EEC 686/785
Modeling & Performance Evaluation of Computer Systems

Lecture 6

Wenbing Zhao
Department of Electrical and Computer Engineering
Cleveland State University
wenbing@ieee.org

Outline

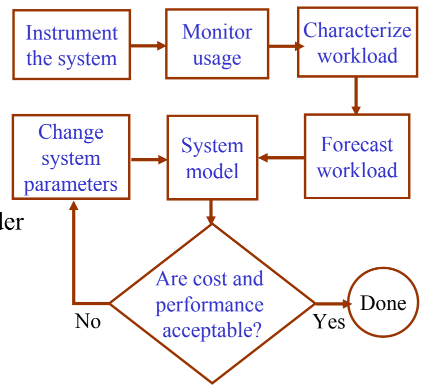
- Review of lecture 5
- The art of data presentation
  - Gnuplot
  - Histogram
  - Gantt charts
  - Kiviat graphs
  - Schumacher charts
- Ratio games

Terminology

- **Capacity planning**: ensuring that adequate computer resources will be available to meet the future workload demands in a cost-effective manner while meeting the performance objectives
- **Capacity management**: ensuring that the currently available computing resources are used to provide the highest performance
- Capacity management is concerned with the present while capacity planning is concerned with the future
- **Benchmarking**: to compare the performance of two competing systems in an objective manner, benchmarks are run on these systems using automatic load drivers

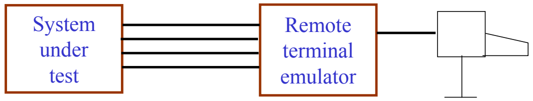
Steps in Capacity Planning and Management

- For planning as well as management, the steps are basically the same:
  - Instrument the system
  - Monitor system usage
  - Characterize workload
  - Predict performance under different alternatives
  - Select the lowest cost, highest performance alternative

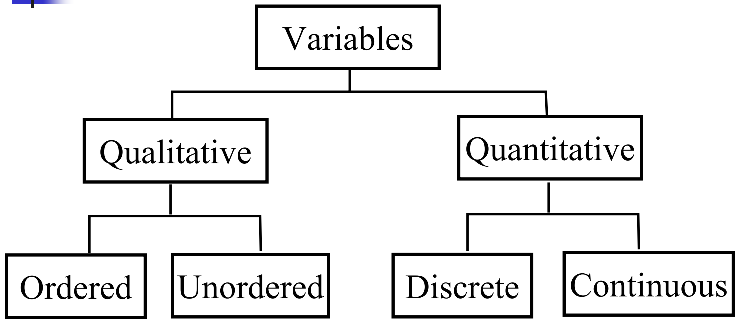


Load Drivers

- **Internal-driver method**: loading programs directly into the memory and executing
  - Problem 1: the effect of the terminal communication overhead is not visible
  - Problem 2: loading overhead may affect the system performance
- **Remote-terminal emulators** (RTEs): the computers that are used to put the load on the system



Types of Variables



Guidelines for Preparing Good Graphic Charts

- **Require minimum effort from the reader**
- **Maximize information**: there should be sufficient information on the graph to make it self-sufficient
- **Minimize ink**: present as much information as possible with as little ink as possible. Too much unnecessary info on a chart makes it cluttered and uninteresting
- **Use commonly accepted practices**: present what people expect
- **Avoid ambiguity**: show coordinate axes, scale division, and origin. Identify individual curves and bars. Do not plot multiple variables in the same chart

Gnuplot

- **Gnuplot** is a portable command-line driven interactive data and function plotting utility for many different operating systems, including MS Windows and Linux
- The software is copyrighted but freely distributed
- Gnuplot supports many types of plots in either **2D** and **3D**
- It can draw using lines, points, boxes, contours, vector fields, surfaces, and various associated text
- It also supports various specialized plot types
- Gnuplot Website: http://www.gnuplot.info/

Histogram

- A **histogram** is defined as a bar graph that shows frequency data
- Differences between histograms and bar graphs
  - Think of histograms as sorting bins
    - Given one variable, sort data by this variable by placing them into "bins"
    - Then count how many pieces of data are in each bin. The height of the rectangle on top of each bin is proportional to the number of pieces in that bin

Histogram

- Bar graphs show the comparison of **several** measurements of **different** items
- The main question a histogram answers is: "**How many** measurements are there in each of the classes of measurements?"
- The main question a bar graph answers is: "**What** is the measurement for each item?"

Gantt Charts

- **Gantt chart** can be used to show the relative duration of any number of Boolean conditions, i.e., conditions that are either true or false
  - A resource being used or being idle is an example of a Boolean condition
  - Each condition is shown to be a set of horizontal line segments
  - The total length of the line segments represents the relative duration of the condition
  - The position of various segments is arranged such that the overlap between different lines represents the overlap between the conditions

Gantt Charts

- Gantt chart is particularly useful to show the utilization profiles of the overlap among different resources
  - Any resource whose utilization is too high is a bottleneck and may degrade the performance
  - A resource with very low utilization represents inefficiency in the system
  - The overlap of utilization of different resources is a good thing

Gantt Charts

- CPU utilization: 60%
- I/O channel utilization: 40%
- Overlap between the CPU and I/O channel is 20%
- Network link utilization: 60%
- During 30% of the time, the CPU and network but not the I/O channel are used
- During 10% of the time all three resources are used
- For 5% of the time, I/O and network are used but the CPU is idle
- Network alone is used for 15% of the time

Procedure to Develop a Gantt Chart from Raw Data

- Example 10.1: four resources:
  - CPU (A)
  - Two I/O channels (B and C)
  - A network link (D)
  - => 16 possible states

Procedure to Develop a Gantt Chart from Raw Data

- First level is divided into two parts representing A and Ā
- A dotted vertical line is drawn to divide all subsequent levels in two parts as well
- The sum of 8 entries with A=1 is 55%
- In the final chart, the division corresponding to A and Ā will have length proportional to 55 and 45, respectively. This is indicated by the labels below A and Ā

Procedure to Develop a Gantt Chart from Raw Data

- Final Gantt chart for example 10.1

Kiviat Graphs

- **Kiviat graph**: a circular graph in which several different performance metrics are plotted along radial lines
  - In the most popular version of the graph, an even number of metrics are used.
  - Half of these metrics are HB metrics so that a higher value of the metrics is considered better.
  - The other half of the metrics measure are LB metrics, and a lower value is considered better
  - **Kiviat graph for an ideal system is star**

Example 10.2

- Eight different metrics
  - **CPU busy** – percentage of time CPU busy, i.e., CPU utilization, a HB metric (the resource is being used and not wasted)
  - **CPU only busy** – percentage of time CPU only is busy, measures the fraction of time when only the CPU is used and none of the I/O channels are used, an LB metric
  - **CPU/channel overlap** – measures the percentage of time the CPU is busy along with at least one I/O channel, a HB metric
  - **Channel only busy** – measures the time when the channel is being used but the CPU is idle. The CPU must be waiting for the I/O, an LB metric

Example 10.2

- Eight different metrics
  - **Any channel busy** – a HB metric
  - **CPU wait** – CPU waiting for I/O completion, resource is wasted, an LB metric
  - **CPU in problem state** – indicates time used executing the user's program, a HB metric
  - **CPU in supervisor state** – indicates the time spent executing the operating system code, an LB metric

Example 10.2

- Kiviat graph for a balanced system
- Problem with this set of metrics: only 4 independent metrics
  - CPU busy = problem state + supervisor state
  - CPU wait = 100 – CPU busy
  - Channel only = any busy – CPU/channel overlap
  - CPU only = CPU busy – CPU/channel overlap

Shapes of Kiviat Graphs

CPU **keelboat** – Kiviat graph for a CPU-bound system

I/O **wedge** – Kiviat graph for an I/O-bound system

Shapes of Kiviat Graphs

I/O **arrow** – Kiviat graph for a CPU- and I/O-bound system

Kiviat Graphs

Which system is more balanced?

Figure of Merit

- The square root of the area of the reversed Kiviat graph can be used to compare the overall goodness of systems
- Consider a Kiviat graph with 2n axes.
  - Let the percentage of performance values for a system be $\{x_1, x_2, \ldots, x_{2n}\}$, where odd values $x_1, x_3, \ldots, x_{2n-1}$ represent good metrics and the even values represent bad metrics
  - All x_i's are percentages and lie between 0 and 100
  - Figure of Merit (FOM) is computed as follows:

$$FOM = \left[\frac{1}{2n} \sum_{i=1}^{n} (x_{2i-1} + x_{2i+1})(100 - x_{2i}) \right]^{1/2}$$

Example 10.3

System	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8
A	100	60	40	0	40	0	40	60
B	70	30	40	30	70	30	40	30

$$FOM_A = \left[\frac{1}{8}\{(100+40)(100-60)+(40+40)(100-0)+(40+40)(100-0)+(40+100)(100-60)\}\right]^{1/2}$$
$$= \left[(5600+8000+8000+5600)/8\right]^{1/2} = \sqrt{28200/8} = 58$$

$$FOM_B = \left[\frac{1}{8}\{(70+40)(100-30)+(40+70)(100-30)+(70+40)(100-30)+(40+70)(100-30)\}\right]^{1/2}$$
$$= \left[(7700+7700+7700+7700)/8\right]^{1/2} = \sqrt{30800/8} = 62$$

Application of Kiviat Graphs

- Most of the literature on Kiviat graphs is on data processing systems
- Kiviat graph can be extended to networks, databases, and other types of computer systems

Application of Kiviat Graphs

Schumacher Charts

- **Schumacher Chart**: another reporting format to show daily, weekly, monthly, quarterly, or yearly resource usage
- Any number of performance metrics can be plotted in tabular form
- The values are normalized with respect to the long-term mean and standard deviation
- Any observations that are beyond the mean plus or minus one standard deviation need to be explained

Decision Maker's Games

- Even if the performance analysis is correctly done and presented, it may not be enough to persuade your audience – the decision makers – to follow your recommendations
- Reasons for rejection:
  - The problem needs more analysis
  - Workload/metrics/configuration/deta is always based on past measurements, their applicability to the current or future environment can always be questioned

Ratio Games

- **Ratios** provide good opportunities for playing performance games with competitors
- Ratios have a *numerator* and *denominator* (*base*)
- **Two ratios with different bases are not comparable**
- However, many examples in published literature where computer scientists have knowingly or unknowingly compared ratios with different bases
- **Ratio game**: the technique of using ratios with incomparable bases and combining them to one's advantage

Ratio Games

- Take an average of each individual system's performance and then take a ratio
- Normalize each system's performance on each workload by that of system A and then take an average of ratios
- Normalize each system's performance on each workload by that of system B and then take an average of ratios

Example of Ratio Games

- Ratio of totals: 6502 is worse. It takes 4.7% more time
- With 6502 as a base: 6502 is better (1% more time)
- With 8080 as a base: 6502 is worse (6% more time)

Example of Ratio Games

- Sum: Z8002 requires 9% less code than RISC-I

Example of Ratio Games

- RISC-I as base: Z8002 has 18% more code than RISC-I

Using an Appropriate Ratio Metric

- Another form in which ratio games are used to show better performance is by choosing a suitable performance metric which is a ratio of two different metrics

Network	Throughput	Response
A	10	2
B	4	1

Network	Throughput	Response	Power
A	10	2	5
B	4	1	4

Using Relative Performance Enhancement

- If the performance metric is already specified, it is possible to show that relative increase in the performance is better using one type of enhancement than another *provided the two are tried on different machines* (**correct way to compare is to try on the same machine**)

Alternative	Without	With
Floating-point accelerator A on X	2	4
Floating-point accelerator B on Y	3	5

Alternative	Without	With	Ratio
A on X	2	4	2.00
B on Y	3	5	1.66

Ratio Games with Percentages

- Percentages are basically ratios. They allow playing ratio games in ways that do not look like ratios
- Example 11.3. =>
- In fact, neither comparison used compatible bases:
  - (a) number of tests used in each test as base
  - (b) number of tests combined as base

Ratio Games with Percentages

- Other ways to misuse percentages
  - They can be used to have a psychological impact on the reader.
    - For example, 1000% improvement in performance sounds more impressive than an 11-time improvement
    - This is particularly useful when the performance before the improvement and the absolute increase in the performance are not impressive

Ratio Games with Percentages

- Other ways to misuse percentages
  - Small sample sizes can be neatly disguised by specifying the percentage.
    - Saying that 83.3% of the universities in town use system X is more impressive than saying that 5 out of 6 universities use the system
  - The base in the percentage should be the initial value
    - The memory prices have gone done 400% – as if they are now paying you to buy memory – it should be the current price is 1/5 of the original

Strategies for Winning a Ratio Game

- If one system is better on all benchmarks, contradicting conclusion cannot be drawn by any ratio game technique

Strategies for Winning a Ratio Game

- Even if one system is better than the other on all benchmarks, a better relative performance can be shown by selecting the appropriate base
  - In table 11.9, A is 40% better than B using raw data, 43% better using A as base, 42% better using B as base
  - A designer would prefer to use the raw-data averages

Strategies for Winning a Ratio Game

- If one system is better on some benchmarks and worse on others, contradicting conclusions can be drawn in some cases
  - The easiest way to verify whether a contradictory conclusion can be drawn for a particular data set is to try all possible bases
  - The next several rules may help select the base

Strategies for Winning a Ratio Game

- If the performance metric is an LB metric, it is better to use your system as the base
  - The execution time is an LB metric, and as shown in table 11.9, system A designers would be better off using system A as the base. B designers would prefer to use B as the base

Strategies for Winning a Ratio Game

- If the performance metric is an HB metric, it is better to use your opponent as the base
  - The throughputs, efficiency, MPS, and MFLOPS are examples of HB metrics
  - A higher average ratio would be obtained for A if B is used as a base

Strategies for Winning a Ratio Game

- Those benchmarks that perform better on your system should be elongated and those that perform worse should be shortened
  - The time duration of the benchmarks is often adjustable
  - For example, in Sieve benchmark, the number of prime numbers to be generated can be set by the experimenter
  - Elongating a benchmark in effect increases the weight on the favorable benchmark

Derivation of the Rules

- Assume the performance metric is an HB metric. Consider the case of two systems A and B on two benchmarks I and J etc. as shown in table 11.10

Derivation of the Rules

- Using raw data, A is better if and only if:

$$y < -\frac{a}{b}x + \frac{a+b}{b}$$

- Using A as base, A is considered better if and only if:

$$y < 2 - x$$

- Using B as base, A is considered better if and only if:

$$y < \frac{x}{2x-1}$$

Derivation of the Rules

- This explains why it is better to use your opponent's system as base for HB metric

Correct Analysis

- The main reason why the analysis of ratios results in contradicting results is that the approach of taking a mean of the ratio is wrong
- The approach completely ignores the fact that the performance is affected by several factors and by experimental errors
- A statement about one factor can be made only if the effects of different factors and errors are first isolated
- The isolation requires developing a model of the way these factors and experimental errors interact with each other
- Techniques for this require knowledge of several probabilistic, statistical, and experimental design concepts