Computer Systems Performance Evaluation

Carey Williamson
Department of Computer Science
University of Calgary

Motivation

- Often in Computer Science you need to:
  - demonstrate that a new concept, technique, or algorithm is <u>feasible</u>
  - demonstrate that a new method is <u>better</u> than an existing method
  - understand the <u>impact</u> of various factors and parameters on the performance, scalability, or robustness of a system (e.g., sensitivity analysis)

Performance Evaluation

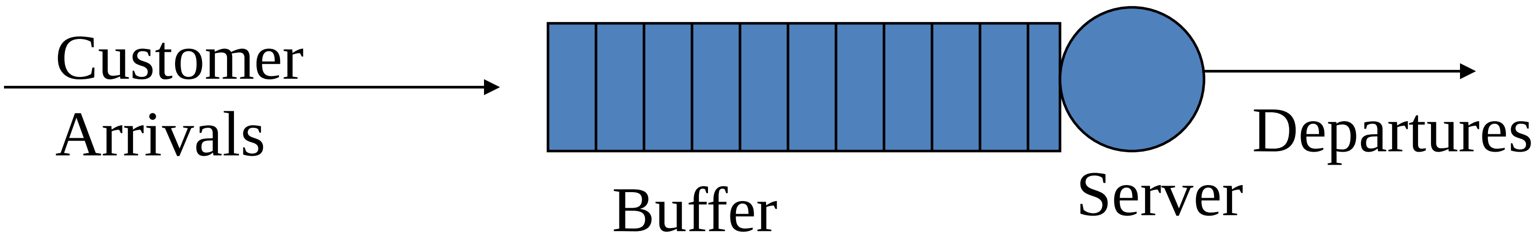
- There is a whole field of computer science called computer systems performance evaluation that is devoted to exactly this issue (e.g., [Ferrari 1978])
- One classic book is Raj Jain's "The Art of Computer Systems Performance Analysis", Wiley & Sons, 1991
- Much of what is outlined in this presentation is described in more detail in [Jain 1991]

Performance Evaluation: An Overview

- There are three main methods used in the design of performance evaluation studies:
- Analytic approaches
  - the use of mathematics, Markov chains, queueing theory, Petri Nets, LP form, Lyapunov optimization,…
- Simulation approaches
  - design and use of computer simulations and simplified models to assess performance
- Experimental approaches
  - measurement and use of a real system

Analytical Example: Queueing Theory

- Queueing theory is a mathematical technique that specializes in the analysis of queues (e.g., customer arrivals at a bank, jobs arriving at CPU, I/O requests arriving at a disk subsystem, requests at a Web server, lineup at Tim Hortons)
- General diagram:



Queueing Theory (cont'd)

- The queueing system is characterized by:
  - Arrival process (M, G)
  - Service time process (M, D, G)
  - Number of servers (1 to infinity)
  - Number of buffers (infinite or finite)
- Example notation: M/M/1, M/D/1
- Example notation: M/M/∞, M/G/1/K

Queueing Theory (cont'd)

- There are well-known mathematical results for the mean waiting time and the number of customers in the system for several simple queueing models
- E.g., M/M/1, M/D/1, M/G/1
- Example: M/M/1
  - $q = \rho / (1 - \rho)$ where $\rho = \lambda/\mu < 1$

Queueing Theory (cont'd)

- These simple models can be cascaded in series and in parallel to create arbitrarily large complicated queueing network models
- Two main types:
  - closed queueing network model (finite population)
  - open queueing network model (infinite population)
- Software packages exist for solving these types of models to determine steady-state performance (e.g., delay, throughput, utilization, occupancy)

Simulation Example: TCP Throughput

- Can use an existing simulation tool, or design and build your own custom simulator
- Example: ns-2 network simulator (or ns-3 now!)
- A discrete-event simulator with detailed TCP protocol models
- Configure network topology and workload
- Run simulation using pseudo-random numbers and produce statistical output

Simulation Issues

- Simulation run length
  - choosing a long enough run time to get statistically meaningful results (equilibrium)
- Simulation start-up effects and end effects
  - deciding how much to "chop off" at the start and end of simulations to get proper results
- Replications
  - ensure repeatability of results, and gain greater statistical confidence in the results given

Experimental Example: Benchmarking

- The design of a performance study requires great care in experimental design and methodology
- Need to identify
  - experimental factors to be tested
  - levels (settings) for these factors
  - performance metrics to be used
  - experimental design to be used

Experimental Factors

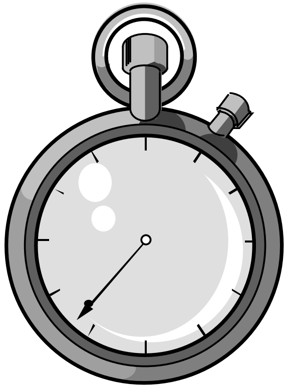
- Factors are the main "components" that are varied in an experiment, in order to understand their impact on performance
- Examples: request rate, request size, response size, number of concurrent clients, read/write ratio
- Need to choose factors properly, since the number of factors affects size of study

Levels for Factors

- Levels are the precise settings of the factors that are to be used in an experiment
- Examples: req size S = 1 KB, 10 KB, 1 MB
- Example: num clients C = 10, 20, 30, 40, 50
- Need to choose levels realistically
- Need to cover useful portion of the design space

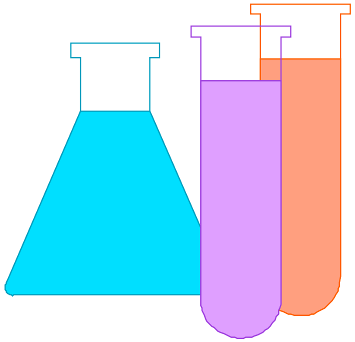
Performance Metrics

- Performance <u>metrics</u> specify what you want to measure in your performance study
- Examples: response time, throughput, packet loss
- Must choose your metrics properly and instrument your experiment accordingly



Experimental Design Methodology

- Experimental design refers to the organizational structure of your experiment
- Need to methodically go through factors and levels to get the full range of experimental results desired
- There are several “classical” approaches to experimental design



Examples of Experimental Design

- One factor at a time
  - vary only one factor through its levels to see what the impact is on performance
- Two factors at a time
  - vary two factors to see not only their individual effects, but also their interaction effects, if any
- Full factorial
  - try every possible combination of factors and levels to see full range of performance results

UNIVERSITY OF CALGARY

Summary

- Computer systems performance evaluation defines standard methods for designing and conducting performance studies
- Great care must be taken in experimental design and methodology if the experiment is to achieve its goal, and if results are to be fully understood
- We will see examples of these methodologies and their applications over the next few months