Disk Space Issues

Bernd Panzer-Steindel, CERN/IT
Draft v4 3.Nov 2006

The current requirements of the experiments for disk space are based on some basic parameters (trigger rates, event sizes) and computing model assumptions (definition of physics data sets, number of re-processing activities per year, number of physicists involved, distribution of data sets on different sites, number of data set copies, etc.). The resulting amount of space is then increased by an efficiency factor of 70% (x1.4) to cover experiment specific issues (fluctuations in event size, overlap of physics channels, fluctuations in the number of calibrations events, etc.).

These calculations do not take into account any I/O performance considerations or other overheads and of course assume that this is usable disk space and not 'raw' disk capacity.

The difference between 'raw' disk capacity and usable capacity can be large and depends on a few factors :

1. how many file systems per box ? the creation of a file system reduces the space by about 2%
2. how are the disks combined into file systems ? how many RAID controllers
3. does the disk space management allow for 100% usage ? Different policies : durable pools , pinning, lifetime guarantees
4. Fragmentation effects performance and depends on the read/write usage patterns and how full the file systems are

Taking a single disk server with 16 data disks as an example:

- raw capacity is 16 * 320 GB = 5120 GB
- the RAID controller has 2x 8 ports and can't span RAID systems; we use RAID5 with one spare disk; 2% file system creation overhead and run at 90% fill rate :
- (2 * (8-2) * 320) * 0.98 * 0.9 = 3387 GB

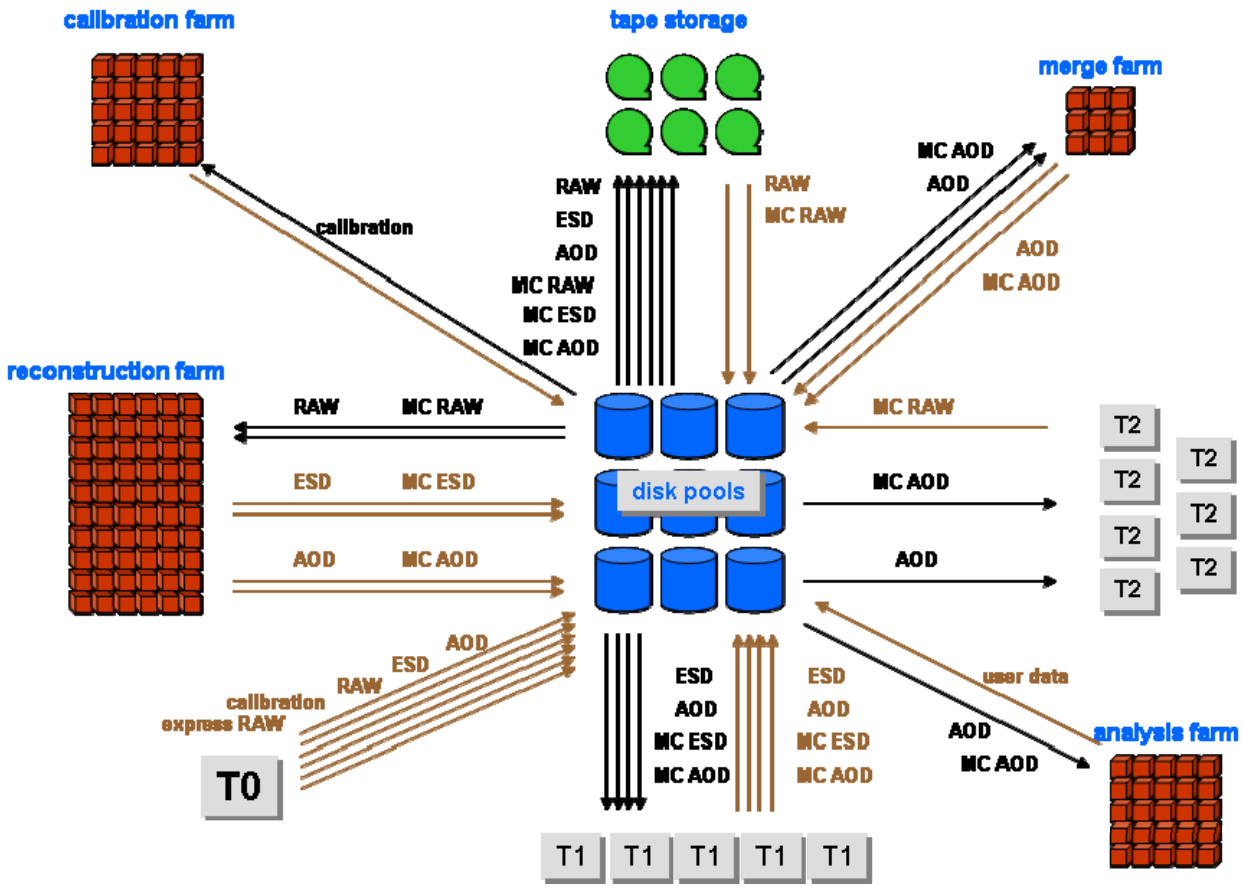
→ thus the usable space is in this case is 66% of the 'raw' capacity.

The influence of the I/O performance on the disk space calculation depends on more parameters and has to take into account the characteristics of the whole computing setup, the expected data rates and the usage patterns:

- number of concurrent jobs = number of cores * 1.x = more than 4 jobs on a 4-core node to improve the CPU utilization efficiency (I/O wait-time hiding)
- number of available tape drives

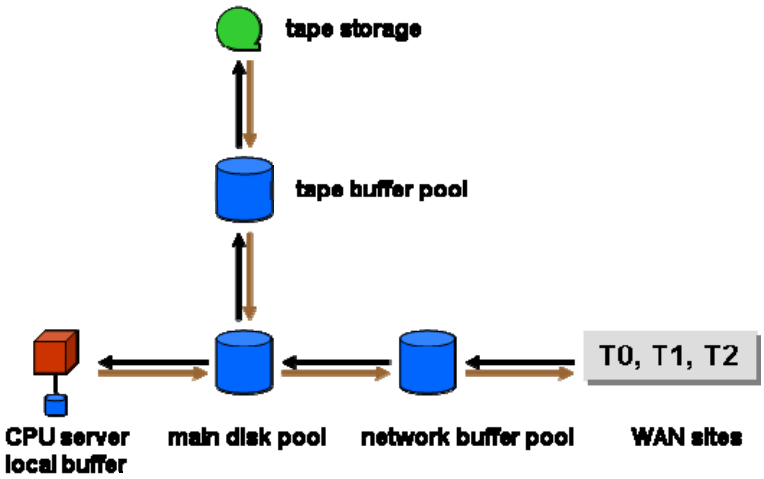
- file sizes
- efficiency of the mass storage system
- number of disk servers
- file system layout, (RAIDx) and performance (write over read preference)
- IO transport application overhead (dCap, RFIOD, rootd, GridFTPd, etc.) and footprint
- Level of data set replicas
- Aggregate IO performance
- Number of concurrent IO streams and IO operations/s
- Number of disk pools and their functional separation
- …….

The following picture shows the principle data flow in a typical T1 center which has to work on quite a number of different concurrent tasks : several times a year the re-processing of RAW data and production of AOD and ESD data, RAW data transfers from the T0, import of MC data, export of ESD and AOD data, analysis of ESD and AOD data, calibration data processing, etc.



There is a priori no need to separate all these activities into separate disk pools but rather use one large pool. Limitations in hardware and software requires in real life the split into different pools with different characteristics based on the application access patterns. In addition there is the need to have an extra disk buffer layer between the pools and the surrounding computing infrastructure (tape storage, WAN, computing nodes).

In a simplified picture of the data flow one can identify the three different buffer layers:



Merging

During the processing of the RAW or ESD data smaller AOD files are produced. As it is better to deal with larger files some merging activity needs to take place. At CERN this was exercised in the last two ATLAS T0 data challenges at nominal ATTLAS performance (20 MB/s in + 20 MB/s out). Three disk server were used (5TB, 3 file systems each) to cope safely with the number of concurrent streams (up to 200, many writes and few reads) and to provide a redundant facility. With the mentioned access pattern the provided pool should be able to cope also with 2-3 times this data rate.

WAN buffer

In the current configuration we have seen that for reaching high aggregate bandwidth between sites there has to be a large number of concurrent transfers.
To calculate how much disk space is necessary for a certain bandwidth I assume the following input parameters

- a disk server on GB Ethernet can do 80 MB/s input plus 80 MB/s output
- a disks server has three file systems with a total of 5 TB usable space

- file systems don't like a mixture of read and write operations, thus we assume a double buffering scheme : each file systems has either read or write operations
- we can have a few write or read operations simultaneously per file systems (5)
- there is a certain footprint of the GridFTP clients and server programs on the disk server and thus we want to limit this to 25 per server. Each physical stream is in addition split into several parallel TCP streams occupying buffer memory on the node.
- the average speed per transfer is about 2 MB/s

Thus to receive 200 MB/s from the WAN and distributing the data to another pool one would need :

1. from pure disk server aggregate performance : 400 / 160 = 2.5 + some redundancy = 5 disk server (25TB)
2. with 200 MB/s aggregate bandwidth and 2 MB/s per stream we have 100 low performance write streams and maybe 25 higher performance read streams for the further distribution of the data.
   - → with a max of 25 streams per server we would need 5 server plus two for redundancy = 7 server (35 TB)
   - → with a max of 5 write streams per file system and a non-overlapping read and write we would need 100/5 + 25/5 = 25 file systems which means ~8 server plus 2 for redundancy = 10 server (50 TB)

The assumed numbers here depend strongly on the type and performance of disk servers, file systems, tuning of GridFTP and FTS and the used HSM system.

Tape buffer

The tape storage infrastructure is very expensive and the key point here is to use the tape drives in the most efficient way. A simple cost comparison shows that one tape drive (plus the tape server = 35 KCHF) is equivalent to about 4 disk server (9 KCHF for a 5 TB node on GB Ethernet). Modern drives (IBM, STK, LTO) can reach 100 MB/s read and write performance which was confirmed by our measurements but of course only for streaming mode and GB file sizes. Based on the experience from the various data challenges it is far more realistic to assume production performances of 40-50 MB/s. To calculate the needed buffer space for a good 'impedance' matching between a disk pool used for re-processing (or the 'WAN' pool) and the tape storage system I make the following assumptions in this example:
The required performance of 300 MB/s requires 8 tape drives to operate constantly and the streams are distributed in a double buffer procedure over 16 disk server. So each disk server has either read operations or write operations ongoing but never both at the same time. Thus this operation requires 16*5TB=80 TB of extra disk space. If one is less conservative and assumes that the separation of read and write is only done on the file system level (without overloading the disk server) one can reduce the 80 TB to 40 TB.

Another consideration for the size of the buffer is the protection against failures in the tape storage system. If one just wants to be covered for a failure over night a 16h buffer would be sufficient (16h * 300 MB/s = 17 TB) while a weekend would need about 65 TB. This space would be in addition to the 'impedance' buffer mentioned before.
All this depends of course heavily on the capabilities of the used HSM system

CPU node buffer

The manner in which the data is accessed from the CPU serves by the experiment programs plays also an important role for the design and sizing of the disk pools.
There are two different policies which have different effects :

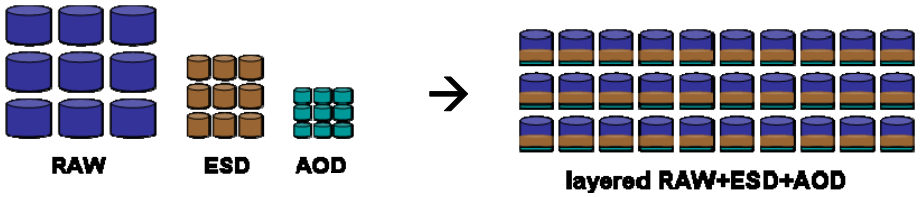
1. Direct I/O connections
   In this case the experiment programs open direct I/O transfers from the disk servers, thus files are opened and read/written during the lifetime of the program. This creates a large number of concurrent I/O streams but with low performance per stream. The load and possible limitation on the disk servers depend on the footprint of the involved IO daemons (rfio, dcap, etc.).
2. Extra cache layer
   The programs first copy the data file they want to access to the local disk of the worker node and also write the output locally before moving these files back to a disk pool. This seems to be today a preferred solution by the experiments.
   As a consequence the access to the disk pool changes from many low performance IO streams to much fewer high performance streams, which the disk pools might be able to cope with in a better manner. On the other side one has added higher load to the CPU server. Several programs (n-core system) are now loading the local disk with considerable read and write operations. Currently the worker nodes have just a single disk attached (<200 GB). With the new dual-cpu dual-core nodes this model will start to cause probably local disk congestions.
   The cost of increasing the local buffer space and improve performance would increase the costs of the CPU nodes by about 5% - 10% :
   - → a modern CPU node (4-cores, 8 GB memory, 160 GB disk) costs about 4500 CHF, by replacing the 160 GB disk with 2 X 250 GB disks we have to add about 220 CHF per node = ~ 5%
   - → adding a third disk and a controller for much better performance would increase the cost by about 10% (1U CPU nodes can have a maximum of three disks attached, front access to hot-swap disk would increase the costs further

   The extra copy of data introduces additional IO overheads for the CPU nodes. There is on one side the extra CPU resource usage for the copy of the data and the latency of the copy process. Both are in the 1-2 % range for typical re-processing exercises, but could be much larger in the case of data analysis

Analysis issues

During the analysis phase a number of users want access to relatively small (TB) data sets in a more random fashion. This requires

- Spread of data sets over as many file systems as possible.
  If one has different data sets of different size in the whole system that one would ideally move from pools with separated functionality and data sets to a spread of data of file systems to optimize the random access behavior.



  But this requires quite some sophistication from the data management system

- RAID1 file systems would give better overall performance.
  It is the number of concurrent IO streams and their individual performance related to the number of disk spindles. But as disks are coupled through RAID file systems it would be better to use the number of file systems as a reference instead of pure spindles; e.g. the random IO performance of a 8-disk RAID5 system is NOT equivalent to 8 single disks, the systems stops scaling with 3-4 disks.

  Moving from a RAID5 to a RAID1 configuration would need about 30% more RAW disk capacity.

- Replication of hot data sets
  Maybe 10 % of the data on disk need 2-3 replica sets

- The PROOF model would require an upgrade of the CPU servers with 2-3 extra disks

Thus one has to assume that the actual disk space needed for the analysis procedures has to be doubled compared with the actual size of the data sets.

Generic overheads

A few tens of TB of disk space has to be available in addition to cope with ‘transitions’ :

- Exchange of data sets
  the old version of data set X has to still stay on disk for a certain time to cross-check with the new version Y

- Preparation of a new re-processing
  Several iterations before stability is reached with multiple smaller ESD and AOD data sets, which need to be checked before the major production starts

And we have also the still unsolved issue of ‘private’ data sets from multiple users.

Some conclusions

This is a complicated area with a large parameter phase space. There are not only the hardware performance values but more important are the boundary conditions from the underlying software (HSM system), the computing models and the ‘impedance’ matching between the computing systems and sites.

T1 sites will have several PB of disk space installed based on the latest experiment requirements. Taking this into account and adding the different overheads discussed in the previous chapters one can probably assume that the possible overall additional disk space overhead is about 10%-15%.
But there is a non-negligible error bar on this number and this is of course site depended.