

# Metrics for Performance Advertisement

## Performance Evaluation of Computer Systems

Vojtěch Horký   Peter Libič   Petr Tůma

Department of Distributed and Dependable Systems  
Faculty of Mathematics and Physics  
Charles University

2010 – 2021

# Outline

- 1 Overview
- 2 Operation Frequency Related Metrics
- 3 Operation Duration Related Metrics
- 4 Benchmark Workloads

# Performance Advertisement

Measuring for the purpose of publishing performance information.

Requirements:

- Well defined meaning.
- Simple to understand.
- Difficult to game.

Pitfalls:

- Publication makes results subject to pressure.
- Often too simple to convey meaningful information.
- Performance in contemporary computer systems is never simple.

# Speed Related Metrics

## Responsiveness Metrics

- Time (*Task Response Time*)
- How long does it take to finish the task ?

## Productivity Metrics

- Rate (*Task Throughput*)
- How many tasks can the system complete per time unit ?

## Utilization Metrics

- Resource Use (*Utilization*)
- How much is the system loaded when working on a task ?
- Share of time a resource is busy or over given load level.
- Helps identify bottlenecks (most utilized resources in the system).

# Outline

- 1 Overview
- 2 Operation Frequency Related Metrics**
- 3 Operation Duration Related Metrics
- 4 Benchmark Workloads

## Clock Rate

Clock rate (frequency) of the component (CPU, bus, memory) in MHz.  
Most often we talk about CPU frequency.

### Not Reliable

CPU with higher frequency does not run all applications faster.

- Ignores IPC.
- Ignores how much of the work done is actually used (speculative execution, pipelining ...).
- Ignores that CPU might not be a bottleneck of an application.

### Not Repeatable

Clock rate is not constant on many platforms.

- Dynamic frequency scaling.
  - ▶ CPU can run on lower frequency to save energy and heat.
  - ▶ CPU can boost frequency to give more performance online.
- This can sometimes be monitored or adjusted.

# MIPS

Millions of instructions executed per second.

Defined for a given instruction mix.

**Gibson Mix (IBM)** for scientific applications

34% int math, 13% float math, etc.

**Whetstone Mix** for floating point computations

**Dhrystone Mix** for system programming

## Not Linear, Not Reliable, Not Consistent

- Results depend on the code executed and cannot be generalised.
- With the same code, instructions on different platforms do different amount of work:
  - RISC** simple instructions, more needed
  - CISC** complex instructions, fewer needed

# MFLOPS

Millions of floating point operations executed per second.  
Assumes certain similarity for basic floating point operations.

## Not Reliable

Makes only sense when floating point operations are the major factor of performance (scientific computing).

## Not Independent

Different platforms support different operations:

- Division sometimes directly supported, sometimes implemented using other operations (Cray, Itanium).
- Sin, Cos, Log sometimes single operation, sometimes look up and approximations (Taylor).
- Are these single or multiple operations ?
- Interpretation prone to marketing games.



# Outline

- 1 Overview
- 2 Operation Frequency Related Metrics
- 3 Operation Duration Related Metrics**
- 4 Benchmark Workloads

# Wall Clock Time I

```
start = get_real_time ();  
// run the operation of interest  
end = get_real_time ();  
return (end - start);
```

Operation time that would have been measured by a person with a stop watch.

## Pros

- Very intuitive metric in units everyone understands.
- Reliable – for representative benchmarks.
- Consistent – seconds are the same with all systems.
- Independent – if the benchmarks are not optimized against.

# Wall Clock Time II

## Cons

- Only applies to a particular operation (usually generalized using benchmarks).
- Typically sensitive to background load:
  - ▶ Non random load (scheduled tasks) can bias the results.
  - ▶ Random load is not easily reproducible.
  - ▶ Realistic background load might make sense, but must be made part of controlled experiment.

Also think about exact operation boundaries:

- User oriented metrics would prefer end-to-end times:
  - ▶ From click to end of page rendering.
  - ▶ From application launch to result display.
- Developer oriented metrics would prefer measuring within single domain:
  - ▶ Separate communication time, queueing time, processing time.
  - ▶ Separate data load and save time from computation time.

## Processor Time I

```
start = get_thread_consumed_time ();  
// run the operation of interest  
end = get_thread_consumed_time ();  
return (end - start);
```

Aggregate work time that would have been reported by workers working in parallel.

### Pros

- Counts only actually consumed time.
- Can distinguish kernel time and user time.

## Processor Time II

### Cons

- Possibly low precision (depends on accounting mechanism).
- Does not include necessary waiting (I/O, synchronization).
- Still may be affected by background load (caches, TLB, memory).

A possible compromise is to collect both processor and wall clock time.  
Think about processor to wall clock time ratio:

- High ratio indicates high parallelism.
- Low ratio indicates blocking.

# Outline

- 1 Overview
- 2 Operation Frequency Related Metrics
- 3 Operation Duration Related Metrics
- 4 Benchmark Workloads**

# Standard Benchmarks

Report performance of a well known (standardized) benchmark.  
The question is who should standardize such benchmarks.

## Industrial Standards

Benchmarks developed through cooperation between multiple vendors.  
Focus on transparent process and fair comparison.  
For example SPEC or TPC benchmarks.

## Research Standards

Benchmarks developed for evaluating research results.  
Focus on insight into particular research topic.  
For example DaCapo or NPB benchmarks.

## Popular Standards

Benchmarks developed often for fun but with popular acceptance.

# Standard Performance Evaluation Corporation

A non profit consortium developing industry standard benchmarks.

Provides a set of benchmark suites for different systems and workloads:

- CPU – SPEC CPU2017 ...
- Power – SPECpower ssj2008, SERT ...
- Graphics – SPECviewperf 13, SPECwpc, SPECapc ...
- Computational – SPEC ACCEL, SPEC MPI2007, SPEC OMP2012 ...
- Java – SPECjvm2008, SPECjEnterprise2018, SPECjbb2015 ...
- Cloud – SPECvirt sc2013, SPEC Cloud IaaS 2018 ...

More information on <http://www.spec.org>.



# SPEC CPU Benchmarks

Reporting combined performance of multiple benchmarks.

## Characteristics

- Set of (about 40) diverse benchmark tasks (compilation, compression, rendering ...)
- Run each benchmark program, measure execution time.
- Provide geometric mean of normalised benchmark execution times.

Benchmark metric comments:

- Geometric mean perhaps a sensitivity compromise.
- Not linear with program execution time.
- Not always reliable.
- Not very intuitive.
- Weights unclear.

# SPEC CPU Benchmarks

## Reliability ?

- Good for individual benchmarks, but these not always of interest.
- For general applications, low level benchmarks (SPECint, SPECfp) less reliable than application benchmarks (SPECjbb, SPECjvm).

## Independence ?

- Vendors are known to optimise for SPEC benchmarks.
- Partial solution is use of base and peak profiles.
  - ▶ Base compiles all benchmarks with the same flags.
  - ▶ Peak permits benchmark specific flags and feedback directed optimization.
- However, developers should not include optimizations that are unlikely to improve real applications.

# SPEC JBB Benchmarks

Reporting transaction rate of a model application.

## Characteristics

- A model application of a supermarket chain backend.
  - ▶ Customers buy in markets.
  - ▶ Markets order from suppliers.
  - ▶ Headquarters perform data mining.
- Multiple deployment models (local and distributed).
- Gradually increase workload and look at transaction processing.
  - ▶ Report critical-jOPS as throughput under response time constraints.
  - ▶ Report max-jOPS as peak throughput with correctness constraints.

Benchmark metric comments:

- Easily related to practical performance.
- Response time constraints sensitive to disruptions.
- Low resolution due to coarse workload steps in implementation.

# Transaction Processing Performance Council

A non profit consortium developing standard data processing benchmarks.

Provides a set of benchmark suites for different systems and workloads:

- TPC-C – order entry system
- TPC-DI – data transformation (ETL)
- TPC-DS – decision support system for retail supplier
- TPC-E – financial brokerage system
- TPC-H – decision support system for product distribution industry
- TPC-VMS – extending benchmarks to virtualized environments
- TPCx-BB – big data benchmark using Hadoop queries
- TPCx-HS – big data benchmark using Hadoop filesystem
- ...

More information on <http://www.tpc.org>.