DOI: 10.14529/jsfi190201

Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community

Vladimir V. Voevodin[1], *Alexander S. Antonov*[1], *Dmitry A. Nikitenko*[1], *Pavel A. Shvets*[1], *Sergey I. Sobolev*[1], *Igor Yu. Sidorov*[1], *Konstantin S. Stefanov*[1], *Vadim V. Voevodin*[1], *Sergey A. Zhumatiy*[1]

© The Authors 2019. This paper is published with open access at SuperFrI.org

The huge number of hardware and software components, together with a large number of parameters affecting the performance of each parallel application, makes ensuring the efficiency of a large scale supercomputer extremely difficult. In this situation, all basic parameters of the supercomputer should be constantly monitored, as well as many decisions about its functioning should be made by special software automatically. In this paper we describe the tight connection between complexity of modern large high performance computing systems and special techniques and tools required to ensure their efficiency in practice. The main subsystems of the developed complex (Octoshell, DiMMoN, Octotron, JobDigest, and an expert software system to bring fine analytics on parallel applications and the entire supercomputer to users and sysadmins) are actively operated on the large supercomputer systems at Lomonosov Moscow State University. A brief description of the architecture of Lomonosov-2 supercomputer is presented, and questions showing both a wide variety of emerging complex issues and the need for an integrated approach to solving the problem of effectively supporting large supercomputer systems are discussed.

Keywords: supercomputer, peak performance, sustained performance, efficiency, parallel computing, supercomputer center, software tools, scalability, monitoring, system level data, data analytics.

Introduction

From the very beginning of the appearance of the first computers, there were always large computing systems at Lomonosov Moscow State University. The first domestic mass-production computer Strela [1] was installed at the Computing Center of Moscow State University in 1956. Basic parameters of the machine were: 500 μs cycle time, performance of 2 thousand operations per second, 300 square meters of footprint area, power consumption of 150 kW. After Strela, there were several dozens of systems in the computer fleet of the Computing Center with various architectures, including self-developed machines based on the ternary number system.

In 1999, the first computing cluster was deployed, consisting of 12 dual-processor compute nodes connected by a fast SCI network, with a peak performance of 12 GFlops. This cluster marked the beginning of a new stage in the development of the computing resources of Moscow University, based on the active use of parallel computing technologies. These are new technologies that are more difficult to use than the conventional sequential approach, but parallel computing is a serious modern trend with enormous potential which is used in modern computers and will be used in all future computing systems.

In 2009, the first petaflops range supercomputer Lomonosov [2] produced by the T-Platforms company was installed at MSU. The supercomputer was built in several stages, and its final configuration has the following parameters: 12346 multi-core Intel processors, 2130 NVIDIA Tesla X2070/2090 graphics processors, 92 TB of RAM, QDR Infiniband as the primary interconnect, parallel data storage system, power consumption of 2.6 MW. The peak performance of the supercomputer is 1.7 PFlops, performance on the Linpack test — 901 TFlops. This is an

[1]Lomonosov Moscow State University, Moscow, Russian Federation

exceptionally large system, which requires about 1000 square meters (including the engineering infrastructure), serving hundreds of users from different organizations that solve tasks from various areas.

The huge number of hardware and software components, together with a large number of parameters affecting the performance of each application, makes ensuring the efficiency of the Lomonosov supercomputer extremely difficult. In this situation, all the basic parameters of the supercomputer should be constantly monitored, and many decisions about its functioning should be made by special means automatically. And this is not a unique feature of this particular supercomputer: an increase in the degree of parallelism and grow of complexity are objective trends of all high-performance computing systems [3]. To confirm, it is enough to analyze the list of the Top500 most powerful supercomputers in the world [4]. In practice, this fact cannot be ignored; otherwise, the efficiency of systems of this scale will be negligible. This was the reason, together with the advent of the supercomputer Lomonosov, to start research aimed at developing technologies to ensure the quality of large supercomputer systems at the Research Computing Center of Moscow State University.

The appearance of Lomonosov-2 supercomputer at MSU fully confirmed this decision. In this paper we would like to describe the tight connection between complexity of modern large high performance computing systems and special techniques and tools required to ensure their efficiency in practice. Further in the paper, Section 1 will be devoted to a brief description of the architecture of Lomonosov-2 supercomputer. In Section 2, we discuss questions showing both a wide variety of emerging complex issues and the need for an integrated approach to solving the problem of effectively supporting large supercomputer systems. Conclusion summarizes the study.

1. Lomonosov-2 Supercomputer

The first stage of Lomonosov-2 supercomputer was installed at Lomonosov Moscow State University in 2014. This system was also created by the T-Platforms company and had four stages in its development:

1. Year 2014, May: Intel Xeon E5-2680v2 10C 2.8GHz, NVIDIA K40s, 6400 cores, Infiniband FDR, peak performance 423 TFlops.
2. Year 2014, October: Intel Xeon E5-2697v3 14C 2.6GHz, NVIDIA K40s, 37120 cores, Infiniband FDR, peak performance 2.575 PFlops.
3. Year 2016, May: Intel Xeon E5-2697v3 14C 2.6GHz, NVIDIA K40s, 42688 cores, Infiniband FDR, peak performance 2.962 PFlops.
4. Year 2018, April: Intel Xeon E5-2697v3 14C 2.6GHz, NVIDIA K40s, Intel Xeon Gold 6126 12C, 2.6 GHz, NVIDIA P100, 64384 cores, Infiniband FDR, peak performance 4.946 PFlops.
5. Year 2019, June: data storage upgrade by 2.5 PBytes up to 3 Pbytes.

Since its inception in 2014, Lomonosov-2 has been included in the global Top500 ranking with the highest position #22 in November of 2014. Since spring of 2015, Lomonosov-2 steadily has been ranking #1 in the Top50 of the most powerful CIS supercomputers [5], thus confirming its leading position in the Russian supercomputer industry.

1.1. System Overview

Today Lomonosov-2 contains 1679 compute nodes in 7 racks (logically divided into three partitions "Compute", "Pascal" and "Test"), 6 management nodes, 10 service nodes, 14 distributed file system servers and 2 storage system appliances. Each compute node is an A-Class solution by the T-Platforms company. There are 7 T-Platforms A-Class racks, 6 of them are fully equipped with 256 compute nodes, and the 7th rack is partially equipped with 160 compute nodes. Infiniband and Ethernet switch systems are also installed in the A-Class system rack. All equipment in these racks excluding PSUs are liquid-cooled by hot water (up to 45 degrees Celsius inlet temperature) to provide better energy efficiency. Key parameters of the Lomonosov-2 system are presented in Tab. 1.

Table 1. Lomonosov-2 supercomputer features

Feature / Partition	Compute / Test	Pascal
Nodes	1487 / 32	160
X86 cores	20818 / 448	1920
GPUs	1487 / 32	320
Memory per node	64 GB	96 GB
GPU memory	11.56 GB	16.3 GB
GPU model	NVidia Tesla K40s	NVidia Tesla P100
CPU model	Intel Haswell-EP E5-2697v3, 2.6 GHz	Intel Xeon Gold 6126, 2.6 GHz

All compute nodes of the Compute and Test partitions have the same configuration described in Tab. 1. One rack contains up to 256 compute nodes (grouped by four on the one assembly with a single coldplate) organized into 8 pools, 2 assemblies of management node, Ethernet switch and auxiliary network Infiniband switch. Each pool contains up to 32 compute nodes (in the 8 assemblies), four 36 ports FDR Infiniband switch systems for communication network connectivity, up to 2 Ethernet switches and one FDR Infiniband switch system to provide auxiliary network connectivity. Compute nodes are connected to the switches via backplane without extra cables.

Compute nodes of the Pascal partition have the same form-factor but they are equipped slightly differently: each node of the partition has 96 GB memory, one Intel Xeon Gold 6126 processor with 12 physical cores and two NVIDIA P100 GPUs.

Mellanox dual-ports ConnectIB-based network module is installed in each compute node as well as the Gigabit Ethernet controller. There are two independent FDR Infiniband networks: communication network for MPI-like exchanges and auxiliary network for I/O operations for Lustre file system.

The communication network is used for MPI communications. Only compute nodes are connected to this network. The network is implemented using 36 ports FDR Infiniband switches which are installed in the A-Class racks. These switches are connected using the flattened butterfly topology $4 \times 8 \times 8$, which allows to extend up to 4D flattened butterfly $4 \times 8 \times 8 \times 8$. This topology was chosen for the system after different topologies simulation based on the requirement for extending the cluster up to 16K compute nodes. Each switch has 8 internal ports connected to the backplane for compute nodes connections and 28 external FDR Infiniband ports for switch-to-switch connectivity.

Management and Service network is based on the 10G/1G Ethernet protocols and used for compute nodes boot, job scheduling, monitoring and remote control. Additionally, Panasas storage system is accessible via the management network.

1.2. System Software and Programming Systems

The operating system of Lomonosov-2 is Centos-7. The only additions are Mellanox Infiniband drivers, Panasas drivers and Lustre drivers. Lomonosov-2 uses xCAT to control all boot images for all nodes and power control via IPMI.

Several OpenMPI versions are available (1.8.4, 1.10.7 and 2.1.1), as well as other MPI implementations, but only OpenMPI supports the flattened butterfly topology of the Infiniband network. Compiling can be done with GNU GCC/GFortran 4.8.5 or Intel Compiler. Intel MPI is not officially supported due to lack of support for the flattened butterfly topology.

For GPU utilization, CUDA versions 5.5, 6.5 and 8.0 are installed. Jobs control on Lomonosov-2 is secured by SLURM 15.08.1 [6] and the GLURMO custom job scheduler. System statistics are collected by collectd and nmond monitoring systems and then processed by Octotron [7] (anomaly detection). Data about actually compiled and used applications and computational packages are collected by XALT software [8].

A wide variety of preinstalled packages are available for users: abinit, espresso, lammps, namd, nwchem, vasp, cp2k, gromacs, magma, etc. Most packages are compiled with CUDA support, all of them support MPI. Intel MKL is available for users to improve performance of their applications.

Lmod [9] compatible with Environment modules was used to control environments for different versions of software.

User access to the supercomputer via ssh and sftp is possible using key-based authentication only. For users management and troubleshooting, Octoshell [10] system is actively used.

Table 2 sums up the software configuration of Lomonosov-2 supercomputer.

2. Lomonosov-2 Supercomputer and Efficiency Issues

As in any large supercomputer system with hundreds of users, there are a lot of components in Lomonosov-2 that affect the efficiency (in a broad sense) of its work, and the state of the components must be carefully controlled. Here are just some of them:

- hardware components (~25000 units);
- software components (~100 units);
- applications (600);
- partitions (10);
- projects (400);
- licenses (100);
- users (2500);
- organizations (300);
- queues (15);
- statuses (20);
- quotas (30);
- jobs (1000 per day)...

Table 2. Lomonosov-2 basic software components

Component	Software
Access Node OS	CentOS 7.1
Compute Node OS	CentOS 7.1
Home Filesystem	Panasas
Scratch Filesystem	Lustre 2.11
Compilers	Intel Compilers (C,C++,Fortran) 15.0; GCC Compilers (C,C++,Fortran) 4.8.5; CUDA 5.5; CUDA 6.5; CUDA 8.0
MPI	OpenMPI 1.8.4; OpenMPI 1.10.7; OpenMPI 2.1.1
Libraries	Intel MKL 2019.2
Resource Manager	Slurm 15.08.1
Job Scheduler	GLURMO
Cluster Manager	Octoshell 2
Monitoring and Analysis Tools	Collectd, nmond, Tentaviz, Octotron, XALT, DiMMoN
Packages, Libraries, Applications	Abinit, Amber, AmberTools, Athena, Charm++, CP2K, CRYSTAL-17, DL_POLY, Firefly (PC-GAMESS), Flow Vision, FMMLIB3D, Gmsh, Gromacs, Lammps, Magma, Materials Studio, Matlab, Molpro, Namd, netCFD, NWChem, Octave, OpenFOAM, Quantum Espresso, Rosetta, Schrodinger, SPILADY, Turbomole, VASP, WIEN2k, WRF...

Analyzing this list, it is necessary to take into account an important feature: there are not only many different types of components in a supercomputer, but the number of different entities within each type varies from tens of units to tens and hundreds of thousands. We already mentioned earlier a large number of components in supercomputers, and here this property becomes obvious: the numbers of entities for Lomonosov-2 are shown in brackets, and the state of each entity of each component must be controlled to ensure the supercomputer as a whole works effectively.

It may seem that some positions of this list are obvious and their processing is simple, but the guarantee of the effectiveness of a supercomputer requires not only maximum details, but also constant monitoring of changes in their state. Let us consider “licenses” and other issues related to software. For each package, library, and tool we have to keep and track all necessary details to ensure ready-to-use status for each software component:

- title and version;
- license current status;
- contacts on license;
- contacts on technical support;
- license key, license activation code;
- license expiration date;
- support termination date;
- restrictions and limitations of the license;
- license update cost, support update cost;

- path to the package, home directory;
- description of installation and fine tuning procedures, basic parameters in use;
- description of testing and checking procedures after upgrades;
- path to reference guides and users manuals;
- person responsible for installation and upgrades;
- contacts of local experts on the software;
- users, projects and organizations who are eligible to use the software.

If the license is not updated on time, or the necessary budget for software update for the next year is not allocated, or the new version of the package has not been tested by an expert in this field, then the efficiency of users, and, consequently, of the supercomputer center as a whole, decreases.

Constant control of the state of each component should be designed in such a way that at any moment it would be possible to find answers to the whole set of questions concerning the efficiency of the supercomputer. To give a feeling of a huge variety of issues that are important to control a status of a supercomputer, we give only a few examples of questions:

- What is a distribution of CPU hours consumed by different software packages for the last year? (Should we spend money for the package X next year?)
- What is average intensity of Infiniband interconnect usage for different partitions? (How large should Infiniband island be in future configurations of supercomputers?)
- How many nodes/cards/disks/cables fail every month?
- How often has Infiniband re-sent packages for the last week?
- How often does LoadAVG exceed number of cores on computing nodes?
- What is a min/max/average level of cache misses for applications of a particular user?
- What is the distribution of waiting time in queues?
- How does LoadAVG behave in my application during execution?
- Who are 5% of the most inefficient applications/users? (regarding CPU load, or LoadAVG, or cache misses, or...)
- What software packages consume 80% of supercomputer time?
- What software components being used in the supercomputer center run with efficiency less than 10%?
- What projects of the supercomputer center use Gromacs with minimal efficiency?
- What is the Top10 list of projects with the lowest CPU load?
- What is the variation in the efficiency of the supercomputer among all the projects when using the Lammps package?
- What is the Top5 list of projects of the supercomputer center by the total amount of consumed CPU hours which do not use the Infiniband interconnect network for MPI-communications?
- What is the total amount of CPU hours consumed by projects of the supercomputer center with highly intensive usage of the fast Infiniband interconnect network?
- What is the Top10 list of users by the maximum number of jobs with abnormal behavior?

There are indeed many questions that arise, since all the components mentioned above need to be considered in close relationship with each other. To guarantee complete control over the operation of supercomputers and prompt responses to all such questions, a set of tools has been developed in the RCC MSU. The main subsystems of the complex are:

- Octoshell — HPC center management system [10];

- DiMMoN — a system for deep monitoring of supercomputer parameters [11];
- Octotron — a system to ensure reliable and autonomous functioning of supercomputers [7];
- JobDigest — a visual tool to analyze the dynamic characteristics of parallel applications [12];
- an expert software system to bring fine analytics on parallel applications and the entire supercomputer to users and sysadmins [13, 14].

The subsystems of the complex are actively used on Lomonosov-2 supercomputer, providing operational data for users and administrators of the supercomputer center [15].

Conclusion

The main objective of this paper is to show the strong correlation between the high complexity of large scale HPC systems and their proper support. There are thousands of components in modern supercomputers that affect the efficiency of parallel applications, and therefore they all require constant deep monitoring. The increasing complexity of computer architecture and the growth of the degree of parallelism are characteristic features that are typical for all, without exception, modern large supercomputer systems. This fact must necessarily be taken into account in any supercomputer center, otherwise its productivity will be in doubt. A set of advanced software tools aimed at solving this problem was developed at MSU Research Computing Center, and the first experience of its use on Lomonosov-2 supercomputer showed both the correctness of the proposed approach and the need to continue and expand work in this direction in the future.

Acknowledgements

The results were obtained with the financial support of the Russian Foundation for Basic Research (grant No. 18-29-03230). The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University.

This paper is distributed under the terms of the Creative Commons Attribution-Non Commercial 3.0 License which permits non-commercial use, reproduction and distribution of the work without further permission provided the original work is properly cited.

References

1. Strela (in Russian). `http://www.computer-museum.ru/histussr/strela0.htm`, accessed: 2019-06-20

2. Sadovnichy, V., Tikhonravov, A., Voevodin, Vl., Opanasenko, V.: "Lomonosov": Supercomputing at Moscow State University. In: Contemporary High Performance Computing: From Petascale toward Exascale (Chapman & Hall/CRC Computational Science), pp. 283–307. Boca Raton, USA, CRC Press (2013)

3. Dongarra, J., Beckman, P. et al.: The International Exascale Software Roadmap. International Journal of High Performance Computer Applications 25(1), 3–60 (2011), DOI: 10.1177/1094342010391989

4. TOP500 Supercomputer Sites. `https://www.top500.org/`, accessed: 2019-06-20

5. Top50 supercomputers of Russia (in Russian). `http://top50.supercomputers.ru/`, accessed: 2019-06-20

6. Slurm workload manager. `http://slurm.schedmd.com/slurm.html`, accessed: 2019-06-20

7. Antonov, A., Nikitenko, D., Shvets, P., Sobolev, S., Stefanov, K., Voevodin, Vad., Voevodin, Vl., Zhumatiy, S.: An approach for ensuring reliable functioning of a supercomputer based on a formal model. In: Parallel Processing and Applied Mathematics. 11th International Conference, PPAM 2015, Krakow, Poland, September 6–9, 2015. Revised Selected Papers, Part I Lecture Notes in Computer Science, vol. 9573, pp. 12–22. Springer International Publishing (2016), DOI: 10.1007/978-3-319-32149-3_2

8. Agrawal, K., Fahey, M.R., McLay, R., James, D.: User environment tracking and problem detection with XALT. In: Proceedings of the First International Workshop on HPC User Support Tools, 21–21 Nov. 2014, New Orleans, LA, USA. pp. 32–40. IEEE Press (2014), DOI: 10.1109/HUST.2014.6

9. McLay, R.: Lmod: Environmental Modules System. `http://www.tacc.utexas.edu/tacc-projects/lmod`, accessed: 2019-06-20

10. Nikitenko, D., Voevodin, Vl., Zhumatiy, S.: Resolving frontier problems of mastering large-scale supercomputer complexes. In: Proceedings of the ACM International Conference on Computing Frontiers (CF'16), May 16–19, 2016, Como, Italy. pp. 349–352. ACM New York, NY, USA (2016), DOI: 10.1145/2903150.2903481

11. Stefanov, K., Voevodin, Vad., Zhumatiy, S., Voevodin, Vl.: Dynamically Reconfigurable Distributed Modular Monitoring System for Supercomputers (DiMMon). In: 4th International Young Scientist Conference on Computational Science. Procedia Computer Science, vol. 66, pp. 625–634. Elsevier B.V Netherlands (2015), DOI: 10.1016/j.procs.2015.11.071

12. Nikitenko, D., Antonov, A., Shvets, P., Sobolev, S., Stefanov, K., Voevodin, Vad., Voevodin, Vl., Zhumatiy, S.: Jobdigest — detailed system monitoring-based supercomputer application behavior analysis. In: Third Russian Supercomputing Days, RuSCDays 2017, Moscow, Russia, September 25–26, 2017, Revised Selected Papers. Communications in Computer and Information Science (CCIS), vol. 793, pp. 516–529. Springer Cham (2017), DOI: 10.1007/978-3-319-71255-0_42

13. Nikitenko, D., Shvets, P., Voevodin, Vad., Zhumatiy, S.: Role-dependent resource utilization analysis for large HPC centers. In: Parallel Computational Technologies. Communications in Computer and Information Science (CCIS), April 2–6, 2018, Rostov-on-Don, Russia. vol. 910, pp. 47–61. Springer (2018), DOI: 10.1007/978-3-319-99673-8_4

14. Shaykhislamov, D., Voevodin, Vad.: An approach for detecting abnormal parallel applications based on time series analysis methods. In: Parallel Processing and Applied Mathematics. Lecture Notes in Computer Science, September 10–13, 2017, Lublin, Poland. vol. 10777, pp. 359–369. Springer International Publishing (2018), DOI: 10.1007/978-3-319-78024-5_32

15. Nikitenko, D., Voevodin, Vad., Zhumatiy, S.: Deep analysis of job state statistics on Lomonosov-2 supercomputer. Supercomputing Frontiers and Innovations, 5(2), 4–10 (2018), DOI: 10.14529/jsfi180201