Performance

- Tony McGregor
- G.1.05
- tonym@cs.waikato.ac.nz

Introduction

- How does one measure, report and summarise performance?
- Complexity of modern systems make it more difficult to access performance
- See through marketing hype
- Need to understand performance at multiple levels
  - Understand why a program performs poorly on a particular processor

c.f. Commercial Airplanes

- Which of the following planes has the best performance?

Airplane	Passengers	Range (mi)	Speed (mph)	Throughput
Boeing 737-100	101	630	598	60,398
Boeing 747	470	4150	610	286,700
BAC/Sud Concorde	132	4000	1350	178,200
Douglas DC-8-50	146	8720	544	79,424

- Can also define computer performance in several ways

Performance

- Purchasing perspective
  - given a collection of machines, which has the
    - least cost ?
    - best performance ?
    - best performance / cost ?
- Design perspective
  - faced with design options, which has the
    - least cost ?
    - best performance improvement ?
    - best performance / cost ?
- Both require
  - basis for comparison
  - metric for evaluation
- Our goal is to understand cost & performance implications of architectural choices

Two notions of "performance"

Plane	DC to Paris	Speed	Passengers	Throughput (pmph)
Boeing 747	6.5 hours	610 mph	470	286,700
BAD/Sud Concorde	3 hours	1350 mph	132	178,200

Which has higher performance?

- **Time to do the task (Execution Time)**
  - execution time, response time, latency
- **Tasks per day, hour, week, sec, ns. .. (Performance)**
  - throughput, bandwidth

Response time and throughput are sometimes in opposition

Definitions

- Performance is in units of things-per-second
  - bigger is better
- If we are primarily concerned with response time
  - performance(x) = $\frac{1}{\text{execution\_time}(x)}$

" X is n times as fast than Y" means

$$n = \frac{\text{Performance}(X)}{\text{Performance}(Y)}$$

Example

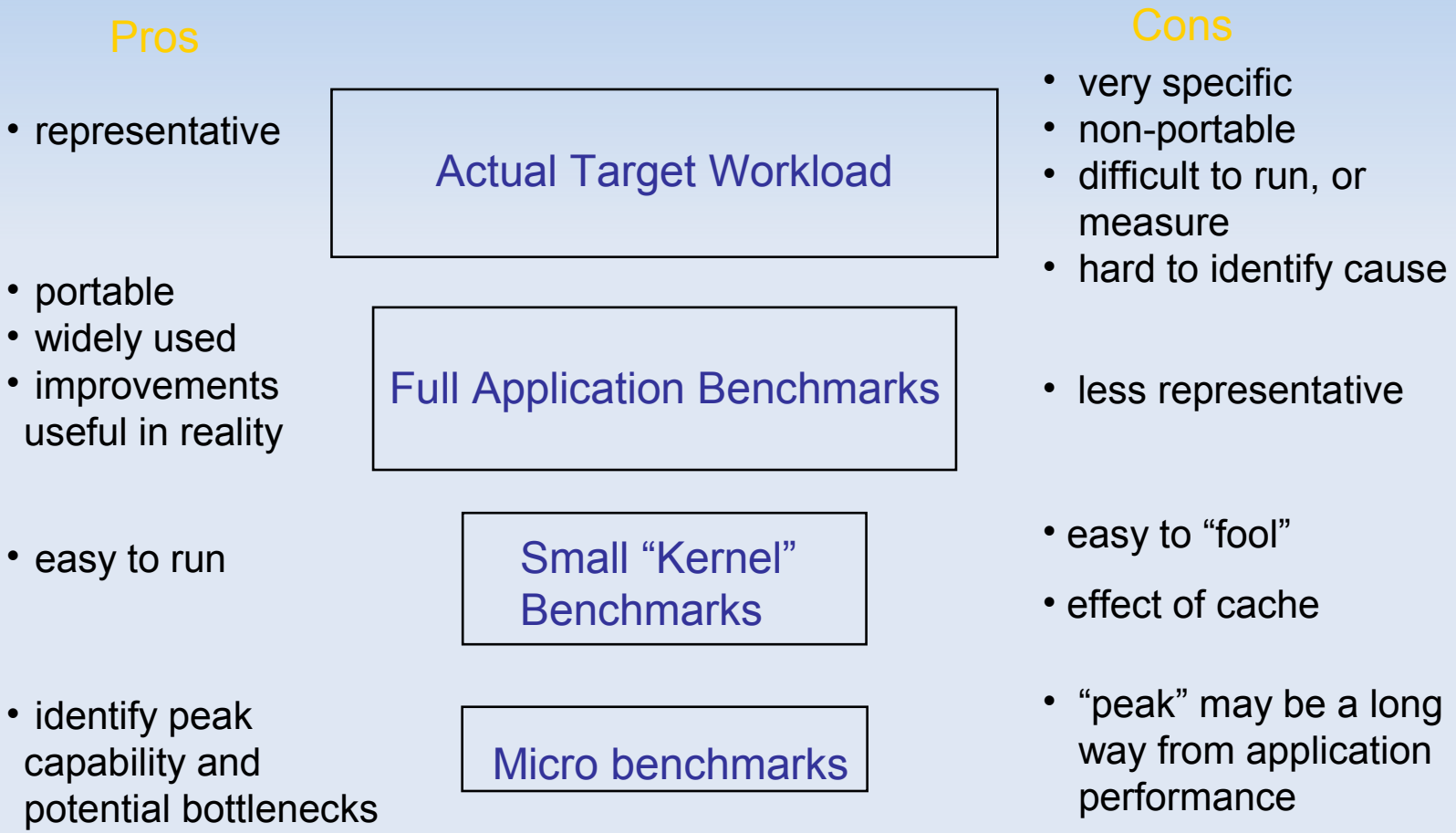
- Time of Concorde vs. Boeing 747?
  - Concord is 1350 mph / 610 mph = 2.2 times as fast
    = 6.5 hours / 3 hours
- Throughput of Concorde vs. Boeing 747 ?
  - Concord is 178,200 pmph / 286,700 pmph = 0.63 times as fast
  - Boeing is 286,700 pmph / 178,200 pmph = 1.6 times as fast
- Boeing is 1.6 times (“63%”) faster in terms of throughput
- Concord is 2.2 times (“120%”) faster in terms of flying time

We will focus primarily on execution time for a single job

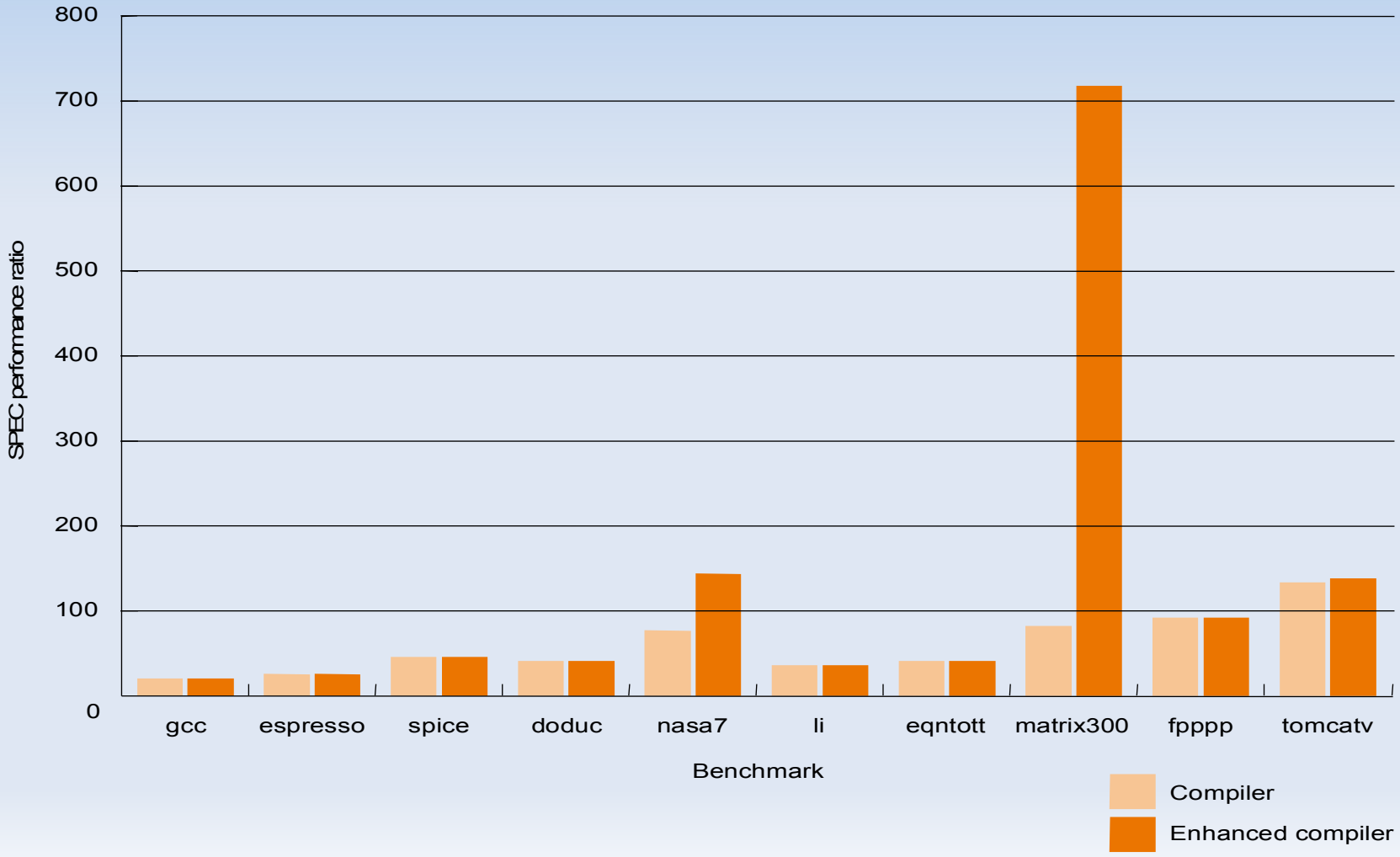
CPU Statistics

- Clock speed
  - CPI
  - multi-core
- MIPS
  - instruction complexity
  - word size
  - pipeline effects
- FLOPS
  - multiple FPUs
  - other types of workload

Basis of Evaluation



Benchmark Problems



SPEC95

- Eighteen application benchmarks (with inputs) reflecting a technical computing workload
- Eight integer
  - go, m88ksim, gcc, compress, li, ijpeg, perl, vortex
- Ten floating-point intensive
  - tomcatv, swim, su2cor, hydro2d, mgrid, applu, turb3d, apsi, fppp, wave5
- Must run with standard compiler flags
  - eliminate special undocumented incantations that may not even generate working code for real programs

Measuring Performance

- Execution Time can be measured in several ways:
  - Elapsed Time
    - counts everything *(disk and memory accesses, I/O , etc.)*
    - a useful number, but often not good for comparison purposes
  - CPU time
    - doesn't count I/O or time spent running other programs
    - can be broken up into system time, and user time
- Our focus: user CPU time
  - time spent executing the lines of code that are "in" our program

Measuring Performance

- On Linux can use **time** command
- E.g.

```
> time tar cvzf pam2001.tgz pam2001
real    0m19.348s
user    0m0.930s
sys     0m0.660s

> time tar cvzf pam2001.tgz pam2001
real    0m1.482s
user    0m0.950s
sys     0m0.230s
```

Summary

- Bottom line performance measure is time
  - $\text{Performance}_A = 1/\text{Execution Time}_A$
- Comparing Performance
  - N = Performance_A / Performance_B

Example

- If a machine A runs a program in 25 seconds and machine B runs the same program in 20 seconds, how much faster is machine B that machine A?

Relating Metrics

- Instead of reporting execution time in seconds, we often use cycles

- So, to improve performance (everything else being equal) you can either

  _____ the number of cycles required for a program, or
  _____ the clock cycle time or, said another way,
  _____ the clock rate.

Example

- Our favourite program runs in 10 seconds on computer A, which has a 1200 Mhz clock. We want to buy a new machine B, that will run this program in 6 seconds. A new machine we are considering has newer (or perhaps more expensive) technology that substantially increases the clock rate, the changes needed to support this clock rate mean that the new machine uses 1.2 times as many clock cycles as machine A for the same program. What clock rate do we need to get our speedup?"

Amdahl's Law

Execution Time After Improvement =

Execution Time Unaffected + (Execution Time Affected / Amount of Improvement)

- Example:

  "Suppose a program runs in 100 seconds on a machine, with multiply responsible for 80 seconds of this time. How much do we have to improve the speed of multiplication if we want the program to run 4 times faster?"

  How about making it 5 times faster?

- *Principle: Make the common case fast*

Amdahl's Law

- More conventianally, Amdahl's law relates to parallel computers:

  maxspeedup = $1/((1-P)+P/N)$

  where:

  - P = proportion of problem parallelised
  - N = number of processors

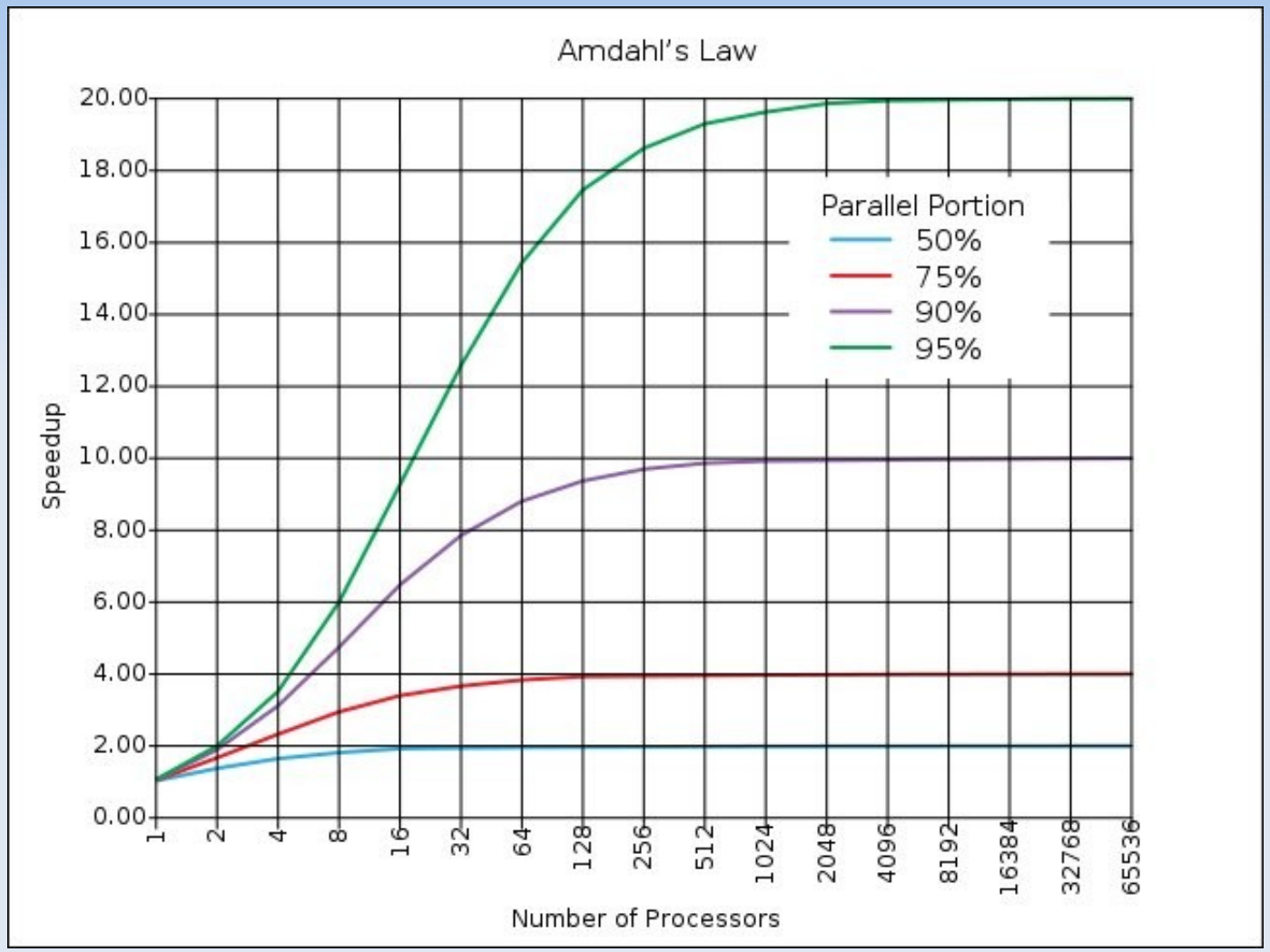
E.G.

- 90% paralleised and 10 processors

  max speedup = 1((1-.9)+.9/10) = 0.19

- 90% paralleised and 1000 processors = 0.1009

Amdahl's Law



Amdahl's Law

- More generally for a set of problem components:
- Speedup = P1/S1 + P2/S2 ... Pn/Sn

  where:

  - P1 = porprtion of problem of first component
  - S1 = speedup of first component
  - P1 + P2 +... + Pn = 1

Amdahl's Law -examples

- Suppose we enhance are compairing two machines with the same specs except that one claims to have floating-point instructions that run five times faster. If the execution time of a benchmark on the slower machine takes 10 seconds with half of that time taken by floating point instructions how long should the benchmark take on the faster machine?

- The supplier of the fater machine above is looking for a benchmark to show off the new floating-point unit and wants the overall benchmark to show a speedup of 3 times. What proportion of a benchmark must be floating-point instructions to show this speedup?

Remember

- Performance is specific to a workload
  - Total execution time is a consistent summary of performance
- For a given architecture performance increases come from:
  - increases in clock rate (without adverse CPI affects)
  - improvements in processor organization that lower CPI
  - compiler enhancements that lower CPI and/or instruction count
  - Increases in the bus width
- Pitfall: expecting improvement in one aspect of a machine's performance to affect the total performance

Evaluating Performance

- Real workloads good but often not possible
- Mostly use benchmarks
  - Real applications best
  - SPEC benchmarks – CPU92, CPU95, CPU2000
  - SPECweb99
- Guiding principle to reporting performance is "*reproducibility*"
- i.e. detailing:
  - CPU details
  - Software (e.g. compiler version etc)

An Asside

- On linux processor information is available in the pseudo file: /proc/cpuifo