CS 147:
Computer Systems Performance Analysis
Selecting Techniques

Overview

Overview

Making Decisions
Techniques
Metrics
Response Time
Processing Rate
Resource Consumption
Error Metrics
Financial Measures
Types of Metrics
Choosing Metrics
Criteria
Classes of Metrics
Requirements

Decisions to Be Made

- Evaluation technique
- Performance metrics
- Performance requirements

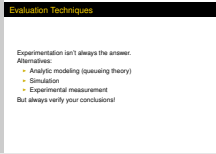
Decisions to Be Made
- Evaluation technique
- Performance metrics
- Performance requirements

Evaluation Techniques

Experimentation isn't always the answer.
Alternatives:

- Analytic modeling (queueing theory)
- Simulation
- Experimental measurement

But always verify your conclusions!



Analytic Modeling

- Cheap and quick
- Don’t need working system
- Usually must simplify and make assumptions

Analytic Modeling
- Cheap and quick
- Don’t need working system
- Usually must simplify and make assumptions

Simulation

- Arbitrary level of detail
- Intermediate in cost, effort, accuracy
- Can get bogged down in model-building

Simulation
- Arbitrary level of detail
- Intermediate in cost, effort, accuracy
- Can get bogged down in model-building

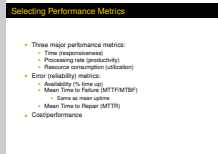
Measurement

- Expensive
- Time-consuming
- Difficult to get detail
- But accurate

Measurement
- Expensive
- Time-consuming
- Difficult to get detail
- But accurate

Selecting Performance Metrics

- Three major perfomance metrics:
  - Time (responsiveness)
  - Processing rate (productivity)
  - Resource consumption (utilization)
- Error (reliability) metrics:
  - Availability (% time up)
  - Mean Time to Failure (MTTF/MTBF)
    - Same as mean uptime
  - Mean Time to Repair (MTTR)
- Cost/performance



Response Time

- How quickly does system produce results?
- Critical for applications such as:
  - Time sharing/interactive systems
  - Real-time systems
  - Parallel computing

Response Time

- How quickly does system produce results?
- Critical for applications such as:
  - Time sharing/interactive systems
  - Real-time systems
  - Parallel computing

Examples of Response Time

- Time from keystroke to echo on screen

Examples of Response Time
- Time from keystroke to echo on screen

Examples of Response Time

- Time from keystroke to echo on screen
- End-to-end packet delay in networks

Examples of Response Time
- Time from keystroke to echo on screen
- End-to-end packet delay in networks

Examples of Response Time

- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time

Examples of Response Time
- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time

Examples of Response Time

- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time
- Leaving Galileo to getting food in Hoch-Shanahan

Examples of Response Time
- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time
- Leaving Galileo to getting food in Hoch-Shanahan

Examples of Response Time

- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time
- Leaving Galileo to getting food in Hoch-Shanahan
  - Edibility not a factor

Examples of Response Time
- Time from keystroke to echo on screen
- End-to-end packet delay in networks
- OS bootstrap time
- Leaving Galileo to getting food in Hoch-Shanahan
  - Edibility not a factor

Measures of Response Time

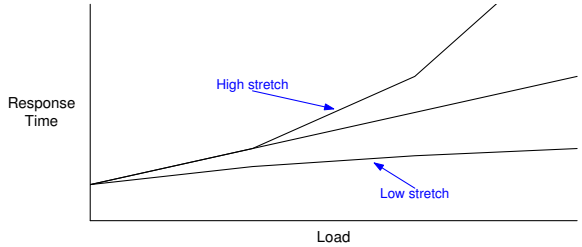
- Response time: request-response interval
- Measured from end of request
- Ambiguous: beginning or end of response?
- Reaction time: end of request to start of processing
- Turnaround time: end of request to end of response

Measures of Response Time

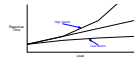
- Response time: request-response interval
- Measured from end of request
- Ambiguous: beginning or end of response?
- Reaction time: end of request to start of processing
- Turnaround time: end of request to end of response

The Stretch Factor

- Response time usually goes up with load
- *Stretch Factor* measures this:

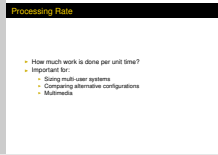


The Stretch Factor
Response time usually goes up with load
Stretch Factor measures this:



Processing Rate

- How much work is done per unit time?
- Important for:
  - Sizing multi-user systems
  - Comparing alternative configurations
  - Multimedia



Examples of Processing Rate

- Bank transactions per hour

Examples of Processing Rate
- Bank transactions per hour

Examples of Processing Rate

- Bank transactions per hour
- File-transfer bandwidth

Examples of Processing Rate
- Bank transactions per hour
- File-transfer bandwidth

Examples of Processing Rate

- Bank transactions per hour
- File-transfer bandwidth
- Aircraft control updates per second

Examples of Processing Rate
- Bank transactions per hour
- File-transfer bandwidth
- Aircraft control updates per second

Examples of Processing Rate

- Bank transactions per hour
- File-transfer bandwidth
- Aircraft control updates per second
- Jurassic Park customers per day

Examples of Processing Rate
- Bank transactions per hour
- File-transfer bandwidth
- Aircraft control updates per second
- Jurassic Park customers per day

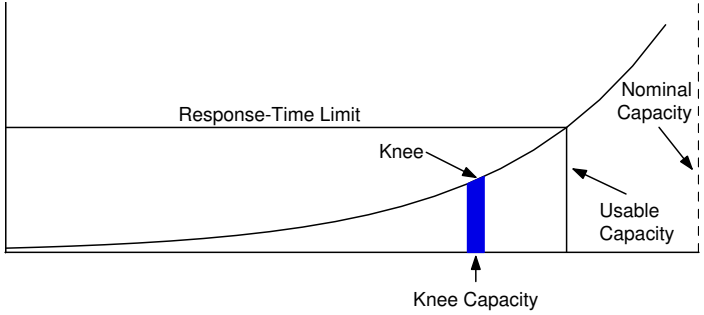
Measures of Processing Rate

- *Throughput*: requests per unit time: MIPS, MFLOPS, Mb/s, TPS
- *Nominal capacity*: theoretical maximum: bandwidth
- *Knee capacity*: where things go bad
- *Usable capacity*: where response time hits a specified limit
- *Efficiency*: ratio of usable to nominal capacity

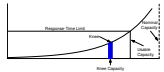
Measures of Processing Rate

- *Throughput*: requests per unit time: MIPS, MFLOPS, Mb/s, TPS
- *Nominal capacity*: theoretical maximum: bandwidth
- *Knee capacity*: where things go bad
- *Usable capacity*: where response time hits a specified limit
- *Efficiency*: ratio of usable to nominal capacity

Nominal, Knee, and Usable Capacities



Nominal, Knee, and Usable Capacities



Resource Consumption

- How much does the work cost?
- Used in:
  - Capacity planning
  - Identifying bottlenecks
- Also helps to identify “next” bottleneck

Resource Consumption

- How much does the work cost?
- Used in:
  - Capacity planning
  - Identifying bottlenecks
- Also helps to identify “next” bottleneck

Examples of Resource Consumption

- CPU non-idle time

Examples of Resource Consumption
- CPU non-idle time

Examples of Resource Consumption

- CPU non-idle time
- Memory usage

Examples of Resource Consumption
- CPU non-idle time
- Memory usage

Examples of Resource Consumption

- CPU non-idle time
- Memory usage
- Fraction of network bandwidth needed

Examples of Resource Consumption
- CPU non-idle time
- Memory usage
- Fraction of network bandwidth needed

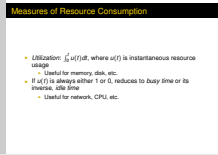
Examples of Resource Consumption

- CPU non-idle time
- Memory usage
- Fraction of network bandwidth needed
- Square feet of beach occupied

Examples of Resource Consumption
- CPU non-idle time
- Memory usage
- Fraction of network bandwidth needed
- Square feet of beach occupied

Measures of Resource Consumption

- *Utilization*: $\int_0^t u(t)dt$, where $u(t)$ is instantaneous resource usage
  - Useful for memory, disk, etc.
- If $u(t)$ is always either 1 or 0, reduces to *busy time* or its inverse, *idle time*
  - Useful for network, CPU, etc.



Error Metrics

- Successful service (speed)
  - (Not usually reported as error)
- Incorrect service (reliability)
- No service (availability)

Error Metrics
- Successful service (speed)
  - (Not usually reported as error)
- Incorrect service (reliability)
- No service (availability)

Examples of Error Metrics

- Missed disk seeks

Examples of Error Metrics
- Missed disk seeks

Examples of Error Metrics

- Missed disk seeks
- Dropped Internet packets

Examples of Error Metrics
- Missed disk seeks
- Dropped Internet packets

Examples of Error Metrics

- Missed disk seeks
- Dropped Internet packets
- ATM down time

Examples of Error Metrics
- Missed disk seeks
- Dropped Internet packets
- ATM down time

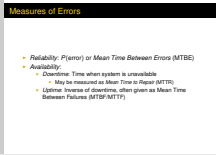
Examples of Error Metrics

- Missed disk seeks
- Dropped Internet packets
- ATM down time
- Wrong answers from IRS

Examples of Error Metrics
- Missed disk seeks
- Dropped Internet packets
- ATM down time
- Wrong answers from IRS

Measures of Errors

- *Reliability*: *P*(error) or *Mean Time Between Errors* (MTBE)
- *Availability*:
  - *Downtime*: Time when system is unavailable
    - May be measured as *Mean Time to Repair* (MTTR)
  - *Uptime*: Inverse of downtime, often given as Mean Time Between Failures (MTBF/MTTF)



Financial Measures

- When buying or specifying, *cost/performance* ratio is often useful
- Performance chosen should be most important for application

Financial Measures
- When buying or specifying, *cost/performance* ratio is often useful
- Performance chosen should be most important for application

Characterizing Metrics

- Usually necessary to summarize
- Sometimes means are enough
- Variability is usually critical
  - A mean I-210 freeway speed of 55 MPH doesn't help plan rush-hour trips

Characterizing Metrics
- Usually necessary to summarize
- Sometimes means are enough
- Variability is usually critical
  - A mean I-210 freeway speed of 55 MPH doesn't help plan rush-hour trips

Types of Metrics

- Global across all users
- Individual

First helps financial decisions, second measures satisfaction and cost of adding users

Types of Metrics
- Global across all users
- Individual

First helps financial decisions, second measures satisfaction and cost of adding users

Choosing What to Measure

Pick metrics based on:

- Completeness
- (Non-)redundancy
- Variability

Choosing What to Measure
Pick metrics based on:
- Completeness
- (Non-)redundancy
- Variability

Completeness

- Must cover everything relevant to problem
  - Don't want awkward questions from boss or at conferences!
- Difficult to guess everything *a priori*
  - Often have to add things later

Redundancy

- Some factors are functions of others
- Measurements are expensive
- Look for minimal set
- Again, often an interactive process

Redundancy
- Some factors are functions of others
- Measurements are expensive
- Look for minimal set
- Again, often an interactive process

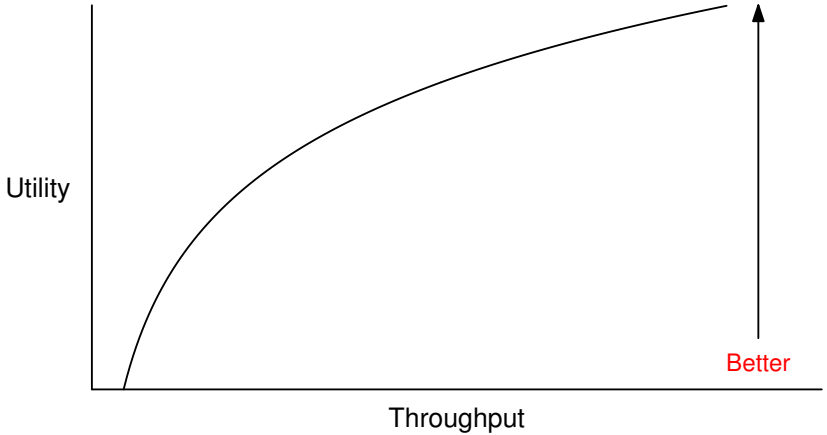
Variability

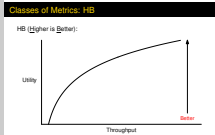
- Large variance in a measurement makes decisions impossible
- Repeated experiments can reduce variance
  - Expensive
  - Can only reduce it by a certain amount
- Better to choose low-variance measures to start with

Variability
- Large variance in a measurement makes decisions impossible
- Repeated experiments can reduce variance
  - Expensive
  - Can only reduce it by a certain amount
- Better to choose low-variance measures to start with

Classes of Metrics: HB

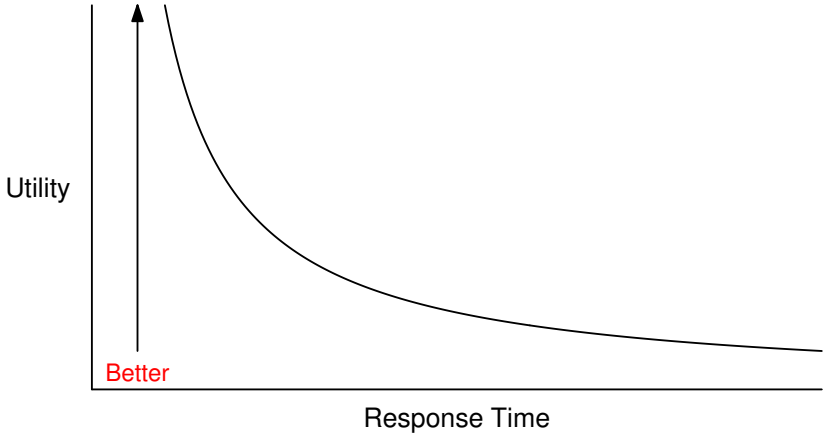
HB (Higher is Better):

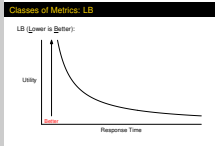




Classes of Metrics: LB

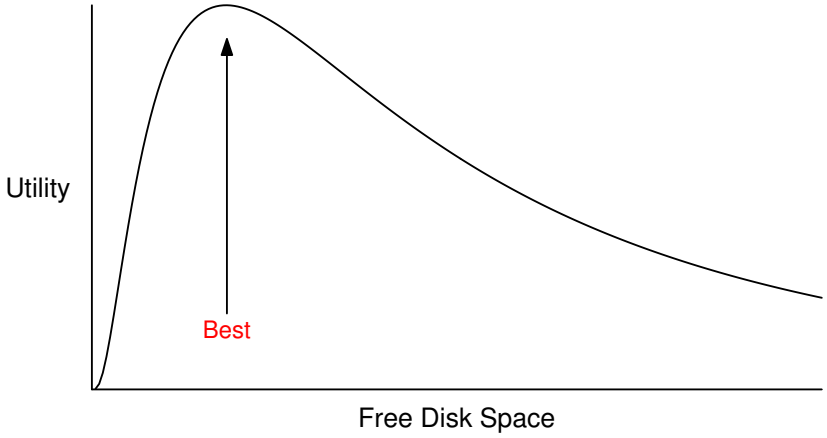
LB (Lower is Better):

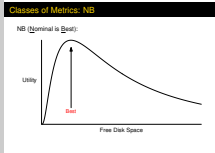




Classes of Metrics: NB

NB (Nominal is Best):





Setting Performance Requirements

Good requirements must be SMART:

- Specific
- Measurable
- Acceptable
- Realizable
- Thorough

Setting Performance Requirements

Good requirements must be SMART:

- Specific
- Measurable
- Acceptable
- Realizable
- Thorough

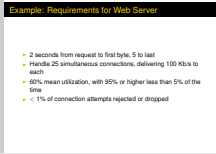
Example: Web Server

- Users care about response time (end of response)
- Network capacity is expensive want high utilization
- Pages delivered per day matters to advertisers
- Also care about error rate (failed & dropped connections)

Example: Web Server
- Users care about response time (end of response)
- Network capacity is expensive want high utilization
- Pages delivered per day matters to advertisers
- Also care about error rate (failed & dropped connections)

Example: Requirements for Web Server

- 2 seconds from request to first byte, 5 to last
- Handle 25 simultaneous connections, delivering 100 Kb/s to each
- 60% mean utilization, with 95% or higher less than 5% of the time
- $<$ 1% of connection attempts rejected or dropped



Is the Web Server SMART?

- Specific: yes

Is the Web Server SMART?
- Specific: yes

Is the Web Server SMART?

- Specific: yes
- Measurable: may have trouble with rejected connections

Is the Web Server SMART?
- Specific: yes
- Measurable: may have trouble with rejected connections

Is the Web Server SMART?

- Specific: yes
- Measurable: may have trouble with rejected connections
- Acceptable: response time, number of connections, and aggregate bandwidth might not be enough

Is the Web Server SMART?
- Specific: yes
- Measurable: may have trouble with rejected connections
- Acceptable: response time, number of connections, and aggregate bandwidth might not be enough

Is the Web Server SMART?

- Specific: yes
- Measurable: may have trouble with rejected connections
- Acceptable: response time, number of connections, and aggregate bandwidth might not be enough
- Realizable: requires good link; utilization depends on popularity

Is the Web Server SMART?

- Specific: yes
- Measurable: may have trouble with rejected connections
- Acceptable: response time, number of connections, and aggregate bandwidth might not be enough
- Realizable: requires good link; utilization depends on popularity
- Thorough? You decide

Remaining Web Server Issues

- Redundancy: response time is closely related to bandwidth, utilization
- Variability: all measures could vary widely
  - Should we specify variability limits for other than utilization?

Remaining Web Server Issues
- Redundancy: response time is closely related to bandwidth, utilization
- Variability: all measures could vary widely
  - Should we specify variability limits for other than utilization?