UNIVERSIDAD DE BURGOS

Quantifying performance

Benchmarks

V 1.0

José M. Cámara
(checam@ubu.es)

UNIVERSIDAD
DE BURGOS

Motivation

- ❑ Performance must be measured to:
  - ➢ Assess the behavior of a computing system.
  - ➢ Compare various systems.
  - ➢ Optimize utilization.
  - ➢ Remove bottlenecks.
- ❑ Quantifying performance is a daunting task:
  - ➢ Systems are distinct from one another.
  - ➢ They are very complex.
  - ➢ Handle a broad variety of applications and data.

UNIVERSIDAD DE BURGOS

Metrics

- ❑ Latency: time to complete an action: deliver a message, execute a program, fulfill a request… According to the scenario it can be expressed in terms of: execution time or response time too.
- ❑ Throughput: tasks completed per time unit: instructions, messages, queries…
- ❑ Throughput = 1/ latency only when no overlap is produced (instruction or message pipe-line). Otherwise throughput > 1/ latency.

UNIVERSIDAD DE BURGOS

Benchmarks

- Concept: application program used to quantify computer's performance.
- Goal: results must be numeric, objective and fair.
- Types:
  - Real programs.
  - Synthetic.
  - Kernels.
  - Toys.
  - Suites.

UNIVERSIDAD
DE BURGOS

Benchmarks

- Real programs may seem the most objective option but in many occasions their results are hard to interpret since too many computer systems are affected by them in uncertain and variable ways.
- Synthetic benchmarks are designed to reflect the performance of certain subsystems.
- Therefore, in order to evaluate all subsystems, usually a suite is preferred.
- Kernels are close to real programs. They eliminate all that is not relevant, such as user interface, calculation results, etc.
- Toy benchmarks are short programs that produce results already known to the user.

Evaluating results

- Benchmark suites are a common tool but since they are composed by a number of programs. Systems may perform differently under each one.
- Direct comparison is not possible then. A more elaborated metric must be obtained:
  - Arithmetic mean:$AM = \frac{\sum_i^N r_i}{N}$. Not accurate when the tests are unrelated. The longest test tends to prevail.
  - Geometric mean:$GM = \sqrt[N]{\prod_i^N r_i}$. Its utility is unclear.
  - Harmonic mean:$HM = \frac{N}{\sum_i^N \frac{1}{r_i}}$
- Results r_i may be absolute execution times (or their inverses) or speed-ups (referred to a precise system).

UNIVERSIDAD DE BURGOS

Comparing means

- Let's think of a set of tests where the execution times are: 1s, 4s and 10s.
- The arithmetic mean of them is 5s. This is correct but, taking into account that the shortest programs may be executed more times than the longest, maybe they should weigh more on the average.
- The harmonic mean is 3/1,35 = 2,22s. That could reflect more accurately the expected performance of the system.

UNIVERSIDAD
DE BURGOS

Selecting benchmarks

- ❑ Distinct systems require distinct benchmarks.
- ❑ We are considering two types of servers: supercomputers & data centers.
- ❑ For supercomputers, regardless the subsystem we intend to stress, what matters is execution time (throughput is also relevant in many cases). Results are usually given in terms of FLOPS.
- ❑ For data centers the response time is what the user perceives as “performance”. Throughput doesn’t make much sense in this environment. Results may be given in terms of SLO/SLA ratio. For example: 99% of responses below 100ms.

UNIVERSIDAD DE BURGOS

Benchmarking supercomputers

- By far, the most popular benchmark for supercomputers is Linpack. Provides performance in FLOPS under the execution of a kernel based on the resolution of systems of linear equations. It has a number of variations to adapt itself to many computer architectures. Is the test bed for the Top500 supercomputer rank.
- NAS NPB: suite of kernels developed by NASA. Kernels try to reflect the core of calculations commonly performed by fluid mechanics applications and other usual programs related to its activity. All benchmarks impose heavy calculation on the system; differ on the problem they solve and the amount of communication traffic they generate:
  - **EP**: ***E**mbarrassingly **P**arallel.* It involves calculation but very little communication between processors.
  - **MG**: ***M**ulti**g**rid.* Unlike the former one, it requires communication between both close and remote processors.
  - **CG**: ***C**onjugate **G**radient.* Communication is low and scattered among close and remote nodes.
  - **FT**: ***F**ourier **T**ransform.* Heavy communication pattern evenly distributed.
  - **IS**: ***I**nteger **S**ort.* Communication, although heavy and uniform is not as heavy and uniform as in the previous case.
  - **LU**, **SP & BT**: address the same mathematical problem using different algorithms. BT is “less parallel” than the rest.

UNIVERSIDAD
DE BURGOS

Linpack

- It first release (Linpack 100) dates back to 1977.
  - It used n=100 size matrices.
  - It didn’t allow for code manipulation.
  - Only compiler level optimizations were allowed.
  - It used level 1 BLAS (Basic Linear Algebra Subprograms).
- The next release (Linpack 1000) date back to 1986.
  - Matrices were n=1000 in size.
  - Manual optimizations of code were permitted.
  - Using level 3 BLAS.
- Top500 rank is based on HPL (high performance Linpack) released in 1991.
  - Matrices can be any size.
  - Manual optimizations are allowed.
  - It is a message passing parallel implementation.
  - For mathematical operation it uses BLAS o VISPL.
  - The use of derived versions of there libraries, allows HPL to extend to heterogeneous systems. E.g. cuBLAS for CPU + GPU systems.
  - MPI is used for message passing.

UNIVERSIDAD
DE BURGOS

Benchmarking data centers

- Benchmarking data centers is not simple. There are two possible approaches:
  - From the users point of view, response time to queries, searches, etc, is what matters.
  - To provide low response times, managers need to test many subsystems.
- Benchmarking data centers is a recent challenge. Nevertheless, there is already certain agreement on what good benchmarks should comprise.
  - Include diverse workloads.
  - Workloads and software stacks must be representative.
  - Involve state of the art techniques.
  - Must be “usable”, that is, easy to deploy, configure and run.

UNIVERSIDAD
DE BURGOS

Benchmarking data centers

- Today's top Internet sites are (http://www.alexa.com/topsites/global;0):
  - Search engines: 40%. Handle mainly text data on queries.
  - Social networks: 25%. Handle graph data.
  - E-comerce: 15%. Table data.
- Common algorithms used by theses sites are:
  - Sort.
  - Scan.
  - Classification.
  - Graph mining.
  - Segmentation.

UNIVERSIDAD DE BURGOS

Example

- A 6 processor, 24 cores cluster takes 91,07 seconds to complete a matrix multiply program on 5000x5000 size matrices.
  - Determine performance in FLOPS.
  - Find out the position of this cluster in the Top500 historical database.

References

- ❑ Advanced Computer Architecture: Introduction to quantitative analysis. Prof. Sherief Reda. School of Engineering. Brown University.
- ❑ BigDataBench: a Big Data Benchmark Suite from Internet Services. Lei Wang et al. The 20th IEEE International Symposium On High Performance Computer Architecture (HPCA-2014), February 15-19, 2014, Orlando, Florida, USA.
- ❑ DCBench: a Benchmark Suite for Data Center Workloads. Zhen Jia. HVC tutorial in conjunction with The 19th IEEE International Symposium on High Performance Computer Architecture (HPCA 2013).
- ❑ Notes on Calculating Computer Performance. Bruce Jacob and Trevor Mudge. Advanced Computer Architecture Lab. EECS Department, University of Michigan.
- ❑ The LINPACK Benchmark: past, present and future. Jack J. Dongarra, Piotr Luszczek and Antoine Petitet. CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE. *Concurrency Computat.: Pract. Exper.* 2003; **15**:803–820 (DOI: 10.1002/cpe.728).
- ❑ The NAS parallel benchmarks. D.H. Bailey et al. The International Journal of Supercomputer Applications (1991).