



dCache Workshop

"dCache in Service Challenge 3"

James Casey, IT-GD, CERN

DESY, 31th August 2005



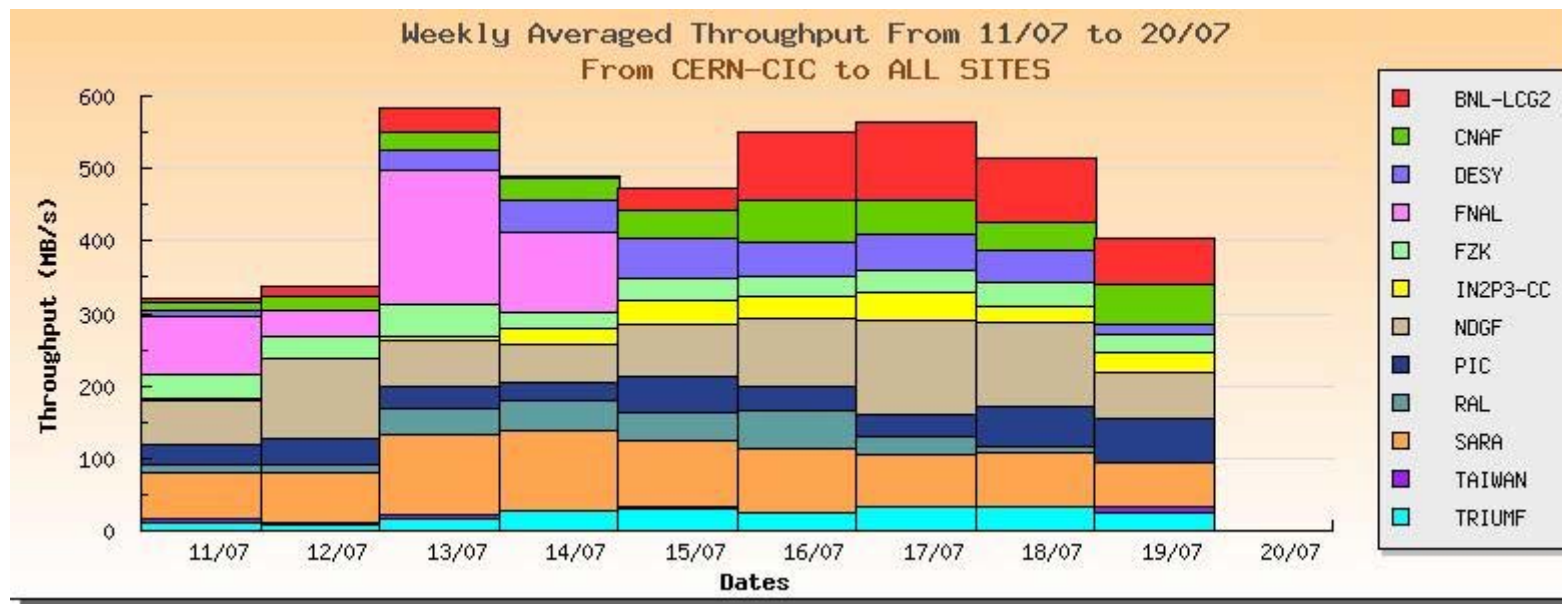


Overview



- Summary of SC3
- Open Issues
 - Long latency network issues
 - Procedures
 - Transfer issues - best practices for #streams & rates
- First results from debugging phase
- Summary

- We haven't met our throughput goals
 - Running at roughly same level as SC2
- We do have all the sites actively involved in transfers





Top daily averages for dCache sites



Site	Daily Average (MB/s)
BNL	107
FNAL	185
GRIDKA	42
IN2P3	40
RAL	52
SARA	111
TRIUMF	34



Issues (1/3)



- Performance on transatlantic networks
 - Very slow per-file transfer rate (~1-2MB/s)
 - Even when multi stream (10/20)
 - Solution is to put a lot of files onto the network at once
 - BNL achieved 150MB/s but with 75 concurrent files
 - We see a lot of timeouts happening
 - FTS retries and the transfers have a high success rate but we lose effective bandwidth
 - These sites have a lot of bandwidth that we don't use
 - e.g. ASCC have 2G/s but it's hard to fill even with TCP based iperf
- Q: How do we up the single file transfer rate on transatlantic sites?
 - Do we need to go back to per site network tuning?



Issues (2/3)



- SRM cleanup procedures are not understood
 - Often we see something going wrong on the transfers and we diagnose and solve the problem e.g. all allocated transfers have timed but movers not cleaned up
 - But the effect tends to go on longer
 - We see degraded performance afterwards and often the sites ends up just rebooting everything
- Q: How can we create, document and share standard procedures, so we don't have to reinvent the wheel 11 times?



Issues (3/3)



- During SC2, we tended to run with few transfers and a single stream per transfer
 - INFN – 10 single stream file transfers 100MB/s
 - FZK – 3 single stream file transfers – 150MB/s
- Now we don't see this
 - INFN has good file transfer rates (~10-15MB/s) but we only get 60% utilization of the network
 - FZK sees very low file transfer rate (~1-2MB/s) for many file transfers (but some seem to run much faster)
 - PIC (& IN2P3/SARA) work best when doing 10 concurrent streams
- Q: How can we reduce number of streams and get individual file rate higher (and more stable) ?

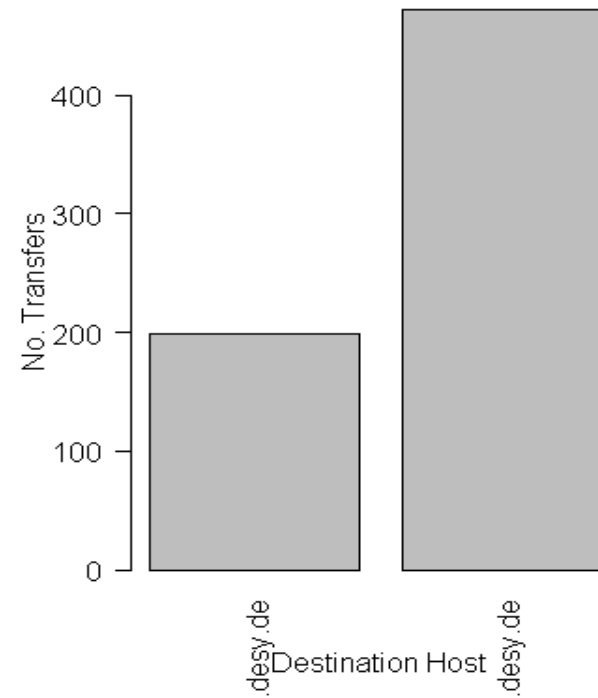
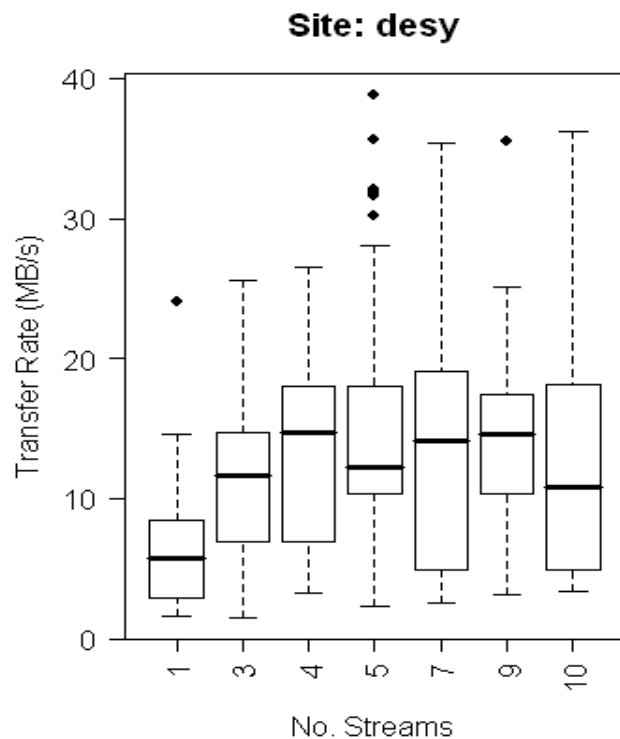


Debugging Phase

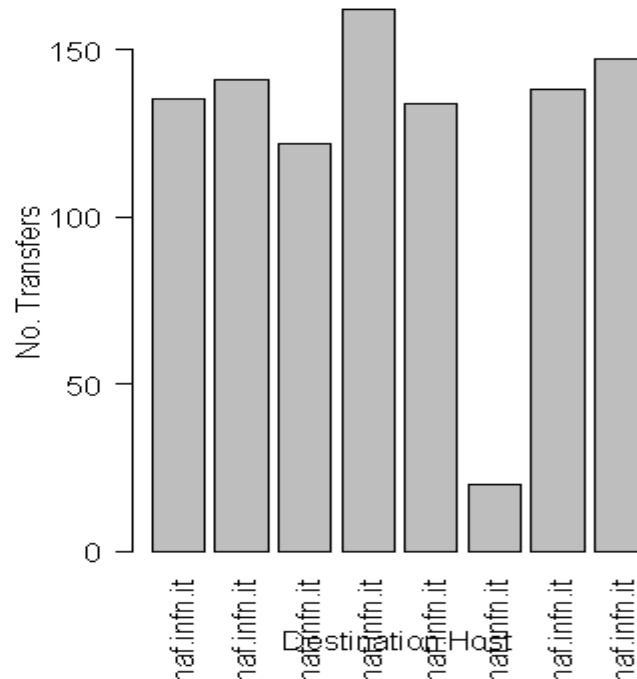
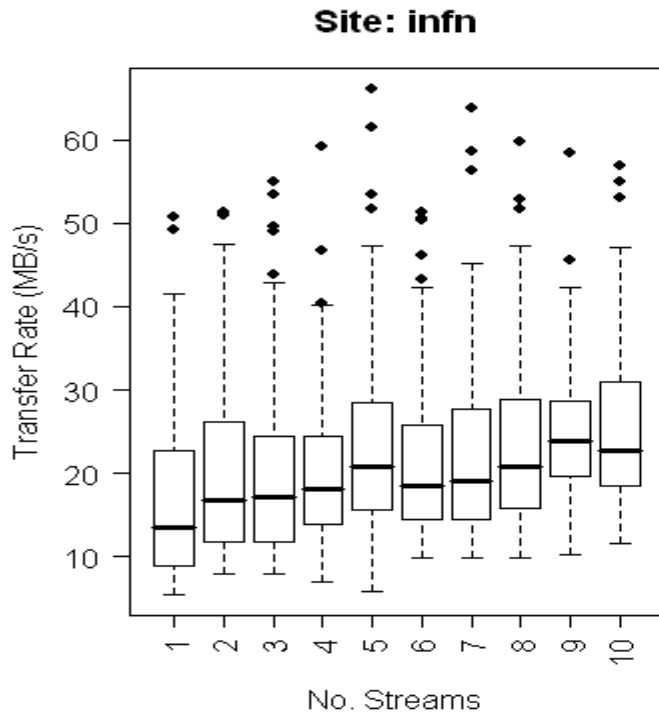


- Tackled the third problem:
 - How can we get higher and more reliable file transfer rates?
 - Looked to answer several questions :
 - What is an ideal node kernel tuning?
 - How many streams are best?
 - What is effect of using SRM Copy?
 - Restricted to low-latency sites, since network issues seem to play bigger role in high latency network routes
 - Tested with DESY to see how a well-tuned system should behave
 - Comparative results against INFN for CASTOR
- This workshop is a step to tackle second problem:
 - Procedures and knowledge sharing.

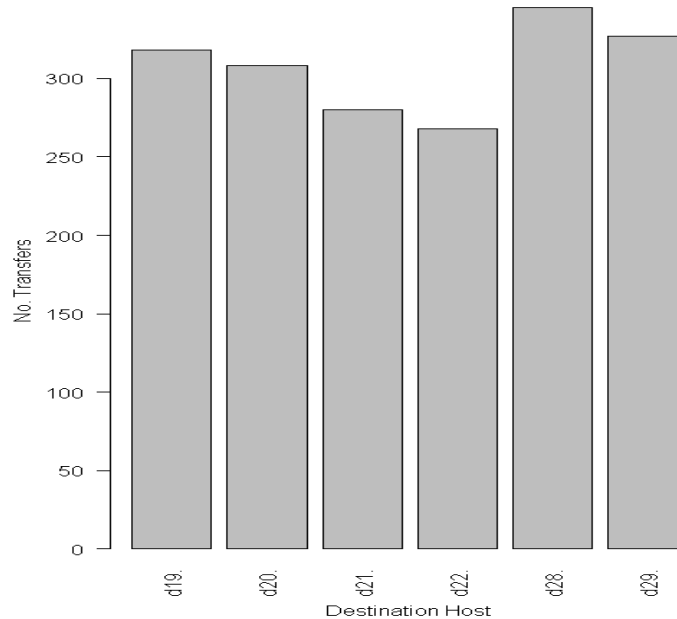
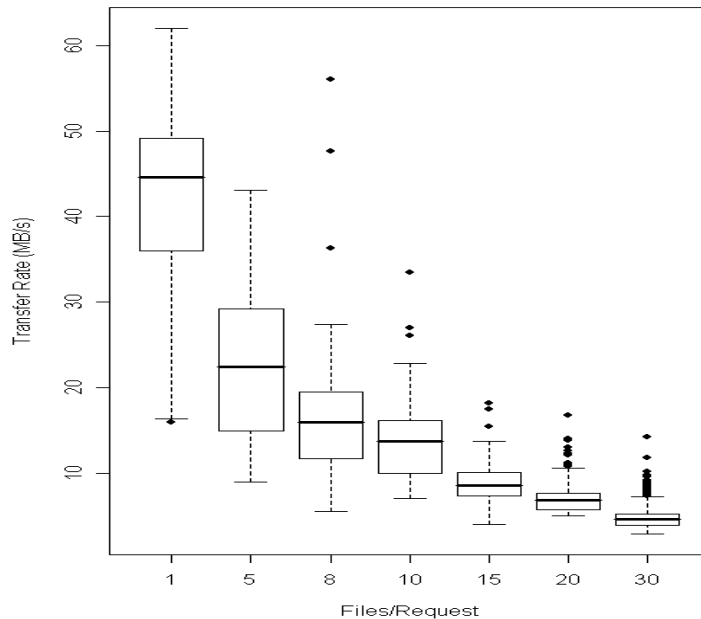
- With dCache transfer rate does not seem to scale with no. streams.
 - "# streams x #files ~ 50"



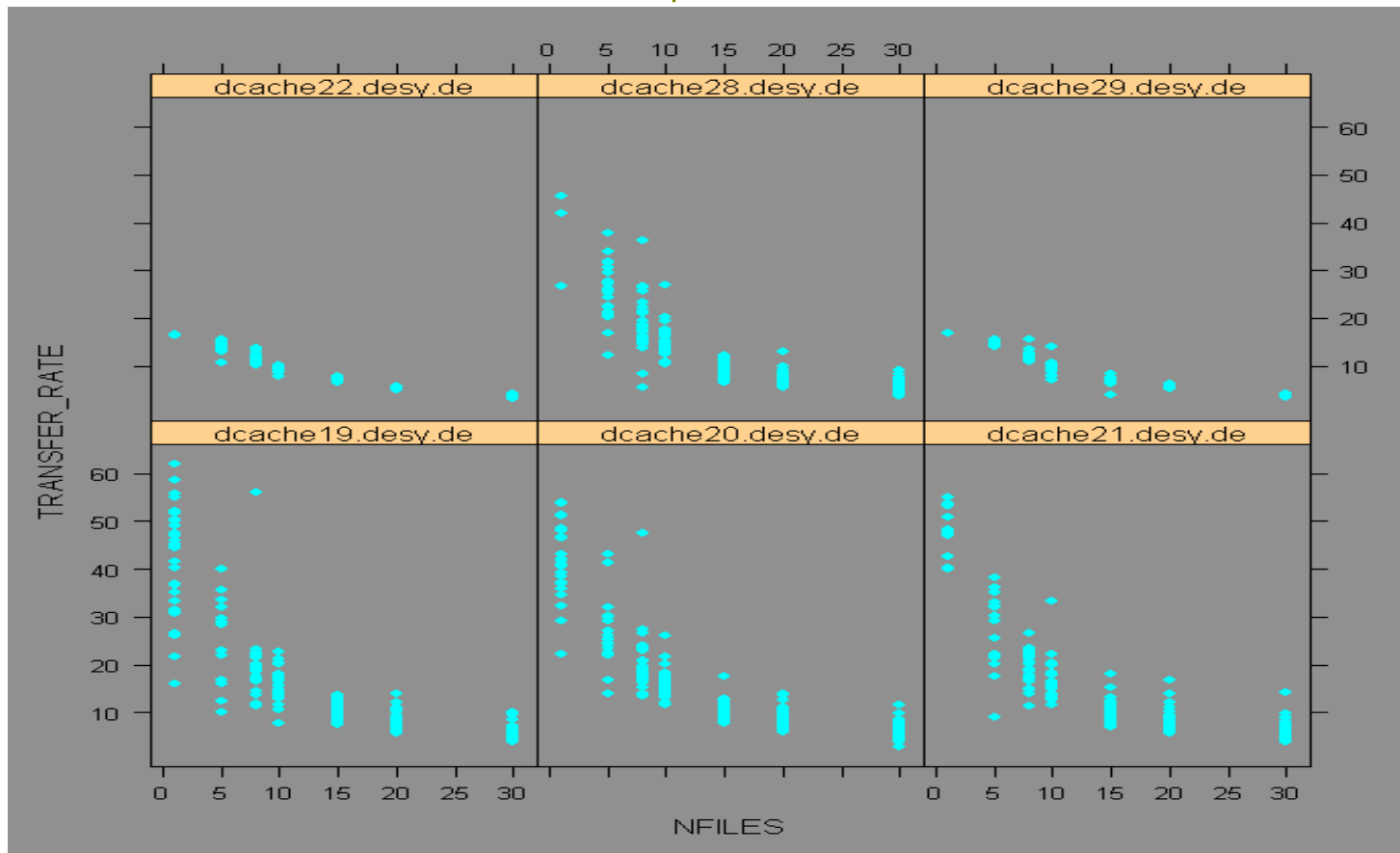
- Slight increase with no of streams (fixed to 10 concurrent files)
 - But total bandwidth did not translate to $\sim 20\text{MB} \times 10$ – was in the range of 60-80MB/s.



- We tended to fill bandwidth
 - but single file bandwidth inv. prop. to # streams
 - CASTOR returns all TURLs immediately, so dCache transfers them
 - Resource management needs to be done on both sides



- Note effect of different TCP buffer sizes
 - 22+29 had 64K buffers, the rest had 2M buffers





Monitoring ongoing transfers



- FTS used gridftp performance markers
 - Has 120 seconds marker-to-marker timeout
 - Has global transfer time set much higher (~1hr)
- dCache does not send the performance markers
 - This initially caused all long-hop transfers to time out
 - Have to disable this feature
- Had the effect of if any problem occurs, it takes 1hr to fail !
 - Bad for channel utilization

- 1 DPM pool node out of 6 started to fail on gridftp
 - SRM kept scheduling to that node
 - Reminiscent of Globus gridftp black holes from SC2
- Rate drops from 150MB/s to 80MB/s





Summary



- Added some knowledge
 - 5 streams is a good number for 10 concurrent files with FTS
 - But dCache does seem capable of running high speed single stream transfers
 - Srmcp gives better load balancing over door nodes
 - With FTS, all pool nodes were used for storage, but door node usage wasn't balanced
 - But throttling needed in other SRM implementations to stop dCache overloading them
 - #files x file transfer rate != throughput
 - Significant lossage, due to SRM overhead and FTS scheduling
 - Managed storage is a good thing for resilient and "self-healing" system