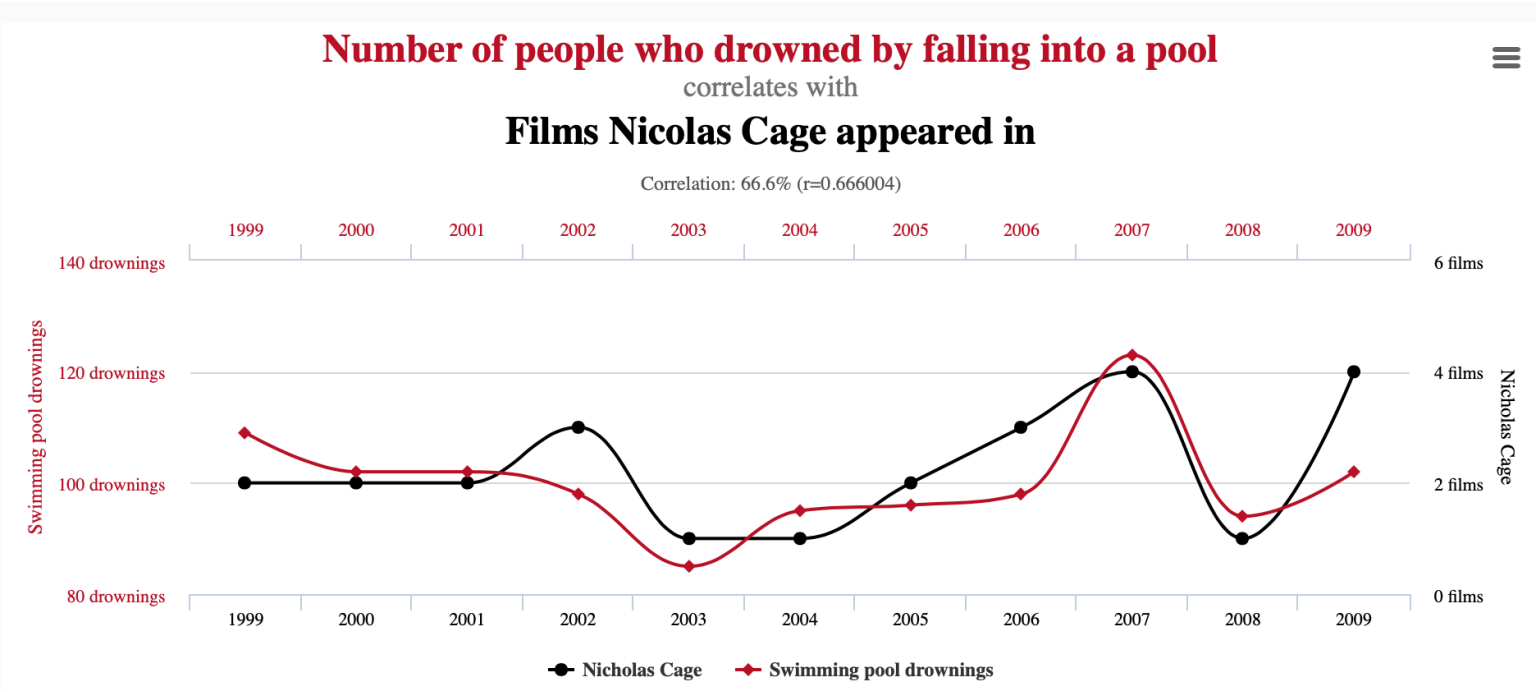
CS 677: Big Data

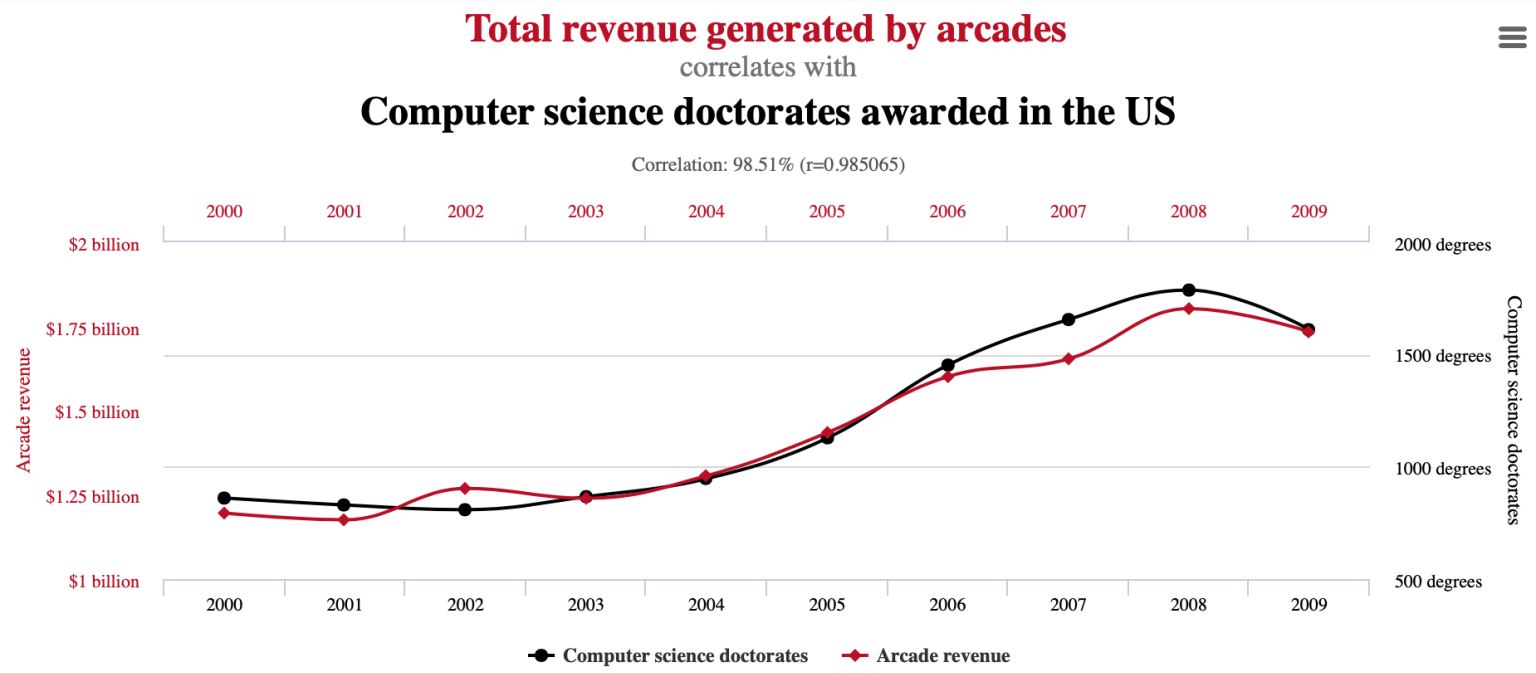
Warehouse-Scale Computing

Lecture 7

Spurious Correlations



Spurious Correlations



Rethinking the Network

- Handling massive datasets requires distributing the computations and storage
- To deal with this, Google pioneered the concept of *warehouse-scale computing*
- To get started, let's talk about a bit of theory…

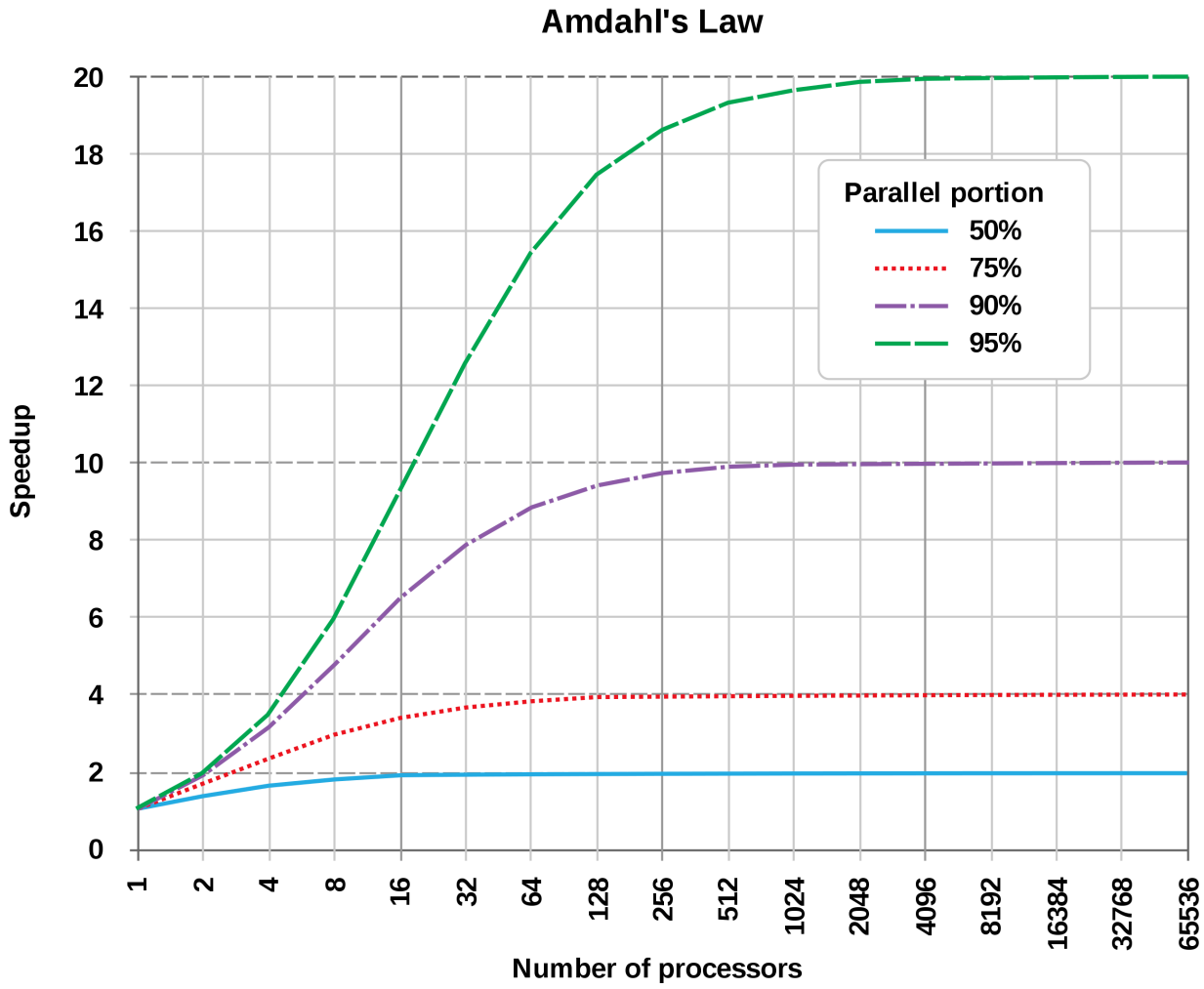
Amdahl's Law [1/2]

- In the best case scenario, doubling the number of cores/CPUs/processing units will halve execution time
  - In practice, this is difficult
- There is overhead associated with parallelism
- Amdahl's law puts a bound on potential speedup:

$$S_N = \frac{1}{(1-P)+\frac{P}{N}}$$

  - S – speedup
  - P – parallelizable portion of the program
  - N – number of cores/CPUs/processing units

Amdahl's Law [2/2]



Distributed Computation

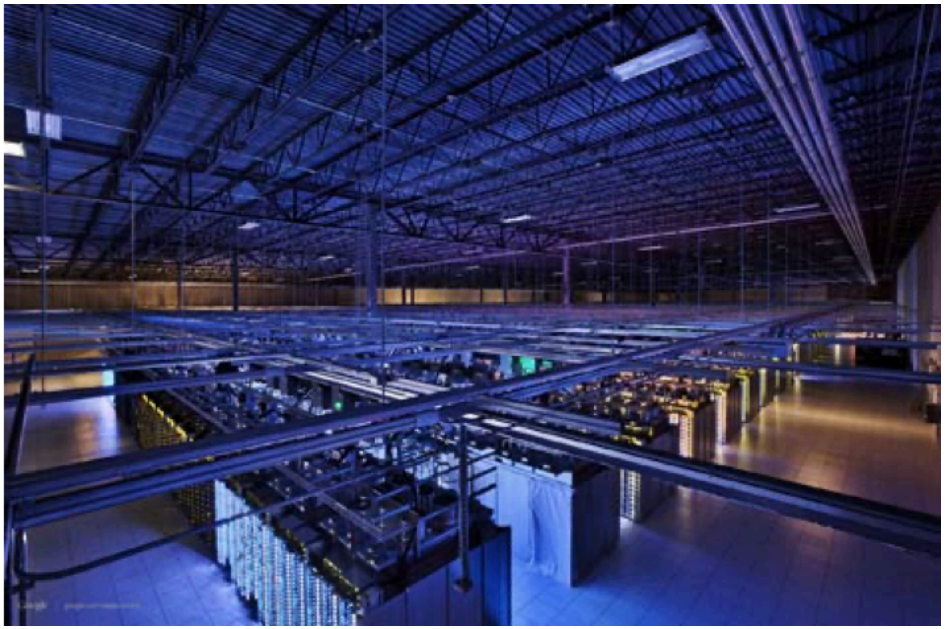
- The whole goal of many Big Data computation frameworks is to maximize P
  - (the parallelizable portion)
- Sometimes this requires an entirely new programming model
  - **MapReduce** paradigm
  - **Restrict** what programmers can do
    - The upside: automatic parallelization (or close enough)

Warehouse-Scale Computing

- Google's *warehouse scale computing* concept touches on both the software and hardware stacks
  - Barroso et al., *The Datacenter as a Computer*
- In this model, data centers are filled with commodity hardware with the best **dollar:performance** ratio
  - Don't buy a machine with a Core i9/i7… get an i5
- Connect these **nodes** with a reasonably-priced interconnect
- View the entire datacenter as a single computer and design software that matches this model

WSCs





Rationale

"as computation continues to move into the cloud, the computing platform of interest no longer resembles a pizza box or a refrigerator, but a warehouse full of computers"

-- Barroso et al., The Datacenter as a Computer

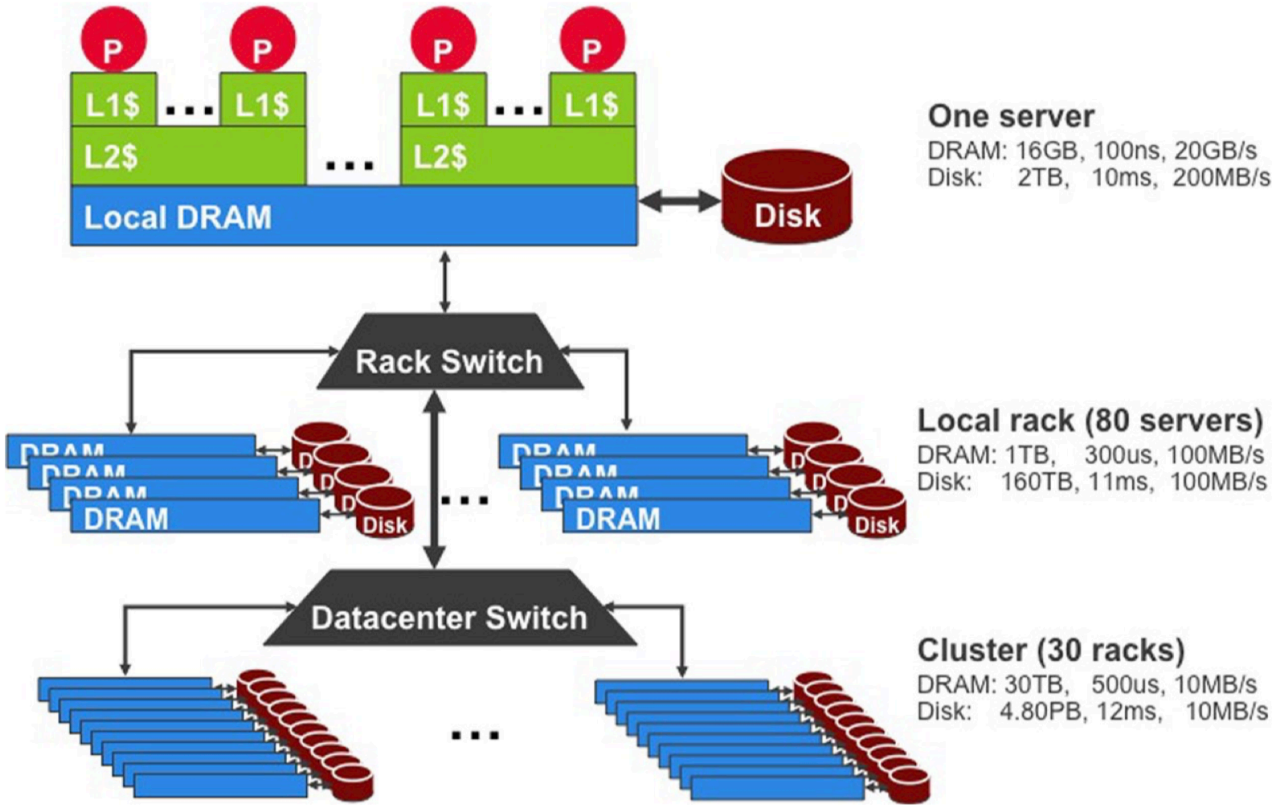
Google Modular Data Center

- Google has a patent for data centers based on shipping containers
  - Others have employed similar technology
- Pack a shipping container with cooling, racks of servers, and a fast network
- Need more capacity? Stack a few more shipping containers on the heap!

Warehouse-Scale Software

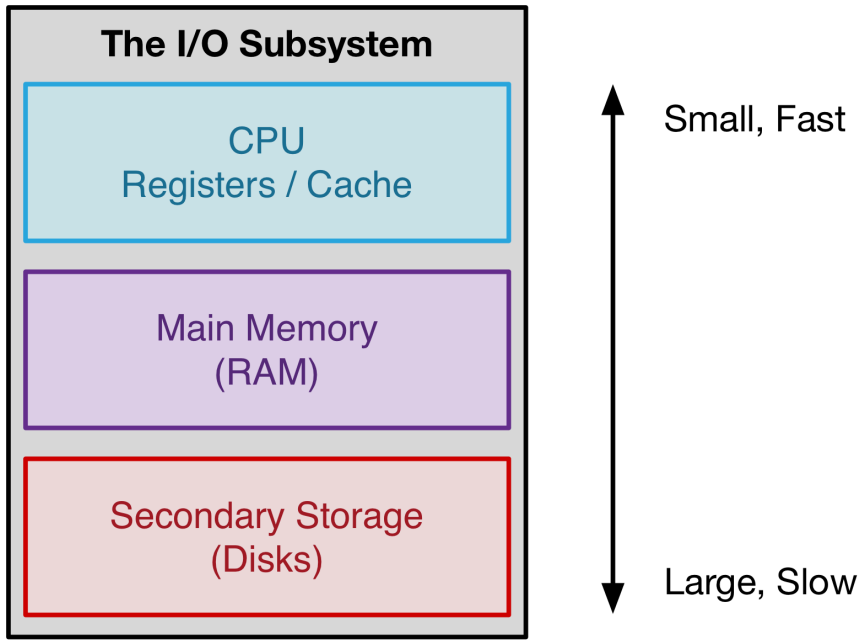
- Google's next challenge: if our datacenter is a "computer," what software does it run?
- We need a way to compute things
  - Like the process abstraction we have on operating systems
  - Led to the creation of **MapReduce**
- We need a way to store and retrieve things
  - Led to the creation of **Google File System (GFS)**
- These technologies laid the foundation for modern data-intensive computing approaches

WSC Storage Hierarchy



The I/O Subsystem

- Big Data spans the whole stack: hardware, low-level (OS) software, networking, algorithms…
- There is a *speed differential* in the hardware memory hierarchy
- Most big data is on **secondary storage**
  - The challenge? Getting it off of the disks and into cache, memory, or even the network
- Thought experiment:
  - Memory accesses are measured in ns
  - Hard disk drive accesses are measured in ms
  - If we scale up, say 1 ns = 1 s, then a single hard disk access takes at least 11.5 days!



The General Idea

- Avoid the disk at all costs
  - If you must touch the disk, then try to limit the number of passes over the data
- Sequential read/write patterns are often faster than random
  - Even with many models of SSD!
- One of the big benefits of Spark over Hadoop MapReduce we'll see later in class is being able to cache data in memory

Designing for WSCs

- Let's take a minute or two to think about this…
- What factors do we have to consider when designing a warehouse-scale computing platform?
  1. Low-level (systems)
  2. Network
  3. Algorithmic
  4. …Something else?
- Chat with the person next to you for a couple minutes, then let's brainstorm as a group

File Transfer Lab

- With the time remaining, let's discuss and work on the file transfer lab
- Find a partner to share your `.proto` with
  - Your projects should be separate implementations, but compatible at the wire format level