Seminar Report

Modern Benchmarking Strategies of HPC Systems

Nils Rosenboom

MatrNr: 21879930

Supervisor: Dr. Narges Lux

Georg-August-Universität Göttingen
Institute of Computer Science

March 30, 2024

Abstract

- Benchmarking plays a crucial role in assessing the performance of high-performance computing (HPC) systems, guiding decision-making and optimization efforts in this rapidly evolving field.
- The complexity and heterogeneity of modern HPC architectures pose significant challenges for benchmarking, including the lack of representative benchmarks that capture real-world application performance accurately.
- Traditional benchmarks like LINPACK and NAS Parallel Benchmarks (NPB) have been widely used but suffer from limitations such as focusing on low-level operations and failing to reflect the diversity of HPC workloads accurately.
- We introduce SPEChpc 2021, a benchmark suite specifically designed for heterogeneous computing systems, offering a comprehensive evaluation of system performance across diverse computational tasks and configurations.
- Through case studies and performance metrics, we demonstrate the effectiveness of SPEChpc 2021 in identifying system-level issues, optimizing performance, and providing valuable insights for decision-makers in the HPC community.

Declaration on the use of ChatGPT and comparable tools in the context of examinations

In this work I have used ChatGPT or another AI as follows:

☐ Not at all

☑ During brainstorming

☐ When creating the outline

☐ To write individual passages, altogether to the extent of -% of the entire text

☐ For the development of software source texts

☐ For optimizing or restructuring software source texts

☑ For proofreading or optimizing

☑ Further, namely: Transforming bullet points into continuous sentences to the extend of 20% of the entire text

I hereby declare that I have stated all uses completely.
Missing or incorrect information will be considered as an attempt to cheat.

Contents

List of Tables

1 Introduction

“Benchmarking is the practice of comparing business processes and performance metrics to industry bests and best practices from other companies. Dimensions typically measured are quality, time and cost.” [1]

As many of the audience are probably familiar with benchmarking personal computers, such as measuring the framerate in a videogame or similar activities, it’s important to note that benchmarking is also crucial in the realm of high-performance computing. However, in the given example, it is noteworthy that you might want to run a standardized 3D benchmark to assess how a system performs in different games without measuring the framerate in each game individually. Therefore, these benchmarks should measure something relevant to assess the performance in real tasks that a system is about to perform.

As HPC systems continue to evolve, becoming increasingly complex and heterogeneous, benchmarks that offer comparable metrics are more challenging to create. However, benchmarking is crucial to assess a system’s performance against expectations and industry standards. It can be employed to identify hardware or software configuration issues and validate a system’s reliability. Also, it serves as a foundation for informed decision-making regarding upgrades or changes to a system.

The challenges arising from benchmarking modern HPC systems include rapid progress in hardware development, difficulties in designing well-scaling benchmarks, defining the metrics to measure, dealing with heterogeneous task fields, and variations in system performance across different task sets.

This is especially underlined as many widely used Benchmarks today are outdated or only measure the performance of low-level operations which do not reflect real world applications in many cases. [2] In this paper, we explore widely used benchmarks and concepts while highlighting their issues. Also, general guideline and considerations for benchmarking modern HPC systems are discussed.

1.1 Parallel Computing

HPC systems are utilized for executing large-scale parallel systems, introducing additional complexity that aligns with the substantial computing power they effectively employ. Multiple types of parallelism, such as node-level parallelization and communication between nodes, exist in HPC systems, which consist of numerous compute nodes communicating with each other. Additionally, each node features multiple CPU cores and different accelerator cards. Common techniques for node-level parallelization include OpenMP or OpenACC, while the Message Passing Interface facilitates communication between nodes, enabling message parsing in systems with distributed memory. [3]

It is important to highlight that performance is not limited by just raw compute power but instead also depends on communication between nodes and thus network infrastructure as well as resource utilization. In Addition, performance can also be dependent on how well software is compiled for the System that is to be tested. [3]

Figure 1: Examples for Parallel Computing Tasks `https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial`

Common applications for HPC systems encompass scientific simulations, such as galaxy formations, planetary movements, climate change modeling, weather forecasts, and traffic predictions as can be seen in Figure 1:

1.2 Benchmarks presented in this Paper

Currently widely used and known Benchmarks for HPC Systems include Linpack [4], HPC-Challenge [5], NAS [6] and SPEChpc 2021 [6]. While the first have been developed decades ago, we also delve into the newly developed SPEChpc 2021 Benchmark Suite to see how Benchmarking has evolved and what is the state of the Art in terms of Benchmarking.

Also, the Spec consortium, which represents different partners from industry and academia, provides guidance on what a good hpc benchmarks should look like and how to assess their results. The topic is especially important as many currently used Benchmarks are outdated or only represent low level processes, while it is difficult to choose the right benchmark for a given system and workload. [7]

2 Traditional HPC Benchmarking

In the past we have seen many Benchmarks that measure the performance based on low level operations. A famous example for this is the LINPACK Benchmark as it is also used in the largely PR relevant TOP500 ranking. [8]

2.1 LINPACK

The LINPACK Benchmark, conceived by Jack Dongarra in the 1970s, has stood as a tool for evaluating the floating-point computing prowess of high-performance computing systems. As we delve into this topic, it becomes apparent that LINPACK is not without its constraints, and the utilization of this benchmark in the TOP500 ranking prompts reflections on its relevance in representing actual HPC performance. [9]

LINPACK's limitations stem from its design, with a primary focus on floating-point performance to the detriment of other vital aspects of HPC systems. The benchmark tends to disregard factors such as I/O efficiency, memory hierarchy, and interconnect efficiency, thereby providing an incomplete picture of a system's capabilities. [9]

Moreover, LINPACK's reliance on a specific LU factorization algorithm introduces a level of algorithmic specificity that may not align with the requirements of all HPC applications. Systems optimized for this particular algorithm may demonstrate disproportionately favorable results, potentially leading to misguided assessments of overall performance. Another facet of concern is LINPACK's emphasis on single-precision performance. While this may be relevant for certain applications, it can create a skewed perspective on a system's computational efficiency, as real-world applications often demand a mix of precision types. [9]

Perhaps most crucially, LINPACK may measure performance that is practically unattainable in real applications unless meticulously optimized for a specific system. This raises doubts about the benchmark's applicability to diverse HPC workloads, where optimization for a single system may prove impractical or counterproductive. [9]

2.2 TOP500

The reliance on the LINPACK Benchmark in the TOP500 List raises concerns about its representativeness for real-world applications due to its outdated development in the 1970s. Consequently, decision-makers may struggle to find systems that meet their specific needs accurately. Moreover, the TOP500 list's lack of representation for real-world applications and transparency regarding test circumstances further diminishes its practical utility and credibility, hindering stakeholders' ability to make well-informed decisions. [10]

Lastly, the TOP500 list, often influenced by public relations considerations, may prioritize superficial rankings over nuanced understandings of a system's performance characteristics. [10]

2.3 NAS Parallel Benchmark

The NAS Parallel Benchmark (NPB), developed by NASA Ames Research Center in the 1990s, has been a crucial tool for assessing the capabilities of parallel computing systems. [6]

The NPB was conceived at a time when parallel computing was gaining prominence in scientific and research domains. NASA Ames Research Center spearheaded the initiative to create a standardized set of benchmarks that could effectively measure the performance of parallel systems. The benchmarks were designed to simulate real-world scientific applications, providing a comprehensive evaluation of different aspects of parallel computing. [11]

The NPB encompasses a range of scientific applications, each serving as a representative workload for specific computational tasks. Notable examples include Integer Sort, random memory access, Conjugate Gradient, discrete 3D fast Fourier Transform, and all-to-all communication. These applications are supposed to mirror the computational challenges faced in scientific simulations and data-intensive computations, making the NPB a versatile tool for assessing the performance of parallel systems. [11]

As technology advanced, the NPB underwent modifications to adapt to new parallel computing paradigms. Modern versions of NPB utilize parallel programming models such as MPI (Message Passing Interface) and OpenMP (Open Multi-Processing). This evolution ensures that the benchmark remains relevant and applicable to contemporary HPC architectures, providing a consistent framework for assessing the performance of parallel computing systems. [6]

The NPB categorizes its benchmarks into different classes based on problem size, facilitating a graded evaluation of computing systems. These classifications include:

1. Class S: Small, designed for quick test purposes.
2. Class W: Workstation size, reflecting the computing power of 1990s workstations, now considered relatively small.
3. Classes A, B, C: Standard test problems, with a fourfold increase in size going from one class to the next.
4. Classes D, E, F: Large test problems, featuring a substantial 16X size increase from each of the previous classes.

2.4 HPC Challenge

The High-Performance Computing Challenge (HPCC) presents a comprehensive suite of benchmarks aimed at evaluating HPC systems. This suite incorporates diverse benchmarks such as LINPACK, DGEMM, STREAM, PTRANS, and RandomAccess to provide an assessment of various system characteristics. One of the notable strengths of HPCC lies in its ability to address a broader range of attributes than the traditional LINPACK benchmark alone. However, some critics contend that despite its inclusivity, HPCC may still fall short of capturing all facets of real-world HPC applications. Additionally, concerns have been raised about the potential for over-optimization towards specific benchmarks within the suite, potentially diverting focus from other essential aspects of HPC performance. [12]

Despite these considerations, HPCC remains a valuable tool for benchmarking and evaluating the capabilities of hpc systems. Yet in contrast to the TOP 500 there cannot be only one metric to determine how good a system as many different tests are performed. [12]

2.5 Conclusion of Traditional Benchmarking

In conclusion, traditional benchmarking methods, while scalable through different versions, suffer from significant limitations. They lack real-world representation, often relying on precompiled benchmarks that may not accurately reflect the diverse performance requirements of modern applications. As a result, decision-makers face challenges in selecting HPC systems that align with their specific needs and priorities, if they rely only on those.

3 SPEC

The Standard Performance Evaluation Corporation (SPEC) is a notable non-profit consortium that plays a pivotal role in the realm of performance evaluation. Established with the primary goal of developing and maintaining benchmark suites, SPEC operates as a collaborative effort involving various entities, including industry leaders, universities, and international standards bodies.

The High-Performance Group within SPEC boasts a diverse membership that includes major players in the technology sector such as AMD, Cisco, Dell, HP, Intel, Lenovo, NVIDIA, and Supermicro. Additionally, universities from countries like the USA, China, South Korea, and Germany actively contribute to SPEC's initiatives. The consortium's involvement extends to various specialized groups, including the International Standards Group, Open Systems Group, and Research Group, highlighting the widespread collaboration and global perspective embedded in SPEC's mission. [13]

SPEC's history is marked by the continuous evolution of benchmark suites tailored to assess the performance of computing systems. Some noteworthy benchmarks in this trajectory include SPECaccel 2023, SPEC ACCEL, SPEChpc 2021, SPEC MPI 2007, and SPEC OMP 2012. These benchmarks are specifically designed to evaluate the capabilities of HPC systems, addressing various aspects such as acceleration, parallel processing, and shared memory performance. [14]

Moreover, SPEC recognizes the importance of performance evaluation beyond HPC systems. To cater to a broader range of computing environments, SPEC has developed benchmarks for non-HPC systems, including Java Client/Server, Storage, Power, and Cloud. A key aspect of SPEC's operation is the review and publication of submitted results. This transparency ensures that benchmarking processes are fair, standardized, and reliable, allowing stakeholders to make informed decisions based on the performance metrics generated by SPEC benchmarks.

SPEC stands as a collaborative force, bringing together industry leaders, academic institutions, and international bodies to develop and maintain benchmark suites that set the standard for performance evaluation. Its history and diverse set of benchmarks illustrate a commitment to addressing the evolving landscape of computing systems, encompassing both HPC and non-HPC environments.

3.1 Common Mistakes in Benchmarking

The SPEC Consortium has identified common misconceptions that often arise in the advertisement of benchmarks, shedding light on potential pitfalls and offering insightful solutions to enhance the reliability of performance evaluations. [15]

1. Running Loops 1 Billion Times

One frequently encountered challenge involves the repetition of loops, where compiler variations can significantly impact results through "dead code elimination." To address this, benchmarks should meticulously select loop operations that produce observable outputs, preventing compiler optimizations. It is essential to adopt compiler-neutral coding practices to mitigate disparities between different compilers, ensuring the evaluation's intended rigor and accuracy.

2. Unchecked Floating Point Differences

Another common issue arises when benchmarks print answers without validating them, leaving room for minor floating-point variations to lead to divergent program paths or incomplete error handling. Addressing this challenge requires the implementation of a robust validation mechanism within a defined tolerance. Furthermore, understanding the impact of these variations on the overall execution path is crucial to prevent potentially misleading conclusions.

3. Benchmark Already Compiled

When benchmarks are pre-compiled, there is limited flexibility for testing new hardware, operating systems, or compilers. The solution lies in advocating for source code benchmarks, offering adaptability to different systems and enabling testing across a broader range of environments. This flexibility is paramount for ensuring the benchmark's relevance and applicability in diverse computing scenarios.

4. Benchmark Measurement of X

Verification challenges emerge when benchmark descriptions lack thorough analysis, leading to skewed results dominated by setup time. Rigorous profiling of data before release is essential to accurately measure intended operations. Identifying and addressing potential bottlenecks in the setup process ensures meaningful performance metrics, with verification mechanisms confirming that the benchmark primarily measures targeted operations rather than setup overhead.

5. Modified Version of Well-Known Benchmark

The modification of well-known benchmarks introduces complexities, with a lack of documentation risking broken comparability. Addressing this requires a systematic process involving detailed write-ups of modifications, ensuring comparability is maintained. Establishing a verification protocol provides confidence in the reliability and relevance of the modified benchmark.

6. Benchmark of Low-Level Operations

Challenges arise when questioning the representativeness of low-level operations for real-world applications. The preference for benchmarks derived from real applications mitigates this concern, offering a more accurate reflection of system performance under practical scenarios. This approach aligns benchmark results with the challenges posed by actual applications, enhancing relevance and applicability.

3.2 SPEChpc 2021 Benchmark Suite

The SPEChpc 2021 benchmark is specifically crafted for heterogeneous computing systems, emphasizing versatility and adaptability across a spectrum of computational tasks. This benchmark is characterized by its inclusion of a diverse array of tasks stemming from various fields, making it a tool for evaluating the performance of heterogeneous systems in real-world scenarios. Offering flexibility in usage, SPEChpc 2021 is available in different sizes, catering to the varied requirements of computing environments. [15]

Moreover, it provides users with the option to employ different extensions, ensuring compatibility with a range of programming paradigms. These extensions encompass pure MPI for distributed memory parallelism, MPI + OpenACC for accelerated computing, MPI + OpenMP for shared-memory parallelism, and MPI + OpenMP with target offload, allowing users to tailor their evaluation based on specific computing configurations and requirements. [15]

SPEChpc 2021 is deliberately designed to comprehensively assess the performance of heterogeneous computing systems by intentionally considering various key components beyond just the processor. The benchmark considers the following critical aspects:

Processor: This includes the CPU chip(s) and, optionally, acceleration devices such as GPUs.

Memory: The memory hierarchy, covering aspects like caches and main memory, is a crucial factor in system performance.

Interconnects: The communication infrastructure between nodes in a cluster is a pivotal element in distributed computing.

Compilers: The benchmark intentionally evaluates the performance of C, C++, and Fortran compilers, including their optimizers, so it can also be used to compare different compilers and optimizations.

MPI Implementation: SPEChpc 2021 accounts for the Message Passing Interface (MPI) implementation, a critical element in parallel and distributed computing.

Notably, SPEChpc 2021 has a specific focus and is not intended to test graphics, Java libraries, or the I/O system. Relying on all the different components also enables to test individual components so the benchmark can also be used to compare different compilers or network interfaces. [15]

3.3 Running SPEChpc 2021

SPEChpc 2021 is freely available for non-commercial use, providing users with a valuable tool for assessing the performance of heterogeneous computing systems. To utilize this benchmark, certain requirements must be met: [15]

- Main Memory: The benchmark demands varying amounts of main memory based on the chosen configuration. The specifications include a minimum of 40GB for the Tiny configuration, 480GB for the Small configuration, 4TB for the Medium configuration, and 14.5TB for the Large configuration.
- Disk Space: Users are required to allocate a minimum of 50GB of disk space for the benchmark.
- Compilers: C, C++, and Fortran compilers are essential prerequisites for running the benchmark as it is distributed as source code.
- MPI Implementation: The benchmark specifically calls for a Message Passing Interface (MPI) implementation configured for use with the compilers specified earlier.
- CPU Architecture: To run SPEChpc 2021, users must have a system with an ARM, Power ISA, or x86_64 CPU(s).

3.4 Results

The results from SPEChpc 2021 are presented through a set of metrics, with a single composite score indicating performance, where a higher score is considered better. These scores are designed for easy comparison with other results within the same suite, providing a relative measure of system performance. The primary metrics employed in SPEChpc 2021 include time, representing the seconds required to complete a workload, and throughput, denoting the amount of work accomplished per unit of time, such as jobs per hour. Notably, SPEChpc 2021 operates as a time-based, strong scaling metric, emphasizing its focus on evaluating system performance as it scales under increasing workloads. [16]

A reference machine serves as a benchmark against which the performance of other systems is evaluated. For each benchmark, a performance ratio is calculated by dividing the time taken on the reference machine by the time taken on the System Under Test (SUT). For instance, if the reference machine completed a benchmark task like 505.lbm_t (Fluid Dynamics) in 2250 seconds, and a particular SUT accomplished the same task in 444 seconds, the performance ratio would be calculated as:

$2250 / 444 = 5.067567$

Notably, TU Dresden's Taurus System functions as the reference system, always having a performance ratio score of 1. This method enables a standardized comparison of the performance of various systems relative to the reference machine. [15]

In the context of performance measurement, the base metric serves as a fundamental reference point for consistency and comparability. To establish a base metric, benchmarks are compiled using identical flags and in the same order, ensuring uniformity in the compilation process. Additionally, the base metric mandates the use of the same node-level parallel model, the same number of ranks, and the same number of host threads per rank during the benchmarking process. This meticulous alignment in compilation and execution parameters ensures that the base metric accurately reflects the baseline performance of a system under specific conditions. Importantly, all reported results must include the base metric, providing a standardized foundation for evaluating and comparing performance across different systems and configurations.

In the measurement framework of Base and Peak metrics, the Peak metric introduces a level of flexibility to accommodate variations in system configurations and optimization strategies. Unlike the Base metric, the Peak metric allows for the utilization of different compiler options, enabling benchmarking under diverse compilation scenarios. Moreover, varying node-level parallel models may be employed for each benchmark within the Peak metric, allowing for a more tailored approach to parallelization strategies. The number of ranks and threads can be individually set for each benchmark, offering a more nuanced evaluation of performance across different computing tasks. Additionally, the Peak metric permits limited source code modifications to fine-tune directive models, such as OpenACC and OpenMP, to better align with the specific characteristics of the underlying system. [16]

In the current Ranking Nvidias DGX SuperPOD is leading the list for the medium sized benchmark. [17] It utilizes the OpenMP + ACC version of the benchmarks as it demonstrates the performance of a GPU heavy system.

3.5 Case Studies

In the case studies conducted with SPEChpc 2021, two noteworthy instances shed light on system performance discrepancies and the impact of underlying issues.

Case Study 1: RWTH Aachen

The examination of RWTH Aachen revealed a significant performance deficit compared to similar HPC systems. Analysis of the performance data indicated variations in execution times, particularly in MPI time, with a notable impact on MPI_Allreduce. Further investigation revealed the presence of faulty Memory DIMMs, which could be fixed to achieve the expected performance. [2]

Case Study 2: TU Dresden

Similar challenges were identified in TU Dresden, highlighting the importance of diagnosing and resolving system-level issues. The performance profile of TU Dresden mirrored that of RWTH Aachen, indicating a comparable situation. The root causes included a faulty BIOS configuration on several nodes, a kernel bug, and an unfavorable SLURM configuration. These findings underscore the need for comprehensive diagnostics to optimize system performance. [2]

3.6 Limitations

SPEChpc 2021, like any standardized benchmark, has inherent limitations that users should be mindful of when interpreting results and making decisions. One fundamental limitation is that the ideal benchmark for selecting a vendor or product is often one tailored to your specific workload and application. While SPEChpc 2021 provides valuable insights, it may not perfectly mirror the intricacies and nuances of your unique computing environment.

Another critical consideration is that no standardized benchmark can perfectly model the realities of a particular system and its user community. Systems vary in terms of architecture, usage patterns, and optimization requirements, making it challenging for a universal benchmark to capture every relevant aspect accurately. Given these limitations, it is crucial to recognize the uniqueness of your workload and application when assessing benchmark results. [15]

4 Efficient Computing

Factors such as the escalating cost of electricity and a growing environmental consciousness have propelled the computing community towards more energy-efficient solutions. Initiatives like the Green500 and benchmarking standards such as SPECpower have emerged as vital benchmarks, providing metrics that evaluate computational prowess against power consumption.

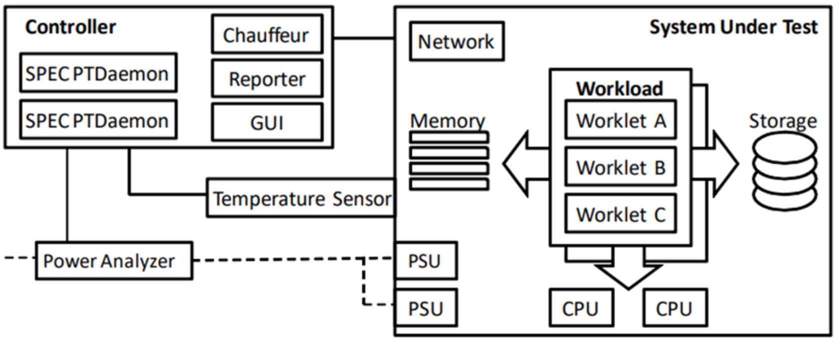

Figure 2: `https://www.spec.org/sert2/SERT-designdocument.pdf`

As displayed in Figure 2, SPECpower utilizes a separate controller system, which is connected to the system under test through a network interface. The benchmark is launched through its GUI from this system. It also employs a temperature sensor and a power analyzer. This setup allows the controller system to monitor which tasks are performed, when they occur, and how much power is consumed during specific tasks. In addition to that, temperature can also be monitored to provide additional insight.

In the SERT (Server Efficiency Rating Tool) framework, various components contribute to the assessment of energy efficiency. All SERT worklets, excluding the Idle state, operate at multiple load levels. Each of these load levels undergoes a distinct energy efficiency calculation. The per load level energy efficiency, denoted as Effload, is determined by the ratio of Normalized Performance to Power Consumption.

$$\mathsf{Eff}_{\mathsf{load}} = \frac{\mathsf{Normalized\ Performance}}{\mathsf{Power\ Consumption}}$$

This formulation captures the essence of energy efficiency, where the efficiency of the system is evaluated concerning its computational performance and the corresponding power consumption. The inclusion of multiple load levels in the assessment ensures a comprehensive understanding of energy efficiency across varying operational states, providing valuable insights into the system's overall efficiency under diverse workloads. [18]

5 Discussion

In our exploration of modern HPC benchmarking methods, we've delved into a variety of approaches, including LINPACK, HPCC, NPB, and SPEC benchmarks (SPEC 2021 and SPECpower).

LINPACK, a long-standing benchmark, has been widely used for its simplicity and accessibility. However, its focus on solving dense systems of linear equations may not capture the full spectrum of tasks typical in HPC applications. This limitation has prompted the development of more comprehensive benchmarks like HPCC and NPB. It was also highlighted that lists like TOP500 and Green500 should be interpreted with caution as they're not measuring representative tasks for real world applications and also lack transparency when it comes to documenting under which circumstances tests have been performed.

HPCC evaluates systems across multiple dimensions, including computational performance, communication efficiency, and memory access patterns. Similarly, NPB provides a suite of benchmarks covering various scientific computations, offering a more nuanced assessment of HPC systems' capabilities.

The SPEC benchmarks, particularly SPEC 2021 and SPECpower, introduce standardized methodologies for evaluating both performance and power consumption. SPEC 2021 encompasses a diverse set of benchmarks representing real-world applications, providing a more realistic evaluation of HPC systems' performance across different workloads. Meanwhile, SPECpower focuses specifically on energy efficiency, enabling comparisons based on power consumption metrics.

Despite their strengths, each benchmarking method has its limitations. For instance, some benchmarks may focus too narrowly on specific performance metrics or fail to adequately represent the diversity of real-world workloads. Additionally, while power consumption measurements have become increasingly important, they may not always be integrated into benchmarking processes.

Looking ahead, the integration of power consumption measurements, as demonstrated in SPEC 2021 and SPECpower, holds promise for more comprehensive evaluations of HPC systems. By considering both performance and energy efficiency metrics, future benchmarking efforts can provide a more holistic understanding of system capabilities and inform decisions related to system design, optimization, and sustainability.

Literature

1. Benchmarking. In: Wikipedia [Internet]. 2023 [cited 2024 Mar 18]. Available from: `https://en.wikipedia.org/w/index.php?title=Benchmarking&oldid=1184618407`

2. Brunst H, Chandrasekaran S, Ciorba FM, Hagerty N, Henschel R, Juckeland G, et al. First Experiences in Performance Benchmarking with the New SPEChpc 2021 Suites. In: 2022 22nd IEEE International Symposium on Cluster, Cloud and Internet

Computing (CCGrid) [Internet]. Taormina, Italy: IEEE; 2022 [cited 2024 Jan 12]. p. 675–84. Available from: `https://ieeexplore.ieee.org/document/9826013/`

3. Introduction to Parallel Computing Tutorial | HPC @ LLNL [Internet]. [cited 2024 Mar 29]. Available from: `https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial`

4. LINPACK [Internet]. [cited 2024 Mar 29]. Available from: `https://netlib.org/linpack/`

5. HPC Challenge [Internet]. [cited 2024 Mar 29]. Available from: `https://hpcchallenge.org/hpcc/`

6. NAS Parallel Benchmarks [Internet]. [cited 2024 Mar 18]. Available from: `https://www.nas.nasa.gov/software/npb.html`

7. The SPEC Organization [Internet]. [cited 2024 Mar 29]. Available from: `https://www.spec.org/spec/`

8. The Linpack Benchmark | TOP500 [Internet]. [cited 2024 Mar 29]. Available from: `https://www.top500.org/project/linpack/`

9. Dongarra JJ, Luszczek P, Petitet A. The LINPACK Benchmark: past, present and future. Concurr Comput Pract Exp. 2003;15[9]:803–20. 10. Kramer WTC. Top500 versus sustained performance: the top problems with the top500 list - and what to do about them. In: Proceedings of the 21st international conference on Parallel architectures and compilation techniques [Internet]. New York, NY, USA: Association for Computing Machinery; 2012 [cited 2024 Mar 29]. p. 223–30. (PACT '12). Available from: `https://dl.acm.org/doi/10.1145/2370816.2370850`

11. Bailey DH, Schreiber RS, Simon HD, Venkatakrishnan V, Weeratunga SK, Barszcz E, et al. The NAS parallel benchmarks—summary and preliminary results. In: Proceedings of the 1991 ACM/IEEE conference on Supercomputing - Supercomputing '91 [Internet]. Albuquerque, New Mexico, United States: ACM Press; 1991 [cited 2024 Mar 29]. p. 158–65. Available from: `http://portal.acm.org/citation.cfm?doid=125826.125925`

12. Luszczek P, Bailey D, Dongarra J, Kepner J, Lucas R, Rabenseifner R, et al. S12-The HPC challenge (HPCC) benchmark suite. SC '06: Proceedings of the 2006 ACM/IEEE Conference on Supercomputing. 2006. 213 p.

13. The SPEC Consortium [Internet]. [cited 2024 Mar 18]. Available from: `https://www.spec.org/consortium/`

14. SPEC Benchmarks and Tools [Internet]. [cited 2024 Mar 18]. Available from: `https://www.spec.org/benchmarks.html`

15. Overview - SPEChpc 2021 [Internet]. [cited 2024 Mar 29]. Available from: `https://www.spec.org/hpc2021/docs/overview.html`

16. SPEChpc 2021 Results [Internet]. [cited 2024 Mar 29]. Available from: `https://www.spec.org/hpc2021/results/`

17. Fourth Quarter 2022 SPEC HPC2021 Results [Internet]. [cited 2024 Mar 18]. Available from: `https://www.spec.org/hpc2021/results/res2022q4/`

18. SPECpower_ssj® 2008 [Internet]. [cited 2024 Mar 29]. Available from: `https://www.spec.org/power_ssj2008/`