Northeastern University - Seattle

CS6650 Building Scalable Distributed Systems

Professor Ian Gorton

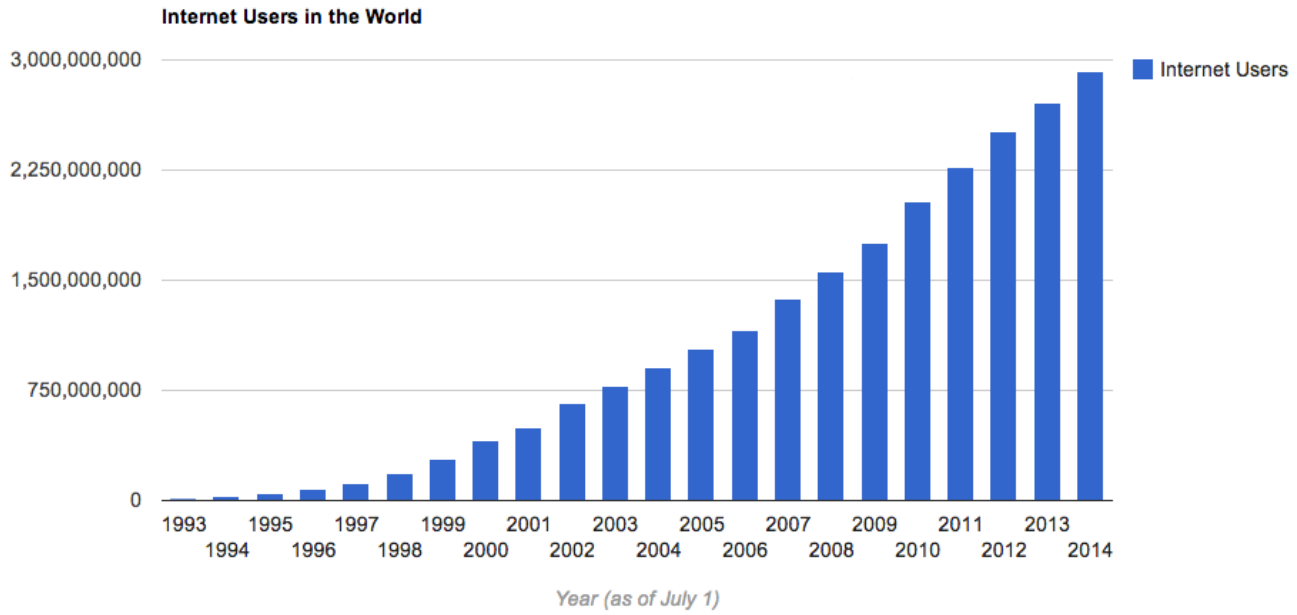
Building Scalable Distributed Systems

Week 1 – Introduction to Scalable Systems

(Part 3)
What is Scalability?

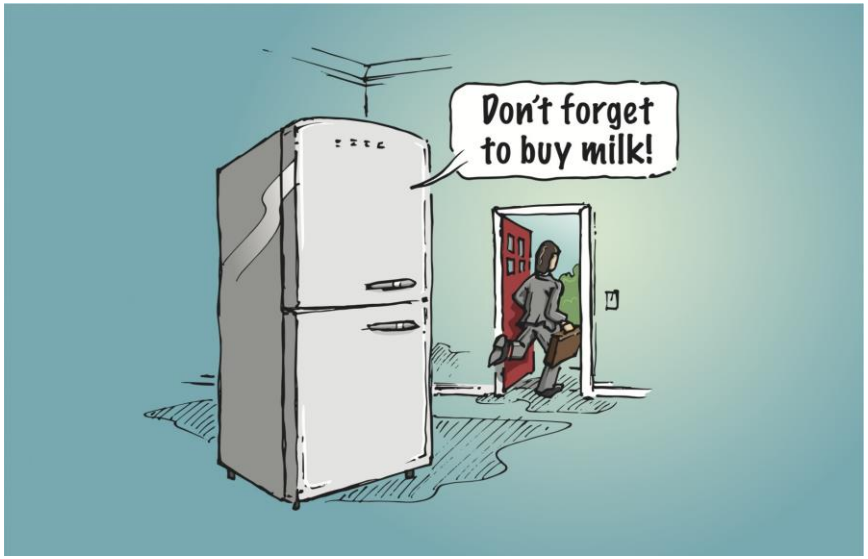
Scale is being driven by …

Around 40% of the world population has an internet connection today (view all on a page). In 1995, it was less than 1%.
The number of internet users has increased tenfold from 1999 to 2013.
The **first billion** was reached in 2005. The **second billion** in 2010. The **third billion** in 2014.
The chart and table below show the number of global internet users per year since 1993:

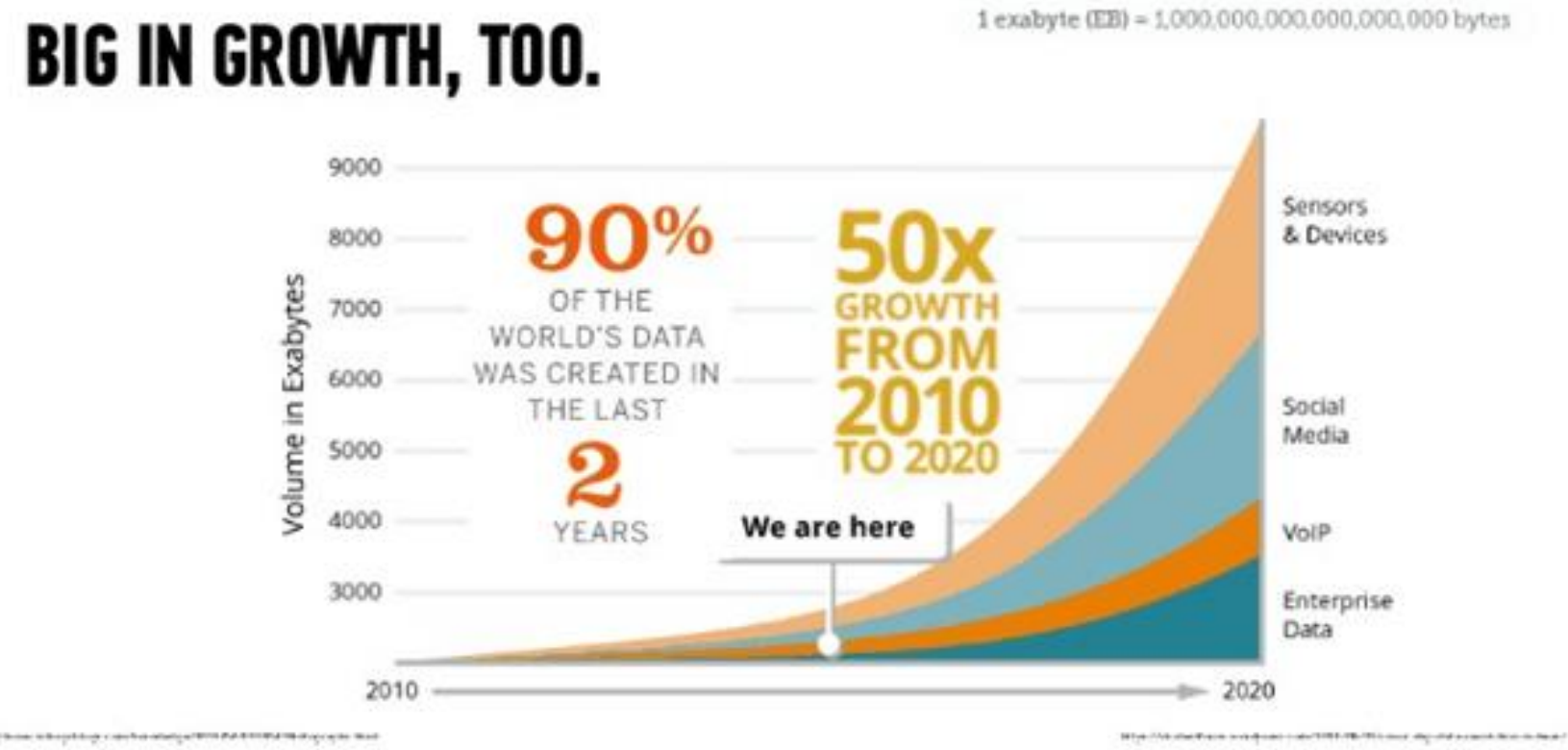


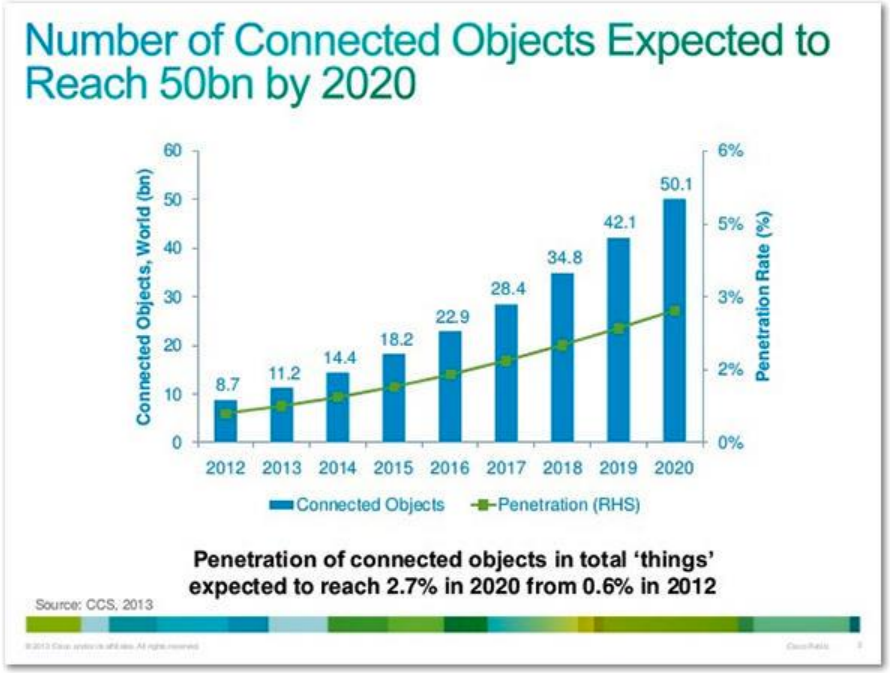
As well as ….


I LOVE IT WHEN
YOU CALL ME BIG DATA
memegenerator.net


Don't forget to buy milk!

Data Volumes





2025???

http://www.zdnet.com/article/the-internet-of-things-and-big-data-unlocking-the-power/

Question

- How are usage statistics derived in a scalable system?
- Information must be gathered about every single request
  - Session length
  - Session origin
  - Session behavior
- In pornhub.com's case, billions of visits a year
- Lot of data …..
  - A big data system?


The future is about
Big Data
Big Problems
@Web Scale

Web scale

Globally accessible

(Practically) infinite user base

Constant usage (24x7)

Ability to handle surges and spikes in requests

Controllable costs

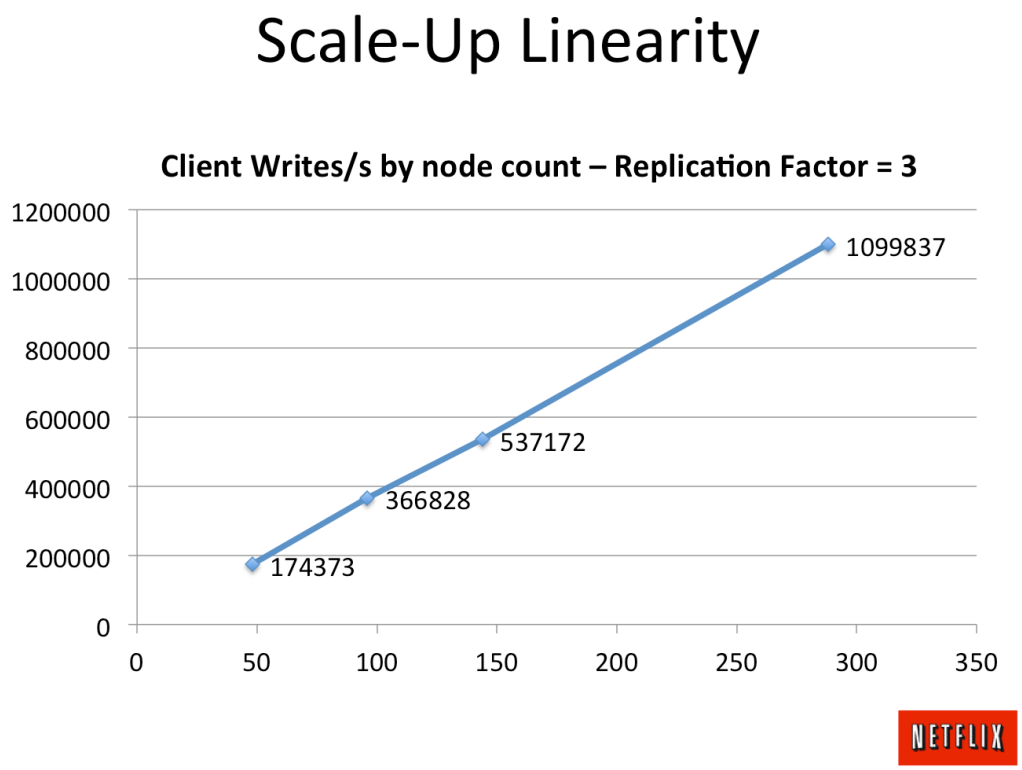
Scalability

- The ability of a system to increase or decrease capacity in response to changing demands, while continuing to satisfying service level agreements for latency and availability
- Hyperscalability – as above, while supporting exponential growth with linear costs

Just need more machines and disks?

Major Internet Companies have 1m+ servers in 2013

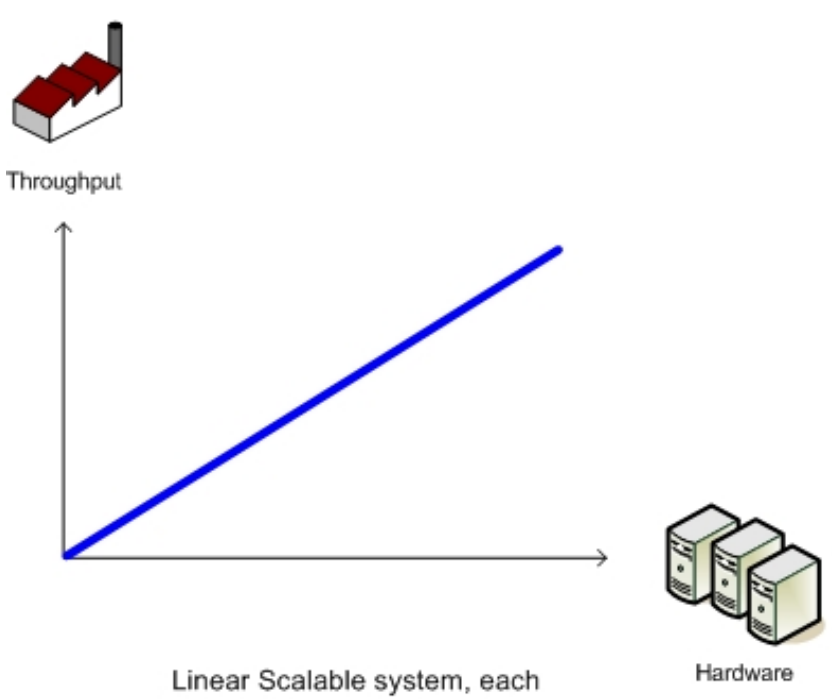
And Scale Linearly ….



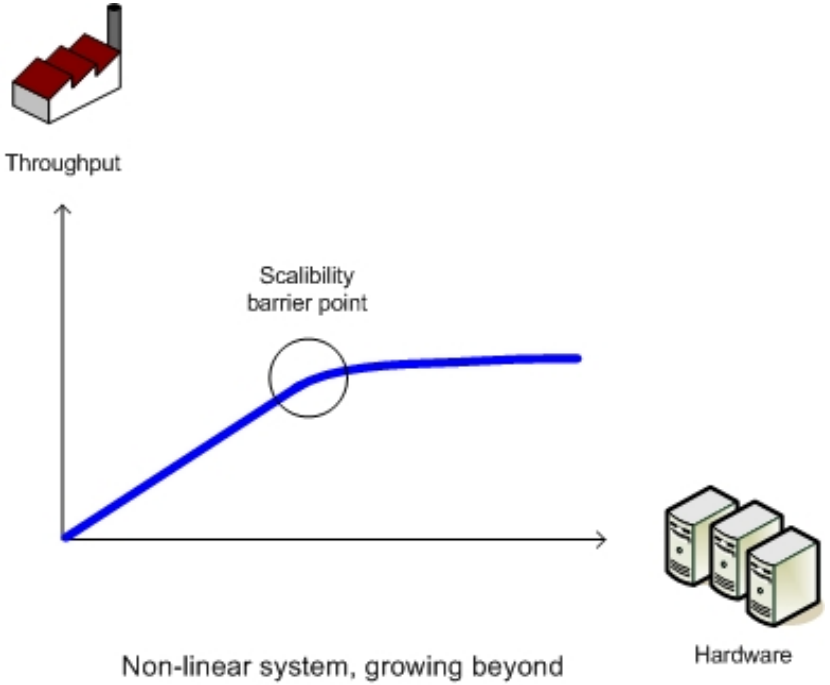
http://www.datastax.com/2012/01/choosing-the-right-architecture-for-big-data-scale


IT'S
Easy

Anyone heard of Amhdahl's law?

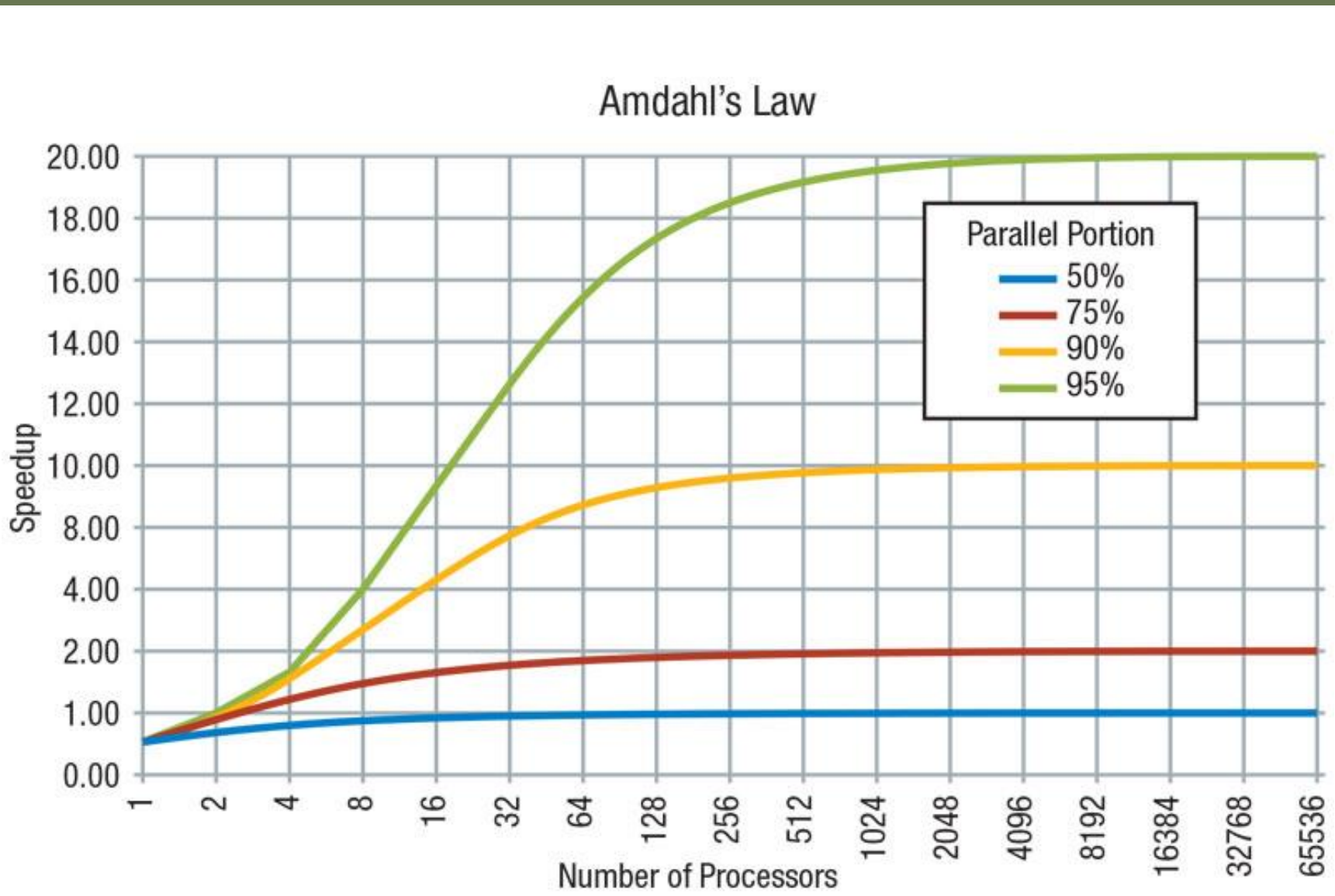


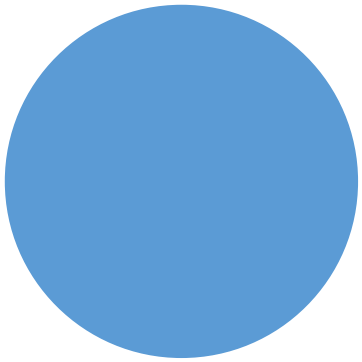
Linear Scalable system, each addition of an hardware unit adds a throughput unit



Non-linear system, growing beyond a certain barrier point adds complexity that hinders throughput increase

Amhdahl's Law

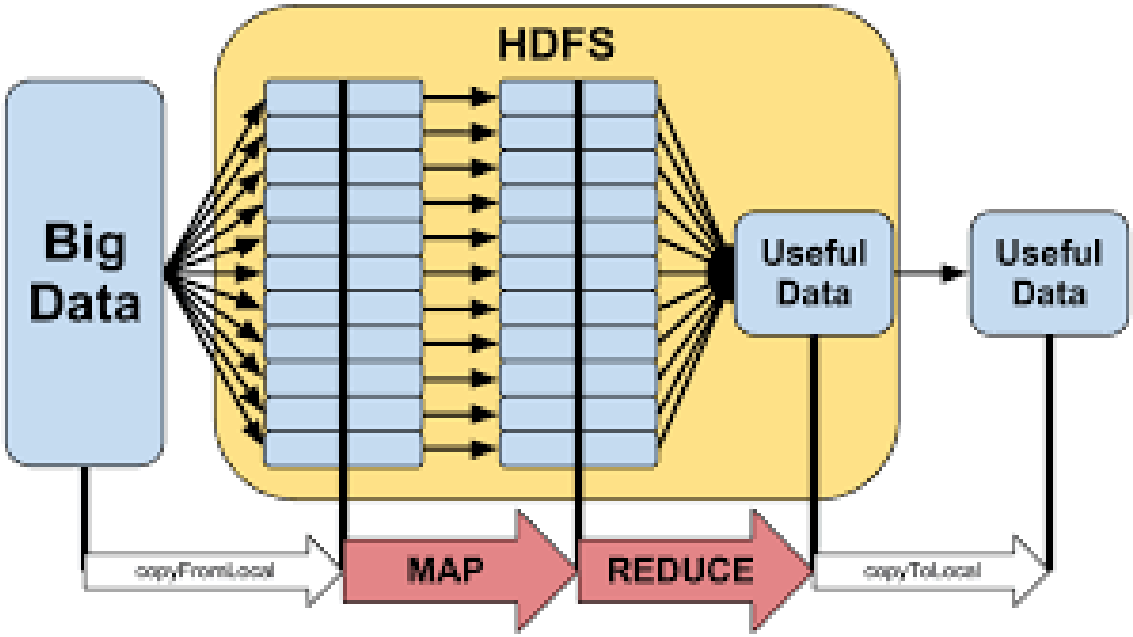

Amdahl's Law
Speedup
20.00
18.00
16.00
14.00
12.00
10.00
8.00
4.00
2.00
1.00
0.00
1
2
4
8
16
32
64
128
256
512
1024
2048
4096
8192
16384
32768
65536
Number of Processors
Parallel Portion
50%
75%
90%
95%



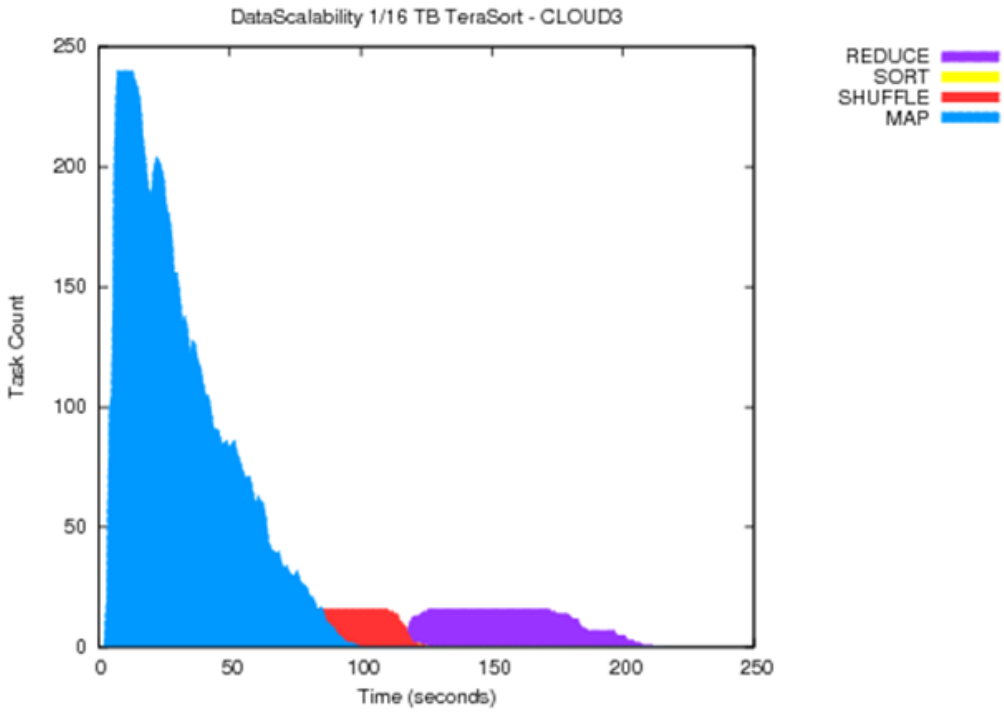

Amhdahl's Law in 2025???

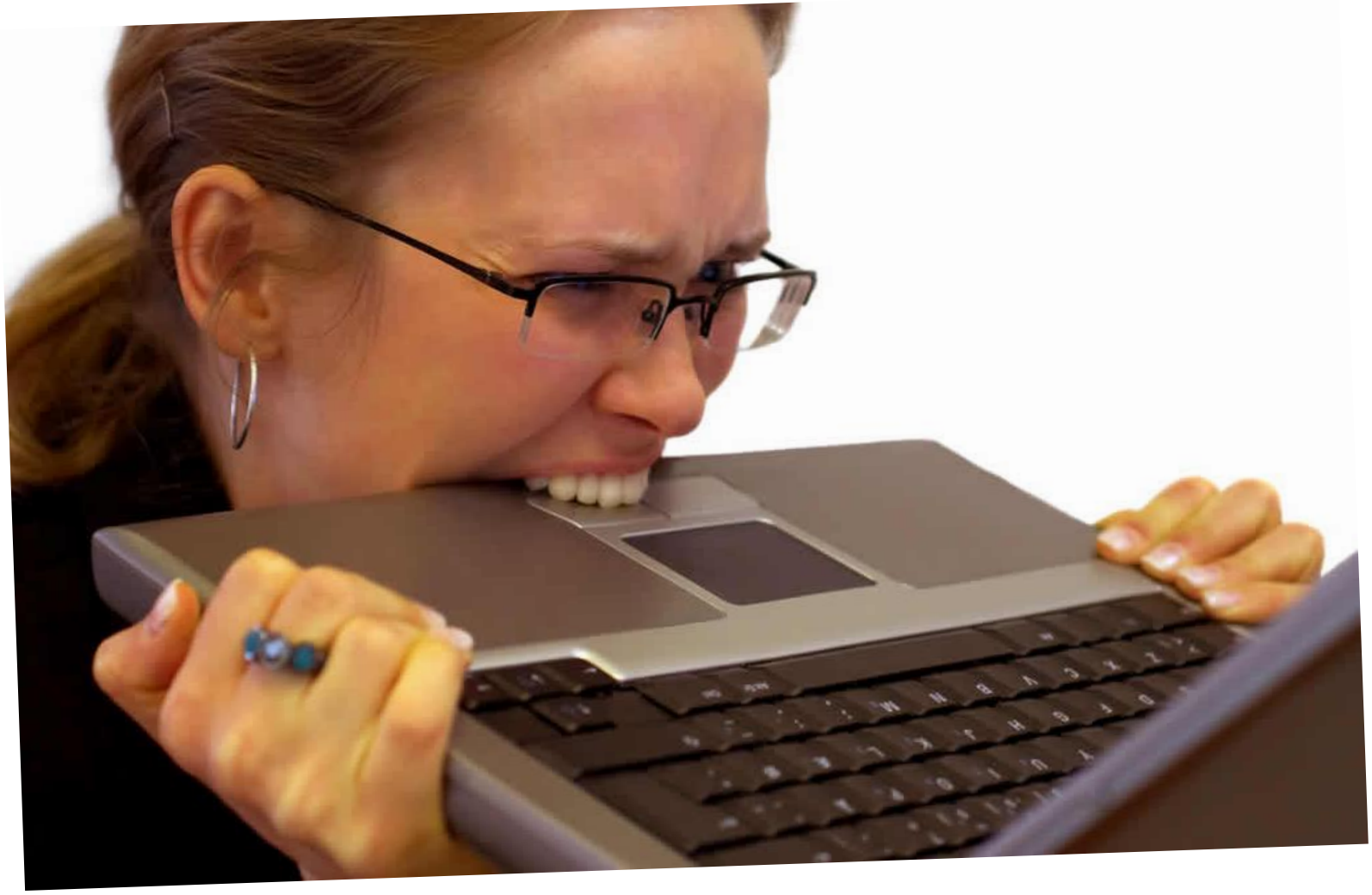

NO
CHANGE

Many
problems
are
easily
parallelized
though?


HDFS
Big Data
Useful Data
Useful Data
copyFromLocal
MAP
REDUCE
copyToLocal

Data Skew???


DataScalability 1/16 TB TeraSort - CLOUD3
Task Count
Time (seconds)
0
50
100
150
200
250
REDUCE
SORT
SHUFFLE
MAP



The future is Big Problems

That are difficult to design

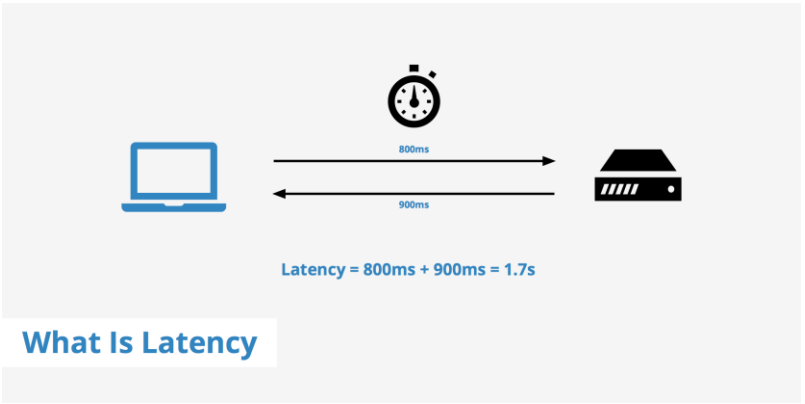
And difficult to build

And difficult to deploy

Engineering Challenges

- High traffic:
  - Consider ad exchanges that are responding to 1,000,000 requests/second
- Huge volumes of data:
  - Often terabytes or petabytes of data
- Strict requirements for systemic properties:
  - Response time, throughput, security, availability, compute cost, …

Response time



- Also known as latency
- Round trip time from when a request sent until a response is received

Response Time

Response time has an impact on user behavior

Many studies show a decrease in sales and customer retention as the response time increases

http://www.martinlugton.com/page-load-speed-important-impact-site-speed-conversions-revenue/

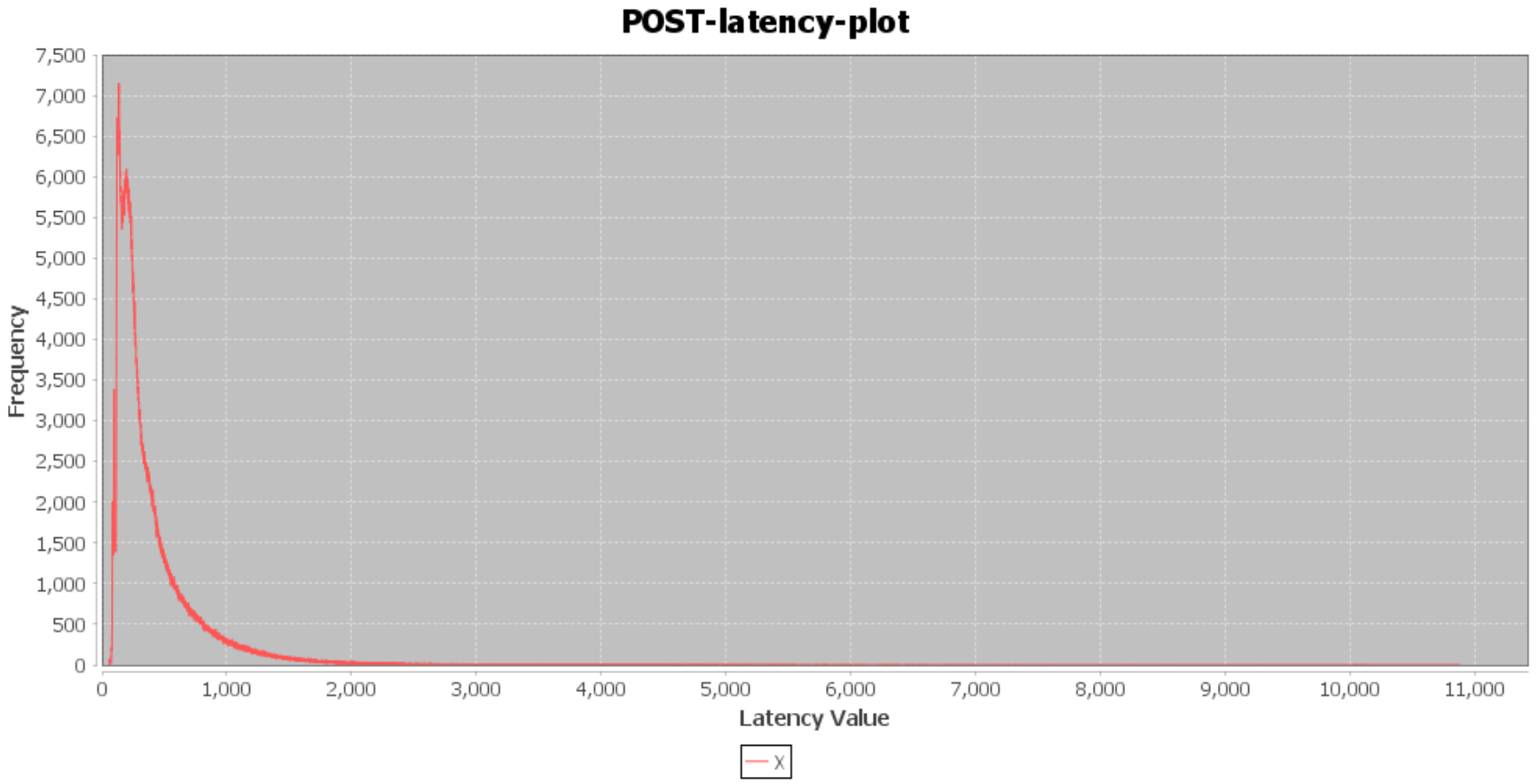
“Google: increasing page load time from 0.4 seconds to 0.9 seconds decreased traffic and ad revenues by 20%”

In some cases it’s required by contract

Responses to ad-exchanges for example need to be within 40 milliseconds

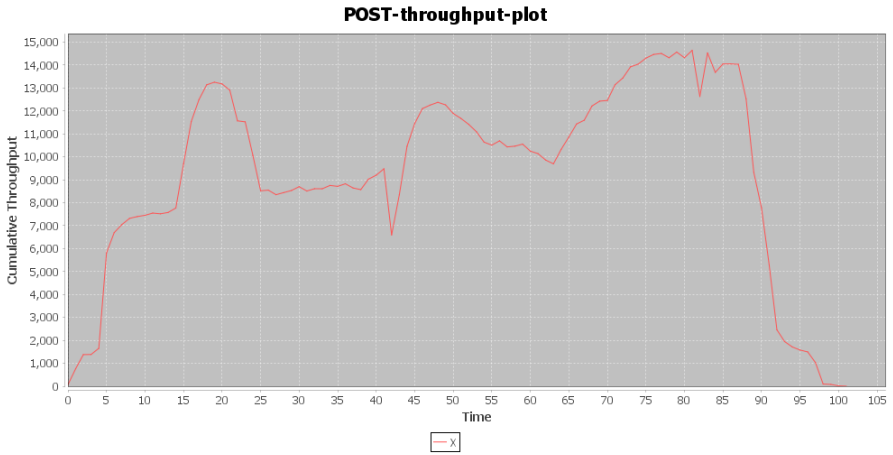
Website response time impacts google rankings

One factor google uses when determining ad listings



Typical Response Time – Long Tail

Throughput



- The capacity of a system to process requests
- Typically measured as number of requests/second
- Cf latency:
  - Latency is a measure for a single request
  - Throughput is a bulk measure of many requests

Throughput

Massive throughput is needed at scale

Millions of requests per second

Before web scale systems, we just used bigger more powerful systems

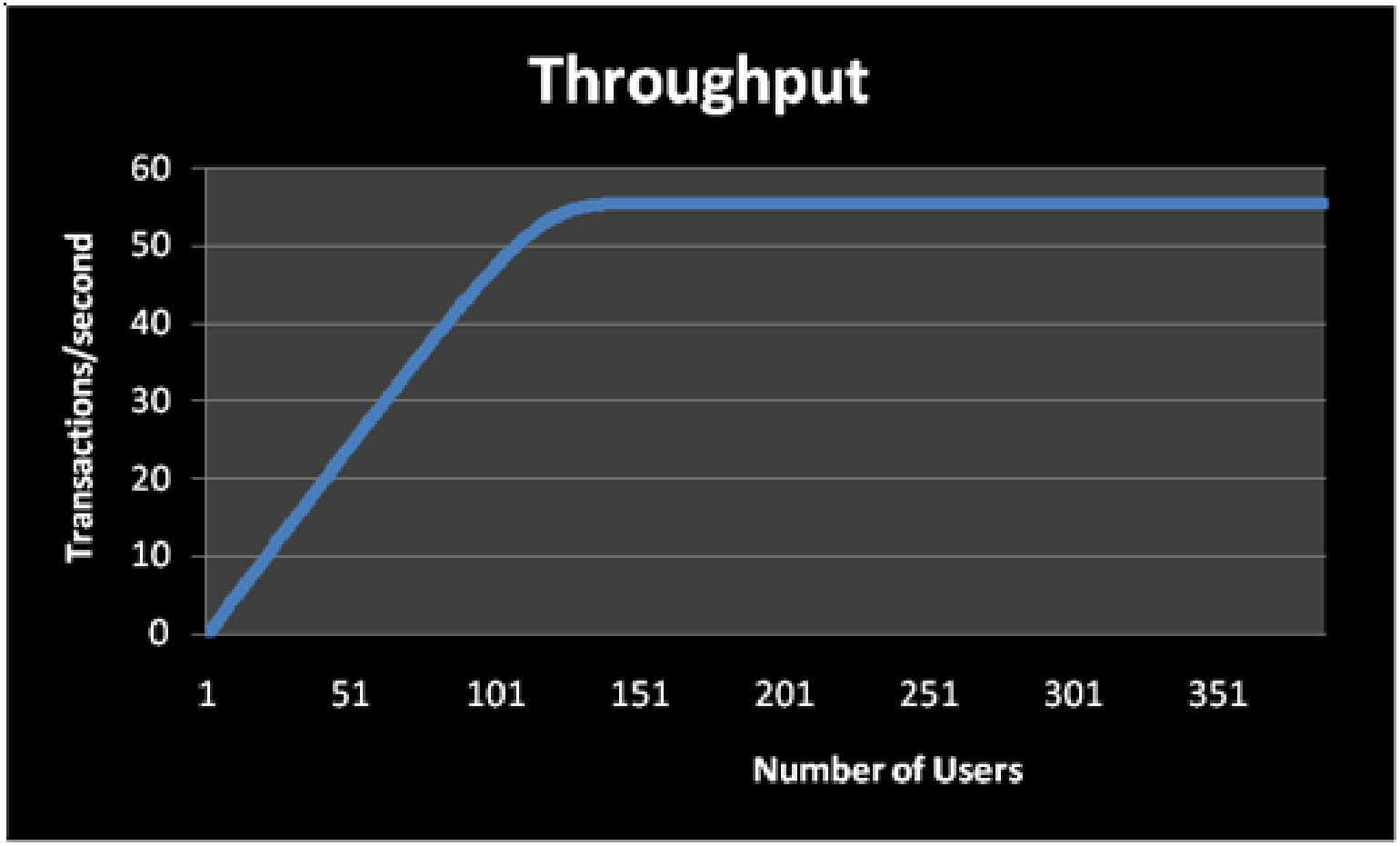
Particularly for the database server

That's no longer an option at web scale

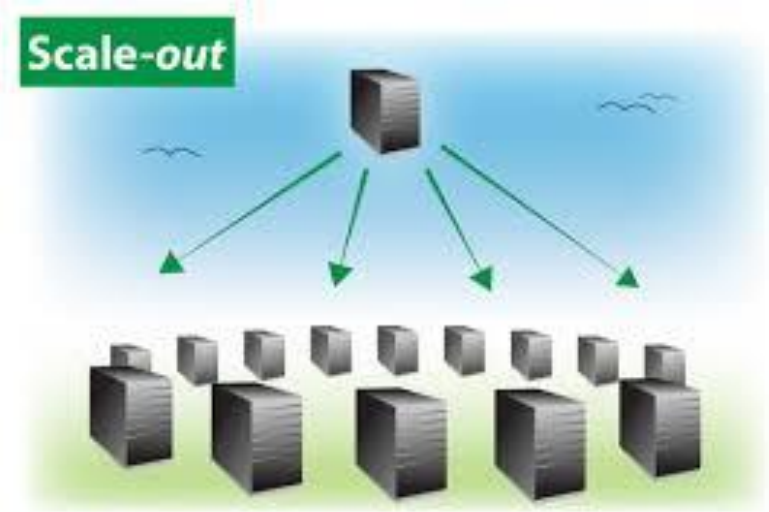
We need to be able to efficiently scale out

In other words be able to split the load across multiple nodes

Throughput with fixed capacity



Scale Out



- **Scale** horizontally (or **scale out**) means adding more nodes to a system
- Increases capacity and hence throughput
- 100s to 1000s of nodes needed at scale
- Scale in/down equally important
  - Why???

Availability

The percentage of time a system is accessible to user

At web scale

Global customer base

High availability required

How high?

99%

99.999%

99.999999%

Availability

- If a system has 1 component
  - It if not available if the element (software/hardware) fails
  - Single Point of Failure (SPoF)
  - So need very highly reliable component
- To achieve scalability, we need to scale out
  - 100s and 1000s of components
  - Do they all need to be always available?
- At scale *everything* breaks …
  - 3% of disks fail in a data center each year
- Systems need to be able to deal with regular component failures
  - In other words, *fault tolerant*
  - ***Built to handle problems as failure is expected***

Managing Tradeoffs

- There is not one way to build such systems
- Requirements and context are going to differ from one system to another
- Optimizing on one system property is inevitably going to compromise another concern, eg
  - Throughput versus cost
  - Performance versus availability
- As an engineer you need to be able to understand:
  - the concerns and context of a given system
  - the nature of the tradeoff that results from a set of decisions
  - And use metrics to guide your designs

Summary

This course examines the trade-offs needed to build a scalable system

We do this though studying scalable distributed system principals and technologies that enable them

We build a series of assignments which give practical insights into building scalable systems

Next – course outline