1. Introduction to Parallel Computer Systems

1.1. Introduction

The term parallel computation is generally applied to any data processing, in which several computer operations can be executed simultaneously. Achieving parallelism is only possible if the following requirements to architectural principles of computer systems are met:

- **independent functioning of separate computer devices** – this requirement equally concerns all the main computer system components - processors, storage devices, input/output devices;
- **redundancy of computer system elements** – redundancy can be realized in the following basic forms:

  – *use of specialized devices* such as separate processors for integer and real valued arithmetic, multilevel memory devices (registers, cache);

  – *duplication of computer devices* by means of using separate processors of the same type or several RAM devices, etc.

Processor pipelines may be an additional form of achieving parallelism when carrying out operations in the devices is represented as executing a sequence of subcommands which constitute an operation. As a result, when such devices are engaged in computation several different data elements may be at different processing stages simultaneously.

Possible ways of achieving parallelism are discussed in detail in Patterson and Hennessy (1996), Culler and Singh (1998); the same works describe the history of parallel computations and give particular examples of parallel computers (see also Xu and Hwang (1998), Culler, Singh and Gupta (1998) Buyya (1999)).

Considering the problems of parallel computations one should distinguish the following modes of independent program parts execution:

– *Multitasking* (*time shared*) *mode*. In multitasking mode a single processor is used for carrying out processes. This mode is pseudo-parallel when only one process is active (is being carried out) while the other processes are in the stand-by mode queuing to use the processor. The use of time shared mode can make computations more efficient (e.g. if one of the processes can not be carried out because the data input is expected, the processor can be used for carrying out the process ready for execution - see Tanenbaum (2001)). Such parallel computation effects as the necessity of processes mutual exclusion and synchronization etc also manifest themselves in this mode and as a result this mode can be used for initial preparation of parallel programs;

– *Parallel execution*. In case of parallel execution several instructions of data processing can be carried out simultaneously. This computational mode can be provided not only if several processors are available but also by means of pipeline and vector processing devices;

– *Distributed computations*. This term is used to denote parallel data processing which involves the use of several processing devices located at a distance from each other. As the data transmission through communication lines among the processing devices leads to considerable time delays, efficient data processing

in this computational mode is possible only for parallel algorithms with low intensity of interprocessor data transmission streams. The above mentioned conditions are typical of the computations in multicomputer systems which are created when several separate computers are connected by LAN or WAN communication channels.

In this book we will discuss the second type of creating parallelism in multiprocessor computing systems.

1.2. Overview of parallel computer systems

The diversity of parallel computing systems is virtually immense. In a sense each system is unique. Such systems use various types of hardware: processors (Intel, IBM, AMD, HP, NEC, Cray, …), interconnection networks (Ethernet, Myrinet, Infiniband, SCI, …). They operate under various operating systems (Unix/Linux versions, Windows , …) and they use different software. It may seem impossible to find something common for all these system types. Obviously it is not so. Later we will try to formulate some well-known variants of parallel computer systems classifications based on some fundamental principles, but before that we will analyze some examples.

1.2.1. Supercomputers

The year 1976 when the first vector system Cray 1 came into being can be rightfully considered the beginning of supercomputer era. The results demonstrated by the first vector system were so impressive in comparison with those of the other systems in spite of a limited at that time set of applications that it was deservedly termed a "supercomputer". For a long period of time it was determining the development of the whole highly effective computation industry. However, the conjoint evolution of architectures and software caused the appearance of new systems on the market, - the systems with drastically differing features. Therefore the very concept "supercomputer" got more than one sense and had to be repeatedly revised.

The attempts to define the term *supercomputer* based on its processing power only inevitably lead to the necessity to continually raise the processing power standards to show the difference between a supercomputer and a work station or even a conventional desktop. Thus according to the definition given by Oxford Computer Dictionary 1986 to bear the glorious name of a supercomputer a computational machine should have the processing power 10 MFlops[1)]. It's well-known that nowadays the processing power of ordinary desktops is two orders higher.

Of all the alternative definitions the following two are the most interesting ones: the economic and the philosophical. The former says that a supercomputer is a system at the price of more than 1-2 million dollars. The latter declares that a supercomputer is a computer the power of which is only an order less than it is necessary for solving present days problems. From the general point of view we can define a supercomputer as a computer system which performance is the most powerful among all computers at some particular period of time.

1.2.1.1 ASCI Programme

One of the main purposes of the programme ASCI (Accelerated Strategic Computing Initiative – see http://www.llnl.gov/asci/) supported by US Department of Energy is the creation of a supercomputer with the performance of 100 TFlops.

The first ASCI system - **ASCI Red** developed by Intel Corp. in 1996 became the world first computer with the performance of 1 TFlops (later the performance was enhanced up to 3 TFlops).

Three years later **ASCI Blue Pacific** by IBM and **ASCI Blue Mountain** by SGI came into being. They became the first for that period of time supercomputers with performance f 3 TFlops.

Eventually in June 2000 the system **ASCI White** (http://www.llnl.gov/asci/platforms/white/), was brought into being. The peak performance was higher than 12 TFlops (the value of the performance which was actually demonstrated in LINPACK test was 4938 GFlops for that period of time; later it was enhanced up to 7304 GFlops).

ASCI White hardware is IBM RS/6000 SP system with 512 symmetric multiprocessor nodes (SMP). Each node has 16 processors. In total the system has 8192 processors. The system RAM is 4 TBytes, the capacity of the disk memory is 180 TBytes.

All system nodes are IBM RS/6000 POWER3 symmetric multiprocessors with 64 –bit architecture. Each node has its own memory, operating system, local disk and 16 processors.

[1)] MFlops – million of floating point operations per second, GFlops – billion, TFlops – trillion accordingly.

POWER3 processors are superscalar 64-bit pipeline chips with two floating point computing devices and three integer computing devices. They are able to execute up to eight instructions per clock cycle and up to four floating point instructions per clock cycle. The clock cycle of each processor is 375 MHz.

ASCI White software supports a mixed programming model which means message transmission among the nodes and multi-treading among an SMP node.

The operating system is a UNIX – IBM AIX version. AIX supports both 32 and 64-bit RS/6000 systems.

Parallel software environment on ASCI White includes parallel libraries, debuggers (in particular TotalView), profilers, IBM utility programs and systems for analyzing the execution efficiency. It also includes a MPI library, a compiler with the OpenMP support, a POSIX thread library and a translator of IBM directives. Moreover, there is also an IBM parallel debugger.

1.2.1.2 BlueGene supercomputer system

The most powerful supercomputer in the world for the time being has been created by IBM. To be more precise, it is still being developed. At present the full name of the system is "BlueGene/L DD2 beta-System". This is the first phase of the complete computer system. Its peak performance is forecasted to reach 360 TFlops by the time the system is put into operation.

The system developers consider hydrodynamics, quantum chemistry, climate modeling etc. to be the main areas of system application.

The features of the current system variant are the following:

- 32 racks with 1024 dual-kernel 32-bit PowerPC 440 0.7 GHz processors in each;
- peak performance is approximately 180 TFlops;
- the maximum performance demonstrated by LINPACK test is135 TFlops.

1.2.1.3 MVS-1000 System

One of the best known supercomputers in Russia Multiprocessor Computing System MVS-1000M is installed in Interdepartmental Supercomputer Center (ISC) of Russian Academy of Science.

It was being designed from April 2000 to August 2001.

According to the official data (http://www.jscc.ru) the system includes the following components:

- 384 dual-processor modules based on Alpha 21264 667 MHz (cache L2 4 Mb) assembled as 6 units with 64 modules in each;
- control server;
- NetApp F840 file server;
- Myrinet 2000 network;
- Fast/Gigabit Ethernet networks;
- Network monitor.

Each computational module has 2 Gb RAM, HDD 20 Gb, Myrinet (2000 Mbit/s) and Fast Ethernet (100 Mbit/s) interconnection networks.

When modules exchange data using MPI protocols in Myrinet the bandwidth in MCS-1000M is 110 - 150 Mb per second.

The system software includes:

- The operating systems is OS Linux RedHat 6.2 with SMP support;
- Parallel software environment – library MPICH for GM;
- Telecommunication network software (Myrinet, Fast Ethernet);
- Optimized compilers for the programming languages C, C++, FORTRAN by Compaq;
- Parallel program debugger TotalView;
- Software tools for parallel program profiling;
- Software tools for parallel managing.

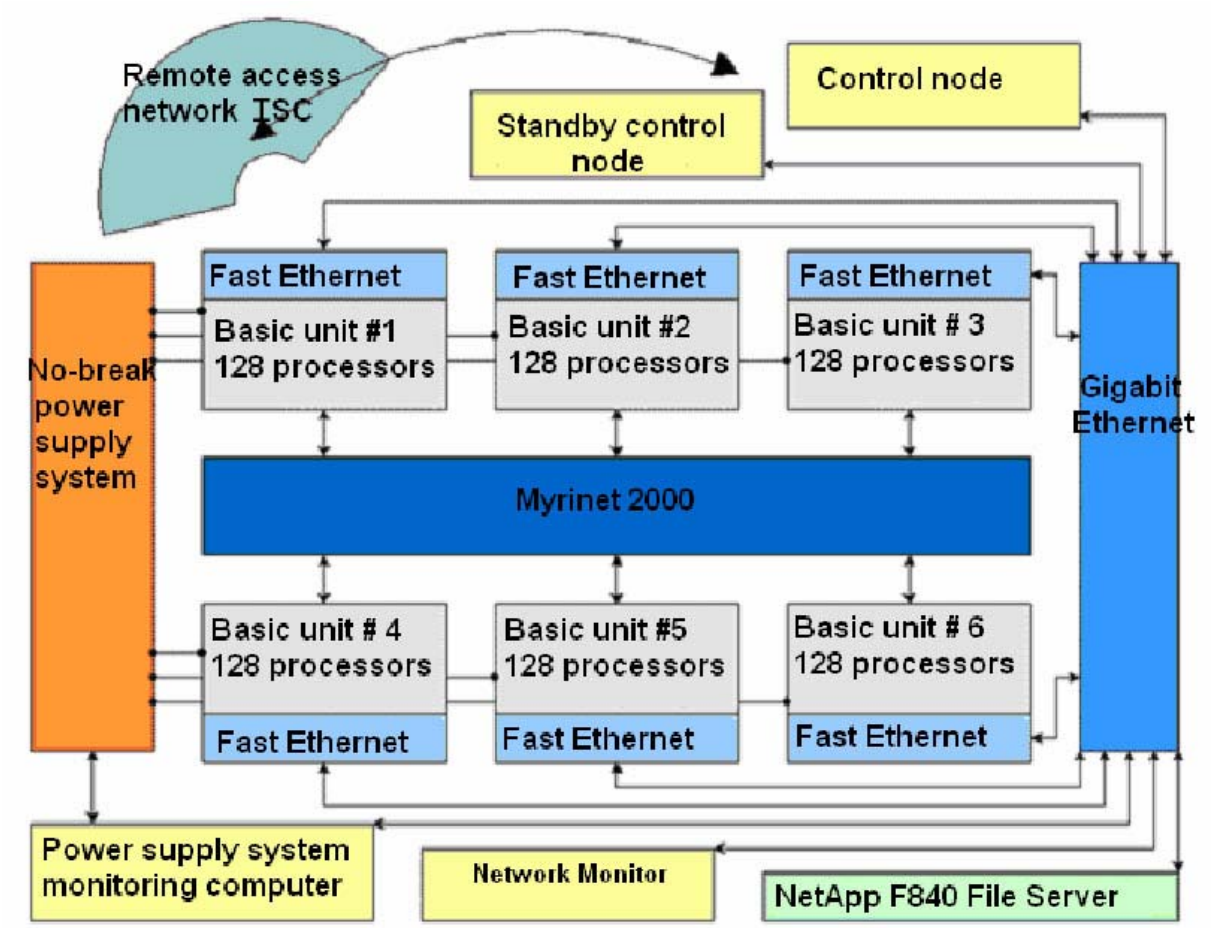

Figure 1.1. MVS-1000M Supercomputer

MVS-1000M is maintained by the two main components:

- Remote control and monitoring subsystem;
- Multiple access subsystem.

In the Top500 list in the summer of 2004 MVS-1000M with the peak performance 1024 GFlops and the maximum performance demonstrated in LINPACK test of 734 GFlops took the 391st position.

1.2.1.4 MVS-15000 System

Currently the most powerful supercomputer in Russia is being put into operation in ISC of Russian Academy of Science (according to the latest edition of Top50 list – http://supercomputers.ru/index.php).

MVS-15000 computational nodes includes:

- 2 IBM PowerPC 970 processors with clock frequency 2.2 GHz, cache L1 96 Kb and cache L2 512 Kb;
- 4 Gb RAM per node;
- 40 Gb IDE hard disc;
- 2 embedded Gigabit Ethernet adaptors;
- M3S-PCIXD-2-I Myrinet adaptor.

MVS-15000 software configuration:

- SuSe Linux Enterprise Server operating systems, version 8 for the platforms x86 and PowerPC;
- GM 2.0.12 package as Myrinet communication environment;
- MPICH-GM package as parallel software environment;
- Software for job management and batch processing.

Currently (at the beginning of 2005) MVS-15000 system has the total number of nodes 276 (552 processors), peak performance 4857.6 GFlops and the maximum one demonstrated in LINPACK test 3052 GFlops.

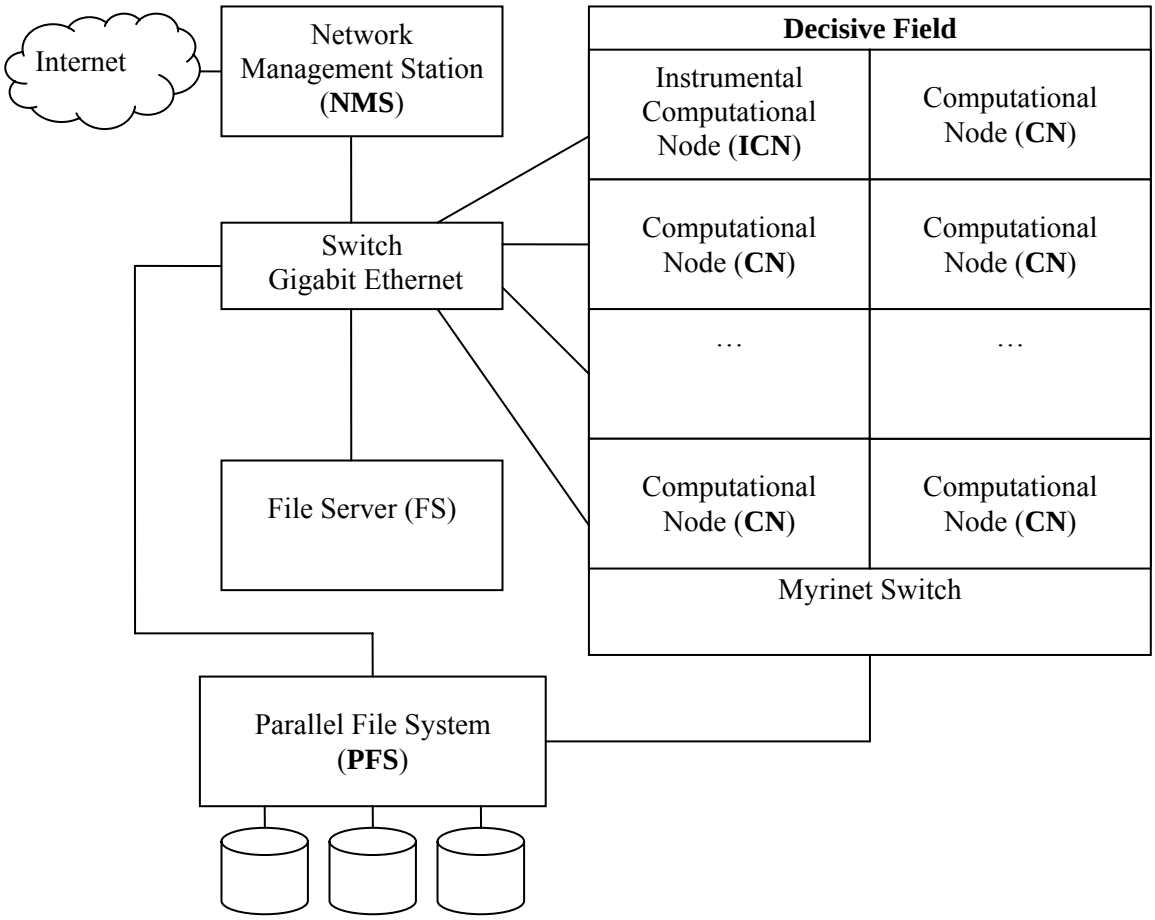

Figure 1.2. MVS-15000 Supercomputer structural scheme

1.2.2. Clusters

A cluster is group of computers connected in a local area network (LAN). A cluster is able to function as a unified computational resource. It implies higher reliability and efficiency than an LAN as well as a considerably lower cost in comparison to the other parallel computing system types (due to the use of standard hardware and software solutions).

The beginning of cluster era was signified by the first project with the primary purpose of establishing connection among computers - ARPANET[2] project.

That was the period when the first principles were formulated which proved to be fundamental. Those principles later lead to the creation of local and global computational networks and of course to the creation of world wide computer network, the Internet. It's true however that the first cluster appeared more than 20 years later.

Those years were marked by a giant breakthrough in hardware development, the emergence of microprocessors and PCs which conquered the market, the accretion of parallel programming concepts and techniques, which eventually lead to the solution to the age-long problem, the problem of each parallel computational facility unicity which was the development of standards for the creation of parallel programs for systems with shared and distributed memory. In addition to that the available solutions in the area of highly efficient systems were very expensive at that time as they implied the use of high performance and specific components. The constant improvement of PC cost/performance ratio should also be taken into consideration. In the light of all those facts the emergence of clusters was inevitable.

[2] ARPANET – the project of Advanced Research Projects Agency, DARPA US Department of Defense aimed at creating a computer network for carrying out experiments in the area of computer communications, maintaining communications in case of nuclear war, developing decentralized management concept (1966 – 1969).

The advantages of the new approach to the creation of very powerful computing systems which were recognized right after the first such system was introduced have only increased since that time supported by the constant increase of standard component performance.

Currently clusters account for the greater part of TOP500 list of the most powerful systems (294 installations).

1.2.2.1 Beowulf Cluster

The first cluster in the world was probably the one created under Thomas Sterling and Don Becker's supervision in the NASA research center – Goddard Space Flight Center – in the summer of 1994. The cluster named after the hero of a Scandinavian saga who had the strength of thirty people, consisted of 16 computers based on 486DX4 processors with 100 MHz. clock frequency. Each node had 16 Mb RAM. The connection of the nodes was provided by three 10Mbit/s network adaptors operating in parallel. The cluster functioned under the control of Linux operating system and used GNU compiler and supported MPI based parallel programs. Cluster nodes processors were too fast in comparison to the conventional Ethernet network bandwidth. That is why to balance the system Don Becker rewrote Ethernet drivers for Linux to create duplicated channels and distribute the network traffic.

The idea "to assemble the supercomputer with one's own hands" became popular first of all among academic communities. The use of standard ("commodity off-the-shelf") hardware and software components produced on a large scale led to a considerable decrease of system development and implementation cost. In addition the performance of the computational system was sufficient to solve a considerable number of problems which required a big amount of computations. "Beowulf cluster" systems began to appear all over the world.

Four years later in Los Alamos national laboratory (US) the astrophysicist Michael Warren and some other scientists from the group of theoretical astrophysics designed **Avalon** computer system. It was a Linux cluster on the basis of Alpha21164A processors with the clock frequency 533 MHz. Originally it included 68 processors. Later it was expanded up to 140. Each node contained 256 Mb RAM, 3 Gb HDD, Fast Ethernet card. The total cost of the Avalon project slightly exceeded USD 300 000.

At the moment when it was put into operation in Autumn 1998 the cluster with the peak performance 149 GFlops and the one of the 48.6 GFlops demonstrated in LINPACK test occupied the 113d position on Top500 list.

The same year the Avalon creators presented the report "Avalon: An Alpha/Linux Cluster Achieves 10 GFlops for $150k" at the most prestigious conference in the area of highly efficient computations Supercomputing'98. The report was awarded the first prize in the nomination "The best price/performance ratio".

Nowadays a "Beowulf" type cluster is a system which consists of a server node and one or more client nodes which are connected with the help of Ethernet or some other network. The system is made of ready-made mass manufactured ("commodity off-the-shelf") components able to operate under Linux or Windows, standard Ethernet adaptors and switches. It does not contain any specific hardware and can be easily reproduced. The server node manages the cluster and serves as a file-server for client nodes. It is also the cluster console and the communication server for an external network. Large Beowulf systems can have more than one server node, they can also have specialized nodes, for instance, consoles of monitoring stations. In most cases the client nodes in Beowulf are passive. They are configured and managed by the server nodes and execute only what has been assigned to them by the server node.

1.2.2.2 AC3 Velocity Cluster

AC3 Velocity Cluster installed in Cornell University, US, (http://www.tc.cornell.edu) was the result of the university collaboration with AC3 (Advanced Cluster Computing Consortium) established by Dell, Intel, Microsoft, Giganet and 15 more software manufacturers for the purpose of integrating different technologies and designing cluster architectures for educational and state institutions.

Cluster structure:

- 64 four-way servers Dell PowerEdge 6350 based on Intel Pentium III Xeon 500 MHz, 4 GB RAM, 54 GB HDD, 100 Mbit Ethernet card;
- 1 eight-way server Dell PowerEdge 6350 based on Intel Pentium III Xeon 550 MHz, 8 GB RAM, 36 GB HDD, 100 Mbit Ethernet card.

The four-way servers are rack-mounted in eights and work under the operating system Microsoft Windows NT 4.0 Server Enterprise Edition. The connection among the servers operates at the rate of 100 Mbyte per second through Cluster Switch by Giganet Company.

The jobs in a cluster are managed by Cluster ConNTroller developed in Cornell University. AC3 Velocity peak performance is 122 GFlops and the cost is 4-5 times less than the one of a supercomputer with the analogous characteristics.

In the summer of 2000 the cluster with the performance 47 GFlops demonstrated in LINPACK test occupied the 381st position on TOP500 list.

1.2.2.3 NCSA NT Supercluster

The cluster was designed in 2000 in NCSA (National Center for Supercomputing Applications) on the basis of Hewlett-Packard Kayak XU PC workstation (http://www.hp.com/desktops/kayak/). Microsoft Windows was chosen as the operating system for the cluster. The designers called it "NT Supercluster" (http://archive.ncsa.uiuc.edu/SCD/Hardware/NTCluster/).

When it was put into operation the cluster with the performance 62 GFlops demonstrated in LINPACK test and the peak performance 140 GFlops took the 207th position of TOP500.

The cluster is made of 38 two-way nodes on the basis of Intel Pentium III Xeon 550 MHz, 1 Gb RAM, 7.5 Gb HDD, 100 Mbit Ethernet card.

The node connection is based on Myrinet (http://www.myri.com/myrinet/index.html).

The cluster software:

- Operating system – Microsoft Windows NT 4.0;
- Compilers – the Fortran77, C/C++ languages;
- Message passing level is based on HPVM (http://www-csag.ucsd.edu/projects/clusters.html).

1.2.2.4 Thunder Cluster

At present the number of systems on the basis of Intel processors and represented on the TOP500 list is 318 items. The most powerful supercomputer which is a cluster based on Intel Itanium2 is installed in Livermore National Laboratory in the USA.

Thunder cluster hardware configuration (http://www.llnl.gov/linux/thunder/):

- 1024 servers with 4 Intel Itanium 1.4 GHz processors in each;
- 8 Gb RAM per node;
- total disc capacity 150 Tb;

Software includes:

- operating system CHAOS 2.0;
- parallel library MPICH2;
- Parallel debugger TotalView;
- compilers Intel и GNU Fortran, C/C++.

At present Thunder Cluster with its performance 22938 GFlops and the maximum one shown in LINPACK test 19940 GFlops takes the 5th position of the Top500 (in the summer of 2004 it occupied the 2nd position).

1.2.3. NNSU Computational Cluster

The computational cluster of Nizhni Novgorod State University the equipment for which was donated to the University in 2001 within the frame of Intel Academic program is another sample of clusters to be considered in this chapter. The cluster is composed of (Figure 1.3):

– 2 computational servers, each has 4 processors Intel Pentium III 700 Mhz, 512 MB RAM, 10 GB HDD, 1 Gbit Ethernet card;

– 12 computational servers, each has 2 processors Intel Pentium III 1000 Mhz, 256 MB RAM, 10 GB HDD, 1 Gbit Ethernet card;

– 12 workstations based on Intel Pentium 4 1300 Mhz, 256 MB RAM, 10 GB HDD, 10/100 Fast Ethernet card.

Heterogeneity is an essential distinctive feature of the cluster. The cluster includes workstations equipped with Intel Pentium 4 and connected with a relatively slow network. It also includes 2- and 4-way servers the data exchange between which is executed with the help of fast data transmission channels (1000Mbit/s). As a result the cluster can be used not only for solving complicated time-consuming computational problems but also for carrying out various experiments concerned with the analysis of multiprocessor cluster systems and parallel methods of solving research and engineering problems.

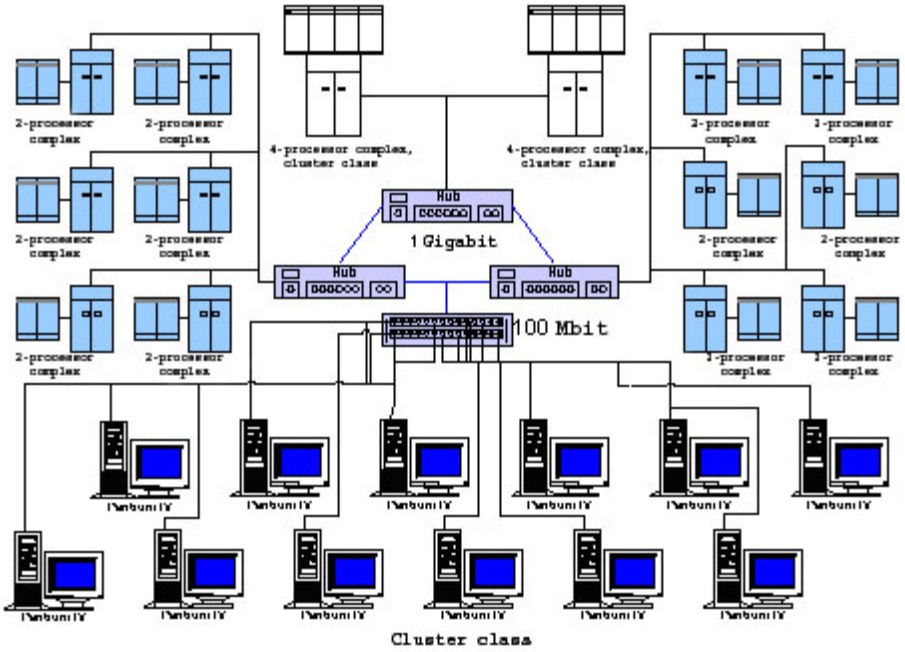

Figure 1.3. Structure of computing cluster in NNSU

Operating systems of Microsoft Windows family have been chosen as the system platform for the cluster design (there is a possibility to use the operating system Unix for carrying out certain experiments). The choice of the solution was determined by a number of reasons. The most important are the following ones:

– Microsoft Windows family operating systems (as well as Unix operating systems) are widely used for designing clusters; it can be noted the operating system Unix was the prevailing system solution, the present days tendency is to create clusters that work under the operating system Microsoft Windows (see for instance www.tc.cornell.edu/ac3/, www.windowclusters.org etc.);

– Software application development is mainly carried out with the use of Microsoft Windows;

– Microsoft Corporation donated all the necessary software to the University (OC MS Windows 2000 Professional, OC MS Windows 2000 Advanced Server etc.).

The cluster software determined by this solution is the following:

– Computing servers work under the operating system Microsoft® Windows® 2000 Advanced Server; Microsoft® Windows® 2000 Professional is installed at the workstations;

– Microsoft® Visual Studio 6.0 is used as the development environment; Intel® C++ Compiler 5.0 built in Microsoft® Visual Studio can be used for carrying out research experiments;

– The following libraries are installed at the workstations;

  - Plapack 3.0 (see www.cs.utexas.edu/users/plapack);
  - MKL (see developer.intel.com/software/products/mkl/index.htm);

– The two MPI standard implementations are installed as the data transmission tools;

  - Argonne MPICH (www.unix.mcs.anl.gov/mpi/MPICH/);
  - MP-MPICH (www.lfbs.rwth-aachen.de/~joachim/MP-MPICH.html).

– The system of parallel program development DVM (www.keldysh.ru/dvm/) is in experimental testing.

1.3. Classification of Computer Systems

Flynn's systematics is one of the most widely used methods of computer classification. In terms of Flynn's systematics the emphasis in analyzing the computing system architecture is laid on the interaction of the streams of executed instructions and the streams of processed data. The approach leads to the following differentiation between the main system types (Flynn (1996), Patterson and Hennessy (1996)):

- **SISD** (Single Instruction, Single Data) – the system where there is a single instruction stream and a single data stream; conventional sequential computers can be referred to this type;
- **SIMD** (Single Instruction, Multiple Data) – the system with the single instruction stream and multiple data stream. This class comprises multiprocessor computer systems in which the same instruction for processing

several data components can be executed at every instant. Multiprocessor systems with a single control unit have this type of architecture. The approach was widely used (ILLIAC IV system or CM-1 by Thinking Machines). Recently the use of the approach has been restricted and now it is mainly to develop specialized systems;

- **MISD** (Multiple Instruction, Single Data) – the systems characterized by a multiple instruction stream and a single data stream. There is much controversy about this type of system: some experts say that the computers of the kind do not exist yet and introduce this class only to make the classification complete. Others refer to this type systolic computing systems (Kung (1982), Kumar et al. (1994)) or pipeline data processing systems.

- **MIMD** (Multiple Instruction, Multiple Data) – the systems with multiple instruction stream and multiple data stream. The majority of parallel multiprocessor computer systems belong to this class.

It is to be noted that Flynn's systematics though widely used to specify the computer system types leads to the situation when all the parallel systems despite their considerable heterogeneity belong to the same group MIMD. As a result many researchers attempted to make Flynn's systematic more detailed. For example, for MIMD class a widely recognized structural scheme was proposed (see Xu and Hwang (1998), Buyya (1999)). According to the scheme the further classification of multiprocessor systems was based on the ways of RAM arrangements used in such systems (see figure 1.4). This approach allows differentiating between the two important multiprocessor system types: *multiprocessors* or systems with shared memory and *multicomputers* or systems with distributed memory.

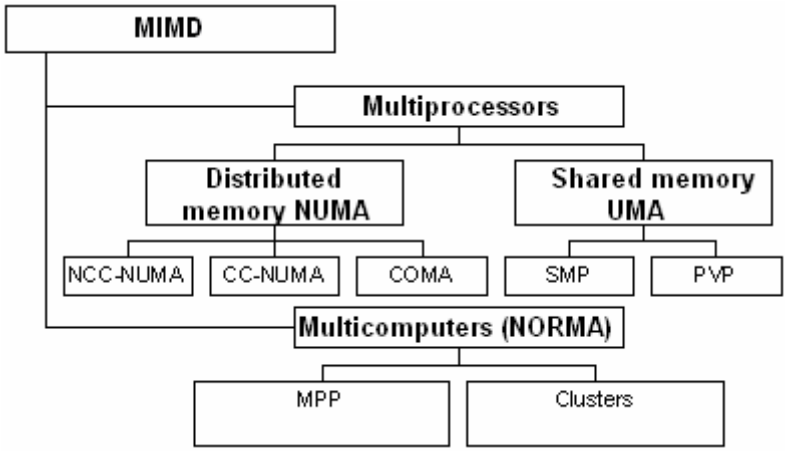

Figure 1.4. Multiprocessor computing systems classification

1.3.1. Multiprocessors

The way of shared memory arrangement is taken into account to make the classification of multiprocessors more detailed. A viable solution is the use of single centralized shared memory – Figure 1.5. This approach ensures *uniform memory access* or *UMA* and serves as the basis for designing *parallel vector processor* or *PVP* and *symmetric multiprocessor* or *SMP*.

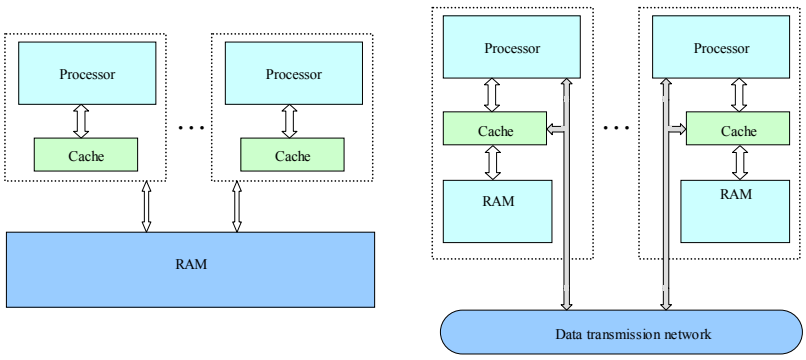

Figure 1.5. Multiprocessor shared memory system architecture**:** (a) uniform memory access systems and (b) non-uniform memory access systems

The first group can be illustrated by Cray T90 supercomputer, the second by IBM eServer, Sun StarFire, HP Superdome, SGI Origin etc.

One of the main problems arising in arrangement of parallel computations in such systems is access to the shared data from different processors and providing in this connection the coherence of different cache contents (*cache coherence problem*). The fact is that if data are shared the copies of the same variable values may appear in the caches of different processors. If in this situation one of the processors changes the value of the shared variable the values of the copies in the caches of the other processors will not correspond to reality and their use will lead to incorrect calculations. Cache coherence is usually provided at hardware level. After the change of shared variable value all the copies of this variable in caches are marked as invalid. The subsequent access to the variable will necessarily require a RAM access. It should be also noted that the necessity to provide coherence leads to some decrease of computation speed and hinders the creation of systems with a considerable number of processors.

The availability of shared data in executing parallel computations causes the necessity to synchronize the interactions of simultaneously carried out instruction streams. Thus for instance if the change of shared data requires some sequence of actions it is necessary to provide *mutual exclusion* so that the change could be performed by only one instruction stream at any moment. Mutual exclusion and synchronization problems are referred to the classical problems. Consideration of such problems is one of the key issues of parallel programming.

Common access to the data may be also provided in case of physically distributed memory (naturally in this case access duration will not be the same for all memory elements) – Figure 1.5. This approach is also called *non-uniform memory access or NUMA*. The systems with such memory type fall into the following groups:

– The systems which use solely the cache of the available processors for data placement (*cache-only memory architecture or COMA*); the examples of such systems are KSR-1 and DDM;

– The systems in which local cache coherence of different processors is provided (*cache-coherent NUMA or CC-NUMA)*; among such systems there are SGI Origin 2000, Sun HPC 10000, IBM/Sequent NUMA-Q 2000;

– The systems in which common access to local memory of different processors is provided without support at the cache coherence hardware level (*non-cache coherent NUMA or NCC-NUMA*); Cray T3E system is an example of this type.

The use of *distributed shared memory* or *DSM* simplifies the problems of multiprocessor creation. Nowadays one can come across systems with several thousands processors. However, the rising problems of distributed shared memory efficient use (access time in case of access to local and remote memory may be several orders different) causes a significant increase of parallel programming complexity.

1.3.2. Multicomputers

Multicomputers (multiprocessor systems with distributed memory) are already unable to provide common access to all available in systems memory (*no-remote memory access or NORMA*) – see Figure1.6. Though this architecture is similar to distributed shared memory systems (Figure 1.5b) multicomputers have the following principle distinction – each system processor is able to use only its local memory while getting access to the data available on other processors requires explicit execution of *message passing operations*. This approach is used in developing two important types of multiprocessor computer systems (Figure 1.4) - *massively parallel*

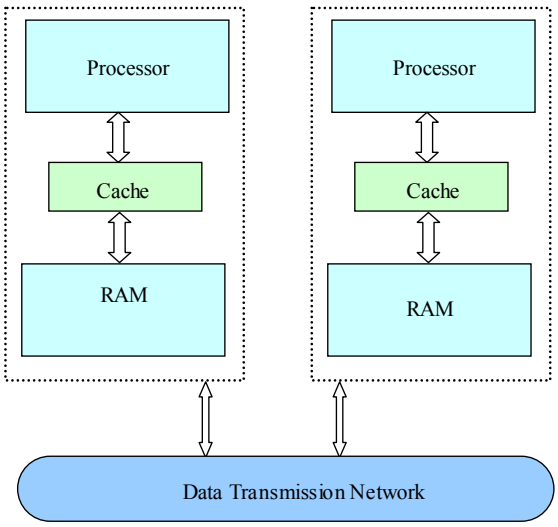

Figure 1.6. Architecture of multiprocessor systems with distributed memory

processor or MPP and *clusters*. IBM RS/6000 SP2, Intel PARAGON, ASCI Red, Parsytec transputer system etc. are representatives of the first type. The examples of clusters are AC3 Velocity systems and NCSA NT Supercluster.

Cluster type multiprocessor computer systems are rapidly developing. The general features of this approach are given for instance in the review edited by Barker in 2000. The cluster is usually defined (see Xu and Hwang (1998), Pfister (1998)) as a set of separate computers connected into a network. Single system image, availability of reliable functioning and efficient performance for these computers are provided by special software and hardware. Clusters can be either created on the basis of separate computers available for consumers or constructed of standard computer units, this allows to cut down on costs. The use of cluster can also contribute to the solution of problems related to parallel algorithm and software development as the increase of computational power of separate processors makes possible to create clusters using a relatively small number (several tens) of separate processors (*lowly parallel processing*). It allows separation of only large independent calculation parts (*coarse granularity*) in the solution algorithm of computational problems for parallel execution and, therefore, simplification of the complexity of parallel methods used and a decrease of the transmitted data streams among the cluster processors. But it should be noted that arranging the computational cluster nodes interaction with the help of data transmission usually leads to considerable time delays which mean additional restrictions for the type of parallel algorithm and program being developed.

Some experts purposefully emphasize the difference between the cluster and the *network of workstations or NOW* concepts. The data transmission networks used to create a local computer network are simpler (in the order of 100Mbit/sec) than those used to create a cluster. The network computers are usually more distributed and can be used to perform some additional jobs.

In conclusion we should also mention that there are other computing systems classifications (a rather complete review of different approaches is given in Patterson and Hennessy (1996)). It would be also useful studying this aspect of parallel computations to pay special attention to the method of structural notation for computer architecture description which permits to describe many typical features of computer systems with a high degree of accuracy.

1.4. Overview of Interconnection Networks

Data transmission among the processes of computing environment is used to provide interaction, synchronization and mutual exclusion of executed parallel processes in arranging parallel computations. Time delays in data transmission over the communication lines may be considerable in comparison to the processor operation speed. As a result the communication complexity of the algorithm has a considerable influence on the choice of parallel problem solution methods.

1.4.1. Topologies of Interconnection Networks

The structure of communication lines among the computing system processors (data transmission network topology) is determined as a rule with due account for the possibilities of efficient technical implementation. The analysis of the information stream intensity in parallel solution of the most widely spread computing problems

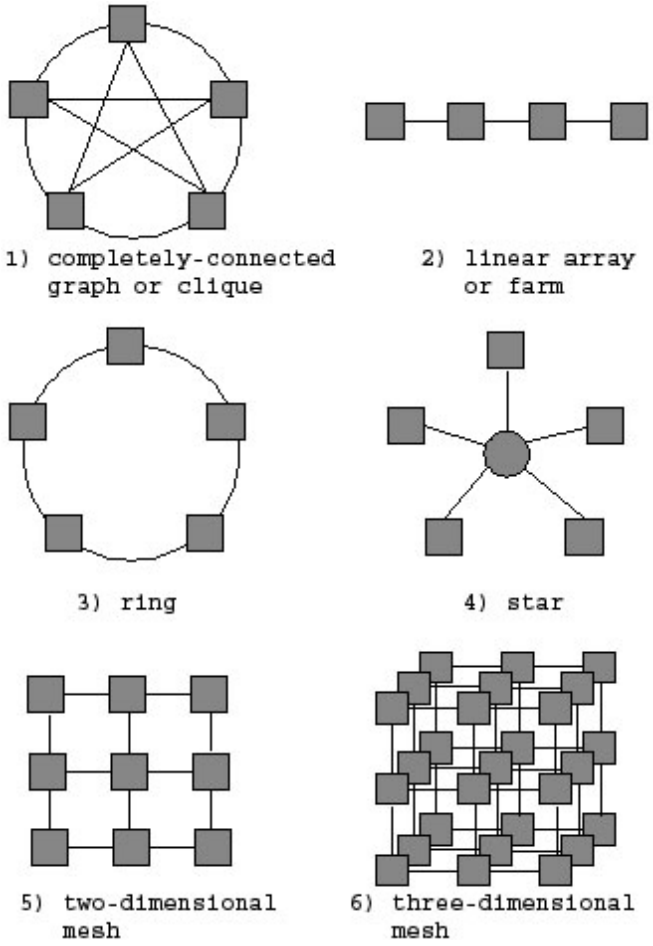

Figure 1.7. Multiprocessor computing systems topologies

also plays an important part in choosing the network structure. The following processor communication schemes are usually referred to the basic topologies (Figure 1.7):

- ***Completely-connected graph* or *clique topology*.** It is a system where each pair of processors is connected by means of a direct communication line. As a result this topology requires minimum costs in data transmission. However it is very difficult to construct when the number of processors is big enough;
- ***Linear array* or *farm topology***. In this system all the processors are enumerated in order and each processor except the first and the last ones has communication lines only with the adjacent (the preceding and the succeeding) processors. This system is easy to implement. It also corresponds to the structure of data transmission in solving many computing problems for instance in arranging pipeline computations;
- ***Ring topology.*** This topology can be derived from a linear array if the first processor of the array is connected to the last one;
- ***Star topology***. All the processors in this system are connected by means of communication lines to some managing processor. This topology is very effective for instance in arranging centralized parallel computation schemes;
- ***Mesh topology***. The graph of the communication lines of the this systems creates a rectangular mesh (usually two- or three-dimensional one); this topology is rather easy to implement and besides it may be very efficient in parallel execution of many numeric algorithms (i.e. in implementing the method of the analysis of mathematical models described as differential equations in partial derivatives);
- ***Hypercube topology.*** This topology is a particular case of mesh topology. In the hypercube there are only two processors on each mesh dimension (that is a hypercube contains 2^N processors when the dimension is equal to N). This option of data transmission network arrangement is widely used in practice and has the following distinct features:
  - two processors have a connection if the binary representations of their numbers have only one distinctive position;
  - in a N - dimensional hypercube each processor is connected with N neighbors precisely;
  - an N -dimensional hypercube can be divided into two $(N\text{-}1)$ -dimensional hypercubes (altogether there are N possible divisions);

– the length of the shortest distance between any two processors coincides with the number of differing bits in the processors numbers (this value is know as *the Hamming distance*)

Additional information on topologies of multiprocessor computer systems may be found in Patterson and Hennessy (1996), Culler and Singh (1998), Xu and Hwang (1998), Buyya (1999). It should be taken into account that data transmission line scheme may be implemented at hardware level. It can also be provided on the basis of the available physical topology through corresponding software. The introduction of virtual (software realized) topologies contributes to the developed parallel programs mobility and reduces software development costs.

1.4.2. Cluster Network Topology

In many cases a switch through which all cluster processors are connected with each other is used to build a cluster system. In this case the cluster network topology is a completely connected graph (Figure 1.7). In this case the data transmission may be organized between any two network processors. However the simultaneity of several switching operation execution is limited – at any given moment of time each processor can participate only in one data transmission operation. As a result, only these switching operations in which the interacting operation of a pair of processors do not overlap each other can be executed in parallel.

1.4.3. Network Topology Characteristics

The following characteristics are typically used as the basic data transmission network topology:

- *Diameter.* This characteristic is determined as the maximum distance between two network processors (the distance is the value of the shortest path between processors). This value can characterize the maximum time necessary to transmit the data between processors as the data transmission time is usually directly proportional to the path length;
- *Connectivity.* This parameter characterized the existence of the different data transmission routes among network processors. The concrete type of this parameter can be determined for instance as the minimum number of arcs which have to be necessary removed for partitioning the data transmission network into two disconnected parts;
- *Bisection width.* This value is defined as the minimum number of arcs which have to be obligatory eliminated for partitioning the data transmission network into two disconnected parts of the same size;
- *Cost*. This parameter can be determined as the total number of data transmission lines in a multiprocessor computing system.

Table 1.1 gives the values of the characteristics described above for different data transmission network topologies.

Table1.1. Data transmission network topology characteristics
(***p*** – the number of processors)

Topology	**Diameter**	**Bisection Width**	**Connectivity**	**Cost**
Completely connected graph	1	$p^2/4$	p-1	p(p-1)/2
Star	2	1	1	p-1
Complete binary tree	2log((p+1)/2)	1	1	p-1
Linear array	p-1	1	1	p-1
Ring	⌊p/2⌋	2	2	p
Mesh N=2	$2(\sqrt{p}-1)$	$\sqrt{p}$	2	2(p-$\sqrt{p}$)
Torus-mesh N=2	$2\lfloor\sqrt{p}/2\rfloor$	$2\sqrt{p}$	4	2p
Hypercube	log p	p/2	log p	(p log p)/2

1.5. Overview of Cluster System Platforms

To be appeared.

1.6. Summary

In this chapter we have discussed the general characteristics of parallel computation arrangement ways and have shown the difference between multitask, parallel and distributed program execution modes. To illustrate the possible approaches we have considered a number of parallel computer systems that allows to make a conclusion about the considerable diversity of parallel system architectures.

The diversity of the computer systems caused the necessity to classify them. In this chapter we have described one of the best-known classification ways – *Flynn's systematics* - based on data stream and instruction stream concepts. The classification is clear enough. However in the framework of this approach all multiprocessor computer systems fall in one and the same group – MIMD class. For further classification of possible system types we have also considered the widely used multiprocessor computer system classification which makes possible to single out two important system groups: the systems with shared and distributed memory - *multiprocessors* and *multicomputers*. The best known samples of the first group are *parallel vector processors or PVPs and symmetric multiprocessors or SMPs. Massively parallel processors or MPPs and clusters* refer to *multicomputers.*

Further in this chapter we have focused on data transmission network characteristics in multiprocessor computer networks. We have discussed the samples of network topologies and shown the peculiarities of data transmission organization in clusters. We have also described the topology parameters which significantly influence the communication complexity of parallel computation methods.

In conclusion we have given the general description of system platforms for cluster design.

1.7. References

Additional information on parallel computer system architecture can be found in Patterson and Hennessy (1996), Culler, Singh and Gupta (1998); useful information is also provided in the works by Xu and Hwang (1998), Buyya (1999).

To review the possible data transmission network topologies in multiprocessor systems and the methods of the implementations we can recommend for instance the work by Dally and Towles, B.P. (2003).

The issues concerned with cluster computing system development and use are considered in detail in Xu and Hwang (1998), Buyya (1999). The works by Sterling (2001, 2002) give some practical recommendations on cluster design for various platform systems.

1.8. Discussions

1. What are the basic ways to achieve parallelism?
2. What are the possible differences of parallel computing systems?
3. What is Flynn's classification based on?
4. What is the essence of multiprocessor systems subdivision into multiprocessors and multicomputers?
5. What types of systems are known as multiprocessors?
6. What are the advantages and disadvantages of symmetric multiprocessors?
7. What system types are known as multicomputers?
8. What are the advantages and disadvantages of cluster systems?
9. What data transmission network topologies are widely used in multiprocessor systems development?
10. What are the peculiarities of data transmission networks for clusters?
11. What are the basic characteristics of data transmission networks?
12. What system platforms can be used for cluster design?

1.9. Exercises

1. Give some additional examples of parallel computer systems.
2. Consider some additional methods of computer systems classification.
3. Consider the ways of cache coherence provision in the systems with common shared memory.
4. Make a review of the software libraries which provide carrying out data transmission operations for the systems with distributed memory.

5. Consider the binary tree data transmission network topology.
6. Choose efficiently realized problems for each data transmission network topology type.

References

Barker, **M.** (Ed.) (2000). Cluster Computing Whitepaper at http://www.dcs.port.ac.uk/~mab/tfcc/WhitePaper/.

Buyya, R. (Ed.) (1999). High Performance Cluster Computing. Volume1: Architectures and Systems. Volume 2: Programming and Applications. - Prentice Hall PTR, Prentice-Hall Inc.

Culler, D., Singh, J.P., Gupta, A. (1998) Parallel Computer Architecture: A Hardware/Software Approach. - Morgan Kaufmann.

Dally, W.J., Towles, B.P. (2003). Principles and Practices of Interconnection Networks. - Morgan Kaufmann.

Flynn, M.J. (1966) Very high-speed computing systems. Proceedings of the IEEE 54(12): P. 1901-1909.

Hockney, R. W., Jesshope, C.R. (1988). Parallel Computers 2. Architecture, Programming and Algorithms. - Adam Hilger, Bristol and Philadelphia.

Kumar V., Grama A., Gupta A., Karypis G. (1994). Introduction to Parallel Computing. - The Benjamin/Cummings Publishing Company, Inc. (2nd edn., 2003)

Kung, H.T. (1982). Why Systolic Architecture? Computer 15 № 1. P. 37-46.

Patterson, D.A., Hennessy J.L. (1996). Computer Architecture: A Quantitative Approach. 2d ed. - San Francisco: Morgan Kaufmann.

Pfister, G. P. (1995). In Search of Clusters. - Prentice Hall PTR, Upper Saddle River, NJ (2nd edn., 1998).

Sterling, T. (ed.) (2001). Beowulf Cluster Computing with Windows. - Cambridge, MA: The MIT Press.

Sterling, T. (ed.) (2002). Beowulf Cluster Computing with Linux. - Cambridge, MA: The MIT Press.

Tanenbaum, A. (2001). Modern Operating System. 2nd edn. – Prentice Hall

Xu, Z., Hwang, K. (1998). Scalable Parallel Computing Technology, Architecture, Programming. – Boston: McGraw-Hill.