Advanced Systems Lab
Tutorial III
Planning Experiments

G. Alonso
Systems Group
http://www.systems.ethz.ch
Why Experiments?
Quantitative questions about systems

• Absolute or comparative performance analysis
  – How many operations can a system run per second? How long does an operation take?
  – How many concurrent clients does a system support?
  – Do SSDs make an application faster than hard disks?
  – Should I use quick sort instead of merge sort for an online catalogue?
  – Where is the bottleneck in the system?
How to answer such questions?

• Experiments
  – You implement / install „system(s) under test“ (SUT)
  – You run benchmarks and measure observable results

• Modeling
  – You build a model of the „system(s) under test“
  – You calculate results with model

• Simulation
  – You implement a system that behaves like SUT
  – You run benchmarks and measure computed results
Experiments vs. Modeling

- Experiments
  - Often expensive to implement
  - Specific to environment (e.g., hardware used)
  - Accurate (quantitative) results
  - Sometimes misleading

- Modeling
  - Typically cheap
  - General
  - Qualitative results
  - You always learn something

- Use modeling whenever you can
  - Unfortunately, modern systems are too complex
Methodology

1. Ask the right question
   – Define the „system(s) under test“
   – Define what to measure and understand why
   – Define relevant workloads, understand parameters

2. Make a hypothesis
   – „A good scientist predicts the results and explains later why something totally different happened."

3. Carry out experiment (real system, model)
   – Run workloads, measure metrics

4. Report results, analyze results, gotoStep 1
   – Give answer to question, possibly refine question
Making a Hypothesis

• Use the same format as the final results
  – Draw graphs with expected results
  – Even try to predict variance and statistical properties
  – Make bullet points with explanations
  – Use „modeling“ to make hypothesis

• Share hypothesis with your customer
  – Validates whether you are asking the right question
  – i.e., can you make decisions if results turn out like that

• Comparison of expected vs. real results
  – Essential to find bugs in your experiments
  – Essential to understand real results
Data distributions
Accuracy vs. Precision

Accuracy = how close to the real value (often unknown)
Precision = similarity of the results of repeated experiments
Dealing with Accuracy

• Understand the System under Test

• Understand the environment in which the tests are being done:
  – Identify potential interference and sources of noise
  – Characterize those sources and try to estimate their value

• Correct (if possible and meaningful) the measured values to improve accuracy
Dealing with Precision

• When measuring, we are trying to estimate the value of a given parameter

• The value of the parameter is often determined by a complex combination of many effects and is typically not a constant

• Thus, the parameter we are trying to measure can be seen as a RANDOM VARIABLE

• The assumption is that this random variable has a NORMAL (GAUSSIAN) DISTRIBUTION
Central limit theorem

- Let $X_1, X_2, X_3, ... X_n$ be a sequence of independently and identically distributed random variables with finite values of
  - Expectation ($\mu$)
  - Variance ($\sigma^2$)

as the sample size $n$ increases, the distribution of the sample average of the $n$ random variables approaches the normal distribution with a mean $\mu$ and variance $\sigma^2/n$ regardless of the shape of the original distribution.
How does it work?

[Images showing the Central Limit Theorem progression: original, twice, three times, four times]

Normal or Gaussian distribution

http://en.wikipedia.org/wiki/Normal_distribution
The standard deviation gives an idea of how close are the measurements to the mean
– important in defining service guarantees
– important to understand real behavior

Mean is 50 but standard deviation is 20, indicating that there is a large variation in the value of the parameter
Mean of a sample

- To interpret a given measurement, we need to provide complete information
  - The mean
  - The standard deviation around the mean

Mean and standard deviation

• The standard deviation defines margins around the mean:
  – 64% of the values are within $\mu \pm \sigma$
  – 95% of the values are within $\mu \pm 2\sigma$
  – 99.7% of the values are within $\mu \pm 3\sigma$

• For a real system is very important to understand what happens when the values go beyond those margins (delays, overload, thrashing, system crash, etc.)
Calculating the standard deviation

• Mean and standard deviation:

\[ \mu = \frac{\sum_{i=1}^{N} x_i}{N} \]
\[ \sigma = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)^2} \]

• In practice, use:

\[ s = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (x_i - \bar{x})^2} \]
Comparisons

• What is better?

Deterministic behavior is often more important than good performance
In practice

• In many systems, the standard deviation is almost more important than the mean:
  – 90% of the queries need to be answered in less than X seconds
  – No web request can take longer than 5 seconds
  – Changes have to be propagated in less than 10 seconds
  – Guaranteed bandwidth higher than X 90% of the time

• Achieving determinism is often done at the cost of performance
Percentile Example

Query Latency in msec

Query Load in Queries/sec
Amazon Example (~2004)

• Amazon lost about 1% of shopping baskets
  – Acceptable because incremental cost of IT infrastructure to secure all shopping baskets much higher than 1% of the revenue

• Some day, somebody discovered that they lost the *largest* 1% of the shopping baskets
  – Not okay because those are the premium customers and they never come back
  – Result in much more than 1% of the revenue

• Be careful with correlations within results!!!
Putting it all together
Look at all the data

• Make sure you are looking at the complete picture of the experiment and your measurements do not include side effects (warm up, cool down, repetition effects)

• Once you are sure you have identified the valid data and that it looks reasonable, then apply statistics to it
Standard deviation

• All measurements and graphs have to be accompanied by the standard deviation, otherwise they are meaningless
  – Provides an idea of the precision
  – Provides an idea of what will happen in practice
  – Provides an idea of how predictable performance is

• Repeat the experiments until you get a reasonable standard deviation (enough values are close enough to the mean)
Advice

• It is a good idea to run a long experiment to make sure you have seen all possible behavior of the system:
  – Glitches only every 3 hours
  – Memory leaks after 1 M transactions

• In reality, tests have to resemble how the system will be used in practice
Designing an experiment
Experiments, but which ones?

• What does it mean to design an experiment?
  • Performance is affected by a large number of factors
    – Workloads
    – Systems
    – Knobs

• We are interested in:
  – Which ones are the most important?
  – Which ones are related?

• Goal: get the most information with least effort (minimum number of experiments)
A **response variable** is the outcome of an experiment - typically the measured performance of the system (e.g., throughput, response time)

A **factor** is any variable that affects the response, and which has several alternatives (amount of memory, number of cores, data sizes)

**Levels** are the values that a given factor can assume – the alternatives for a factor.

**Primary factors** are those whose effects need to be quantified

**Secondary factors** are those that impact performance but whose effect we are not interested in quantifying

**Replications** are the number of times each experiment is to be repeated with particular levels for each factor.
An experiment

• An experimental design consists of:
  – the number of different experiments
  – the factor level combinations for each experiment
  – the number of replications of each experiment

• An experimental unit is any entity used for the experiment
Interaction

• Two factors interact if the effect of one depends on the level of the other.
• Interaction considerably complicates the business of interpreting experimental results

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>$B_2$</td>
<td>6</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>$A_1$</th>
<th>$A_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$B_1$</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>$B_2$</td>
<td>6</td>
<td>9</td>
</tr>
</tbody>
</table>
Avoid mistakes

• Try to avoid the following:
  – Ignoring the variation due to experimental errors
  – Not controlling important parameters (secondary factors)
  – Not isolating the effects of different factors
  – Overly simple (and very inefficient designs)
  – Ignoring interactions between factors
  – Conducting too many experiments
    • Take it slowly!
    • Break up the project into steps
Exploring the space

• Given a number of factors, what to do?

• Bad idea:
  – Vary one factor at a time
  – Find best value, fix it
  – Repeat for each factor

• Why is this a bad idea?: too many experiments, will get stuck in local minimum
$2^k$ Factorial Designs

• Experimental technique to find the relative weight of different factors
  – Pick K factors
  – Pick two levels for each factor
  – Behavior of factors must be unidirectional or monotonic in the range explored (!)
Example $2^2$ Factorial Design

Observation: can update book examples by multiplying by 1000!

<table>
<thead>
<tr>
<th>Cache size (MB)</th>
<th>Memory size</th>
<th>4GB</th>
<th>16GB</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>15</td>
<td>45</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>25</td>
<td>75</td>
</tr>
</tbody>
</table>

Define variables $x_A$ and $x_B$ to represent levels for each factor:

$$x_A = \begin{cases} 
-1 & \text{if 4 GB main memory;} \\
1 & \text{if 16 GB main memory.} 
\end{cases}$$

$$x_B = \begin{cases} 
-1 & \text{if 1 MB cache,} \\
1 & \text{if 2 MB cache.} 
\end{cases}$$
Solving the model

A useful fiction: non-linear regression model for performance:

\[ y = q_0 + q_{A \times A} + q_{B \times B} + q_{A B \times A \times B} \]

This means we can write:

\[ 15 = q_0 - q_A - q_B + q_{AB} \]
\[ 45 = q_0 + q_A - q_B - q_{AB} \]
\[ 25 = q_0 - q_A + q_B - q_{AB} \]
\[ 75 = q_0 + q_A + q_B + q_{AB} \]
Relative weights on response variable

Solving:

\[ y = 40 + 20x_A + 10x_B + 5x_Ax_B \]

What does this mean?

- Mean performance is 40
- Effect of memory is 20
- Effect of cache is 10
- Interaction between the two accounts for 5
In the form of a table

The same calculation can be done using a sign table:

<table>
<thead>
<tr>
<th>l</th>
<th>A</th>
<th>B</th>
<th>AB</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>1</td>
<td>15</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>45</td>
</tr>
<tr>
<td>1</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>25</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>75</td>
</tr>
<tr>
<td></td>
<td>160</td>
<td>80</td>
<td>40</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>20</td>
<td>10</td>
<td>5</td>
</tr>
</tbody>
</table>
Repetitions and errors

• Look in the book
  – How to allocate variation
  – How to consider repetitions of the experiments to look for errors

• For the report, please use repetitions to get meaningful results.