Performance Evaluation

Tutorial

Felix Rath, Mirko Stoffers

Outline

Outline

Audience

You are

- A current thesis writer at Comsys
- A Hiwi at Comsys
- For some other reason interested in performance evaluation

You have

- Designed some kind of “system”
- Implemented this “system” (or you are currently doing this)

You want to

- Evaluate your system

Why Evaluation?

Maybe…

- …to show how great my tool is?
- …to get a degree?

No!

Rather:

- To evaluate the performance of my tool
  - Possible results: good, …, bad
  - The result is not fixed beforehand

Why Evaluation?

But my tool is so great!

- It's always better than everything else!
  - Ok, go for it, do the formal proof!
- Well ... ok ... it might have some disadvantages...
  - Then show them
  - Makes your evaluation more credible

A good evaluation

- Shows the advantages
- Shows the (obvious) disadvantages

Methods to Evaluate

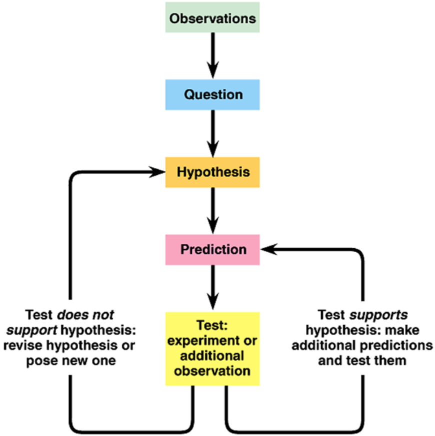
- Descriptive: Textual description of pros and cons
- Analytical: Mathematical results
- Simulation: Results obtained from running and abstract model
- Emulation: Combination of simulation and testbed
- Testbed: Measurement on physical hardware

In the following

- Measurements in simulation, emulation, or testbeds

Outline

The Scientific Method



- **Observation:** We (your supervisor) found a problem in state-of-the-art
- **Question:** Can we solve it?
- **Hypothesis:** Maybe we can solve the problem by applying ...
  <insert your thesis here>
- **Prediction:** If I set these parameters, it will...
- **Test:** ... break :(

Steps in a Performance Evaluation Study

6 steps to come up with a good performance evaluation study

1. Problem formulation and system definition
2. Choice of metrics, factors, and levels
3. Data collection
4. Implementation and verification
5. Validation
6. Experimentation, analysis, and presentation

Problem Formulation / System Definition

- Define the goal of the study and the system under study
- Goals of a study need to be unbiased
  - The goal of study must be unknown
  - There must not be a preferred outcome
  - Any outcome must be acceptable
  - Do not abuse an investigation to show something
- Definition of the system under study
  - What belongs to the system?
  - Maybe more important: What does not belong to the system?

Example Goals

- Compare idea A with idea B
- Not: Idea A is better than idea B, isn't it?

Choice of Metrics, Factors, and Levels

Definitions

- A *metric* is a measure for system performance (“output”)
  - Throughput, delay, jitter, runtime, speedup, ...
- A *factor* is a parameter of a system that is modified (“input”)
  - Amount of traffic, network topology, available bandwith, ...
- *Levels* are the considered numerical values for the factors
  - 10 MBit/s, 1 GBit/s, star, bus, 10 nodes, 10000 nodes, ...

Choosing the “correct” metrics, factors, and levels is crucial!

Data Collection

Understanding the system

- Do you believe in every result your evaluation pops out?
- There might be obvious relationships:
  - The datarate over that 1000Base-X link can't be greater than 1 GBit/s
- There might be relationships that need more investigation:
  - Battery lifetime of this sensor node can't be greater than 42 hours when CPU is busy at least 10 % of the time
- Go and understand your system!
- Read research papers
- Find those relationships and write them down
- Define validation scenarios!

Implementation and Verification

- Now implement your stuff
- And verify the implementation
  - Assure that the code does what it should do

Validation

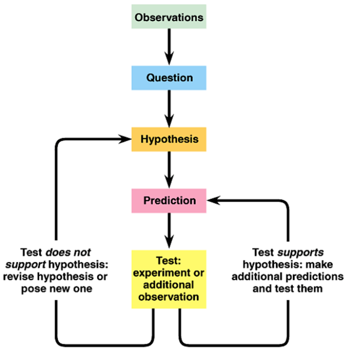
- Assure that the results are correct
- You did define validation scenarios, didn't you?
- Now, assure that your code really behaves as expected
- Run it with the inputs you defined before
- Check whether the results are correct

Verification vs. Validation

- Verification: My code does, what I want it to do
- Validation: The "what I want it to do" makes sense

Experimentation, Analysis, Presentation

- Now we can run the experiments
- Follow the scientific method
- Analyze the results carefully
- Finally, present them accurately
  - More details later



And then?

- Iterate!
- This will take time!

Steps in a Performance Evaluation Study

6 steps to come up with a good performance evaluation study

1. Problem formulation and system definition
2. Choice of metrics, factors, and levels
3. Data collection
4. Implementation and verification
5. Validation
6. Experimentation, analysis, and presentation

Outline

How to do Measurements

The situation

- You have defined everything
- You know how you want to set up your system
- You know what you want to measure (metrics)

The Question: How do I measure now?

How (not) to do Measurements

First shot

- I just let it run, and measure the results

Are you sure you get a similar result next time?

Second shot

- I let it run 5 times, and take the average

Does this allow for comparisons?

Example

(bigger numbers are better)

System A

- Average: 6
- Measurements: 3, 8, 4, 8, 7

System B

- Average: 7
- Measurements: 6, 1, 7, 14, 7

So: System B is better than System A!

Or was it just bad luck?

And another Problem...

Example: A queuing system at a checkout

- Want to know average queuing time at a supermarket at 10am
- Setup: Simulation of the checkout as a queuing system
- Measure queuing time of each customer
- Take the average

Where is the problem?

- Queuing time of first customer is 0
- Queuing time of second customer is rather short
- …

Initial transient vs. steady state!

And one more Problem...

Example: A queuing system at a checkout

- Want to know average queuing time at a supermarket at 10am
- Setup: Simulation of the checkout as a queuing system
- Start the simulation at 9 am
- Measure queuing time of each customer from 9.55 to 10.05
- Take the average

Where is the problem?

- Results: 35 s, 40 s, 38 s, 45 s, 50 s, ...
- So the average is: 42 s

This is not the average over independent measurements!

Summary of the Problems

You have to care for:

- Initial transient vs. steady state
- Getting independent values
- Taking the average doesn’t suffice to compare results

Initial Transient

Ways to cope with initial transient

- Start measurements after end of initial transient
  - Need to know where it ends
- Run “long enough” such that the effect gets negligable
  - How long is “long enough”?
- Initialize correctly
  - What does correctly mean?
- Statistical methods (e. g., batch means)
  - Statistical way that tells you the end of the transient
  - Ask your supervisor / me for details
- Also care for the end of your experiments
  - Stop measurements on time!

Getting Independent Values

How to get independent values

- Do independent runs
  - On real experiments: make sure they are really independent
  - When using pseudo random numbers (e. g., in simulation): use different seeds (truely random, e. g., from `http://www.fourmilab.ch/hotbits/`)
- Average over “long enough” batches
  - Batch means helps again
  - Compute average over each batch
  - Compute autocorrelation
  - Increase batch size if autocorrelation too big
  - Ask your supervisor / me for details

Doing Comparisons

You always want to do comparisons

- Need "i.i.d." random variables (see Central Limit Theorem!)
- Independent: see above
- Identically distributed
  - Don't mix different configurations
  - Careful with experiments: avg SINR between 11 and 12am not identically distributed with avg SINR between 3 and 4am!

With "i.i.d." random variables you can

- Compute the average
- Compute the confidence interval

What is a confidence interval?

Confidence Interval

True mean vs. empircal mean

- The average of N runs yields an empirical mean
- Example: throw a die 100 times, mean: 3.47
- But what is the true mean?
- For the die: 3.5
- In general: we don't know!

What is a confidence interval?

- An interval around the average
- Has a parameter (defined by you), e. g., 99 %
- Meaning: true average in this interval with confidence of 99 %
- For the die this could be: $[3.41; 3.53]$

Comparing two Fair Dice

Without confidence intervals

- First die average over 100 runs: 3.47
- Second die average over 100 runs: 3.51

Interpretation: The second die yields bigger results!

With confidence intervals

- First die 99 % confidence interval: [3.41; 3.53]
- Second die 99 % confidence interval: [3.43; 3.59]

Interpretation: We don't know which die is better!

Computing Confidence Intervals

Steps to compute confidence intervals

1. Define the confidence level
2. Make sure you have enough samples, and they are i.i.d.
3. Why? Take a look at the Central Limit Theorem!
4. Compute the average $\bar{x}$
5. Compute standard deviation σ
6. Compute lower endpoint of confidence interval: $\bar{x} - Z \cdot \frac{\sigma}{\sqrt{n}}$
7. Compute upper endpoint of confidence interval: $\bar{x} + Z \cdot \frac{\sigma}{\sqrt{n}}$
   - n: number of samples
   - Z: depends on confidence level
     - 95 %: 1.96
     - 99 %: 2.58
     - More values on the Internet ;)

Things to take care of

My confidence intervals are too big

- So you can't compare your values?
- Do more measurements
- Decrease confidence level (don't go below 95 %!)
- Well, maybe you just can't...

How many samples?

- Target for 30 (actually the theory only holds for ∞)
- If you don't have enough time: at the very least: 5
- If your intervals are too big, increase n!

Summary of the Problems

You have to care for:

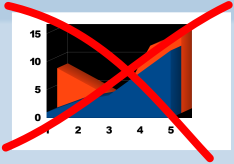
- Initial transient vs. steady state
- Getting independent values
- Taking the average doesn't suffice to compare results

Outline

How to Present your Results

General rules

- Scientific – not fancy!
- Results must be clearly observable
- Show the results, don't hide them!
- Don't include more data in a single graph than possible
- Rather make more plots



Elements of a graph

- x-axis: parameters / different configurations
- y-axis: metric(s)
- Must include
  - labels, tick labels, units, legend, confidence intervals!

Creating Plots

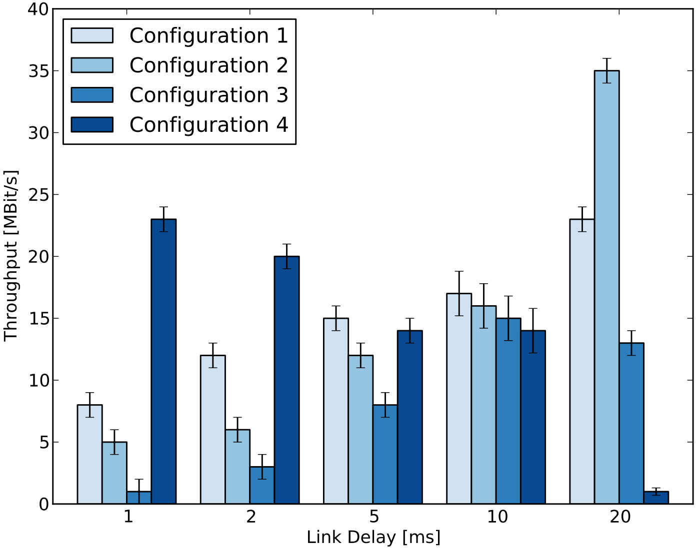
Ask yourself

- What do you want to show?
- What graph type makes sense?
- How can I arrange the data?

Tools to use

- Matplotlib: Python package (you can do everything with your data that Python can)
- Gnuplot / GLE: Script languages for plots (can also do some automation)
- ~~Excel / OpenOffice (automation? confidence intervals? non-ugly graphs? at least: make vector graphics!)~~

Example Plot



Discussing Results

Describe the evaluation setup

- Introduce all details of your setup
- Tell them about your configurations
- Make sure that the reader believes that the chosen configurations make sense
- and that they are not chosen “to prove that your system is better”

Describe the evaluation results

- Describe the structure of the plots
- Describe the meaning of the numbers

Discussing Results

Discuss the evaluation results

- Most important part
- Discuss the general (expected) effects
- Discuss the unexpected effects
- Point to the configurations where your system is better
- But also show the configurations where your system is worse
- and discuss why

In general

- Stick to a clear structure
- Make sure the reader gets it
- Use proper methodology, discuss proper methodology

Summary

What is important?

- Follow scientic method
- Care for initial transient
- Care for statistical confidence
- Discuss and present your results properly
- Further reading and more links:
  `http://www.sigplan.org/Resources/EmpiricalEvaluation/`

On questions

- Ask now (if general)
- Ask your supervisor (if more special)

Slides available at /projects/tutorials/perfeval.git

```
git clone login.comsys.rwth-aachen.de:/projects/tutorials/perfeval.git
```