EEC 686/785
Modeling & Performance Evaluation of Computer Systems

Lecture 19

Wenbing Zhao
Department of Electrical and Computer Engineering
Cleveland State University
wenbing@ieee.org
(based on Dr. Raj Jain's lecture notes)

Outline

- Simulation of a Single-Server Queueing System
- Review of midterm #2

Simulation of a Single-Server Queueing System

- Simulation model
- Performance metrics to be measured
- State variables
- Manual simulation steps

Workload Characterization Techniques

- **Workload characterization**: the process of studying the real-user environments, observe the key characteristics, and develop a workload model that can be used repeated
- The measured workload data consists of services requested or the resource demands of a number of users on the system
- **Workload component** or **workload unit -** the entity that makes the service requests at the SUT interface
- **Workload parameters** or **workload features**
  - Measured quantities, service requests, or resource demands

Workload Characterization Techniques

- Averaging
- Single-parameter histograms
- Multiparameter histograms
- Principal component analysis
- Markov models
- Clustering

Principal Component Analysis

- Key idea: use a weighted sum of parameters to classify the components
- Let x_{ij} denote the *i*th parameter for *j*th component

$$y_j = \sum_{i=1}^{n} w_i x_{ij}$$

- Principal component analysis assigns weights w_i's such that y_j's provide the maximum discrimination among the components
- The quantity y_j is called the **principal factor**
- The factors are ordered. First factor explains the highest percentage of the variance

Markov Models

- Markov => the next request depends only on the last request
- Described by a transition matrix

From/To	CPU	Disk	Terminal
CPU	0.6	0.3	0.1
Disk	0.9	0	0.1
Terminal	1	0	0

- Given the same relative frequency of requests of different types, it is possible to realize the frequency with several different transition matrices

Clustering

- Take a sample, that is, a subset of workload components
- Select workload parameters
- Select a distance measure
- Remove outliers
- Scale all observations
- Perform clustering
- Interpret results
- Change parameters, or number of clusters, and repeat 3-7
- Select representative components from each cluster

Experimental Designs - Terminology

- **Response variables**: outcome
- **Factors**: variables that affect the response variable
- **Levels**: the values that a factor can assume
- **Primary factors** and **secondary factors**
- **Replication**: repetition of all or some experiments
- **Design**: the number of experiments, the factor level and number of replications for each experiment
- **Experimental Unit**
- **Interaction**: effect of one factor depends upon the level of the other

Types of Experimental Designs

- **Simple Designs**: vary one factor at a time
- **Full Factorial Design**: all combinations
- **Fractional Factorial Designs**: use only a fraction of the full factorial design

2^k Factorial Designs

- *k* factors, each at two levels
  - Easy to analyze
  - Helps in sorting out impact of factors
  - Good at the beginning of a study
- Valid only if the effect of a factor is unidirectional, i.e., the performance either continuously decreases or continuously increases as the factor is increased from min to max
  - E.g., memory size, the number of disk drives

Computation of Effects: Sign Table Method

I	A	B	AB	y
1	-1	-1	1	15
1	1	-1	-1	45
1	-1	1	-1	25
1	1	1	1	75
160	80	40	20	Total
40	20	10	5	Total/4

- For a 2^2 design, the effects can be computed easily by preparing a 4×4 sign matrix as shown above
- Next, multiply the entries in column I by those in column y and put their sum under column I; perform similar calculation for other columns
- The sums under each column are divided by 4 to give the corresponding coefficients of the regression model

Allocation of Variation

- Importance of a factor = proportion of the *variation* explained
- Sample Variance of $y = s_y^2 = \frac{\sum_{i=1}^{2^2}(y_i - \bar{y})^2}{2^2 - 1}$
- Variation of $y \triangleq Numerator$

$$= \sum_{i=1}^{2^2}(y_i - \bar{y})^2$$
$$= sum\ of\ squares\ total\ (SST)$$

- For a 2^2 design $SST = 2^2 q_A^2 + 2^2 q_B^2 + 2^2 q_{AB}^2$
- Variation due to A=SSA= $2^2 q_A^2$, etc.

$2^k r$ Factorial Designs with Replication

- r replications of 2^k experiments
  - $2^k r$ observations
  - Allows estimation of experimental errors
- Model: $y = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B + e$
  - e = experimental error
- Computation of effects: use sign table
- Estimation of errors:

$$e_{ij} = y_{ij} - \hat{y}_i = y_{ij} - q_0 + q_A x_{Ai} + q_B x_{Bi} + q_{AB} x_{Ai} x_{Bi}$$

$2^k r$ Factorial Designs with Replication

- Allocation of Variation:

$$SST = \sum_{ij}(y_{ij} - \bar{y}_{..})^2 = 2^2 r q_A^2 + 2^2 r q_B^2 + 2^2 r q_{AB}^2 + \sum_{i,j} e_{ij}^2$$
$$SST = SSA \quad + SSB \quad + SSAB \quad + SSE$$

- Effects are random variables
  - Errors ~ $N(0, \sigma_e) \Rightarrow y \sim N(\bar{y}_{..}, \sigma_e)$
  - q_0 is normal with variance $\sigma_e^2 / (2^2 r)$

Confidence Intervals for Effects

- Variance of errors: $s_e^2 = \frac{1}{2^2(r-1)} \sum_{ij} e_e^2 = \frac{SSE}{2^2(r-1)} \triangleq MSE$
- Estimated variance of q_0: $s_{q_0}^2 = s_e^2 / (2^2 r)$
- Similarly, $s_{q_A} = s_{q_B} = s_{q_{AB}} = s_e \big/ \sqrt{2^2 r}$
- Confidence intervals (CI) for the effects:

$$q_i \mp t_{[1-\alpha/2; 2^2(r-1)]} s_{q_i}$$

- CI does not include a zero => significant

Multiplicative Models for 2^2r Experiments

- **Additive model**. Not valid if effects do not add
- E.g., execution time of workloads
  Execution time $y_{ij} = v_i \times w_j$
  The two effects **multiply**. Logarithm => additive model:
  $$\log(y_{ij}) = \log(v_i) + \log(w_j)$$
- **Correct model**: $y'_{ij} = q_0 + q_A x_A + q_B x_B + q_{AB} x_A x_B + e_{ij}$
  where, $y'_{ij} = \log(y_{ij})$
- Taking an antilog of additive effects *q*. to get the multiplicative effects *u*.

2^{k-p} Fractional Factorial Designs

- Large number of factors
  - Large number of experiments
  - Full factorial design too expensive
  - Use a fractional factorial design
- 2^{k-p} design allows analyzing *k* factors with only 2^{k-p} experiments
  - 2^{k-1} design requires only half as many experiments
  - 2^{k-2} design requires only one quarter of the experiments

Preparing Sign Table for 2^{k-p} Design

- Prepare a sign table for a full factorial design with *k-p* factors
- Mark the first column I
- Mark the next *k-p* columns with the *k-p* factors
- Of the ($2^{k-p} - k + p - 1$) columns on the right, choose *p* columns and mark them with the *p* factors which were not chosen in step 1

Algebra of Confounding

- Given just one confounding, it is possible to list all other confoundings
- Rules:
  - I is treated as unity
  - Any term with a power of 2 is erased

Design Resolution

- Order of an effect = number of factors included in it
- Order of ABCD=4, order of I=0
- Order of a confounding = sum of the order of two terms
  - E.g., AB=CDE is of order 5
- Resolution of a design = minimum of orders of confoundings
- Notation: R_{III} = Resolution III = 2^{k-p}_{III}

One Factor Experiments

- Used to compare alternatives of a single **categorical** variable
  - For example, several processors, several caching schemes
- Model: $y_{ij} = \mu + \alpha_j + e_{ij}$

r = number of replications

y_{ij} = ith response with jth alternative

μ = mean response

α_j = effect of alternative j

e_{ij} = error term

$$\sum \alpha_j = 0$$

Analysis of Variance (ANOVA)

- Importance $\neq$ significance
- **Important** => explains a high percent of variation
- **Significance** => high contribution to the variation compared to that by errors
- Degree of freedom = number of independent values required to compute

$$SSY = SS0 + SSA + SSE$$

$$ar = 1 + (a-1) + a(r-1)$$

Note that the degrees of freedom also add up

F-Test

- Purpose: to check if SSA is *significantly* greater than SSE
  - Errors are normally distributed => SSE and SSA have chi-square distributions
  - The ratio (SSA/ν_A)/ (SSE/ν_e) has an F distribution. Where $\nu_A = a - 1$ = degree of freedom for SSA $\nu_e = a(r - 1)$ = degree of freedom for SSE
  - Computed ratio > $F_{[1-\alpha;\nu_A,\nu_e]}$ => SSA is **significantly** higher than SSE
  - SSA/ν_A is called mean square of A or (MSA). Similarly, MSE = SSE/ ν_e

Simulations - Terminology

- State variables
- Event: a change in the system state
- Continuous-time and discrete-time modes
- Continuous state and discrete state models
- Deterministic and probabilistic models
- Static and dynamic models
- Linear and nonlinear models
- Open and closed models
- Stable and unstable models

Types of Simulations

- Emulation: using hardware or firmware
- Monte Carlo simulation
- Trace-driven simulation
- Discrete event simulation
- Process-oriented simulation

Components of Discrete Event Simulations

- Event scheduler
- Simulation clock and a time-advancing mechanism
- System state variables
- Event routines
- Input routines
- Report generator
- Initialization routines
- Trace routines
- Dynamic memory management
- Main program

Analysis of Simulation Results

- Model verification techniques
  - **Verification** => debugging, correct implementation of the model
- Model validation techniques
  - **Validation** => model = real world, valid assumption
- Transient removal
- Terminating simulations
- Stopping criteria: variance estimation
- Variance reduction

Model Verification Techniques

- Top down modular design
- Antibugging
- Structured walk-through
- Deterministic models
- Run simplified cases
- Trace
- Online graphic displays
- Continuity test
- Degeneracy tests
- Consistency tests
- Seed independence

Model Validation Techniques

- Aspects to validate
  - Assumptions
  - Input parameter values and distributions
  - Output values and conclusions
- Techniques
  - Expert intuition
  - Real system measurements
  - Theoretical results

Transient Removal

- General steady state performance is interesting
  - Remove the initial part. No exact definition => heuristics
- Transient removal methods
  - Long runs
  - Proper initialization
  - Truncation
  - Initial data deletion
  - Moving average of independent replications
  - Batch means

Stopping Criteria: Variance Estimation

- Run until confidence interval is narrow enough

$$\bar{x} \pm z_{1-\alpha/2}\sqrt{Var(\bar{x})}$$

- Independence not applicable to most simulations. Large waiting time for *i*th job => large waiting time for (*i*+1)th job
- For correlated observations: Actual variance » Var(*x*)/*n*
- Solutions:
  - Independent replications
  - Batch means
  - Method of regeneration

Random-Number Generation

- A key step in developing a simulation: have a routine to generate random values for variables with a specified random distribution
  - **Random number generation**: a sequence of random numbers distributed uniformly between 0 and 1
  - **Random variate generation**: the sequence is transformed to produce random values obeying the desired distribution

Terminology

- **Seed**: x_0
- **Pseudo-random**: deterministic yet would pass randomness tests
- **Fully random**: not repeatable
- **Cycle length**, **Tail**, **Period**:

Desired Properties of a Good Generator

- It should be efficiently computable
- The period should be large
- The successive values should be independent and uniformly distributed

Types of Random-Number Generators

- Linear congruential generators
- Tausworthe generators
- Extended Fibonacci generators
- Combined generators

37

Seed Selection

- Do not use zero
- Avoid even values
- Do not subdivide one stream
  - Might lead to strong correlation
- Use non-overlapping streams
- Reuse seeds in successive replications
- Do not use random seeds such as the time of day
  - Can't reproduce. Can't guarantee non-overlap

38

Myths About Random-Number Generation

- A complex set of operations leads to random results
- A single test, such as the chi-square test, is sufficient to test the goodness of a random-number generator
- Random numbers are unpredictable
- Some seeds are better than others
- Accurate implementation is not important
- Bits of successive words generated by a random-number generator are equally randomly distributed

39

Chi-Square Test

- Most commonly used test. Can be used for any distribution
- Prepare a histogram of the observed data
- Compare observed frequencies with theoretical D=…
- D=0 => exact fit
- D has a chi-square distribution with $k-1$ degrees of freedom => compare D with $\chi^2_{[1-\alpha;k-1]}$
  Pass with confidence α if D is less

40

Kolmogorov-Smirnov Test

- Designed for continuous distributions
- Difference between the observed CDF (cumulative distribution function) $F_o(x)$ and the expected cdf $F_e(x)$ should be small
- K^+ = maximum observed deviation above the expected cdf
  K^- = maximum observed deviation below the expected cdf

$$K^+ = \sqrt{n}\max_x(F_o(x) - F_e(x)) \quad K^- = \sqrt{n}\max_x(F_e(x) - F_o(x))$$

- $K^+ < K_{[1-\alpha;n]}$ and $K^- < K_{[1-\alpha;n]}$ => pass at α level of significance
- For $U(0, 1)$: $F_e(x) = x$, $F_o(x) = j/n$,
  where $x > x_1, x_2, \ldots, x_{j-1}$ $\quad K^+ = \sqrt{n}\max_j\left(\frac{j}{n} - x_j\right) \quad K^- = \sqrt{n}\max_j\left(x_j - \frac{j-1}{n}\right)$

Chi-Squire vs. K-S Test

K-S Test	Chi-Square Test
Small samples	Large sample
Continuous distributions	Discrete distributions
Differences between observed and expected CDFs	Differences between observed and hypothesized probabilities (pdfs or pmfs)
Use each observation in the sample without any grouping => makes a better use of the data	Groups observations into a small number of cells
Cell size is not a problem	Cell sizes affect the conclusion but no firm guidelines
Exact	Approximate

k-Dimensional Uniformity or *k*-Distribution

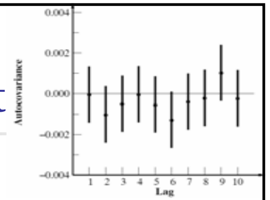
- ***k*-distributed** if:

$$P(a_1 \le u_{n-1} < b_1, \ldots, a_k \le u_{n+k-1} < b_k) = (b_1 - a_1)\cdots(b_k - a_k)$$

For all choices of a_i, b_i in [0, 1), with $b_i > a_i$, i=1,2,…, k.

- k-distributed sequence is always (k – 1)-distributed. The inverse is not true
- Two tests:
  - Serial test
  - Spectral test

Serial-Correlation Test

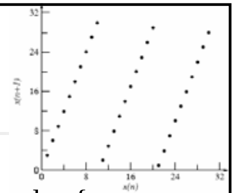


- Nonzero covariance => dependence
  - The inverse is not true
- R_k = autocovariance at lag k = Cov[x_n, x_{n+k}]
- For large n, R_k is normal distributed with a mean of zero and a variance of 1/[144(n-k)]
- 100(1-α)% confidence interval for the autocovariance is

$$R_k \mp z_{1-\alpha/2} / (12\sqrt{n-k})$$

- Check if CI includes zero

Spectral Test



- **Goal**: to determine how densely the k-tuples $\{x_1, x_2, \ldots, x_n\}$ can fill up the k-dimensional hyperspace
- The k-tuples from an LCG all on a finite number of parallel hyperplanes
- Successive pairs would lie on a finite number of lines
- In three dimensions, successive triplets lie on a finite number of planes
- **Spectral test**: determine the max distance between adjacent hyperplanes
  - Larger distance => worse generator

Random-Variate Generation

- General techniques
  - Inverse Transformation
  - Rejection
  - Composition
  - Convolution
  - Characterization

How to Select a Random-Variate Generation Technique

Commonly Used Distributions

- Bernoulli, binomial, negative binomial, Poisson, geometric, Pascal, normal
- Gamma, Erlang, Exponential, Beta, uniform, Pareto, χ^2, Cachy, Lognormal, t, F, Weibull