Hybrid Supercomputer Desmos with Torus Angara Interconnect: Efficiency Analysis and Optimization

Nikolay Kondratyuk[1,2], Grigory Smirnov[1], Ekaterina Dlinnova[3], Sergey Biryukov[4], and Vladimir Stegailov[1](✉)

[1] Joint Institute for High Temperatures of the RAS, Moscow, Russia
stegailov.vv@mipt.ru

[2] Moscow Institute of Physics and Technology, Dolgoprudny, Russia

[3] National Research University Higher School of Economics, Moscow, Russia

[4] JSC NICEVT, Moscow, Russia

Abstract. The paper describes the first experience of practical deployment of the hybrid supercomputer Desmos at the Joint Institute for High Temperatures of the Russian Academy of Sciences (JIHT RAS). We consider job scheduling statistics, energy efficiency, case studies of GPU acceleration efficiency and benchmarks of the distributed storage with a parallel file system.

Keywords: Job accounting and statistics · Energy efficiency GPU acceleration · Parallel I/O

1 Introduction

Desmos is a supercomputer targeted to molecular dynamics (MD) calculations that was installed in the JIHT RAS in December 2016. Desmos is the first application of the Angara interconnect for a GPU-based MPP system [1,2].

Modern MPP systems can combine up to 10^5 nodes for solving one computational problem. For this purpose, MPI is the most widely used programming model. The architecture of the individual nodes can differ significantly and is usually selected (co-designed) for the main type of MPP system deployment. The most important component of MPP systems is the interconnect. The interconnect properties have a major influence on the scalability of any MPI-based parallel algorithm. In this work, we describe the Desmos supercomputer, which is based on cheap 1CPU+1GPU nodes connected by the original Angara interconnect.

The JIHT team was supported by the Russian Science Foundation (grant No. 14-50-00124). The Desmos supercomputer is a part of the Supercomputer Centre of JIHT RAS. The authors acknowledge the Shared Resource Center "Far Eastern Computing Resource" IACP FEB RAS (http://cc.dvo.ru) for granting access to the IRUS17 supercomputer.

L. Sokolinsky and M. Zymbler (Eds.): PCT 2018, CCIS 910, pp. 77–91, 2018.
https://doi.org/10.1007/978-3-319-99673-8_6

The Angara interconnect is a Russian-designed communication network with a torus topology. The interconnect ASIC was developed by JSC NICEVT and manufactured by TSMC using the 65 nm process. The Angara architecture uses some principles of both the IBM Blue Gene L/P and the Cray Seastar2/Seastar2+ torus interconnects. The torus interconnect developed by EXTOLL is a similar project [3]. The Angara chip supports deadlock-free adaptive routing based on bubble flow control [4], direction ordered routing [5,6] and initial and final hops for fault tolerance [5].

The results of the benchmarks confirmed the high efficiency of commodity GPU hardware for MD simulations [2]. The scaling tests for electronic structure calculations also showed the high efficiency of MPI-exchanges over the Angara network.

In this paper, we combine the results of the Desmos supercomputer performance analysis. These results pave the way to optimizations of the supercomputer efficiency and could be relevant for other HPC systems.

2 Related Work

Job scheduling determines the efficiency of a supercomputer practical deployment and is a very important topic in parallel systems (see, e.g., [7]). The everyday work of supercomputer centers shows a need for separation of cloud-like jobs (which do not require a high-bandwidth low-latency interconnect between nodes) from regular parallel jobs. Such a separation is a way for increasing efficiency of supercomputer deployment [8]. There have been some attempts of statistical analysis of supercomputers operation in Russian HPC centers (see, e.g., [9]).

The increase of power consumption and heat generation of computing platforms is a significant problem. Measurement and presentation of the results of performance tests of parallel computer systems become more and more often evidence-based [10], including the measurement of energy consumption, which is crucial for the development of exascale supercomputers [11].

Nowadays, partial use of single precision in MD calculations with consumer-grade GPUs cannot be regarded as a novelty. The results of such projects as Folding@Home confirmed the broad applicability of this approach. Recent developments in optimized MD algorithms include the validation of the single-precision solver (see, e.g., [12]). The authors of [13] give very instructive guidelines for achieving the best performance at minimal cost in 2015.

The success of the TeraChem package [14] illustrates the amazing perspectives of GPU usage for quantum chemistry.

The ongoing increase of data generated by HPC calculations leads to the requirement of a parallel file system for rapid I/O operations. However, benchmarking parallel file system is a complicated (and usually expensive!) task, which is why accurate results of particular case studies are quite rare (see, e.g. [15]).

3 Statistical Data of Desmos Deployment

The batch system for user jobs scheduling of Desmos is based on Slurm, which is an open-source workload manager designed for Linux clusters of any size [16]. It is used in many HPC centers worldwide (the paper [16] has been cited more than 500 times). Slurm has the following main features:

- allocates exclusive and/or non-exclusive access to resources (Compute Nodes) to users for some time so they can perform a work;
- provides a framework for starting, executing, and monitoring work (normally a parallel job) on the set of allocated nodes;
- arbitrates conflicting requests for resources by managing a queue of pending work.

The SlurmDB daemon stores data into a MySQL database. The SlurmDB daemon runs on the management node. In September 2017, the SlurmDB database was activated on Desmos, giving us the possibility of detailed analysis of supercomputer load statistics. The default Slurm tool *sreport* has quite limited functionality. That is why we use SQL-queries for accessing the SlurmDB for statistical analysis. For example, the following command retrieves and calculates the duration of allocated jobs:

```
select timestampdiff(second, from_unixtime(time_start),
from_unixtime(time_end)) as running,
timestampdiff(second, from_unixtime(time_submit),
from_unixtime(time_start)) as waiting,t.*
from desmos_job_table;
```

Figure 1 shows the distribution of jobs over the number of cores used and over running time t_R. GPU floating point performance is not taken into account when drawing the iso-levels of $R_{\text{peak}} * t_R$ constant value. This quantity corresponds to the total number of floating-point operations that CPUs deployed for the particular job are able execute theoretically during time t_R.

Parallel algorithms can be executed either slowly on a modest number of cores (nodes) or quickly if their parallel scalability justifies using a large number of processing elements efficiently. Two iso-levels separate three regions of total number floating-point operations corresponding to individual jobs: less than 10 PFlos, between 10 and 100 PFlops, and above 100 PFlops. The percent values shown in the blue boxes correspond to the share of each region in the Desmos total workload since the beginning of SlurmDB logging. We see that the major part of all the jobs executed on Desmos have been essentially supercomputer-type jobs.

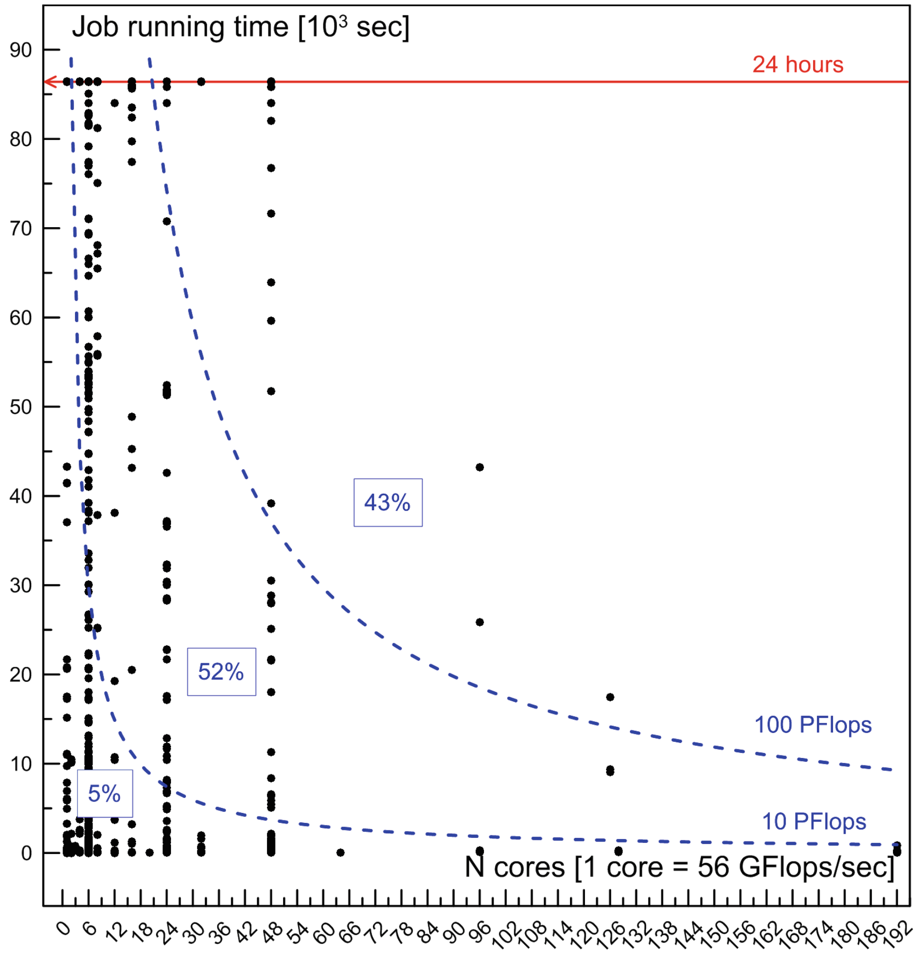

Fig. 1. Job running time vs. job size. Each point corresponds to one job (Color figure online)

At the same time, we see that there are jobs that were executed on six cores or less, i.e. on a single node. This type of jobs can be easily moved away from the supercomputer either to the cloud or to a personal workstation.

The efficiency of the supercomputer job scheduling policy can be evaluated by such type of graphs. The more points we see on the right side of the graph, the more efficient is the end-user collective deployment of the supercomputer. Users should be motivated to use scalable codes and to choose larger number of nodes for speeding calculations up. The following Slurm batch system partitions have been created on the Desmos supercomputer:

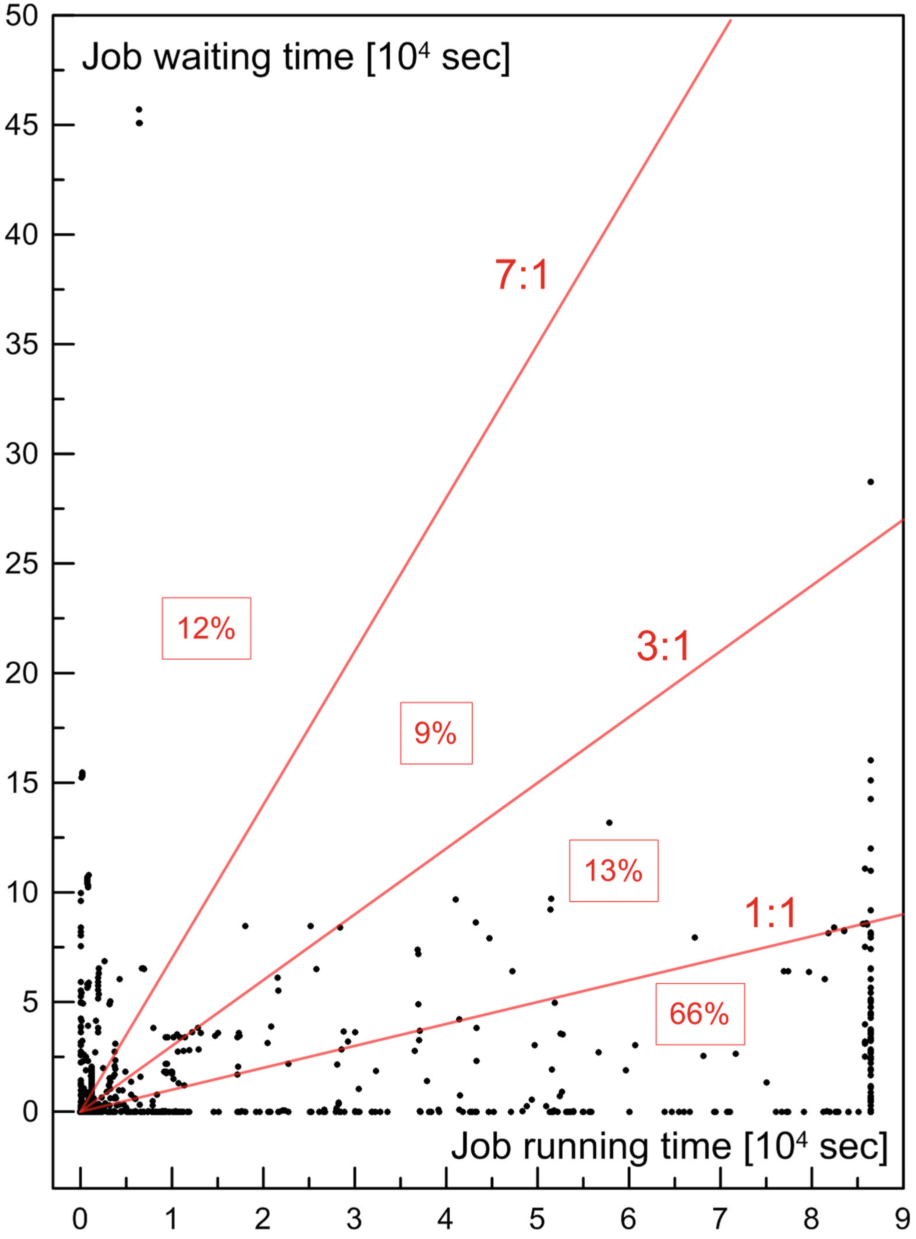

Fig. 2. Job waiting time vs. job running time. Each point corresponds to one job. The percent values shown in the red boxes correspond to the share of each region in the supercomputer total workload (Color figure online)

- test: max time = 15 min, any number of nodes;
- max1n: max time = 1440 min, min/max number of nodes = 1;
- max8n: max time = 1440 min, min/max number of nodes = 4/8;
- max16n: max time = 720 min, min/max number of nodes = 4/16;
- max32n: max time = 360 min, min/max number of nodes = 4/32.

This policy motivates users to deploy higher numbers of nodes. Also, it prevents overloading the supercomputer with small one-node or two-node jobs.

Another aspect of job scheduling is the waiting time for the job to start calculation after being submitted to the job queue. Figure 2 shows the correlation of the job running time t_R to the job waiting time t_W (i.e. the time between the moment of job submission into the batch queue and the moment of job execution). Three levels of ratios t_W/t_R are shown in Fig. 2. Fortunately, the majority of jobs (66%) fall into the category with $t_W < t_R$. Obviously, jobs with $t_W > t_R$ should be regarded as inefficient. Diminishing the number of such jobs is another criterion of supercomputer efficient usage.

4 Energy Consumption Optimization: The VASP Case Study

Our recent studies of energy consumption at frequency variation [17–19] show that a variation of CPU frequency can have a positive effect on reducing energy consumption for memory-bound algorithms. Here we extend this type of analysis from the level of a single CPU to the level of a whole supercomputer.

VASP 5.4.1 was compiled for Desmos using gfortran 4.8, Intel MPI and linked with Intel MKL for BLAS, LAPACK and FFTW calls. The model represents a GaAs crystal consisting of 80 atoms in the supercell. All 32 nodes are used for the benchmark runs. Each run corresponds to one complete iteration for electron density optimisation that consists of 35 steps. We use digital the logging capabilities of the UPS for digital sampling of the power consumed during the benchmark runs.

The results of the CPU frequency variation from the 3.5 GHz to 1.2 GHz are presented in Fig. 3. We see how the level of consumed power decreases when the CPU frequency decreases. At the same time, the time-to-solution increases.

The lower graph shows the variation of the total energy consumed in two cases:

- The real benchmark of Desmos shows that no energy saving regime can stem from CPU frequency variation.
- The hypothetical case with all fans in the chassis switched off shows a shallow minimum of total energy consumed. The total power consumption of the chassis fans working at full speed is about 4 kW. If we subtract the fan-determined power draw from the total power level (e.g., it could be the case if liquid immersion cooling would be used). This minimum corresponds to saving about 3.4% of energy at the cost of about 3.8% longer calculations.

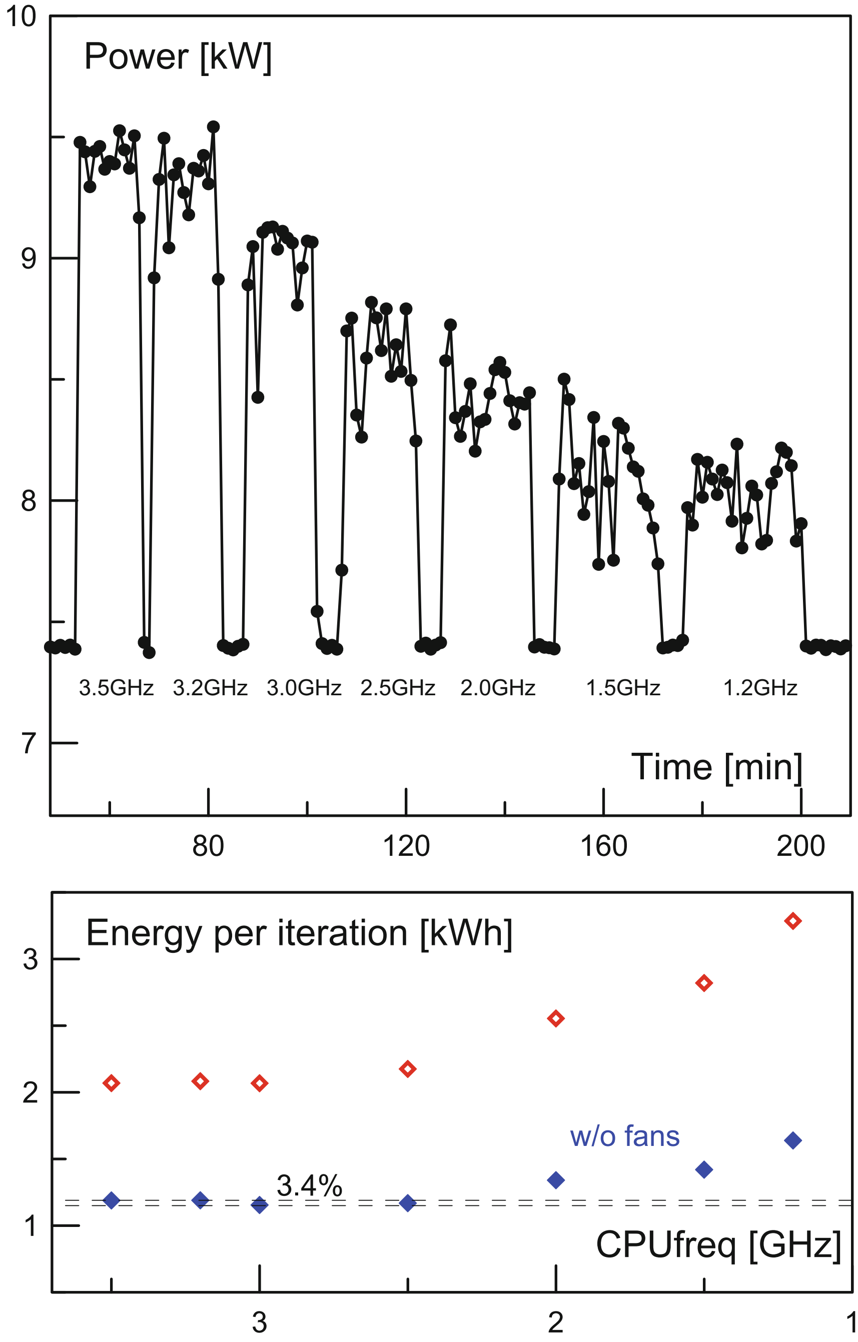

Fig. 3. Power variation and consumed energy variation on the same VASP test benchmark at different CPU frequencies

5 Case Studies of GPU Efficiency

NVIDIA CUDA technology was released in 2007. The past decade became a time of gradual adoption of this programming paradigm. Nowadays, the CUDA-enabled software ecosystem is quite mature. The GPU usage in HPC is motivated not only by energy efficiency but by cost efficiency as well. Consumer cards with teraflops performance in single precision represent an attractive option for cheap computational acceleration. The deployment of such commodity GPU accelerators in the Desmos supercomputer was a carefully planned decision [2]. However, the absence of double-precision capabilities narrows the spectrum of potential problems that can be solved using this hardware.

In this context, we present benchmarks showing the efficiency of the Desmos supercomputer for certain workloads.

5.1 Classical Molecular Dynamics with Gromacs

Classical molecular dynamics is an important modern scientific tool (see, e.g., [20–24]). In Fig. 4, we can see the results of the Intel Xeon-based supercomputer IRUS17 and the Desmos supercomputer on two detailed biomolecular benchmarks [13] from the Gromacs package.

The cost of each node in Fig. 4 consists of the price of the computational resources with the corresponding infrastructure excluding the costs of interconnect. The prices of single nodes are estimated according to the price lists from the ThinkMate.com website at the end of November 2017.

A Desmos node without GPU costs about $ 2600, while a Desmos node with one GTX 1070 costs $ 3100. An IRUS17 node with two Intel Xeon E5-2698 v4 costs $ 11 000 (IRUS17 consists of dual-node blades in an enclosure), and an IRUS17 node with two Intel Xeon E5-2699 v4 costs $ 13 000. The labels in Fig. 4 show the amount of nodes. The cost is multiplied by the corresponding number of nodes.

We see that Desmos is ahead of IRUS17 for these benchmarks, in terms of both maximum attainable speed of calculation (ns/day) and cost-efficiency.

5.2 Quantum Molecular Dynamics with TeraChem and GAMESS-US

Quantum chemistry and electronic structure calculations are among the major consumers of HPC resources worldwide (see, e.g. [25–30]). The TeraChem package is a rare example of CUDA-based software that deploys very efficiently single-precision floating point operations of NVIDIA GPU accelerators. In this work, we compare the performance of TeraChem with the well-know quantum chemistry package GAMESS-US.

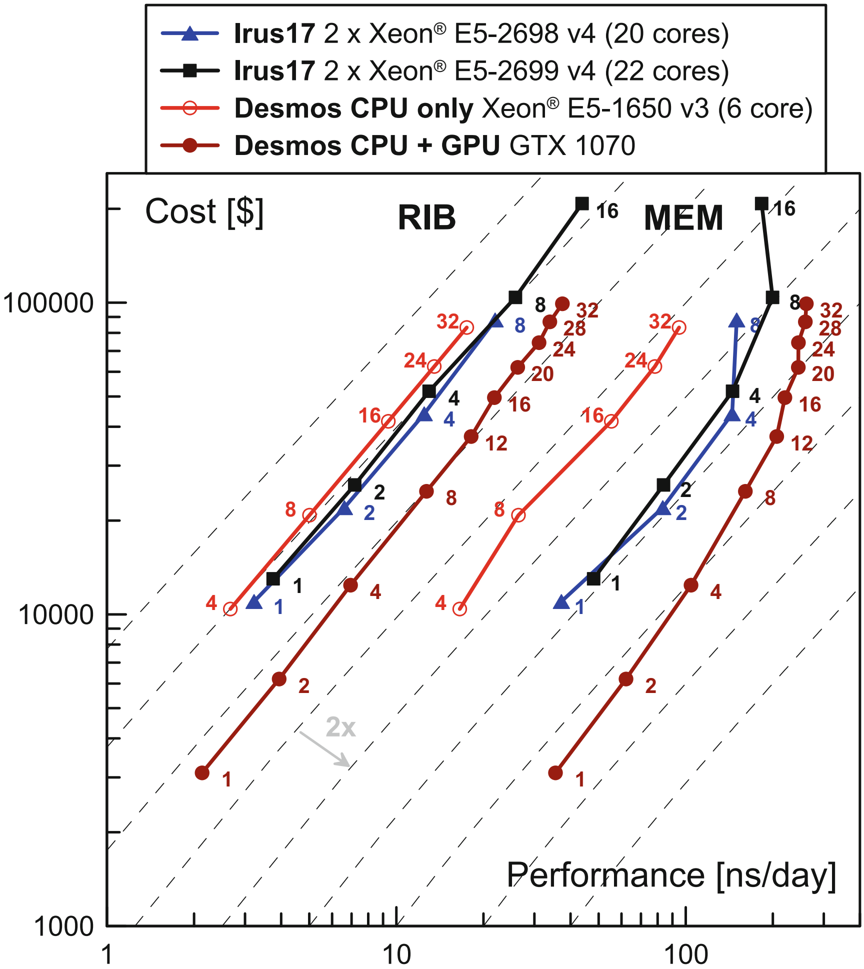

Fig. 4. Comparison of the supercomputers Desmos and IRUS17, on two biomolecular benchmarks (RIB: 2 million atoms, MEM: 82 thousand atoms; see [13])

The test model is the *ab initio* DFT molecular dynamics of the molecule of malondealdehyde $CH_2(CHO)_2$. The 6–31g basis is used together with the B3LYP exchange-correlation functional.

TeraChem is not MPI-parallelized and runs on a single node of Desmos (on a single core with a GTX 1070 accelerator). This hardware gives 0.5 s per one MD step in this test benchmark for TeraChem. The same level of performance we see in the CPU-only MPI-parallelized GAMESS-US calculation on 12 Desmos nodes (0.5 s per one MD step).

It is instructive to compare the peak performance of the hardware under consideration in these two tests. Twelve Desmos nodes have 4 TFlops of double-precision peak performance and 540 GB/s DRAM total memory bandwidth. One GTX 1070 accelerator has 6 TFlops of single-precision peak performance and 256 GB/s DRAM memory bandwidth. These numbers allow us to conclude that, with respect to GAMESS-US, the Desmos supercomputer is equivalent to a 128-TFlop supercomputer (= 32 nodes × 4 TFlops) based on Intel Xeon Broadwell CPUs.

6 Parallel File System Benchmarks

Many scientific HPC codes generate huge amounts of data. For example, in classical MD, the limits of the system size are trillions of atoms [31]. Desmos allows for GPU-accelerated modeling of MD systems with up to 100 million atoms. On-the-fly methods of data processing help considerably but cannot substitute post-processing completely. Another unavoidable requirement is the saving of control (or restart) points during or at the end of the calculation.

All 32 nodes of Desmos have been equipped with fast SSD drives, and the BeeGFS parallel file system has been installed in order to use all these disks as one distributed storage.

For comparison, we consider the Angara-K1 supercomputer, located at JSC NICEVT. This cluster is based on the Angara interconnect as well. Angara-K1 has a dedicated storage server (hardware RAID-adapter Adaptec 5405z, RAID level: 6 Reed-Solomon, HDD: 8 × 2 TB SATA2, FS: Lustre 2.10.1, FS type: ext3).

The schemes of the Desmos and Angara-K1 supercomputers and relevant parameters are given in Fig. 5.

The standard Lennard-Jones benchmark was run with the LAMMPS molecular dynamics package (the benchmark is based on the model "melt" from the LAMMPS distribution package, the model corresponds to a f.c.c. crystal of Lennard-Jones particles and has been replicated to 16 million particles).

LAMMPS has two variants for output of large amounts of data. It is possible to use either standard output methods or MPI-IO capability.

Figure 6 depicts the results of the benchmarks for different sizes of the MD model. We see that the absolute values of the calculation time are higher for Angara-K1 than for Desmos. However, the performance degradation due to storing large files is more pronounced for Desmos. The MPI-IO output gets the evident benefits of the distributed storage of Desmos.

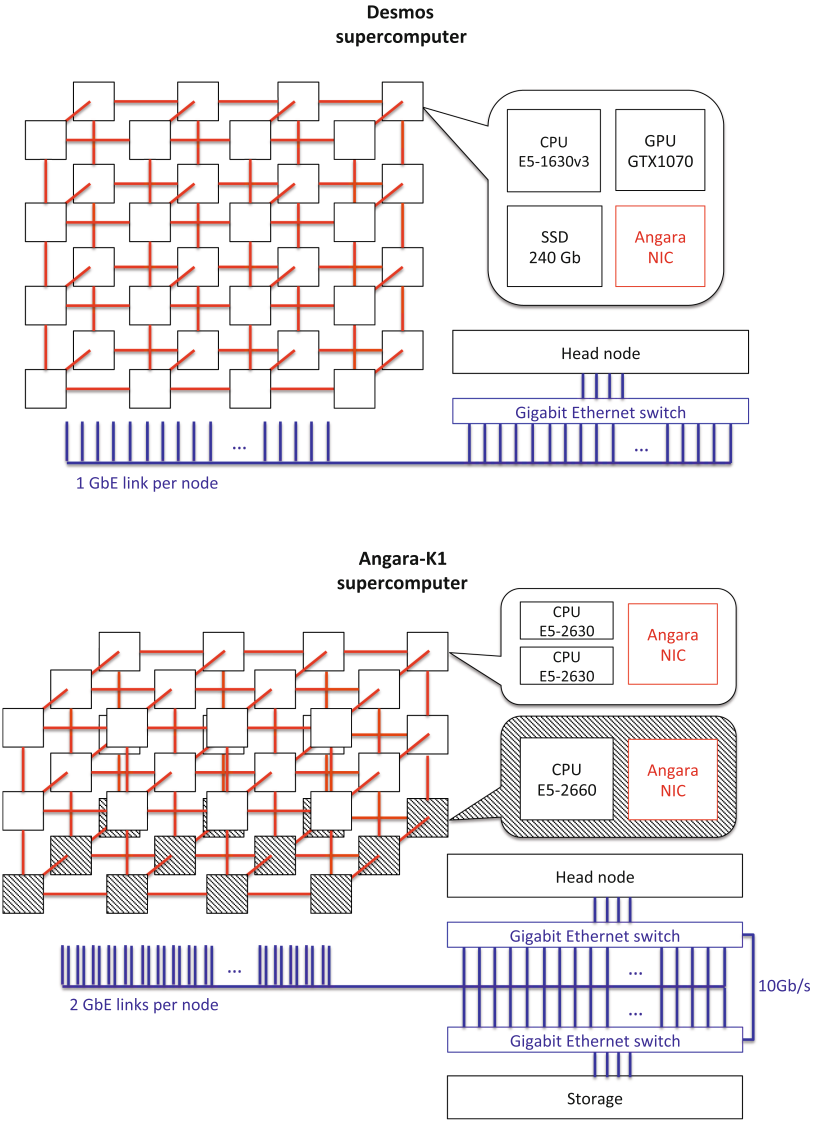

Fig. 5. The schemes of the supercomputers Desmos and Angara-K1

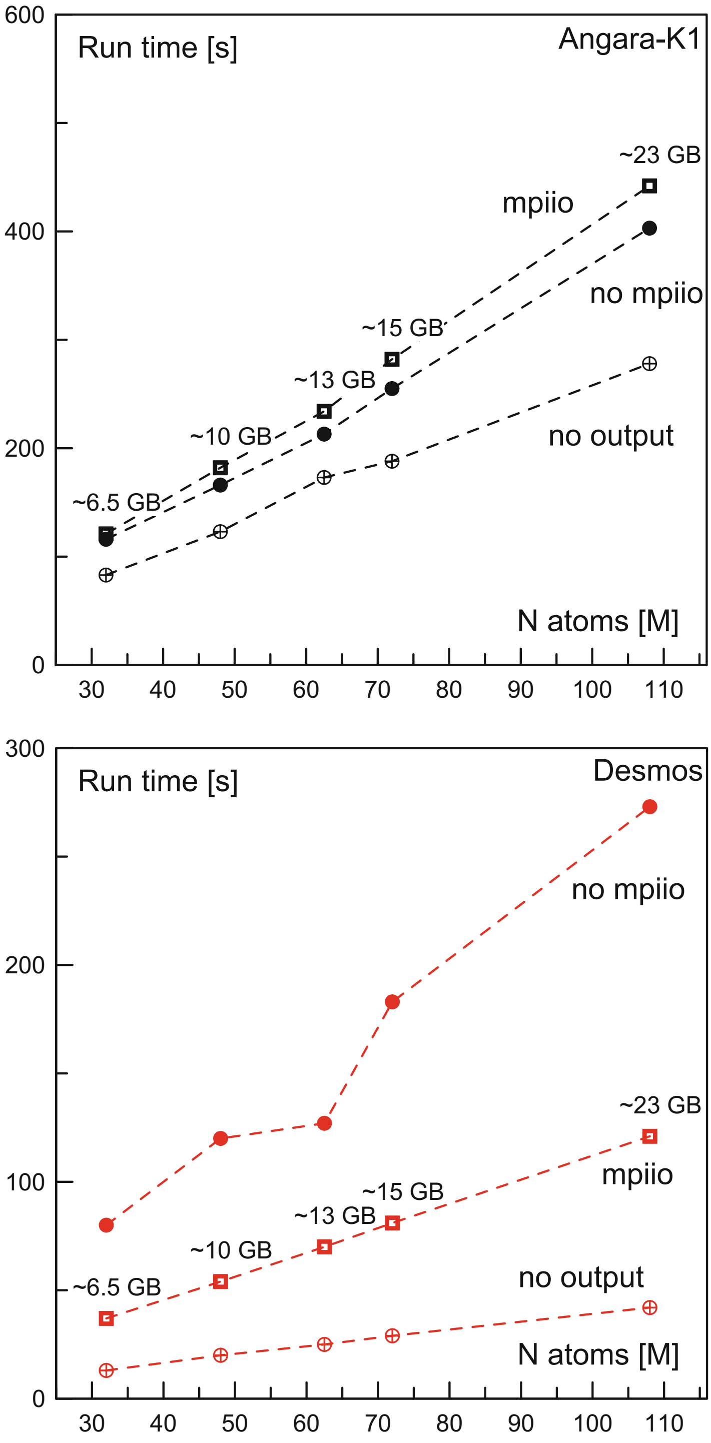

Fig. 6. Parallel output benchmarks based on the LAMMPS test model for Angara-K1 and Desmos supercomputers

7 Conclusions

The paper presents the results of efficiency and performance analyses of the Desmos supercomputer.

- The job accounting statistics of the Desmos supercomputer were reviewed. Two methods of quantitative efficiency monitoring were proposed.
- A variation of CPU frequency was attempted for energy optimization. The effect of lower energy consumption does indeed show up but the figures promise no practical benefits.
- GPU-accelerated classical MD with Gromacs runs faster and is more cost effective on supercomputers similar to Desmos than on wide-spread supercomputers based on expensive Intel Xeon multi-core CPUs.
- GPU-accelerated quantum MD can be effectively computed on Desmos nodes using single precision. A comparison with the GAMESS-US package shows that TeraChem is able to efficiently substitute double-precision CPU performance with single-precision GPU performance for solving *ab initio* problems.
- It is shown that BeeGFS effectively combines the distributed storage units located on the Desmos nodes into a single drive. MPI-IO shows a very good speed in storing data from the LAMMPS MD calculation on the Desmos parallel file system. However, LAMMPS MPI-IO shows no benefits in the case of a conventional storage benchmarked on the Angara-K1 supercomputer.

References

1. Stegailov, V., et al.: Early performance evaluation of the hybrid cluster with torus interconnect aimed at molecular-dynamics simulations. In: Wyrzykowski, R., Dongarra, J., Deelman, E., Karczewski, K. (eds.) PPAM 2017 Part I. LNCS, vol. 10777, pp. 327–336. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78024-5_29
2. Vecher, V.S., Kondratyuk, N.D., Smirnov, G.S., Stegailov, V.V.: Angara-based hybrid supercomputer for efficient acceleration of computational materials science studies. In: Proceeding of International Conference Russian Supercomputing Days 2017, pp. 557–571 (2017)
3. Neuwirth, S., Frey, D., Nuessle, M., Bruening, U.: Scalable communication architecture for network-attached accelerators. In: 2015 IEEE 21st International Symposium on High Performance Computer Architecture (HPCA), pp. 627–638 (2015). https://doi.org/10.1109/HPCA.2015.7056068
4. Puente, V., Beivide, R., Gregorio, J.A., Prellezo, J.M., Duato, J., Izu, C.: Adaptive bubble router: a design to improve performance in torus networks. In: Proceedings of the 1999 International Conference on Parallel Processing, pp. 58–67 (1999). https://doi.org/10.1109/ICPP.1999.797388
5. Scott, S.L., Thorson, G.M.: The Cray T3E network: adaptive routing in a high performance 3D torus. In: HOT Interconnects IV. Stanford University, 15–16 August 1996 (1996)
6. Adiga, N.R., et al.: Blue Gene/L torus interconnection network. IBM J. Res. Dev. **49**(2), 265–276 (2005). https://doi.org/10.1147/rd.492.0265

7. Gómez-Martín, C., Vega-Rodríguez, M.A., González-Sánchez, J.L.: Fattened backfilling: an improved strategy for job scheduling in parallel systems. J. Parallel Distrib. Comput. **97**(Suppl. C), 69–77 (2016). https://doi.org/10.1016/j.jpdc.2016.06.013
8. Kraemer, A., Maziero, C., Richard, O., Trystram, D.: Reducing the number of response time SLO violations by a Cloud-HPC convergence scheduler. In: 2016 2nd International Conference on Cloud Computing Technologies and Applications (CloudTech), pp. 293–300 (2016). https://doi.org/10.1109/CloudTech.2016.7847712
9. Mamaeva, A.A., Voevodin, V.V.: Methods for statistical analysis of large supercomputer job flow. In: Proceeding of International Conference Russian Supercomputing Days 2017, pp. 788–799 (2017)
10. Hoefler, T., Belli, R.: Scientific benchmarking of parallel computing systems: twelve ways to tell the masses when reporting performance results. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, pp. 73:1–73:12. ACM, New York (2015). https://doi.org/10.1145/2807591.2807644
11. Scogland, T., Azose, J., Rohr, D., Rivoire, S., Bates, N., Hackenberg, D.: Node variability in large-scale power measurements: perspectives from the Green500, Top500 and EEHPCWG. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2015, pp. 74:1–74:11. ACM, New York (2015). https://doi.org/10.1145/2807591.2807653
12. Höhnerbach, M., Ismail, A.E., Bientinesi, P.: The vectorization of the Tersoff multibody potential: an exercise in performance portability. In: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2016, pp. 7:1–7:13. IEEE Press, Piscataway (2016). https://doi.org/10.1109/SC.2016.6
13. Kutzner, C., Pall, S., Fechner, M., Esztermann, A., de Groot, B.L., Grubmuller, H.: Best bang for your buck: GPU nodes for gromacs biomolecular simulations. J. Comput. Chemis. **36**(26), 1990–2008 (2015). https://doi.org/10.1002/jcc.24030
14. Luehr, N., Ufimtsev, I.S., Martínez, T.J.: Dynamic precision for electron repulsion integral evaluation on graphical processing units (GPUs). J. Chem. Theory Comput. **7**(4), 949–954 (2011). https://doi.org/10.1021/ct100701w
15. Mills, N., Alex Feltus, F., Ligon III, W.B.: Maximizing the performance of scientific data transfer by optimizing the interface between parallel file systems and advanced research networks. Futur. Gener. Comput. Syst. **79**(Part 1), 190–198 (2018). https://doi.org/10.1016/j.future.2017.04.030
16. Yoo, A.B., Jette, M.A., Grondona, M.: SLURM: simple linux utility for resource management. In: Feitelson, D., Rudolph, L., Schwiegelshohn, U. (eds.) JSSPP 2003. LNCS, vol. 2862, pp. 44–60. Springer, Heidelberg (2003). https://doi.org/10.1007/10968987_3
17. Vecher, V., Nikolskii, V., Stegailov, V.: GPU-accelerated molecular dynamics: energy consumption and performance. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2016. CCIS, vol. 687, pp. 78–90. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-55669-7_7
18. Stegailov, V., Vecher, V.: Efficiency analysis of intel and AMD x86_64 architectures for Ab initio calculations: a case study of VASP. In: Voevodin, V., Sobolev, S. (eds.) RuSCDays 2017. CCIS, vol. 793, pp. 430–441. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-71255-0_35

19. Stegailov, V., Vecher, V.: Efficiency analysis of Intel, AMD and Nvidia 64-Bit hardware for memory-bound problems: a case study of Ab Initio calculations with VASP. In: Wyrzykowski, R., Dongarra, J., Deelman, E., Karczewski, K. (eds.) PPAM 2017 Part II. LNCS, vol. 10778, pp. 81–90. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78054-2_8
20. Smirnov, G.S., Stegailov, V.V.: Anomalous diffusion of guest molecules in hydrogen gas hydrates. High Temp. **53**(6), 829–836 (2015). https://doi.org/10.1134/S0018151X15060188
21. Orekhov, N.D., Stegailov, V.V.: Simulation of the adhesion properties of the Polyethylene/Carbon nanotube interface. Polym. Sci. Ser. A **58**(3), 476–486 (2016). https://doi.org/10.1134/S0965545X16030135
22. Pavlov, S.V., Kislenko, S.A.: Effects of carbon surface topography on the electrode/electrolyte interface structure and relevance to li-air batteries. Phys. Chem. Chem. Phys. **18**, 30830–30836 (2016). https://doi.org/10.1039/C6CP05552D
23. Antropov, A.S., Fidanyan, K.S., Stegailov, V.V.: Phonon density of states for solid uranium: accuracy of the embedded atom model classical interatomic potential. J. Phys.: Conf. Ser. **946**(012094), 94 (2018). https://doi.org/10.1088/1742-6596/946/1/012094
24. Logunov, M.A., Orekhov, N.D.: Molecular dynamics study of cavitation in carbon nanotube reinforced polyethylene nanocomposite. J. Phys.: Conf. Ser. **946**(1), 2044 (2018). https://doi.org/10.1088/1742-6596/946/1/012044
25. Stegailov, V.V., Orekhov, N.D., Smirnov, G.S.: HPC hardware efficiency for quantum and classical molecular dynamics. In: Malyshkin, V. (ed.) PaCT 2015. LNCS, vol. 9251, pp. 469–473. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21909-7_45
26. Aristova, N.M., Belov, G.V.: Refining the thermodynamic functions of scandium triflouride SCF3 in the condensed state. Russ. J. Phys. Chemis. A **90**(3), 700–703 (2016). https://doi.org/10.1134/S0036024416030031
27. Kochikov, I.V., Kovtun, D.M., Tarasov, Y.I.: Electron diffraction analysis for the molecules with degenerate large amplitude motions: intramolecular dynamics in arsenic pentafluoride. J. Mol. Struct. **1132**, 139–148 (2017). https://doi.org/10.1016/j.molstruc.2016.09.064
28. Stegailov, V.V., Zhilyaev, P.A.: Warm dense gold: effective ionioninteraction and ionisation. Mol. Phys. **114**(3–4), 509–518 (2016). https://doi.org/10.1080/00268976.2015.1105390
29. Minakov, D.V., Levashov, P.R.: Melting curves of metals with excited electrons in the quasiharmonic approximation. Phys. Rev. B **92**, 224102 (2015). https://doi.org/10.1103/PhysRevB.92.224102
30. Minakov, D., Levashov, P.: Thermodynamic properties of LiD under compression with different pseudopotentials for lithium. Comput Mat. Sci. **114**, 128–134 (2016). https://doi.org/10.1016/j.commatsci.2015.12.008
31. Eckhardt, W., et al.: 591 TFLOPS multi-trillion particles simulation on SuperMUC. In: Kunkel, J.M., Ludwig, T., Meuer, H.W. (eds.) ISC 2013. LNCS, vol. 7905, pp. 1–12. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38750-0_1