and readaheads**
- 4 Alternative technologies
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination
- 6 Summary / Future plans

Rfio buffer sizes

- Set in `/etc/shift.conf` or (in most recent DPM versions) with `RFIO_IOBUFSIZE` variable.
- Data access still not sequential enough for buffered data to be effectively used.
- So for direct rfio access - smaller buffers *still* give better efficiencies for single atlas analysis jobs.
- True even with TTreeCache on for main collection Tree.

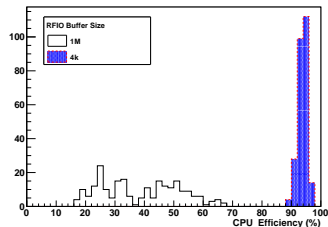
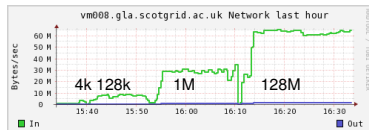
Rfio buffer sizes

- Set in `/etc/shift.conf` or (in most recent DPM versions) with `RFIO_IOBUFSIZE` variable.
- Data access still not sequential enough for buffered data to be effectively used.
- So for direct rfio access - smaller buffers *still* give better efficiencies for single atlas analysis jobs.
- True even with TTreeCache on for main collection Tree.



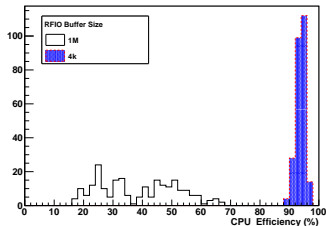
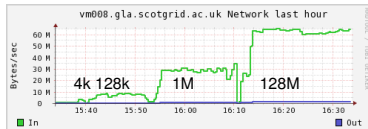
Rfio buffer sizes

- Set in `/etc/shift.conf` or (in most recent DPM versions) with `RFIO_IOBUFSIZE` variable.
- Data access still not sequential enough for buffered data to be effectively used.
- So for direct rfio access - smaller buffers *still* give better efficiencies for single atlas analysis jobs.
- True even with TTreeCache on for main collection Tree.



Rfio buffer sizes

- Set in `/etc/shift.conf` or (in most recent DPM versions) with `RFIO_IOBUFSIZE` variable.
- Data access still not sequential enough for buffered data to be effectively used.
- So for direct rfio access - smaller buffers *still* give better efficiencies for single atlas analysis jobs.
- True even with TTreeCache on for main collection Tree.



But smaller buffers **not used** because....

Rfio buffer sizes - copy to WN

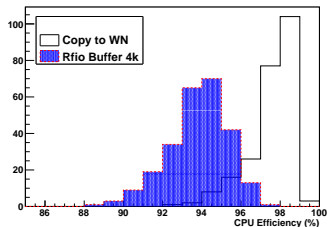
- Smaller buffers lead to more IOPS.
- Causes considerable load on Tier 2 disk servers.
- Reduces efficiency for all users.
- So for most UK DPM T2 sites still best for ATLAS jobs to copy file to WN before running - can use large buffers and keep load down with high CPU efficiencies
- Needs to be tuned for site - 0.5M is common in the UK.

Rfio buffer sizes - copy to WN

- Smaller buffers lead to more IOPS.
- Causes considerable load on Tier 2 disk servers.
- Reduces efficiency for all users.
- So for most UK DPM T2 sites still best for ATLAS jobs to copy file to WN before running - can use large buffers and keep load down with high CPU efficiencies
- Needs to be tuned for site - 0.5M is common in the UK.

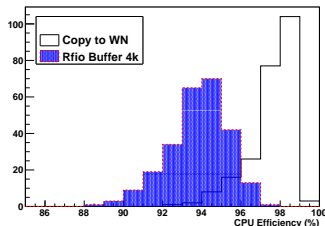
Rfio buffer sizes - copy to WN

- Smaller buffers lead to more IOPS.
- Causes considerable load on Tier 2 disk servers.
- Reduces efficiency for all users.
- So for most UK DPM T2 sites still best for ATLAS jobs to copy file to WN before running - can use large buffers and keep load down with high CPU efficiencies
- Needs to be tuned for site - 0.5M is common in the UK.



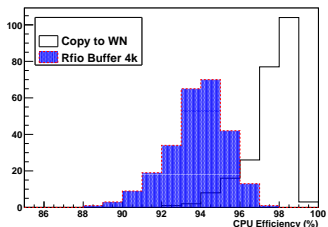
Rfio buffer sizes - copy to WN

- Smaller buffers lead to more IOPS.
- Causes considerable load on Tier 2 disk servers.
- Reduces efficiency for all users.
- So for most UK DPM T2 sites still best for ATLAS jobs to copy file to WN before running - can use large buffers and keep load down with high CPU efficiencies
- Needs to be tuned for site - 0.5M is common in the UK.



Rfio buffer sizes - copy to WN

- Smaller buffers lead to more IOPS.
- Causes considerable load on Tier 2 disk servers.
- Reduces efficiency for all users.
- So for most UK DPM T2 sites still best for ATLAS jobs to copy file to WN before running - can use large buffers and keep load down with high CPU efficiencies
- Needs to be tuned for site - 0.5M is common in the UK.



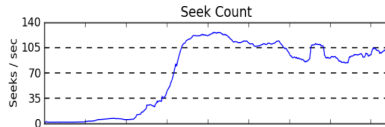
But copy to WN depends on
WN disk ...

Copy to WN

- Analysis on Worker Node - can move bottleneck.
- Efficiencies can deteriorate if node saturated with analysis jobs and only a single SATA drive available.
- Options to maintain efficiency include SSDs or Raid 0 pair of disks.
- See Sam Skipsey's talk for an evaluation.

Copy to WN

- Analysis on Worker Node - can move bottleneck.
- Efficiencies can deteriorate if node saturated with analysis jobs and only a single SATA drive available.
- Options to maintain efficiency include SSDs or Raid 0 pair of disks.
- See Sam Skipsey's talk for an evaluation.

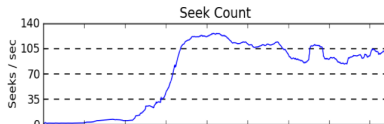


Fill (jobs / 8 core node)	Cpu Eff.
1	90%
2	85%
8	70%

Copy to WN

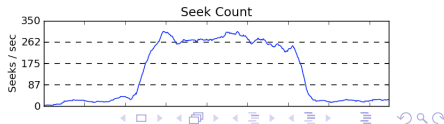
- Analysis on Worker Node - can move bottleneck.
- Efficiencies can deteriorate if node saturated with analysis jobs and only a single SATA drive available.
- Options to maintain efficiency include SSDs or Raid 0 pair of disks.
- See Sam Skipsey's talk for an evaluation.

SATA Disk:



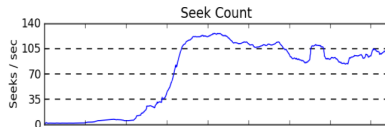
Fill (jobs / 8 core node)	Cpu Eff.
1	90%
2	85%
8	70%

SSD:



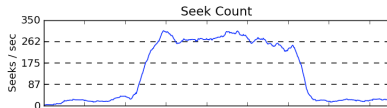
Copy to WN

- Analysis on Worker Node - can move bottleneck.
- Efficiencies can deteriorate if node saturated with analysis jobs and only a single SATA drive available.
- Options to maintain efficiency include SSDs or Raid 0 pair of disks.
- See Sam Skipsey's talk for an evaluation.



Fill (jobs / 8 core node)	Cpu Eff.
1	90%
2	85%
8	70%

SSD:



Blkdev read aheads

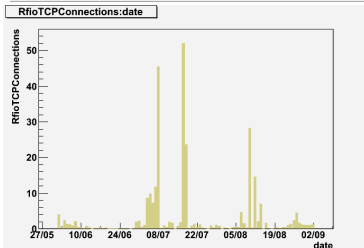
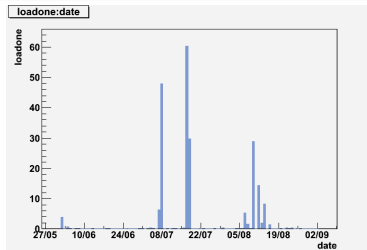
- Even with larger rfio buffers, many UK sites seeing high loads on storage servers.
- Particularly after move to SL5 with xfs filesystems - reason not understood.
- Can be mitigated by moving to ext4.
- Or by setting larger read aheads in the block device - e.g Liverpool set 8MB from 64kB, noticed immediate drop in load spikes.

Blkdev read aheads

- Even with larger rfio buffers, many UK sites seeing high loads on storage servers.
- Particularly after move to SL5 with xfs filesystems - reason not understood.
- Can be mitigated by moving to ext4.
- Or by setting larger read aheads in the block device - e.g Liverpool set 8MB from 64kB, noticed immediate drop in load spikes.

Blkdev read aheads

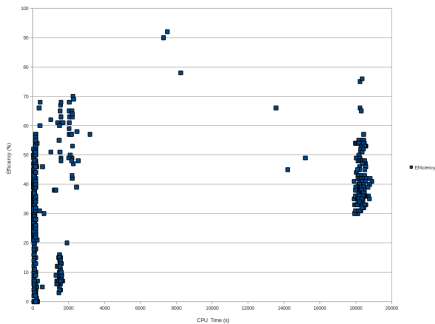
- Even with larger rfio buffers, many UK sites seeing high loads on storage servers.
- Particularly after move to SL5 with xfs filesystems - reason not understood.
- Can be mitigated by moving to ext4.
- Or by setting larger read aheads in the block device - e.g Liverpool set 8MB from 64kB, noticed immediate drop in load spikes.



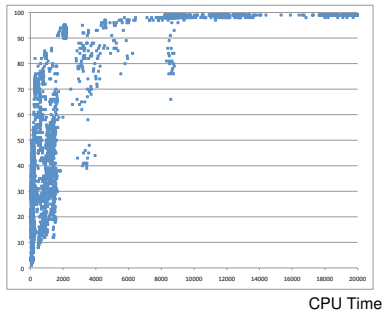
Blkdev read aheads

Manchester Cpu Efficiency Vs CPU Time.

Before readahead tuning



After



Outline

- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 Application tuning
 - ROOT I/O
- 3 Filesystem / protocol tuning
 - Rfio buffer sizes and readaheads
- 4 **Alternative technologies**
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination
- 6 Summary / Future plans

Fuse with DPM

- Users like POSIX access.
- NFS v4.1 in DPM soon (see Ricardo Rocha's talk)
- But in the meantime - reviving a GFAL FUSE module written by J-P Baud some years ago.
- RFIO is still the backend, but can benefit from page cache
- Cp performance here poor but Athena performance at least as good as normal rfio
-need whole site scale/
stability test.

Fuse with DPM

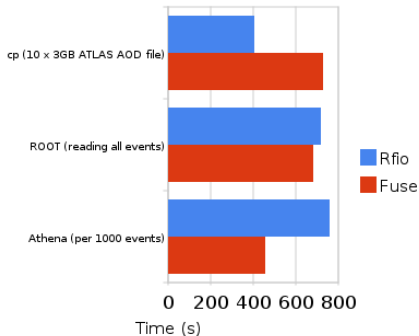
- Users like POSIX access.
- NFS v4.1 in DPM soon (see Ricardo Rocha's talk)
- But in the meantime - reviving a GFAL FUSE module written by J-P Baud some years ago.
- RFIO is still the backend, but can benefit from page cache
- Cp performance here poor but Athena performance at least as good as normal rfio
-need whole site scale/
stability test.

Fuse with DPM

- Users like POSIX access.
- NFS v4.1 in DPM soon (see Ricardo Rocha's talk)
- But in the meantime - reviving a GFAL FUSE module written by J-P Baud some years ago.
- RFIO is still the backend, but can benefit from page cache
- Cp performance here poor but Athena performance at least as good as normal rfio
-need whole site scale/
stability test.

Fuse with DPM

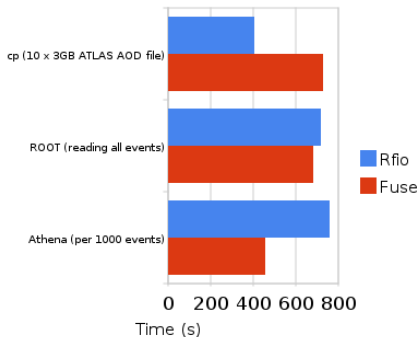
- Users like POSIX access.
- NFS v4.1 in DPM soon (see Ricardo Rocha's talk)
- But in the meantime - reviving a GFAL FUSE module written by J-P Baud some years ago.
- RFIO is still the backend, but can benefit from page cache
- Cp performance here poor but Athena performance at least as good as normal rfio -need whole site scale/stability test.



Issue in using for some Atlas jobs as file is opened twice (resolved for this test by opening via rfio for configuration).

Fuse with DPM

- Users like POSIX access.
- NFS v4.1 in DPM soon (see Ricardo Rocha's talk)
- But in the meantime - reviving a GFAL FUSE module written by J-P Baud some years ago.
- RFIO is still the backend, but can benefit from page cache
- Cp performance here poor but Athena performance at least as good as normal rfio -need whole site scale/ stability test.



Issue in using for some Atlas jobs as file is opened twice (resolved for this test by opening via rfio for configuration).

HDFS and Ceph

Hadoop HDFS:

- Widely used in industry - Facebook, HP , etc. etc.
- Particularly used to aggregate Worker Node disk
- Being used in production for US CMS T2s

HDFS and Ceph

Hadoop HDFS:

- Widely used in industry - Facebook, HP , etc. etc.
- Particularly used to aggregate Worker Node disk
- Being used in production for US CMS T2s

Ceph:

- Not yet thought production ready
- Client in linux kernel since v2.6.34
- Tested both kernel client and FUSE in SL5.

HDFS and Ceph

Hadoop HDFS:

- Widely used in industry - Facebook, HP , etc. etc.
- Particularly used to aggregate Worker Node disk
- Being used in production for US CMS T2s

Ceph:

- Not yet thought production ready
- Client in linux kernel since v2.6.34
- Tested both kernel client and FUSE in SL5.

Testbeds: (by Efstathia Mouzeli see [link](#) for details)

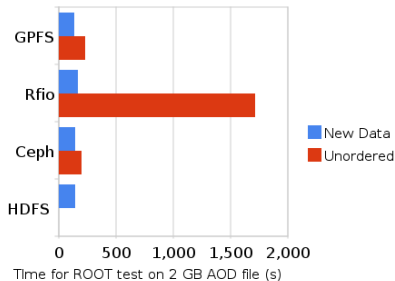
- Using same 5 worker nodes at Edinburgh.
- Both with 2 block level replicas.
- StoRM SRM frontend but basic/hacked as no full POSIX acls.

HDFS and Ceph

- Both HDFS/ Ceph easily configured and performant (comparable to site's DPM and GPFS).
- Eg. for ROOT jobs reading all events in ATLAS AOD files.
- Ceph like GPFS seems to handle well random access of "old" ATLAS AOD files.
- HDFS test maxed out network on unordered file - but maybe some tuning could help.

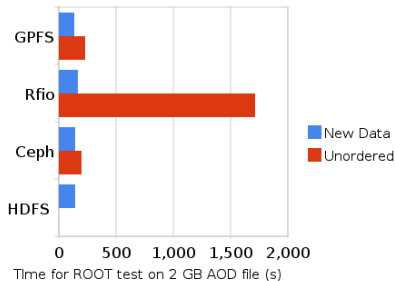
HDFS and Ceph

- Both HDFS/ Ceph easily configured and performant (comparable to site's DPM and GPFS).
- Eg. for ROOT jobs reading all events in ATLAS AOD files.
- Ceph like GPFS seems to handle well random access of "old" ATLAS AOD files.
- HDFS test maxed out network on unordered file - but maybe some tuning could help.



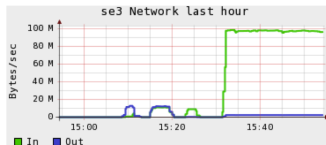
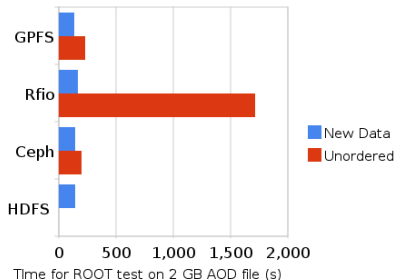
HDFS and Ceph

- Both HDFS/ Ceph easily configured and performant (comparable to site's DPM and GPFS).
- Eg. for ROOT jobs reading all events in ATLAS AOD files.
- Ceph like GPFS seems to handle well random access of "old" ATLAS AOD files.
- HDFS test maxed out network on unordered file - but maybe some tuning could help.



HDFS and Ceph

- Both HDFS/ Ceph easily configured and performant (comparable to site's DPM and GPFS).
- Eg. for ROOT jobs reading all events in ATLAS AOD files.
- Ceph like GPFS seems to handle well random access of "old" ATLAS AOD files.
- HDFS test maxed out network on unordered file - but maybe some tuning could help.



Outline

- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 Application tuning
 - ROOT I/O
- 3 Filesystem / protocol tuning
 - Rfio buffer sizes and readaheads
- 4 Alternative technologies
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination**
- 6 Summary / Future plans

Coordination

GridPP Storage Group

- UK based group - but open to all!
- Currently around 70 members in 10 different countries.
- List (gripp-storage@jiscmail.ed.ac.uk) and weekly meeting.
- Some central effort but largely community based.

Provides:

- Liason with experiments, WLCG.
- Forum for experiences and advice.

Develops:

- Supporting tools.
- Middleware/hardware evaluations.
- Services - such as integrity checking and cleanup (esp. for smaller VOs).

Outline

- 1 LHC data access and storage
 - Problems
 - Solutions
 - Tests performed
- 2 Application tuning
 - ROOT I/O
- 3 Filesystem / protocol tuning
 - Rfio buffer sizes and readaheads
- 4 Alternative technologies
 - Fuse with DPM
 - HDFS and Ceph
- 5 Coordination
- 6 Summary / Future plans

Some future plans



TestsInABox: For site sysadmins

Some future plans



TestsInABox: For site sysadmins



Real workload: Proper comparisons with tests

Some future plans



TestsInABox: For site sysadmins



Real workload: Proper comparisons with tests



Evaluate hot topics e.g. NFS 4.1 and xrootd in DPM

Summary

- Efficiencies for LHC data analysis must improve.
- Considerable recent progress: improvements in (experiments use of) ROOT IO and tuning of existing systems.
- Also many emerging technologies.
- Tuning depends on site hardware, mix of jobs etc. - so ongoing
- Increasing understanding of how to test and monitor.

Summary

- Efficiencies for LHC data analysis must improve.
- Considerable recent progress: improvements in (experiments use of) ROOT IO and tuning of existing systems.
- Also many emerging technologies.
- Tuning depends on site hardware, mix of jobs etc. - so ongoing
- Increasing understanding of how to test and monitor.

Summary

- Efficiencies for LHC data analysis must improve.
- Considerable recent progress: improvements in (experiments use of) ROOT IO and tuning of existing systems.
- Also many emerging technologies.
- Tuning depends on site hardware, mix of jobs etc. - so ongoing
- Increasing understanding of how to test and monitor.

Summary

- Efficiencies for LHC data analysis must improve.
- Considerable recent progress: improvements in (experiments use of) ROOT IO and tuning of existing systems.
- Also many emerging technologies.
- Tuning depends on site hardware, mix of jobs etc. - so ongoing
- Increasing understanding of how to test and monitor.

Summary

- Efficiencies for LHC data analysis must improve.
- Considerable recent progress: improvements in (experiments use of) ROOT IO and tuning of existing systems.
- Also many emerging technologies.
- Tuning depends on site hardware, mix of jobs etc. - so ongoing
- Increasing understanding of how to test and monitor.

Summary

- Efficiencies for LHC data analysis must improve.
- Considerable recent progress: improvements in (experiments use of) ROOT IO and tuning of existing systems.
- Also many emerging technologies.
- Tuning depends on site hardware, mix of jobs etc. - so ongoing
- Increasing understanding of how to test and monitor.

Co-Authors

Sam Skipsey, Chris Walker, John Bland, Phil Clark, Efstathia Mouzeli

Other Acknowledgements

Jens Jensen, Alessandra Forti and others in GridPP Storage
Daniel van der Ster for Hammercloud help.