UNIVERSITY OF CAMBRIDGE

P51(bis): High Performance Networked-Systems

Prof. Andrew W. Moore

With copious thanks to Dr Noa Zilberman (Oxford), Richard Sites (formerly DEC, HP, MSR, Google), and many others whom I've bounced off ideas.

Finally, a **huge** thank you to Eben Upton, Raspberry Pi Foundation, and PiHut people for enabling this incarnation of the module at incredibly short notice.

An explanation

This subject is not as advertised:

- Popularity + Equipment constraints meant refactoring

Things that are the same:

- Hands on
- Performance-centered
- Deadlines are similar (one intermediate submission around week 5/6 and one end of term submission) 20% and 80% respectively.

Thanks that are better:

- More than Networks: Systems *and* Networks; together.
- Prerequisites: Undergraduate courses in digital communication, good working knowledge of C/C++, ~~ECAD~~, **Unix (**look at `https://www.cl.cam.ac.uk/teaching/current/UnixTools/)`
- All work is assessed as individual (no team submissions); however, collaborations are fine*

Things that are uncertain:

- How things will go. Regardless of planning, there are always issues.

* Always credit all work of others.

Administrivia

Scope:

- Understanding high-performance networked-systems

Course structure:

- Lectures – 6 x 1 hour – Tuesday P51 SW02: (Wks 1,2,4,5,6,7)

- Lab time – 5 x 2 hours – Friday P51 SW02 (Friday 15:00-17:00)

  P51 SW02: (Wks 2,4,5,6,7) - We skip next Friday

Assessment:

- Lab writeup (20%) – 23/02/2023 12:00
- Principle Assignment (80%) – 16/03/2023 12:00

Some logistics for Michaelmas 2022-2023

Web page: http://www.cl.cam.ac.uk/teaching/current/P51/

Repository: https://github.com/cucl-srg/P51a/ NOTE THE ‘a’. Work in progress

Moodle: *https://www.vle.cam.ac.uk/course/view.php?id=245002*

Grades:

MPhil (ACS) – Mark out of 100

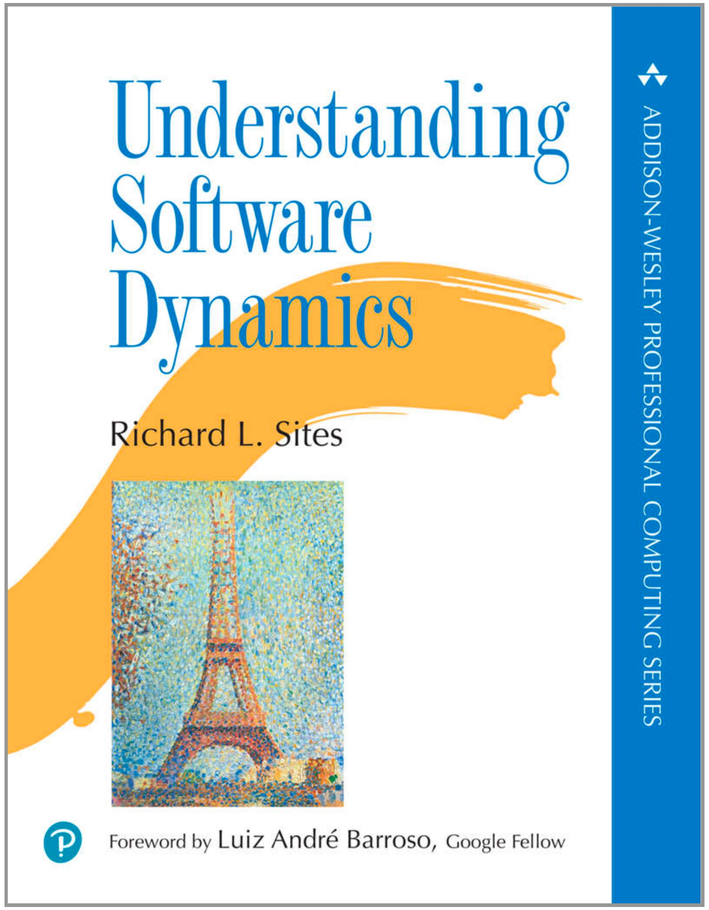
All others (DTC/CDT) – Mark out of 100

High Performance Networked-Systems

On completion of this module, students should:

- Describe the role of high performance networked-systems and where they are used;
- Develop an appreciation of best-practices by performing hands-on measurement and analysis;
- Understand the architecture of a high performance networked-system;
- Understand with greater insight high performance networking devices;
- Understand challenges and solutions to high performance measurement;
- Understand how to select or implement tools to measure high performance measurement;
- Utilise representation techniques to understand and interpret networked-systems;
- Evaluate the performance of example high performance networked-systems

Textbook



Includes a number of discussions about getting information mostly leading to the ***kutrace*** tool

We will **loan** you a copy for the duration of the course.

The book says “Software” but don’t be fooled, computer software is our (human) gateway.

If the **hardware** doesn’t make the **software** go fast – it isn’t high-performance.

This book covers much more than we will have time to cover here – **consider investing** in a copy **for** your **professional library**.

Yes I **do** want the loaners back at the end of term.

- The disks get faster
  - The CPUs get faster (at least a bit faster)
    - The Memory get faster
      - The Networks get faster – a lot faster

      - So why doesn't your program go faster too?

Because nothing is simple. Sorry.

High Performance….

"My system is performing badly"

"Well, how badly *should* it be performing?"

Just how good should your system be?

Jeff Dean's 'Numbers Everyone Should Know'

L1 cache reference	0.5 ns	O(1) ns
Branch mispredict	3.0 ns	O(10) ns
L2 cache reference	4.0 ns	O(10) ns
Mutex lock/unlock	17.0 ns	O(10) ns
Main memory reference	100.0 ns	O(100) ns
Compress 1K bytes with Zippy	2,000.0 ns	O(1) us
Read 1 MB sequentially from memory	4,000.0 ns	O(10) us
Send 2K bytes over 1 Gbps network	20,000.0 ns	O(10) us
Read 1 MB sequentially from SSD	62,000.0 ns	O(10) ms
Round trip within same datacenter	500,000.0 ns	O(1) ms
Read 1 MB sequentially from spinning disk	947,000.0 ns	O(10) ms
Disk seek	3,000,000.0 ns	O(10) ms
Read 1 MB sequentially from network	10,000,000.0 ns	O(10) ms
Send packet CA->Netherlands->CA	150,000,000.0 ns	O(100) ms

Taken from a mashup of two slides stacks and some updated numbers from URL below
http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/people/jeff/stanford-295-talk.pdf
http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf
https://kousiknath.medium.com/must-know-numbers-for-every-computer-engineer-6338a12c292c
https://norvig.com/21-days.html#answers

So….

- How long does it take to read a file?
  - a 10KByte file
    - With random seeking
      - Over the network
      - …..

- How long does it take to finish a matrix multiplication?
  - Of 10,000,000 elements
    - 8,000,000 times?

- How quickly will a particular web page be retrieved and rendered?
- How many frames of graphics per second can I render?
- How does everything affect me so?

Don’t be afraid to *do the math*

Networked Systems – some sad news – nothing is straightforward

- Two computers connected with 100GbE will .never. NEVER move data from one computer memory to another computer memory at 100Gbit/s

- Spinning disks are very weird and techniques to make them go faster have been around since the 50's

- Main memory is SERIOUSLY not sensible

- SSD and NVMe disks are just memory that (hopefully) remembers without power

- Don't get me started on the sneaky tricks CPUs use!

Measure what is measurable, and make measurable what is not so
Galileo Galilei

- If you don’t measure something – how do you know its too slow?

- Measurements are mostly over time – sometimes we want to understand quantities too.

    - How fast is the CPU? – frequency
    - How long will this job take? - seconds
    - How big is the file? – storage size
    - How much power is used? – watts (literally power)

System Measurements

Can be used to answer questions such as:

- Is this system working as expected?
- Is this system better that another system?
- What are the limitations of my system?
- Where are the system’s bottleneck?

Performance – not just bits per second

Second order effects

- Image/Audio quality

Other metrics…

- Network efficiency (good-put *versus* throughput)
- User Experience? (World Wide Wait)

- Network connectivity expectations

- Others?

What is a high-performance system?

Supercomputer?

Maybe in the 1980's

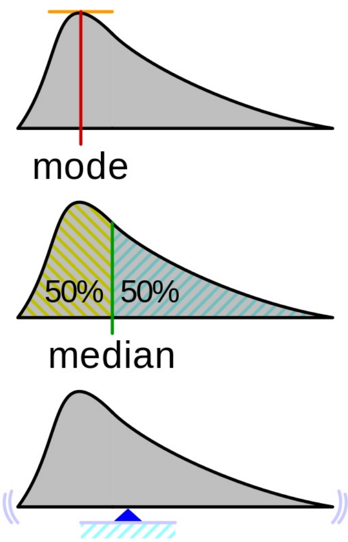
Now it's datacenters

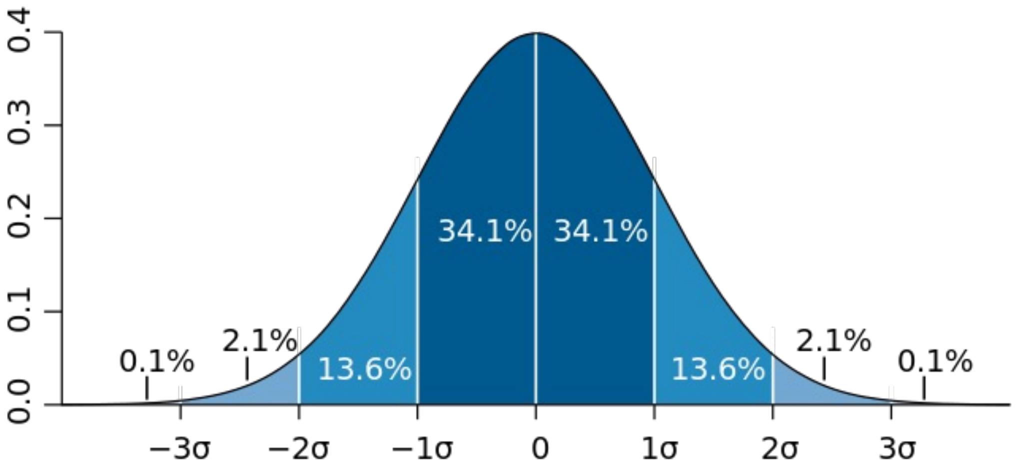
At least at the extreme

Statistics in Measurements
Terms and limits

- Mean
- Median
- Standard deviation
- Independence
- Heavy tail distribution (and where it all goes wrong)
- Probability density function / Histogram

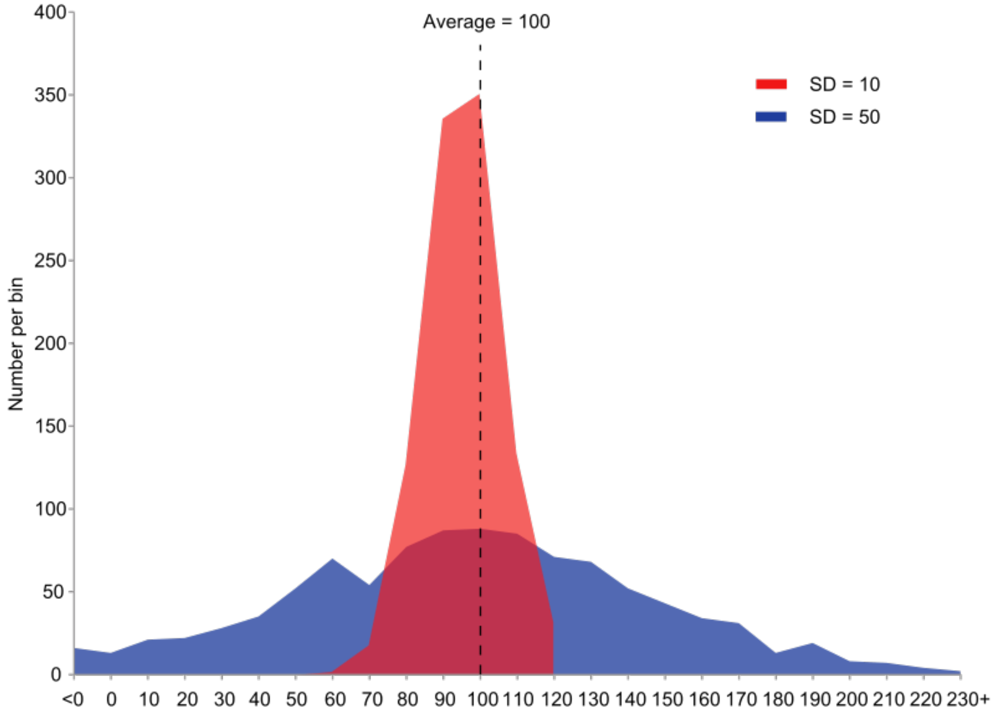
  Cumulative density function (CDF) and CCDF
- Tests (two variable or hypothesis: t-test, multivariable: ANOVA)





Standard Deviation in a Normal Distribution

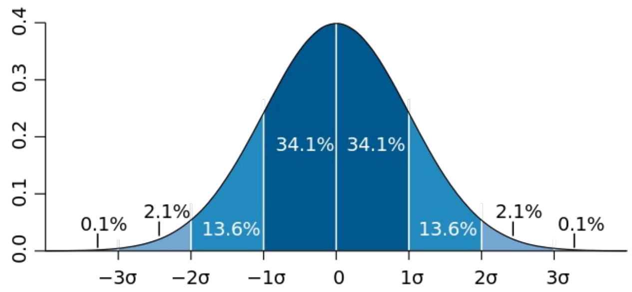
Two sets of samples with the same mean and different Standard Deviations



Confidence Intervals? Error Bars? Sample Size?

- **Confidence Interval** is the interval (range) of values you have confidence a given sample will fall within….

- **Error Bar** represents the range of all values for an experiment (sometimes the confidence interval is used – this makes assumptions!)



- **Sample Size** is the number of (measurements) made

  certain *tests* (eg t-test) can assist us in deciding on a sample size

  when we don't choose the sample size those same tests will declare

  the confidence to hold in how representative the sample-set was

Why our most-basic assumptions are wrong

or Why Independence is not a great assumption…

We measure the use of electricity in a neighbourhood over a day

There is a popular TV programme

A commercial break sees much of the population in the neighbourhood *putting the kettle on*

This is a correlated event (non-independent)

Correlation is common in the Internet too

At many timescales (weekly, daily, hourly, predictable functions of time, distance, computer-type, application-type, favourite soda….)

Why our most-basic assumptions are wrong

Independence – why we care

- Some(many/most) analysis techniques assume independence
  - Highly correlated events may mean ***non-representative*** measurements

- We might use measured data as-if it was independent/representative

What can we do?

- Constant vigilance:
- Look at the data, best-practice, *think*.

Why our most-basic assumptions are wrong

- Why Poisson distribution is not a great assumption…

We measure the use of electricity of 1000 households to determine average use as a representation (informed guess) for the nation….

Households have a high prevalence of solar panels

Not so presentative…..

This example might give a skewed distribution

This is only one cause of normal distribution failure

Distributions

- Normal Poisson Binomial….. Not the same and often ‘jumbled up’

- A **Normal distribution** is continuous
- A **Poisson distribution** is discrete
- A **Poisson** random variable is always $[0,\infty)$

- It is common to mean Poisson even if people say Normal….

Why our most-basic assumptions are wrong

Poisson distributions– why we care

- Poisson distributions make analysis and interpretation easy
  (e.g. mean, standard deviation, etc.)

What can we do?

- Look at the data, best-practice, *think*.
  - Particularly when the dataset is small

- Did I mention that normal distributions assume independence?
  <sad face>

Central Limit Theorem or "Mix enough to get Normal"

- CLT says that statistics computed from the mean (eg the mean itself) are approximately normally distributed – regardless of the distribution of the population

(OR ANOTHER WAY)

- CLT says the more data you have the more the observed mean will become the true mean

- Sadly CLT can say nothing about variance!

Law of Large Numbers or “You just need more data”

- LLN is actually a handy idea that says ”given enough data and obey the rules, the sample (measurements/overvations) will better represent the population (causal) characteristics”

- Sadly the rules are
  - Independence (again)
  - Population should not be skewed (eg be larger than *30, or is it 40? 400?....)*

- LLN is useful, it tells us lots of things:
  - <if rules> - the average of more data observations becomes the mean of the source of observations
  - But LLN says nothing about the variance.

When Standard Deviations go wrong…

- Standard Deviations (SD) indicate the *dispersion* of the underlying data

but SD measures build in some assumptions: symmetry and common computation assume a Poisson distribution….

Sometimes they simply don’t capture the nature of the data, nothing showed this up more clearly than the heavy-tail distribution…..

Heavy Tails… (condensing a lot into a slide)

- Certain phenomena (eg correlated events) can cause unusual (rare) events
- These events led to very large (wide) distributions, ones where the tail(s) has more values than a Poisson distribution would predict
- The more dispersed the data : the larger the Standard Deviation measure
- One definition of heavy tails is where Standard Deviation tends to infinity….
- Sadly, heavy tail distributions are very (VERY) common

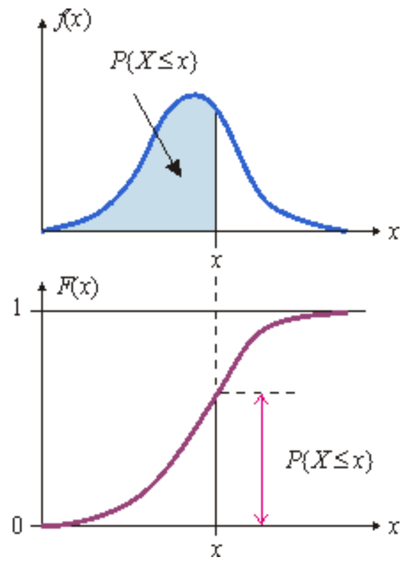
"1 in a million events occur about 9 times out of ten" – T. Pratchett

How to read a PDF CDF and CCDF…..

- A Probability Density Function tells me the probability for a specific value

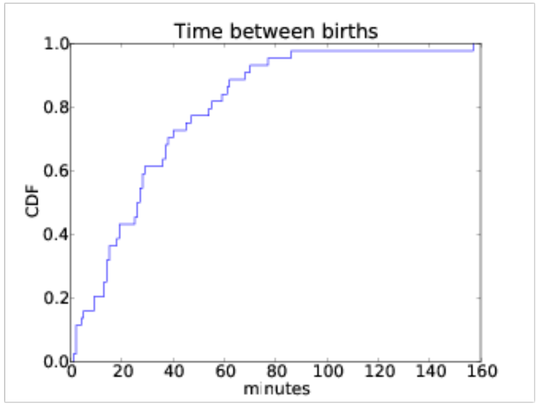
- A Cumulative Density Function is a sum of probabilities

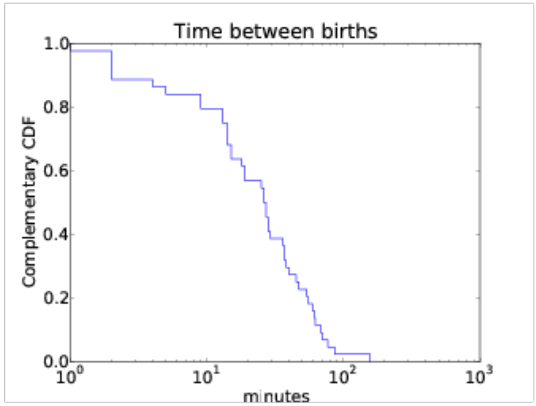
That is: “is the probability that the random variable will take a value less than or equal to a particular level.”



How to read a (C)CDF…..

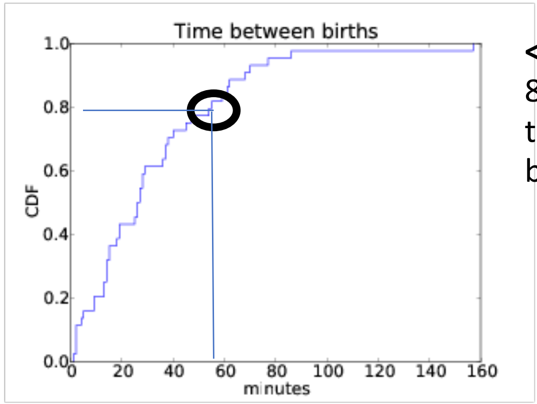
- A Complementary Cumulative Density Function 1-the sum of probabilities
  - Useful for “how often the random variable is(at or) *above* a particular level.”





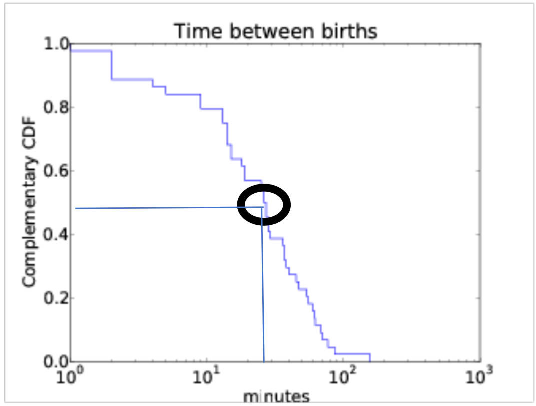
How to read a (C)CDF…..

- A Complementary Cumulative Density Function 1-the sum of probabilities
  - Useful for “how often the random variable is(at or) *above* a particular level.”



<- CDF
80% of the time it was less than 55 minutes between births

CCDF ->
Over half the *time between births* Were longer than 20 minutes



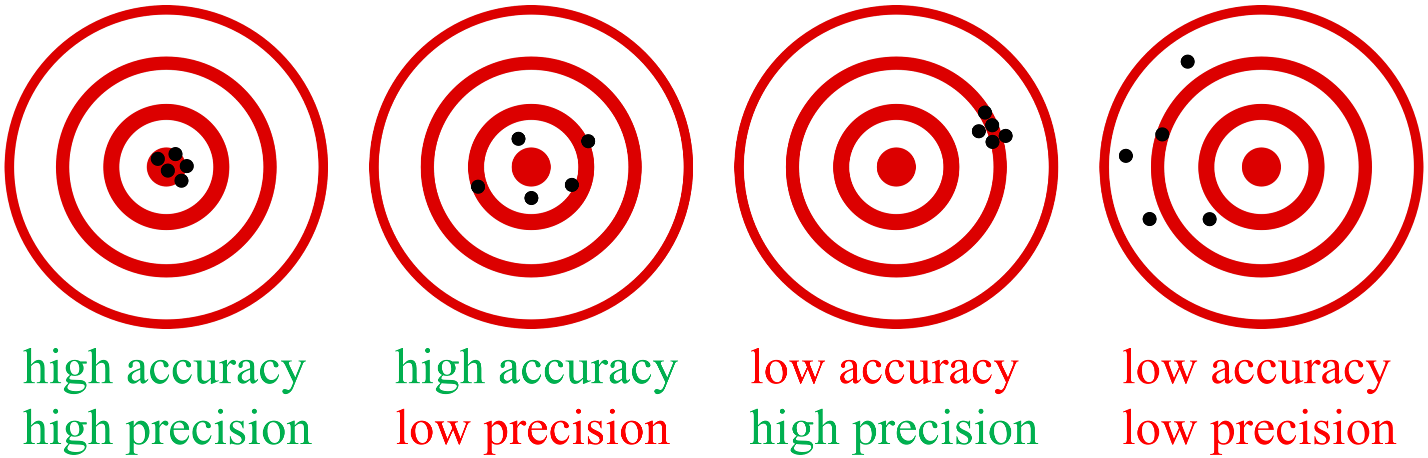
Terminology Matters!

… in greater depth in following weeks

Precision, Accuracy and Resolution

Accuracy – How close is the measured value to the real value?

Precision – How variable are the results?



Precision, Accuracy and Resolution

Resolution – The smallest measurable interval.

The resolution sets an upper limit on the precision.

high resolution

low resolution

In our experiments, resolution many times is determined by clock frequency (directly or indirectly)

Bandwidth, Throughput and Goodput

- Bandwidth – how much data can pass through a channel.
- Throughput – how much data actually travels through a channel.
- Goodput is often referred to as application level throughput.

But bandwidth can be limited below link’s capacity and vary over time, throughput can be measured differently from bandwidth etc…..

Speed and Bandwidth

- Higher bandwidth does not necessarily mean higher speed

- E.g. can mean the aggregation of links
  - ➢ 100G = 2x50G or 4x25G or 10x10G
  - ➢ A very common practice in interconnects

RTT, Latency and FCT

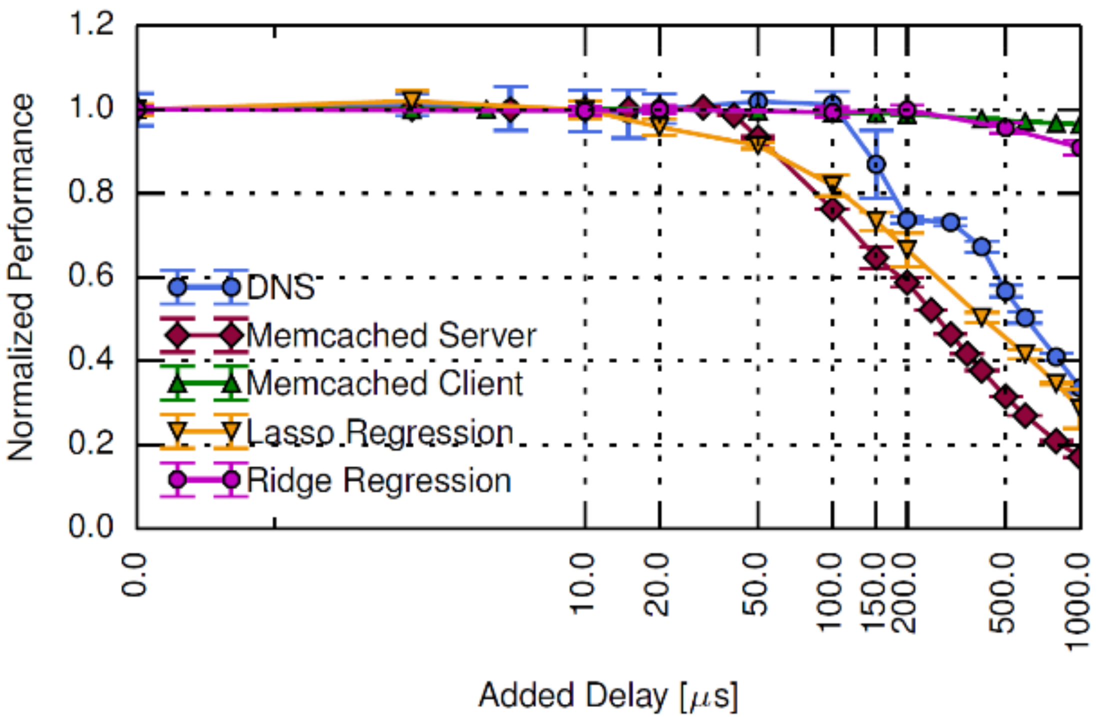
Measures of time:

- Latency – The time interval between two events.
- Round Trip Time (RTT) – The time interval between a signal being transmitted and a reply is being received.
- Flow Completion Time (FCT) – The lifetime of a flow.

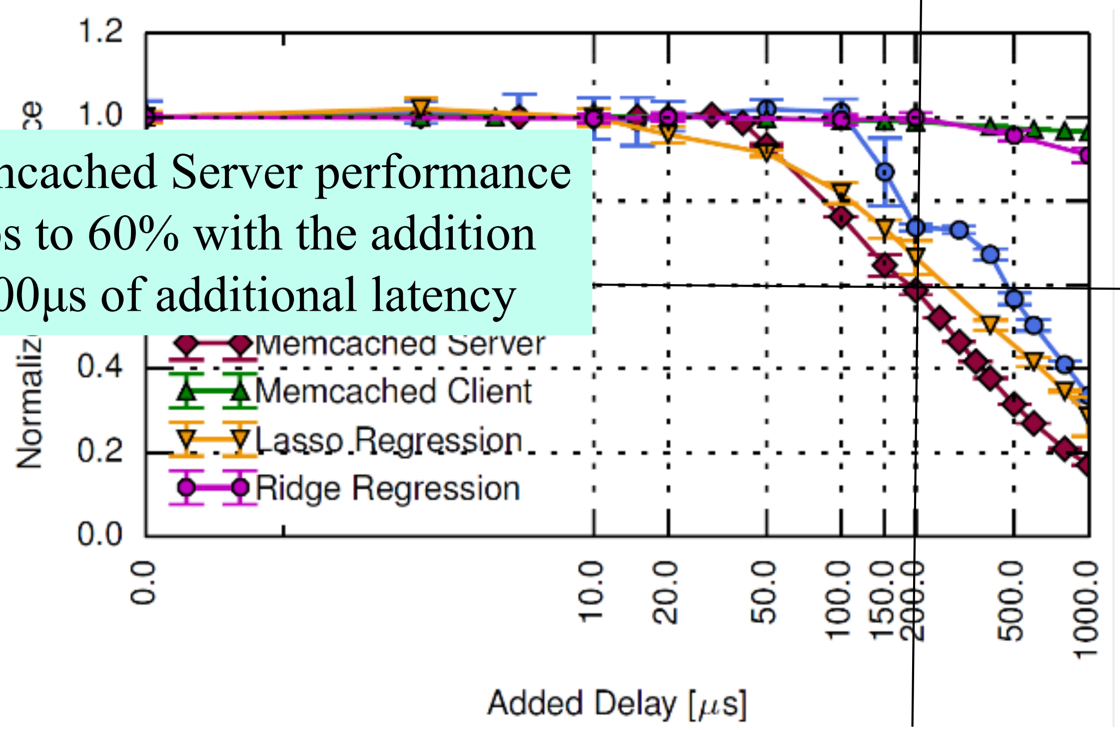
Performance Metrics

- Throughput, FCT etc. are measures of *Performance.*
- Bandwidth, RTT, packet loss etc. don’t indicate (directly) how good or bad the application / system / network perform

Example: The Effect of Latency on Application's Performance



Example: The Effect of Latency on Application's Performance



Memcached Server performance drops to 60% with the addition of 200µs of additional latency

Types of Measurements

Measurement Techniques

- Active
  - ➢Issue probe, Analyse response
- Passive
  - ➢Observe events

Example: Active vs. Passive RTT Measurement

- Active measurement – `ping`
  - Sends ICMP Echo Request message
  - Waits for Echo Reply message
  - RTT is the time gap between the request and the reply.
- Passive measurement – `tcptrace`
  - Uses TCP dump files
  - Calculates RTT according to timestamps logged in the dump.

Comparison

Passive	Active
Can only measure in the presence of activity / traffic	Measures even when tapping activity / traffic is not possible
Measures user experience, behaviour Measures protocol exchanges	Measures system, network, application performance
Raise privacy concerns	Adds probing load: - Overload system/network - May bias inferences

Measurement Vantage Point

- Point where measurement host connects to system / network
- Observations often depend on vantage point
  - Do you have enough vantage points?
  - How are the vantage points distributed?
- Can affect, e.g.:
  - Topology discovery
  - Bandwidth analysis

Possible Vantage Points

- End-hosts
  - Active measurements of end-to-end paths
  - Passive measurements of host’s traffic
- Routers/Measurement hosts in network
  - Active measurements of network paths
  - Passive measurements of traffic, protocol exchanges, configuration