CERN Mass Storage Issues

Experience from the last 4 month of Data Challenges and current running experiments (I)

Few issues :

1. Disk server instabilities caused problems (inefficiencies, lack of Castor error-recovery) and shortage in available disk space
   → server instability understood and fixed, now more or less back to normal running conditions
   → redundancy/error-recovery improvements in the new stager

2. Overload of the nameserver due to experiment specific queries
   -- large directory listings (300000 files during LHCb production))
   -- use of the nameserver as experient production control system (regular, large status) queries
   → for the future have one nameserver per experiment
   → throttling/priority mechanisms for Oracle queries (not easy)
   → better collaboration between Experiments and IT

Experience from the last 4 month of Data Challenges and current running experiments (II)

Few issues :

3. Open connections from/to disk servers from large number of batch jobs for long times
   → use the lxbatch local disks as extra cache layer
   improves the situation (ntof), but needs more investigations to understand the full implications user perception ?!

4. large numbers of single file requests from tape, which leads to tape drive inefficiencies (mount-unmount time is ~ 120 s)
   → new Castor design improves the aggregation of requests
   → multi-file staging possibilities needs to be discussed/organized with the experiments

Experience from the last 4 month of Data Challenges and current running experiments (III)

5. Large numbers of small files

a) limits in the current stager , only one million files can be kept
on disk (no tape limit), but the performance gets already very slow with 500 K files
due to the catalogue structure
this was seen with CMS and partly with Atlas
major problem with ALICE, production phase 1 produced more than
2 million files (size between 50 KB and 1 GB)
needed 2 more stager nodes for this
→ limit will disappear in the new stager, which is currently been tested
(to be used also in the ALICE-IT computing DC)

Experience from the last 4 month of Data Challenges and current running experiments (IV)
Major problem

b) large numbers of small files are creating large overheads and thus inefficiencies

example : 1 MB files versus 1 GB files (ALICE wants to import in phase II of their DC 7 million 1 MB files)

major point is the tape system, as tapes are designed to run most efficient with large files. There is more than 1s real time spent per file on tape (tape mark)
thus 1MB file == < 1MB/s transfer speed, 9940B tape drive runs at 30 MB/s
➔ inefficiency is > 97 % (for 2008 we have foreseen ~ 200 drives for 4 GB/s tape performance assuming sequential read/write with GB files
extreme : with 1 MB files we need 10000 drives)

Experience from the last 4 month of Data Challenges and current running experiments (V)
Major problem

other effects :

- open+close of remote connections (e.g. RFIO) takes 0.1 s (== 10 MB/s max)
  → more streams per server and disk needed to get the aggregate performance
  e.g. in the current calculations for the large CDR disk buffer for 2008 we need from the pure performance (GB/s / perf. per 1 TB disk) point of view about 50 TB of disk space, but if one includes the # of streams we need 500 TB (already with large files…) → more streams per disk moves the perf. from sequential to random I/O = factor 10 perf. reduction
- RLS insertion 3s per file (from the CMS DC)
- response time of large number of files in a directory
- OS caching issues
- security checks per file (GridFTP >= 1s)
- large buffers for effective WAN transfers (> 100 MB ??)
- ……..

3 possible solutions

1. bigger files in the experiments (RAW, EST/DST, AOD, NTUPLE !?)
   >1 GB now AND moving to >10 GB in 2008

2. start an intensive investigation about the overhead of small size files everywhere
   (HSM, OS, transfers, protocols, filesystems, disks, tapes, etc.)
   what is the effect ? how big ? scaling ? how to avoid/overcome limits?
   manpower and money ?

3. see whether we can find an optimal compromise between the two
   (for the AOD, NTUPLE part ?)
   mass storage versus shared filesystem