GridPP
UK Computing for Particle Physics

Tier-2 optimisation of dCache and DPM

Greig A Cowan

University of Edinburgh

Graeme A Stewart, Jamie K Ferguson

University of Glasgow

GridPP
UK Computing for Particle Physics

Outline

- Background: a Tier-2 perspective of storage
- Objective of this work
- Details of optimisation tests:
  - Pool filesystems
  - Linux kernels
  - Transfer parameter configuration
- Results, comment and analysis
- Future work
- Conclusions

A ‘typical’ Tier-2

No such thing as ‘typical’, but there are some commonalities, i.e.

- Limited hardware resources:
  - One or two nodes attached to a few TB of RAID’ed disk.
  - Some storage NFS mounted from another disk server.
  - No tape storage.
- Limited manpower to spend on administering/configuring an SRM.
- Choice of SRM solutions (dCache, DPM, StoRM . . .)
- Require the SRM they choose to be optimised in order to be able to handle the data flows that are expected when the LHC comes online.
  - GridPP service challenge set target that all T2s should be able to sustain T1$\rightarrow$T2 transfer rate of $\geq$ 300Mb/s.

GridPP
UK Computing for Particle Physics

Objective

To use LCG's File Transfer Service (FTS) to study a typical T2 SRM setup, looking at how changes in the:

- disk pool filesystems
- Linux kernels
- FTS transfer parameters

affect the **data transfer rate** when writing into the test SRM.

- GridPP uses both dCache and DPM, so run tests for both.
- Want to be able to make recommendations to sites about the optimal setup to use.

Hardware used

- Representative of Tier-2 hardware.
- Single Node running both admin **and** pool services of dCache/DPM.
  - Dual core Xeon.
- 5TB RAID-5 disk, 64K stripe. Partitioned into 3*1.7TB filesystems.
- Source SRM was a local DPM, capable of reading data at a sufficiently high rate that it would not act as a bottleneck.
- Gb/s network between the two SRMs (no firewalls or other annoyances in the way).

Pool filesystems and kernels

- Four different filesystems on 2.4 and 2.6 series kernels.
- Could not run xfs under vanilla SL3.0.5 2.4 kernel - use CERN 2.4 xfs kernel instead.

	SL3.0.5 vanilla 2.4	SL3.0.5 CERN 2.4 xfs	SLC3.0.6 2.6
ext2	Y	Y	Y
ext3	Y	Y	Y
jfs	Y	N	Y
xfs	N	Y	Y

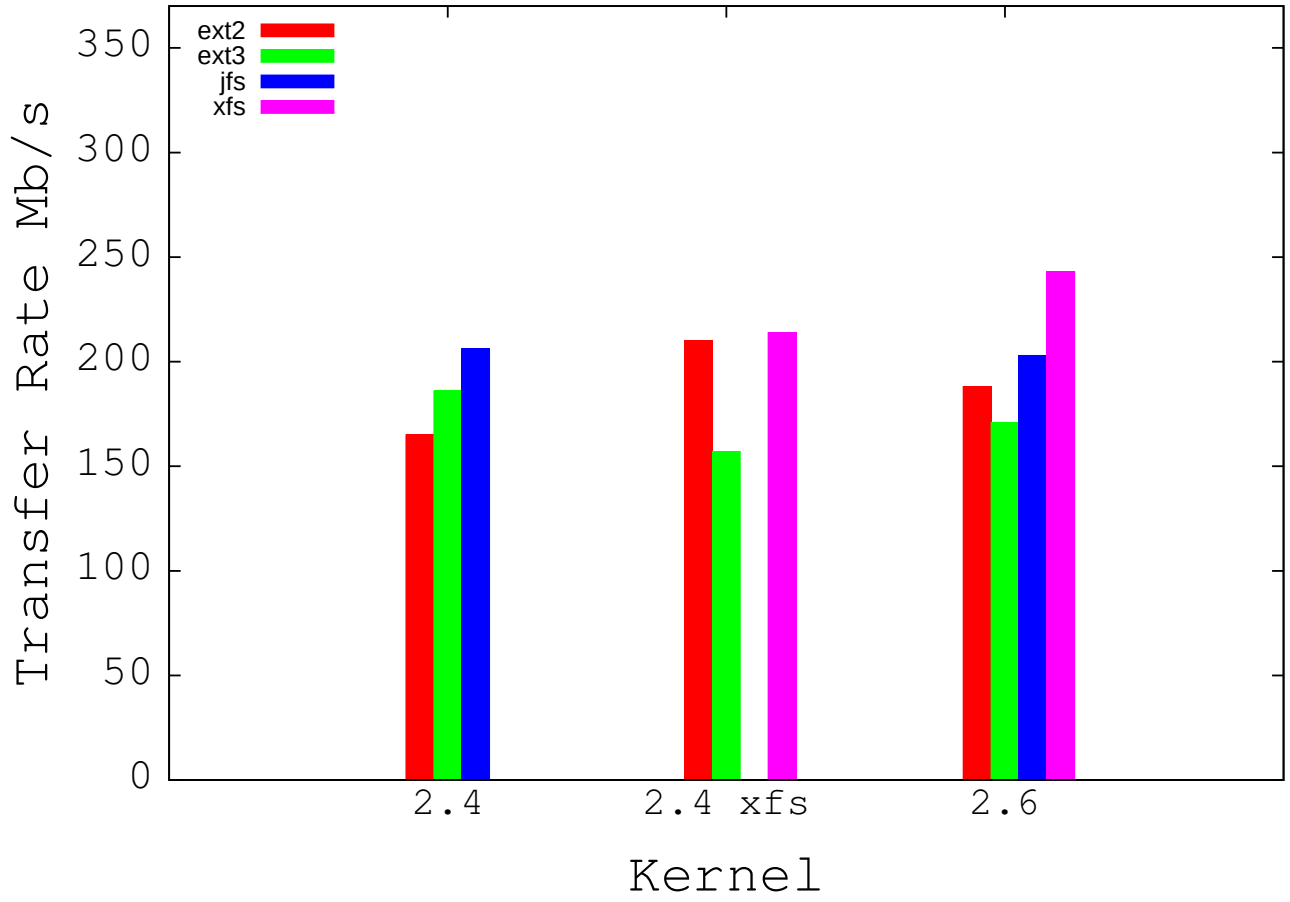
Transfer parameters

Many possibilities here, but we only looked at two that could be modified via FTS:

1. Number of concurrent files (i.e. number of files that FTS attempts to simultaneously transfer). $N_f \in \{1, 3, 5, 10\}$
2. Number of parallel streams (i.e. number of GridFTP streams used per file transfer). $N_s \in \{1, 3, 5, 10\}$

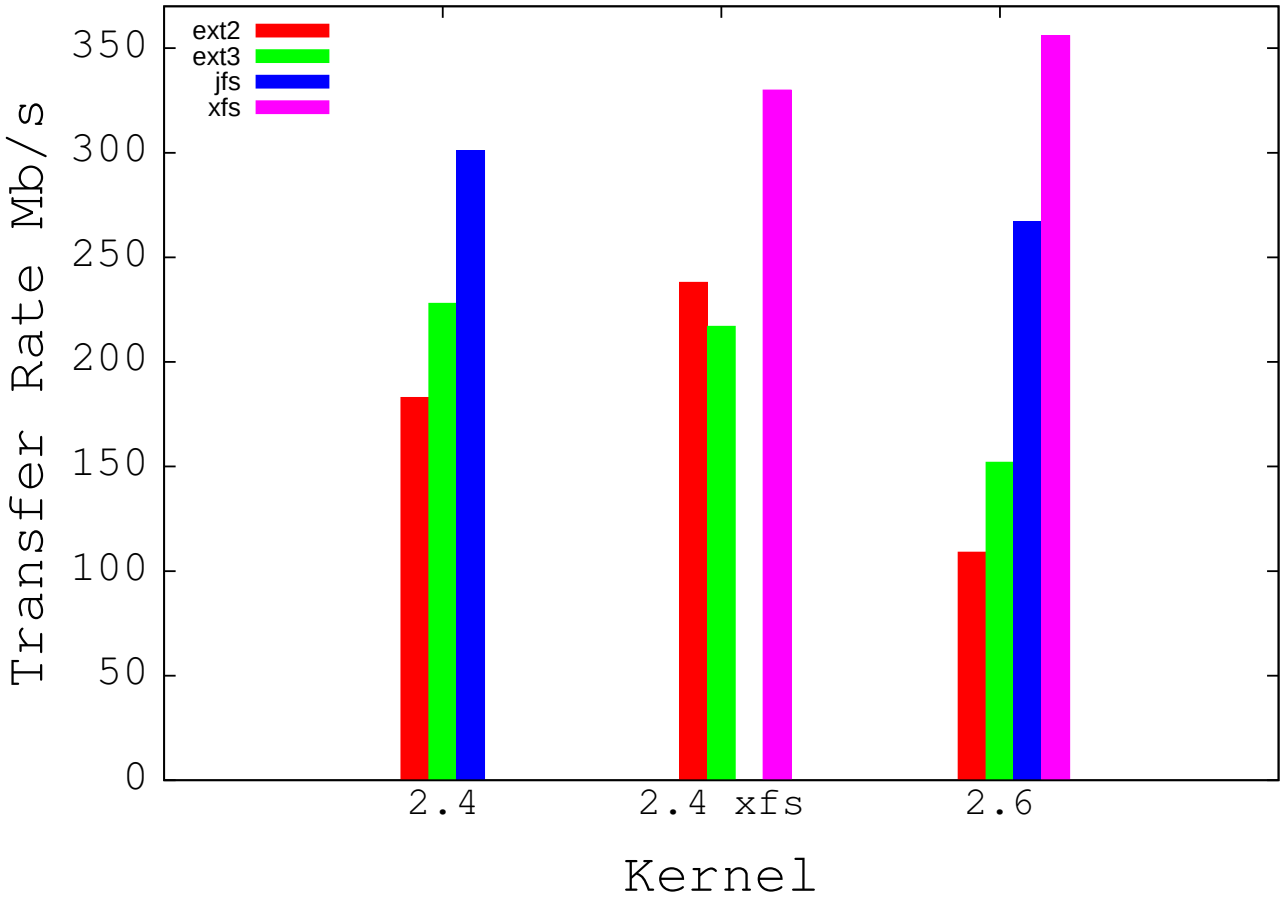
Submitted FTS job to transfer 30*1GB files from source DPM into our test SRM. Using FTS allowed us to monitor the status of the jobs (Done, Waiting. . .)

dCache: $N_f = N_s = 3$



No files failed to transfer.

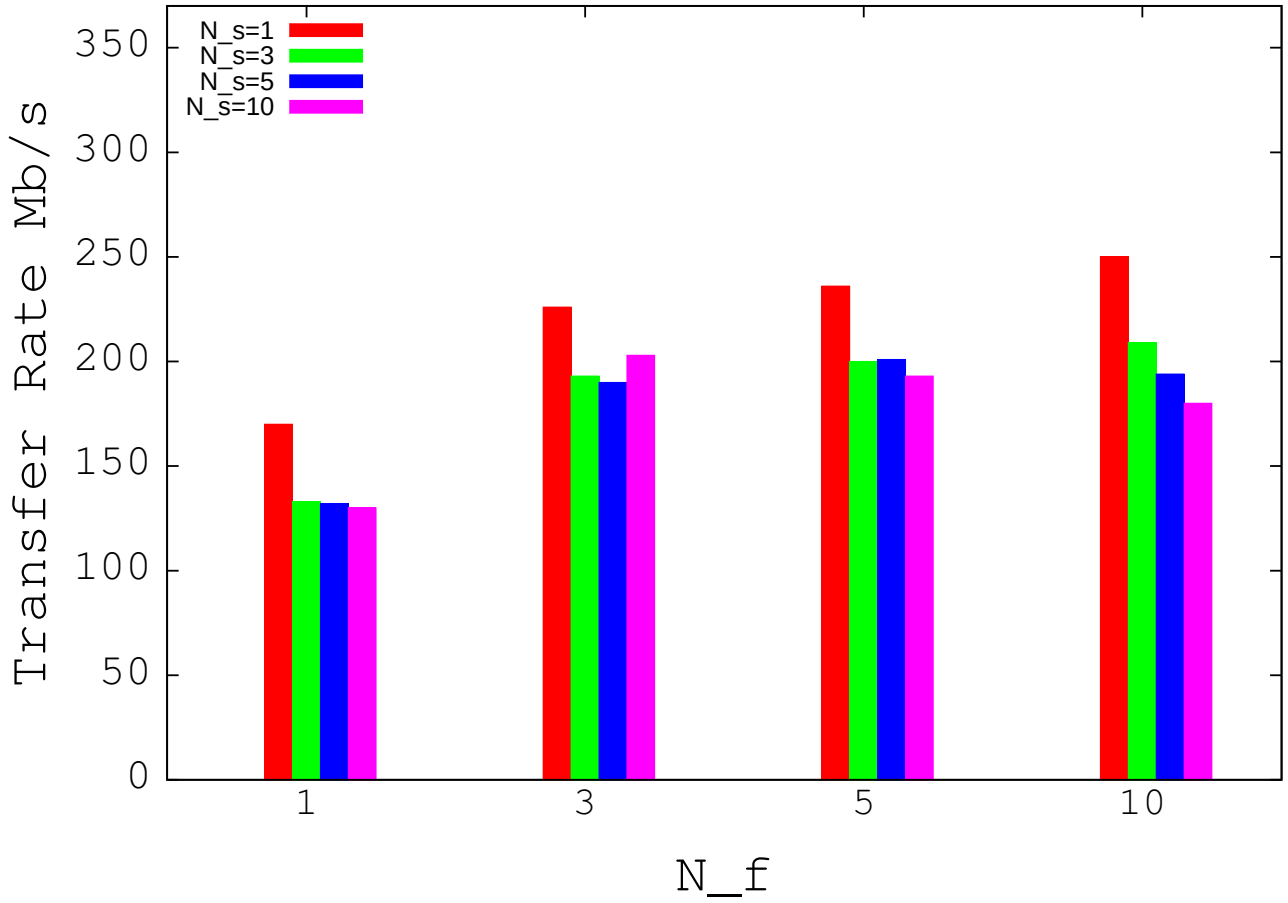
DPM: $f = \quad s = 5$



Small # failed file transfers for:

- ext2,3 with 2.6 kernel and jfs with vanilla 2.4 kernel.

dCache, SLC306, 2.6 kernel, jfs pool



Clearly using **single stream** leads to highest rate. For $N_f = 10$ there is a 20% improvement between $N_s = 3$ and $N_s = 1$.

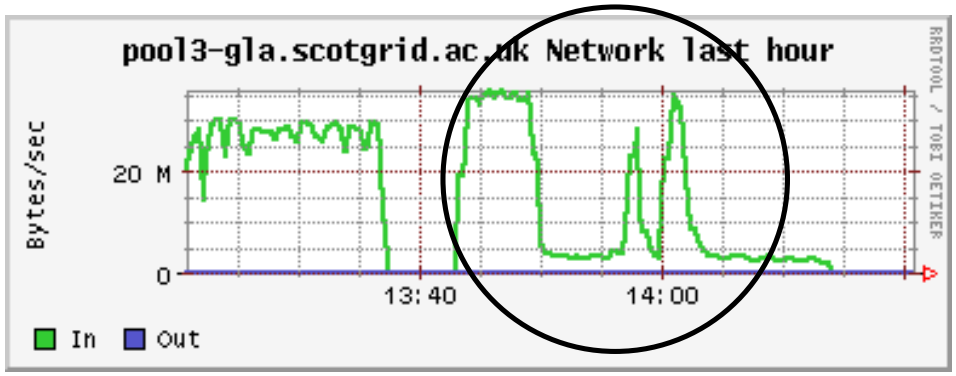
dCache/DPM results

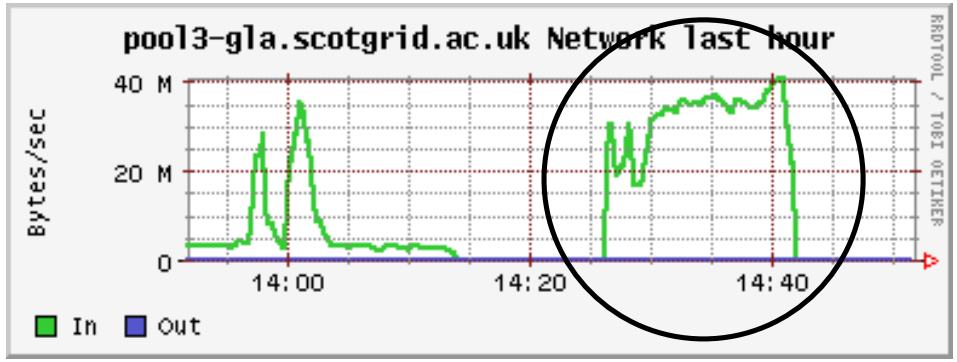
Observed highest transfer rates with the following setup:

- Pool filesystem: xfs
- OS/Kernel: SLC 3.0.6, 2.6 kernel
- FTS parameters: $N_f \sim 10$, $N_s = 1$

Comments 1

- Did not perform systematic study for case of $N_f > 10$.
- Initial tests show that problems occur if N_f is large since it leads to high load on machine after first batch of files transferred. Possibly due to SRM negotiation.
- For example 30*1GB files into jfs dCache pool:





$N_f = 15$

142Mb/s, 15 failed

$N_f = 1 \rightarrow 15$

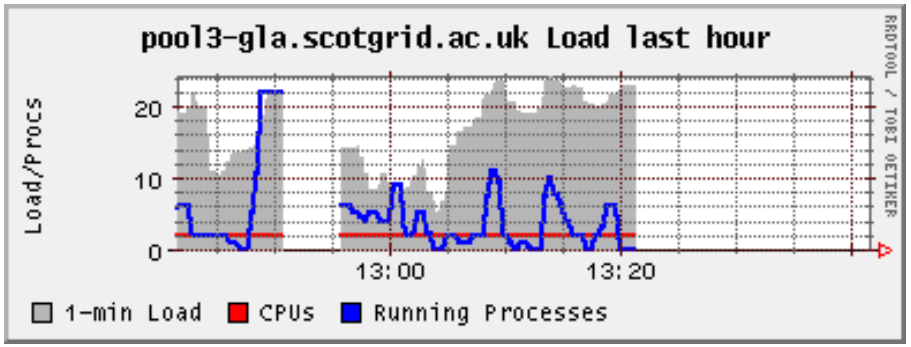
249Mb/s, 0 failed

- Would be better if FTS **staggered** the start times of transfers.

Comments 2

Additional dCache testing:

- Changed maximum TCP buffer sizes to match those set in the main dCache configuration file (1MB default).
- Saw 10% performance improvement with 2.4 kernel. No failed file transfers.
- Saw 20% performance improvement with 2.6 kernel running xfs. Again, no files failed. This was not the case for other filesystems where high machine load was observed. For ext2,3 this caused machine to crash:



Future Work

- Other filesystems i.e. ReiserFS, GPFS, Lustre
  - If looking at GPFS, then could make comparison to StoRM. See talk tomorrow for more information.
- Make further investigations of kernel-network tuning parameters for 2.6 kernels.
  - Look at TCP BIC for 2.6 kernels (see T. Ferrari's talk from Monday).
- Repeat tests for different RAID stripe sizes.

GridPP
UK Computing for Particle Physics

Conclusions

- Era of SRM at Tier-2 sites is upon us.
- Sites need to deploy and configure their SEs hardware and software in order to meet the needs of the experiments computing models and to provide efficient service to users.
- Tier-2's typically do not have time to carry out this optimisation themselves. They need guidelines/recommendations.
- Tests have shown that xfs leads to highest file transfer rate when used on dCache and DPM disk pools.
- Still further tuning work to be done to extract optimal performance from the SRM systems.

dCache configuration

- `parallelStreams` in dCacheSetup file had no effect on the FTS transfers. Only has effect when using `srmcp` to initiate transfer.