Advanced Systems Lab

G. Alonso, D. Kossmann
Systems Group
http://www.systems.ethz.ch

ADMINISTRATION

Overview of the Course

- Lecture: Tuesdays, 5-7pm, CAB G61
  - cover material from the text book
  - ~9 sessions (probably done mid November)
- Project Work (40% of the grade; 80% of the effort)
  - done in groups of 3 students
  - Java programming
  - Meetings with supervisors, Thursdays, 5-7pm, CAB/CHN
  - FIRST MEETING: NEXT THURSDAY!!! -> CAB G11, not in the usual rooms!
- Exam (60% of the grade)
  - Sessionsprüfung, details to be announced
- Web page

  http://www.systems.ethz.ch/courses/fall2012/ASL

Literature

- Raj Jain: The Art of Computer Systems Performance Analysis, John Wiley &Sons
  - (we will not cover Part V „Simulation“)
  - Library has copies. Nevertheless, worth buying.
  - Exam based on book chapters and book exercises
- Almost all „Systems“ publications have examples
  - Networks, operating systems, databases, ...
  - Master and PhD theses of Systems Group
  - ACM SIGMETRICS: specialized on performance eval.

Requirements

- Things we assume you know:
  - Programming
  - Basic statistics and probability theory
  - Operating systems (threads, scheduling, memory management)
  - Databases (SQL)
  - Networks (RPC, TCP/IP, routing, sockets)
- If you do not have the necessary background, you need to acquire it before taking this course

Project for this course

- Three milestones
  * Milestone 1: Build System
    - persistent messaging system

  * Milestone 2: Performance Experiments
    - study performance of the system you have implemented under various parameters

  * Milestone 3: Modeling
    - estimate/model performance of the system you have implemented based on queueing models

Objective of this course

- Quantitative evaluation of computer systems
- Learn how to answer questions of the form...
  - What is better: A or B?, Is A good enough?
  - What are the limits of A?, How can I improve A?
- A, B, C, ... are computer systems
  - software + hardware
  - often only components of a bigger system
- Almost always the answer is: „It depends!“
  - That is what makes performance eval. interesting.

Goals

- Things you will know at the end of the course
  - System building principles
  - Experimental design and analysis
  - Answering quantitative questions about systems
  - Experiments presentation
  - Queuing theory applied to system performance analysis
  - Insights on the systems used in the project

THE COURSE

Schedule

- Intro
- Basic concepts
- Metrics and workloads
- Workloads
- Review of statistics
- Queuing theory

Quantitative questions about systems

- In scope questions (although not well specified)
  - Can we move our web server to the Amazon cloud?
  - Are SSDs better than hard disks?
  - Should I use quick sort instead of merge sort for my online catalogue?
  - Is MySQL fast enough for my online catalogue?
- Out of scope questions
  - Is a product better than another?
  - Feature analysis
  - Costs (e.g., $ per tpcm)

How do you 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 whenver you can
  - Unfortunately, modern systems are too complex

Methodology

1. Ask the right question
   - Define the „system(s) under test“
   - Define what „better“ means
   - 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

Methodology

1. Ask the right question
   - Define the „system(s) under test“
   - Def eans
   - 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

Difficult

On vegetarians (from Wikipedia)

A 1999 metastudy combined data from five studies from western countries.[68] The metastudy reported mortality ratios, where lower numbers indicated fewer deaths, for fish eaters to be 0.82, vegetarians to be 0.84, occasional meat eaters to be 0.84. Regular meat eaters and vegans shared the highest mortality ratio of 1.00. The study reported the numbers of deaths in each category, and expected error ranges for each ratio, and adjustments made to the data.

However, the "lower mortality was due largely to the relatively low prevalence of smoking in these [vegetarian] cohorts"

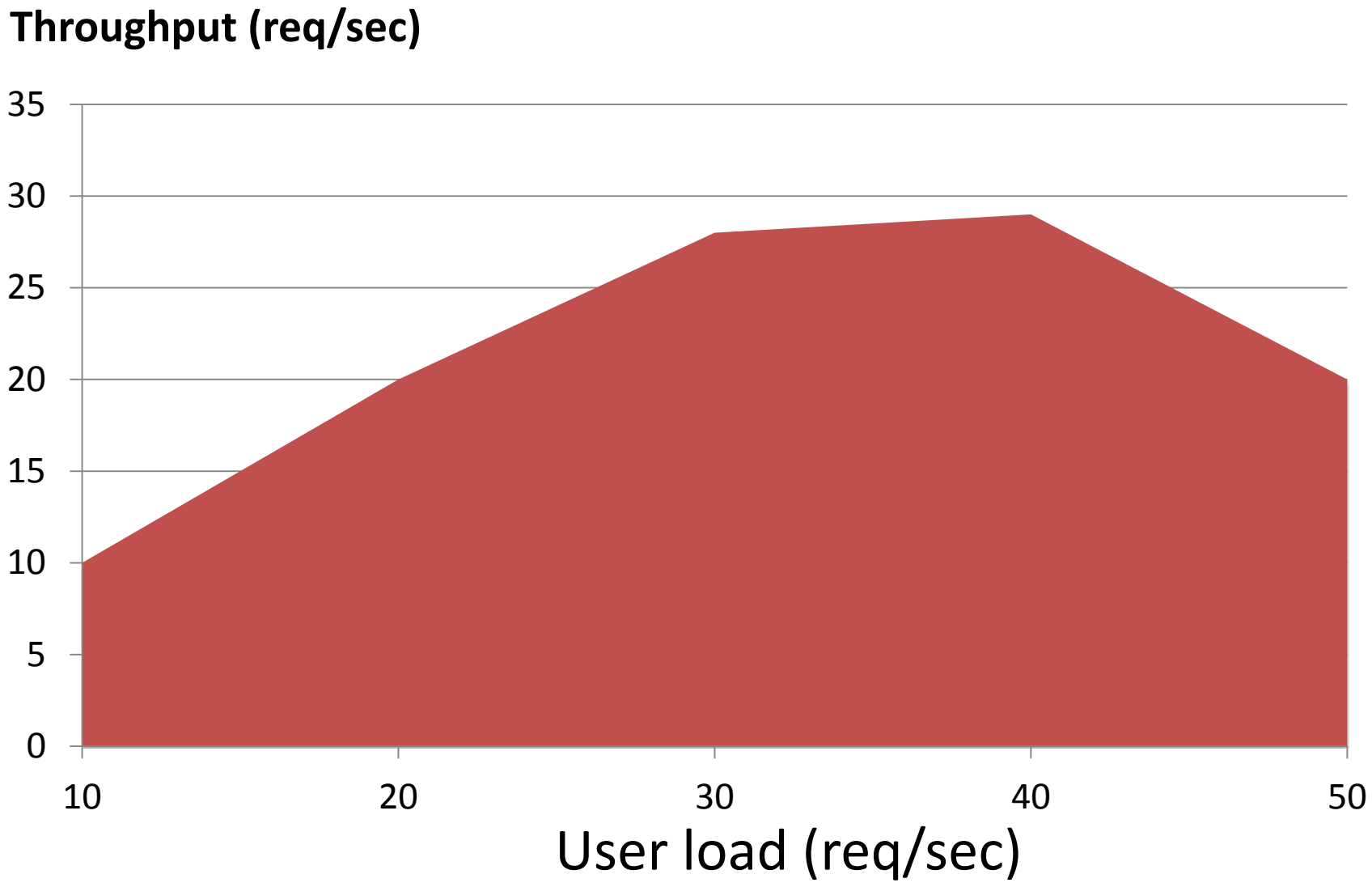
Making a Hypothesis

- 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 (advisor)
  - 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

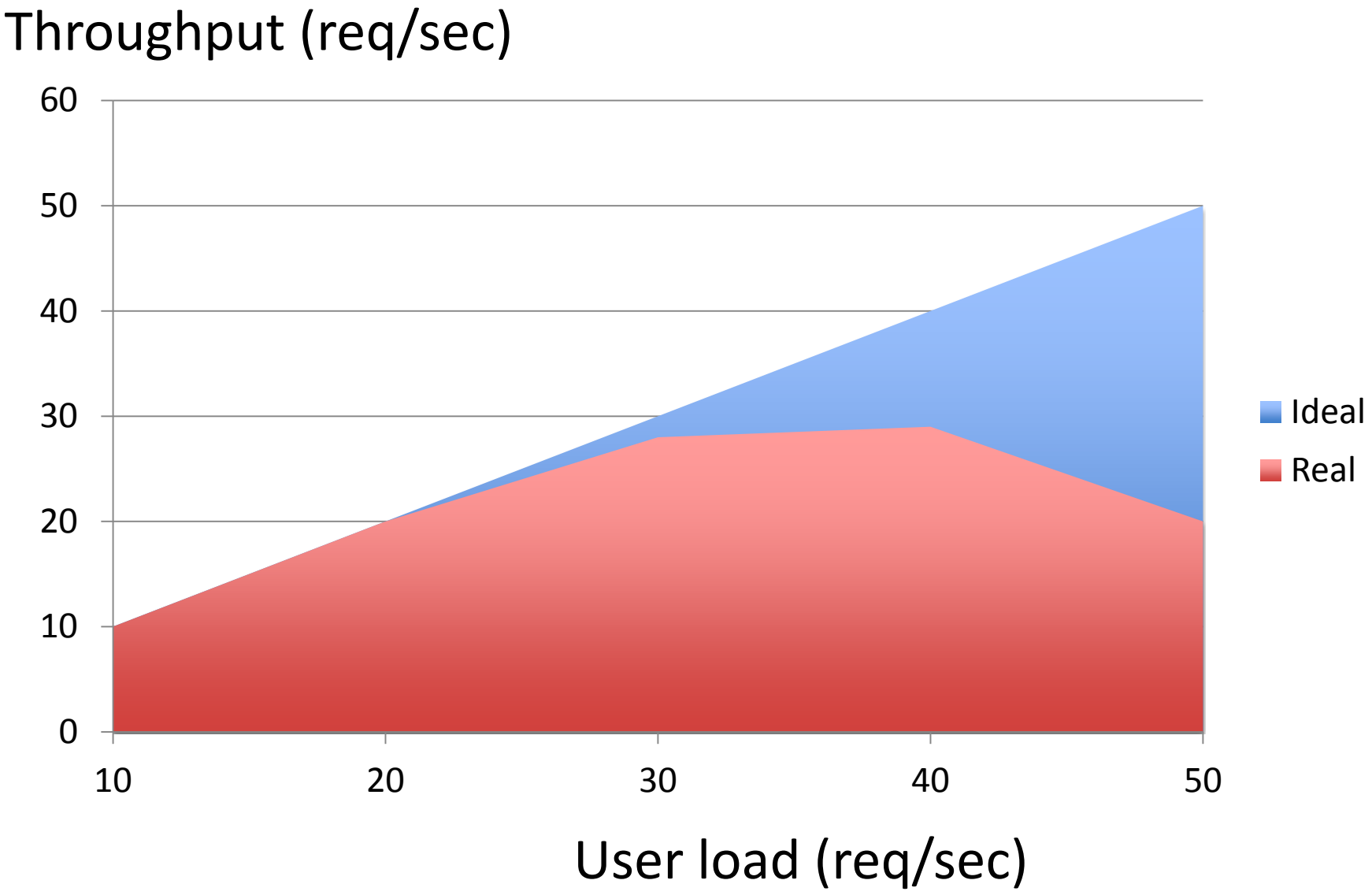
Example

- Metric 1: Throughput (y-axis of graphs)
  - requests completed per unit of time (secs)
  - count only “successful” requests (no error, < timeout)
- Metric 2: Response Time (y-axis of graphs)
  - max/min/avg time (secs) to complete a request
- Parameter: User Load (x-axis of graphs)
  - number of requests arriving per unit of time (secs)

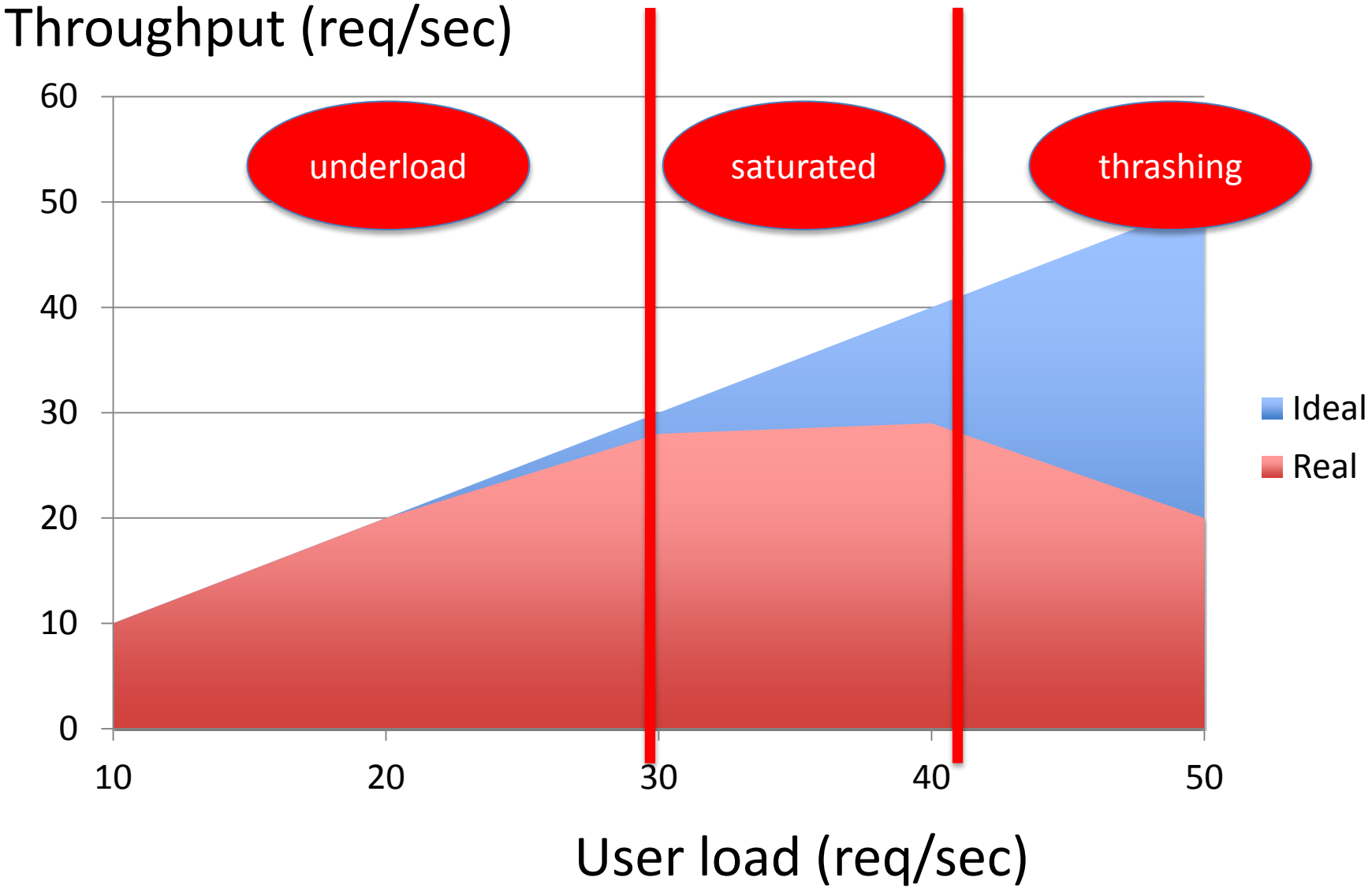
Example: Throughput



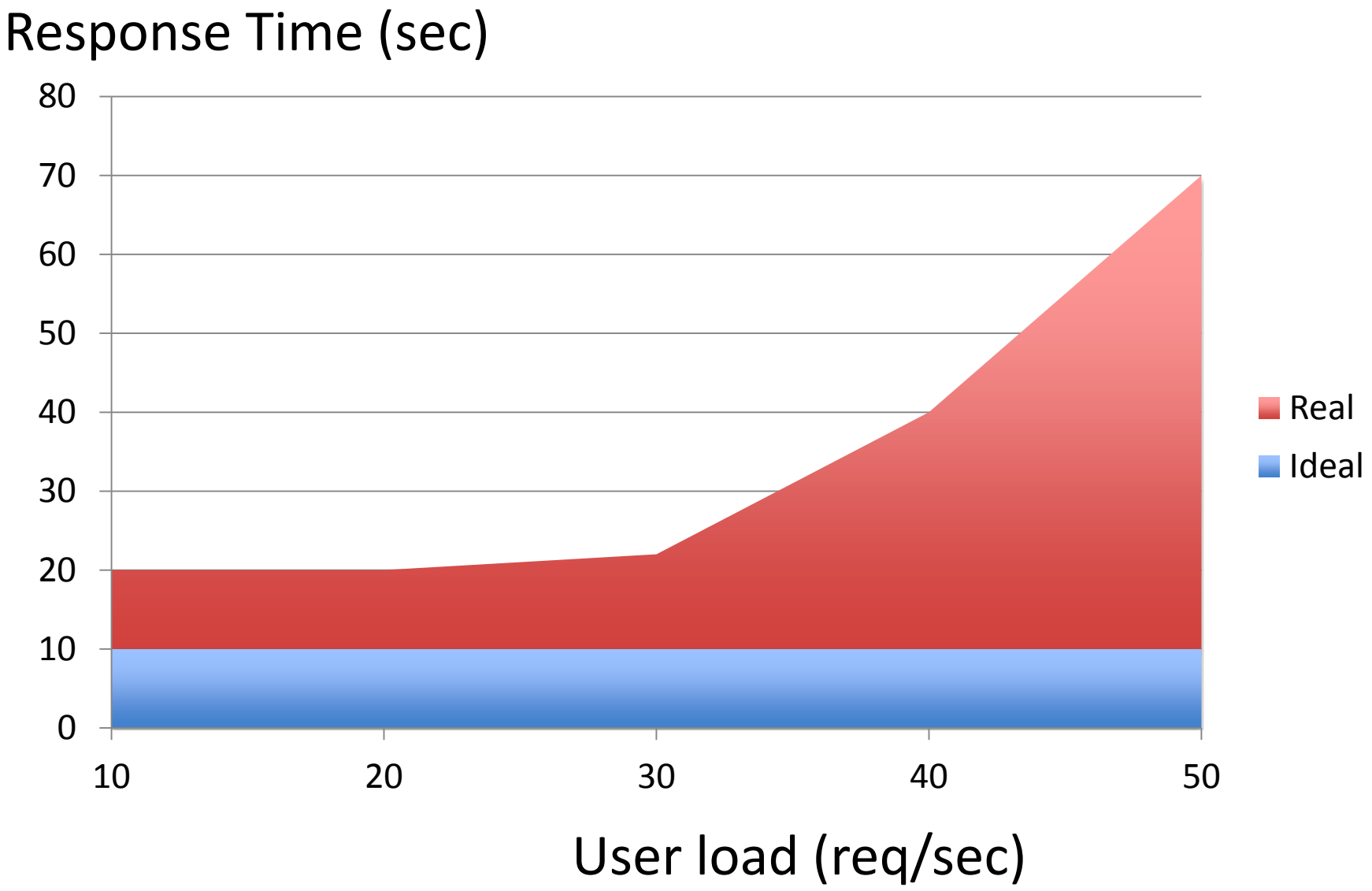
Example: Throughput



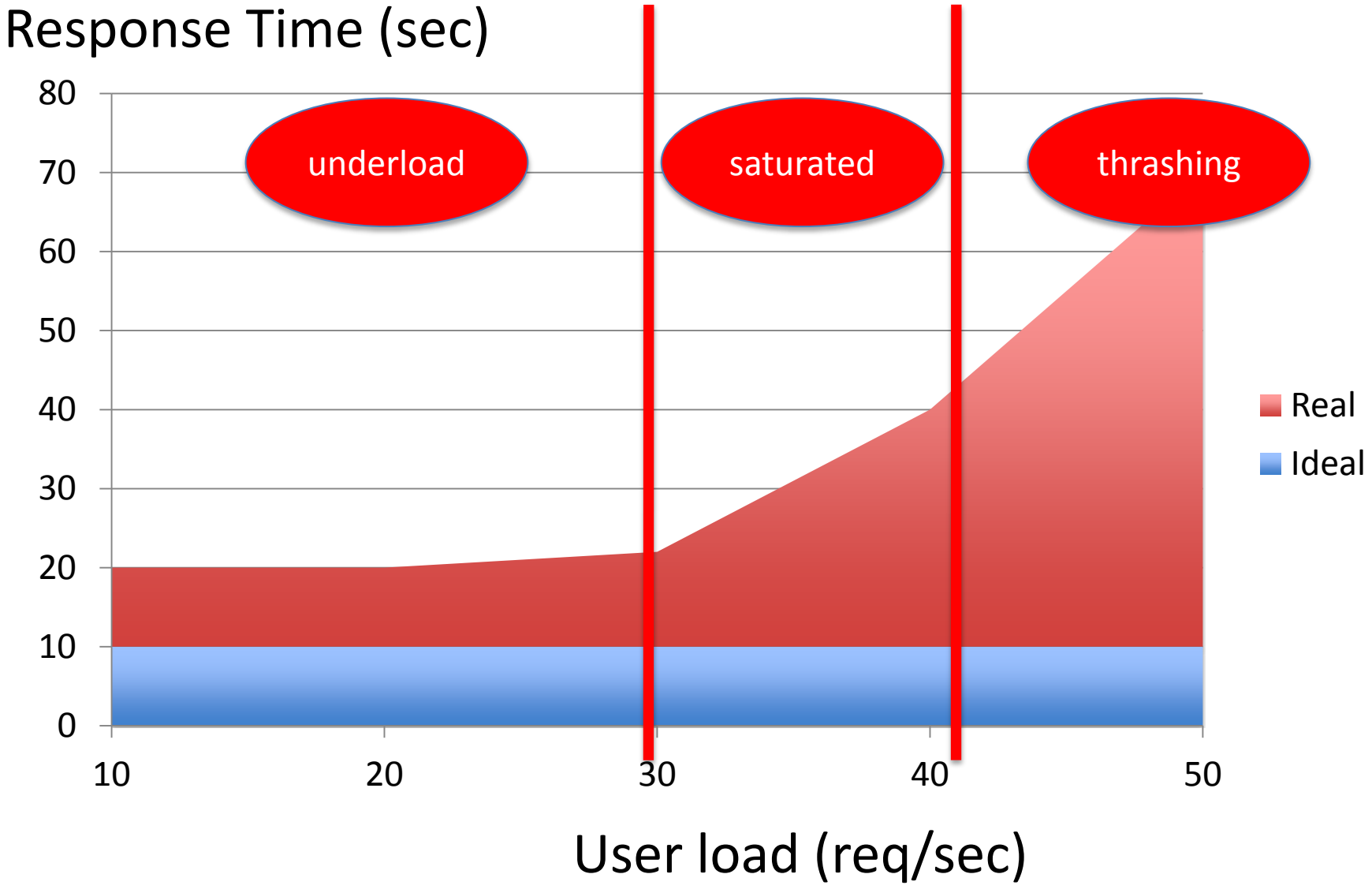
Example: Throughput Analysis



An Example: Avg. Response Time



An Example: Avg. Response Time



The most dangerous profession

- In a study in 1685 of the ages and professions of deceased men, it was found that the profession with the lowest average age of death was “student.”
- Why does being a student appear to be so dangerous?

http://www.stat.columbia.edu/~gelman/bag-of-tricks/chap10.pdf

How to lie with statistics

Absolute Performance

System	Workload 1 (tps)	Workload 2 (tps)	Average (tps)
System A	20	10	15
System B	10	20	15

Relative Performance: Baseline B

System	Workload 1 (rel.)	Workload 2 (rel.)	Average
System A	2x	0.5x	1.25x
System B	1x	1x	1x

Ratio games

- Using ratios instead of direct measurements allows in some cases to reach arbitrary conclusions (see book, chapter 11)
- Changing the weights of different workloads can be used to make a system look better
- Inventing ratio measurements can make a system look better than it is
- …

Example ratio games with %

- System A (total tests passed 20.6%)

Test #	Total	Passed	Passed in %
1	300	60	20 %
2	50	2	4 %

- System B (total tests passed 9%)

Test #	Total	Passed	Passed in %
1	32	8	25 %
2	500	40	8 %

How to run experiments

- Understand the subject area
  - E.g., lectures on „systems“
- Learn your math (lectures, pencil & paper)
  - Statistics
  - Modeling (Queueing Models)
  - Example designs
- Experience, experience, experience
  - Project as part of this class
  - Lab projects, Master thesis, ...
- Common sense ☺