ARCHITECTURE Of SMALL COMPUTING CLUSTERS IN HIGH ENERGY PHYSICS

A.N. Lodkin, A.A. Oreshkin, A.Y. Shevel, T.S. Serebrova

1. Introduction

In last years we see that large **High Energy Physics (HEP)** collaborations use high performance computing clusters. Most of the clusters are integrated into World Grid infrastructure. At the same time, we observe that small clusters do continue to play their own role in the computing life. We mean ***small*** for the cluster up to about 50 machines or so. It is quite obvious that such small cluster is supposed to serve relatively small physics group, may be about 10 physicists. A good time is now to discuss a realistic scenario how physics data might be processed and analyzed and what computing architecture might be used by small physicist's teams.

Contemporary architecture of small computing clusters in **HEP** for physics analysis is based on the commodity hardware. In most of cases it is a Personal Computer (**PC)** based on Intel compliant microprocessors. Scientific Linux is the main operating system. It is better when the computing cluster is a member of a Grid Virtual Organization(s). Here we plan to discuss several aspects of computing cluster design and implementation and how small clusters are related to large computing installations.

2. Large clusters

Several important clusters of Tier 1 & Tier 2 are mentioned in Table 1 (N/A means that the information is *Not Available*). Each cluster of level Tier 1 has tens of **FTEs** in the staff. Most of such clusters have 10 Gbit or more external connectivity. Most of Tier 2 clusters have external channels in between 1 Gbit and 10 Gbit. There are expectations that in 2010 many Tier 1 clusters will have 1 Tbit channels.Each cluster uses a batch system – usually one or two from the range: **LSF**, **Condor**, **Torque**/**PBS**, **SGE**.

Table

Number of hosts and data volumes on several computing clusters in HEP (info from HEPiX October 2006)

Laboratory	Facility type	Number of hosts in clusters	Data volume on disks	Data volume in Mass Storage
BNL	Tier 1	~2.0 K	~400 TB	~4.0 PB
Canadian GridX	Tier 1	~1.2 K	~100 TB	~0.4 PB
FNAL Grid Computing Center	Tier 1	~3.0 K	~700 TB	~4.0 PB
GridPP/RAL	Tier 1	~2.9 K	~168 TB	N/A
NIKHEF	Tier 1	~0.3 K	~ 70 TB	N/A
SLAC	Tier 2	~1.7 K	~755 TB	N/A
GRIF(France)	Tier 2	~1.2 K	~700 TB	N/A
INFN	Tier 1	~1.3 K	~600 TB	N/A

Of course clusters are included into one or several Grid Virtual Organizations (**VO**). One of the consequences is that the bulk data moving over World Area Network (**WAN)** must be planned and performed with Grid tools. Each VO has its own rules on the data moving. There is special person (manager) responsible for data moving process. At the same time if the bulk data moving is started it is difficult to predict when the operation would be accomplished. Due to this fact in many large clusters the user jobs are planned to be cancelled if the data requested by jobs are not available in local disk space.

Apparently clusters Tier 1 (or even Tier 2) are main source of physics data and main repository for the software of almost any kind. Often it is required to create new data with special selection algorithm from other data before start an analysis. Because resulted data are peculiar or even private the small physicist's team needs to keep the data in own disk space which could be obtained on large cluster for relatively short period of time. In many cases a small computing cluster is best place to keep own data for long time.

When small physicist’s team plans to use existing large cluster it important to know not only power of the cluster but the administrative conditions on the cluster [1,2]. For example if any non privileged user can keep just 40 jobs in the run stage (permitted to use 40 CPUs), it does not matter for him that the cluster consists of 10K machines. Of course in Grid architecture someone can easily send jobs to another cluster where the data are, i.e. it possible to use more than one computing cluster. However in most of real situations some data moving and other additional operations are required. Real advantage to use more than one cluster can be gained if specific conditions are taken place [3]. In this discussion it is assumed that clusters are really stable. The stability in general and difficulty to predict future status of the Grid clusters is a hot topic for now. Anyway, there is an expected advantage with using of two computing clusters instead of one.

3. Estimates for accelerating of computing with two clusters

Let us imagine two clusters where we can do some data analysis or data simulation. For example one cluster is in one research institute and another cluster is in another institute or university, the clusters have independent administration. For simplicity we will use terms *local cluster* and *remote cluster*. We consider the ***computation*** as bunch of simulation (data generation) jobs or data analysis jobs which could be performed on any of two clusters. Usually the bunch of jobs is performing many hours or many days. All jobs are considered as accomplished when the computing itself is ended and the data are moved to final (target) place, for instance to small computing cluster.

Let us introduce several variables:

T – the time for computing task with using one (local) cluster;

τ – the time for computing with two clusters;

t_l – average time for processing of one portion of the data on one (local) cluster;

t_r – average time for processing of one portion of the data on one (remote) cluster;

t_o – the average overhead time which is required to process one portion of the data on remote cluster; we can include in this time anything we need to make computation possible in remote cluster, for example, time to transfer the data to (from) remote cluster;

D – total number of the data portions which have to be processed;

S – the number of data portions which are processed per time unit;

α – acceleration of the computing due to use of additional cluster.

Now we can write for only (local) cluster $S = \frac{1}{t_l}$ and total time for data processing is

$$T = \frac{D}{S} = D \cdot t_l .$$

For two clusters (local and remote) we can write $S = \frac{1}{t_l} + \frac{1}{t_0 + t_r}$ and total time for the computing is

$$\tau = \frac{D}{\left(\frac{1}{t_l} + \frac{1}{t_0 + t_r}\right)} .$$

It is assumed that the jobs will be sent to remote cluster only when local cluster if fully loaded with our jobs. Now accelerating is

$$\alpha = \frac{T}{\tau} = t_l \cdot \left(\frac{1}{t_l} + \frac{1}{t_0 + t_r}\right) = 1 + \frac{t_l}{t_0 + t_r} .$$

Above might be rewritten as

$$\alpha = \frac{(t_0 / t_r + 1 + t_l / t_r)}{(t_0 / t_r + 1)} .$$

From above formula we see that the accelerating is quite sensitive to the overhead time $\boldsymbol{t_o}$ (please see in Fig. 1) Even in case when remote cluster has huge throughput, when value $\boldsymbol{t_r}$ can be considered close to 0, we have the acceleration is limited by $\boldsymbol{t_o}$.

It is possible to conclude:

- the expected acceleration for the computing with two clusters can be relatively easy estimated;
- the use of two computing clusters instead one cluster does not mean guaranteed decreasing the computation time in all cases, concrete overheads are very important;
- in long term plan (many days or weeks) the probability that the remote cluster is up and running properly is important.

If we plan to keep the data on small cluster apparently it is more effective to analyze the data on the same cluster.

All above reasons do lead to understanding that in a range of cases the use of small computing cluster is very helpful. Let us consider the architecture of such clusters.

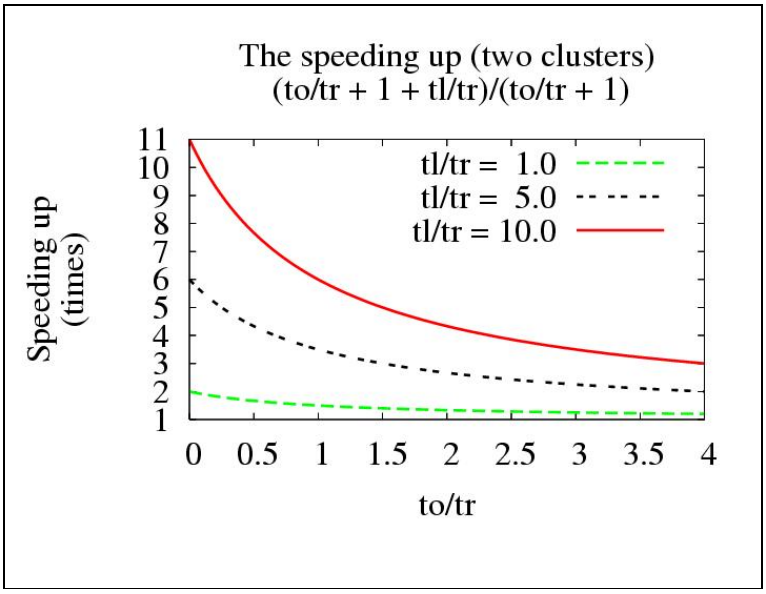

Fig. 1. Computing acceleration with two clusters with different conditions

4. Cluster architecture considerations

Last years **CPU** (microprocessor) performance is increasing at the rate around 60% per year. At the same time disk read/write speed is increasing about 10% per year. Also disk volume per spindle is increasing at the rate around 60% per year. It means that access to disk space (especially in **HEP**, due to extremely large volume of the data) is more and more often becomes a bottleneck. There are many reasons to think that such the relations in between mentioned values will be kept at least for several years in the future.

In the light of such facts and facing the need to analyze tremendous volume of the data the methods how to make read/write operations in parallel are most hot topic. One of the methods is to use advanced **RAID** controllers and advanced cluster file system, for example, **Lustre**, **PVFS**, **GFS**, **GPFS**, **Panasas**, **StorNext**, and the like.

Another very simple method was suggested in the presentation [4]. The approach assumes two opposite configurations and many in between. First is to use one central machine in the cluster where connected all **RAID**s. Opposite configuration is to use the disk space on each machine in the cluster as shown in Fig. 2. Here all file systems are mounted by **NFS** over separate network channel for each file system. More specialized systems like **xrootd** are addressing the same target: to make I/O operation in parallel. Another important point in small cluster is architectural features which permit to reduce the requirements of local maintenance to the minimum.

5. Remote maintenance for small computing cluster

When new cluster design is under consideration we need to take into account the trends with less available manpower in future years. In other words we have to plan as much as minimum local maintenance activity. To permit remote experts to do their job the cluster must be equipped with appropriate components.

Vital and stable solution may be implemented with using the special hardware. We mean type of devices so called **KVM** switch. **KVM** switch (or just **KVM**) is device which connects all control lines (keyboard, mouse, monitor) from each cluster node. **KVM** has connectors to connect real keyboard, monitor, and mouse. Also **KVM** has ability to connect (logically) real keyboard, mice, monitor to any desired server in the cluster.

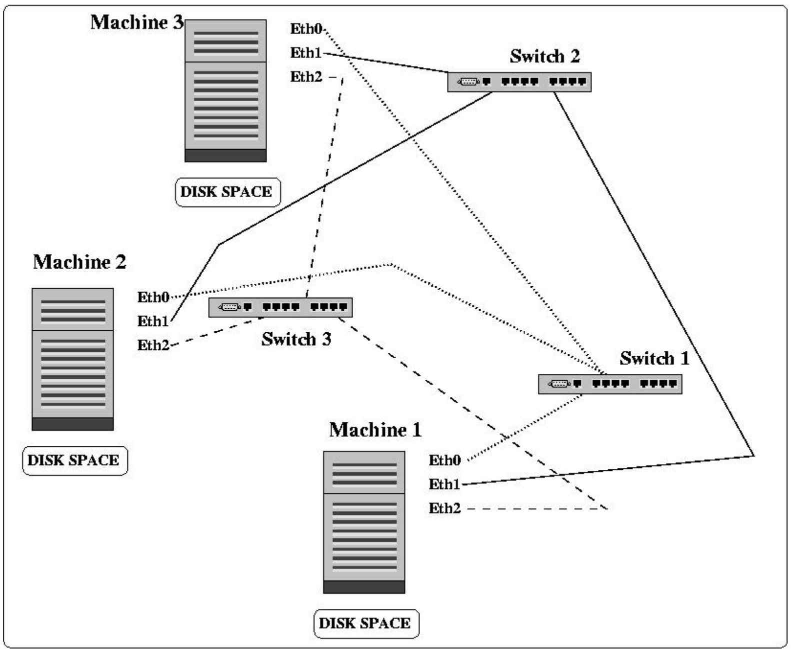

Fig. 2. Cluster scheme where each machine has own disk space

Another device entitled *remote IP console* (**RIPC)** is used as interface in between **KVM** and Internet. An expert on remote computer can use Web browser (with enabled Java functionality) to make connection to the device **RIPC** with protocol **https** and see the screen on **Java** applet. **Java** applet displays redirected console screen, keyboard, and mouse of the cluster where **RIPC** is attached. In other words the expert is able to use local for him keyboard, mouse as they would be attached to the cluster.

Among other mandatory components of the cluster we can mention **UPS** – to make electricity power more stable – and **PDU -** to have ability to switch electrical power **on** and **off** for specified machine in the cluster. An air conditioning is also in range of requirements.

The basic requirements were enumerated. Now other details concerning the software have to be discussed.

6. Technological cluster software and middleware

A range of technological services are seemed very important in contest of easier maintenance procedures. Usually it is very important to have reserve copy of critical data - backup service. Much better is to have reserve copy on the tape cartridges. The tape drive might be used for reserve copy for the physics data as well. Personal subsets of physics data copied on the tape cartridge are often great advantage for physicist.

All cluster users must be in cluster user mailing list. This list is to be used for the information about the changes on the cluster. Such the mailing list may be implemented in several places. One of the useful ways to keep mailing list for small physics team is site http://groups.google.com/.

Cluster administrative system is aimed to deploy, configure and support cluster (to keep all system parameters in consistent state) and is in active use on many clusters. Good examples of such the systems are **OSCAR**, **ROCKS**, and the like.

Batch processing on the cluster is usually performed with one of several batch systems: **Torque/PBS**, **Condor**, **Sun Grid Engine**, **LSF**. Fortunately batch systems in use on large clusters are pretty same as for small clusters. One of the specialized batch tools is **PROOF.** It is the system for users who does like to be all the time inside **ROOT** environment. Part of physicists do use **PROOF** for data analysis.

Monitoring systems are almost same as for large clusters, for instance **Ganglia** and **Nagios.**

Grid middleware is another large software component which is required if the cluster is included into one or more Grid Virtual Organizations. The part of middleware components are the same for all VOs (basic **Globus** toolkit and the like.). A good fraction of middleware is developed by concrete VO and it is not supposed to be used outside the VO.

7. Application software and related databases

Application software dedicated for concrete physics is large fraction of all software tools on the computing cluster. This software has as a rule many versions (may be several tens). Usually people keep all versions on the cluster. Such the software is kept on leading clusters (Tier 1) in **AFS** tree. In general it is possible to use it over **AFS,** however real experience shows that much better to keep local copy of the software. It is performed with one or another set of mirroring mechanisms for example once a week or so. It is quite safe for small physics team especially if some problems are appeared on leading clusters (**AFS** server is down, application software tree becomes unusable). Almost the same we can repeate about such databases as geometry/calibrations database, *etc*.

8. Small cluster support over years

The cluster is running, data analysis is in progress but time is going on and new versions of software (application, system, middleware) are appeared. Quite often you have to do upgrade the software to guarantee that you have same common software packages as your colleagues do. That means somebody has to be careful about consistency of the software.

Furthermore with the time (one, two, three years) you might (very probable) discover that some machines in the cluster are broken or out of date and do not meet newest requirements. Common rule is to remove from the cluster any machine which gives any kind of problems. In average if we like to keep the computing cluster abilities on top – we have to plan to change about 1/3 machines in the cluster every year. It means to remove old machine from cluster and add new machine instead. The machines retired from the cluster might be used as personal machines for students, physicists, *etc*. The reach source of useful information about computing clusters is available at the site http://hepd.pnpi.spb.ru/ClusterGate.RU.

9. Conclusion

Obviously it is not possible to use small computing clusters ***instead of*** large clusters of Tier 1 or 2. It is very useful as a complement to large computing installations; also they make use of large computing facility more optimal. Taking into account that prices for disks and CPUs are going down, it is clear that about 20–50 machines with 50–100 TB of disk space are foreseeable for small physicist’s team for most of analysis needs.

References

1. B. Jacak, R. Lacey, S. Mioduszewski, D. Morrison, A. Shevel and I. Sourikova, talk presented at *the Conference on Computing in High Energy Physics CHEP2003* (La Jolla, USA, 24 – 28 March 2003), presentation TUCT009 in http://www.slac.stanford.edu/econf/C0303241/proceedings.html
2. A. Shevel, B. Jacak, R. Lacey *et al.*, in *the Proceedings of the Conference on Ccomputing in High Energy Physics CHEP2004* (Interlaken, Switzerland, 24 September – 1 October 2004), Geneva, 2005. p. 974.
3. A. Shevel and R. Lacey, talk presented at *the Clobus World Conference* (Boston, USA, 7 – 11 February 2005).
4. A. Shevel and R. Lacey, poster presentation at *the Conference on Computing in High Energy Physics CHEP2006* (Mumbai, India, 13 – 17 February 2006).