Note Title 10/20/2009

# Performance Analysis

measure
interpret
communicate
a computer system's speed
or
size ← space performance

Why do performance analysis?

1. compare alternate systems.
2. improve performance of your applications.

# How to do performance analysis?

1. Measure actual systems

   → high external validity

   but

   1. results are limited to actual systems measured.
   2. not always possible to set levels of the indep. var.

2. Simulation

→ easy to change levels of indep. var.

BUT

simulations are always simpler than real systems

↓ external validity.

3. Mathematical Models

— very simple ↓ external validity

— fast and easy

Experiment

- find relationship b/w $X$ and $Y$

  ($X$ ← indep. var; $Y$ ← performance)

approximate $Y$ with measurable $\hat{Y}$ (measurable $\hat{Y}$ = metric)

<u>performance metrics</u>

counts, time durations, size of some parameter high internal validity

→ if you report a relationship between $\hat{X}$ and $\hat{Y}$, and there really is a relationship between $\hat{X}$ and $\hat{Y}$.

high external validity

→ if $\hat{Y}$ is a good approx. of $Y$?

## Metrics

→ rate metric or throughput

$$\frac{\text{count of something}}{\text{time}}$$

example: $\frac{\text{cycle}}{\text{sec}}$ or Hz

→ scalar metrics

not rates. counts, time interval

# Characteristics of a good performance metric

① reliability

if $\hat{Y}_1 > \hat{Y}_2$, then $Y_1 > Y_2$.

② repeatability

redo experiment → get same value of metric

③ consistency

metric defn stays the same from system to system.

## Some more metrics

MIPS

million instructions /sec

reliable ? not really, no. ignores memory.
repeatable ? maybe
consistent ? NO!

MFLOPS

millions of floating point opns / sec

SPEC benchmark , SPEC metric

specific set of programs

1. Measure time required to execute each program.
2. Normalize each time value by dividing by the time needed on a standard base machine.
3. Compute the geometric mean of normalized values.

SPEC : sadly not reliable ☹

# Scalar Metrics

not repeatable

1. Wall clock time

   time at finish − time at start

   → includes time spent on other apps

2. CPU time

   ↑ not reliable! doesn't account for memory delays

3. Response time

   time result appears − time made request

**Speedup** of system 2 wrt system 1

$$S_{2,1} = \frac{R_2}{R_1} = \frac{\frac{D}{T_2}}{\frac{D}{T_1}} = \frac{T_1}{T_2}$$

"D": "distance" measured

→ ex: MIPS  D = million instructions

if $T_2 < T_1 \implies S_{2,1} > 1$

if $T_2 > T_1 \implies S_{2,1} < 1$ "slowdown"

# Means

A good mean (in systems)

① is directly proportional to total time.

double $\sum x_i \longrightarrow$ double $\bar{x}$

$1+1+1+1 \quad \Sigma=4 \qquad \bar{x}=1$

$2+2+2+2 \quad \Sigma=8 \qquad \bar{x}=2$

$1+1+1+5 \quad \Sigma=8 \qquad \bar{x}=2$

} arithmetic mean meets #1

② a mean of rates should be <u>inversely</u> proportional to the total time.

10 Hz. If cycles take twice as long, mean should be cut in ½ → 5 Hz.

For scalar measurements, like time

(1) $\bar{x} \propto \underbrace{\sum T_i}_{\text{total time}}$

true for $\bar{x}_A$ if $\bar{x}_A = k \sum_i T_i$

well $\bar{x}_A = \frac{1}{n} \sum_i T_i \Rightarrow k = 1/n$.

(2) For rates

$$\bar{x} = \frac{k}{\sum T_i}$$

true for $\bar{x}_A$ if $\bar{x}_A = \frac{k}{\sum T_i}$

2 Hz     6 Hz     1 Hz

2 cyc/s     $6 \frac{cyc}{s}$     $1 \frac{cyc}{s}$

$$\bar{x}_A = \frac{k}{\sum T_i}$$

$$\bar{x}_A = \frac{1}{n} \sum_i \frac{(\# cyc)_i}{T_i}$$

normalize on the # cyc

$\frac{1 \, cyc}{1/2 \, sec}$     $\frac{1 \, cyc}{1/6 \, sec}$     $\frac{1 \, cyc}{1 \, sec}$

$$\bar{x}_A = \frac{1}{n} \sum_i \frac{1}{T_i} \Rightarrow \sum \frac{1}{T_i} \neq \frac{1}{\sum T_i}$$

Arithmetic mean fails criteria #2.

## Harmonic Mean

Two criteria for a good mean

1. $\bar{x} = k \sum T_i$ (scalar)
2. $\bar{x} = k / \sum T_i$ (rate)

$$\bar{x}_H = \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}$$

fails criteria #1
meets criteria #2

good for rates

## Geometric Mean – fails criteria #1 AND #2

$$\bar{x}_G = \sqrt[n]{x_1 x_2 x_3 \cdots x_n} = \left( \prod_{i=1}^{n} x_i \right)^{1/n}$$

Vibha sez: good for nothing

# Pythagorean Means

$n = 2 \qquad x_1 = 2$

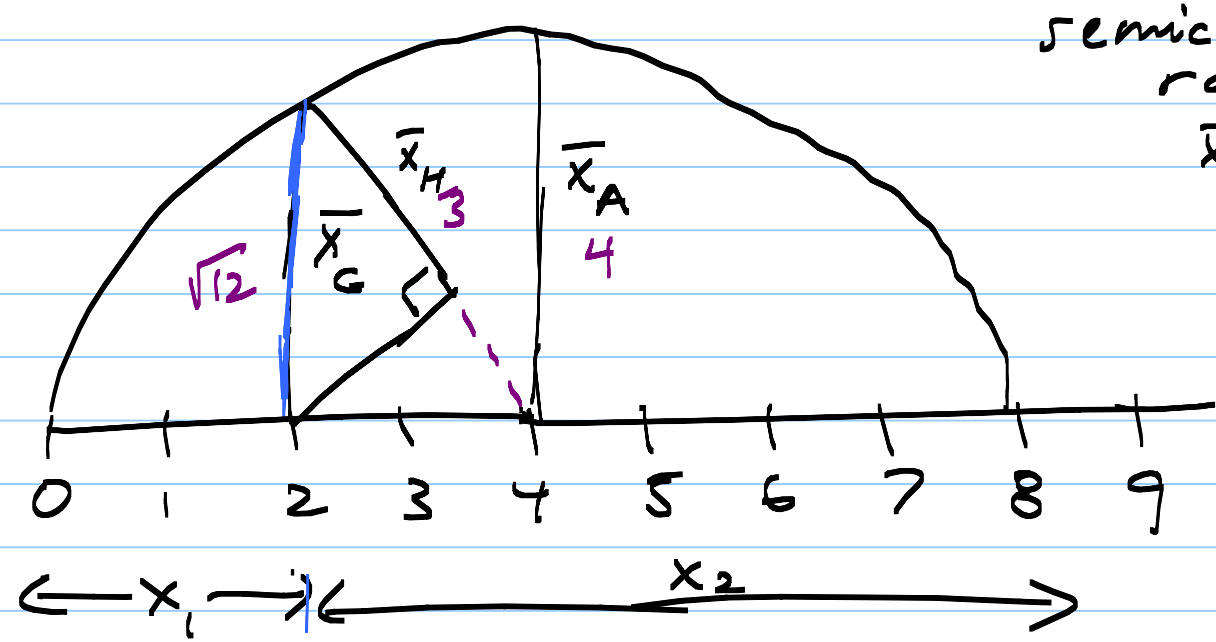
$\qquad\qquad x_2 = 6$

$$\bar{x}_H = \frac{2}{\frac{1}{2} + \frac{1}{6}} = 3$$

$$\bar{x}_G = \sqrt{2 \cdot 6} = \sqrt{12}$$

semicircle radius: 4

$\bar{x}_A = 4$



What are we measuring?

① count of events
ex: counting how many times
a function is called.

② document value of state at an event
ex: measure call stack depth
at each start of function call

③ time to complete something

④ proportion of time spent in Function F

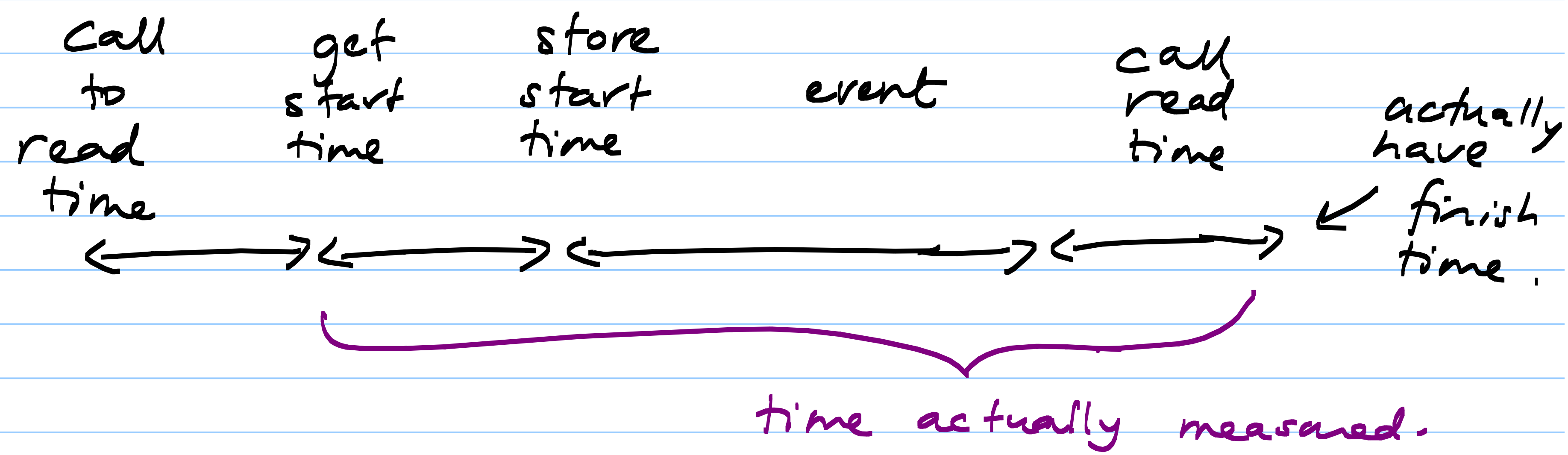
profile: overall view of where its execution time is spent

- trace: order of operations

Measuring a system <u>perturbs</u> the system.

<u>examples</u>

(1) timer overhead



Measuring a system perturbs the system.

- slow down or affect program execution because you collect and store all these counts.

# Measurement Approaches

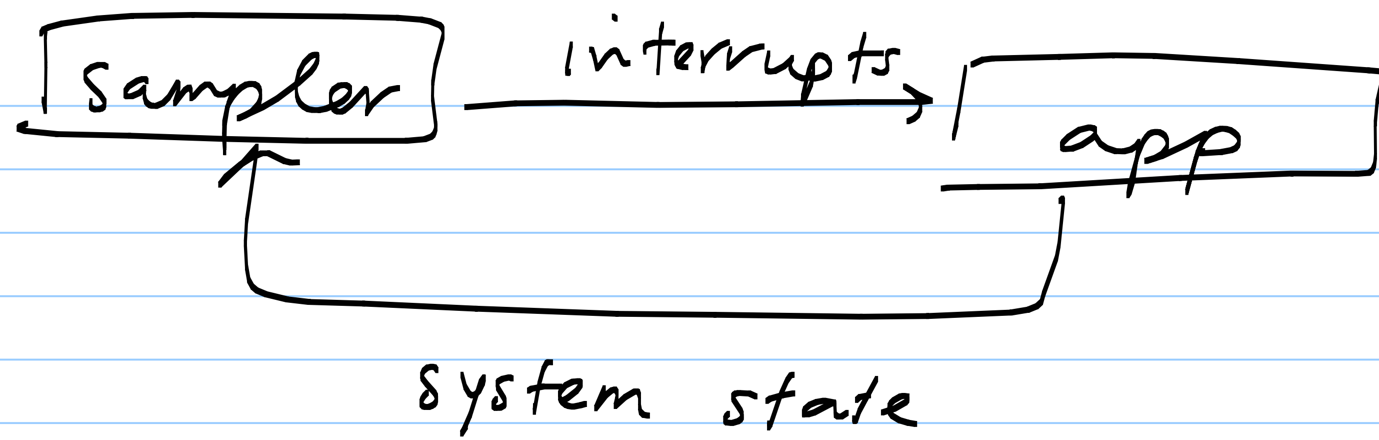
1. Insert measurement code into source code.

2. Smart compiler can add instrumentation to the executable.

3. Instruct virtual machine

4. Insert code into a running program.
   ⟹ Jeff Hollingsworth
   Dyninst Project

5. Sample: At certain times, interrupt program.


sampler
interrupts
app
system state

# Proportions

What fraction of time do you spend in function F?

use sampling.

sample 800 times. 157 times you're in F.

sample proportion is $\frac{157}{800}$

**Shortcut**: compute a confidence interval using only one sample proportion!!

each time we interrupt the system,
we are either in F or not in F

in F → 1, not in F → ∅: Bernoulli variable

$$p = \text{sample proportion of time in } F = \frac{\text{sum of bernoulli variables}}{\text{\# of times you interrupt}}$$

hey, a sample mean!

$np \geq 10$

as $n \to \infty$,

distribution of $P$ approaches a normal distribution

with $\mu = p$

$\sigma^2 = \frac{p(1-p)}{n}$

compute a 95% confidence interval

$$p \pm z(0.025)\sqrt{\frac{p(1-p)}{n}}.$$