COMP 190-088: Systems Performance Analysis

Simulations

Jasleen Kaur

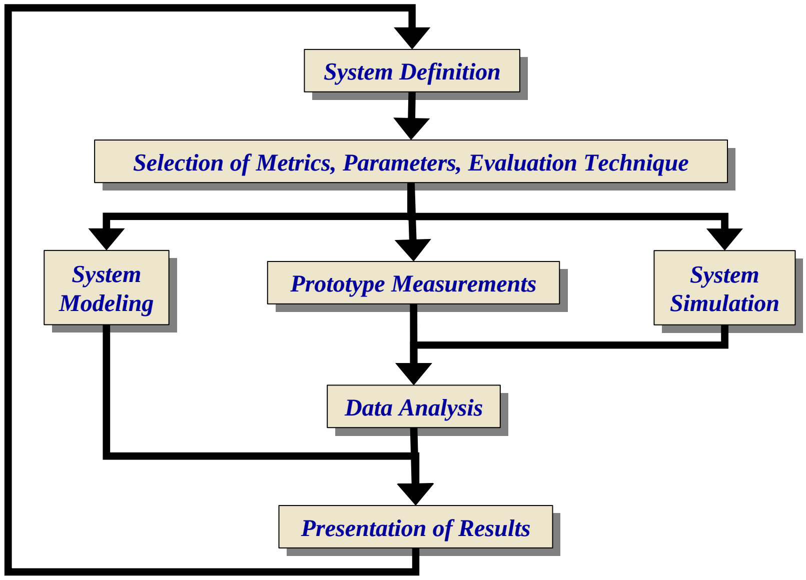
Department of Computer Science
The University of North Carolina at Chapel Hill

Spring 2005

Course Outline

- Selection of metrics
- Performance Evaluation Methodologies
- Workload selection
- Measurements tools
- Analysis and visualization of measured data
- System Modeling
- **Simulations**
- Distributed monitoring infrastructures
- PA in the Research and Industrial communities

Performance Analysis Methodology



Simulation Basics

- **Terminology**
- Types of simulations
- Implementing a simulator

System Simulation

- Basic Idea:
  - Computationally model system state as it varies in response to inputs
- Required skills:
  - Programming skills
    - How should pending events be maintained?
  - Level-of-detail modeling skills
    - At what detail should system interactions be modeled?
  - Experimental design skills
    - How many and what simulation experiments should be run?
  - Statistical analysis skills
    - What confidence should be placed in the results?

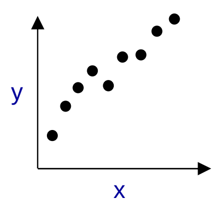
More simulation efforts are terminated due to lack of above skills, or due to incorrect estimation of development time

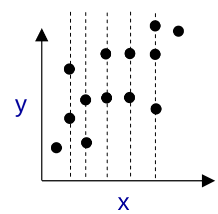
Terminology

- State variables:
  - Define the state of the system
  - eg, length of job queue for CPU performance modeling
- Event:
  - A change in the system state
  - eg, three events in CPU scheduling:
    - Arrival of a job
    - Beginning of a new execution
    - Departure of a job
- Continuous-time (CT) vs. Discrete-time (DT) models:
  - Depending on whether state is defined at all times or specific instants
    - eg, CPU scheduling is CT
    - eg, class attendance for COMP190 is DT

Terminology

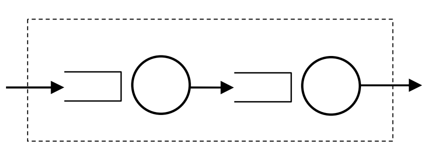
- Continuous-state (CS) vs. Discrete-state (DS) models
  - Depending on whether state variables are continuous or discrete
  - Examples:
    - Time spent by students on COMP190 is CS
    - Queue length in CPU scheduling is DS
  - Continuity of time does not imply continuity of state
- Deterministic vs. Probabilistic models
  - Depending on whether repetitions with same set of input parameters yields same or different results

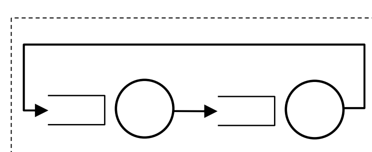




Terminology

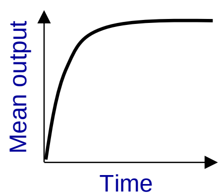
- Static vs. Dynamic models:
  - Depending on whether system state changes with time or not
  - eg, model of matter-to-energy transformation: $E = mc^2$
  - eg, model of CPU scheduling
- Linear and Non-linear models:
  - Depending on whether the output is a linear function of the input
- Open and Closed models
  - Depending on whether input is external and independent of system

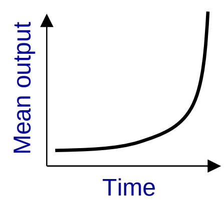




Terminology

- Stable and Unstable models:
  - In stable models, the dynamic behavior of the model settles down to a steady state, that is independent of time
  - In unstable models, system behavior is continuously changing





Simulation Basics

- Terminology
- **Types of simulations**
  - **Emulation**
  - **Monte-Carlo simulations**
  - **Trace-driven simulations**
  - **Discrete-event simulations**
- Implementing a simulator

Emulation

- An emulation is a simulation using hardware or firmware
- Examples:
  - A processor-emulator
    - Emulates the instruction set of one processor on another
  - A network emulator
    - Emulates a real network-link on another
  - Dummynet:
    - Emulates different link capacities
    - Emulates packet loss and delay dynamics
  - Modelnet:
    - Recent emulator from Duke
    - Emulates several network links on the same physical link
    - Enables emulation of large-scale networks

Monte Carlo Simulation

- A static simulation in which time is not modeled
- Requires generation of pseudo random numbers
- Used for:
  - Modeling probabilistic phenomena that do not change characteristics with time
  - Evaluating non-probabilistic expressions using probabilistic methods

Evaluating Expressions With Monte-Carlo Simulations

- Problem: Evaluate the integral: $I = \int_0^2 e^{-x^2} dx$
- Consider:
  - A uniform random variable: X ~ Uniform(0,2)
    - Density function: $f(x) = \frac{1}{2} \quad iff \quad 0 \le x \le 2$
  - A function, y, computed as: $y = 2e^{-x^2}$

$$E(y) = \int_0^2 2e^{-x^2} f(x) dx = \int_0^2 e^{-x^2} dx = I$$

- Integral I can be evaluated by:
  - Generating uniformly distributed: x_i ~ Uniform(0,2)
  - Computing y_i: $y_i = e^{-x_i^2}$
  - Averaging y_i: $I = E(y) = \frac{1}{n} \sum_{i=1}^{n} y_i$

Trace-driven Simulation

- A simulation that uses a trace as its input
- Quite common in computer system analyses
  - Generally used in analyzing resource-management algorithms
    - Simulation models different algorithms
    - Trace of resource demand is used as input to simulation
  - Examples:
    - Paging algorithms
    - Cache analysis
    - CPU scheduling
    - Deadlock prevention algorithms
    - Dynamic allocation of storage
- Traces should be independent of the system under study
  - Trace of: pages fetched from the disk or pages referenced ?

Advantages of Trace-driven Simulations

- Credibility of results
  - Input not generated randomly using an assumed distribution
- Easy validation of simulation model
  - Can collect system performance characteristics along with trace
- Accurate workload
  - Preserves workload characteristics; no simplification required
- Less randomness in output
  - A trace is a deterministic input; less variance in output across runs
- Fair comparison of different alternatives
  - Using same input
- Similarity to actual implementation
  - Good feel for complexity of implementation in reality

Disadvantages of Trace-driven Simulations

- Complexity:
  - A trace-driven simulation requires more system details to be modeled
- Representative-ness:
  - Traces of one system may not be representative of another system
  - Even on a single system, workload may change with time
    - Traces may become obsolete fast
- Finiteness:
  - Results from a small simulated time may not be applicable at all times
- Single point of validation:
  - An algorithm that is best for one trace may not be best for others
- Detail:
  - Takes long time to read each trace line and perform computations
- Little control over workload parameters:
  - Difficult to change workload characteristics in a trace