CPS311 Lecture: Performance Evaluation

NEED: Web access, projectable of Null table 11.6

1. Measuring the performance of a computer system is important for a number of reasons.

ASK for examples

- Deciding which system to purchase
- Determining if a specific application is feasible.
- "Tuning" a system to optimize performance

But how do we measure the performance of a computer system?

2. Manufacturers tend to publish numbers that make their products appear impressive

ASK for examples

- CPU clock speed
- bus speed
- FLOPS (floating operations per second), MFLOPS, GFLOPS
- MIPS (millions of instructions per second)

3. However, such numbers are not a good way of comparing performance.

ASK Why?

- Requirements of different tasks vary greatly:
  - Compute bound versus memory bound versus IO bound tasks.
  - Tasks requiring extensive FP computation (scientific tasks) versus those requiring little or none.
- In the case of measures like FLOPS and MIPS, the instruction mix is critical - e.g. on any machine floating multiply is more complex than floating add; on CISCs different instructions can take widely varying numbers of cycles.
- "A chain is a strong as its weakest link".
  - The notion of compute bound versus memory bound versus IO bound depends on relative speed of components. The same task may be IO bound on one system, memory bound on another system, and compute bound on another.
  - "VonNeumann bottleneck" (memory). Hierarchical memory attempts to yield a system that is close to as fast as its fastest component, but degree of success depends on proper sizing of component parts and may be application specific (e.g. applications with greater inherent locality do much better than those with little locality).
  - Other issues such as network communication ...

4. A better way to measure performance is through use of benchmarks that test a system's performance in a workload that is perceived to be similar to that in which the intended system will be used.

ASK if you've ever seen these

- Trade magazines typically have suites they use for product reviews.

Example: http://www.macworld.com/article/143698/2009/11/speedmark6_intro.html

- There are industry-standard benchmark suites. The most-widely known are the SPEC suites (Standard Performance Evaluation Corporation), which has

been developing suites since 1988 (continually being revised, of course!)

Example: http://www.spec.org/ - then navigate to graphics/workstations, then published results for SPECviewperf@10, then Dec 2, 2009 results summary

NOTE: A problem with benchmarks is that a manufacturer may actually optimize with a particular benchmark in mind. This has happened - sometimes in very significant ways (e.g. special compiler flags used when compiling a particular benchmark program in order to optimize the code in a benchmark-specific way). This sort of thing is still allowed with suites such as SPEC - but SPEC publishes two numbers for a given system - with and without optimization flags.

5. Two issues with benchmark suites

- No system is typically best on every test.

  Example: look at example SPEC results (high numbers are better).

- A host of number can get confusing, especially when system price is also considered

6. Some things one may do:

- Calculation of an overall score

  (SPEC does this for each of the individual benchmark suites, but does not attempt to produce an overall score considering all suites)

  Example: navigate to results for one suite - note individual tests, weighted average which is then carried forward into overall.

  Example: MacWorld SpeedMark 5 overall score

- Calculation of cost / performance ratio

  Example: show final column in SPEC suite details (Cost per composite).
  [again, SPEC does not do this over all suites of a given type, just within an individual suite]

- Normalization to a specific system

  Example: MacWorld data looked at earlier does this for overall summary (Show discussion in 3rd paragraph - all scores noramlized to score for 2.13 GHz MacMini w/2 GB of RAM as 100)

7. How one calculates a mean is also an issue.

  - A simple arithmetic average of raw data is usually not very good.

    ASK Why?

    - Suppose we had the following raw data (high numbers better)

	Test 1	Test 2	Test 3	Test 4
System A	1000	5	5	10
System B	980	10	10	20

    System A is slightly better than B on Test 1, but B is twice as good as A on all the other tests. What is the simple arithmetic average?

A: 1020/4 = 255
B: 1020/4 = 255

Averages suggest both are equal, which is really misleading in this case.

- Often, a weighted average is better than a simple average

Example: SPEC details - note weights

However, this has a problem, too.

ASK

Assumes some knowledge of workload mix in order to properly assign weights.

- A geometric mean may be better.

The geometric mean of n numbers is (x1 * x2 * ... * xn) ^ 1/n (nth root)
Note: geometric means are calculated using normalized values:

Example: using above data, normalizing to A

	Test 1	Test 2	Test 3	Test 4
System A	1	1	1	10
System B	.98	2	2	2

A: 4th root of 1 = 1
B: 4th root of 7.84 = 1.67
Ratio of geometric means = 1.67:1

Ratio is independent of choice of base system - e.g. normalizing to B

	Test 1	Test 2	Test 3	Test 4
System A	1.02	0.5	0.5	0.5
System B	1	1	1	1

A: 4th root of .1275 = 0.6
B: 4th root of 1 = 1
Ratio of geometric means = 1:0.6 = 1.67:1

Geometric mean is not useful for predicting performance - i.e. the above data does _not_ mean that typically system B is 1.67 times faster than A!

- When rates are involved, one must use the harmonic mean

Example: Suppose we made a 30 mile trip, going 10 miles at 10 mph, 10 at 40 mph, and 10 at 70 mph. What is the average speed? It is _not_ (10 + 40 + 70) / 3 = 40 mph!

Clearly, going 10 miles at 10 mph takes 60 minutes. Going 10 miles at 40 mph takes 15 minutes, and going 10 miles at 70 mph takes about 8.6 min. So the total trip takes 83.6 minutes, for an average speed of only 21.5 mph.

When we have rates, the harmonic mean is calculated as 1 over the average of the reciprocals - e.g.

1/((1/10 + 1/40 + 1/70) / 3) = 1/((.1 + .025 + .014)/3) = 1/ .0463 = 21.5

- Summary table on measures of central tendency: project Null table 11.6