Journal of Electrical and Electronic Engineering
2019; 7(5): 101-106
http://www.sciencepublishinggroup.com/j/jeee
doi: 10.11648/j.jeee.20190705.11
ISSN: 2329-1613 (Print); ISSN: 2329-1605 (Online)

Analysis of World Experience in Creating Parallel Computing Systems Designed to Effectively Solve DIS-tasks

Andrey Molyakov

Institute of Information Technologies and Cybersecurity, Russian State University for the Humanities, Moscow, Russia

Email address:
andrei_molyakov@mail.ru

To cite this article:
Andrey Molyakov. Analysis of World Experience in Creating Parallel Computing Systems Designed to Effectively Solve DIS-tasks. *Journal of Electrical and Electronic Engineering.* Special Issue: *Science Innovation*. Vol. 7, No. 5, 2019, pp. 101-106.
doi: 10.11648/j.jeee.20190705.11

Received: August 22, 2019; **Accepted**: September 23, 2019; **Published**: October 9, 2019

Abstract: Author describes world experience in creating parallel computing systems by example Cray XE6 and network chip Gemini, designed to effectively solve Data intensive tasks (DIS-tasks). Most often, in modern supercomputers (SC), architecture options with shared (shared) memory are used to provide effective solutions to problems of high capacitive complexity, including those that contain mostly irregular work with memory. It is possible to provide support for a programming model with shared (shared) memory in various ways using hardware, as well as using virtualization software. Different options for implementing a shared memory programming model may vary in functionality and timing of memory accesses. The problem of the "*memory wall*" is that if arithmetic-logical operations take several processor cycles, then operations directly with the memory take several hundred cycles. If the memory is formed from the memories of computing nodes connected by a communication network, then the execution time of such a call includes the time of operation with the network to transfer addresses and data. This already increases the memory access time to several thousand cycles. The problem is that such delays in accessing data cause idle functional units of the processor - they cannot perform arithmetic and logical operations on data, because they simply do not exist due to the large delays in performing operations with memory.

Keywords: DIS-tasks, Irregular Work with Memory, Information Security, Supercomputer, Shared Memory

1. Introduction

Most often, in modern supercomputers (SC), architecture options with shared (shared) memory are used to provide effective solutions to problems of high capacitive complexity, including those that contain mostly irregular work with memory. It is possible to provide support for a programming model with shared (shared) memory in various ways using hardware, as well as using virtualization software. Different options for implementing a shared memory programming model may vary in functionality and timing of memory accesses [1-4]. Let me briefly explain the problems of implementing *shared globally addressable memory* (GAS/PGAS) and ensuring its effective operation in an irregular access mode. This will be required to understand and evaluate the architectural solutions of this implementation considered below.

The memory wall problem is that if arithmetic-logical operations take several processor cycles, then operations directly with the memory take several hundred cycles. If the memory is formed from the memories of computing nodes connected by a communication network, then the execution time of such a call includes the time of operation with the network to transfer addresses and data. This already increases the memory access time to several thousand cycles. The problem is that such delays in accessing data cause idle functional units of the processor - they cannot perform arithmetic and logical operations on data, because they simply do not exist due to the large delays in performing operations with memory.

Massive multi-thread architecture is considered to be a promising processor option. It provides a large stream of simultaneously accessed memory accesses, which, when executed in parallel with pipelines, will allow to achieve

efficiency that depends not on the latency of memory operations, but on the speed of their execution. To ensure this, the communication network must provide the ability to transmit to the memory modules a large number of short messages - requests for memory operations, and the memory must ensure their execution. These solutions simultaneously take into account the need for increased resiliency, including security against information attacks. To use them, special computational models and programming tools are required. Partial solutions for the architecture of all three components are possible, which is further demonstrated when considering different systems.

The problems of energy efficiency are mainly associated with the storage and movement of data, their solution is possible with the use of a new architecture, but, for the most part, with the use of new technologies of optoelectronics and 3D-, 4D-assembly.

2. Cray XE6 Supercomputer

The Cray XE6 supercomputer is Cray's latest high-end supercomputer. It can be considered as an improved version of the current XT6 SK. In the new SC, as in the XT6, the 6th generation of AMD Opteron processors is used, but the communication subsystem is significantly improved. The new Gemini system network delivers significantly better performance and scalability. Currently, Cray XE6 SC is the most high-performance in the Cray product line, providing scalability up to 1 million processor cores and a petaflops level of effective productivity. Cray XE6 is not the first SC to provide effective performance at the level of 1 petaflops. But it is worth noting that he is the first in this major league of petaflops SC, built on the basis of publicly available x86 architecture processors (the SC switch, however, is built on the basis of non-commercial VLSI). At the development stage, this SC had the name (code) Baker. In turn, Cray XE6 SC is a step towards the implementation of the Cray Cascade project funded by DARPA [5-8]. Commercial deliveries of Cray XE6 started in 2010. Already, there is a significant package of orders worth more than $200 million. As was the case with previous scalable Cray systems, in 2011, it is expected the announcement of mid-range XE6m systems up to 6 racks in size [9, 10] (which will use the SeaStar system of the previous generation).

Most often, in modern supercomputers, architecture options with shared (shared) memory are used to provide effective solutions to problems of high capacitive complexity, including those containing predominantly irregular work with memory. In the majority of supercomputers with a mass level of parallelism, built on the MPP architecture with distributed memory, there is no hardware support for working with a single address space. Work within the framework of models with shared memory is possible using virtualization software (for example, ScaleMP or 3 Leaf), but the time characteristics may not be at a high enough level for effective implementation of applications. Cray XE6 is the first of the highest performance SC (from 100 TFlops to several Pflops), built on the MPP architecture, with hardware-supported features of global memory addressing.

In Cray XE6, the capabilities and efficiency of using in different modes of the main memory physically distributed over the control unit are determined by the capabilities and characteristics of the Gemini system network. The distinctive properties of the Gemini system network, distinguishing it from other system networks, include:

a) Support for global addressing (access) to memory;
b) High speed transmission of small messages;
c) Hardware support for programming languages that use the PGAS model when executing put/get/amo primitives.

The structure of the custom VLSI Gemini provides special functional units whose task is to support fast memory access (FMA) modes, transfer large data blocks, cache atomic operations, and others. The FMA mechanism bypasses the OS and is characterized by low data exchange overhead. This allows you to use it to transfer data of small volume. The pipelining of write and read accesses to global memory is supported. This can speed up calculations for irregular memory accesses. Atomic operations with memory are also supported, with the help of which synchronization is performed with one-way accesses to the memory. Support for global memory addressing in Cray XE6 allows the application to use some memory area in the remote WU as its own, without resorting to the OS. This functionality is often called one-way communications. Its presence makes it possible to use programming languages based on the partitioned global address space (PGAS) memory model, such as CAF — Co-Array Fortran and UPC — Unified Parallel C, or the SHMEM library for developing parallel applications.

It is reasonable to be able to evaluate architectural decisions on the organization of supercomputers experimentally. Evaluation testing is used for this. In particular, to evaluate the effectiveness of the memory subsystem, you can use the APEX-MAP standard test that has already become "de facto". Figuratively speaking, this test allows you to see the *memory wall* - the APEX-MAP test builds a surface that displays the dependence of the average number of clocks per read operation on the spatio-temporal mode of working with memory.

According to the profile of working with memory, this point corresponds to the Random Access test, which estimates the efficiency of working with memory in units of GUPS (Giga-updates per second) when accessing memory at random addresses. We denote this point by the letter G. Tasks of this type or close to them are commonly called DIS-tasks (Data Intensive Systems, tasks with intensive work with data, this is a historically established term). Note that the number of ticks at points L and G differ by two orders of magnitude.

If such surfaces are constructed for a multitude of computing nodes, then the delays in accessing memory at points L and G differ by no less than four orders of magnitude. The numerical characteristics presented on the APEX-MAP test allow us to better understand, for example,

the main goals of the DARPA HPCS program, in which the goal of overcoming the *memory wall problem* was set for the first time in world practice. She expressed that when starting work on DARPA HPCS at the beginning of the last decade, the goal was to increase the efficiency of solving problems with irregular intensive memory access (point G) by 3-4 orders of magnitude, while increasing the efficiency of tasks with good localization (point L) factor of. In other words, the goal was to create supercomputers for which the APEX surface would be horizontal and correspond to the level of good spatial-temporal localization (point L). The supercomputers discussed later in this section have actually "grown" out of the DARPA HPCS program or similar Chinese and Japanese programs. Among domestic specialists, an opinion has often been and still is that there are practically no problems approaching the memory profile to a test of the Random Access type (point G). Indeed, it is difficult to imagine such a profile in a practical task, it is more convenient and calmer to not think about it, but this exists and is an urgent problem.

Researching results have shown that in the area close to point G there are also many real computational problems, not only problems solved in modern information systems, and this is considered to be a serious problem in modern works on exaflops topics. In the normalized region of the spatiotemporal localization, the number of measures actually occurring in the performed numerical operation is shown for the applications that are significant for exaflops topics. Essentially, this is the coefficient of deceleration of the task's calculation in comparison with its calculation at peak performance. Most tasks have poor temporal localization and there are problems that approach the worst spatial-temporal localization. The list of tasks is given below.

AVUS (Air Vehicle Unstructured Solver) - CFD program (gas dynamics) used in the development of airframes. WRF (Weather Research and Forecasting) is a weather prediction program. AMR (Adaptive Mesh Refinement) is a representative benchmark for adaptive improvement of the computational grid, suitable for many application areas. Hycom is an ocean modeling program. Overflow is a CFD program used by NASA. RFCTH - a program from the field of physics of explosions (poorly considered in almost all modern machines).

For comparison, three evaluation test programs were also taken:

HPL - Linpack Test STREAM - McCulpin Data Transfer Test. Small Random Access - A test of irregular data access for small amounts of memory (working with memory is similar to fast Fourier transform)

Thus, the problem of effectively providing solutions to problems with irregular access to memory is inherent not only in the field of special applications, but also in traditional computational areas of applications in general. True, for intelligence services and the military this problem is most acute, therefore, DARPA, in the first place, was engaged in this direction.

Memory of such a volume will be characterized by huge delays in performing operations with it, so the problem of ensuring the tolerance of supercomputers and application efficiency will be very difficult. Meanwhile, even taking into account the special orientation of the architecture of supercomputers, a rather complicated issue requiring study and gaining experience is the problem of programming tasks with irregular access to memory. The fact is that ideal tolerance (independence from delays) in practice is not achieved completely now, and it is unlikely that this will be in the future. Therefore, in applications, it is required to take into account the heterogeneity in access time to different parts of the memory and much more.

So, in order to solve the *memory wall problem*, which is key in providing the ability to solve problems of high capacitive complexity with predominantly irregular work with memory, a range of architectural solutions is possible. They should concern three components - a processor, a communication network, and memory. Communication network involves the use of specialized high-performance switches (Gemini project).

3. Gemini System Network

In Cray XE6 the Gemini network is system-forming, performing the functions of combining all the most important supercomputer resources. It is used not only for communication between the control units, but also for connecting the control units with input/output nodes and data storages. The characteristics of the communication subsystem largely determine the capabilities and characteristics of the SC as a whole. Given the widespread use of standard solutions for building other subsystems of supercomputers (computing, data storage, etc.), without a large error, it can be argued that a mass-parallel computer is as effective as its communication subsystem [11].

The Gemini system network can be seen as an intermediate stage between the SeaStar family of system switches (SeaStar in Cray XT3, SeaStar2 in XT4 and SeaStar2 + in XT5) and the Cascades hybrid aircraft system switch (funded by DARPA). The Cascades project system switch received the code name Aries (Aries) and it uses PCI-Express links instead of Hyper-Transport links in SeaStar and Gemini switches to connect to the control unit [12].

This decision is based on various processors - Intel, AMD, graphics and FPGA [13]. The Gemini system network consists of communication nodes combined according to the 3D-tor topology.

Physically consists of one custom VLSI, which is manufactured at TSMC using 90nm technology. The new VLSI Gemini serves two WUs and corresponds in this indicator to two SeaStar modules of the previous generation. Compared to SeaStar2, the power consumption remained at about the same level (even slightly increased), but the functionality and performance characteristics were significantly improved [14, 15].

When designing the Gemini system network, the goals

were to provide the following characteristics and improvements regarding the SeaStar (SeaStar2) network:

a) Scalability to 1 million processor cores;
b) 100-fold increase in the number of MPI messages transmitted per unit time (up to millions per second);
c) Latency for MPI message at the level of 1 μs (3-fold improvement);
d) Providing support for each VLSI Gemini of two WUs, scalability up to a level of over 100,000 network nodes, peak router throughput of 168GB/s;
e) Ensuring that the network interface controller supports MPI, one-way MPI, Shmem, and PGAS languages (UPC, Co-array FORTRAN);
f) Optimization of the transmission of long messages using a DMA block (like SeaStar);
g) Support for adaptive routing.

The distinctive properties of the Gemini system network, distinguishing it from other system networks, include:

a) Support for global addressing (access) to memory;
b) High speed transmission of small messages;
c) Hardware support for programming languages that use the PGAS model when executing put/get/amo primitives (amo is an atomic operation).

Another group of properties of the Gemini system network is related to reliability and ease of use. These include:

a) Recovery with a faulty link using adaptive routing;
b) "Warm" VM replacement (without stopping the system).

The Gemini VLSI structure provides special functional units (FUs) whose task is to support Fast Memory Access modes (FMA FUs), transfer large data blocks (BTE FUs - Block transfer engine), and caching atomic operations (FU AMO) and others. During initialization, descriptors and corresponding windows are created in the global address space. Writing to a window pane causes the execution of put/get/amo primitives. By writing to the descriptors, the base addresses of the windows can be changed. It is also software-controlled reflection of the address to the memory of the slave.

The FMA mechanism bypasses the OS and is characterized by low data exchange overhead. This allows you to use it to transfer data of small volume. It supports pipelining write and read accesses to global memory. This should speed up calculations with irregular memory accesses. Atomic operations with memory are supported, with the help of which synchronization is performed with one-sided accesses to memory.

An important functional feature of Cray XE6 is the support of the global address space at the hardware level using the capabilities of the Gemini network. Such support is not peculiar to mass-parallel computers with distributed memory and communication subsystems. They can use a software-supported virtual global address space (for example, using the appropriate ScaleMP or 3 Leaf products), but the temporal characteristics may not be at a high enough level for efficient application implementation. Programmable support for global memory addressing in Cray XE6 allows the application to use a certain memory area in the remote WU as its own, without resorting to the OS. Such functionality allows one-way communications. This ensures the use of programming languages based on the PGAS memory model when developing parallel applications.

Not all organizations and even countries can afford to develop high-performance GAS/PGAS implementations through the use of special hardware solutions of the level mentioned above in subclause 2 of the conclusions. At the same time, the attractiveness of GAS/PGAS models, on the one hand, and the inaccessibility for various reasons of effective GAS/PGAS implementations, on the other hand, make us look for implementation options that are accessible to a wide range of developers and users. The essence of these works is to find suitable components on the open market that, with some refinement of the hardware and developed new software, would provide, to one degree or another, three basic conditions for the practically acceptable efficiency of GAS/PGAS implementation:

a) The processor or computing node must support a large number of simultaneously performed operations with memory and the network;
b) The communication network should have the smallest possible diameter and have a high throughput when transmitting short packets that implement memory and network operations;
c) RAM modules must have high parallelism and pipelining of memory accesses.

Modern multi-socket boards with multi-core superscalar microprocessors and direct intercrystal connections through, for example, Hyper-Transport and QPI interfaces, to some extent meet these conditions, which was experimentally confirmed.

The communication network that unites such nodes is optimized by the introduction of special fragments (subnets) with high reactivity when transmitting short messages, although with a small number of nodes (several tens). For this, PCI-express and Hyper-Transport interfaces are used, switch chips available on the market, and special software. Thus, a hierarchy is introduced into the organization of such supercomputers, and the diameter of the network is reduced. To reduce the overhead during transmission over a standard communication network that unites all nodes, various types of aggregation of short messages are introduced by software.

The lack of efficiency inherent in such GAS/PGAS implementation methods initiated, within the framework of this area of work, the optimization of the use of GAS/PGAS models due to various methods of localizing memory accesses and organizing optimal interaction of parallel processes operating on shared memory. It was shown in experiments that such methods can increase the efficiency up to ten times in comparison with the direct using of the GAS/PGAS model. The results of these studies will certainly be useful for supercomputers with special hardware implementations of GAS/PGAS.

4. Conclusion

The amount of RAM available for the application for all architectures is determined mainly by the total amount of RAM available in the computer. Shared-memory architectures allow the use of the entire RAM as a single shared resource. They support programming models, which many consider more convenient and natural for developing parallel applications, and in some cases the only possible ones. For many applications, it is possible to separate the processed data of large volume (for example, matrices in linear algebra algorithms) between the computing nodes (CN) of the SC so that each CN carries out operations only with locally located data (periodically exchanging data using a fixed pattern with other CN). But for a number of applications, each CN must process data arbitrarily located in large-volume structures (for example, when working with large tables or databases). In these cases, it is necessary to use one of the variants of the programming model with shared (shared) memory.

The greatest interest is the promising level associated with the use of massive multi-thread microprocessors. It is noted that this level may become the basis for the creation of future energy-efficient exaflops supercomputers, and the basis for this will probably be the development of a hybrid microprocessor combining the performance with the memory of modern massive multi-thread microprocessors (Threadstorm (Cray), CT-2 (NUDT)) with many asynchronous threads, as well as the processing power of GPUs (NVIDIA Fermi) with many synchronous threads.

The first results in this direction were obtained in China and USA. The GAS implementation line using commercially available components is typical for organizations and even countries that do not have sufficient economic and technological resources. At the same time, multi-socket boards with multi-core microprocessors and highly reactive networks (possibly of their own design) are used to create highly-connected fragments of supercomputers. This GAS implementation option compensates for the lack of special hardware (typical for the first line) using special programming techniques for applied tasks. Note that such programming techniques can be useful in GAS implementations using special supercomputer technologies, since these implementations are also imperfect.

In Russia, they work in all directions, most actively - in the generally accessible, least - in the long-term. In leading foreign countries in the field of GAS implementation - USA and China. Moreover, the average level of activity in solving this problem in Russia is much lower than in these countries.

To solve the memory wall problem, various architectural methods are used, which are used in the supercomputers considered below:

a) Localization of data in fast cache memories to reduce real-time memory access;
b) Localization of calculations for data due to hardware support for remote procedure calls by transmitting active messages (RPC, remote procedure call);
c) The use of parallel and pipelined methods for executing memory accesses, usually two architectural approaches are used: vector and multi-thread (with low multi-threading of 2-4 threads per processor core and large multi-threading with 64-256 threads per core), these methods allow you to work with memory with efficiency, which is determined not by the delay in the execution of the call, but by the pace of the memory access;
d) The application of methods for supporting streaming computing, these methods can eliminate memory access in general, since computational processes exchange data through fast localized resources.

In the ideal case, as already noted, using special architectural methods, they try to ensure that the APEX surfaces are flat, i.e. there was no dependence on the spatio-temporal localization of memory accesses.

References

[1] Cray's Baker pops out of the oven as company «re-learns» how to make great systems, by John West, 06. 28. 2010, http://insidehpc.com/category/business-of-hpc.

[2] Cray launches Gemini super interconnect, Timothy Prickett Morgan, Posted in HPC, 25 May 2010, www.theregister.co.uk/2010/05/25/cray_xe6_baker_gemini/page2.html.

[3] Cray Unveils "Baker" Supercomputer, by Michael Feldman, HPCwire, May 25, 2010, www.hpcwire.com/features/94828804.html.

[4] Shahbazi Karim, Eshghi Mohammad, Mirzaee Reza Faghih. Design and Implementation of ASIP-based cryptography processor for AES, IDEA, and MD5. Engineering Science and Technology, an International Journal, 20, 2017, 1308-1317.

[5] NSF Awards PSC $2.8M toward the Purchase of World's Largest Coherent Shared-Memory System, July 29, 2010.

[6] SGI ASIC is first step to exascale system, Rick Merritt, 9/9/2010, www.eetimes.com/electronics-news/4207500/SGI-ASIC-is-first-step-to-exascale-system.

[7] SGI Colors New Shared Memory Machines Ultraviolet, by Michael Feldman, HPCwire, November 16, 2009, www.hpcwire.com/features/SGI-Colors-New-Shared-Memory-Machines-Ultraviolet-70198797.html.

[8] Molyakov, A. S. New Multilevel Architecture of Secured Supercomputers/A. S. Molyakov//Current Trends in Computer Sciences & Applications 1 (3) – 2019. – PP. 57-59. – ISSN: 2643-6744 – https://lupinepublishers.com/computer-science-journal/special-issue/CTCSA.MS.ID.000112.pdf. – DOI: 10.32474/CTCSA.2019.01.000112.

[9] Molyakov, A. S. Technological Methods Analysis in the Field of Exaflops Supercomputers Development Approaching/A. S. Molyakov, L. K. Eisymont//Global Journal of Computer Science and Technology: Information & Technology. – 2017. – № 1 (17). – PP. 37-44.

[10] Molyakov, A. S. A Prototype Computer with Non-von Neumann Architecture Based on Strategic Domestic J7 Microprocessor/A. S. Molyakov//Automatic Control and Computer Sciences. – 2016. – № 50 (8). – PP. 682-686.

[11] Molyakov, A. S. Token Scanning as a New Scientific Approach in the Creation of Protected Systems: A New Generation OS MICROTEK/A. S. Molyakov//Automatic Control and Computer Sciences. – 2016. – № 50 (8). – PP. 687-692.

[12] Molyakov, A. S. Model of hidden IT security threats in the cloud computing environment/A. S. Molyakov, V. S. Zaborovsky, A. A. Lukashin//Automatic Control and Computer Sciences. – 2015. – № 49 (8). – PP. 741-744.

[13] Alam S. R., Barrett R. F., McCurdy C. B., Roth P. C., Vetter J. S. Characterizing Applications on the Cray MTA-2 Multithreading Architecture. ORNL, Cray User Conference, 2006, 13 pp.

[14] Taylor M. B. Bitcoin and the Age of Bespoke Silicon. Proc Int'l Conf. Compilers, Architectures and Systems for Embedded Systems. 2013, 9 pp.

[15] Trader T. STARnet Alliance Seeks Revolution in Chip Design. HPCWire, January 23, 2013.