Parallel Computing: Technology Trends
I. Foster et al. (Eds.)

doi:10.3233/APC200087

Energy Consumption of MD Calculations on Hybrid and CPU-Only Supercomputers with Air and Immersion Cooling

Ekaterina DLINNOVA [a,b,1], Sergey BIRYUKOV [c] and Vladimir STEGAILOV [a,b]
[a] *National Research University Higher School of Economics, Moscow, Russia*
[b] *Joint Institute for High Temperatures of RAS, Moscow, Russia*
[c] *JSC NICEVT, Russia*

Abstract. The article presents the energy consumption and efficiency analysis based on the data from three small-size supercomputers installed in JIHT RAS. One system is the air-cooled hybrid supercomputer Desmos with AMD FirePro GPUs and two others are the air-cooled and liquid-cooled segments of the supercomputer Fisher based on AMD Epyc Naples CPUs. To collect data, we implement the same real-time analytics infrastructure on all three supercomputers. We consider classical molecular-dynamics problem as a benchmarking tool. Our results quantify the energy savings that are provided by the GPU-based calculations in comparison with CPU-only calculations and by liquid cooling in comparison with air-cooling. During strong scaling benchmarks, we detect an interesting minimum of energy consumption in the CPU-only case.

Keywords. supercomputers, monitoring systems, statistics analysis, energy profiles, energy consumption

1. Introduction

The effective use of supercomputer resources is an extremely important task in the field of high-performance computing [1]. However, currently there are no standard generally accepted methods that allow to collect, to analyze and to evaluate the optimal use of supercomputer resources. Energy efficiency is becoming an increasingly decisive requirement for supercomputers, as race in industrial production involves the use of the next generation of powerful computing systems. Technology has come a long way, but it is clear that there are still some difficult but crucial work that industry needs to do in the area of energy efficiency. Enlarging computing power without increasing energy consumption will undoubtedly require a deep transformation that extends across all aspects of HPC systems design.

Graphics processing units (GPUs) became widely used as accelerators for scientific and HPC applications due to their energy efficiency and high memory bandwidth. And

[1]Corresponding Author: Ekaterina Dlinnova, International Laboratory for Supercomputer Atomistic Modelling and Multi-scale Analysis NRU HSE, 34 Tallinskaya Ulitsa, 123458, Moscow, Russia; E-mail: edlinnova@hse.ru

in this work we are giving real-life values for comparison of GPU-accelerated and CPU-only computations.

Cooling technologe is another avenue for improving energy efficiency of HPC systems. And in this work we compare one systems based on a de facto standard air-cooling with a similar system that uses immersion oil cooling technology.

2. Related works

In [2], the authors raise the problem that large-scale distributed systems consume a huge amount of energy. To solve this problem, it is proposed to use job shutdown policies that can dynamically adapt the amount of resources to the actual workload. The sheer amount of energy consumed by large-scale computing and network systems, such as data centers and supercomputers, is causing serious concern in a society increasingly dependent on information technology. Trying to solve this problem, the research community and industry have proposed many methods to curb the energy consumed by IT systems.

The article [3] discusses methods and solutions aimed at improving the energy efficiency of computing and network resources. It discusses methods for estimating and modeling the energy consumed by these resources, and describes methods that work at the distributed system level, trying to improve aspects such as resource allocation, planning, and managing network traffic. This work is aimed at reviewing the state of technology in the field of energy efficiency in order to further facilitate research on the creation of networks and computing resources more efficient. Several indicators have been suggested that are most commonly used for infrastructure, such as data centers.

In order to assess how optimally the supercomputer complex consumes electricity, it is necessary to enter certain indicators and characteristics. For example, the authors of [4] characterize the energy efficiency of the Cray XC30 supercomputer system in three metrics: time to solution for a given workload; workload power consumption and energy efficiency (PUE) of the data center where the system resides. The decision time and energy consumption are closely related. For example, choosing a processor can reduce your solution time by increasing power consumption.

In [5] and [6], new methods for distributing resources were presented that take into account the topology of the machine, the patterns of interaction of tasks, and the characteristics of the application to select the best node among those available on the platform.

The article [7] discusses the current state of energy-efficient methods of parallel computing in order to achieve optimal resource consumption in conditions of limited energy consumption. The authors of [7] believe that the power consumption of modern high-performance computing systems should be reduced by at least one order of magnitude before they can be increased to ExaFLOP performance. Although new hardware technologies and architectures can be expected to contribute to this goal, software technology such as proactive and energy-efficient planning, resource allocation, and fault-tolerant computing should also bring significant success.

The paper [8] presents the energy consumption model for a hybrid supercomputer (which ranked first in Green500 in June 2013), which combines CPU, GPU, and MIC technologies to achieve high levels of energy efficiency. This model takes into account both the characteristics of the workload, the amount and location of resources that are used by each task at a certain time, and it also calculates the predicted energy consumption at the system level.

Table 1. Specifications of supercomputers considered

Supercomputer	Major components
Desmos	32 nodes Intel Xeon E5-1650v3, 6 cores, 3.5 GHz AMD FirePro S9150 GPU, Angara interconnect (4D torus), Air cooling
Fisher Air segment	18 nodes 2 x AMD Epyc7301, 16 cores, 2.7 GHz (8 DIMMs per socket) InfinibandFDR interconnect (switch) , Air cooling
Fisher Immersion segment	24 nodes 2 x AMD Epyc7301, 16 cores, 2.7 GHz (4 DIMMs per socket) Angara interconnect (switch), Immersion oil cooling

Article [9] provides a detailed analysis of the problems and possibilities of supercomputer computing in various fields of human activity: in machine learning, astronomy, medicine, materials science and energy efficiency. The authors discuss the scalability problems of both technical equipment and software and algorithms. In this regard, the discussion of the problems of efficiency from the point of view of the future of exascale-computing and analysis of large data is extremely relevant and important.

3. Hardware

For this study, we analyze the statistics of three supercomputers, all of which were installed at the Joint Institute for High Temperatures of RAS.

The first supercomputer Desmos consists of 32 nodes with AMD FirePro S9150 graphics accelerators, interconnected with a low-latency high bandwidth Angara interconnect [10]. The supercomputer is aimed at carrying out calculations by the classical molecular dynamics method, and can also effectively accelerate the calculations of the electronic structure of materials.

The second supercomputer is the Fisher supercomputer that consists of air-cooled and oil-cooled segments with AMD Epyc 7301 CPUs (see Table 1). The air-cooled segment consists of 18 dual-socket nodes connected by Infiniband FDR. The oil-cooled segment consists of 24 dual-socket nodes connected by Angara network (its switch-based fat-tree variant). The immersion cooling system was designed by the Immers company.

4. Model used for benchmarks

We analyse the power consumption of the calculation for the same resource-intensive scientific code, namely, the large-scale molecular dynamics problem in the LAMMPS package, running on the three above-mentioned supercomputers. For the benchmarking, we use a typical molecular-dynamics (MD) problem of 4 millions Lennard-Jones atoms.

Figure 1. Real-time Desmos visualization in Grafana. We can see the dependence between energy consumption and time on Desmos while LAMMPS is running on different number of nodes (1,2,4,8,16,32).

5. Monitoring system

To collect, analyze and visualize statistics on the use of supercomputers, we used a set of applications:

- Telegraf is a utility for collecting time series measurements.
- InfluxDB is a clustered database specifically designed for storing time series.
- Grafana is a time series visualization tool. Web application for setting up charts and dashboards.

Telegraf is an agent written in Go for collecting performance metrics from the system it is running on and the services running on that system. Data aggregation infrastructure is based on InfluxDB. Grafana is an open source platform for visualizing, monitoring and analyzing data. Grafana allows users to create dashboards with panels, each of which displays certain indicators for a set period of time. Each dashboard is universal, so it can be customized for a specific project or taking into account any needs. Grafana builds visualization of the cluster state in real-time. For example, we can verify the energy consumption of the Desmos supercomputer (Figure 1).

Another possible option for building a cluster monitoring system is to use a software stack consisting of Elasticsearch, Logstash, and Kibana (the so-called ELK stack). ELK is specially designed to solve the problems of collecting, storing and processing system logs.

However, the ELK stack is designed for highly loaded web-projects, which are based on the products of companies that contain hundreds of servers of the same type. It is advisable to use ELK if you intend to analyze hundreds of megabytes of logs every day, hundreds of production servers on which you want to catch events, and also have your own highly loaded application and it needs to be monitored. In addition, logstash consumes server resources for each rule, since before processing data, it first processes

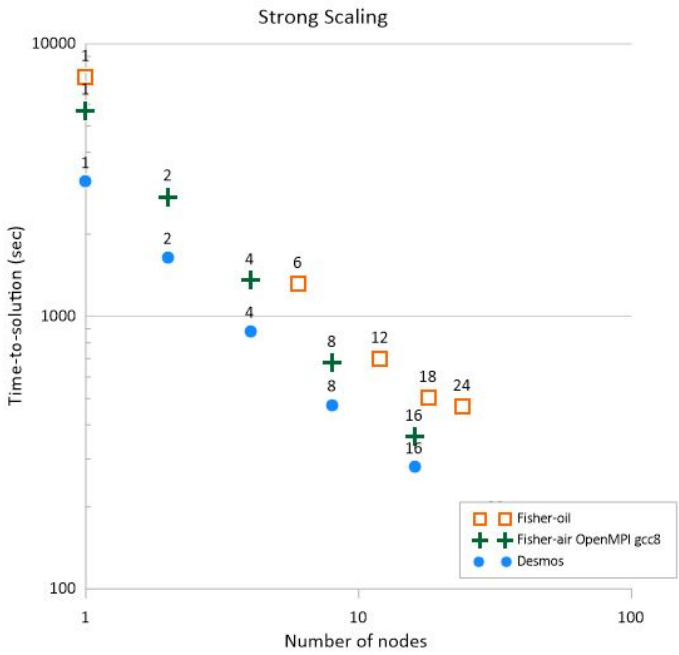

Figure 2. Strong scaling of the LAMMPS test problem: the dependence between the time-to-solution and the number of nodes for three supercomputers considered.

them. For the above reasons, we have decided that using the ELK stack is less suitable for building an HPC monitoring system.

6. Possible levels of energy consumption analysis

To collect data, we implement the following three-tier infrastructure on all three supercomputers:

- Level 1: RAPL-like protocols for CPU/DIMM energy consumption,
- Level 2: IPMI protocol at the node level (with limitations for FirePro GPUs),
- Level 3: SNMP protocol for collecting data from UPS smart-cards.

Recent Intel processors support the Running Average Power Level (RAPL) interface, which among other things provides estimated energy measurements for the CPUs, integrated GPU, and DRAM. AMD Epyc CPUs have compatible interface [2]. These measurements are easily accessible by the user, and can be gathered by a variety of tools, including the Linux event interface. This allows an easy access to energy information when designing and optimizing an energy-aware code.

[2]This interface, however, demonstrates some problems, see https://community.amd.com/thread/243717

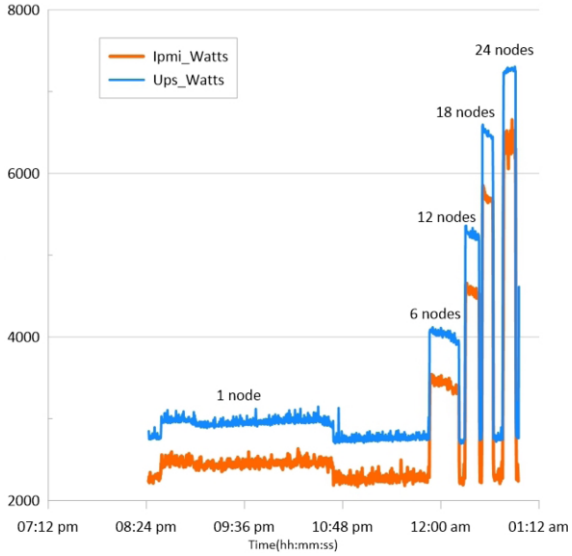

Figure 3. The Fisher immersion segment energy profile. We can see the time profile of the consumed power when LAMMPS is running on a different number of node. The blue line shows the data collected from the power supply. The orange line shows the data collected using IPMI.

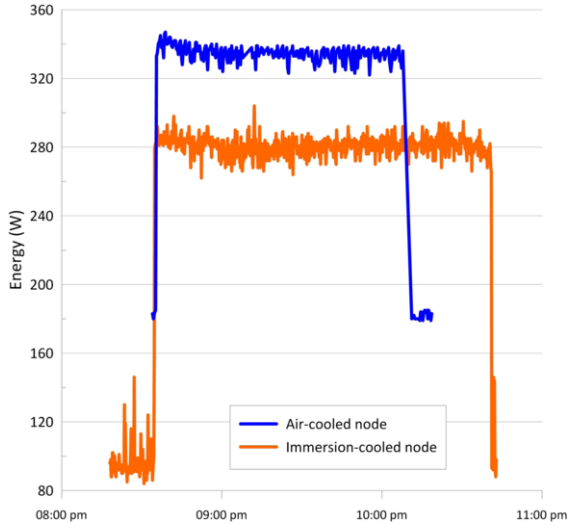

Figure 4. Comparison of the power consumption profiles (collected via IPMI) for the single node runs on the air-cooled and immersion segments of the Fisher supercomputer.

While greatly useful, on most systems these RAPL measurements are estimated values, generated on the fly by an on-chip energy model. The values are not documented well, and the results (especially the DRAM results) have limited validation.

Through the Intelligent Platform Management Interface (IPMI), it is possible to connect remotely to the server and manage its operation:

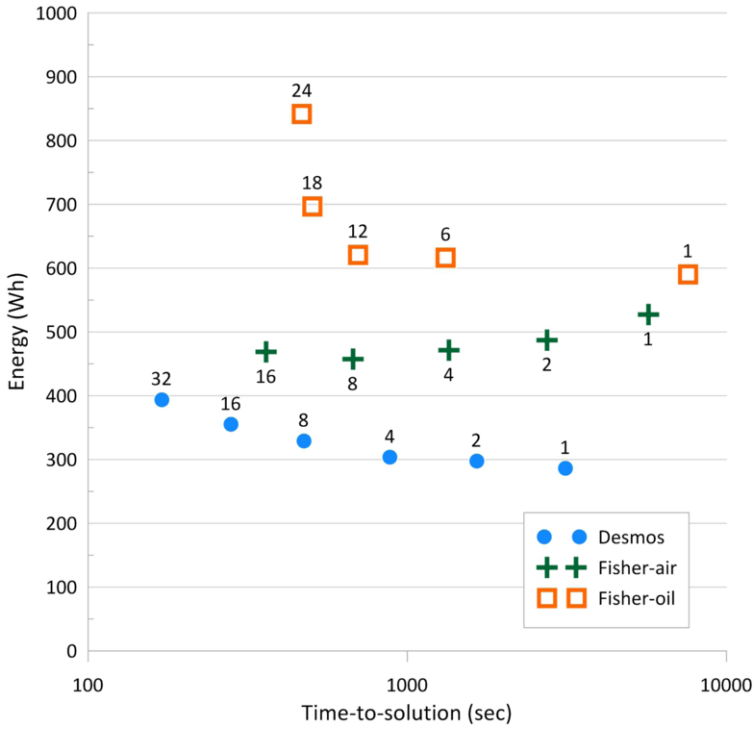

Figure 5. The dependence between the energy-to-solution and the time-to-solution for the test LAMMPS runs on three supercomputers considered.

- Monitor the physical condition of the equipment, for example, check the temperature of the individual components of the system, voltage levels, fan speed, energy consumption.
- Restore the server in automatic or manual mode (remote system reboot, power on / off, loading ISO images and updating software).
- Manage peripheral devices.
- Keep an event log.
- Store information about the equipment used.

Using the Simple Network Management Protocol (SNMP) allows to monitor almost any server, workstation or network equipment. SNMP monitoring is a standard way to obtain the characteristics of network resource utilization by routers and other network equipment. Many other parameters, such as disk space or CPU utilization, can also be obtained from the target device via SNMP. In this paper we present the results gathered at the second level only (via IPMI).

7. Benchmarking procedure and results

In this section the benchmarking procedure is described. First of all, we freeze the tasks queue on three supercomputer considered, waite for all already started jobs finish and start our measurements. We execute sequentially LAMMPS on different number of nodes

on Desmos, Fisher Air and Fisher Immers. The energy consumption information is collected using special Telegraf-exec plugin, after parsing it was inserted into InfluxDB database. We analyze the real-time energy consumption visualization using Grafana and its dashboards.

Figure 2 depicts the strong scaling results for three supercomputers. We can see in the log-log scale the dependence between the time-to-solution and the number of nodes.

Figure 3 depicts the results of the benchmarks for the Fisher supercomputer segment with immersion oil cooling. We see instantaneous values of energy consumption (W) when we execute the LAMMPS code with different number of nodes.

Figure 4 depicts the profiles of power consumption for the single node runs on the air-cooled and immersion-cooled segments of Fisher supercomputer.

To calculate total energy consumption during the running time period in kWh, we used the InfluxDB function *integral*:

```
SELECT
FROM "ipmitool_raw"
WHERE time >= '2019-09-29T20:19:00Z'
AND time <= '2019-09-29T20:25:00Z'
AND host='10.2.1.101'
```

A subtle question is how to choose the start and finish time points. In this work we choose them manually looking at the power profile in Grafana. Changing these times even by a few seconds can result in significant changes of the integral value. So this selection of time periods is very important, and in the future works we plan to use task manager synchronisation to determine start and finis time points.

The benchmark results of three supercomputers are presented on Figure 5. We can see the dependence between the total consumed energy-to-solution and the time-to-solution. We see that hybrid computations are more energy efficient despite we compare the novel CPUs (Zen microarchitecture uses 14 nm FinFET) and slightly old Haswell CPUs (22 nm FinFET) and FirePro GPUs (28 nm CMOS). Immersion cooling does not demonstrate evident benefits (despite the fact that at this stage we have not taken into account the energy consumption of the liquid transfer subsystem and the heat exchanger). The immersion segment demonstrates longer values of time-to-solution due to the reduced memory bandwidth of its nodes (see Table 1), their lower power consumption does not compensate this fact (see Figure 4) and the energy integrals is larger than for the air-cooled segment.

The results for the air-cooled segment show an interesting feature: the energy consumption decreases when we increase the number of nodes and there is a minimum energy consumption at 8 nodes. The origin of this effect is presumably connected with the dynamic fan speed control in the nodes during the benchmarks and deserves a separate study in the future.

8. Conclusions

We have implemented identical energy monitoring systems for real-time analytics of power and energy consumption on three supercomputers: the hybrid air-cooled Desmos

supercomputer and the air-cooled and the liquid-cooled CPU-only segments of the Fisher supercomputer.

Benchmarking results based on a single MD model calculation example show the following:

- The excess consumption of an air-cooled system compared to an immersion-cooled system is on average 30% or 1.4 kW.
- The energy efficiency gain of a hybrid air-cooled system is 200 Wh (46%) for 1 node and decreases with an increase in the number of nodes used for the test calculation.
- We detected a minimum of total energy consumption for the test problem on CPU-only systems.

Acknowledgment

The study has been partially supported by the grant of the President of Russian Federation for support of leading scientific schools grant NSh-5922.2018.8 and supported within the framework of the Basic Research Program at the National Research University Higher School of Economics (HSE) and within the framework of a subsidy by the Russian Academic Excellence Project 5-100.

References

[1] F. Mantovani and E. Calore, "Performance and power analysis of HPC workloads on heterogeneous multi-node clusters," *Journal of Low Power Electronics and Applications*, vol. 8, no. 2, 2018.

[2] A. Benoit, L. Lefevre, A.-C. Orgerie, and I. Raïs, "Reducing the energy consumption of large-scale computing systems through combined shutdown policies with multiple constraints," *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 176–188, 2018.

[3] E. Jaureguialzo, "Pue: The green grid metric for evaluating the energy efficiency in dc (data center). measurement method using the power demand," in *2011 IEEE 33rd International Telecommunications Energy Conference (INTELEC)*, pp. 1–8, IEEE, 2011.

[4] G. Pautsch, D. Roweth, and S. Schroeder, "The Cray® XC supercomputer series: Energy-efficient computing," tech. rep., Technical Report, 2013.

[5] Y. Georgiou, E. Jeannot, G. Mercier, and A. Villiermet, "Topology-aware job mapping," *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 14–27, 2018.

[6] C. Gómez-Martín, M. A. Vega-Rodríguez, and J.-L. González-Sánchez, "Performance and energy aware scheduling simulator for HPC: evaluating different resource selection methods," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 17, pp. 5436–5459, 2015.

[7] A.-C. Orgerie, M. D. d. Assuncao, and L. Lefevre, "A survey on techniques for improving the energy efficiency of large-scale distributed systems," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 47, 2014.

[8] A. Sîrbu and O. Babaoglu, "A data-driven approach to modeling power consumption for a hybrid supercomputer," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 9, p. e4410, 2018.

[9] C. Jin, B. R. de Supinski, D. Abramson, H. Poxon, L. DeRose, M. N. Dinh, M. Endrei, and E. R. Jessup, "A survey on software methods to improve the energy efficiency of parallel computing," *The International Journal of High Performance Computing Applications*, vol. 31, no. 6, pp. 517–549, 2017.

[10] V. Stegailov, E. Dlinnova, T. Ismagilov, M. Khalilov, N. Kondratyuk, D. Makagon, A. Semenov, A. Simonov, G. Smirnov, and A. Timofeev, "Angara interconnect makes GPU-based Desmos supercomputer an efficient tool for molecular dynamics calculations," *The International Journal of High Performance Computing Applications*, vol. 33, no. 3, pp. 507–521, 2019.