Proceedings of the 27th International Symposium Nuclear Electronics and Computing (NEC'2019)
Budva, Becici, Montenegro, September 30 – October 4, 2019

MAKING SUPERCOMPUTERS SMART: THE MOSCOW STATE UNIVERSITY EXPERIENCE[1]

S.I. Sobolev, Vl.V. Voevodin, A.S. Antonov, D.A. Nikitenko, P.A. Shvets, K.S. Stefanov, Vad.V. Voevodin, S.A. Zhumatiy

Research Computing Center, Moscow State University, Leninskie gory 1 bld. 4, Moscow, 119234, Russia

E-mail: sergeys@parallel.ru

Just after the computer era has been started, the Research Computing Center of the Moscow State University was equipped with the most modern computing hardware. These days RCC MSU still operates large scale supercomputers including Lomonosov and Lomonosov-2. Supercomputers are open for research and education society supporting hundreds of projects. The huge numbers of hardware and software components and parameters together with the complexity of architectures implemented raise an extremely important question – how efficiently the supercomputers are used. The efficiency study requires deep monitoring and analysis of all processes inside supercomputers. To improve the efficiency, a set of tools and techniques should be created to make a quick and automated decisions in all sides of supercomputer functioning. The paper is a brief overview of RCC MSU experience in supercomputers productivity improving by use of smart software and analytical techniques.

Keywords: HPC, supercomputers, deep monitoring, efficiency analysis, efficient computing

Sergey Sobolev, Vladimir Voevodin, Alexander Antonov, Dmitry Nikitenko, Pavel Shvets, Konstantin Stefanov, Vadim Voevodin, Sergey Zhumatiy

[1] The research is carried out using the equipment of the shared research facilities of HPC computing resources at Lomonosov Moscow State University. The results described in Section 3 of this paper were achieved at Lomonosov Moscow State University with the financial support of the Russian Science Foundation (agreement No. 17-71-20114).

1. Prerequisites

From the very beginning of the appearance of the first computers, there were always large computing systems at Moscow State University. The first Russian mass-production computer Strela with 2KOps performance was installed at Research Computing Center of MSU in 1956. In 1961, the M-20 computer was installed; in 1966, 20 KOps BESM-4 arrived, and in 1968 Strela was replaced with 1 MOps BESM-6, Russia's highest-performance computer at the time. From 1955 until the beginning of the 1990s more than 25 high-performance systems of different architectures, including self-developed ones, had been installed and were actively used at the MSU. In 1999, RCC MSU had chosen the cluster architecture as the basis for projected new computer systems. The first self-made 12 GFlops cluster consisted of 18 nodes connected by SCI network. In 2002, a cluster with a standard Fast Ethernet technology for communication reached the peak performance of 82 GFlops. In 2004, 700 GFlops Hewlett-Packard cluster was installed. In 2008 the Chebyshev supercomputer was put into operation with 625 nodes and 60 TFlops peak performance. In 2009, the first petaflops-range supercomputer Lomonosov [1] was installed. The supercomputer was built in several stages, and in its final configuration 1.7 PFlops peak performance was achieved. In 2014, the first rack of Lomonosov-2 arrived at MSU. Similar to the previous supercomputer, Lomonosov-2 also had several development stages. For now, its peak performance has reached 4.95 PFlops. [2, 3] These days Lomonosov and Lomonosov-2 together form the base of MSU HPC center providing its facilities for almost 3 000 users.

Every supercomputing center has a number of unique properties. Modern supercomputer is a very complex and expensive installation consisting not only of pure computing part; e.g. it also has sophisticated software, network, storage, power, climate, and security infrastructure. Complexity on every level leads to difficulties in support and maintenance of supercomputer. Supercomputers mean huge numbers; for instance, talking about Lomonosov only, we have to manage 6 200 compute nodes with 12 346 CPUs and 2 130 GPUs (51 168 CPU cores and 954 240 GPU cores), and 1.75 PBytes of storage; it uses 252 m^2 and consumes 2.7 MW. On the other hand, a supercomputer has a huge potential that should be provided to its users in the most efficient way. To implement this, every hardware and software component must be monitored, measured, and analyzed, since it may have an impact on user application performance and on general system health. Certainly this cannot be done manually due to system scale. To improve the supercomputer productivity we need a smart approach aimed to provide self-control of every point of potential efficiency loss.

2. Brief Tools and Techniques Overview

Speaking more detailed, first, we must control everything what is necessary to control the efficiency of supercomputer permanently and exactly. Second, we need a guarantee of coincidence between our expectations and reality. Third, we must describe everything that needs to be controlled. To secure complete control over the operation of supercomputers, a set of tools and techniques has been developed in the RCC MSU. The base bricks of this set are Octoshell, DiMMoN, Octotron, and JobDigest subsystems.

Octoshell [4] is a HPC center management system. It was designed to help resolving routine problems in mastering and administering of any computing center from a scale of a stand-alone system up to the top-rank HPC centers that include a number of totally different installations. The toolkit implements a flexibly configurable variety of essential tools in a single interface. It also features useful means of automation for typical administering and management multi-step procedures. It allows be installing and using without any significant changes to existing administrating tools and system software. Octoshell is not integrated with target machines system software; it is executed on a remote server. It runs scripts on HPC systems via SSH as a dedicated user with limited access permissions to perform certain actions. Octoshell web service is the entry point for users and administrators. It provides registration of users and projects, helpdesk features, annual user reports, etc. For MSU HPC facilities, Ocstoshell takes care of two supercomputers, 15 partitions, 400 projects, 2 900 users, 300 organizations, 1 000 jobs per day, etc.

DiMMoN [5] is a system for deep monitoring of supercomputer parameters. It was designed as a dynamically reconfigurable distributed modular monitoring system framework. Its design allows combining both monitoring tasks (supercomputer 'health' monitoring and performance monitoring) in one monitoring system. Our approach allows different parts of the monitoring system process only the data needed for the task assigned to these parts. This helps to process a lot of performance data and to get information about dynamic features of heavy parallel tasks. Another feature of the DiMMoN framework is the ability to calculate performance metrics on-the-fly, dynamically creating processing modules for every job or other objects of interest.

Octotron [6] is a system to ensure reliable and autonomous functioning of supercomputers. Octotron is based on a formal model of computing system that describes system components and their interconnections in graph form. The model determines relations between data describing current supercomputer state (monitoring data) under which all components are functioning properly. Relations are given in form of rules, with the input of real monitoring data. If these relations are violated, Octotron registers the presence of abnormal situation and performs one of the predefined actions: notification of system administrators, logging, disabling or restarting faulty hardware or software components, etc.

JobDigest [7] is a visual tool to analyze the dynamic characteristics of parallel applications. It features application behavior analysis for every job run on HPC system providing: the set of dynamic application characteristics – time series of values representing utilization of CPU, memory, network, storage, etc. with diagrams and heat maps; the integral characteristics representing average utilization rates; job tagging and categorization with means of informing system administrators and managers on suspicious or abnormal applications.

The subsystems of the complex are actively used on Lomonosov and Lomonosov-2 supercomputers, providing operational data for users and system administrators. Based on these bricks, an expert software suite (TASC) was designed to bring fine and smart analytics on parallel applications and the entire supercomputer to users and administrators [8, 9].

3. Smart HPC Analytics: Two Main Targets

Two main targets were established for the TASC analytical software suite:

- to help users to find problems with efficiency of their application in order to improve the efficiency if possible;
- to help administration to estimate overall supercomputer efficiency.

If we want to help users in this way, we have to examine every application and every job. We have to perform on-the-fly analysis so users could get the information just after the job was ended; in special cases it should be done even during the job execution. We have to analyze everything connected to job runs, including performance monitoring data for every node used by job, and usage of libraries linked to the application. We have to use smart techniques, e.g. machine learning to find and highlight low efficiency issues. At last, we have to provide it to user in the comfort way.

	Найденные проблемы	ID задачи	Начало счета	Конец счета	Статус	Число узлов	Время счета (часы)	Размер задачи (ЦПУ-часы)	Загрузка ЦПУ	Загрузка ГПУ	Load average	IPC	Получено байт по MPI (МБ/с)	Передано байт по MPI (МБ/с)
		697858	2018-07-25 13:12:09	2018-07-25 13:36:56	cancelled	1	0.4	5.8	86.7	0.0	25.4	0.36	0.0	0.0
		697859	2018-07-25 13:12:09	2018-07-25 13:36:56	cancelled	1	0.4	5.8	83.9	0.0	24.7	0.35	0.0	0.0
		697857	2018-07-25 13:12:06	2018-07-25 13:36:56	cancelled	1	0.4	5.8	55.2	0.0	18.7	0.36	0.0	0.0
		697867	2018-07-25 13:40:02	2018-07-25 14:23:28	completed	1	0.7	10.1	34.7	0.0	10.7	0.46	0.0	0.0

Заметный дисбаланс внутри узлов по использовании памяти.

Figure 1. Efficiency tab in Octoshell shows analysis results for all user applications

Figure 1 presents the job overview for users available within Octoshell. Every row represents a job; small icons on the left show inefficiency issues determined by our analytical software. Crossed arrows mean incorrect partition choice: GPU partition was chosen, but accelerators were not used. Libra sign means imbalanced node memory usage.

	Найденные проблемы	ID задачи	Начало счета	Конец счета	Статус	Число узлов	Время счета, в часах	Размер задачи (ЦПУ-часы)	Загрузка ЦПУ	Загрузка ГПУ	Load average	IPC	Получено байт по MPI, в МБ
≡		852079	2018-10-20 07:37:52	2018-10-21 11:46:24	completed	24	28.1	9455.8	47.1	0.0	14.0	1.68	550.0
≡		852332	2018-10-20 11:11:53	2018-10-21 17:18:01	completed	24	30.1	10114.3	46.9	0.0	14.0	1.71	514.2

Figure 2. Example of bad network locality

Figure 2 shows more complicated example. Two similar runs of the same application with the same parameters resulted in different run times: 28.1 and 30.1 hours. Analysis discovered weak network locality issue marked by the leftmost icon for the 2nd run. Application uses intensive MPI communication. The set of nodes chosen by batch system for the 1st run was connected with 1 Infiniband switch, while for the 2nd run data communications had to pass 3 switches.

For now, there are about 30 issues that can be discovered and highlighted for the further user analysis. JobDigest is available for every completed task.

Going further towards estimation of overall supercomputing efficiency, there are much more data we have to take into account. In addition to performance monitoring and library usage data, we have to analyze a whole job flow, users and projects specific data, hardware and software failures, file system and storage health, etc. It is quite complicated to integrate and to present such variety of data flows. Generally speaking, this is one of the most important research directions for us. Here are several example of what we can do now.

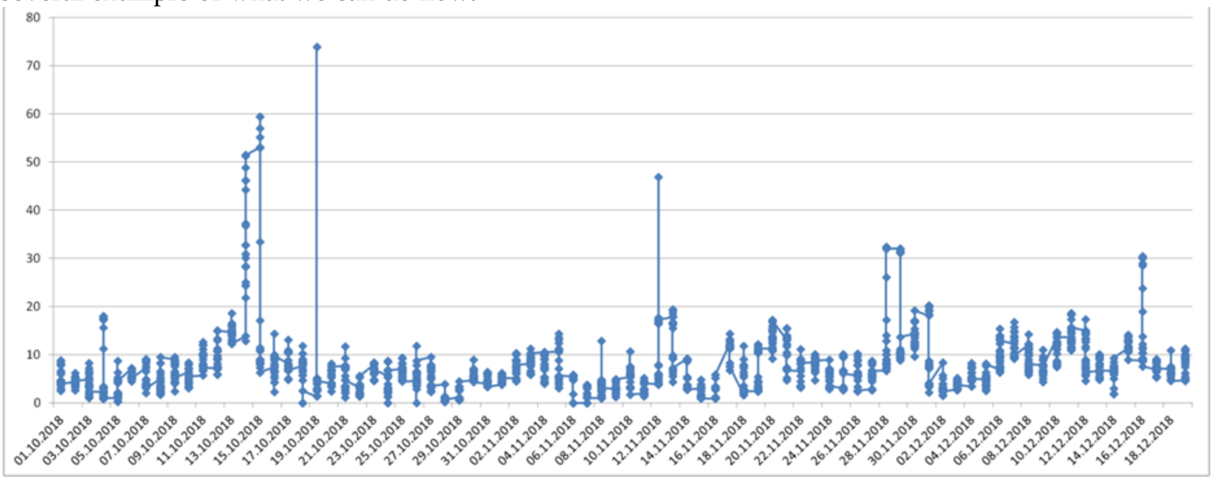

Figure 3. Fraction of compute nodes with low CPU load

Figure 3 show the fraction of compute nodes with low CPU load value (in %) collected for about 2.5 months. There are several points per each day since data are collected every hour. For the most of the days we observe quite a low fraction of weak loaded nodes. High surges on the graph is a matter of investigation – they mean overall weak supercomputer load.

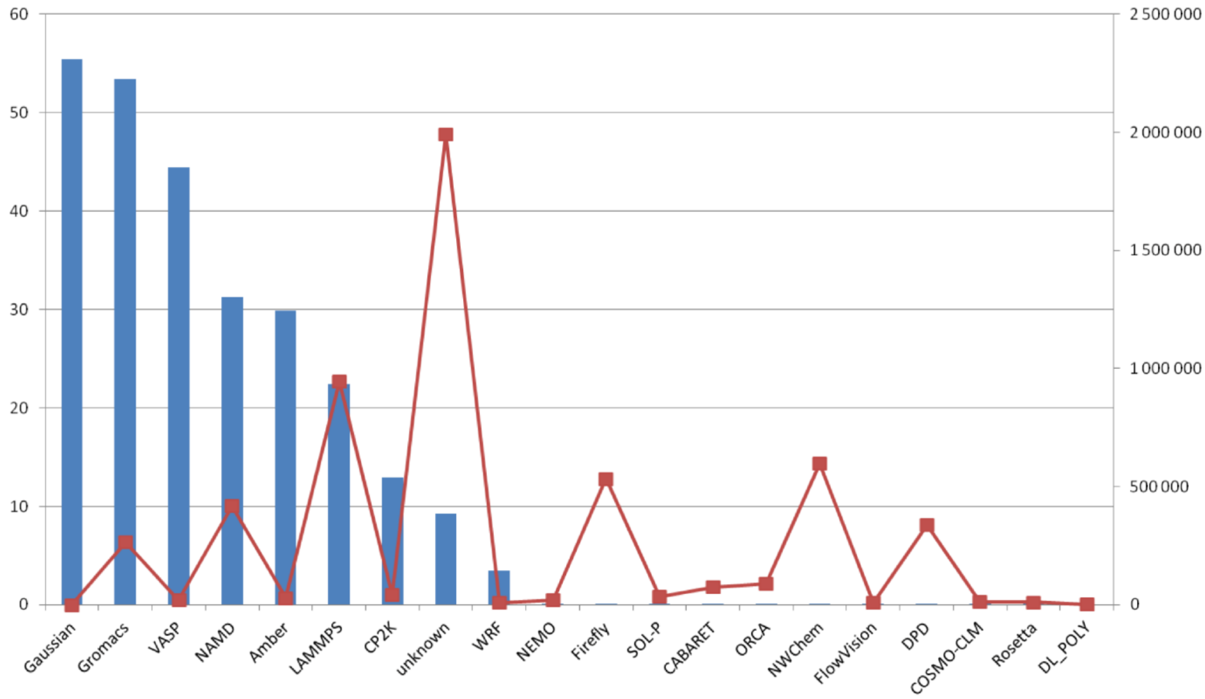

Figure 4. GPU Usage by software packages

Another load issue is shown on the Figure 4. It presents GPU usage statistics by software packages. Blue columns represent average GPU load (scale is on the left), red line represents node-hours (scale is on the right). "Unknown" package unites own code written by users (i.e. not recognized as a specific package by XALT). In general, we can do nothing with GPU-inefficient packages; on the other hand, most packages utilize GPUs more efficiently than home-grown user codes. One lesson learned from this analysis example is the necessity to persuade users to use preinstalled software packages where possible.

4. Unusual Conclusion

Here we presented a brief overview of RCC MSU experience in supercomputers productivity improving by use of software and analytical techniques which a going to or pretending to be a smart enough. Every subsystem mentioned is in extensive development now.

The importance of investigation of supercomputer seems to be obvious; but what about real practice? During the Russian Supercomputing Days 2019 Conference (September 23-24, 2019, Moscow, Russia) there was a special workshop devoted to HPC centers efficiency. Before the workshop, a survey was carried out for users and also for administrators of supercomputers. A number of survey answers aimed to know about administrators' interest in efficiency issues were discouraging. For instance, 77% of people answered do not know about top applications with the lowest CPU load; 66% do not use any tools for regular efficiency reports; 66% do not collect statistics on software packages used. The reasons for that discussed during the workshop are quite simple: lack of time, lack of man power, lack of tools. Definitely this should be changed. RCC MSU is ready to share its experience how to provide the high-quality computing service — and how to estimate the efficiency it is used.

References

[1] V. Sadovnichy, A. Tikhonravov, Vl. Voevodin, and V. Opanasenko. "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.

[2] Voevodin Vladimir, Antonov Alexander, Nikitenko Dmitry, Shvets Pavel, Sobolev Sergey, Stefanov Konstantin, Voevodin Vadim, Zhumatiy Sergey, Brechalov Andrey, and Naumov Alexander. Lomonosov-2: Petascale supercomputing at Lomonosov Moscow State University. In Contemporary High Performance Computing: From Petascale toward Exascale, volume 3, pp. 305-330, Boca Raton, United States, 2019.

[3] Vl. Voevodin, A. Antonov, D. Nikitenko, P. Shvets, S. Sobolev, I. Sidorov, K. Stefanov, Vad. Voevodin, S. Zhumatiy. Supercomputer Lomonosov-2: Large Scale, Deep Monitoring and Fine Analytics for the User Community. In Journal: Supercomputing Frontiers and Innovations, Vol.6, No.2 (2019). pp.4–11. DOI: 10.14529/jsfi190201

[4] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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