Joint DM-SM TEG workshop

[Day 1 - January 24th - Notes by *D. Bonacorsi* - v.1.0 - last update February 2nd]

Data Placement (Dirk)

Main Agenda: https://indico.cern.ch/conferenceDisplay.py?confId=165687
This talk: https://indico.cern.ch/getFile.py/access?sessionId=1&resId=1&materialId=0&confId=165687

Note: in the following "*{C,Q}/**Name***" refers to a *Comment* or *Question* by *Name*. In case the person who gave the {C/Q} is not clearly identifiable as from my notes, the field *Name* is set to *Unknown*.

Status of the POOL project

This is discussed in this section since it does not fit into any other one. It's quite self-contained. One slide by Dirk. Input by Andrea Valassi.
CMS and LHCb stopped using POOL. ATLAS has produced a statement about the features of POOL, that will go into the document. As from Dirk's slide 2: "*ATLAS will continue to need support for POOL, including any relevant software patches and releases, for as long as the 2012 production version of the ATLAS software is actively used. It is based on the LCG61 configuration, which includes a version of POOL centrally built and maintained by the core team in IT-ES (using ROOT 5.30). ATLAS will no longer need support for POOL from IT-ES for their releases based on the LCG62 configuration (using ROOT 5.32), where a custom software package derived from POOL is built and maintained by ATLAS as part of their internal software. The first such release already exists and will be used as an ATLAS development release in 2012; this will eventually become the production version of the ATLAS software, by end 2012-beg 2013*".

Data Placement

Data has been placed ("pushed" to sites) since the start of WLCG.

Traditional transfer method implemented by FTS. All experiments are placing data - also ALICE is placing/ pushing data (even via xroot, but same concept)

Pre-placement. In the past experiments have decided to send a consistent set of data at a given site and then consume it on-site. Pre-placement has unbeatably low latency, but has constraints, e.g. 1) the availability of the local space; 2) (C by Paul) also the ability to state where the data is going to be needed. Connected to the last point, interestingly the experiments have recently changed the way they deal with the data movement process: efforts done mainly by ATLAS and CMS towards "data popularity" information to drive a more clever data placement/movement, i.e. predict or measure the popularity of data and trying to adjust the use of the available resources.

- *Discussion*. C/**Miron**: an important reason for federations or remote access is the error conditions, it's not a matter of popularity, you come to the site and made this constrained assumption that unless the data is there you cannot run, and one file or one byte is missing and... this is missing in the overall design, the errors in the early assumptions have been addressed and this must propagate to the entire system, because you can never assume that you know what you are going to fail once you are going after the bite, and the question is how does all this propagates back, this is a fundamental thing we are missing in our entire software data stack, how do we deal when we want to access, and the data is not there. C/**Dirk**: we will come back to it during the federation discussion. We will continue to do data placement, we will not do everything by federations, but the latter concept is being widely adopted already by 3 out of 4 experiments to complement. C/**Philippe**: need to define what you call by "complement" here, i.e. for which purpose you complement: if it is for fail-over, you can add LHCb (i.e. all 4 experiments then), instead if you plan to use it to be able to submit jobs everywhere and the jobs will pull the data around without any control, then certainly LHCb does not agree. C/**Dirk**: agreed, we need to define what problem the federation approach is intended to solve. For data for which we exactly know how they will be used, pre-placement is a brilliant way of doing it because it has no latencies and it's easier to do. But, there's the errors case of Miron and other cases which need to be treated differently. C/**Wahid**: can we quantify how often the error code happens? E.g. file corruption occurrencies. C/**Philippe**: also disk-servers not available. C/**Ian**: corrupted files are very rare, unavailable files is measured in few %, at some place between 2 and 5%.

Dirk discussed the "data placement responsibilities" point, and the split among data management and storage management. According to how the system was working until now, Dirk underlines that the experiments were placing data and therefore they were doing "data management" by defining geographical data(-sets) distribution within available resources according to current priorities, i.e. deciding where should datasets (and not single files) be available within some priorities envelopes. On the sites side, the sites provide and maintain stable storage for data placed by the experiments. Dirk takes the split among the two TEG working groups (DM and SM) as the first guess on the distinction among data management (experiment datasets centric) and storage management (service centric), as outlined above. If you look deeper into this and what actually happened, the sites are supposed to provide the files that the experiments needs, but experiments regularly happen to take more and more responsibilities in the repair tasks, and frequently take over. The reasons for this are various: experiments have easier way of doing it, of checking whether a site is consistent with the experiment expectation of what's at the site. very often the site cannot check better than the experiment can. So we need to progress on the definition of the responsibilities split among experiments and sites.

- *Discussion*. C/**Miron**: it's important to decide what the sites provide. Does the site provide storage space? or a guarantee that if you put the data on-site the data will remain on-site until they are removed? (which is a very different statement). The item goes back to the allocation of resources. If a site just provides the space, and a disk burns out, the site would provide a brand new disk back, not the data(-sets) back, and it's an experiment responsibility to fill it up again. C/**Dirk**: agreed, this needs to be defined in a layered model that describes the responsibilities today and the modifications triggered in the next future from recent developments. E.g. statement by Ian at the Ops TEG meeting (Monday 23th) about experiments to rely less on the storage side: it probably does not mean that the sites should not look after the data any amore, but it could mean that there is a change in the responsibilities the experiments think can be realistically taken over by the sites, and if that would be better defined it would help e.g. the whole sites community to do the right thing and not compete in areas where they may not undertake repair tasks because they do not have the needed knowledge. C/**Brian**: there had been a shift in the last couple of years away from the idea that once one puts a byte on a site storage I just can get it back and if it's not possible I file a ticket. This assumption has changed a bit over recent times, and a file loss has become a normal and acceptable event. C/**Dirk**: we have never defined this, and we probably should now. Some experiments have probably seen so often that one can doubt the service offered by the site was to keep the byte forever; on the other hand, some experiments may still think that they should just file a ticket when it happens. C/**Miron**: see yesterday's discussion. We need to think about it in a broader context, i.e. not just the LHC experiments. What does it mean that a VO comes to your site and park data there? what's the semantics of it? Is it just disk space? If you loose the disk how quickly would you replace it? With CPUS we know this, anything can happen and we can restart the job. C/**Ian**: there was something agreed upon, but only at the T1 level: at the T1s there was an agreement for the % of files that could be lost in a given year (10^6). C/**Miron**: lost from archive, but not from caching for analysis. C/**Philippe**: T0D1 is different from caching. C/**Ian**: this was in the CMS Computing TDR for the T1s, I am assuming there as an equivalent statement in the ATLAS TDR? C/**Jeff**: I do not remember any such statement (in the ATLAS TDR). C/**Philippe**: the sites are also here to support the access to the data; providing storage space is one thing, but providing bandwidth from the storage to the WNs is absolutely mandatory. We have a drawback in the WLCG, since the storage requirements are expressed in terms TBs/PBs but there is nothing about the access pattern, and how many jobs can access the very same file or the same disk-server at the same time. C/**Dirk**: agreed, there is no other area in which this happens, e.g. on the DB side you never discuss volumes without discussing IO. C/**Philippe**: exactly. The amount of data you put on a single disk-server has increased of a factor of ~10 in the past ~3 years, and still there is a 1 Gbit connection between storage and the WNs, and the number of spindles you have to serve your job is certainly not enough. C/**Miron**: agreed, but not only for local access: the bandwidth is broader than the job itself if you are getting remote access. C/**Philippe**: but already if it does not manage to provide local access to your jobs you are already in a bad situation. C/**Dirk**: this comes from a premature conversion of "$" figures to "PB" figures, because all the discussion about pledges should be in $ but for some reason they convert it into PB beforehand, so this kind of split into PB volumes or bandwidth to read them is lost.

But here we can make a statement here that this kind of conversion of budget into PB has an unclear split into how much of this goes into I/O access bandwidth and external network bandwidth (manpower as well, btw), but not much more. C/**Michel**: but the experiments talk about PBs and not bandwidth. C/**Philippe**: not true, it's the boards that do not take this into account. C/**Ian**: the information you provide have only one number, but the information we provide have by far more than that. And it's actually done by core, i.e. the expected average input per core. C/**Michel**: but when it's summarized that information is lost. C/**Philippe**: this is a message we should pass to the relevant committees, this is not a single number and we need to have these numbers and we need the sites to react. C/**Vladimir**: for INFN T1 I confirm we need more than a simple storage space figure. C/**Dirk**: we agree there is a problem here. C/**Unknown**: part of the reason why ATLAS has started to figure out popularity is that previous placement model were so brute force, we were busy in terms of network usage compared to actual reading, so it seems to me that smart placement needs to be taken into account. Placement has an effect in terms of external networking usage. C/**Brian**: we say data placement is done according to "current priorities" and one needs to accept the fact that current priority is not something that changes once per year. C/**Paul**: one last thing, there is a functionality mismatch here. One thing coming in particular from ATLAS is the idea that storage size is where I put files and datasets is where experiments work with. There is an impact of this, we are getting larger disk servers and one of the thing we can do is to try to spread the access pattern over multiple disk-servers to increase the overall bandwidth which you need to provide. So, there is a bandwidth-valuable access that would help, e.g. if the sites knew more of the datasets. C/**Dirk**: if a site does not know what a dataset is, this cannot be a responsibility of the site, also if the site does not know how to make a dataset complete it cannot do the repair. Depending on how exactly you define the info that is known to the site you also define the responsibility accordingly, otherwise that's impossible for them to take actions.

Dirk underlines that the data placement on disk (e.g. for t1s) used to be a SM decision (at least it was optimized by the SE software), whereas today it's a conscious DM (experiment) decision. Rephrasing: the T1s are placing data on disk, by this meaning "this data should be online" but is it really a DM decision if you leave it on a system that does not guarantee that the data is online? The HSM, e.g. speaking for Castor, does not guarantee this. Admittedly a bit provocative, but: if experiments place data on disk and expect to find them when they come back to read it, then they should not use HSM, or disable that feature; if experiments think being the data "online" or not is a SM decision, then you have to live with the consequences. Experiments more and more place data on disk and expect to find it there.

- *Discussion*. C/**Vladimir**: it depends on the SRM solution and implementation. C/**Dirk**: it mostly depend on the pinning implementation. It is really a question to the experiments: if the experiments see this as a DM decision (their decision) then we have to make to make sure that the storage system enforces this. C/**Miron**: it may take 30 secs or 30 years, unless the whole system is not designed under the assumption that something will happen, but it only takes into account what do we do if that does not happen, we would be in a much better shape. Our systems assume we write to an infinite disk, with no a-priori space checks, for example. C/**Dirk**: there is some planning by the experiments. C/**Miron**: ok, only some, though. When you have a storage hierarchy, when you have a fault you need to identify the fault and put in something that deals with it. C/**Mario**: ATLAS answers are: yes, we want to tell you put this data on disk, and no, we don't care if it's hierarchical or not.

Dirk underlines that the split among SM and DM is actually a still rather minimal model. It's useful if we manage to produce "the diagram", this will give us the split among components and people responsibilities. It would also address interface problems. E.g. the existence of dark data can make it hard for sites to leave experiments full DM responsibilities without getting stuck (disk full). And single sites cannot easily check consistency of data file content, consistency with non-local catalogues. Also, what are the "data/storage repair" responsibilities/agreements between site and experiments?

So, we need to work on "the diagram". Dirk asks: "Where is the rest of the model?". If there wasn't a concrete one when the current system was designed, can we produce a strawman model for what we think is in place now? This should cover:
1. Relationship between DM, transfer components, SM, storage components;
2. Implied responsibility split between experiments & sites;
3. Areas where joint operation is required and how the s/w supports the organization of these joint tasks
And - also - it should account for how recent changes are "changing the model" itself.