The Art of Performance Evaluation

- Given same data, two analyst may draw different conclusions
- Example:
  - Throughput of two systems in transactions per second is as follows:

System	Workload 1	Workload 2
A	20	10
B	10	20

- Three possible ways to compare
  - Compare the average throughput
  - Compute throughputs wrt system B
  - Compute throughputs wrt system A

Example (Cont'd)

- Comparison of Averages

System	Workload 1	Workload 2	Average
A	20	10	15
B	10	20	15

  - Conclusion: two systems are equally good
- Compare the ratio of throughputs with system B as reference

System	Workload 1	Workload 2	Average
A	2	0.5	1.25
B	1	1	1

  - Conclusion: system A is better than B
- Compare the ratio of throughput with system A as reference

System	Workload 1	Workload 2	Average
A	1	1	1
B	0.5	2	1.25

  - Conclusion: system B is better than A

The Art of Perf. Evaluation (Cont'd)

- Similar games can be played in:
  - Selecting the workload
  - Measuring the systems
  - Presenting the results
- Some of these games are played intentionally to create a market "hype" by certain vendors
- What can help?
  - Thorough understanding of systems under test
  - Thorough understanding of workload and its impact on system performance
  - Ability to use various tools to run multiple measurement based experiments and analyze their results
  - Mastering the "science" before practicing the "art" of performance evaluation

Resources for Further Information

- Professional organizations
  - ACM Sigmetrics
  - ACM Sigsim
  - CMG: The Computer Measurement Group, Inc.
  - IFIP: International Federation for Information Processing
  - SIAM: Society for Industrial and Applied Mathematics
- Conferences
  - Sigmetrics
  - CMG
  - PERFORMANCE
  - MASCOTS
- Journals
  - Performance Evaluation Review: quarterly by ACM Sigmetrics
  - CMG Transactions: quarterly by CMG
  - Performance Evaluation: twice a year by Elsevier Science Publishers
  - IEEE Transactions on Software
  - IEEE Transactions on Computers
  - ACM Transactions on Computers

Related Courses

- Later, you may consider taking some of these courses to enhance your background:
  - Stochastic processes
  - Time-series analysis
  - Statistical inference operations research
  - Queuing theory
  - Clustering and pattern recognition
  - Decision theory
  - Simulation

Performance Projects

- Select a system/subsystem
  - Network
    - Hardware: ATM, Ethernet, GigabitEthernet, routers, switches, etc.
    - Software: TCP/IP stacks, tools, and applications
    - QoS: DiffServ, IntServ, IP based multicast, and content netowrking
  - Processor
  - Memory/cache
  - I/O
  - Operating system
  - Server: database, LDAP, web, proxy, streaming, etc.
- Do:
  - Perform measurements
  - Analyze the collected data
  - Simulate
  - Analytically model the subsystem
- Group: up to 2 students per group

Project Suggestions

- Memory subsystem
  - On-chip counter based measurements
  - OS level tools/instrumentation
  - Profiling
- Server performance measurement
  - Web server performance
  - Proxy server performance
  - Streaming media server performance
  - L4 switch performance
- Network QoS measurement
  - Real-time network applications
  - L4 switch performance
  - VoIP server (gateway) performance
- Workload characterization
  - Proxy logs
  - Network traffic logs
  - Accounting logs

Raj Jain's Project Suggestions

- Measure and compare the performance of two microprocessors
- Simulate and compare the performance of two multicomputer interconnection networks
- Characterize the workload on a typical networked workstation in a department
- Characterize the workload of a campus web proxy server
- Measure and analyze the performance of a distributed information system
- Measure and identify the factors that result in memory/cache overhead for a sizeable application
- Develop a software monitor to measure the performance of a distributed system
- Compare several network congestion control algorithms

Chapter 2

Common Mistakes and How To Avoid Them

Common Mistakes

- No goals
  - No general purpose model
  - Goals => techniques, metrics, workload
  - Biased goals
    - “OUR system is better than THEIRS”
  - Analysis without understanding the problem
- Inappropriate experiment design
  - Unsystematic approach
  - Incorrect performance metrics
  - Unrepresentative workload
  - Overlook important parameters and significant factors
  - No sensitivity analysis

Common Mistakes (Cont.)

- Inappropriate level of detail
- No analysis
  - Erroneous analysis
  - No sensitivity analysis
  - Improper treatment of outliers
  - Ignoring errors in input
  - Ignoring variability
  - Too complex analysis
- Improper presentation of results
  - No analysis that can help the decision maker
  - Ignoring social aspects
  - Omitting assumptions and limitations

Checklist for Avoiding Common Mistakes

- Is the system correctly defined and the goals clearly stated?
- Are the goals stated in an unbiased manner?
- Have all the step of analysis followed systematically?
- Is the problem clearly understood before analyzing it?
- Are the performance metrics relevant for this problem?
- Is the workload correct for this problem?
- Is the evaluation technique appropriate?
- Is the list of parameters that affect performance complete?
- Have all parameters that affect performance are chosen as factors to be varied?
- Is the experimental design efficient in terms of time and results?
- Is the level of detail proper?
- Is the measured data presented with analysis and interpretation?
- Is the analysis statistically correct?

Checklist (Cont'd)

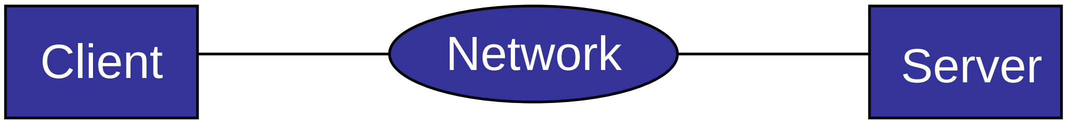
- Has the sensitivity analysis been done?
- Have the outliers in the input or output been treated properly?
- Has the variance of input been taken into account?
- Has the variance of the results been analyzed?
- Is the analysis easy to explain?
- Is the presentation style suitable for its audience?
- Have the results been presented graphically as much as possible?
- Are the assumptions and limitations of the analysis clearly documented?

A Systematic Approach to Perf Evaluation

- State goals and define the system
- List services and outcomes
- Select metrics
- List parameters
- Select factors to study
- Select evaluation technique
- Select workload
- Design experiments
- Analyze and interpret data
- Present results
- Repeat

Case Study: Remote Pipes vs. RPC

- System definition:



- Services:
  - Small data transfer or large data transfer
- Metrics:
  - No errors and failures; correct operation only
  - Rate, time, resource per service
  - Resource = client, server, network

This leads to:

1. Elapsed time per call
2. Maximum call rate per unit of time, or equivalently, the time required to complete a block of n successive calls
3. Local CPU time per call
4. Remote CPU time per call
5. Number of bytes sent on the link per call

Case Study (Cont'd)

- Parameters:
  - System parameters
    - Speed of the local CPU
    - Speed of the remote CPU
    - Speed of the network
    - OS overhead for interfacing with the channels
    - OS overhead for interfacing with the networks
    - Reliability of the network affecting the number of retransmissions
  - Workload parameters
    - Time between successive calls
    - Number and sizes of the call parameters
    - Number and sizes of the results
    - Type of channel
    - Other loads on local and remote CPUs
    - Other loads on the network

Case Study (Cont'd)

- Factors:
  - Type of channel: remote pipes and remote procedure calls
  - Size of the network: short distance and long distance
  - Sizes of the call parameters: small and large
  - Number n of consecutive calls: 1,2,4,8,16,32,...,512, and 1024
  - Notes:
    - Fixed: type of CPUs and OS
    - Ignore retransmissions due to network errors
    - Measure under no other load on the hosts and the network

Case Study (Cont'd)

- Evaluation technique:
  - Prototypes implemented => measurements
  - Use analytical modeling for validation
- Workload:
  - Synthetic program generating the specified types of channel requests
  - Null channel requests => resources used in monitoring and logging
- Experimental design:
  - A full factorial experimental design with 2^3x11 = 88 experiments
- Data analysis:
  - Analysis of variance for the first three factors
  - Regression for number n of successive calls
- Data presentation:
  - The final results will be plotted as a function of the block size n

Chapter 3

Selection of Techniques and Metrics

Selecting an Evaluation Technique

Criterion	Analytical modeling	Simulation	Measurement
Stage	Any	Any	Post-prototype
Time required	Small	Medium	Varies
Tools	Analysts	Computer languages	Instrumentation
Accuracy	Low	Moderate	Varies
Trade-off evaluation	Easy	Moderate	Difficult
Cost	Small	Medium	High
Salability	Low	Medium	High

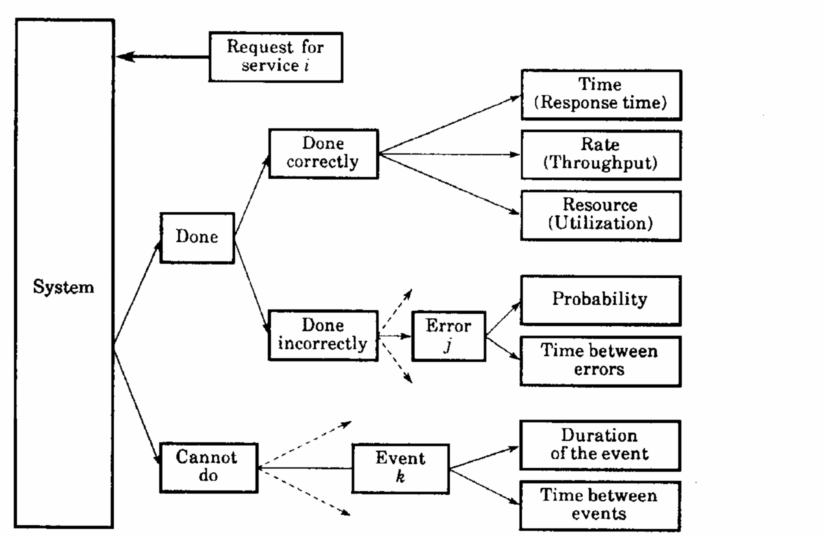
Three Rules of Validation

- Do not trust the results of a simulation mode until they have been validated by analytical modeling or measurements
- Do not trust the results of an analytical model until they have been validated by a simulation model or measurements
- Do not trust the results of a measurement until they have been validated by simulation or analytical modeling

Two or more techniques can be used sequentially

Selecting Metrics

- Include:
  - Performance: time, rate, and resource
  - Error rate/probability
  - Time to failure (MTBF) and duratio
- Consider including:
  - Mean and variance
  - Individual and global
- Selection criteria:
  - Low-variability
  - Non-redundancy
    - Example: Avg. waiting time = queue length x arrival rate
    - Using both waiting time and queue length will be redundant
  - Completeness



Case Study: Two Congestion Control Algs

- Service: send packets from specified source to specified destination in order
- Possible outcomes
    1. Some packets are delivered in order to the correct destination
    2. Some packets are delivered out of order to the destination
    3. Some packets are delivered more than once (duplicates)
    4. Some packets are dropped on the way (lost packets)
- Performance: for packets delivered in order
    1. Response time: the delay inside the network
    2. Throughput: the number of packets per unit of time
    3. Processor time per packet on the source end system
    4. Processor time per packet on the destination end systems
    5. Processor time per packet on the intermediate systems
- Variability of response time => retransmissions
- Out-of-order packets consume buffers => probability of out-of-order arrivals

Case Study (Cont'd)

- Duplicate packets consume the network resource
- Lost packets packets require retransmission => probability of lost packets
- Too much loss cause disconnection =>prob. of disconnect
- Shared resource => fairness
  - Fairness index is given by:

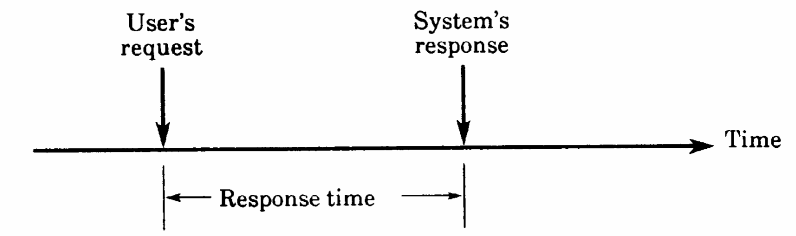
$$f(x_1, x_2, \ldots, x_n) = \frac{\left(\sum_{i=1}^{n} x_i\right)^2}{n \sum_{i=1}^{n} x_i^2}$$

  - Fairness properties:
    - Represents variability of user throughputs
    - Always lies between 0 and 1
    - Equal throughput => fariness = 1
    - If k on n receive x and n-k users receive zero throughput: the fairness index is k/n
- Throughput and delay were found redundant => use power

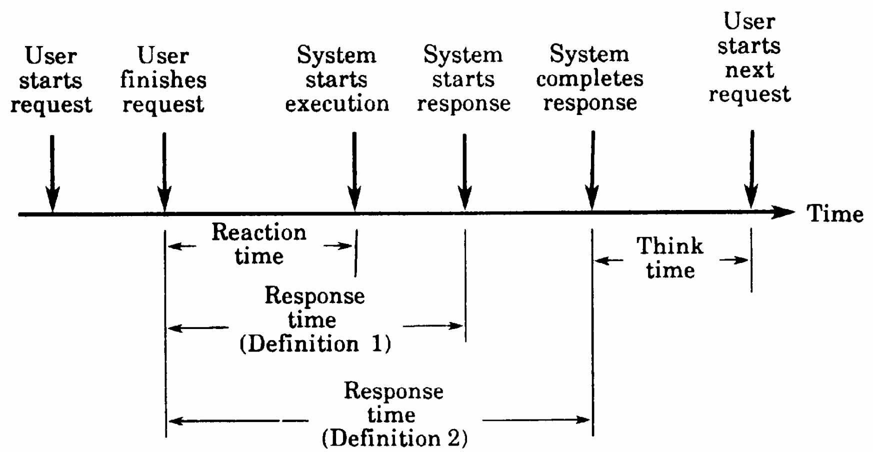
  Power = Throughput / response time
- Variance in response time redundant with the probability of duplication and the probability of disconnection
- Total 9 metrics

Commonly Used Performance Metrics

- Response time and reaction time



(a) Instantaneous request and reponse.



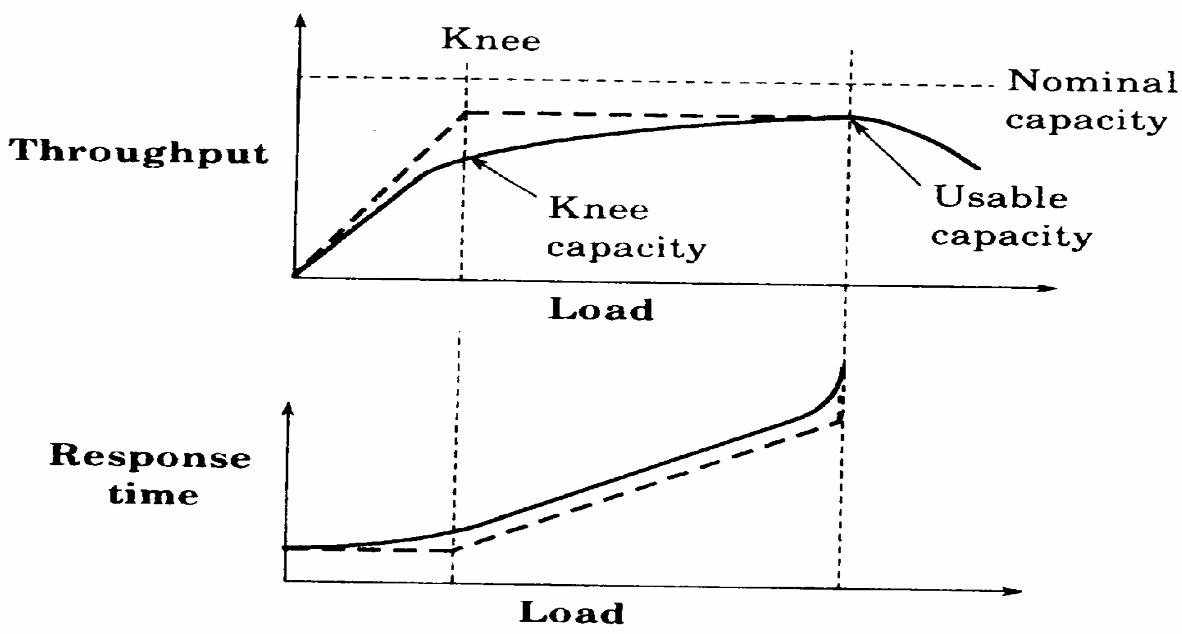
(b) Realistic request and reponse.

Commonly Used Perf Metrics (Cont'd)

- Turnaround time = time between the submission of a batch job and the completion of its output
- Stretch factor: the ratio of the response time with multiprogramming to that without multiprogramming
- Throughput: rate (requests per unit of time)
  - Examples:
    - Jobs per second
    - Requests per second
    - Millions of instructions per second (MIPS)
    - Millions of floating point operations per second (MFLFOPS)
    - Packets per second (PPS)
    - Bits per second (bps)
    - Transactions per seconds (TPS)

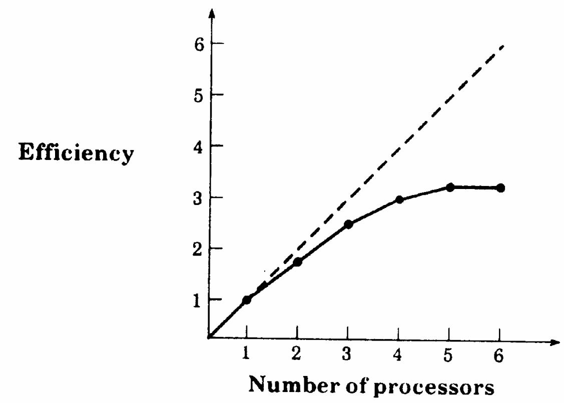
Commonly Used Perf Metrics (Cont'd)

- Capacity:
  - Nominal capacity: Maximum achievable throughput under ideal workload conditions
    - Example: bandwidth in bits per second. The response time at maximum throughput is too high
  - Usable capacity: Maximum throughput achievable without exceeding a prespecified response time limit
  - Knee capacity: Knee = Low response time and high throughput



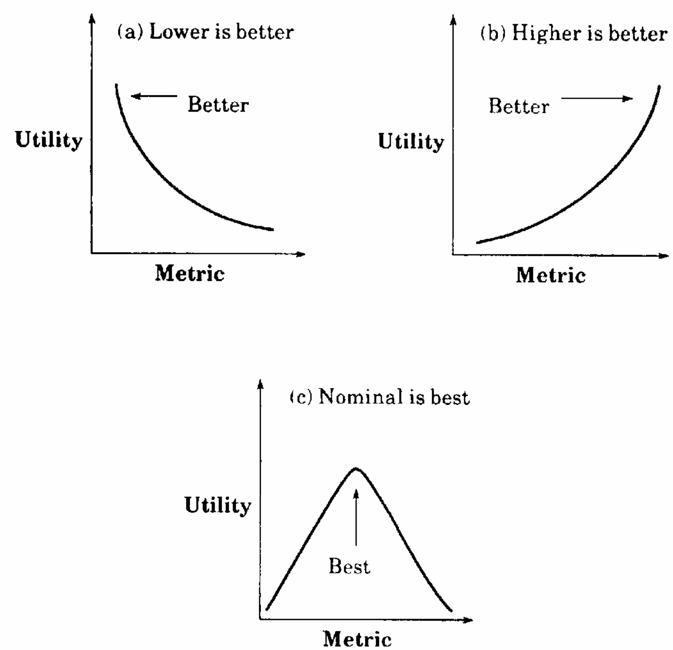
Commonly Used Perf Metrics (Cont'd)

- Efficiency: Ratio of usable capacity to nominal capacity. Or, the ratio of the performance of an n-processor system to that of a one-processor system is its efficiency
- Utilization: The fraction of time the resource is busy servicing requests. Average fraction used for memory
- Reliability:
  - Probability of errors
  - Mean time between errors (error-free seconds)
- Availability:
  - Mean Time to Failure (MTTF)
  - Mean Time to Repair (MTTR)
  - MTTF/(MTTF+MTTR)



Utility Classification of Performance Metrics

- Higher is Better or HB
  - Example: throughput
- Lower is Better or LB
  - Example: response time
- Nominal is best or NB
  - Example: utilization



Setting Performance Requirements

- Examples:
  - The system should be both processing and memory efficient. It should not create excessive overhead
  - There should be an extremely low probability that the network will duplicate a packet, deliver a packet to the wrong destination, or change the data in a packet
- Problems:
  - Non-specific
  - Non-measurable
  - Non-acceptable
  - Non-realizable
  - Non-thorough
- Solution: SMART (specific, measurable, acceptable, realizable, and thorough)

Case Study: Local Area Networks

- Service: send frame to D
- Outcomes:
  - Frame is correctly delivered to D
  - Incorrectly delivered
  - Not delivered at all
- Requirements:
  - Speed
    - The access delay at any station should be less than one second
    - Sustained throughput must be at least 80 Mbits/sec
  - Reliability
    - Five different error modes
    - Different amount of damage
    - Different level of acceptability
  - Availability

Case Study (Cont'd)

- Reliability criterion
  - The probability of any bit being in error must be less than $1x10^{-7}$
  - The probability of any frame being in error (with error indication set) must be less than 1%
  - The probability of a frame in error being delivered without error indication must be less than $1x10^{-15}$
  - The probability of a frame being misdelivered due to an undetected error in the destination address must be less than $1x10^{-18}$
  - The probability of a frame being delivered more than once (duplicate) must be less than $1x10^{-5}$
  - The probability of losing a frame on the LAN (due to all sorts of errors) must be less than 1%

Case Study (Cont'd)

- Availability: Two modes of failure
  - Network reinitialization
  - Permanent failures
- Availability criterion:
  - The mean time to initialize the LAN must be less than 15 msec
  - The time between LAN initializations must be at least one minute
  - The mean time to repair a LAN must be less than one hour (LAN partitions may be operational during this period)
  - The mean time between LAN partitioning must be at least one-half an week