CLUSTER SUPERCOMPUTER ARCHITECTURE

Andrey Golovinskiy, Sergey Ryabchun, Anatoliy Yakuba

Abstract: The paper describes the architecture of SCIT - supercomputer system of cluster type and the base architecture features used during this research project. This supercomputer system is put into research operation in Glushkov Institute of Cybernetics NAS of Ukraine from the early 2006 year. The paper may be useful for those scientists and engineers that are practically engaged in a cluster supercomputer systems design, integration and services.

Keywords: supercomputer, 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. Introduction

In 2004-2005 years small developer team from the Glushkov Institute of Cybernetics NAS of Ukraine built and put into research operation two high-performance supercomputer systems with cluster architecture SCIT-1 and SCIT-2 on the basis of modern unicore Intel microprocessors. The developed supercomputers allow to solve essentially new challenges of the big dimension in the field of a science, economy, ecologies, an agriculture, in space branch and other branches.

Dynamics of managerial processes during task flow computing inside a supercomputer system, adaptation of existing and creation of new architectural means for maximization of global characteristics of supercomputer productivity is the investigation objects in the current work of research team.

2. Architecture Components of Cluster Supercomputer

The architecture of the multi-server, cluster system is a multi-plane combination of hardware-software means, particularly at the level of interaction of the server operating systems, distributions of processes of computation on processors and synchronization of these processes, effective maintenance of queries to the centralized or distributed files systems.

Specific of tasks for the cluster computing. The supercomputer of cluster type is the computer system with asymmetrical multiprocessing and strongly connected nodes and task, intended for execution in a such computation environment, have features:

- each task consists of great number of interactive processes having an *identical* code, they are started on the different cluster nodes and each of them executes some part of *common* work;
- during the task execution processes can exchange intensively between themselves;
- interprocess data exchange results in smoothing of productivity of each process on speed of the *slowest*;
- each process, as a rule, occupies the large volume of main memory for the period of execution.

A cluster task is located in the user domestic catalogue and for implementation is got on competition basis two cluster resources - processor resource and timing resource. The long continuous execution of task is connected to an opportunity to not receive results in time (for example, from failures or at the large load by other tasks), therefore the double timing resources is given tasks – both full time of execution of task as the session time i.e. time of noninterapted execution, after a session a check point keeping received intermediate result should be formed. The technology of programing with check points is one of basic components of organization of task execution, as allows repeatedly to interrupt execution and to proceed in it on intermediate results.

Simplified structure of supercomputer. The supercomputer of cluster type is the array of computing nodes, each of which is a multiprocessor server with symmetric mulitprocessing in the field of common main memory

(SMP-architecture), incorporated by a few local areas networks of different purpose and productivity; from the array of computing nodes can be abstracted frontend servers (managing nodes) for the centralized process for handling of task executions. Besides this, there are the servers, specialized on the management by shareable files resources (file server) and external access of users to the cluster (access server) – see fig.1.

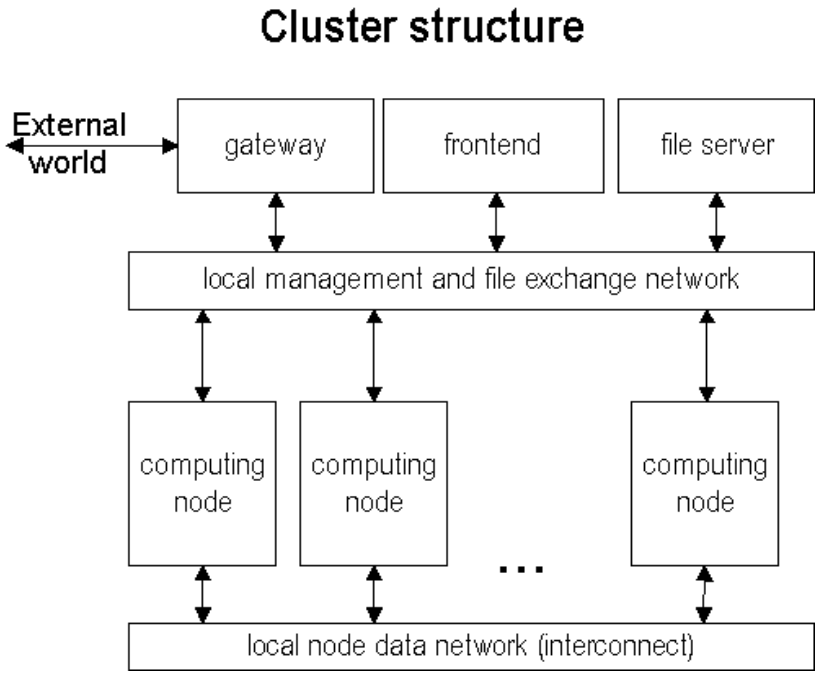

Fig. 1. Simplified cluster structure

Some hardware and software features of SCIT supercomputers are resulted in table 1.

Table 1. SCIT clusters hardware characteristics

	SCIT-1	SCIT-2
Computing nodes quantity	**24**	**32**
Node processor quantity	**48 (Xeon 2,67 GHz)**	**64 (Itanium2 1,4 GHz)**
Frontend quantity	**1**	**1**
Processor cache (Mbyte)	**1**	**3**
All main memory (GByte)	**48 (DDR SDRAM PC-2100 ECC)**	**64 (DDR SDRAM PC-2100 ECC)**
Interconnect network	**Infiniband**	**SCI (Scalable Coherent Interface)**
File managing network	**Gigabit Ethernet**	**Gigabit Ethernet**
Storehouse (TByte)	**1.6 (Common to both clusters)**	
Operating system (Linux)	**Fedora Core 4**	**CentOS 4.2**
Linux kernel	**2.6.12**	**2.6.12**
Global File system	**Lustre 1.4.5**	**Lustre 1.4.5**
Parallel programming system	**Open MPI**	**Open MPI, Scali**
Programmming language	**C, C++, Fortran-77**	**C, C++, Fortran-77**

The cluster operating system are chosen Linux FedoraCore4 and CENTOS 4.2, they are established both on frontend and computing nodes of clusters. As the root file system for computing nodes is used NFS, and as the distributed file system are chosen **Lustre** [1].

The cluster software accessible to the user includes for programming languages C/C++, Fortran compilers of the GNU and Intel different versions, for the parallel calculations - optimized libraries of ATLAS[2], BLACS[3], SCALAPACK[4], Intel MKL[5], application packages of GROMACS[6], WIEN2K[7], GAMESS[8] et al. As a parallel interface various realization of MPI interface - SCAMPI[9], OPENMPI[10] are used.

Task management and file exchange networks. In the environment of tasks management and file management two logical networks are allocated - management network (MN) and file exchange network (FEN). FEN service generally should give opportunities:

- remote management of computing node through protocol of WakeOnLan;
- access of node to the data on a network configuration (protocols of DHCP);
- loadings of the operating system in a computing node (protocols of TFTP);
- access of node to root file system (protocols of NFS);
- deliveries at the computing node of the task data (protocols of NFS).

MN service generally case provides an opportunity of access to the node from outside for:

- operative management by a node;
- receptions of statistical information on loading of processors, employment of memory, the indication of gauges of temperature, speed of rotation of fans;
- start and further control of task processes.

Let's consider in detail use of a network of data exchange for the process to start the cluster computing node. Each node is configured on inclusion at reception by the network interface of the special package *wake-on-lan* and on the load through a network interface by PXE-protocol. A frontend sends the formed package through FEN and a node initiates the load process.

A node sends the broadcast inquiry and from the server DHCP which is established on a frontend, receives all data necessary for loading system, downloads a kernel and minimum root file system from the TFTP server, which is also established on the frontend, unpacks a kernel and starts its implementation.

Farther the process of initializing of the system, being based on received on DHCP data, mounts on NFS file system located at storehouse, does it by a root and completes initializing, having transferred management to the starting scripts located on the new root file system. Since this moment loading of system on a network or from a local disk does not differ practically. Further additional sections NFS with working data, by users directories and etc necessarily are mounted.

Such scheme supposed that root file systems at all cluster nodes are the same, essentially facilitates administration, update, installation of the new software, as works with all cluster entirely, and on orders reduces an opportunity to make a mistake. Root file systems of all nodes are identical, except for a few catalogues which really should be unique at everyone, but also they are located in main memory of node. Processes of each activated task, working as everyone on a separate node, all the same work in the same catalogue located at the storehouse, read and write from/to the same files.

As we see, practically all on the file input-output work is done on the data exchange network, therefore the requirements to throughput of this network very high. It is necessary to specify, that a «bottle neck» in this chart is the network interface of server, throughput of the network interface of node suffices much.

The start of task execution on the cluster nodes can be carried out by various ways depending on a task, i.e. a MPI-task is started by the command of *mpirun*, and the ordinary not parallel program can be activated by the command of *ssh* or *rexec*. For the operative control after the started task state, for its forced completion and liberation of resources busy at a task access is also used to the node on protocol of SSH, it non-obvious implies, that the node should be accessible.

Necessity for the base remote management in each cluster node separately, an opportunity of implementation of such operations as startup-shutdown of node, the console with the output of load of node have demanded

installation of **ServNET** [11]. Further it is planned to use nodes only with support of the IPMI interface [12] of version above 1.5, node providing the remote startup-shutdown at presence only of the Ethernet cable and feed connected to the node, and the function of SERIAL-OVER-LAN in IPMI 2.0+ allows even remotely to adjust the node's BIOS.

IP-network. As supporting in a cluster systems an IP-network is used with a few ranges of private IP-addresses: 10.0.0.0 – 10.255.255.255, 172.16.0.0 – 172.31.255.255, 192.168.0.0 – 192.168.255.255, thus for the cluster nodes used private range is 10.0.0.0 – 10.254.254.254 as most spacious. The following chart of distributing of subnet of IP-addresses is applied in the last:

- A computing node has an IP-address 10.N.M.X, where N is number of cluster, M is number of switch, X is number of port in the switch. Thus, 10.1.1.1 is the first node of the cluster #1, and 10.3.1.24 is twenty fourth node of the cluster #3.
- Mask of IP-address 255.0.0.0, i.e. all network infrastructure is fully attainable from any point, here differentiating of different clusters is executed by VLANs. The nodes of different clusters are mutually invisible as a result, but the systems of storehouse will be accessible even in case if the functions of storehouse and managing cluster node are laid on one device. The switch of networks of cluster management has the fixed IP-address: 10.N.M.250, where N is number of cluster, M is the number of switch.
- The frontend (managind node of cluster) has the fixed IP-address: 10.N.M.254, where N is number of cluster, M is the number of switch.
- The subnet 10.0.0.0/16 is given for services, so 10.0.0.254 is an access server address, (10.0.0.11 – 10.0.0.15) are devices of UPS and etc

File service. As a rule, parallel tasks are focused on the computing connected to huge files of the initial, intermediate or final data. *So, the analysis of results of nuclear researches with terabyte size of the initial data, and a tasks in a package of quantum chemistry Gamess* [8] *can use hundreds creates time files in the size in some gigabytes on process with usual read - write the intermediate results by small, fine packages.* Therefore the extremely important problem is to give to nodes high-speed access to storehouse systems of the huge sizes.

For increase in throughput of system of a storehouse it is used PORT TRUNKING - aggregation of 2-4 network interfaces in one for increasing the common new throughput though a linear gain is not possible.

As the distributed file system of cluster system SCIT still recently it was used NFS. As cluster nodes have no own disks each node during initial loading mounts root file system by NFS. Besides by NFS operating system sections with the working data of tasks were mounted also. Choice of NFS has been caused by the several reasons is a standard network file system, NFS is present in any UNIX-system, NFS is very easily adjusted and configured.

Operating experience NFS within one year as the basic file system has shown, that NFS is an excellent choice only for small (on 4-8 nodes) clusters, for clusters a level the SCIT, on 16-32 nodes, NFS can be a quite good choice under condition of use for the account of tasks with a small amount of operations of input-output with disk files. However NFS becomes a unacceptable choice at use for the task execution with an intensive input - output. Therefore file service has been modified, and the primary goals of modification were:

- a choice of the optimal distributed file system with an opportunity of scaling as on volume about an opportunity to node existing storehouse various clusters in one common data storage and on the maximal throughput;
- transition from use NFS on partial or full use of the chosen file system.

The following candidates for a role of the distributed file system were examined:

GFS manufactures RedHat (earlier SISTINA.COM) [13], for today last version 6.1. GFS uses as the distributed storehouse mounted simultaneously in all units GNBD (global network block device) atop of which GFS works actually with the manager of blocking.

GFS has many advantages - free-of-charge decisions very much, development by the largest manufacturer RedHat Linux, work «it is direct from a box» at use RedHat Enterprise Linux 4 and is higher or Fedora Core 4 and is higher, quite good scalability on volume, ease in installation and configuration.

During too time there are also lacks - bad scalability on the general throughput, and it means, that for escalating capacity it is necessary to use expensive hardware decisions such as FiberChannel as all nodes are shared with one block device refusal of one of nodes can lead to some damages of file system.

GFS is a quite good choice for the finished decision when it is not planned to increase computing cluster capacity and volume of system of a storehouse, i.e. delivery on a turn-key basis. For use in our case supposing the further escalating of capacity on computing resources and volumes of disk space, approaches a little.

OCFS2 manufactures ORACLE [14], the successor with open codes OCFS. While the stable version of system still is not present, but it already is in kernel Linux, is supported by various distribution kits Linux. Actually represents distributed on all cluster nodes a RAID5-file that gives both high speed of read - write, and some fault tolerance of all file. However, if some units have broken down, the file can collapse down to loss of all data, i.e. the system demands a highly reliable disk subsystem on each cluster node, that very strongly increases a total price of the decision. OCFS2 it is optimum for processing the big databases for what actually and it was created.

Lustre manufactures CLUSTERFS.COM [1], the commercial, free-of-charge version, leaves with some backlog, is maximum for one year. It is very hard in installation, but it is very simple in configuration. It is perfectly scaled both on volume, and on throughput.

The system uses a set of patches to a kernel and consequently there can be problems of its construction, is especial in case of use of a various sort of the non-standard equipment. Problems basically are connected to a binding of interconnect drivers and driver **Lustre** to determined and not always to the same versions of kernel Linux. especially big such discrepancies arise at simultaneous use of architecture IA64, proprietary drivers SCI from firm SCALI and file system **Lustre**.

Thus it is very simple in configuration, it is enough to tell, that after big amount of works on construction **Lustre** and starting adjustment of cluster startup of node demands literally some minutes.

From the point of view of system **Lustre** looks as usual local file system with all pluses as aggressive caching *inodes* and *dentry*. Realization of full compatibility with POSIX is expected at third quarter of 2006 (the call *flock/lockf* doesn't realized now). Escalating of volume and throughput is made by simple addition in system of one or several nodes with disks (OSS). As each file can "be stripped" on several OSS and thus access to it made be parallel throughput grows practically in an arithmetic progression, i.e. the more at us is established OSS, the above speed of read-write. Thus very high degree of recycling of devices of an input-output so File I/O exceeds 90 % raw bandwidth disks is reached{achieved}, and single GigE end-to-end throughput reaches 118 MB/s at a physical maximum of the interface 125 MB/s. Additional plus is that as the network interface in **Lustre** any interface supporting report IP can practically act, and in some cases and more low level protocol (for example, Infiniband).

High enough requirements on reliability are showed to storehouse, that is clear, as now integrity of file system depends on serviceability of all components entirely. And, though at what refusal or OSS-node cluster nodes can continue work if their data have not been located on the damaged node, but parts of the data all the same can be lost (in following versions **Lustre**, except for a mode of storage of files stripe or RAID0, modes RAID1 for small and RAID5 for the big files will be realized, therefore the probability of full loss of the data will be sharply reduced). Therefore use on OSS-units of RAID-files with redundancy is not the recommendation but the requirement.

Requirements to performance of OSS-nodes are very low. As the node is occupied with one task execution - maintenance of an input-output, any modern processor fulfils this task with success at low final cost. Moreover, idle computing powers are well enough to realize the programmed RAID-array function, i.e. they saved money for expensive controllers (modern RAID-controllers lose to program RAID-arrays in speed for the banal reason - controllers uses weak enough CPU, actually for last years 5 RAID-controllers have found support RAID6,10,50,60, trunks PCI-X and PCI-E, but their computing capacities have remained at the same level of five years' prescription and in competition of frequencies wins more high-speed processor).

As there is some blank in **Lustre** performance – it's a search of files in the catalogue, 5000 op/s is very low figure and in some cases results in falling productivity (for example one of programs of the user created in the working catalogue about hundred thousand files and degradation of speed was appreciable). However this feature easily manages accommodation of working files not in one but in tens or hundreds catalogues.

After the analysis existing parallel (as above-named, and some other) file systems and an estimation of our technical opportunities we were defined that our specifications quite corresponded to requirements to **Lustre** and this file system has been chosen as the major candidate for a role of the distributed file system for ours supercomputers.

Early 2006 clusters SCIT have been transformed to use of distributed file system **Lustre**. It has allowed to unit all storehouses in one common file system in volume 1.7 Tbyte physically general file system settles down on three servers of the data (OSS) with four disk files (OSD) and one server of the metadata (MDS). As at configuration **Lustre** we have specified to distribute a file on all OSD (actually it is classical RAID-0 in application to a file) thus we could distribute loading on a file input - output simultaneously on all servers.

Results of testing of two file systems, **Lustre** and NFS, on a file in the size in 8 Gbyte (in testing are measured: throughput - Kb / c, use of the processor, frequency of search) are resulted in table 2.

Table 2.

Operation	Sequential Output						Sequential Input					
	Per Char		Block		Rewrite		Per Char		Block		Random seek	
	Kbps	%CPU	Kbps	%CPU	Kbps	%CPU	Kbps	%CPU	Kbps	%CPU	K/sec	%CPU
NFS	**26665**	**88.1**	**27907**	**6.3**	**3134**	**95.2**	**29215**	**91.1**	**84975**	**15.3**	**460.7**	**2.8**
Lustre	**27791**	**99.3**	**69991**	**41.3**	**39668**	**58.0**	**28066**	**98.9**	**98254**	**86.1**	**121.6**	**17.3**

Further it is possible to increase volume and throughput of file storehouse by simple connection to the switchboard of additional servers with disk files and small reconfiguration of the system.

3. The selection of architecture features to a supercomputer project

What characteristics may be selected for the new supercomputer project. Proceeding the cluster tasks from mentioned early specific properties, it is possible to formulate the commons requirements to the node of cluster for the effective decision of parallel tasks:

- productivity of node linearly depends on power of processor, and productivity of processor from frequency descriptions of the used bus of main memory and amount of main memory accessible in a node (to some reasonable limit);
- interprocessor data exchange always faster than a interconnect exchange, i.e. preferably to use multiprocessor nodes (with 2–4 processors) and multicore processors;
- productivity of node depends on as used interconnect, two features are here important is latency, i.e. delay arising up at the transmission of minimum package between nodes, and maximal carrying capacity;
- productivity of node depends on intensity of operations of input-output with the devices of storehouse.

Pipeline and systems calls. As a rule, parallel tasks executed at computing node are not used with cyclic algorithms, therefore classic architecture with a short pipeline, used in the processors AMD, much preferably architectures P4 processors INTEL. Every reference to the data of neighbouring process is accompanied by a few transitions in the kernel mode of processor. Price of this transition on the processors of AMD 120-240 times, on the processors of architecture P4 1100-1300 times. However with appearance the recently represented architecture of Intel Conroe and actually by returning of Intel to architecture of P-III and short pipeline, in the second half of 2006 year and first half of 2007 year, i.e. down to the appearance of AMD K8L architecture, placing of forces will be completely other.

HyperThreading. Due to idle time of one of pipelines in incorrectly predicted transition or simply impossibility of parallel execution of instruction on architecture P4 there is possibility of the use of standing resources as a virtual processor (HyperThreading), but in parallel tasks it results only in falling of productivity. The reason is simple – the data exchange between nodes aligns productivity of all processes on speed the slowest and, as on a virtual

processor is no more than 40% real processor, general productivity falls in 2–3 times, i.e. this possibility for clusters is practically unavailing.

64 bits versus 32 bits. For today all modern processors either support 64–bits expansions (AMD64, EM64T) or are pure 64-bits processors. Unfortunately, now a prize from the use of word length in 64 bits the programs requiring for calculations with such arithmetic collect in 64 bits receive only, and that not always, the other only lose. The reasons for this are few (due to size of data and address megascopic twice):

- It is required to increase twice processor cache, otherwise there is falling of productivity at the frequent «washing» of a cache.
- At that width of bus of memory plenty of addresses is required twice to main memory, that gives falling to productivity.
- It is required increases twice the size of node main memory.

Power consumption of the processor. The selected power of processor can non-obvious influence on general productivity of all system – at the overheat of one of processors to it the automatic lowering of frequency will be applied, that will result in the common falling of productivity of all system on the whole.

Main memory. Clusters have 1-2 GByte on every processor core of node:

- More than 2 Gbyte on a processor core expediently either at the use of clean 64-bit architecture or after clarification of specificity of the basic applied tasks of cluster, otherwise memory will be essential to idle, however there are situations, when than the more maximal capacity of main memory of node is anymore, so much the better for a task (main memory modeling of SMP-architecture);
- Frequency on which main memory works should be maximal of all supported the chosen processor architecture;
- The used chipset should be able to support the necessary amount of memory.

Interconnect. As the cost of interconnect lies in a very wide range from the zero of to a few thousand dollars on a site, the choice of interconnect is determined by the basic purpose of the cluster system:

Latency of interconnect is one of major indexes, influencing on the real productivity of the cluster system, this time expended by the operating system and device on the transmission of single package to other cluster node. Because a data exchange between nodes takes place by such transmissions, the latency can be described as the time lost by a process. For tasks with a large data exchange between nodes the large latency can give the catastrophic falling of productivity, at the same time for tasks with a small data exchange between nodes the small latency will give nothing in the plan of winning of productivity, but will result in the enormous increase of budget of project.

The throughput of interconnect practically does not tell on general productivity of the system. There are some minimum scopes, but throughput of any device used today as interconnect, is higher than these scopes, i.e. as yet we did not meet a task to the necessity of which in the throughput of interconnect there would be compared with the necessities of ping-pong benchmark.

Conclusion

Presently productivity of existent cluster systems SCIT suffices only for the simultaneous calculation of a few tasks, therefore for satisfaction of present queries from the side of Institutes of NAN of Ukraine, that the decision of large tasks left off to be a bottleneck, it is necessary to heave up productivity of supercomputer systems on an order (to a few teraflops).

Evaluation the characteristics of the cluster system which would on nearest a few years to answer the vital queries of the mentioned directions in physics, biology, technique and etc, are the following:

1. Quantity of dual-core processors with frequency at (2,2 – 2,8) GHz – 300.
2. Main memory (total) — 1,0 TByte.
3. Bulk storage on local HDD — 2.5 TByte.
4. Global storehouse system — 10 TByte.
5. Highest possible productivity — (4,5 –5.5) Tflops.

Bibliography

1. www.lustre.org/
2. http://www.netlib.org/atlas/
3. http://www.netlib.org/blacs/
4. http://www.netlib.org/scalapack/
5. http://www.intel.com/cd/software/products/asmo-na/eng/perflib/mkl/index.htm
6. http://www.gromacs.org/
7. http://www.wien2k.at/
8. www.msg.ameslab.gov/GAMESS/GAMESS.html
9. http://www.scali.com/
10. http://www.open-mpi.org/
11. www.t-platforms.ru/english/about/dnd.html
12. www.intel.com/design/servers/ipmi/spec.htm
13. www.redhat.com/software/rha/gfs/
14. oss.oracle.com/projects/ocfs2/

Authors' Information

Ahdrey L. Golovinskiy – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; e-mail: tikus@ukr.net

Sergey G. Ryabchun – Institute of Cybernetics NAS Ukraine; Prospekt Academika Glushkova,40, Kiev, 03680 MCP, Ukraine; e-mail: sr67@voliacable.com

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

THE TECHNOLOGY OF PROGRAMMING FOR A CLUSTER COMPUTER BY THE REMOTE TERMINAL WITH OS WINDOWS

Dmitry Cheremisinov, Liudmila Cheremisinova

Abstract. The problem of preparation of a program to perform it on multiprocessor system of a cluster type is considered. When developing programs for a cluster computer the technology based on use of the remote terminal is applied. The situation when such remote terminal is the computer with operational system Windows is considered. The set of the tool means, allowing carrying out of editing program texts, compiling and starting programs on a cluster computer, is suggested. Advantage of an offered way of preparation of programs to execution is that it allows as much as possible to use practical experience of programmers used to working in OS Windows environment.

Keywords: parallel programming, cluster computer, programming technology.

ACM Classification Keywords: D.2.1 [Software engineering]: Requirements/Specifications – Tools; D.1.3 [Programming techniques]: Concurrent Programming – Parallel programming.

Introduction

In recent years, it has become clear that the future of the high performance computing industry is in cluster computing. Cluster computers are found in the Top 500 List [1] of the highest performance computers in the