Bibliography

1. V.N. Koval, Yu.V. Kuk. Distances between predicates in by-analogy reasoning systems, “Information Theories and Applications”, International Journal, vol. 10, N 1, p. 15-22, Sofia, 2003.
2. V.N. Koval, Yu.V. Kuk. Finding Unknown Rules of an Environment by Intelligent Goal-Oriented Systems, “Information Theories and Applications”, International Journal, vol. 17, N 3, p. 127-138, Sofia, 2001.
3. Гладун В.П. Партнерство с компьютером. Человеко-машинные целеустремленные системы. – Киев: «Port-Royal», 2000. –128 с.
4. Величко В.Ю. Розв’язання дослідницьких задач в дискретних середовищах методами виведення за аналогією. – Киев: Кандидатская диссертация. – 2003. – 150 с.

Authors' Information

Valeriy Koval – The Institute of Cybernetics, Head of Department, address: 40, Prospect Glushkova, Kiev, Ukraine; 03680 e-mail: icdepval@ln.ua

Yuriy Kuk – The Institute of Cybernetics, senior scientific researcher, 40, Prospect Glushkova, Kiev, Ukraine; 03680; e-mail: vkyk@svitonline.com .

CLUSTER MANAGEMENT PROCESSES ORGANIZATION AND HANDLING

Valeriy Koval, Sergey Ryabchun, Volodymyr Savyak, Anatoliy Yakuba

Abstract: The paper describes cluster management software and hardware of SCIT supercomputer clusters built in Glushkov Institute of Cybernetics NAS of Ukraine. The paper shows the performance results received on systems that were built and the specific means used to fulfil the goal of performance increase. It should be useful for those scientists and engineers that are practically engaged in a cluster supercomputer systems design, integration and services.

Keywords: cluster, computer system management, computer architecture.

ACM Classification Keywords: C.1.4 Parallel Architectures; C.2.4 Distributed systems; D.4.7 Organization and Design

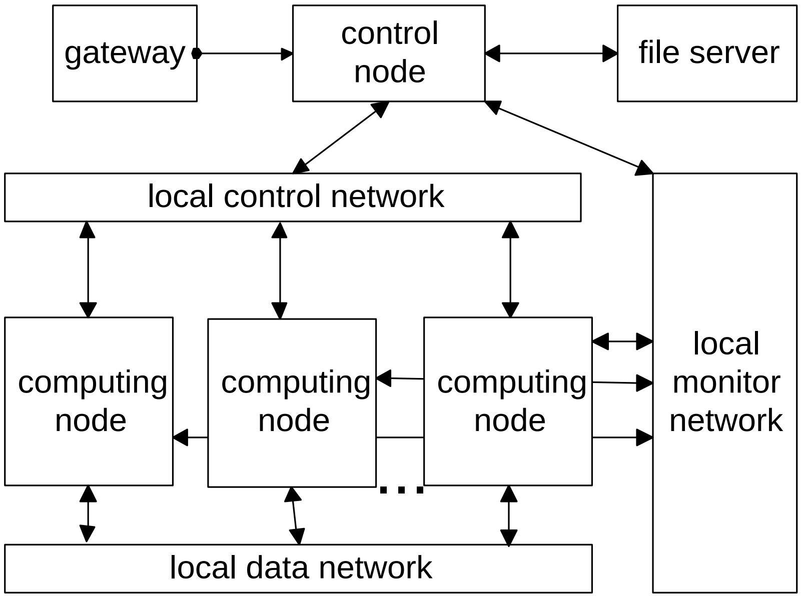
1. Cluster Complex Architecture

Basis cluster architecture is the array of servers (contains computing nodes and the control node), are connected among themselves by several local computer networks - a high-speed network of data exchange between computing nodes, a network of dynamic management of a server array and a network for cluster nodes monitoring. User access to cluster as a whole can cope by the access server - a gateway on which check of the rights of access of users to cluster and preliminary preparation of tasks for execution is realized. File services are given user tasks by a file server through the cluster control node. A file server in a system provides data access on file level protocols, like Network File System (NFS). A file server is connected directly to a local data network via high throughput channel. In some cases, the gateway and/or file server functions may be carried out on the control node.

Cluster computing node is a server, more often dual-processor, for direct execution of one user task in one-program mode. Computing nodes are dynamically united through a network in a resource for a specific task, simultaneously on cluster some problems may be executed, depending on amount of free computing nodes.

The control node of cluster is a server on which are carried out compilation of tasks, assignment of cluster resources (computing modules - cluster nodes, processors) to the user task, global management of processes activated on nodes during task execution, granting to task needed services of a file server.

Cluster structure



2. Dynamic Management with Cluster Nodes

The role of the dynamic management is to manage access to computing nodes and to provide a dynamic reconfiguration of a system. Dynamic management of a cluster system is mostly determined by the used logical systems of a parallel programming (LSPP) (i.e. their architecture and communication libraries). But it can also be influenced by nodes interconnect architecture, rather, a data communication network (means to connect the cluster nodes among themselves and with cluster control node).

A basis of a dynamic cluster reconfiguration under a user task is defined by the list of cluster resources allocated to the task (nodes, processors). After the resources are reconfigured, the system provides a corresponding handling of a user task only within the framework of the appointed resources.

The element of this list of cluster resources is assigning to task the name of node and quantity of processors, which are active in the node. A node always is appointed entirely, whereas the request of a task always specifies necessary amount of processors.

The cluster resources handling system estimates real presence of resources and "collects" the number of processors necessary to a task from the pool of really active nodes at the moment of free nodes request. Processors are allocated always in the cluster node staff, i.e. it is impossible to allocate in one node on one processor to the different tasks, processors of node unused in a task always should stand idle.

In the cluster, where the communication network is based on the switch (Gigabit Ethernet, Infiniband), any of nodes accessible to a task can cope irrespective of other nodes in this configuration up to full restart. Mutual influence of cluster nodes upon serviceability of a communication network does not exist as a whole - it is provided with the switch.

For a network on basis SCI cards the opportunity of a direct handling of the cluster node within the framework of allocated cluster resources is sharply limited, as the communication network "rises" entirely and serviceability of separate node can depend on serviceability of connections with the next nodes essentially [1].

Though at application 2D-and 3D-topology, it is possible the dynamic change of routing that supposes detour short, but defective connection due to working, but longer, connections through other nodes. However if several nodes die, then a general cluster performance is going down up to transition to a disabled condition. On the other hand, when using a central switch (which is not mirrored), the switch causes a death of all the system.

An opportunity of reconfiguration depends also on a usage of local disk memory of the node. For a cluster systems with a distributed storage based on a local node's hard drives there is a problem found with an execution of user tasks in a background batch mode. When a repeated return to a computing process for the task execution is required, it is necessary to receive the same cluster resources for a task that was provided in a previous stage of the task execution (it implicitly demands long reservation of disk resources on all cluster nodes, appointed to a task).

Reduction of negative influence of this restriction is possible only at refusal from the local disk resource for background tasks for the benefit of network file systems (for example NFS) or the general file systems oriented on cluster application (GFS) [2]. This allows do not care about granting the same cluster resources for the task being executed in a background batch mode.

After task is finished, all allocated resources should be returned in a pool of free resources. Rational use of this pool assumes a regular check of resources' state. The system diagnosis and makes a conclusion about an unavailable resources in an emergency configuration to exclude their incorrect usage. This part of a management system is one of the most important parts of all the cluster management software.

3. Management of Cluster Accessibility

There are several approaches known in a field of cluster resources access management. All of them are based on a standard user authentication on a stage of a system user login. After login is made there are following general ways possible:

1. A user receive an access to all cluster nodes, assigned as a resource to one's queued task, i.e. the task is executed on behalf of the user and a user has a full control over the behaviour of nodes, usage of node own resources (main memory, exchanges with a file server and other nodes, employment of the processor) is given to this user.
2. A user receives an access to an interface of a task status control and management of a task execution. Thus, a user has no real access to cluster nodes allocated.

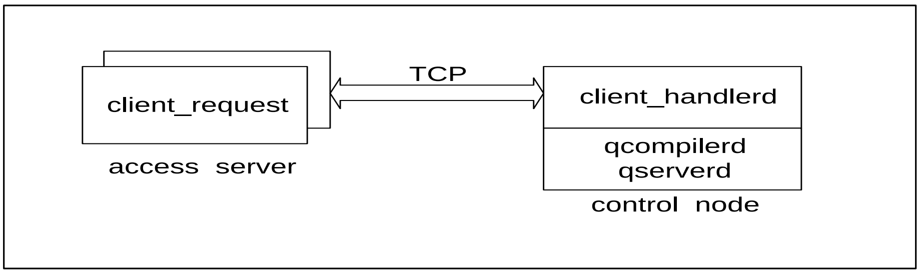
At the first approach the list of users is exported to all cluster parts or the real system user is dynamically created for the period of a task execution. The control over access to the system variables, data and command files of cluster management, nodes essentially becomes complicated, as for communication network SCI this control should be more rigid, than for cluster on the basis of the switch. On the other hand, granting to the user the full access to node allows going to the manual management of task execution up to loading into local disk memory of a node. One of the examples of the mentioned approach implementation is MBC-1000M (Moscow) system [3].

In our opinion, the second approach, despite of considerably big system costs on the organization and support of user work, is represented to more reliable in preservation of integrity of the system software, its functioning and cluster security from the non-authorized access. In this case all works on task execution on nodes are carried out by the specialized pseudo-users existed only on cluster nodes. On behalf of these pseudo-users, the task is executed. For integrity of the approach, an every LSPP has the unique specialized pseudo-user; i.e. the policy of safety does not permit a real user, except for repair managers, to log in into cluster nodes. Such a system provides greater security and reliability of a cluster.

Absence of direct access of the user to cluster nodes is compensated by presence of a specific user interface. An interface allows users to operate a task execution, task queues, to load the data for a task, to supervise a condition of the nodes, which are included in a resource of a task, etc. A program of the user interface cooperates with a demon started on the control node and carrying out all necessary user work. The cluster administrator has the possibility to execute any of these functions.

3. Task Processes Handling

Users, as with the remote access as taking place in a corporate local network, get access only to a gateway - access server, the last holds all user catalogues exported from a cluster file server and support user preparations of the tasks for execution. The subsystem of service of users and their tasks has client-server architecture: the client part settles down on a gateway, the server part - on the control node, connection between these parts is organized under TCP- protocol.



Requests from the user are transferred to the control node and executed by a demon client_handlerd, at this one control node can serve a little clusters with identical architecture. The demon client_handlerd carries out the requested action and returns result of performance to the user.

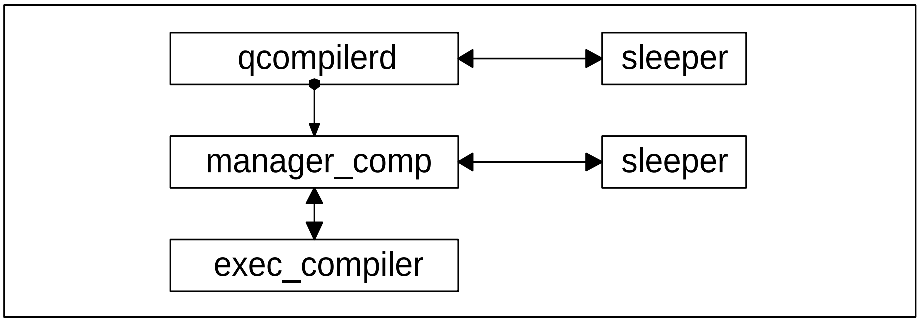
One of such actions is the definition of necessity to compile task and queue it up for compilation with the subsequent placing (at the absence of compiling mistakes) in the execution queue. Each of these queues is served by the demon, correspondingly, qcompilerd and qserverd, their activities on the control node; their Status may be change only by the cluster administrator. In the same way the user receives data about queued tasks, on cluster congestion, presence of free resources, etc.

The qcompilerd functions are:

5 Search of a task (without the control of parameters of task execution);

6 Creation of working structure where this task is compiled;

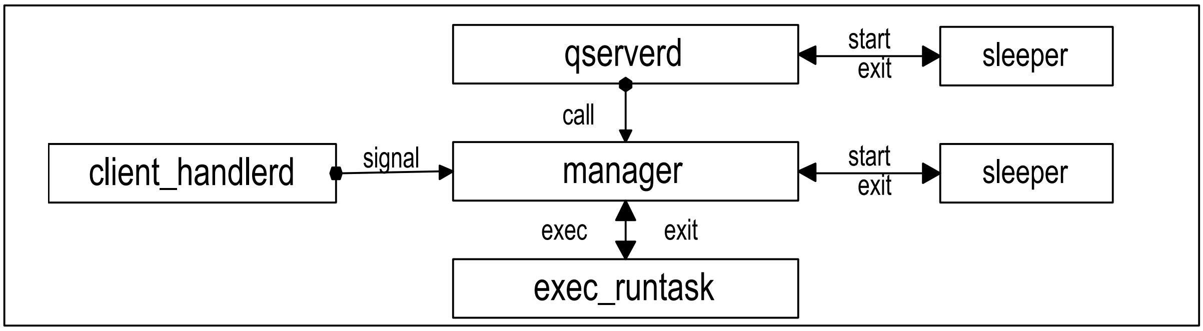
7 Start of the compilation manager, monitoring the specified task, and return to search of other task to compile.



The manager of compilation, in turn, starts as independent process a command file of compilation in a mode *chroot*, expects the end of compilation and returns after that results of compilation in the user individual catalogue.

The qserverd functions are:

- Search of a task (with the control of parameters of task execution);
- Updating or creation of working structure of a task for execution;
- Assignment of resources to a task;
- Export of an environment, start of the manager of execution of a specific target and return to search another task for execution.



The manager of execution (manager), in turn, starts as independent process the command file of execution (exec_runtask) in a mode *chroot*, expects the end of execution exec_runtask or the user signal about task execution stopping - through a demon client_handlerd - and returns after that results of execution to the user individual catalogue.

4. Cluster Management System (Base Functions)

Management system – cluster control facilities, used both the system administrator, and various software systems over the operating system, having for an object "continuous" monitoring of computing process, the equipment and the software. It contains, at least, three obligatory parts:

- A direct control of computing process and functioning of the cluster equipment;
- Management of service means of a task stream processing and user works with cluster;
- Monitoring cluster infrastructure (system of power supplies and cooling, a cluster configuration and availability of the cluster components through its communication networks).

The management system may be resident on one of control nodes with an opportunity to change this place to another, and may be distributed among them is depends only on the rules of functioning of managing means.

Obligatory functions of a management system are:

- Management of start, stop and restart all cluster equipment, and its separate nodes and also active means of the cluster system software, in particular, means of a task stream processing;
- Monitor service of the system administrator needs with results of the analysis of a cluster status, its configuration and availability of its nodes;
- Management reconfiguration of node connections if it allows the accepted circuit of a configuration;
- User authentication at its local or remote login to the cluster, support of its functioning during task preparation, granting of help services both online and offline;
- Support of service means of the user interface at compilation, assembly and task execution, under the control of intermediate results over long task execution, on preservation of results of the task running, maintenance of user tasks with services of a file server and DBMS on it;
- Support of a message exchange between the system administrator and users;
- Remote user maintenance with means upload/download to transfer the data between its local client computer and the cluster client individual catalogue.

5. Support of the User Computing Process Means

Cluster oriented tasks should use the communication libraries, more often implementing the MPI interface. In this interface the task will start on the zero allocated node with the indication of necessary processors quantity, names of a task code file and some other parameters. For example, mpirun-np 16 /test/test2, where mpirun - standard command for task start, np - required number of processors, /test/test2 - a path to the task code file.

Implicitly in this start rights of the owner of the catalogue from which start is carried out, and rights of the owner of a code file are taken into account also. The coordination of these rights and maintenance of start correctness, and also a correctness of access to the data, dissipated upon cluster file system, are assigned to service means.

Compilation of a task is made on behalf of the pseudo-user, representing chosen LSPP, on the control node without attraction of cluster resources with the subsequent transferring the compiled task to queue for execution on cluster nodes under the control of the same pseudo-user determined as the only thing for ordered LSPP.

Client-server means of user's interaction are included into means of support of the user computing process with control facilities tasks also. The accepted principle is the user alienation from executed tasks, that is client, placed in cluster environment, get access only to the gateway – access server physically separated by network addresses from the control node and other cluster nodes, and working areas of the tasks started on execution are placed on the control node. Functioning service means, client-request on the access server and client_handlerd on the control node, having established connection among them, support it activity till the moment of the termination of concrete user request.

The direct task start is connected to significant inconveniences by the rights of access. More effectively to add additional interfacing means to start the task on allocated cluster resources. These interfacing means should coordinate correctly rights of access during start, estimate and prepare for real use the list of cluster resources, check their sufficiency and, maybe, real availability. As unification of LSPP is absent, these means are individually adjusted on each type of LSPP through environment variables of execution PATH, LD_LIBRARY_PATH and specific ones for concrete LSPP.

Cluster tasks, as a matter of fact, are tasks with great volumes of calculations and consequently, the period of the maximal uninterrupted execution cannot be uncertain, that is why the monopolization of cluster as a whole or only some its parts under one task is incorrect, long on time of the task running should represent a chain of consecutive starts and breaks of the execution (i.e. a set of quantums to run the task), alternated by the idle periods waiting the reception of quantum. Service that means to support the execution of such tasks should provide a correctness of the termination of concrete quantum, preservation of the intermediate data and renewal the execution in the other quantum.

One more service means, facilitated work of users, may be the debugger of cluster tasks, it allows with cluster resources limited from above receiving reports as task executions on the concrete processor, as characteristics of data exchanges between cooperating processors. The attitude to such debuggers dual, rough debugging on them goes conveniently enough and naturally, exact debugging is usually connected to searches of opportunities of increase of task productivity, searches of memory "leakage" and adjustment of a task for the big number of processors, that just and cannot really be supported by noncommercial cluster debuggers.

6. System Means for Increasing the Cluster Performance

Among many means to improve the quality of cluster functioning, it is possible to discuss the basic:

- To carry out hardware improvements in a communication network of nodes, in particular, using network adapters SCI-technology instead of switch oriented Gigabit Ethernet, making up the connections on the basis of 2D-topology (or 3D-topology) and choosing the optimal variant of node switching (i.e., for 16-node cluster with processors Xeon only transition from the network based on switch with Gigabit Ethernet to a network based on SCI gives almost 30 % a gain of performance in Linpack test, and replacement of switching 2x8 nodes on switching 4x4 nodes gives a gain on 4-6 %).
- To maximize the using of node main memory due to exact selection of the used software. So, use only a necessary minimum of demons on node allows to achieve employment of all 12-16MB on the unloaded node.
- To use architecturally – optimized libraries and the compilers giving the most effective codes, in particular, Intel compilers for languages C and Fortran or family compilers GCC, use library MKL (Intel Math Kernel Labs) instead of library ATLAS.

Total results of consecutive changes for 16-node cluster with processors Xeon 2.66 GHz at 2 processors and main memory 1 GByte on node (that gives peak performance in 166 Gflops) are resulted in table 1.

The analysis of table 1 shows, that obligatory elements of cluster adjustment, needed for the maximal productivity, should be - "thin" adjustment of a node main memory for system using, installation, adjustment and use of the richest noncommercial libraries, even for rather weak communication network on Gigabit Ethernet. In case of replacement of switch oriented weak network by more powerful (in particular, by SCI as with Infiniband

[4] we did not have experiments) yet it is necessary to choose rational configuration of data connections, recommended the vendor firms, and to use communication library Scali, instead of MPICH-SCI.

Table 1

Changes in structure and the system software	*The measured maximal performance in Linpack test (Gflops)*	*Ratio max/peak performance (%%)*
Initial configuration: Communication network =Gigabit Ethernet, Accessible MM = 0.83 GByte, Compiler = GNU, Library = ATLAS	71	43
Communication network =SCI, Switching = 2x8, Communication library = MPICH-SCI	94	57
Accessible MM = 0.99 Gbyte	99	60
Switching = 4x4	104	63
Library = MKL, Communication library = SCALI	112	67

One more factor influencing the common cluster performance is a rational choice of structure of file system. Generally, when installation of commercial OS Red Hat Cluster Suite which contains cluster oriented file system Global File System is not supposed, and there is a local system of a data storage based on a RAID-array in the structure of control node entering or served by the specialized server, and local disk memory on cluster nodes is absent, the most effective means may appear export of references to contents of a RAID-array to all points of the cluster where work with files is supposed. Thus even for the user individual catalogues which formally should be on a gateway – access server, their physical accommodation in disk memory of the gateway is not supposed, they only there are exported from a file server by the references.

Similar by results of the decision can be offered for access to files in an executing task - despite of accommodation of the big data files in the individual catalogue of the user, direct access to which to the absolute address from node is impossible, and copying of data files in working structure of task execution is comprehensible only to the small sizes of files (for example, tens Mbytes), indirect addressing through tables of address transformation will provide access to the data of great volume without their moving to working task structures.

The reference to databases, which are stored in the same RAID-array, actually does not differ from described. Unfortunately, experiments in this direction just begin, as well as authentic results are absent.

Bibliography

1. http://www.scali.com
2. http://www.redhat.com/software/rha/gfs
3. http://parallel.ru/computers/reviews/MVS1000M.html (In Russian)
4. http://www.mellanox.com

Authors' Information

Valeriy N. Koval – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; e-mail: icdepval@ln.ua

Sergey G. Ryabchun – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; e-mail: sr@emt.com.ua

Volodymyr V. Savyak – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; e-mail: Volodymyr.Savyak@ustar.kiev.ua

Anatoliy A. Yakuba – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; e-mail: ayacuba@voliacable.com