HPS

nils.rosenboom@stud.uni-goettingen.de

Nils Rosenboom

Modern Methods of HPC-Benchmarking

Unveiling Contemporary Approaches in HPC Benchmarking

Table of contents

Benchmarking at home

What is Benchmarking?

- Performance Evaluation
- Standardized testing

Example: Building computer to maximize framerate in favorite video games

Image source: https://www.reddit.com/r/pcmasterrace/comments/db8zew/the_real_pc_benchmark/

Benefits of Benchmarking

Why is benchmarking important?

- Assess Performance Against Expectations
- Pinpoint Hardware and Software Configuration Issues
  - E.g. Misconfigured BIOS (wrong Clock Speeds)
  - Missing RAM DIMMs
- Enable Comparison with Industry Standards
- Validate System Reliability
- Informed Decision-Making for Upgrades or Changes

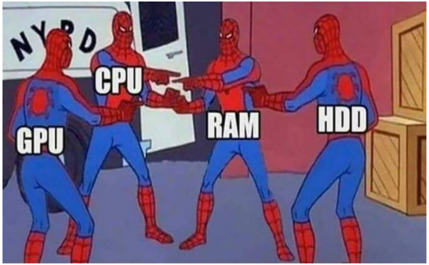


Image source: https://i.pinimg.com/736x/d7/57/7d/d7577d5adb9790df39160c8297f07e5c.jpg

Challenges in Benchmarking

- Increasingly Complex and heterogeneous systems
- Progress in Hardware development is too fast
- Difficult to design well scaling benchmarks
- What are you measuring?
- Heterogeneous fields of tasks
- System performance varies over different tasksets

HPC Systems - What's different?

Execute code on large parallel Systems

- Additional Complexity
- Much larger compute power
- Hugely parallel
- Many CPUs and GPUs or other accelerator cards
- Different Kinds of Parallelism
- Network and Message Parsing between Notes
  - (Network-) Interfaces also become Important
- New Questions arise:
  - How well does the performance scale with the number of nodes?
  - Power Consumption?

Parallel Computing

- Simultaneous execution of multiple tasks, improving performance by dividing a problem into smaller parts and solving them concurrently.
- Different kinds of parallelism:
  - SIMD (Single Instruction, Multiple Data)
  - MIMD (Multiple Instructions, Multiple Data)

Image: https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial

Parallel Computing

Benefits:

- Increased computational speed.
- Efficient utilization of resources.
- Scalability for larger problem sizes.

Considerations:

- Communication overhead in distributed systems.
- Load balancing to ensure optimal resource usage.
- Code complexity and potential for synchronization issues.

Parallel Computing - Examples

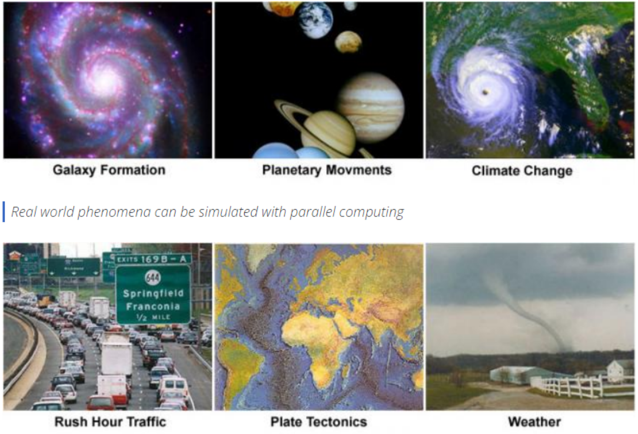


Image: https://hpc.llnl.gov/documentation/tutorials/introduction-parallel-computing-tutorial

Principles of Parallel Computing

- **MPI (Message Passing Interface)**:
  - Communication framework for distributed memory systems.
  - Enables coordination among multiple processors by exchanging messages.
  - Commonly used in cluster and supercomputer environments.
- **OpenMP**:
  - API for shared-memory parallelization (Node Level Parallelism)
  - Adds parallelism to existing code through compiler directives.
  - Facilitates the creation of multithreaded applications for enhanced performance.
- **OpenACC**:
  - Accelerator directive-based approach.
  - Designed for heterogeneous computing environments, targeting GPUs and other accelerators.
  - Simplifies parallel programming by adding directives to high-level languages like C, C++, and Fortran.

Parallel Code & Compiler Optimization

```
#pragma omp parallel
{
    #pragma omp for
    {
        for(int i = 0; i < ARRAY_SIZE; i++)
        {
            arr[i] = arr[i] / arr[i] + arr[i] / 5 - 14;
        }
    }
}
```

Use "-fopenmp" flag to compile:

- g++ hello.cpp -o hello -fopenmp

Popular HPC Benchmarks

- LINPACK
- HPC Challange
- NAS Parallel Benchmark
- SPEChpc 2021

LINPACK Benchmark

- Developed by Jack Dongarra in the 1970s *6*
- Measures a computer's floating-point computing power
- Widely used for ranking supercomputers in the TOP500 list
- However, it has its limitations and may not represent real-world performance for all applications

Weaknesses of LINPACK Benchmark

- **Limited Scope:** LINPACK focuses primarily on floating-point performance, neglecting other important aspects of HPC systems, such as I/O, memory hierarchy, and interconnect efficiency
- **Algorithmic Specificity:** The benchmark relies on the specific LU factorization algorithm
- **Single Precision Emphasis:** LINPACK tends to emphasize single-precision performance
- Measures performance that is unattainable in real applications unless meticulously optimized for one system only

Top500

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Aurora** - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	4,742,808	585.34	1,059.33	24,687
3	**Eagle** - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Microsoft Azure United States	1,123,200	561.20	846.84	

https://www.top500.org/lists/top500/2023/11/

What’s wrong with Top500

- Based on LINPACK
- Outdated benchmark
- Hardly represents any real world application
- Intransparent
  - no Information about the test circumstances
  - no Information about used Hardware & Software
- Mostly PR relevant

NAS Parallel Benchmark (NPB)

- Developed by NASA Ames Research Center in the 1990s *4*
- Mimics a set of scientific applications
- Examples: Integer Sort, random memory access, Conjugate Gradient, discrete 3D fast Fourier Transform, all-to-all communication
- Nowadays different versions exist utilzing MPI and OpenMP
- Different sizes classified as:
  - Class S: small for quick test purposes
  - Class W: workstation size (a 90's workstation; now likely too small)
  - Classes A, B, C: standard test problems; 4X size increase going from one class to the next
  - Classes D, E, F: large test problems; 16X size increase from each of the previous classes

HPCC (High-Performance Computing Challenge)

- A suite of benchmarks designed to assess HPC systems comprehensively *5*
- Includes HPL (LINPACK), DGEMM, STREAM, PTRANS, and RandomAccess benchmarks
- **Strengths:** Addresses a broader range of system characteristics than LINPACK alone
- **Weaknesses:** Some argue that it still does not cover all aspects of real-world HPC applications, and the emphasis on specific benchmarks might lead to over-optimization for those

Standard Performance Evaluation Corporation

- Non-profit consortium
- Develops and maintains benchmark suites
- Reviews and publishes submitted results

Who's involved? *1*

- High Performance Group:
  - AMD, Cisco, Dell, HP, Intel, Lenovo, NVIDIA, Supermicro ...
  - Universities from USA, China, Southkorea, Germany ...
- International Standards Group
- Open Systems Group
- Research Group

History of SPEC Benchmarks

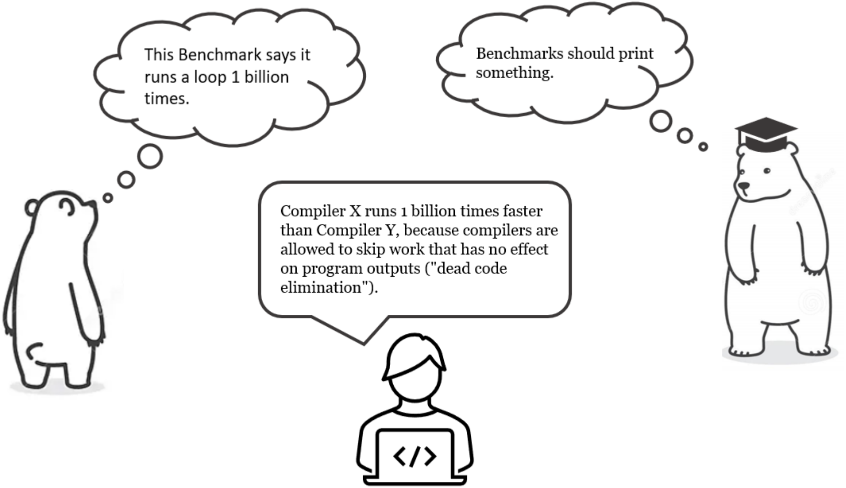
Various Benchmarks for High Performance Computing

- SPECaccel 2023
- SPEC ACCEL
- SPEChpc 2021
- SPEC MPI 2007
- SPEC OMP 2012

Other Benchmarks, also for non HPC Systems

- Java Client/Server
- Storage
- Power
- Cloud

SPEC - Common Benchmarking Mistakes



SPEC - Common Benchmarking Mistakes

If the benchmark description says:	**There may be potential difficulties:**	**Solutions**
The benchmark is already compiled. Just download and run.	You may want to compare new hardware, new operating systems, new compilers.	Source code benchmarks allow a broader range of systems to be tested.
The benchmark measures X.	Has this been checked? If not, measurements may be dominated by benchmark setup time, rather than the intended operations.	Analyze profile data prior to release, verify what it measures.

SPEC - Common Benchmarking Mistakes

If the benchmark description says:	**There may be potential difficulties:**	**Solutions**
The benchmark is a slightly modified version of Well Known Benchmark.	Is there an exact writeup of the modifications? Did the modifications break comparability?	Someone should check. Create a process to do so.
The benchmark is a collection of low-level operations representing X.	How do you know that it is representative?	Prefer benchmarks that are derived from real applications.

Full list: https://www.spec.org/hpc2021/docs/overview.html

What is a good Benchmark?

Table 1: Characteristics of useful performance benchmarks

Specifies a *workload*	A strictly-defined set of operations to be performed.
Produces at least one *metric*	A numeric representation of performance. Common metrics include: • Time - For example, seconds to complete the workload. • Throughput - Work completed per unit of time, for example, jobs per hour.
Is *reproducible*	If repeated, will report similar (*) metrics.
Is *portable*	Can be run on a variety of interesting systems.
Is *comparable*	If the metric is reported for multiple systems, the values are meaningful and useful.
Checks for correct operation	Verify that meaningful output is generated and that the work is actually done. *"I can make it run as fast as you like if you remove the constraint of getting correct answers."* (**)
Has *run rules*	A clear definition of required and forbidden hardware, software, optimization, tuning, and procedures.

(*) "Similar" performance will depend on context. The benchmark should include guidelines as to what variation one should expect if the benchmark is run multiple times.

(**) Author unknown. If you know who said it first, write .

SPEChpc 2021

- Designed to be used for heterogeneus Systems
- Contains a variety of tasks from different fields
- Is available in different sizes
- Is available with different extensions
  - pure MPI
  - MPI + OpenACC
  - MPI + OpenMP
  - MPI + OpenMP with target Offload

SPEChpc 2021 (cont.)

SPEChpc 2021 intentionally depends on all of the below - not just the processor.

- Processor - The CPU chip(s) and optionally, an acceleration device such as a GPU.
- Memory - The memory hierarchy, including caches and main memory.
- Interconnects - The communication between nodes of a cluster.
- Compilers - C, C++, and Fortran compilers, including optimizers.
- MPI - The MPI implementation.

Not intented to test graphics, Java libraries, or the I/O system

SPEChpc 2021 - Overview

Application Name	Benchmark				Language	Approximate LOC	Application Area
	Tiny	Small	Medium	Large			
LBM D2Q37	505.lbm_t	605.lbm_s	705.lbm_m	805.lbm_l	C	9000	Computational Fluid Dynamics
SOMA Offers Monte-Carlo Acceleration	513.soma_t	613.soma_s	Not included.		C	9500	Physics / Polymeric Systems
Tealeaf	518.tealeaf_t	618.tealeaf_s	718.tealeaf_m	818.tealeaf_l	C	5400	Physics / High Energy Physics
Cloverleaf	519.clvleaf_t	619.clvleaf_s	719.clvleaf_m	819.clvleaf_l	Fortran	12,500	Physics / High Energy Physics
Minisweep	521.miniswp_t	621.miniswp_s	Not included.		C	17,500	Nuclear Engineering - Radiation Transport
POT3D	528.pot3d_t	628.pot3d_s	728.pot3d_m	828.pot3d_l	Fortran	495,000 (Includes HDF5 library)	Solar Physics
SPH-EXA	532.sph_exa_t	632.sph_exa_s	Not included.		C++14	3400	Astrophysics and Cosmology
HPGMG-FV	534.hpgmgfv_t	634.hpgmgfv_s	734.hpgmgfv_m	834.hpgmgfv_l	C	16,700	Cosmology, Astrophysics, Combustion
miniWeather	535.weather_t	635.weather_s	735.weather_m	835.weather_l	Fortran	1100	Weather

Image: https://www.spec.org/hpc2021/docs/overview.html

Running SPEChpc Benchmark

- Free for non-commercial use
- Requirements:
  - Main Memory: 40GB (Tiny), 480GB(SMALL), 4TB(Medium), 14,5TB (Large)
  - 50 GB disk space
  - C, C++, and Fortran compilers
  - A MPI implementation configured for use with your compilers
  - ARM, Power ISA, or x86_64 CPU(s)

SPEChpc 2021 - Results

Metrics

- Single composite score (higher is better)
- Can be compared to other results from the same Suite

Typically:

- Time - For example, seconds to complete a workload.
- Throughput - Work completed per unit of time, for example, jobs per hour.

SPEChpc 2021 is a time-based, strong scaling metric.

SPEChpc Reference Machine

For each benchmark, a performance ratio is calculated as:

- Time on a reference machine / time on the SUT
- The reference machine ran 505.lbm_t (Fluid Dynamics) in 2250 seconds.
- A particular SUT (System under Test) took only 444 seconds
- The score is: 2250/444 = 5.067567

TU Dresden’s Taurus System was used as a reference System,

- It’s score is always 1

Base & Peak Measurement

Base Metric

- Compiled using the same flags, in the same order
- Use the same node-level parallel model
- Use the same number of ranks
- Use the same number of host threads per rank

All reported results must include the base metric.

Base & Peak Measurement

The Peak metric allows greater flexibility

- Different compiler options
- Different node-level parallel models may be used for each benchmark
- Number of ranks and threads set individually for each benchmark
- Limited source code modification to tune the directive models (OpenACC and OpenMP) for their system

Actual Results

SPEChpc™ 2021 Medium Result

NVIDIA Corporation Selene: NVIDIA DGX SuperPOD (AMD EPYC 7742 2.25 GHz, Tesla A100-SXM-80 GB)	SPEChpc 2021_med_base = 44.7 SPEChpc 2021_med_peak = Not Run

hpc2021 License:	019	**Test Date:**	Sep-2022
Test Sponsor:	NVIDIA Corporation	**Hardware Availability:**	Jul-2020
Tested by:	NVIDIA Corporation	**Software Availability:**	Mar-2022

Benchmark result graphs are available in the PDF report.

Results Table

Benchmark	Base									Peak								
	Model	Ranks	Thrds/Rnk	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio	Model	Ranks	Thrds/Rnk	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
705.lbm_m	ACC	1024	16	18.3	66.9	**18.2**	**67.2**	18.1	67.6									
718.tealeaf_m	ACC	1024	16	35.3	38.3	35.8	37.7	**35.5**	**38.0**									
719.clvleaf_m	ACC	1024	16	26.8	68.9	27.3	67.7	**27.0**	**68.4**									
728.pot3d_m	ACC	1024	16	**63.8**	**29.0**	63.6	29.1	65.2	28.4									
734.hpgmgfv_m	ACC	1024	16	**66.3**	**15.1**	66.6	15.0	66.3	15.1									
735.weather_m	ACC	1024	16	**23.0**	**104**	23.8	101	22.7	106									
SPEChpc 2021_med_base					44.7													
SPEChpc 2021_med_peak					Not Run													

Results appear in the order in which they were run. Bold underlined text indicates a median measurement.

Screenshot: https://www.spec.org/hpc2021/results/res2022q4/hpc2021-20221017-00137.html

Actual Results

Hardware Summary		Software Summary	
Type of System:	SMP	**Compiler:**	C/C++/Fortran: Version 22.3 of NVIDIA HPC SDK for Linux
Compute Node:	DGX A100	**MPI Library:**	OpenMPI Version 4.1.2rc4
Interconnects:	Multi-rail InfiniBand HDR fabric DDN EXAScalar file system	**Other MPI Info:**	HPC-X Software Toolkit Version 2.10
Compute Nodes Used:	64	**Other Software:**	None
Total Chips:	128	**Base Parallel Model:**	ACC
Total Cores:	8192	**Base Ranks Run:**	1024
Total Threads:	16384	**Base Threads Run:**	16
Total Memory:	128 TB	**Peak Parallel Models:**	Not Run

Screenshot: https://www.spec.org/hpc2021/results/res2022q4/hpc2021-20221017-00137.html

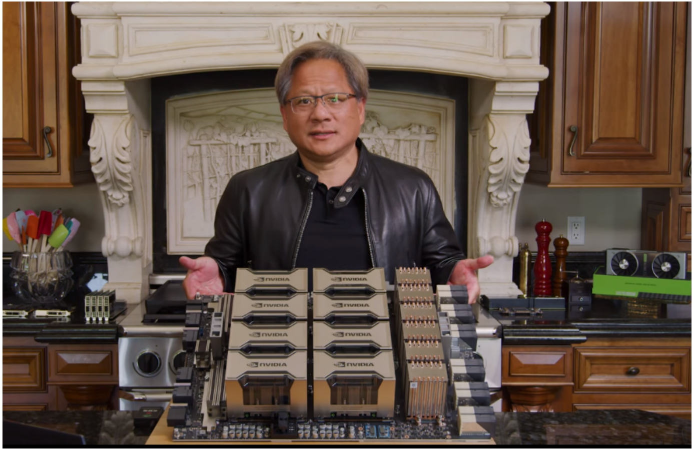
Actual Results

Node Description: DGX A100			
Hardware		**Software**	
Number of nodes:	64	**Accelerator Driver:**	NVIDIA UNIX x86_64 Kernel Module 470.103.01
Uses of the node:	compute	**Adapter:**	NVIDIA ConnectX-6 MT28908
Vendor:	NVIDIA Corporation	**Adapter Driver:**	InfiniBand: 5.4-3.4.0.0
Model:	NVIDIA DGX A100 System	**Adapter Firmware:**	InfiniBand: 20.32.1010
CPU Name:	AMD EPYC 7742	**Adapter:**	NVIDIA ConnectX-6 MT28908
CPU(s) orderable:	2 chips	**Adapter Driver:**	Ethernet: 5.4-3.4.0.0
Chips enabled:	2	**Adapter Firmware:**	Ethernet: 20.32.1010
Cores enabled:	128	**Operating System:**	Ubuntu 20.04 5.4.0-121-generic
Cores per chip:	64	**Local File System:**	ext4
Threads per core:	2	**Shared File System:**	Lustre
CPU Characteristics:	Turbo Boost up to 3400 MHz	**System State:**	Multi-user, run level 3
CPU MHz:	2250	**Other Software:**	None
Primary Cache:	32 KB I + 32 KB D on chip per core		
Secondary Cache:	512 KB I+D on chip per core		
L3 Cache:	256 MB I+D on chip per chip (16 MB shared / 4 cores)		

- Detailed description for Interconnection and Compiler is also available

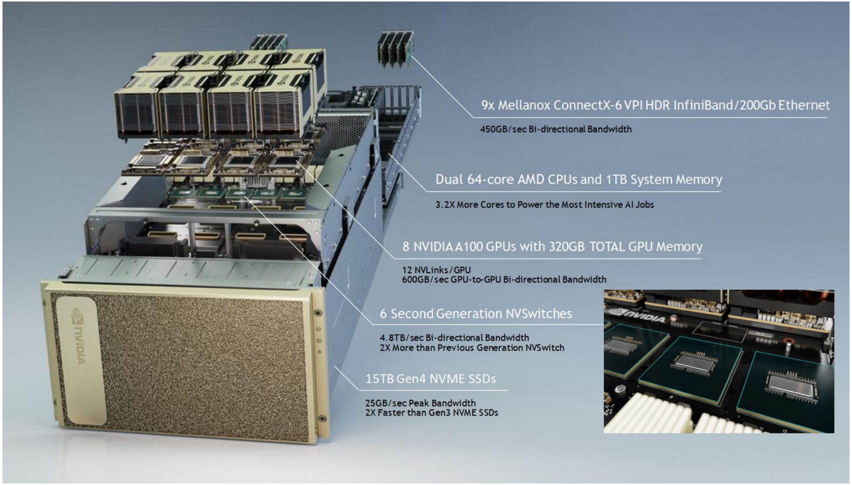
Screenshot: https://www.spec.org/hpc2021/results/res2022q4/hpc2021-20221017-00137.html

DGX A100



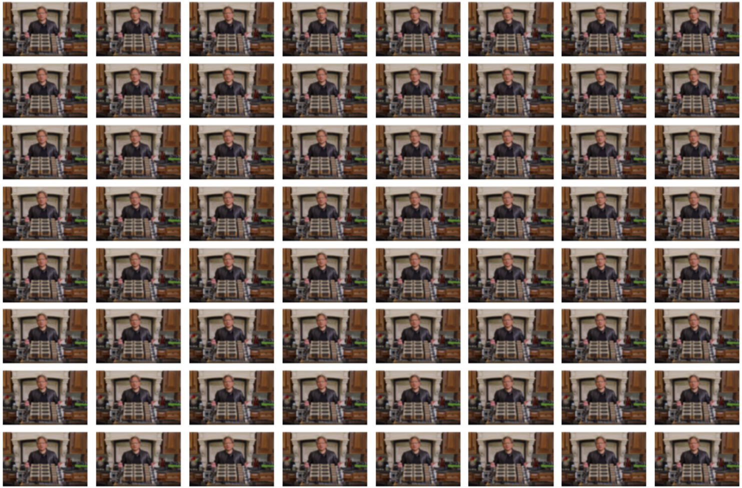
https://8f430952.rocketcdn.me/wp-content/uploads/2020/05/aim_nvidia.png

DGX A100



https://www.skyblue.de/de/erbe-und-exoten/gpu-pcie-karten-server/d-nvidia-dgx-a100

64 x DGX A100



Ranking

Test Sponsor	System Name	System Configuration				Results	
		Node-level Parallelization Model	Compute Nodes Used	MPI Ranks	Base Threads Per Rank	Base	Peak
NVIDIA Corporation	Selene: NVIDIA DGX SuperPOD (AMD EPYC 7742 2.25 GHz, Tesla A100-SXM-80 GB) HTML | CSV | Text | PDF | PS | Config	ACC	64	1024	16	44.7	Not Run
Oak Ridge National Laboratory	Summit: IBM Power System AC922 (IBM Power9, Tesla V100-SXM2-16GB) HTML | CSV | Text | PDF | PS | Config	ACC	700	4200	1	41.3	Not Run
RWTH Aachen University	CLAIX-2018: Intel Compute Module HNS2600BPM (Intel Xeon Platinum 8160) HTML | CSV | Text | PDF | PS | Config	MPI	100	4800	1	2.00	2.32
Technische Universitaet Dresden	Taurus: bullx DLC B720 (Intel Xeon E5-2680 v3) HTML | CSV | Text | PDF | PS | Config	MPI	85	2040	1	1.04	Not Run

Iamge: https://www.spec.org/hpc2021/results/hpc2021medium.html

SPEChpc 2021 - Case Studies

- Case study 1: RWTH Aachen *3*
  - Showed significantly lower performance than similar HPC Systems
  - Performance Data showed that exeution times differ in MPI time
  - Especially MPI_Allreduce
  - Faulty Memory DIMMS
- Case study 2: TU Dresden *3*
  - Similar situation
  - Faulty BIOS configuartion on serveral nodes
  - Kernel bug
  - Unfavorable SLURM configuration

Limitations of SPEChpc 2021

- The ideal benchmark for vendor or product selection is your own workload on your own application.
- No standardized benchmark can perfectly model the realities of your particular system and user community.
- Consider the uniqueness of your workload and application when assessing benchmark results.

Efficient Computing

Efficiency vs. Performance at any Cost

- Rising electricity price
- Increased environmental awareness
- Green500
- SPECpower

Green500

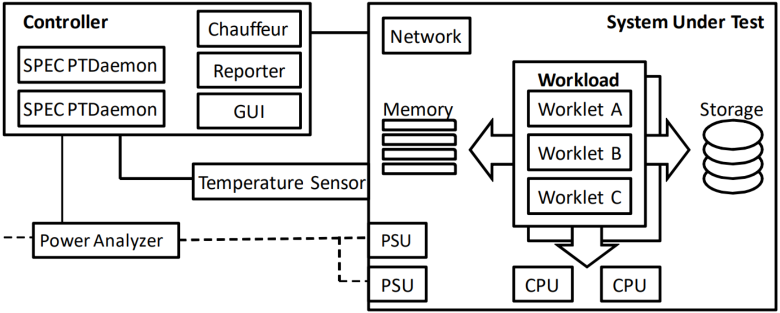
Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
1	293	**Henri** - ThinkSystem SR670 V2, Intel Xeon Platinum 8362 32C 2.8GHz, NVIDIA H100 80GB PCIe, Infiniband HDR, Lenovo Flatiron Institute United States	8,288	2.88	44	65.396
2	44	**Frontier TDS** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	120,832	19.20	309	62.684

https://www.top500.org/lists/green500/2023/11/

SPECpower Committee

- SERT 2 Suite
  - Approximates server efficiency across diverse applications
  - User-friendly with GUI and predetermined tuning parameters.
  - Tested on various 64-bit processors, operating systems, and JVMs
  - Scalable, tested up to 8 processor sockets and 64 nodes.
  - Applicable to standalone servers and multi-node sets with shared infrastructure
  - Written in Java for cross-platform support; accommodates other languages
  - Generates machine- and human-readable results for certification and customer reports

SERT Setup



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

SERT Components

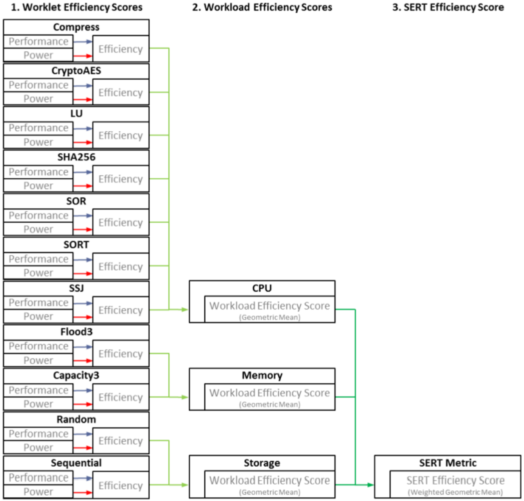


Figure 1: SERT 2 metric calculation

SERT Components

All SERT worklets, except Idle run at multiple load levels. For each of those load levels, energy efficiency is calculated separately. We define per load level energy efficiency Effload as follows:

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

Outlook

- SPECIaaS
- Benchmarks for AI Applications
- SPEChpc 2021 Benchmarks on Ice Lake and Sapphire Rapids Infiniband Clusters: A Performance and Energy Case Study 2
- Trends in efficient computing
  - How do systems evolve?

References

1. https://www.spec.org/consortium/.
2. https://dl.acm.org/doi/pdf/10.1145/3624062.3624197.
3. https://ieeexplore.ieee.org/document/9826013.
4. https://www.nas.nasa.gov/software/npb.html.
5. https://hpcchallenge.org/hpcc/.
6. https://www.top500.org/project/linpack/.