ClusterAlive

G. Caruso, S. Arezzini, A. Ciampa, A. Formuso, E. Mazzoni
INFN Sezione di Pisa, Pisa (ITALY)

E-mail: giuseppe.caruso@pi.infn.it

Abstract. INFN-Pisa Scientific Computing Center is working from many years both in GRID and HPC computing. The monitoring and managing tools have been key components of the center's successful operation. The lessons learned from the use of standard tools, such as Ganglia, have been starting points for the development of new tools specific for our infrastructure. In this note we will illustrate the integration of many different monitoring tools in one single platform called ClusterAlive. Aim of ClusterAlive is to increase the HPC cluster performance and simplify maintenance operations, possibly in a proactive approach.

1. Context Introduction and Objectives
ClusterAlive is a tool developed in the INFN-Pisa Scientific Computing Center intended to perform checks and management tasks in the HPC environment. HPC environment at INFN-Pisa is made of about 5000 cores connected with QDR Infiniband and using two network filesystem (GPFS and AFS). These computing resources are organized in 4 different clusters according to the applications needs. The biggest (named Zefiro) is composed of 2048 cores dedicated to the INFN Theoretical Physics Community. It is a national cluster and is part of SUMA project [1] (SUper MAssive computing project founded by Italian Research and Education Ministry). There are three other clusters for Academic and Industrial collaboration dedicated to advanced simulations, CFD for automotive and mechanical design and specific regional projects with Small & Medium Enterprises.
ClusterAlive is made of shell scripts developed at INFN-Pisa with the aim of monitor and manage HPC clusters.
This tool has been developed to collect information from many sources (both ethernet and Infiniband):
- operating system
- batch scheduler
- network components
- storage
with a plugin approach. All the information are summarized in a dashboard for an immediate view of the cluster’s operating state. In this way the problems are easily brought to the attention of the operator that can deeply investigate with more specific tools.
Additionally ClusterAlive can perform automatic tasks. In order to restart services, adjust parameters, start basic activities both for computing nodes and infrastructure components.
2. Monitoring Tools
The key point of Computing Center management is the monitoring infrastructure. Monitoring systems must take care of:

- environment parameters such as air temperature and humidity
- infrastructure parameters such as power distribution and networks functionality
- server parameters such as node operational, CPU and memory usage
- computational infrastructure usage and accounting

In this chapter we will concentrate on the last two items: server parameters and computational usage.

2.1. Server parameters: GANGLIA

Ganglia [2] is a powerful and scalable distributed monitoring system. It is de facto standard for the monitoring of HPC system. Ganglia uses well consolidated technologies such as XML for data exchange or RRDtool for data archiving. The per-node computational overhead is very low so it has almost no impact on the clusters performances. For all these characteristics we are using it since many years [3]. The hierarchical organization of monitored data is very useful to spot a potential problem and trace it down to the single server.

Figure 1. GANGLIA – CPU, memory and network usage for the entire cluster
2.2. Computational Infrastructure: LSFmon

LSFmon [4] is a tool in house developed to monitor the usage of the LSF batch system and to keep accounting information. LSFmon is composed by a set of programs that collect information from LSF log files and LSF diagnostic tools. All the information collected are used to build web pages showing real-time status of the batch system and accounting information. LSFmon was developed in 2010 for the use in a CMS Tier2 so it was focused on serial jobs. In such framework the status of batch system from the user, queue or CPUs point of view are almost the same. There is a one to one correlation between job slot, users and queue for each submitted job. That is not the case when we consider parallel jobs. First of all a job can be composed by many job slots, moreover in some cluster the definition of job slot is not coincident with the CPU granularity of the systems. The difference between serial and parallel jobs has imposed a major rewrite of the code to have a double vision of the LSF status. The original view from user point of view has been complemented by a queue view. The accounting information are the same for both software versions. The following snapshots give an idea of the information available from LSFmon in a cluster environment. The two versions in production can viewed at the following URLs:

- LSFmon serial jobs: http://farmsmon.pi.infn.it/lsfmon/
- LSFmon HPC parallel jobs: http://farmsmon.pi.infn.it/lsfmonhpc/dev/

Figure 2. GANGLIA – CPU, memory and network usage for a single node
Figure 3. LSFmon – Realtime scheduler information from user point of view (CPU and job information)

Figure 4. LSFmon - Realtime scheduler information from queue point of view (CPU and job information)
3. ClusterAlive
Our experiences with HPC clusters operational suggest that Ganglia and LSFmon are not all the story. Ganglia gives accurate information at the level of single node. LSFmon gives very good information at the scheduler level. There are situation in which a cluster is in good operational condition for both Ganglia and LSFmon but actually it does not work. For example if:

- all the cluster nodes for Ganglia are up but the major part of them are without CPU load
- LSFmon indicates that all the job slots are available but there are many job long waiting in queue

Such events combination could indicate a problem at node level that prevent the job starting. Also the opposite situation with high CPU load (Ganglia) and no running jobs (LSFmon) could indicate a problem. In these case a manual intervention is needed to discover and fix the problem. The lesson learned from these two simple example is:

- an automatic procedure to correlate information from different monitoring systems is useful for the HPC cluster operation
- a system to automatically fix the problems it is also useful

The ClusterAlive project is our response. It is both a monitoring and managing system. From our past experience the most critical things to be controlled for the cluster operation are:

- cluster filesystem
- user authentication and authorization mechanism
- node login mechanism
- high performances network operation
- scheduler information at node level
The entire ClusterAlive code is based on shell scripts. In this way it is very simple to adapt to different cluster's environment. ClusterAlive also provides a simple and intuitive web interface that gives a summary of the collected data and node operational state.

3.1. ClusterAlive monitoring

In our case each cluster is composed by a master node (MN) and a set of worker nodes (WN). MN has no computational role only the WNs take part in the computational jobs. The monitoring part of ClusterAlive is made of two components:

- a control script periodically running on the MN. It creates the list of WN alive in that moment and then starts the checking operation on every WN
- a checking script installed on every WN. This script is activated by the control script mentioned above. It collects all the information necessary for the monitoring and put the results in a file on a common AFS area

The web interface collects the information from the file produced by each WN and creates a synoptic view of cluster parameters.

3.2. ClusterAlive management

The management part of ClusterAlive is conceptually identical to the monitoring but in this case the WN script tries to fix the problem. For example if it detects that ssh daemon is not working it tries a restarting procedure. If the script fails to fix the problem after three attempts it will close the node in LSF to avoid problems for the user jobs. If the problem is not easily solvable but needs a human intervention, for example in case of file system failures, the node is only closed. The monitoring part of ClusterAlive catch this condition of “subsystems not properly working and node closed”.

Figure 6. ClusterAlive – Node status and users job informations on the same web page
4. Next steps
ClusterAlive was originally designed and developed to equip our main cluster Zefiro. All the features described in this article are in production on Zefiro.
Future plans for ClusterAlive are:

- install it on all HPC Cluster of INFN-Pisa data center
- integrate ClusterAlive with the Infiniband switch native monitoring system
- extend ClusterAlive monitoring feature to integrate GPU components that are already present in our data center
- extend ClusterAlive monitoring to other filesystem such as AFS.

5. References
[1] SUMA Project: https://web2.infn.it/SUMA/
[2] http://ganglia.sourceforge.net/
[3] http://farmsmon.pi.infn.it/ganglia-web
[4] Subir Sarkar and Sonia Taneja 2011 A Job Monitoring and Accounting Tool for the LSF Batch System (J. Phys.: Conf. Ser. 331 072064)