Performance of Computer System

Evaluation Technique – Simulation

Dr. Lina Xu
lina.xu@ucd.ie

School of Computer Science,
University College Dublin

September 18, 2018

How You Should See the Problem

- What is system?
- Evaluation techniques
- Metrics to use
- Workload
- Parameters

Evaluation techniques

- SIMULATION AND RELATED ISSUES

Why simulation?

- System under study may not be available
  - Common in design and proving stages
- Simulation may be preferred alternative to measurement
  - Controlled study of wider range of workloads and environments
- Higher accuracy results than analytical modeling

Sometime simulation is not preferred

- Accurate simulation models take a long time to develop
  - Typically the evaluation strategy that takes the longest

Evaluation techniques — Simulation

- Easy to modify and update
- Due to the cost of changing system configurations, simulation is preferred beforehand.
- With simulations, it may be possible to identify the optimal combination, but often it is not clear what the trade-off is among different parameters.
- Time consuming for a comprehensive simulation environment development, off the shell simulations tools:
  - Smart phone application simulator/emulator: Android Studio
  - Networking simulator: NS-3, Matlab
  - Transportation simulation: Simio

Good Tips for Evaluation

- Combining evaluation techniques is useful
  - Analytical model: find interesting range of parameters
  - Simulation: study performance within parameter range
- Until validated, all evaluation results are suspect!
  - Always validate one analysis modality with another
  - Beware of counterintuitive results!

Common Mistakes: Too Much Detail

- Level of detail limited only by time available for development
- A detailed model may not be a better model
- Recipe for success
    - Start with less-detailed model
    - Get some results
    - Study sensitivities
    - Introduce details in key areas that affect results most

Common Mistakes: Initial Conditions

- Initial part of a simulation is generally not representative
  - Transient behaviour rather than steady state
- Initial part of simulation should be discarded
  - Several techniques for identifying beginning of steady state

Common Mistakes: Too Short Simulations

- Simulation run times are often very long
- Temptation is to halt simulations ASAP
- However
  - Results may be heavily dependent on initial conditions
  - may not be representative of a real system until steady state
- Correct length for simulations depends on
  - Accuracy desired (width of confidence intervals)
  - Variance of observed quantities

Common Mistakes: Bad Random Numbers

- Bad random numbers can pollute simulation results
  - Period too short
  - Assume global randomness = local randomness
  - Rely on bit subsets: may not be as random as whole
- Use well-known generator rather than rolling your own

Simulation Types

- Monte Carlo Simulation
- Trace-Driven Simulation
- Discrete-Event Simulations

Monte Carlo Simulation

- Model probabilistic phenomenon that do not change over time
- It is used for evaluating nonprobabilistic expressions using probabilistic methods.

Trace-Driven Simulation

- Trace = time ordered record of events on real system
- Advantages
  - Credibility, Easy Validation, Accurate Workload, Detailed Trade-offs, Less Randomness, Fair Comparison, Similarity to the Actual Implementation.
- Disadvantages
  - Complexity, Representativeness, Finiteness, Single Point of Validation, Detail, Trade-off

Trace is different from Statistics

- Statistics (requests, errors, latency, etc.) are calculated based on the full volume of traces
- Examples:
  - Total requests and requests per second
  - Total errors and errors per second
  - Latency
  - Breakdown of time spent by service/type

Discrete-Event Simulations

- Discrete-event simulations use discrete-state model of system, the simulation will have the following components
  - Event Scheduler: A list of events
  - Simulation Clock and a Time-advancing Mechanism: Absolute time and Relevant time
  - System State Variables: Global variables that describe the state of the system
  - Event Routines: Each event is simulated by its routine
  - Input Routines: Input routines typically allow a parameter to be varied in a specified manner
  - Report Generator: At the end of the simulation
  - Initialisation Routines: Initial state of the system state
  - Trace Routines: Print out intermediate variables as the simulation proceeds for debugging
  - Dynamic Memory Management: Normally automatically
  - Main Program: Brings all the routines together

Discrete-Event Simulations

- Mostly used in Computer Sciences
  - Time driven system: auto-sync programs, alert systems, some street or home lighting systems
  - Event driven system: Almost every computer systems

What Simulation to Use

- To model destination address reference patterns in a network traffic, given that the pattern depends upon a large number of factors.
- To model scheduling in a multiprocessor system, given that the request arrivals have a known distribution.
- To determine the value of π

Good Simulations

- Measuring goodness
  - Validation: are assumptions reasonable?
  - Verification: does model implement assumptions correctly?

invalid, unverified	**invalid, verified**
valid, unverified	**valid, verified**

Correctly implements bad assumptions
Incorrectly implements good assumptions
Correctly implements good assumptions

Strategies for Avoiding Bugs – For Validation

- What to check?
  - Assumptions
  - Input parameter values and distributions
  - Output values and conclusions
- How to check?
  - Expert intuition: most common and practical
  - Measurements of real system
  - Theoretical results, e.g. queueing model

Strategies for Avoiding Bugs – For Verification

- Software engineering
  - Top-down design: layered (hierarchical) system structure
  - Modularity: well-defined interfaces, unit testing
- Assertions to check invariants
  - No. packets received = No. packets sent - No. packets lost - No. in flight
- Structured walk through
- Simplified test cases with easily analysed results

Transient Removal

- Transient state: prefix of simulation before steady state
- Steady state performance is usually that of interest
- Heuristic approaches for removing transient state
  - Long runs
  - Proper initialisation
  - Truncation
  - Moving average of independent replications
  - Initial data deletion
  - Batch means

Long runs

- Long run = steady state results long enough to dominate effects of initial transients
- Disadvantages
  - wastes resources (computer time and real time)
  - difficult to ensure length of run is "long enough"
- **Avoid this method if you have other choice**

Proper initialisation

- Proper initialisation = starting simulation in state close to expected steady state
- Examples:
  - start CPU scheduling simulation with non-empty job queue
  - start WWW cache trace-driven simulation with most frequently referenced files in cache

Truncation

- Assumption: variability of steady state ¡ transient state
- Truncation algorithm:

  input: n observations $x_1, x_2, ..., x_n$

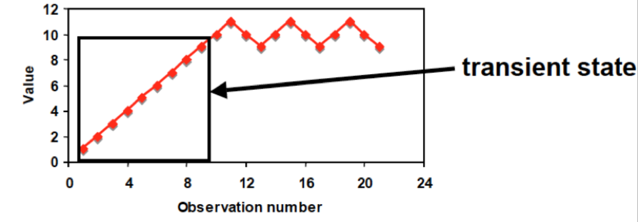
  for k = 2, n

  $min_k = \min(x_k, ..., x_n)$

  $max_k = \max(x_k, ..., x_n)$

  if min_k != x_k && max_k != x_k, break

  post condition: if k != n then k - 1 = length of transient state



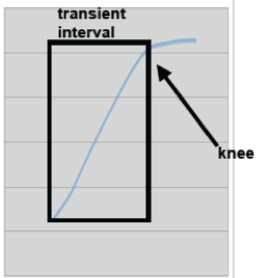
Moving Average of Independent Replications

- Compute mean trajectory by averaging across replications
- Find the knee in the curve.

  for k = 1 to n

  plot trajectory of moving average of successive 2k+1 values

  if trajectory is “sufficiently smooth”, break
- The knee gives the length of the transient phase

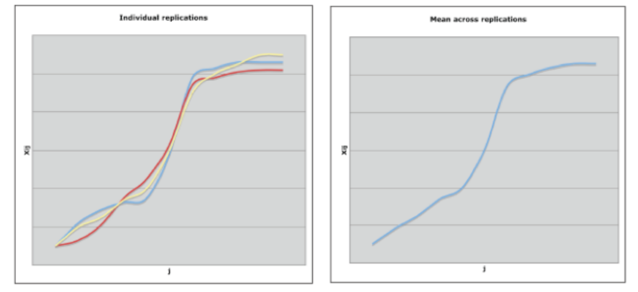


Initial Data Deletion I

- Compute average after some of initial observations omitted
- During steady state average does not change as much as additional observations are deleted
- Problem
  - Randomness in observations causes average to change even in steady state
- Solution
  - Average across several replications

Initial Data Deletion II

1. Compute mean trajectory by averaging across replications
2. Compute overall mean
3. Find knee in a curve showing the relative change in overall mean



Initial Data Deletion III

Find knee in a curve showing the relative change in overall mean

for k = 1, n - 1

assume transient state is of length k

delete first k observations from mean trajectory

compute overall mean from remaining n - k values:

$$\bar{\bar{x}} = \frac{1}{n-k} \sum_{j=k+1}^{n} \bar{x}_j$$

compute relative change in overall mean:

$$\text{Relative change} = \frac{\bar{\bar{x}}_k - \bar{\bar{x}}}{\bar{\bar{x}}}$$

Batch Means

- Run a very long simulation
- Afterward, divide it into several parts of equal duration
- Each part is a batch
- Batch mean = mean of observations in each batch

Evaluation Examples

Criterion	Analytical Modeling	Simulation	Measurement
Stage	any	any	post-prototype
Time required	small	medium	varies
Tools	analysts	programs	instrumentation
Accuracy	low	moderate	varies
Trade-off evaluation	easy	moderate	difficult
Cost	low	medium	high
Saleability	low	medium	high