Table of Contents

Storage Management TEG: Daniele's draft twiki on CMS answers

First of all, a general CMS statement on this process of interaction with the Storage Management TEG and the Data Management TEG (referred to as S-TEG and D-TEG in the following, respectively). We perceive the line between Storage Management and Data Management as somehow blurry, and there are overlaps, also in the questions, so we tried to be inclusive and consistent in a unique set of answers but we are also aware there might be items on which a specific TEG might want to ask us further: if so, we are available to provide deeper and further details as long as we will be asked by both TEGs. Said that, answers (A) to the questions (Q) are drafted below.

The {SM,DM}-TEG questions

Q: " *In your view, what are the 3 main current issues in Storage Management (SM)?* "

From the CMS perspective, some of the main areas (open issues) we identified are:

- **Disk management in a hierarchical storage**
  - i.e. be able to manage the disk in a way we can be sure data is on disk where CMS wants it to be
- **Access to data from disk with increasing volumes and fewer spindles**
  - as we get larger and larger disks we have fewer spindles, more apps and more cores trying to read, the IO of the application and the sw performances get even hotter topics
- **WAN access to storage**
  - i.e. at the moment, refer to the xrootd work by Brian et al, and also to the Federated Storage ws in Lyon (end November 2011)
- **Reliability**
  - an area that's easy to overlook, but it's crucial. In some sense the previous item (i.e. WAN access) stands as a sort of acknowledgement we need a fallback when the storage was not working, but it is worth to list it as a separate item. Think of it as 1) "we wants less errors from the storage systems" as well as 2) "if any, errors should be reported from the storage systems to the CMS application layer that intercepts them in a not-misleading, complete, clear manner".
- **Storage monitoring** (shared with the Operations TEG)
  - We need a better monitoring of what's actually on disk and tape.

Also, but read the caveat in the comments, we would add:

- **intra-VO authorization** (shared with the Security TEG)
  - i.e. one needs to have the right privileges in writing to different storage areas, especially copying data from other sites. E.g. jobs running at the T1s with the /t1access VOMS role trying to write at T2 storage where the priority user role is required to write to the group area. Actually, this can be identified as a *problem* that we definitely have, but not a particular *priority*. At least it's not urgent, in the sense that once we open the T1s to a more intense use by Analysis we will have this problem as a priority. Currently, while we agree it would be nice to have a better solution, we also recognize that we have an *operational* solution in which we solve this issues by granting the needed privileges to users who actually need it.

Q: " *What is the greatest future challenge which would greatly impact the SM sector?* "

Large data volumes and need for access.

Q: " *What is your site/experiment/middleware currently working on in SM?* "

Disk and Tape Management at the Tier-1s.

Q: " *What are the big developments that you would like to see from your site/experiment/storage system in the next 5 years?* "

- More emphasis on the separation on active storage and archival storage. I.e. the concect that you are accessing a fraction of the overall storage available to you frequently and another fraction just for custodial copy or in case a backup recovery is needed. This is probably a longer term goal, and it would profit also of the technologies we might have in the foreseeable future.

- Dynamic data pre-placement using data popularity system
    - this covers not only the intelligent placement of data based on their popularity, but it also includes a full deployment of solutions to enforce automatic discovery of datasets that should be deleted and - if subscribed to "central space" - eventually deleted in a automatic and quick manner, with no manual actions by the site admins

Q: " *In your experience and area of competence, what are the (up to) 3 main successes in SM so far?* "

- The ability to virtualize and operate large scale storage resources at all Tiers of the Computing System
    - if we look back at the ramp-up phase (towards LHC data taking) there were some questions as of whether we would have been able to operate multi-hundreds of TBs at Tiers, and we did, it was a success of data management and storage management
- The ability to move data quickly between sites
    - much easier to populate a site than to actually flash it clean..
- The ability to open data files locally from the application from a variety of storage systems
    - probably now obvious but it was not at the beginning

Q: " *In your experience and area of competence, what are the (up to) 3 main failures or things you would like to see changed in SM so far?* "

- The transaction limits and eventually file size constraints of the wide area interface to storage
    - we spent a lot of time worrying about staging a user file, or having the merge stuff done, because we had a limit in the transaction rate
- The need for disk management and staging without the real ability to track what is managed and staged
    - for the system to work properly we need to place staging requests and make sure all we need gets really staged to disk. But: we do not really have (and we are not sure we want) a very fine-grain over the global system, even if still we might want some more information wrt what we have now.

If we can add a couple more, we would probably pick:

- Reliability and monitoring (see above)

- Better scheme for authorization (see above)

A comment on the latter. There is a specific activity in the Security TEG about the storage security, but AFAWCT there has been no discussion yet, so it's not clear what the boundary is, still. As soon as the activities are set-up on the security side, a joined discussion is needed.

Q: " *We assert that we have a working system, do not need a new system, but the current system (cost and complexity) exceeds what is needed. Would the experiment agree with this statement?* "

CMS does not entirely agree with that statement. The CMS Data Management system PhEDEx and the TFC seems pretty well spec-ed to deal with the problem we have, in terms of moving data around, of providing a scalable way to determine the mapping among LFN and PFN, etc. All this is about the rioght level of complexity. On the other hand, CMS can probably agree to the fact that some elements and functionalities of SRM are beyond what we are currently using and may have exceeded the actually needed complexity. Potentially, FTS may have some excess complexity also. So, we can only agree only partially to the statement in the Q.

Q: " *What elements are being used, and how? Which have been added and why?* "

- used:
  - SRM, FTS, xrootd
- added:
  - PhEDEx: added, because we needed a *dataset-* (and not *file-*) replication system which was scalable and reliable
  - TFC: added, as an "algorithmic" dynamic toll and not a DB-based catalog - e.g. we do not use LFC
  - DBS: added, it covers the physics "metadata catalogue" to be intended in the meaning that the Grid world use for it

Q: " _Which components failed to deliver (larger) functionality needed? Does that mean we can deprecate and obsolete these components, or was it not truly needed?_ "

We mildly disagree on how the Q is made, i.e. "deprecate and obsolete" components which did not offer the pledged functionalities. The quick answer would be that the fact that we are working around some components does not imply we do not need the potential functionalities that such components may offer someday in the future. Of course this must be discussed component by component. Come more details and examples below.

For instance, **FTS** was promised as a more dynamic way of moving files around wrt how it actually is, but the fact we are surviving without some functionalities offered by the tool that was supposed to provide them does not imply that we do not need/like it, it may just mean we do not have it yet, we had of course to develop something ourselves, and we still hope we can use FTS3 and efficiently compare its checksum with the one in our catalogues. In this sense we would object in the Q to the fact that this should imply we then can "deprecate and obsolete" such components. It must be also noted that the fact that we developed some components ourselves mean that the available components did not have the functionalities that we needed. But at the end one of the main reason why we ended up to develop something ourselves had a lot to do also with the achievable performance and not only to the available functionalities. As an example, the development of **DBS** was felt as a need because with other existing tools we would not have been able to reach the scale and maybe the flexibility that we needed. As another example, there was nothing like **PhEDEx** at the level of

dealing with datasets (in the experiment-specific way an experiment defines one) which was something we really needed, and this is also a valid reason in itself. Another example is the **checksum**. This is a functionality we would have expected in the underlying system (i.e. SRM or FTS on the fly) to guarantee consistency but we had to put it in outside in order to have it done: having it done in the underlying system would better than how we are doing it now, still, so this is another functionality we would welcome from the underlying system itself (e.g. we are doing this only at the T1 level, for example).

Q: " *If the middleware dropped complexity and removed functionality, does the experiment have the resources to adopt to the change?* "

Yes, we are interfaced to the middleware in clean ways. Because of the way we interface to data management services, we will probably have a clean change (take e.g. the PhEDEx integration with gLite FTS, instead of other low-level file transfer tools). It must be noted that a question was asked as of why this Q is actually asked :) A discussion on the needs for simplification and low-impact reduction of functionalities followed.

Q: " *What is the experiment's interest in revising data placement models? What kinds of revisions have gone through? What is the experiment's interest in data federations?* "

We are interested in exploring more dynamic data placement systems, as well as data access over the WAN with an increasing scale. On the latter, we started with a simple fail-over model. We have done a little bit about moving into intentionally allocating remote resources with dynamic balancing (i.e. access over the network remotely to look for a file that cannot be accessed locally for any reason), and potentially opportunistic use for smaller sites. We have a roll-out plan, which is intended to be slow. We rely quite a bit on pre-placement for scalability, but also on WAN access to get the ability to gain flexibility as needed and under well-defined circumstances.

Q: " *What is the experiment's interest in data federations? What sort of assumptions does the experiment need to make for data federations work?* "

Not overloading the network of data servers. Also, making sure that there is a global optimization, i.e. 90% PhEDEx and 10% WAN access overall, and not a 50-50. The WAN access solutions may just not work on a scale of the majority access, nut it's fine as a minority solution for well-defined use-cases.

Q: " *Could you work directly with clustered file systems at smaller sites?* "

Yes, this has been demonstrated with Lustre over the WAN.

Q: " *Could you work directly with cloud file systems (i.e., GET/PUT)? Assume "cloud file systems" implies REST-like APIs, transfer via HTTP, no random access within files, no third-party transfer, and possibly no file/directory structure. See Amazon S3 for inspiration.* "

We could work this way for some workflows (any workflow where we could download the entire file locally to the WN over a simple interface) but it might be not suitable for all workflows (any workflow where one just needs to read a small part of each file will be very inefficient).

Q: " _Which components failed to deliver (larger) functionality needed?Does that mean we can depr

ecate ar

Q: " *How thoroughly does the experiment use space management today?* "

Very coarsely.

- In terms of file families, we tend to have reasonably fine-grained file families that are mapped to the namespace.
- In terms of space tokens, we can use it only on sites that need it for the mass storage, otherwise not much.
- For places like the "unmerged" storage areas, we do not apply any concept of space management: we should have people to clean up manually after given amount of time
- If "space management" involves e.g. the distinction among custodial and not custodial data samples, the use we make today of such concept may change and be revised soon, since we are aware that currently (actually for good reasons and choices by the sites) it does not necessarily map "custodial" to MSS resources and "non custodial" to disk resources respectively
- If "space management", instead, is intended in a slightly more general way (higher level), Victor and data placement can be quoted as a way we have to manage the storage space

Note that some of the aforementioned items (at least partially) overlap with activities in the Operations TEG.

Q: " *Does your framework support HTTP-based access?* "

It supports any access that can be read from ROOT, which would include http.

Q: " *Is the archive / disk split an agreed-upon strategy in your experiment? Can HSM be dropped?* "

HSM can t be stopped, but we have a strategy toward a more managed split. Currently, there is a lot of funtionalities e.g. the ones connected to the concept of "recovering" files in a structured way.

Q: " *What role do you see for Data Federation versus Data Push? Is this useful at smaller sites?* "

Data Federation might be easier at smaller sites, both techniques will probably be used.

Q: " *For smaller sites, would caching work for your experiment?* "

Yes, technically it should be possible, but there are some difficulties there. We can consider anyway to think of a system in which we can deal with file loss more like we operate in a T3 than we operate in a T1.

-- DanieleBonacorsi - Nov-2011
