Features of Implementation of High-Performance Clusters and Their Parallelization*

Evgenii I. Milenin[0000-0003-1364-828X] and Maksim A. Davydov[0000-0002-0977-0824]

ITMO University, Kronverkskiy prospekt, 49, St. Petersburg, 197101, Russia
monser2002@gmail.com

Abstract. The article describes the concept of cluster systems and the principle of cloud computing, the classification groups of clusters, the principle of implementation of high-performance clusters, the analysis of existing high-speed solutions and the calculation time of the problem depending on the number of processors involved in the cluster.

Keywords: cluster, computing complex, performance, availability, data rate, packages, interface, architecture, requirements, design.

1 The concept of cluster systems

The aim of the study is to determine how the number of involved processors in the cluster affects the calculation time of the problem, to conduct a comparative characteristic of high-speed interfaces and to determine the advantages and disadvantages of possible means of parallelization in the cluster. Before you begin your research, it is important to determine what a "cluster" is.

Such a concept as "cluster" was first defined by Digital Equipment Corporation in the list of classifications of computer systems. As defined by Digital Equipment Corporation, a "cluster" is a group of interconnected computing machines whose purpose is to function as a common single node processing information for a specific task. The cluster is able to function as a unified system, which means that the user or application task of the cluster will be presented as a single computer, but in fact it is a set of computers. [1]

Initially, clusters at Digital Equipment Corporation were designed and deployed on the Virtual Address eXtension (VAX) in the mid-1970s. The 32-bit architecture of the VAX machine was a logical way to develop the PDP-11 line, which was developing within the framework of the Star project [2]. These clusters are no longer in production, but they are still found in working condition and perform their tasks where they were installed almost half a century ago. And the main thing that is worth noting is that the General principles that were laid down in the process of their construction, still remain relevant in the design of cluster systems today.

The main requirements for the design of cluster systems are high availability (high-availability), high performance (high-performance), scaling (scaling), availability of resources (resource availability) and ease of maintenance (ease of maintenance). Obviously, you cannot create a cluster in which each of the requirements will be a benchmark of quality. Often it is necessary to choose between the quality of each of the characteristics of the cluster, depending on the requirements for it, which are formed on the basis of the problem solved by the cluster. As an example, we can cite a situation where an important criterion for the task is speed, and in order to save on resources in the design of high readiness neglected, paying much less attention to it. [3]

In a global sense, the cluster should function as a multiprocessor system. In this regard, it is extremely important to be able to determine the classification of these systems in the field of distribution of software and hardware resources. There are three types of multiprocessor systems. Closely related multiprocessor system combines a group of interconnected computers in a common cluster that performs a specific application task. (Figure 1.)

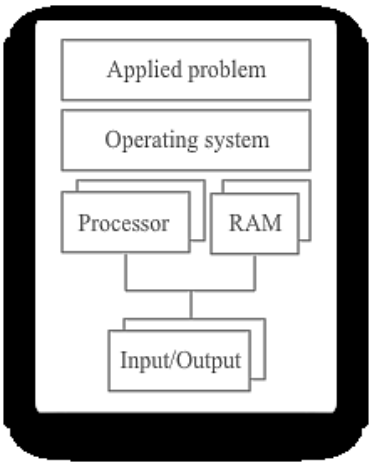

Fig. 1. Closely related multiprocessor system

A moderately coupled multiprocessor system combines a group of interconnected computers into several clusters that perform a common application task. (Figure 2.)

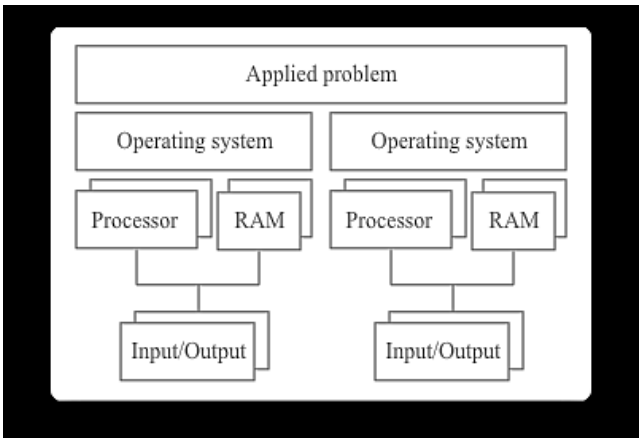

Fig. 2. Closely related multiprocessor system

Loosely coupled multiprocessor system is a set of clusters, each of which performs its intended application task. (Figure 3.)

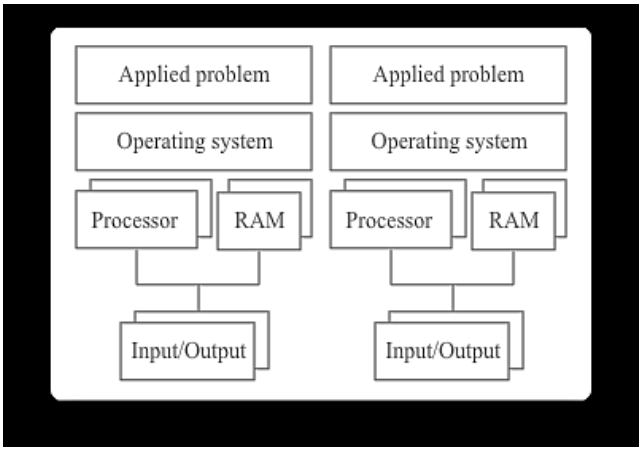

Fig. 3. Closely related multiprocessor system

2 High performance clusters

Varieties of clusters can be divided entirely into three large classification groups – high availability clusters (High Performance, HP), high performance clusters (High Availability, HA) and mixed systems.

High availability clusters, or as they sometimes say – high availability, are used in those situations and for those tasks where the damage from possible downtime of the system is higher than the cost of designing and building a cluster system. These can be

areas such as electronic Commerce and banking transactions, systems for the management of large companies and different billing systems. [4]

Clusters of high performance typically are characterized by embedded in them the enormous computing potential. Such clusters can be used for tasks, the main requirement of which is a large computing power. These may include areas such as automatic image processing, such as facial recognition; various types of research – biochemistry, bio-computer science, genetics, physics and astronomy; and, for example, mathematical modeling in industry. [5]

As for mixed systems – they combine the features of both high-performance and high-availability clusters. It is important to understand that mixed systems are not able to be more effective than each of the nested cluster types. According to both criteria, the mixed system will be significantly inferior to highly specialized types of clusters, respectively.

3 Tools for the implementation of clusters of high performance

Before you start designing any type of cluster, it is important to identify the set of communication technologies that you plan to use to communicate with the internal systems of the cluster architecture and, in General, for deployment. One of the most popular today communication technologies used to build supercomputers DAS (Data Analytics Supercomputer), based on cluster architectures, are primarily popular nowadays Gigabit Ethernet and InfiniBand HDR, and high-performance communication architectures designed for high-performance computing clusters – Intel OPA (Omni-Path), PCI Express 5.0, HyperTransport 3.1. [6]

In the image below you can see the comparative characteristics of the rate of transmission of continuous data flow. (Figure 4.)

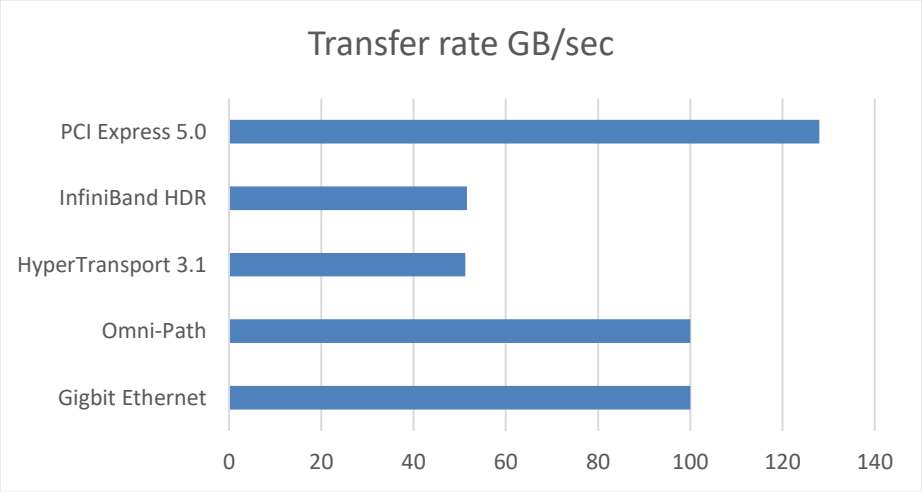

Fig. 4. Continuous data rate

Comparing Ethernet and InfiniBand, it is worth noting that the advantages of the InfiniBand Protocol is a high bandwidth speed and, most importantly, low latency. The standard itself and the equipment allow you to transfer the packet 10 times faster than Ethernet. For high-performance computers and modern data transmission systems, this is crucial. [7]

The diagram perfectly shows the real speed of hardware implementations of various technologies, but it is important to know that in reality, all kinds of hardware platforms such characteristics as latency and data transfer rate are 15-30% worse than the maximum possible. Sometimes this threshold can reach two times the value of deterioration.

Summing up – designing high-speed clusters with a focus on performance indicators, it is always important to consider the performance losses that are associated with the processing and transmission of data in each of the computing nodes of the cluster. Table 1 shows the data of the comparative work of the indicators of bandwidth, delay, the cost of a switch for 8 ports and supporting platforms with the comment reflecting the most significant fact of using the above interfaces.

Table 1. Comparison of high-speed communication interfaces

Technology	Speed, MB/s	Latency s/pack	Review
Gigabit Ethernet	100	33	Convenience of modernization, Guidelines for setting up
InfiniBand HDR	51.6	2	Laid reserve fault tolerance architecture plug-in-play
Omni-Path	100	0.8	Low delay, high speed
HyperTransport 3.1	51.2	1.1	High throughput in both directions
PCI Express 5.0	128	N/A	Architecture for ultra-complex computing

It should be noted an interesting feature of communication interfaces that provide low latency. On the basis of such interfaces, it is possible to build systems with nested NUMA architecture (Non-Uniform Memory Architecture — "Architecture with uneven memory"), the peculiarity of which is the time of access to memory can be determined by its location in relation to the processor. [8]

Also on the basis of such communication interfaces at the software level, it is possible to build systems that have the ability to simulate SMP (Symmetric Multiprocessing) – the architecture of multiprocessor computers, where two or more identical processors are connected in the same order to the shared memory and perform similar functions. The main advantage of the described system will be that you can use standard and familiar operating systems, and software that are focused on SMP-solutions. But due to the high delay of multiprocessor interaction, it will be almost impossible to predict the performance of such a system. [9-10]

4 Tools of parallel high performance clusters

There are different approaches to programming options for parallel computing systems. These can be standard and widely used programming languages that use communication libraries and interfaces to organize inter-processor interaction, such as PVM (Parallel Virtual Machine), MPI (Message Passing Interface), HPVM (High Performance Virtual Machines), MPL (Message Passing Library), OpenMP and ShMem. You can use specialized languages of parallel programming and parallel extensions in such well-known languages as Fortran and C/C++, ADA, Modula-3. [11] We should not forget about the means of automatic and semi-automatic parallelization of sequential programs - BERT 77, FORGE, KAP, PIPS, VAST. It is worth noting the standard programming languages, which can be programmed using parallel procedures from highly specialized libraries. Typically, this method is used to solve problems in certain areas, for example, in the development and definition of genetic algorithms, molecular chemistry, algebra and higher mathematics. [12-14]

Turning to the tools to simplify the design of parallel systems, we can mention CODE and TRAPPER. CODE is a graphical system that allows you to create parallel programs using PVM and MPI libraries. The essence of the work is that the parallel program is displayed as a graph, and its vertices are directly successive parts of the program. TRAPPER is a product of the company Genias, distributed on a commercial basis, which is essentially a graphical programming environment containing components for the design of parallel software. [15]

Based on the user experience of high-speed clusters, the maximum efficiency was observed in programs that took into account the need for interprocessor interaction. At the same time, MPI and PVM libraries are now the most common, even though it is most convenient to write code on packages that use the shared memory interface or automatic parallelization tools.

MPI (The Message Passing Interface) is a standard used to design parallel programs using the messaging model. There are many MPI implementations for C/C++ and Fortran, both free and commercial. They are used for all possible cluster platforms of various kinds, even for High Performance cluster systems built on UNIX, Linux, and Windows hosts. MPI is endowed with the standardization, and in this connection it is endowed with certain rights by the organization of the MPI Forum, which formed the basic tenets of the described standard. The latest version of the standard announced many new useful mechanisms and procedures used to organize the functioning of parallel programs. Such mechanisms as dynamic process control, one-way communication (Put/Get), parallel I/O, which are already actively used in the development and design of cluster solutions, are described. [16-17]

In order to estimate MPI functionality, the calculation time of the problem for solving systems of linear equations depending on the number of processors involved in the cluster was estimated. The cluster was designed for Intel processors and the system of inter-node connections SCI. Of course, the results should not be taken into account as a General model for predicting the performance of the required system. In this case, a particular problem was solved.

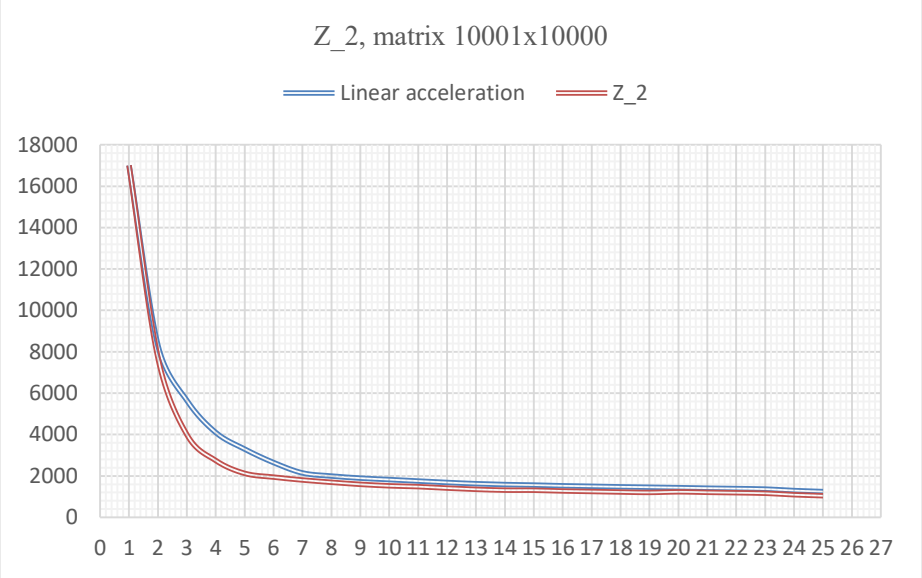

Fig. 5. The dependence of the computation time of the problem of solving systems of linear equations depending on the number of processors involved in the cluster

In figure 5, you can observe two curves. The blue curve is responsible for linear acceleration and the red curve is responsible for the experimental data. It turns out that, using each additional node again, the acceleration will be higher than the linear one. These results are a logical effect of efficient memory cache usage.

5 Summary and conclusions

Before you begin to design a cluster architecture, it is important to identify the goals and objectives to be pursued in your deployment. Depending on the task, you need to choose between high availability (high Performance, HP) clusters, high performance (High Availability, HA) clusters, and mixed systems. High-performance clusters are typically used for tasks that require large amounts of processing power. When designing high-performance clusters, it is important to choose the appropriate implementation tool for your goals, and to increase the cluster computing resources, the most convenient and appropriate parallelization tool.

As a result of the comparative characteristics of high-speed interfaces, it is obvious that the InfiniBand network is the most versatile for complex computing tasks. And in the process of parallelization it is necessary to take into account the need for interprocessor interaction to observe the maximum efficiency of the programs.

References

1. Aliyev T. I. Basics of system design.: St. Petersburg: ITMO University, 2015. – Pp. 120.
2. Bogatyrev V. A., Bogatyrev S. V.: Redundant data transmission through aggregated channels in the real-time network – proceedings of higher educational institutions. Instrumentation - 2016. - Vol. 59. - № 9. - Pp. 735-740.
3. Bogatyrev V. A., Bogatyrev S. V.: Criteria of optimality of multistable fault-tolerant computer systems – journal Scientific and technical of St. Petersburg state University of information technologies, mechanics and optics - 2009. - № 5(63). - Pp. 92-98.
4. Bogatyrev V. A., Bogatyrev A.V., Golubev I. Yu., Bogatyrev S. V.: Optimization of Request distribution between clusters of fault-tolerant computing system – journal Scientific and technical of information technologies, mechanics and optics - 2013. - № 3(85). - Pp. 77-82
5. Bogatyrev V. A., Bogatyrev A.V.: Functional reliability of real-time systems – journal Scientific and technical of information technologies, mechanics and optics - 2013. - № 4(86). - Pp. 150-151
6. Wikipedia. A List of data transmission interfaces bandwidth, https://en.wikipedia.org/wiki/List_of_interface_bit_rates, last accessed 2019/04/19.
7. Wikipedia. InfiniBand, https://ru.wikipedia.org/wiki/InfiniBand, last accessed 2019/04/16.
8. Intel blog. Intel Omni-Path. Data is precious everywhere, https://habr.com/ru/company/intel/blog/370441/, last accessed 2019/04/23.
9. NOU INTUIT. Architectures and topologies of multiprocessor computing systems, https://www.intuit.ru/studies/courses/45/45/info, last accessed 2019/04/15.
10. The website of the company InfiniBand Trade Association, https://www.infinibandta.org/about-the-ibta/, last accessed 2019/04/14.
11. Aleksankov S. M.: Models of dynamic migration with iterative approach and network migration of virtual machines – journal Scientific and technical of information technologies, mechanics and optics. 2015. Vol. 15. No. 6.
12. Qiao Xiang ; J. Jensen Zhang ; X. Tony Wang ; Y. Jace Liu ; Chin Guok ; Franck Le ; John MacAuley.: Fine-Grained, Multi-Domain Network Resource Abstraction as a Fundamental Primitive to Enable High-Performance, Collaborative Data Sciences // SC18: International Conference for High Performance Computing, Networking, Storage and Analysis. Dallas. 2018, pp.58-70.
13. Seyyed Mansur Hosseini, Mostafa Ghobaei Arani. Fault-Tolerance Techniques in Cloud Storage: A Survey // International Journal of Database Theory and Application Vol.8, No.4 (2015), pp.183-190.
14. Chintureena Thingom // International Journal of Interdisciplinary and Multidisciplinary Studies (IJIMS), 2014, Vol 1, No.4, 82- 86.
15. Wubin Li and Ali Kanso. Comparing Containers versus Virtual Machines for Achieving High Availability // Cloud Engineering (IC2E), 2015 IEEE International Conference on. 9-13 March. USA. 2015.
16. Morgan S. Stuart. Mitigating Interference During Virtual Machine Live Migration through Storage Offloading. // Theses and Dissertations. Virginia Commonwealth University. 2016. 61 p.
17. Xiaoyu Fu ; Rui Ren ; Sally A. McKee ; Jianfeng Zhan. Digging deeper into cluster system logs for failure prediction and root cause diagnosis // Ninghui Sun. 2014 IEEE International Conference on Cluster Computing