Computer Systems Performance Analysis and Benchmarking (37-235)

Analytic Modelling

Simulation

Measurements / Benchmarking

Lecture/Assignments/Projects:

Dipl. Inf. Ing. Christian Kurmann

Textbook:

Raj Jain, “The Art of Computer Systems Performance Analysis”, 1991 Wiley & Sons, New York

Topic of Today:

- **A systematic approach to Evaluation**
- **Evaluation Techniques (Selection)**
- **Workloads**

A Systematic Approach

- **State goals and define the system**

  Given same hardware and software, definition varies depending upon the goals.

  Example:

  1. Given: two CPUs
     Goal: impact on response time of interactive users

  Wanted: better timesharing system (external component)

  2. Given: two CPUs with different ALUs
     Goal: which ALU

  Wanted: best implementation of an architecture (internal component)

  What are the boundaries?

- **List services and outcomes**

  Services:

  - Network transports user packets
  - Database system responds to queries

  Outcomes:

  - Packets are lost or delayed.
  - Database queries are answered wrongly or hung due to a deadlock.

  Define what is acceptable - what not?

  The list of services and outcomes helps later in selecting the metrics and workloads.

- **Select Metrics:**

  Criteria to compare performance
  - Speed
  - Accuracy
  - Availability of services

- **List parameters**

  System parameters:
  Software and hardware characteristics which generally do not vary among various installations of the system.

  Workload parameters:
  Characteristics of the user's requests, that typically vary from one installation to another.

  Keep the list as comprehensive as possible!

- **Select factors**

  Factors are the parameters that vary during evaluation, their values are called **levels**.

  Parameters with high impact on systems performance should be factors.

- **Select evaluation technique**
  - analytic modeling
  - simulation
  - measuring a real system
- **Select workload**
  - Analytic model:

    Probability/Distribution of requests.
  - Simulation:

    Traces of requests.
  - Measurement:

    User scripts, sample problem.

- **Design an experiment**

  Sequence of steps (simulations, measurements) that offer maximal information, maximal coverage with minimal effort.

  Fractional factorial experimental design.

  Two phases:
  1. many factors, few levels
  2. reduced factors, increased levels

- **Analyze and interpret data**
  - Deal with randomness and variability.
  - Interpret the results of the analysis.
  - Prepare to draw conclusions.

- **Present data properly**
  - Spreadsheets, charts and graphics
  - no statistical jargon

The complete project consists of several cycles through theses steps

Case Study

Compare **remote messaging** with **remote method invocation**.

Remote messaging does not block the caller, callee runs concurrently, the results (if any) are later returned asynchronously.

Remote method invocations are like procedure calls, the caller waits until the callee returns.

1. System Definition
2. Services
3. Metrics
4. Parameters
5. Factors
6. Evaluation Technique
7. Workload
8. Experimental Design
9. Data Analysis
10. Data Presentation

Selection of Technique

Criterion	Modeling	Simulation	Measurement
1. Stage	Any	Any	Postprototype
2. Time required	Small	Medium	Varies
3. Tools	Analysts	Computer Lang	Instrumentation
4. Accuracy	Low	Moderate	Varies
5. Trade-off Evaluation	Easy	Moderate	Difficult
6. Cost	Small	Medium	High
7. Saleability	Low	Medium	High

Key consideration:

- life-cycle, is system already available

Further considerations:

- available time, yesterday?
- availability of tools, skills
- accuracy level, not identical to correct-ness of conclusion!
- find opt. parameters or compare diffe-rent alternatives,
- cost
- saleability, justifies the expense

Rules of validation

- Do not trust the results of a simulation model 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 simulation or measurements.
- Do not trust the results of a measurement until they have been validated by simulation or analytical modeling.
- ;-)
- Especially rule 3 should be emphasized
- At least expert intuition
- Sometimes it is a good idea to use two techniques.

Selection of Metrics

Three possible outcomes of service requests:

- done correctly - **speed metric**:
  Time (Responsiveness), Rate (Productivity), Resource (Utilization)
- done incorrectly - **reliability metric**:
  Probability, Time between errors
- refused to do - **availability metric**:
  Duration of event, Time between events

Common Metrics:
response time, reaction time, throughput (MIPS, pps), capacity (bandwidth), efficiency (%), utilization, reliability, availability (MTTF), cost/performance ratio...

Workloads

- **test workload**
  - workload used in a performance evaluation
  - real or synthetic
- **real workload**
  - normal operations
  - often not suitable for test
  - not reproduceable
- **synthetic workload**
  - characteristic similar to real
  - controlled
  - repeatedly applied
  - built in measurement capabilities
  - no sensitive data used

Types of test workloads

- Addition Instruction
- Instruction mixes
- Kernels
- Synthetic programs
- Application benchmarks

Addition Instruction

very frequent instruction

good pick

early days of computing

no longer sufficient

Instruction mix

Gibson Mix 1959

TABLE 4.1 Gibson Instruction Mix

1. Load and Store	31.2
2. Fixed-Point Add and Subtract	6.1
3. Compares	3.8
4. Branches	16.6
5. Floating Add and Subtract	6.9
6. Floating Multiply	3.8
7. Floating Divide	1.5
8. Fixed-Point Multiply	0.6
9. Fixed-Point Divide	0.2
10. Shifting	4.4
11. Logical, And, Or	1.6
12. Instructions not using registers	5.3
13. Indexing	18.0
	100.0

IBM 704 and IBM 650 (dinosaurs!)

- Average instruction time
- Clocks per instruction
- Directly MIPS/MFLOPS

Problem: Pipelining, caching, address translation.

Kernels

Specific functions like:

- Scalar Product of two Vectors (BLAS1)
- Matrix Vector Multiply (BLAS2)
- Sieve, Puzzle
- Tree Search
- Ackermann
- Matrix Inversion
- Sorting (NAS sort)

Synthetic Programs

see example in the book...

- Used in a microbenchmarks
- Exercises one or two parameter spaces

Application Benchmarks

Specific applications for:

- Banking
- Airline reservation
- large scientific codes

  includes everything
  - hardware
  - input/output
  - networks
  - operating system
  - databases

  e.g. debit credit benchmark