2018 17th International Symposium on Distributed Computing and Applications for Business Engineering and Science

A few of the most popular tools for evaluating supercomputers

Gang Xie
Institute of Computer Applications
China Academy of Engineering Physics
MianYang, China
xieg@caep.cn

Ya-lin Zhang
Institute of Computer Applications
China Academy of Engineering Physics
MianYang, China

***Abstract*—Computer benchmarks are computer programs that form standard tests of the performance of a computer and the software through which it is used. In this paper we discuss the purpose, significance and method of benchmarking supercomputers, describe the state of the art, and review a few of the mainstream benchmarks for supercomputer evaluation, among them, Linpack is the most popular for the famous TOP500 ranking list. In addition to this, there is the benchmark for the Graph500 ranking list which pays attention to the data processing ability, there are the Rodinia and SHOC benchmarks for evaluating the heterogeneous supercomputers composed of both GPUs and multi-core CPUs, there is still the DEISA benchmark for various other purposes.**

Keywords-Supercomputer;Benchmark;HPL;Graph500; DEISA ; Rodinia ; SHOC

I. INTRODUCTION

As the computer architecture advances, it becomes more difficult to compare the performance of various computer systems simply by looking at their specifications. Therefore, tests are developed to compare platforms, identify performance bottlenecks, and evaluate potential solutions. Benchmarks are designed to mimic a particular type of workload on a component or system. The basic requirements of a benchmark suite for general purpose computing include supporting diverse applications with various computation patterns, employing state-of-the-art algorithms, and providing input sets for testing different situations. Synthetic benchmarks do this by specially created programs that impose the workload on the component. Application benchmarks run real-world programs on the system. While application benchmarks usually give a much better measure of real-world performance on a given system, synthetic benchmarks are useful for testing individual components. Benchmarking is not easy and often involves several iterative rounds in order to arrive at predictable, useful conclusions. Interpretation of benchmarking data is also extraordinarily difficult.

To choose and buy a HPC system is not as easy as buying a PC. As HPC systems are very expensive, users usually need to evaluate a spectrum of platforms carefully and earnestly before decide which to purchase.

Benchmarks are used to decide the performance of a supercomputer by testing its CPU speed, network communication, memory access, and I/O servers. By revealing strengths and weaknesses of different products, benchmarks help users to decide which product can better serve their requirement and purchase accordingly.

Benchmarks aim at providing an objective standard for evaluating fairly the performance of a supercomputer system. But to be really fair is not so easy, as the performance of an application involves many factors, including the hardware, architecture, compiler optimization, programming environment, test conditions, and problem solving algorithms.

Requirement gathering is the foundation of performance evaluation, which means the collection and selection of the representative applications of a special group of users. Collection of typical applications mainly concentrates on areas most demanding for HPC systems. Only by clearly understanding the characteristics of typical applications, can we get scientific basis for selecting test programs. Application programs from different field use different algorithms, and so require different HPC systems. For example, communication intensive algorithms demand high network performance, while computation intensive algorithms demand high processor performance. By analyzing the applications, we catch the characteristics of various application programs, understand their requirements for different components of the computer system, and provide the basis for evaluation and purchase of HPC system.

When analyzing an application program, we focus on its memory access characteristics, communication characteristics, and I/O characteristics. Due to the difference in processor and memory speed, memory access patterns can seriously restrict the performance of a parallel program. Memory access patterns include sequential and random access, which can be described by data locality, while data locality of a program includes temporal locality and spatial locality. Temporal locality refers to repeated references to the same memory address. Spatial locality refers to the trend of accessing nearby memory addresses. For example, sequential access to memory addresses means good spatial locality, while frequent reference to the same data means good temporal locality.

As parallel programs complete large-scale parallel computation through inter process communication and synchronization, so the time, space and capacity characteristics of their communication mode are the major factors impacting their performance and scalability. Time characteristics of communication refer to message frequency. Space characteristics of communication refer to distribution of the message destinations. Capacity of message measures distribution features of message size.

II. STATE OF THE ART

It can be seen from the Linpack performance index of the TOP 500 supercomputers that the peak performance of

978-1-5386-7445-1/18/$31.00 ©2018 IEEE
DOI 10.1109/DCABES.2018.00024

high performance computers upgrades very fast. Because the lifting speed of CPU performance follows Moore's law, much faster than the memory, disk and other promotion, IO throughput gradually becomes the bottleneck of large systems. That’s why the Graph500 benchmark was developed. At the same time, network performance advances beyond people's expectations in recent years. This means supercomputer systems can be made larger and larger with an energy consumption of several MWs. It was this concern about operating costs that gave birth to the Green500 benchmark.

Different benchmark programs serve different purposes. Some test the performance of CPU, some test the performance of file server, some test input and output, some test network communication speed etc.. There is a plurality of ranking lists for high performance computers in the world, each with a different guidance. For example, there is the Linpack benchmark for the TOP500 ranking list which pays attention to the CPU floating point performance, there is the benchmark for the Graph500 ranking list which pays attention to the data processing ability, and there is the benchmark for the Green500 ranking list which pays attention to the performance and power ratio. So the high performance computer evaluation itself has been controversial. But because the TOP500 ranking is very influential in the industry, it has become a glory chased for by each country in the world. This has resulted in the situation of 'big machine but small application'. So the industry needs new benchmarks and ranking lists meeting the requirements of practical applications to guide the design of high performance computers. On another hand, heterogeneous supercomputers composed of both GPUs and multi-core CPUs have become more and more popular as an approach to lift the speed and reduce energy consumption. As a result of this development, corresponding benchmarks need to be established to compare various architecture design and programming environments. Thus, a new generation of evaluation tools for this class of heterogeneous systems, such as SPEC, Rodinia, and SHOC, came into being in the 2010s.

III. LINPACK

Linpack is the most popular benchmark suite in the world for testing the floating point performance of high performance computers, by using the Gauss elimination method solving dense linear algebraic equations, which is a common task in engineering. The aim is to approximate a computer’s peak performance when solving real problems. But it is a simplification, as no single computational task can reflect the overall performance of a computer system. The performance of a computer is a complex issue that depends on many interconnected variables. The performance measured by the LINPACK benchmark consists of the number of 64-bit floating-point operations, generally additions and multiplications, a computer can perform per second, also known as FLOPS. However, a computer's performance when running actual applications is likely to be far behind the maximal performance it achieves running the appropriate LINPACK benchmark. HPL means Highly Parallel Computing, is a portable implementation of Linpack’s benchmark written in C. HPL generates a linear system of equations of order n and solves it using LU decomposition with partial row pivoting. It requires installed implementations of MPI and BLAS to run.

Because it is scalable, simple and easy to operate, and its evaluation method with the CPU floating-point operation ability as the standard is direct and reliable, HPL has become the de facto standard for high performance computer evaluation. Both the world’s TOP500 and China’s TOP100 lists have adopted it as the evaluation criteria to rank high performance computer systems. However, academia and industry have been aware of some of its limitations. For HPL, the dominant cost is CPU-related because computation has higher complexity order than communication: O(n3) versus O(n2). Its performance levels are generally unobtainable, because it only tests the resolution of dense linear systems, what is not representative of all the operations usually performed in scientific computing. Now days, as high performance computer systems become more and more complex, it is too general to reflect the overall performance of a computer system only by its CPU floating point operation ability, because it cannot reflect performance bottlenecks which may exist in a computer system, and thus unable to provide a reliable and practical experience to guide future system design and construction. Especially for multi-core clusters, all the factors, such as the network bandwidth and delay, the mechanism for memory access, sharing and classification , are likely to restrict the system performance, or greatly affect the evaluation results. Thus, it is not accurate enough to evaluate the performance of a supercomputer system only by its CPU floating point operation rate.[1-9]

IV. GRAPH500

In the field of medicine, social network analysis and international security affairs, "Big Data" problems are quite common. These date intensive computing are playing an increasingly important role in HPC workloads and data centers. For big data problems, what is important for a supercomputer is not its arithmetic operation ability, but the ability of storing and exchanging massive data in an irregular, fast changing communication mode. Because current benchmarks and performance metrics do not provide useful information on the suitability of supercomputing systems for data intensive applications, alternative metrics that characterize the performance of a machine in a more holistic manner may be more relevant for many scientific applications, and may be desirable for making purchasing decisions.

The Graph500 is the first serious approach to complement the Top 500 with data intensive applications. It differs from the traditional Linpack by testing a supercomputer's skill at using graph theory to analyze the output streams from simulations in biological, security, social and similar big data problems. The project was announced on International Supercomputing Conference in June 2010. The first list was published at the ACM/IEEE Supercomputing Conference in November 2010. New versions of the list are published twice a year. The main performance metric used to rank the supercomputers is GTEPS, the number of traversed edges per second that can be performed by a supercomputer cluster, which measures both the communication capabilities and computational power of the machine. This is in contrast to the more standard metric of floating point operations per second (FLOPS), which does not give any weight to the communication capabilities of the machine. In this context, an edge is a connection between two vertices on a graph, and the traversal is the ability of the machine to communicate data between these two points.

The benchmark used in Graph500 stresses the communication subsystem, instead of counting double precision floating-point arithmetic. It is based on a breadth-first search in a large undirected graph. There are two computation kernels in the benchmark: the first kernel is a scalable data generator which produces edges containing the start vertex and end vertex for each edge; the second kernel does a parallel BFS search of some random vertices. Both kernels are timed.

As the energy consumption of HPC systems will become a limiting factor in the future, the Green Graph 500 was suggested as a benchmark for testing graph search performance similar to the Graph 500 that collects performance-per-watt metrics and acts as a forum for vendors and data center operators to compare the energy consumption of data intensive computing workloads on their architectures. A high TEPS/W value indicates a high graph search capability per unit of power consumption, or in other words low power consumption. The data is collected in collaboration with the Graph 500 list. The benchmark and the performance metrics are identical with Graph 500. It is also designed to complement the Green500 list with energy metric for data intensive computing.

The Graph 500 may add to the list of metrics that rate a supercomputer's performance. But a Graph 500 score shouldn't be seen as some definitive number any more than the Linpack score used today. As an example of applications, on Aug 6, 2014, SGI announced that the SGI UV 2000 became the fastest single node SMP, as verified by the Graph 500 benchmark test. In addition to this new performance world record, the Green Graph 500 test also revealed it to be the commercial supercomputer with the lowest power-consumption figures.

V. DEISA

DEISA is a benchmark suite for scientific HPC applications.

The Distributed European Infrastructure for Supercomputing Applications (DEISA) is a European Union supercomputer project funded by the European Commission. The project started in 2002 developing and supporting a pan-European distributed high performance computing infrastructure, which coupled eleven national supercomputing centers with a dedicated network connection.

DEISA produced a benchmark suite to assess the performance of parallel supercomputer systems. It provides a structured framework, which allows compilation, execution and analysis to be configured and carried out via standard input files. The benchmark comprises a number of real applications taken from a wide range of disciplines, including astrophysics, fluid dynamics, climate modelling, biosciences, materials science, fusion power and fundamental particle physics.

VI. RODINIA

The Rodinia benchmark suite is designed by University of Virginia to provide parallel programs for the study of heterogeneous systems with OpenMP, OpenCL and CUDA implementations. Rodinia 1.0 was first released on Mar 01, 2010. The newest version of Rodinia is Rodinia 3.0, released on July 23, 2014.

Heterogeneous computer systems that incorporate diverse accelerators and automatically select the best computational unit for a particular task are increasingly popular because they are becoming easier to program and offer dramatically better performance for many applications. These accelerators differ significantly from CPUs in architecture, middleware and programming models, offer parallelism at scales not currently available with other microprocessors. The performance of applications on these architectures requires taking advantage of multithreading, large number of cores, and specialized hardware. However, most of the previous benchmark suites focus on providing applications for conventional, general-purpose CPU architectures rather than heterogeneous architectures containing accelerators. They neither support these accelerators' APIs nor represent the kinds of applications and parallelism that are likely to drive development of such accelerators. Rodinia is released to address this problem. It provides publicly available implementations of applications for both GPUs and multi-core CPUs.

The Rodinia benchmark suite consists of four applications and five kernels, parallelized with OpenMP for CPUs and with the CUDA API for GPUs. The Similarity Score kernel is programmed using Mars' MapReduce API framework [10]. Various optimization techniques and on-chip compute resources are used. The Rodinia applications cover a diverse range of domains, including Graph Algorithms, Fluid Dynamics, Physics Simulation, Pattern Recognition, Molecular Dynamics, Data Mining, etc. Each application represents a representative application from its respective domain. Users are given the flexibility to specify different input sizes for various uses.[10-16]

VII. SHOC

SHOC (Scalable HeterOgeneous Computing) is a spectrum of programs for testing the performance and stability of non-traditional architecture computers. It was created by Jeremy Meredith at Oak Ridge National Laboratory's Future Technologies group in 2010. As the scalable heterogeneous computing systems, composed of GPUs and multi-core CPUs, become more common as one approach to performance improvement and energy efficiency, it is important to be able to compare and contrast architectural designs and programming systems in a fair and open forum. That's why SHOC is developed.

SHOC focus on heterogeneous systems and can be used on clusters as well as individual hosts. The benchmarks are divided into two primary categories: stress tests and performance tests. The stress tests use computationally demanding kernels to identify OpenCL devices with bad memory, insufficient cooling, or other device component problems. The other tests measure many aspects of system performance on several synthetic kernels. At the lowest level, SHOC assesses architectural features of the systems by micro benchmarks. At higher levels, SHOC uses application kernels to determine system-wide performance, such as intra-node and inter-node communication among devices. SHOC includes benchmark implementations in both OpenCL and CUDA in order to provide a comparison of these programming models. [17]

VIII. CONCLUSIONS

Requirement analysis is the first step of supercomputer evaluation and selection. First of all we must figure out what kind of supercomputer we want. Or putting it another way, we need to identify the characteristics of our typical application programs. For example, we should find out whether they are computation intensive, communication intensive, or memory intensive. Next step, on the basis of requirement analysis, we can choose a few benchmarks which represent the characteristics of our mainstream application programs, or choose a few of our real world applications as benchmarks. At present, Linpack is still the most recognized benchmark, especially for general purpose supercomputers. Adding to this, we can objectively evaluate a super computer system by choosing some other representative benchmark as a supplement and contrast.

REFERENCES

[1] The Linpack Benchmark, TOP500 Supercomputing Sites. http://www.top500.org/project/linpack/

[2] Dongarra, Jack J.; Luszczek, Piotr; Petitet, Antoine, "The LINPACK Benchmark: past, present and future", Concurrency and Computation: Practice and Experience (John Wiley & Sons, Ltd.): 803–820, 2003.

[3] HPL-A Portable Implementation of the High-Performance Linpack Benchmark for Distributed-Memory Computers, http://www.netlib.org/benchmark/hpl/.

[4] Gahvari, Hormozd; Hoemmen, Mark; Demmel, James; Yelick, Katherine, "Benchmarking Sparse Matrix-Vector Multiply in Five Minutes", SPEC Benchmark Workshop, 2006.

[5] Dongarra, Jack J., "The HPC Challenge Benchmark: A Candidate for Replacing Linpack in the Top500?", SPEC Benchmark Workshop, 2007.

[6] Wong, P.; van der Wijngaart, R., "NAS Parallel Benchmarks I/O Version 2.4", NAS Technical Report NAS-03-002, NASA Ames Research Center, Moffett Field, CA, January 2003.

[7] Saphir, W.; van der Wijngaart, R.; Woo, A.; Yarrow, M., New Implementations and Results for the NAS Parallel Benchmarks 2, NASA Ames Research Center, Moffett Field, CA

[8] Van der Wijngaart, R., "NAS Parallel Benchmarks Version 2.4", NAS Technical Report NAS-02-007, NASA Ames Research Center, Moffett Field, CA, October 2002.

[9] Jin, H.; Frumkin, M.; Yan, J., "The OpenMP Implementation of NAS Parallel Benchmarks and Its Performance", NAS Technical Report NAS-99-011, NASA Ames Research Center, Moffett Field, CA, October 1999.

[10] B. He, W. Fang, Q. Luo, N. K. Govindaraju, and T. Wang. Mars: a MapReduce framework on graphics processors. In Proceedings of the 17th International Conference on Parallel Architectures and Compilation Techniques, Oct 2008.

[11] M. A. Goodrum, M. J. Trotter, A. Aksel, S. T. Acton, and K. Skadron. Parallelization of Particle Filter Algorithms. In Proceedings of the 3rd Workshop on Emerging Applications and Many-core Architecture (EAMA), in conjunction with the IEEE/ACM International Symposium on Computer Architecture (ISCA), June 2010.

[12] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, Sang-Ha Lee and K. Skadron. "Rodinia: A Benchmark Suite for Heterogeneous Computing". IEEE International Symposiumon Workload Characterization, Oct 2009.

[13] J. Meng and K. Skadron. "Performance Modeling and Automatic Ghost Zone Optimization for Iterative Stencil Loops on GPUs." In Proceedings of the 23rd Annual ACM International Conference on Supercomputing (ICS), June 2009.

[14] L.G. Szafaryn, K. Skadron and J. Saucerman. "Experiences Accelerating MATLAB Systems Biology Applications." in Workshop on Biomedicine in Computing (BiC) at the International Symposium on Computer Architecture (ISCA), June 2009.

[15] M. Boyer, D. Tarjan, S. T. Acton, and K. Skadron. "Accelerating Leukocyte Tracking using CUDA: A Case Study in Leveraging Manycore Coprocessors." In Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS), May 2009.

[16] S. Che, M. Boyer, J. Meng, D. Tarjan, J. W. Sheaffer, and K. Skadron. "A Performance Study of General Purpose Applications on Graphics Processors using CUDA" Journal of Parallel and Distributed Computing, Elsevier, June 2008.

[17] Anthony Danalis, Gabriel Marin, Collin McCurdy, Jeremy S. Meredith, Philip C. Roth, Kyle Spafford, Vinod Tipparaju, Jeffrey S. Vetter, The Scalable HeterOgeneous Computing (SHOC) Benchmark Suite, Proceedings of 3rd Workshop on General Purpose Processing on Graphics Processing Units, GPGPU 2010, Pittsburgh, Pennsylvania, USA, March 14, 2010.