GridPP
UK Computing for Particle Physics

THE UNIVERSITY OF EDINBURGH

UNIVERSITY of GLASGOW

Optimising LAN access to grid enabled storage elements

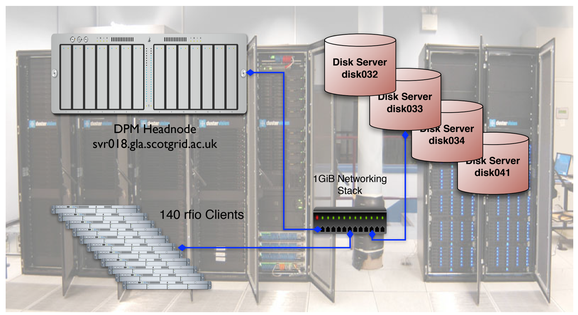
G A Cowan, G A Stewart, B Dunne and A Elwell

Introduction

- Processing HEP data in a grid environment happens in two phases:
  - Gridftp transfer over the wide area network to a site's grid storage element
  - POSIX-like random access to perform physics analysis tasks.
- Here we examine the performance of simulated physics analysis code on the LAN

Test setup

- We utilised the ScotGrid Glasgow production WLCG site (UKI-SCOTGRID-GLASGOW)
  - This site uses the gLite Disk Pool Manager software for its storage system
  - The site has a dedicated DPM headnode and 9 production disk servers with 90TB of storage
  - rfio client side software was run on up to 100 hosts, substantially stressing the system

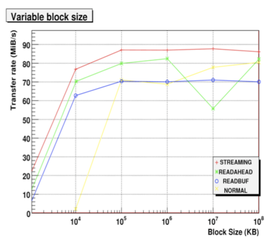


Client measurements

RFIO modes tested

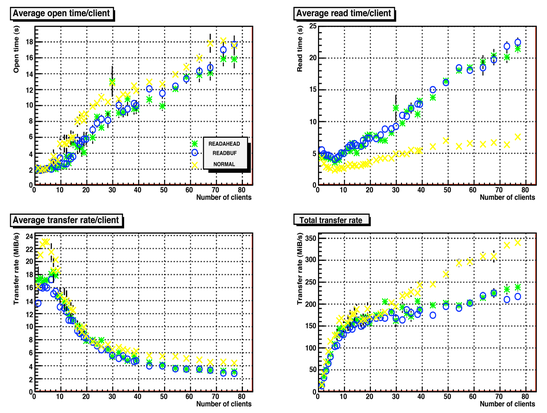
- NORMAL – one call per read
- READBUF – fills internal buffer to service requests
- READAHEAD – user internal buffer and reads until EOF
- STREAM – separate TCP streams for read/control

Transfer rate vs. RFIO block size



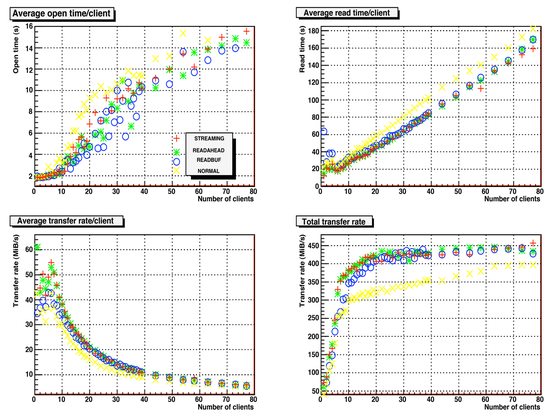
Sequential reading of data. RFIO_STREAM mode is fastest, however all modes benefit greatly from reading with block sizes of greater than 10MB.

File open, read times; transfer rates



Partial File Reads

RFIO performance for simulated physics analysis (reading 10% of a file). Here it can be seen that NORMAL mode obtains the highest overall read rate, because with READBUF and READAHEAD data is read from the server which is not used by the client. (N.B. For this test data was spread over 4 servers.)

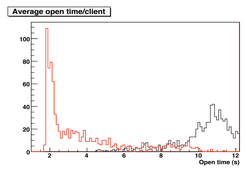


Complete File Reads

RFIO performance for complete file reads. Here it can be seen that NORMAL mode is poorer than all others. (N.B. For this test data was spread over 4 servers.)

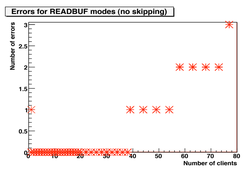
Server Measurements

Open time



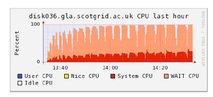
The red histogram shows the distribution of file open times for <23 simultaneous clients. The black histogram shows the case for >23 clients. This shows how the DPM response time changes under load. N.B. all clients start with a file open call within 1s in our tests.

File access errors



In the case where the access to the file is sequential and with a buffer size of 1MB, a small number of errors were recorded for RFIO_READBUF mode. This is a significant improvement compared to versions of DPM before 1.6.5 where the error count grew rapidly after ~20 clients.

Server load



On disk servers network capacity limits the overall transfer rate when complete files are being streamed; however, when partial reads are made i/o wait quickly dominates the load, and the network limit is not reached.

Conclusions

- Utilising STREAM mode is beneficial for a single client when reading entire files, though READBUF anf READAHEAD also perform well.
- However, when many clients are expected to make partial reads from a site, NORMAL mode gives a better rate.
- A data read block size of at least 10MB helps all reading modes.

Further Work

- Server i/o and network optimisations.
- Testing evil client scenarios for DPM robustness.
- Running experiment analysis code.

The authors would like to thank all those who helped in the preparation of this work. Funding for this work was provided by STFC/PPARC via the GridPP project. G Stewart is funded through the EU funded EGEE project.

www.gridpp.ac.uk