# Metacluster System for Managing the HPC Integrated Environment

Victor Gergel and Andrew Senin

N.I. Lobachevsky State University of Nizhni Novgorod, Russia
SeninAndrew@gmail.com

**Abstract.** Clusters became the de-facto standard in modern high-performance computing. At present it is rather often when a single organization has a few clusters and wants to connect them into a multicluster to benefit from reduced task waiting time and increased total available processing power. This paper studies one of possible approaches to the problem of uniting the computing resources which addresses exactly the case of owning clusters by a single proprietor. Such approach was implemented in Metacluster system of Nizhni Novgorod State University, Russia. Current state of Metacluster was reviewed. Key features, component-based architecture and main functions implementation details were described.

**Keywords:** Metacluster, cluster management system, tasks scheduling, computing resources monitoring, Microsoft High Performance Computing Server 2008.

## 1 Introduction

Cluster technologies play a leading role in high performance computing world ([1]). According to the current (as of November, 2009) TOP 500 list of supercomputing sites clusters occupy 417 positions which are 83% of installations ([8]). In most cases clusters offer an optimal ration of price/performance for solving a wide range of computing insensitive tasks such as finance math, physical modeling, new medicine discovering etc. Cluster systems are easy to upgrade, there are a lot of off-the-shelf hardware components available and administration can be done by technicians who do not have deep knowledge in the field of high performance computing. These factors have led to a massive spread of clusters in the scientific community, in industry, and even individual researchers often can afford to purchase or build their own a small cluster.

Increase in the number of cluster systems makes attractive the idea of clusters sharing to increase the total available processing power, as well as to reduce the average waiting time of tasks to start. The latter is achieved through the use of advanced scheduling strategies that take into account temporary fluctuations in workload of various clusters (if one of the clusters is idle then tasks from the most loaded cluster can be transferred to the idle one) and at the same time flexibility in allocation of resource (for example, tasks of cluster owners might have higher priority etc.).

C.H. Hsu and V. Malyshkin (Eds.): MTPP 2010, LNCS 6083, pp. 86–94, 2010.

The idea of clusters sharing is reflected in the concept of grid ([2]). There are currently a few popular technologies available which allow combining computing resources into the grid: Globus Toolkit, gLite, Unicore etc. Some of grid systems combine clusters located on different continents (eg. World Community Grid, [9]), consist of thousands of organizations and millions of computing nodes. But often there is a need to unite computing resources which belong to a single organization: a few clusters or labs of workstations. If these resources are physically close to each other, one can setup a dedicated connection between them with characteristics similar to network within a cluster. This allows the computing resources owner to expect from the derived *multicluster* to be more effective than when combining similar resources on the Internet. For solving the problem above one or another grid implementation can be used. But such solution will be paid by unnecessary complexity of administration and inconveniences for end-users. The concept of grid does not imply the availability of centralized management, which is natural in case the computing resources are owned by different organizations. But in the case of a single organization, this approach introduces additional cost and complicates tasks scheduling.

An alternative approach in creating a multicluster is to centralize the management of all connected computing resources. Such an approach may be inapplicable when managing thousands of resources all around the world. But in the case all clusters that make up a multicluster are owned by a single organization centralized management can be more effective and natural because it reflects the fact of owning the resources by a single proprietor. The high performance computing environment management system Metacluster developed at the University of Nizhni Novgorod is based on such principle. The main purpose of the system is more effective use of computing resources through load balancing between clusters, and effective scheduling strategies within each cluster.

## 2 Metacluster Overview

There were three consistently developed version of Metacluster during its project history. The first version of was Metacluster developed in 2002 and was focused on the organization of effective management of individual clusters while ensuring high reliability and fault tolerance when providing remote access to the cluster via the Internet. An important feature of the developed system was its installation on clusters operated under the family of operating systems Microsoft Windows. Selection of the operating system was due to the desire to simplify the problem of practical usage of high performance cluster systems by end-users – many application developers have more experience in Windows environment and their development of programs for Linux-based clusters could be quite complex. Besides, a large amount of computing resources which can be reused to build clusters (labs of workstations, student's terminals), as a rule, is running Windows. And finally, at the beginning of Metacluster development there were a rather limited number of affordable, reliable and easy to use cluster management systems working with Windows as practice of high performance computing on Windows had not have wide distribution. As shown by subsequent developments, the choice of Windows was justified, because it attracted to the subject

of high performance computing a wide range of users. At present, most of the leading software manufacturers for cluster management also support Windows (Platform LSF, Condor, Microsoft HPC 2008 and others).

The next version of Metacluster was presented in 2005 and was aimed at supporting of multiclustering – the ability to manage simultaneously multiple clusters providing a single access point to all connected computing resources. This has opened great opportunities for users, as they can run the job on a wide range of computing resources united in a multicluster (or so-called integrated high performance computing environment) as well as to improve the overall efficiency of clusters due to dynamic load balancing between them.

The newest version of Metacluster was developed in 2008 with support of Linux clusters as well as integration with third-party cluster management systems including Microsoft High Performance Computing Server 2008.

The key features of Metacluster are the following: multicluster management, ability to work on different operating systems, integration with third-party cluster management systems. Consider each of the characteristics in detail.

### 2.1 Multicluster Management

Metacluster allows connecting to the integrated high performance computing environment an arbitrary number of clusters, possibly remote and not in the same network. The system takes over the problem of delivering necessary input files to the required cluster and to copy results of calculations on a virtual user desktop. This allows users not to think about where the tasks physically are being performed. In case of temporary unavailability of a particular cluster, as well as in the case of new computing resources being connected to Metacluster there is nothing changes for users: the system finds out itself which resource it is more effective to resend the tasks to. All the user needs to do is to specify requirements for a task: architecture and a number of processors, RAM, free space on hard disks and other characteristics. If necessary it is possible to explicitly specify a cluster and later its nodes. But in this case a user will have to restart the tasks by him-/herself in case the selected resources are disconnected.

### 2.2 Multiplatform Cluster Integration

Metacluster enables integration of clusters running different operating systems. This is achieved through specific implementation of platform-specific code and the use of open protocols for component communications. Currently major versions of operating systems Windows and Linux are supported. When formulating a problem user selects a target operating system and implementation of communication mechanism (for example a specific implementation of MPI) required by the application. The task of Metacluster is to choose appropriate resources and to launch the application in a specific for selected operating system and MPI implementation manner.

Flexibility in selecting of operating system allows extending the range of Metacluster users and spectrum of applications. However Metacluster partially retains its focus on work in Windows. Thus, the integrator (distribution of tasks across clusters) and the remote access component work only in Windows.

### 2.3 Cooperation with Third-Party Cluster Management Systems

Metacluster can interact with cluster management systems of third party developers. When connecting of such a cluster Metacluster takes over work on the interaction with the cluster management system: adding of tasks, tracking task statuses, monitoring of computing resources, etc. This allows connecting of clusters to the integrated high performance computing environment management system without disturbing existing users, who can continue operating in the familiar way. Currently integration with Microsoft High Performance Computing Server 2008 is implemented.

## 3 Metacluster Architecture

There are four main components of Metacluster: remote access manager, integrator of clusters, manager of a cluster and inspector of a node.

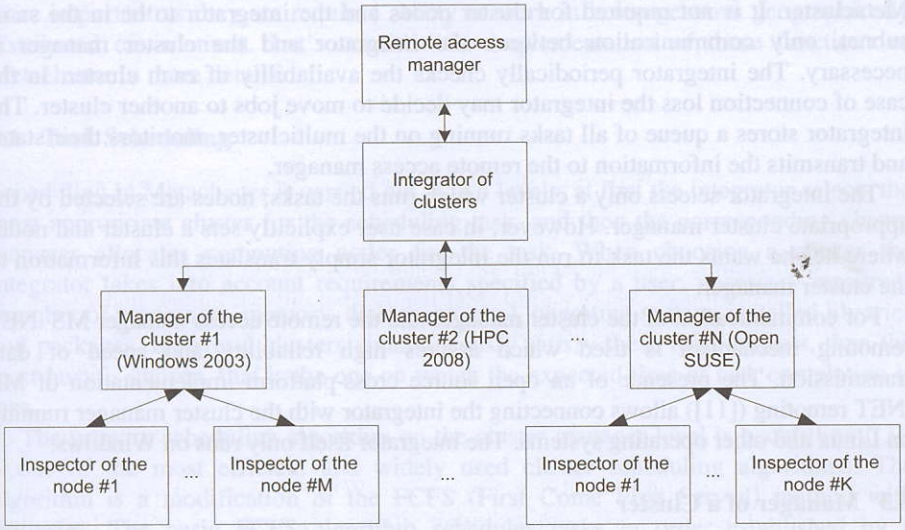


**Fig. 1.** Architecture of Metacluster

### 3.1 Remote Access Manager

The remote access manager is a single access point for all the resources under control of Metacluster. The access is possible either through a Web service ([10]) or through web interface. The web service allows integrating Metacluster with custom user applications that need to use high performance computing resources. The open standards the web service is based to (XML, WSDL, SOAP) help to interact with Metacluster using virtually any programming language and any operating system.

Metacluster web interface provides convenient and easy access to computing resources via the Internet and does not require installing any special software. After the authentication procedure a Metacluster user gets access to his/her a virtual desktop

with uploaded applications and input files as well as results of previous calculations. The virtual desktop supports all basic file management operations: creating of directories, moving/ coping, renaming, zipping/ unzipping, loading/ downloading of files, etc. To add a new task a Metacluster user selects an executable file and sets properties of the computing problem: required computing resources, input parameters, standard output file, task priority, etc. After adding the task its state is shown in the task queue. The web interface can also be used to draw different statistical graphics and reports on the cluster usage during selected periods of time. Almost all of these operations can be performed from a user's program through the utilization of web services - in fact, the web interface is just one of the web service clients.

### 3.2 Integrator of Clusters

The integrator of clusters is the central point of multicluster control. The integrator receives tasks from the remote access manage, distributes them among the clusters and monitors task states. The integrator liaises with each of the clusters included in Metacluster. It is not required for cluster nodes and the integrator to be in the same subnet, only communication between the integrator and the cluster manager is necessary. The integrator periodically checks the availability of each cluster. In the case of connection loss the integrator may decide to move jobs to another cluster. The integrator stores a queue of all tasks running on the multicluster, monitors their states and transmits the information to the remote access manager.

The integrator selects only a cluster which runs the tasks; nodes are selected by the appropriate cluster manager. However, in case user explicitly sets a cluster and nodes where he/she wants the task to run the integrator simply translates this information to the cluster manager.

For communication of the cluster manager and the remote access manager MS .NET remoting mechanism is used which assures high reliability and speed of data transmission. The presence of an open source cross-platform implementation of MS .NET remoting ([11]) allows connecting the integrator with the cluster manager running on Linux and other operating systems. The integrator itself only runs on Windows.

### 3.3 Manager of a Cluster

The cluster manager component is a cluster management system for a single cluster. Job of the component is to distribute tasks on the cluster nodes in accordance with local planning strategy, to monitor of cluster computing resources and to provide fault tolerance of individual sites. The cluster manager has a local task queue of tasks from the integrator and tasks from local users. The cluster manager tracks states of running tasks and transfers this information to the integrator.

A specific cluster manager implementation can be used to connect clusters under control of different operating systems to Metacluster as well as clusters with third-party cluster management systems. In the latest case the cluster manager acts as a communicator of the integrator and the cluster management system by converting integrator command into the local cluster management system commands. Currently Windows and Linux native cluster managers are implemented and a communicator for Microsoft High Performance Computing Server 2008.

### 3.4 Inspector of a Node

The inspector of nodes component must be installed on all cluster nodes. Its job is to execute cluster manager command on remote nodes. The commands may include the following: start/stop of processes, restart of hung services, providing information of node workload and state as well as states of separate processes of interest. For communication with the cluster manager MS .NET remoting mechanism is used.

## 4 Metacluster Implementation Features

One of the most important challenges of cluster management is to ensure the efficient use of available computing resources. Only in this case the maximum possible workload of a computer system can be achieved, which, in turn, will minimize time of tasks execution. Thus, the scheduling and monitoring subsystems are two of the most important functions of cluster management. In case of a multicluster they are even more important as they are required ability to work in heterogeneous geographically distributed environment. Let's consider the implementation of these functions of Metacluster in more details.

### 4.1 Task Scheduling

Scheduling in Metacluster is carried out at two levels: at first the integrator selects the most appropriate cluster for the scheduling task, and then the corresponding cluster manager allocates computing nodes for the task. When choosing a cluster the integrator takes into account requirements specified by a user: necessary resources (number of processors, memory, disk space, etc.), operating system, installed libraries and packages. If several clusters simultaneously satisfy these requirements, then the least busy is chosen, that is the one on which the expected time of task completion is less.

The primary scheduling algorithm on the cluster manager level is backfilling ([3]) – one of the most efficient and widely used cluster scheduling algorithms. The algorithm is a modification of the FCFS (First Come First Served) method with priorities. The basic FCFS algorithm schedules tasks in order established by a weighted sum of several parameters: task waiting time, task priority, user priority etc. For the backfilling algorithm to work it is necessary to set up its expected execution time. With using this information the scheduler can plan less priority but small tasks to be executed before high priority major tasks by utilizing free time slots in the schedule. Thereby the average tasks waiting time is reduced.

The cluster manager can be integrated with Maui – a popular task scheduler on cluster systems. Maui allows an administrator to configure advanced settings of scheduling policies. Integration with Maui was accomplished by exchange of network sockets messages in WIKI protocol ([12]). Details of integration are presented for example in [4].

### 4.2 Monitoring of Computing Resources

The main jobs of the Metacluster monitoring subsystem are the following:

- Providing up-to-date information of dynamic computing nodes metrics: CPU workload, amount of free RAM, amount of HDD free space etc.,
- Graphics with statistics of cluster workload in the Metacluster web interface.

The monitoring subsystem must be cross-platform and work stable on heterogeneous geographically distributed multicluster system. We found that the Ganglia distributed system for monitoring of high performance computing resources ([5]) best suits to the listed requirements. It consists of the following main components:

- Gmond is a service of node monitoring which must be installed on each cluster node,
- Gmetad is a service of statistics gathering which is installed on cluster head node to collect and store statistics into a cyclic database. In case of having multiple clusters the service can accumulate statistics from different gmetad services,
- Web interface is a visual part for graphic display of statistics.

Ganglia is open source software written in C programming language. The node monitoring service has been ported to a wide range of different operating systems, but the service of statistics gathering is intended for use only on Unix-like systems. In particular, according to the developers gmetad does not compile on Windows. Nevertheless, as a part of integration of Metacluster and Ganglia the Metacluster developing team has ported gmetad to Windows – see [6] for more details.

The Metacluster scheduler receives monitoring information from a specially developed library which uses network sockets to communicate with gmetad (format of messages is XML). The Ganglia web interface provides convenient and intuitive means of displaying information about the workload of the cluster. Ganglia Web interface was built into the statistics page of Metacluster web interface to display cluster workload for selected intervals.

### 4.3 Managing of Different Task Types

Metacluster does not restrict types of user applications. Those may be sequential programs, parallel programs written with using different implementations of MPI, tasks which utilize packages installed on a cluster. Such flexibility is achieved by moving parts of the launching logic into Perl scripts, editable by a cluster administrator. An administrator determines what types of tasks are allowed to run on a cluster, for example: sequential executables, scripts in Python, MPICH2 programs, OpenMPI programs, Fluent tasks ([13]) etc. For each type of task an administrator binds its handling script.

The handling scripts accept the following parameters:

- Module name. File selected in the Metacluster web interface to run. In case of running self-written application it may be an executable or a script. In case of utilizing a package installed on a cluster this may be a name of input file with model description and settings,
- Command line arguments,
- Standard output and standard error redirection,
- Working directory. Full path to a directory which corresponds to the virtual folder from which the user started his/her application,

- Environment variables set by a user in the web interface,
- Nodes allocated by the scheduler to run the task.

With using given values the script must create a full command string to run the task. For example in case of MPICH2 application it may be the following command:

```
<MPI install path>\mpiexec -hosts <list of nodes> -env
<environment variable> <module name> <command line
arguments> > <output filename> (some parameters were
omitted for clarty).
```

In addition of forming a command line a script may execute additional service operations in order to prepare the task to run: checking correctness of passed parameters, preparing necessary environment, registering of environment variables, writing messages to a log file etc. An executable file which is actually started as result of executing the script command must be running until the task finishes. Otherwise Metacluster considers this as a completion of the task and starts the release of resources procedure which closes a whole tree of running processes started by the task.

## 5 Conclusions

The high performance computing environment management system Metacluster manages 3 clusters of the Supercomputing technologies center of Nizhni Novgorod state university: a cluster of 64 dual-core dual-processor servers based on Intel Xeon 3.2 GHz, 4 GB RAM and 2 miniclusters on 4 and 5 nodes, respectively. Gigabit Ethernet is the main data network inside the clusters.

Resources under control of Metacluster are actively used by a wide range of research workers and students of the University and some other users. There are some applications developed at the university which utilize high performance computing resources through integration with Metacluster: ParaLab, Global Expert. Metacluster was adapted to act as a part of an experimental grid segment of the SKIF-grid program ([14]). Metacluster is also used in the project "Developing of high performance software complex for the quantum mechanical calculations and modeling of nanoscale atomic-molecular systems and complexes" (see [7]).

There are more than 100 user registered in the system. During peak periods the workload of the system was more than 500 tasks per day. Metacluster proved to be easy to use, stable and extendable system.

Currently the Metacluster team works on using of multiple clusters in one MPI task as well as on improving and extending the remoting manager component options.

## References

1. Gergel, V.P.: Theory and practice of parallel computing. Internet university of information technologies intuit.ru, Moscow (2007) (in Russian)
2. Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The physiology of the grid: An open grid services architecture for distributed systems integration (2002)

3. Lee, C.: Parallel Job Scheduling Algorithms and Interfaces. Department of Computer Science and Engineering. University of California, San Diego (2004)
4. Kustikova, V.D., Senin, A.V.: Integration of clusters management system Metacluster with Maui scheduler. In: Proceeding of Technology Microsoft in theory and practice of programming, NNSU, N. Novgorod (2008) (in Russian)
5. Massie, M., Chun, B., Culler, D.: The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. Parallel Computing 30 (2003)
6. Lozgachev, I.N., Senin, A.V.: Monitoring of high performance computing in Metacluster clusters management system. In: Proceeding of Technology Microsoft in theory and practice of programming, NNSU, N. Novgorod (2008) (in Russian)
7. Vasil'ev, V.N., Buhanovski, A.V., Kozlov, S.A., Maslov, V.G., Roganov, N.N.: High performance software complex for modeling of nanoscale atomic-molecular systems. In: Technologies of high performance computing and computer modeling, SPbSU ITMO. SPbSU ITMO, University telecommunications, Saint Petersburg, vol. 54 (2008) (in Russian)
8. TOP500 Highlights (November 2009), http://www.top500.org/lists/2009/11/highlights
9. World Community Grid official website, http://www.worldcommunitygrid.org/
10. Web Services Glossary, http://www.w3.org/TR/ws-gloss/
11. Mono project official site, http://www.mono-project.com/Main_Page
12. Wiki Interface Specification, version 1.2, http://www.clusterresources.com/products/mwm/docs/wiki/wikiinterface.shtml
13. ANSYS FLUENT Flow Modeling Software, http://www.ansys.com/products/fluid-dynamics/fluent/
14. SKIF-grid initiative official web site, http://skif-grid.botik.ru/