

## Lecture 4: Performance

COS 471a, COS 471b / ELE 375

Computer Architecture and Organization

Princeton University  
Fall 2004

Prof. David August

2

### Notes

- Read Chapters 1-4
- Read Appendix B
- For those that felt uncomfortable with quiz 0 or achieved less than an 8.5, answer any 4 questions from Appendix B.
- TA will examine all those turned in on Tuesday (1 week).

3

### Which Aircraft Is Best?

Aircraft	Passengers	Range (Miles)	Speed (mph)
Boeing 737-100	101	1540	598
Boeing 747	470	4150	610
Concorde	132	4000	1350
Douglas DC-8-50	146	8720	544



## Longest Range?

Aircraft	Passengers	Range (Miles)	Speed (mph)
Boeing 737-100	101	1540	598
Boeing 747	470	4150	610
Concorde	132	4000	1350
Douglas DC-8-50	146	8720	544

- Suitability to task

## Fastest?

Aircraft	Passengers	Range (Miles)	Speed (mph)
Boeing 737-100	101	1540	598
Boeing 747	470	4150	610
Concorde	132	4000	1350
Douglas DC-8-50	146	8720	544

- Suitability to task
- Customer **Latency** (time of a trip)

## Biggest Capacity?

Aircraft	Passengers	Range (Miles)	Speed (mph)
Boeing 737-100	101	1540	598
Boeing 747	470	4150	610
Concorde	132	4000	1350
Douglas DC-8-50	146	8720	544

- Suitability to task
- Customer Latency
- Customer **Bandwidth** (number of passengers in a trip)

## Largest Throughput?

Aircraft	Passengers	Speed (mph)	Passenger-mph
Boeing 737-100	101	598	60,398
Boeing 747	470	610	286,700
Concorde	132	1350	178,200
Douglas DC-8-50	146	544	79,424

- Suitability to task
- Customer Latency
- Customer Bandwidth
- Customer **Throughput** (passenger trips per unit time)

## Which Aircraft Is Best?

Aircraft	Passengers	Speed (mph)	Passenger-mph
Boeing 737-100	101	598	60,398
Boeing 747	470	610	286,700
Concorde	132	1350	178,200
Douglas DC-8-50	146	544	79,424

- Suitability to task
- Customer Latency
- Customer Bandwidth
- Customer Throughput
- Cost to purchase? Operation cost? Safety?

## Defining Performance

What is important to whom?

Computer system user:

- response time - related to program elapsed time
- $\text{elapsed\_time} = \text{time\_end} - \text{time\_start}$
- Lower elapsed time for program is better

Computing center manager:

- throughput - job completion rate
- job completion rate (#jobs/second)
- Larger job completion rate is better

11

## Improving Performance

- Response Time, Throughput, or Both?
- If we upgrade a machine with a new processor what do we increase?
- If we add a new machine to the lab what do we increase?

## Response Time Measurement

$$\text{CPU time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Cycle}}$$

Determined by  
Compiler and  
ISA Design

Determined by  
ISA Design and  
Microarchitecture

Determined by  
Microarchitecture  
and Technology

13

## Response Time Measurement

$$CPU\ time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

- Performance is inverse of CPU time
- Dynamic Instructions
- Instruction-Level Parallelism
  - CPI - Cycles per Instruction
  - IPC - Instructions per Cycle
- Throughput?

14

## Throughput Metrics

- Rates: Units of work per unit time
- Examples:
  - millions of instructions / second (MIPS)
  - millions of floating point instructions / second (MFLOPS)
  - millions of bytes / second (MBytes/sec)
  - millions of bits / second (Mbits/sec)
  - images / second
  - samples / second
  - transactions / second (TPS)

15

## MIPS and MFLOPS

- MIPS = instruction count / (execution time x 10<sup>6</sup>)
- MIPS = clock rate / (CPI x 10<sup>6</sup>)
- MFLOPS - MIPS for floating point operations
- But MIPS has serious shortcomings...
- When is MIPS OK?

16

## Meaningful Rates

Use rates that measure something useful!

### Example: Video Image Processing

- **Bad: MFLOPS**
  - Number of FLOPS depends on algorithm
  - O(n<sup>2</sup>) matrix-vector product vs. O(n log n) FFT
- **Better: frames/sec**
  - A faster running program will process more frames per second
  - Frames/sec measures speed of target application

17

## Peak Rates

- Example:  
The i860 is *advertised* as having a peak rate of 80 MFLOPS (40 MHz with 2 flops per cycle).
- Measured MFLOPS tell a different story:



Kernel	1D FFT	SASUM	SAXPY	SDOT	SGEMM	SGEMV	SPVMA
MFLOPS	8.2	3.2	6.1	10.3	6.2	15.0	8.1
% Peak	11%	4%	7%	13%	8%	19%	10%

- Peak MFLOPS: MFLOPS obtained for some contrived (and mostly likely useless) scenario.
- Peak rates are useless!

18

## Relative Performance

### Absolute time measure:

- Straightforward measurement of time a task takes
- AKA: running time, elapsed time, response time, latency, completion time, execution time

### Relative (normalized) time measures:

- Running time normalized to some reference time
- $\text{task\_time} / \text{reference\_time}$  (time = 1 / performance)
- Used to compare machines:
  - Machine A finishes task in 10 seconds
  - Machine B finishes task in 15 seconds
  - Machine A is (15 seconds / 10 seconds) 1.5x faster than B
  - Machine A is 50% faster than B

20

## Travel Time and Time Travel

- You plan to visit a friend in Turkey
- Concorde to Paris + 737 to Istanbul = \$3500
- 747 to Paris + 737 to Istanbul = \$1200

Equipment	New York to Paris	Paris to Istanbul	Total
747 + 737	8 Hours	4 Hours	12 Hours
SST + 737	3 Hours	4 Hours	7 Hours

- Taking the SST (which is 2.7 times faster) speeds up the overall trip by only a factor of 1.7!
- Teleporter to Paris? (Teleporter is  $10^6$  times faster)
- Time Machine to Paris?

21

## Amdahl's Law

- Fraction optimized limits overall speedup
- Amdahl's Law:

$$\text{Speedup} = \frac{1}{1 - f + \frac{f}{s}}$$

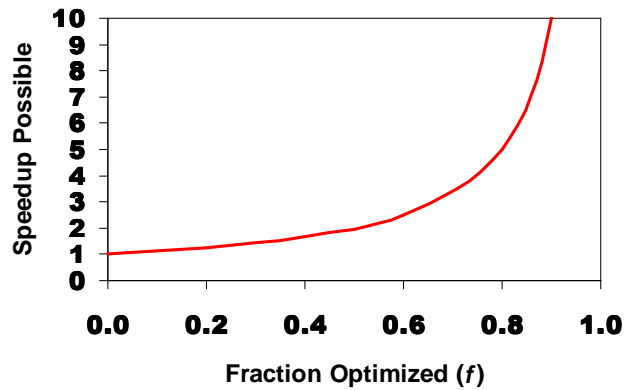
where f is fraction optimized,  
s is speedup of that fraction



22

## Amdahl's Law

Speed Enhancement is limited by fraction optimized:



$$\lim_{s \rightarrow \infty} \frac{1}{1 - f + \frac{f}{s}} = \frac{1}{1 - f}$$

where f is fraction optimized,  
s is speedup of that fraction

23

## Parallelism

Parallel Processing - throw more processors at problem

- 1024 parallel processors - LOTS OF MONEY!
- 90% of code is parallel (f = 0.9)
- Parallel portion speeds up by 1024 (s = 1024)
- Serial portion of code (1-f) limits speedup

$$\lim_{s \rightarrow \infty} \frac{1}{1 - f + \frac{f}{s}} = \frac{1}{1 - f}$$



Serial portion limits to 10x speedup!

24

## Summary

- Beware of metrics in general (MFLOP, MIPS)
- Beware of peak measurements
- Beware of kernels
- Relative and Absolute Performance
- Amdahl's law
- IPC/CPI

$$Speedup = \frac{1}{1 - f + \frac{f}{s}}$$

$$CPU\ time = \frac{Instructions}{Program} \times \frac{Cycles}{Instruction} \times \frac{Seconds}{Cycle}$$

25

## Processor Performance

Aircraft have many applications

Computer systems have many applications

- Scientific computing
- Transaction processing
- Real-time systems
- Multimedia applications
- Commercial workloads
- Software development

Systems will perform differently in each domain

27

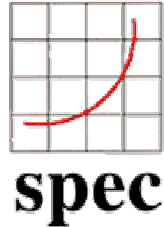
## Use Benchmark Suites

Benchmark suites are designed to standardize the evaluation of machines



Suites just from **Standard Performance Evaluation Corporation:**

[SPECcapc](#), [SPECviewperf](#), [SPEC HPC2002](#), [SPEC OMP](#)  
[SPEC CPU2000](#), [SPECjAppServer2001](#),  
[SPECjAppServer2002](#), [SPEC JBB2000](#) ,  
[SPEC JVM98](#), [SPEC MAIL2001](#), [SPEC SFS97\\_R1](#),  
[SPEC WEB99](#), [SPEC WEB99\\_SSL](#)



Choose the suite to match a particular domain

28

## Beware of “kernels”

Kernels are extracted from programs  
Meant to be the essence of an application

Example: Olden

- Something is often lost in the kernel-ization
  - Monolithic task
  - Small and regular
- Some programs in Olden produce no output!
  - Compiler Optimization

Why is Olden called Olden?

29

## Processor Performance

Benchmark suites are designed to standardize the evaluation of machines



**Beware of performance measures that do not involve full applications!**

30