HPS

aasish-kumar.sharma@gwdg.de

Aasish Kumar Sharma

HPC Benchmarking

High-Performance Computing System Administration (HPCSA)

Table of contents

1 Theory

2 Exercise

Objectives

To understand,

- What benchmarking is, basically, the theoretical knowledge behind it.
- Why benchmarking is done, i.e., necessity of benchmarks.
- How benchmarking is done, i.e., ways of doing benchmarking.

https://languages.oup.com/google-dictionary-en

What is Benchmarking?

Theoretical introduction of benchmarking.

Benchmark (noun)

- A standard or point of reference against which things can be compared.

Benchmarking (verb)

- Process of comparing with a previously defined standards (benchmarks).

https://languages.oup.com/google-dictionary-en

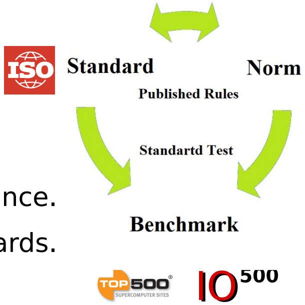
What is Benchmarking? (Contd..)

Standard

- It is a published specification ratified by some organization.
- e.g., ISO 9001 - is an international standard for quality management.

Benchmark

- It is a test to evaluate your system's performance,
- It is often established by general organizational acceptance.
- e.g., IO 500 Benchmark - is a comparison against standards.



Images: https://www.iso.org/modules/isoorg-template/img/iso/iso-logo-print.gif, https://www.vi4io.org/io500/start, https://www.top500.org/news/chinas-tianhe-2-supercomputer-retains-top-spot-on-43rd-edition-of-the-top500-list/

https://de.wikipedia.org/wiki/Standard

Why Benchmarking is Done?

HPC Benchmarking

- Benchmark measures system behavior, so
- Whenever there is question about performance, answer is benchmarking.

Moreover

- Benchmarking measures the relative performance either by,
  - Changing the in/out parameters, or
  - On scaling the system.
- Measured by running a number of standard tests and programs. For e.g.,
  - Running a computer program (micro benchmarking),
  - A collection of programs (macro benchmarking),
  - Other operations (overall benchmarking).

.

Fleming and Wallace, “How Not to Lie with Statistics”

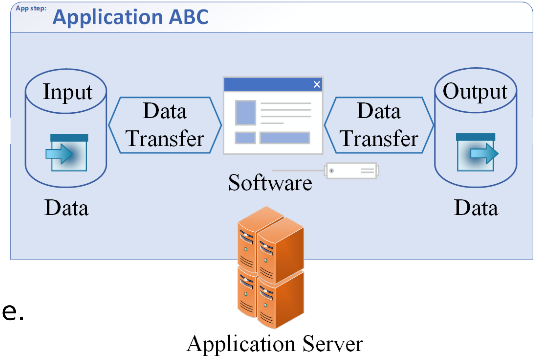
What is an application and how it is benchmarked?

Application means: (System & Workload)

- The configured system,
- Software running on it, and
- Input/Output data required by it.

So to benchmark need

- To calibrate the whole system.
- To allocate proper resources.
- To make sure there is ideal wait time.



How to Benchmark: What are the Benchmarking Key Metrics

Key Metrics

- Micro Benchmarking - Baseline performance
  - Measures benchmarking performance improvement against unit of node.
- Macro Benchmarking - Scaling
  - Measures how the performance changes with the number of nodes/cores.
- Overlall Benchmarking - Performance
  - It is a measure of rate of how well an application is running.
- Other Measures
  - Timing
    - It is a measure of full or partial run-time of an application. Mainly wall clock.
  - Parallel efficiency
    - The ratio of measured scaling to the perfect scaling.

Let us Discussion on Macro Benchmarking - Scaling

- Scalability in case of Speedup for,
  - Hardware: is the ability to handle more workload by scaling compute power.
  - Software: is the parallelization efficiency, given by
    - The ratio of the actual speedup and the ideal speedup for a number of processors.

$$Speedup = t(1)/t(N) \longrightarrow t : time; N : no..of..processors. \tag{1}$$

Discussion on Macro Benchmarking - Scaling (Contd..)

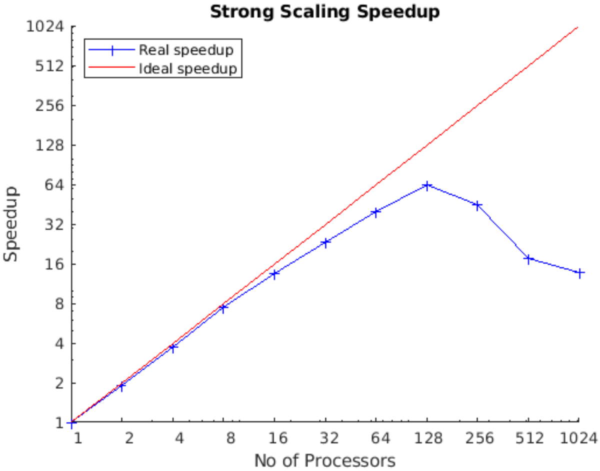
- Applications' speedup scaling test is done in two ways:
  - Strong Scaling
    - Number of processors is increased while the problem size remains constant.
    - e.g., Amdahl's law (1967),

$$Speedup = 1/(s + p/N) \longrightarrow s : serial; p : parallel \tag{2}$$

  - Week Scaling.
    - Both the number of processors and the problem size are increased.
    - e.g., Gustafson's law (1988),

$$Speedup = s + p * N \longrightarrow s : serial; p : parallel \tag{3}$$

An Example of Strong Scaling



Scaling - HPC Wiki

Guidelines on Scaling Benchmark

1. Measure using job sizes that span:
2. Use wall-clock time units or equivalent.
3. Measure multiple independent runs per job size.
4. Various factors must be considered when using more than one node:
   a) Interconnect speed and latency
   b) Max memory per node
   c) processors per node
   d) max processors (nodes)
   e) system variables and restrictions (e.g. stack size)
5. Also, if possible measure using different systems and factors.
6. Use a problem state that best matches the intended production runs.

In Summary

Overall Guidelines:

- Be alert and vigilante to the details,
- Think critically and include all the details,
- Use proper measures and charts to present,
- Adapt and address the changes properly,
- Attempt repetitively and continuously.

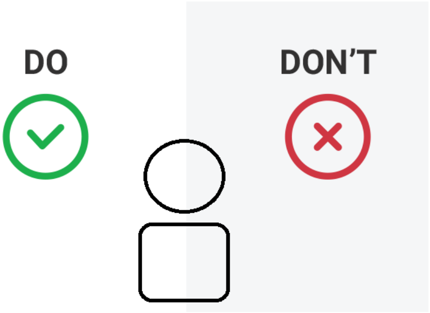


Image: https://thefruitfultoolbox.com/dos-donts-disc/

References

Fleming, Philip J. and John J. Wallace. "How Not to Lie with Statistics: The Correct Way to Summarize Benchmark Results". In: *Commun. ACM* 29.3 (Mar. 1986), pp. 218–221. ISSN: 0001-0782. DOI: 10.1145/5666.5673.

Scaling - HPC Wiki. URL: https://hpc-wiki.info/hpc/Scaling (visited on 04/18/2023).

Outline

1 Theory

2 Exercise