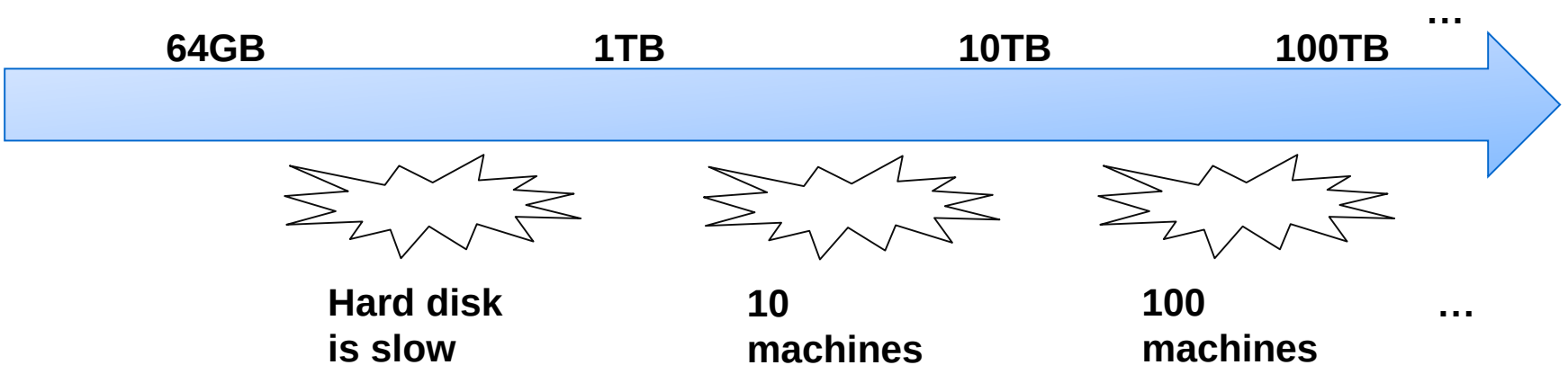
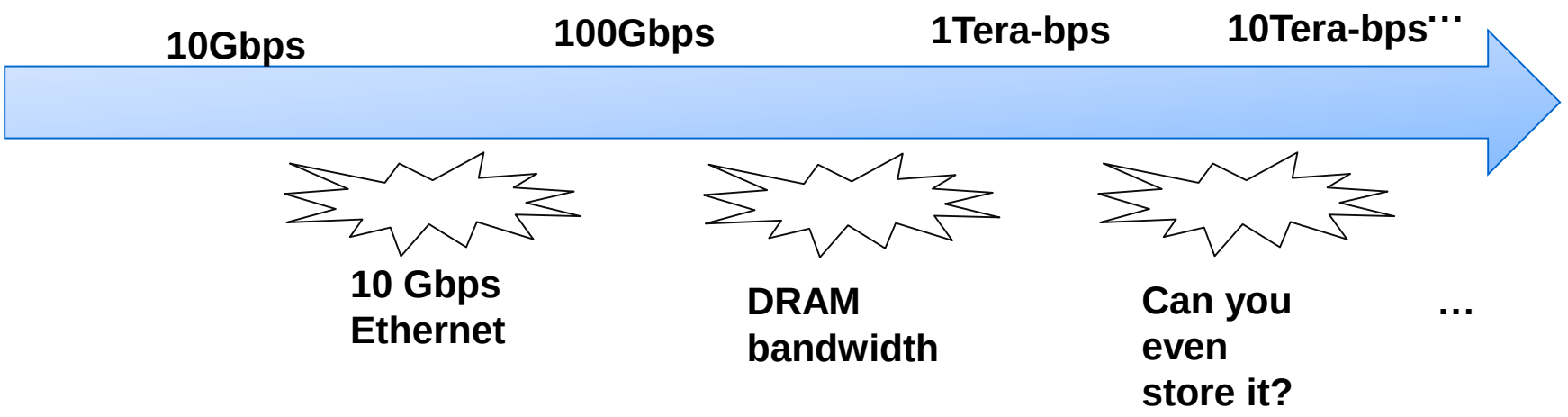
Recap: Volume

- Volume: The scale of data
- The challenges of large volume
  - Performance
  - Cost
  - Reliability
  - Algorithm design complexity



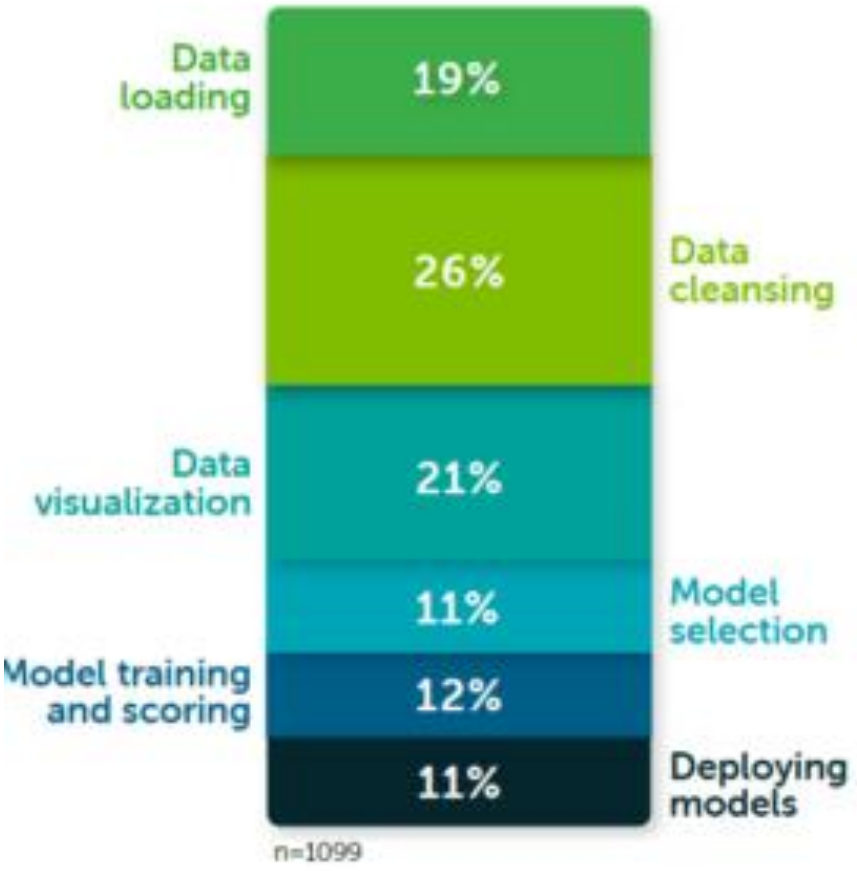
Recap: Velocity

- Velocity: the speed of new data (streaming data)
- The challenges of high velocity
  - Performance
  - Cost
  - Reliability
  - Algorithm design complexity



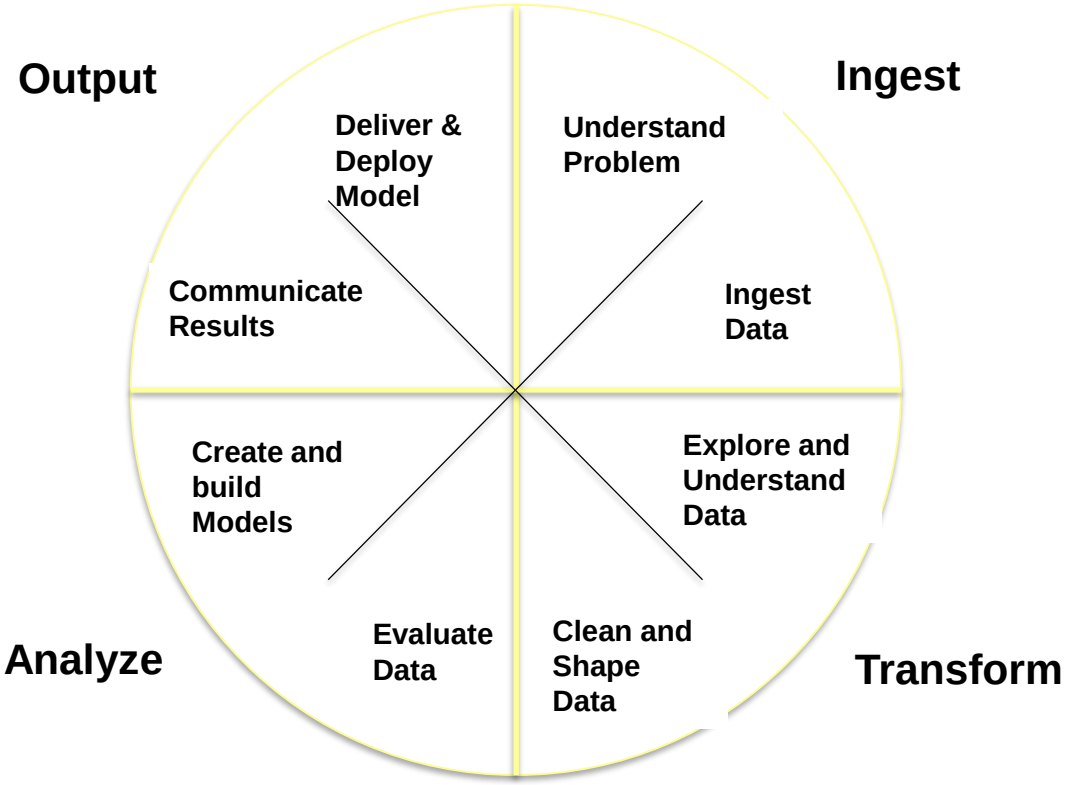
Recap: Variety and Veracity

- Why Variety matters?
  - "One size does not fit all"
  - Data integration
  - Multi-modal learning
- Why Veracity matters?
  - Dirty and noisy data
  - Data provenance
  - Data uncertainty



https://www.datanami.com/2020/07/06/data-prep-still-dominates-data-scientists-time-survey-finds/

Recap: Data Lifecycle



Recap: Cloud Computing, Who cares?

- A source of problems…
  - Cloud-based services *generate* big data
  - Clouds make it easier to start companies that *generate* big data
- As well as a solution…
  - Ability to provision analytics clusters on-demand in the cloud
  - Commoditization and democratization of big data capabilities

CS4225/CS5425 Big Data Systems for Data Science

Principles of Big Data Systems

Bingsheng He
School of Computing
National University of Singapore
hebs@comp.nus.edu.sg

School of Computing

Learning Objectives

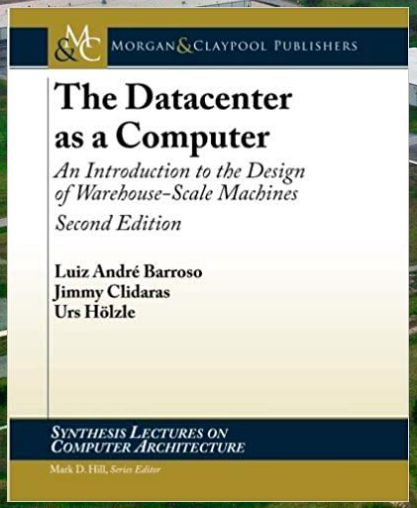
- Learn the storage and memory architectures of data centers as well as the cost of moving data in the data center.
- Understand the four “big Ideas” for building efficient big data systems
- Understand the motivations for the right abstractions of big data systems

Outline

- **Data center architecture**
- The four “Big Ideas”
- Abstractions for big data systems

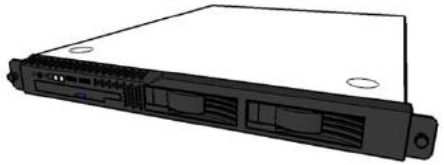
The datacenter *is* the computer!

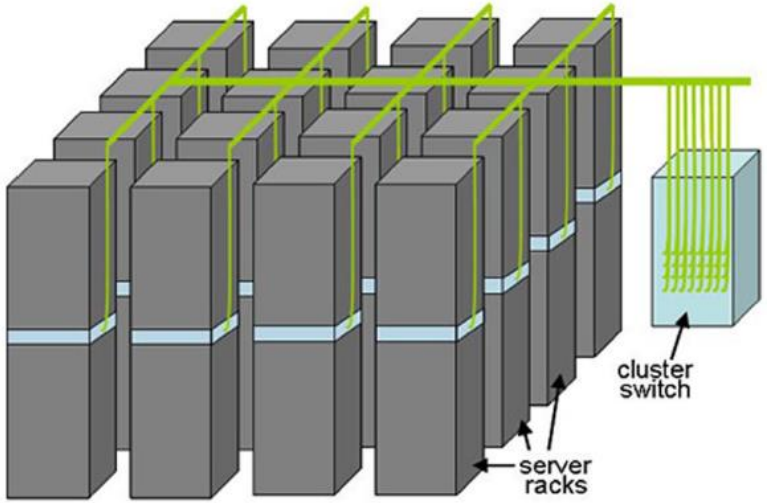
(That is: we think of many machines in a data center as a big "processing unit" being used to solve a problem, rather than just as independent machines)



Source: Google

Building Blocks





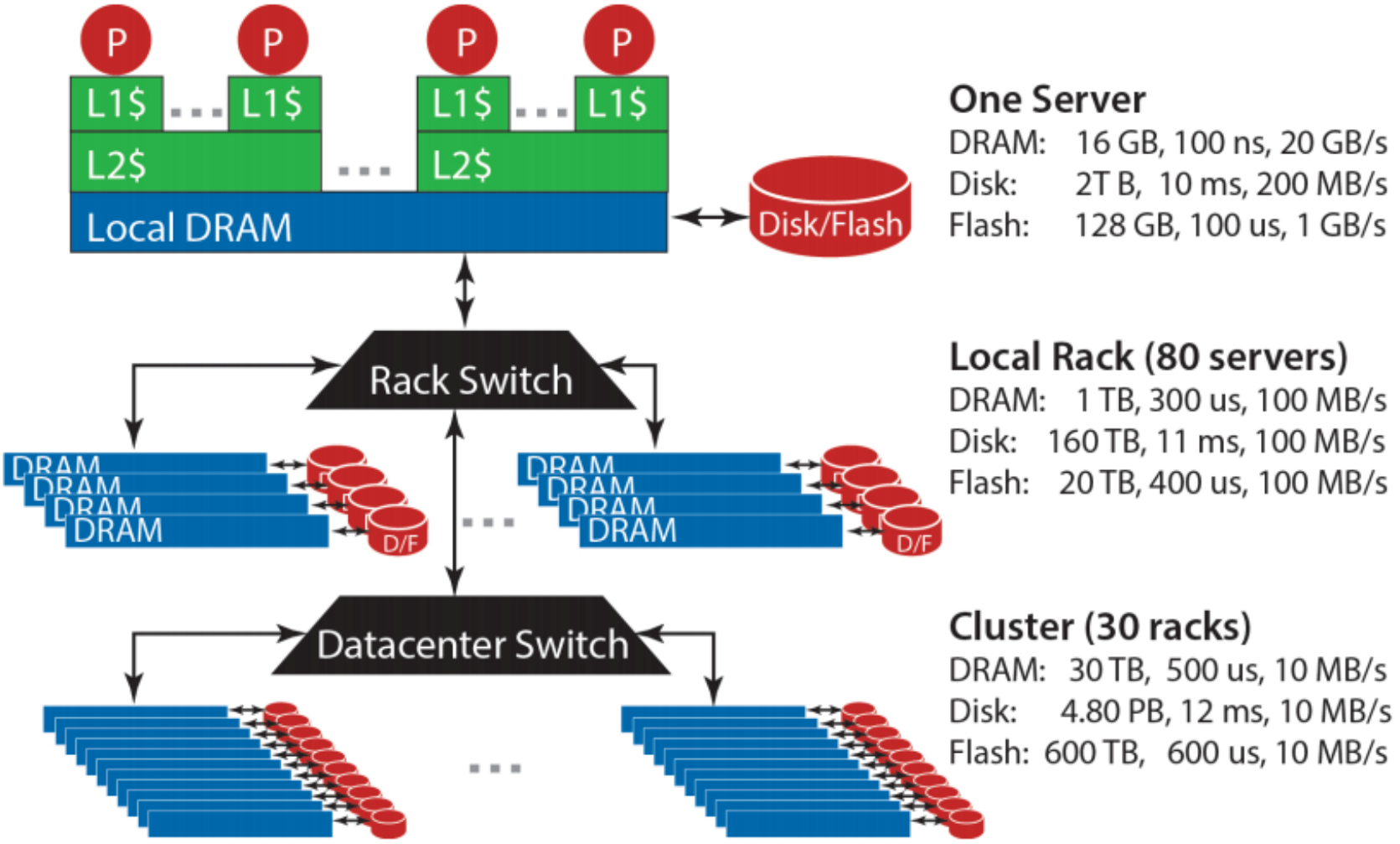
Source: Barroso and Urs Hölzle (2009)

Source: Google

Source: Google

Source: Facebook

Storage Hierarchy



One Server
DRAM: 16 GB, 100 ns, 20 GB/s
Disk: 2T B, 10 ms, 200 MB/s
Flash: 128 GB, 100 us, 1 GB/s

Local Rack (80 servers)
DRAM: 1 TB, 300 us, 100 MB/s
Disk: 160 TB, 11 ms, 100 MB/s
Flash: 20 TB, 400 us, 100 MB/s

Cluster (30 racks)
DRAM: 30 TB, 500 us, 10 MB/s
Disk: 4.80 PB, 12 ms, 10 MB/s
Flash: 600 TB, 600 us, 10 MB/s

Source: Barroso and Urs Hölzle (2013)

Bandwidth vs Latency

- **Bandwidth**: maximum amount of data that can be transmitted per unit time (e.g. in GB/s)
- **Latency**: time taken for 1 packet to go from source to destination (*one-way*) or from source to destination back to source (*round trip),* e.g. in ms
- When transmitting a *large* amount of data, bandwidth tells us roughly how long the transmission will take.
- When transmitting a very *small* amount of data, latency tells us how much delay there will be.
- (**Optional**: **Throughput** is similar to bandwidth, but instead of referring to capacity, it refers to the rate at which some data was *actually transmitted* across the network during some period of time.)

Low Bandwidth

High Bandwidth

Low Latency

High Latency

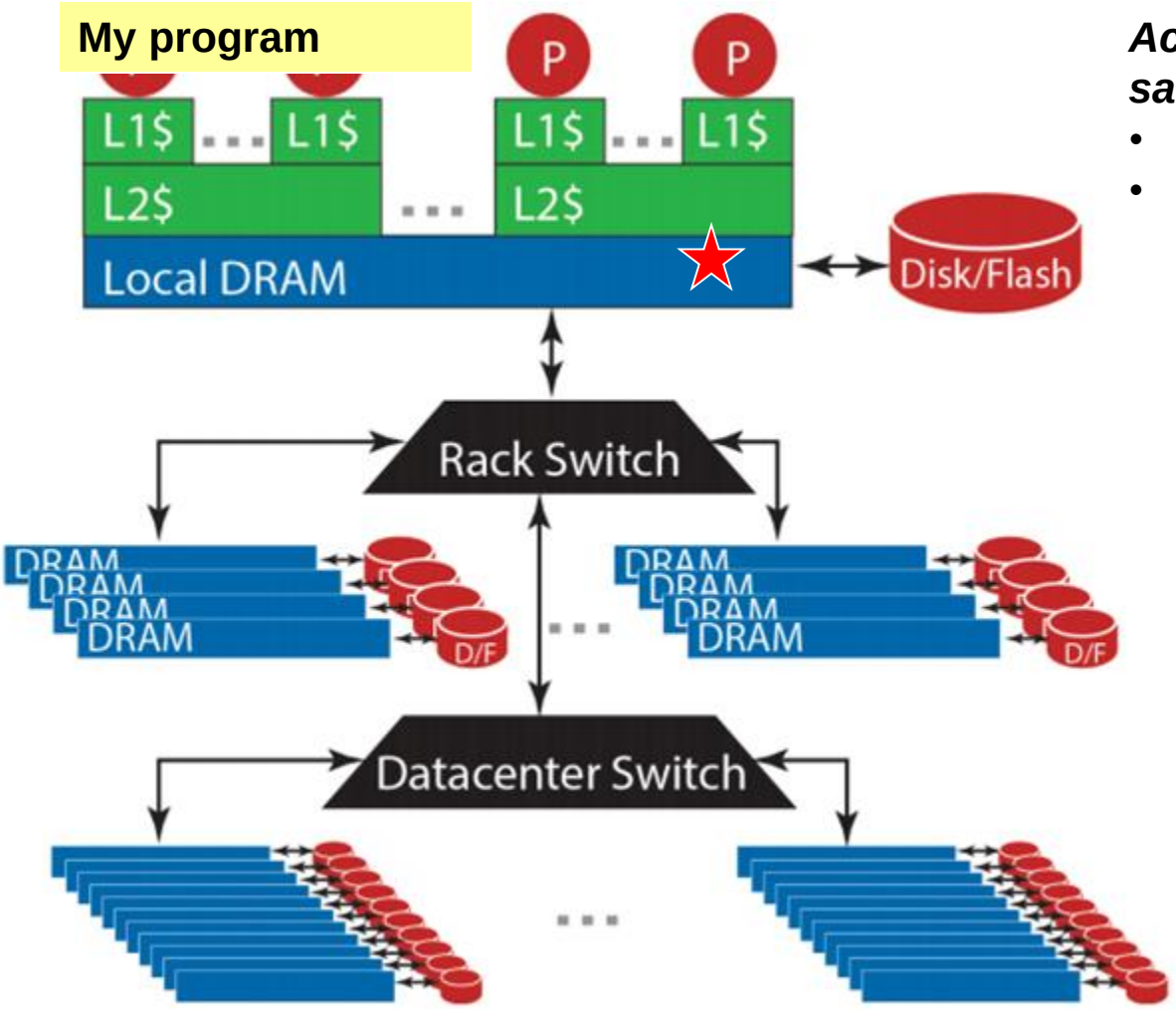
Adding many lanes to a highway: increases bandwidth, but does not decrease latency

Credit: Katie Hempenius (via Jake Archibald)

The Cost of Data Accesses in Data Center

- A data access may cross several components including hard disk, DRAM, rack switch and cluster switch etc.
- We will do some back-of-envelop calculations on the cost
  - Like algorithmic complexity analysis, we approximate for the scale or the order of magnitude, rather than the exact number.
  - Many details are simplified: a) an one-way communication (data -> program), b) no failures etc.
- Latency
  - = The sum of the latency caused in all components in the data access.
- Bandwidth
  - = The minimum of the bandwidths of all components in the data access.
  - Assume the hardware peak bandwidth
  - The actual achieved bandwidth depends on the algorithm.

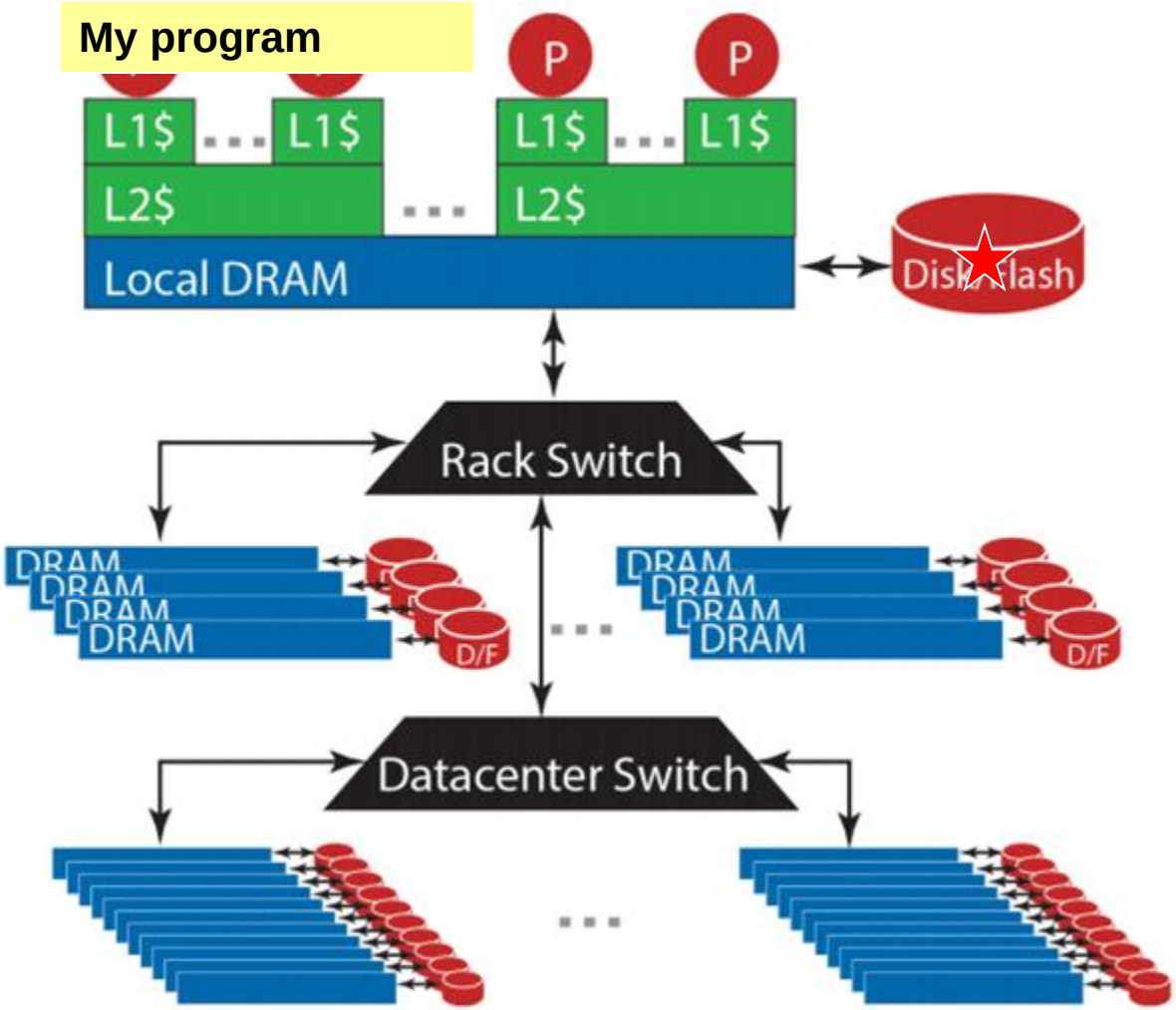
Data Accesses in Local Machine



Accessing the <u>DRAM</u> on the same machine:

- **Latency=100ns**
- **Bandwidth=20GB/sec**

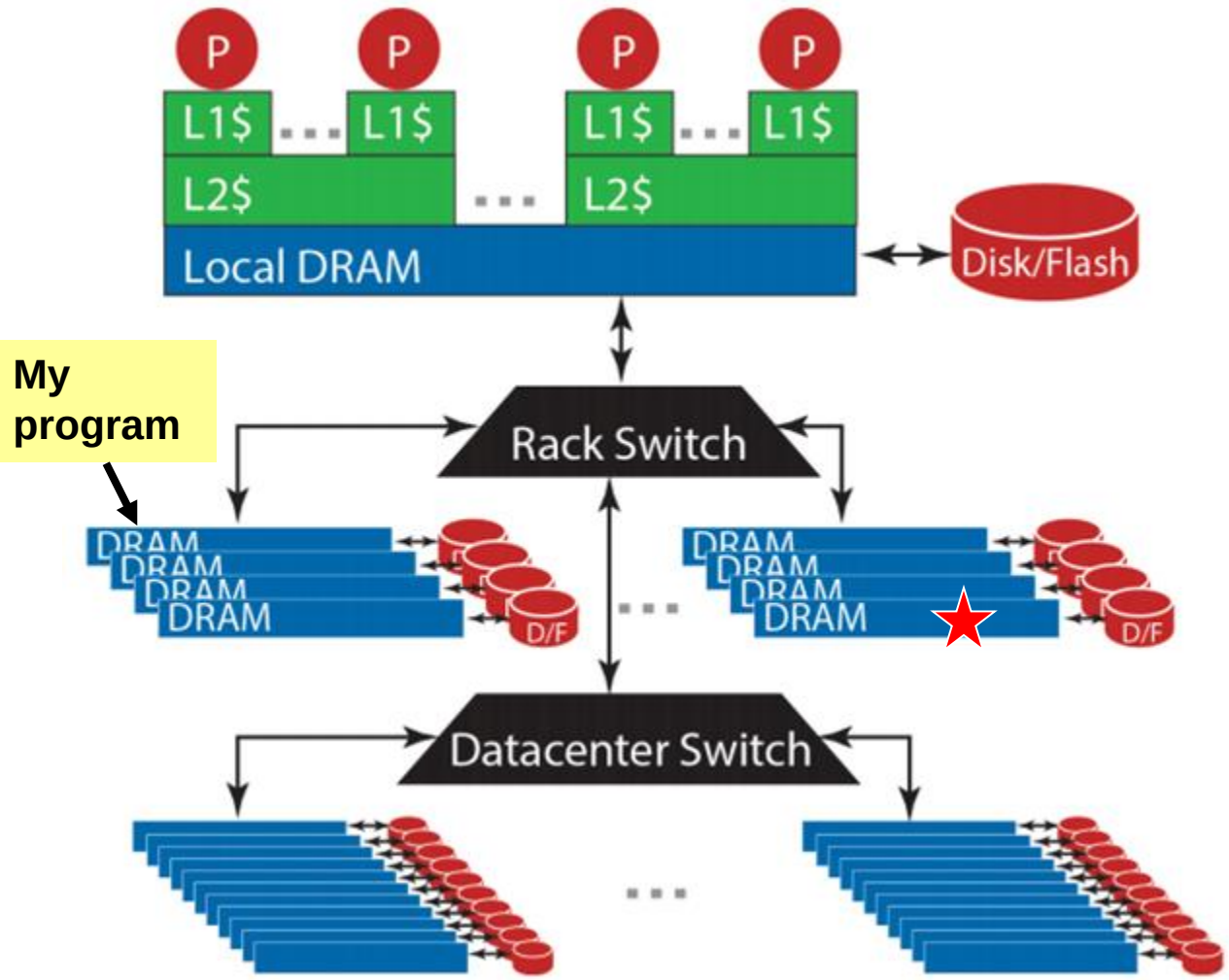
Data Accesses in Local Machine



Accessing the <u>disk</u> on the same machine:

- **Latency=10ms**
- **Bandwidth= 200MB/sec**

Data Accesses within Same Rack

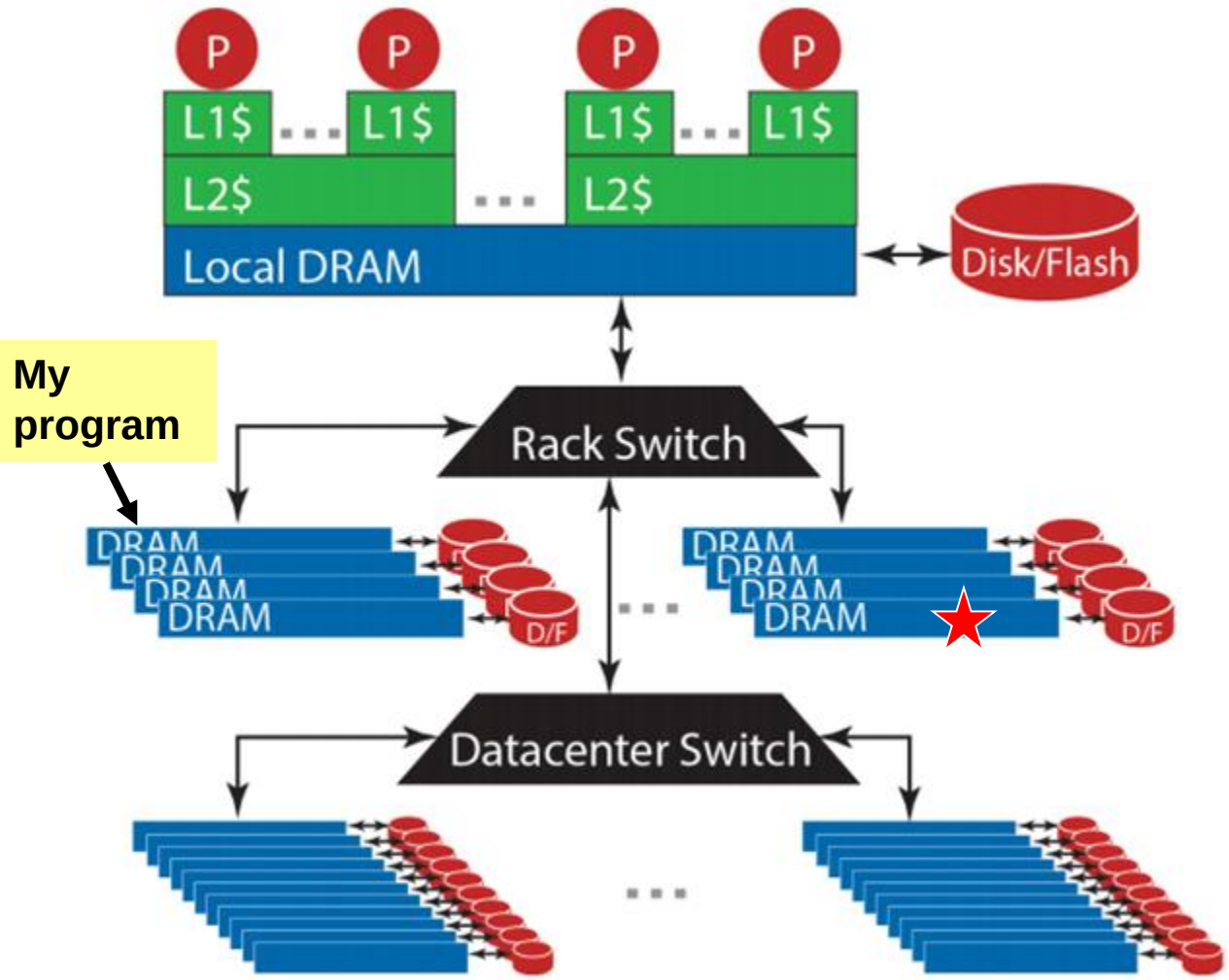


Rack switch (per port):
- Latency=300us
- Bandwidth=100MB/sec

Accessing the DRAM on the another machine in the same rack:

- **Latency= ?**
- **Bandwidth= ?**

Data Accesses within Same Rack

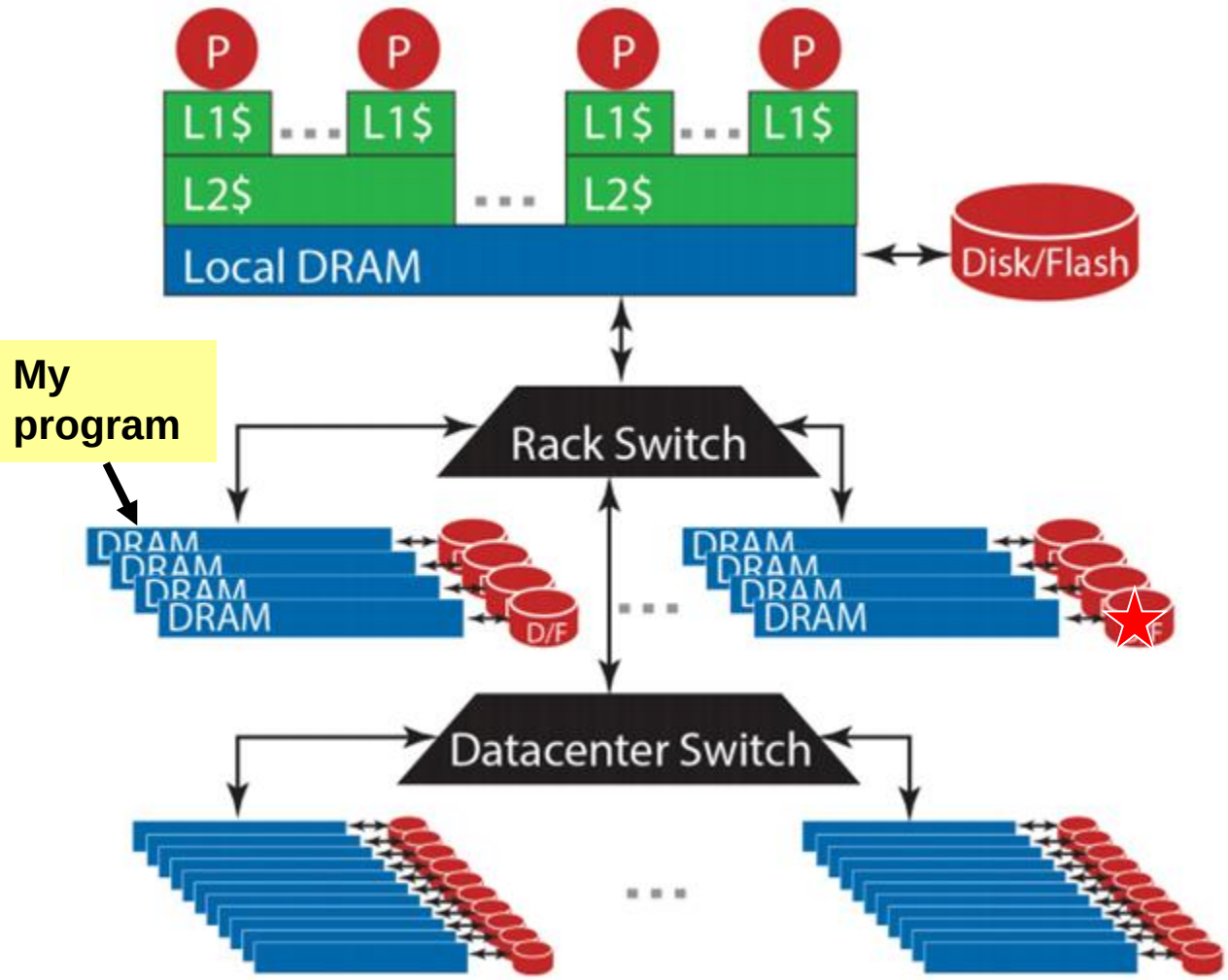


Rack switch (per port):
- Latency=300us
- Bandwidth=100MB/sec

Accessing the DRAM on the another machine in the same rack:

- **Latency= ~300 us**
- **Bandwidth= ~100MB/sec**

Data Accesses within Same Rack

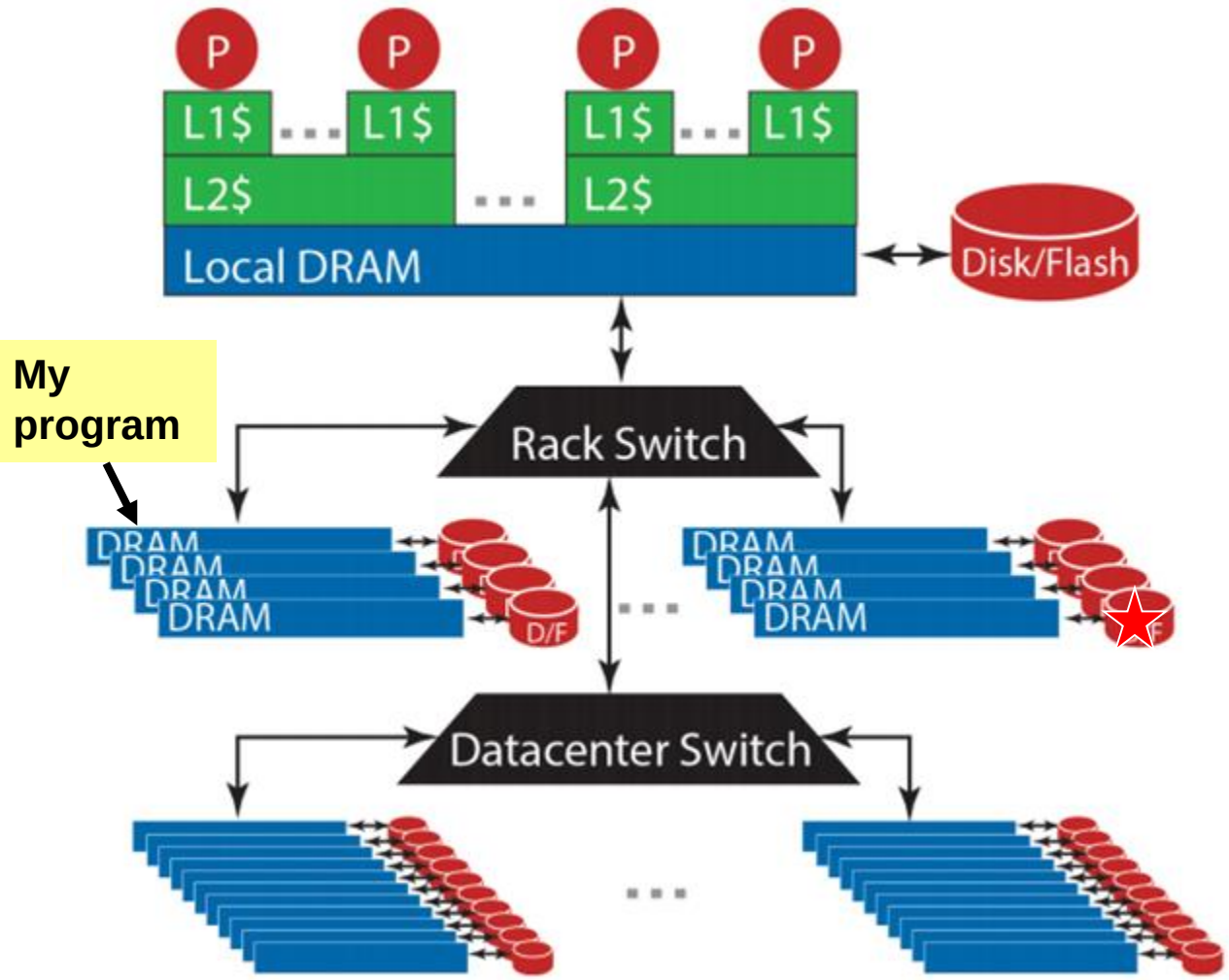


Rack switch (per port):
- Latency=300us
- Bandwidth=100MB/sec

Accessing the <u>disk</u> on the another machine in the same rack:

- **Latency= ?**
- **Bandwidth= ?**

Data Accesses within Same Rack

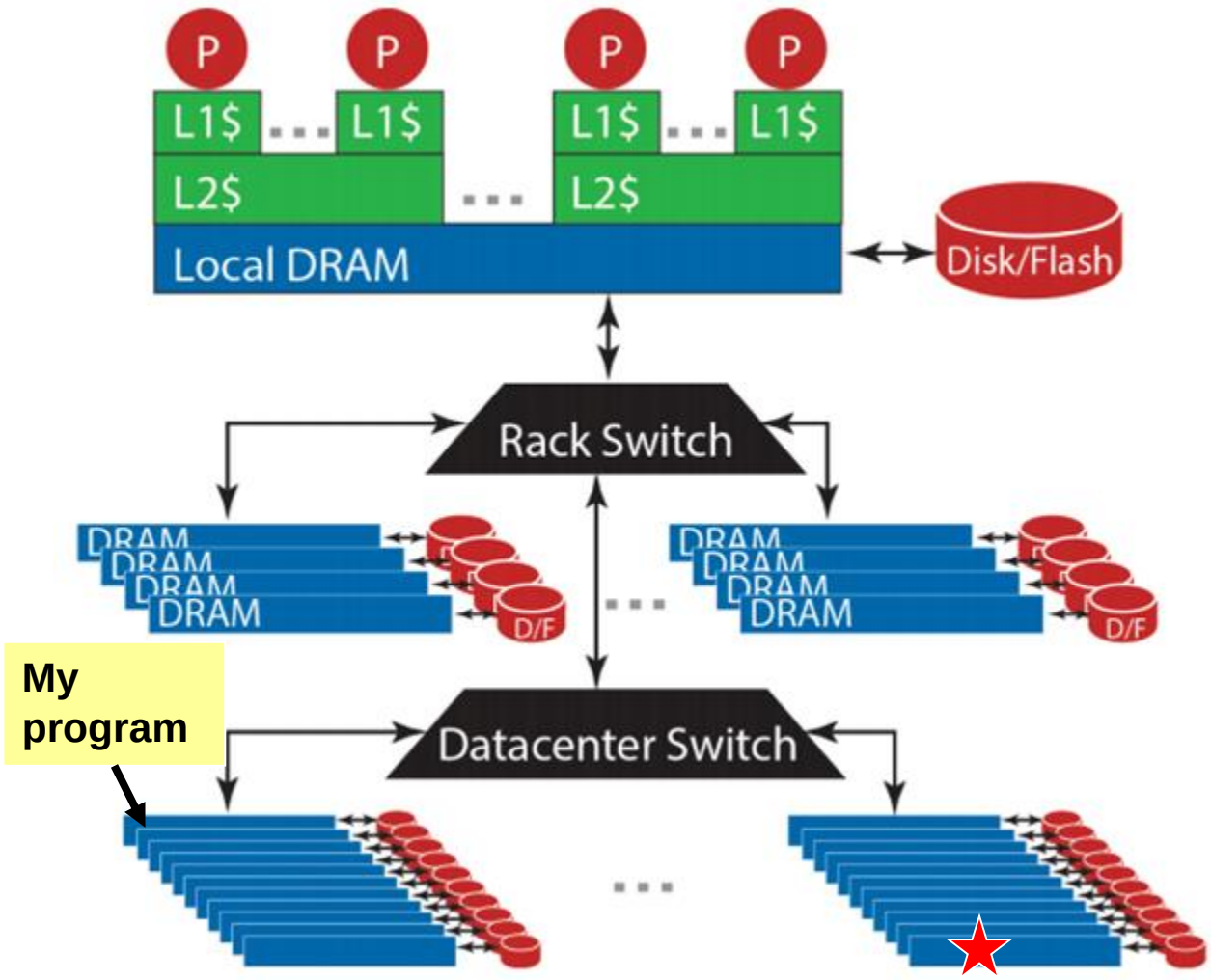


Rack switch (per port):
- Latency=300us
- Bandwidth=100MB/sec

Accessing the <u>disk</u> on the another machine in the same rack:

- **Latency= ~10ms**
- **Bandwidth= ~100MB/sec**

Data Accesses within Data Center (But in Different Racks)

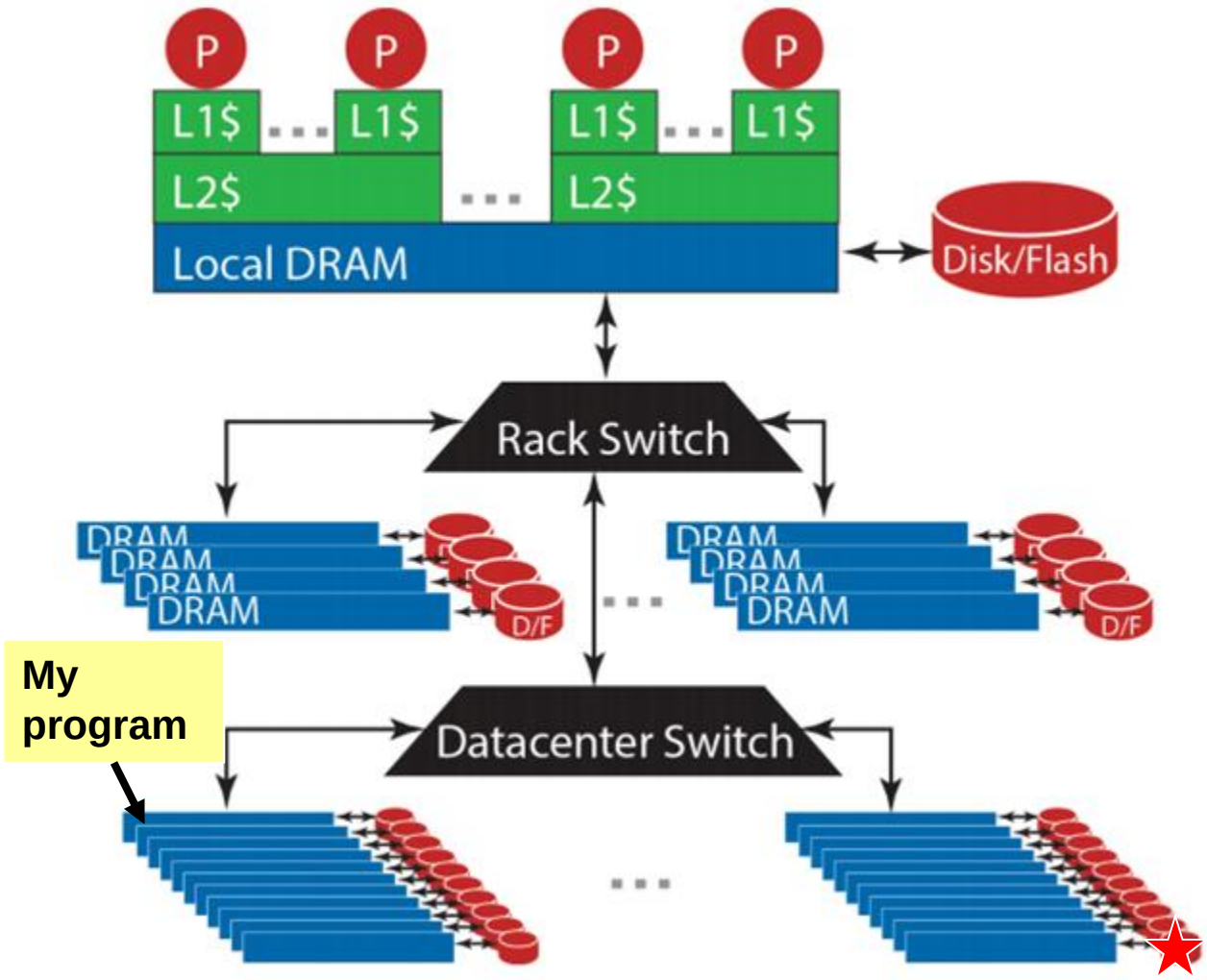


Datacenter switch (per port):
- Latency=500us
- Bandwidth=10MB/sec

Accessing the DRAM on the another machine in different rack:

- **Latency= ~(500+300*2)=1300 us**
- **Bandwidth= ~10MB/sec**

Data Accesses within Data Center (But in Different Racks)

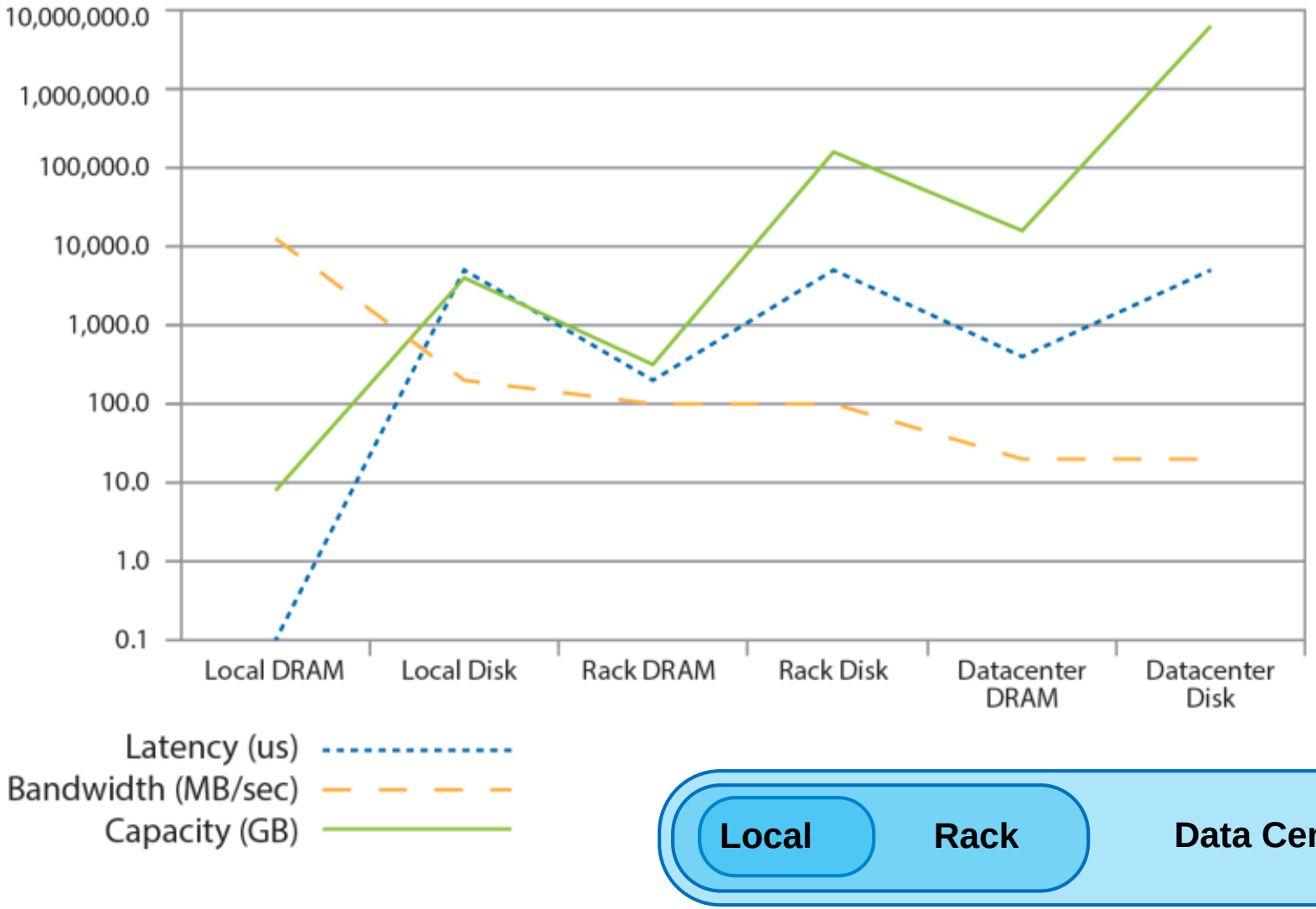


Datacenter switch (per port):
- Latency=500us
- Bandwidth=10MB/sec

Accessing the <u>hard disk</u> on the another machine in different rack:

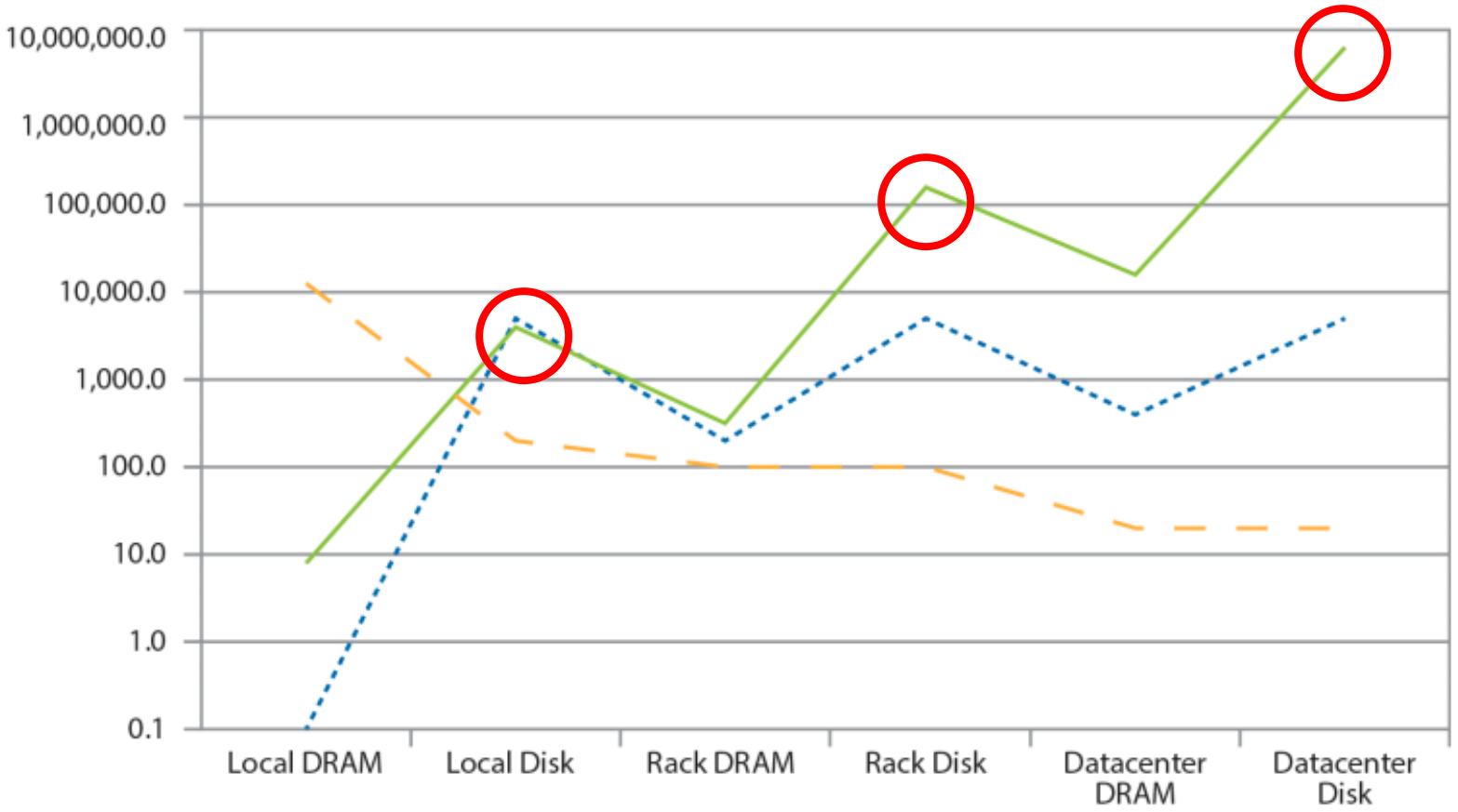
- **Latency= ~10ms**
- **Bandwidth= ~10MB/sec**

The Cost of Moving Data Around Data Center



Source: Barroso and Urs Hölzle (2013)

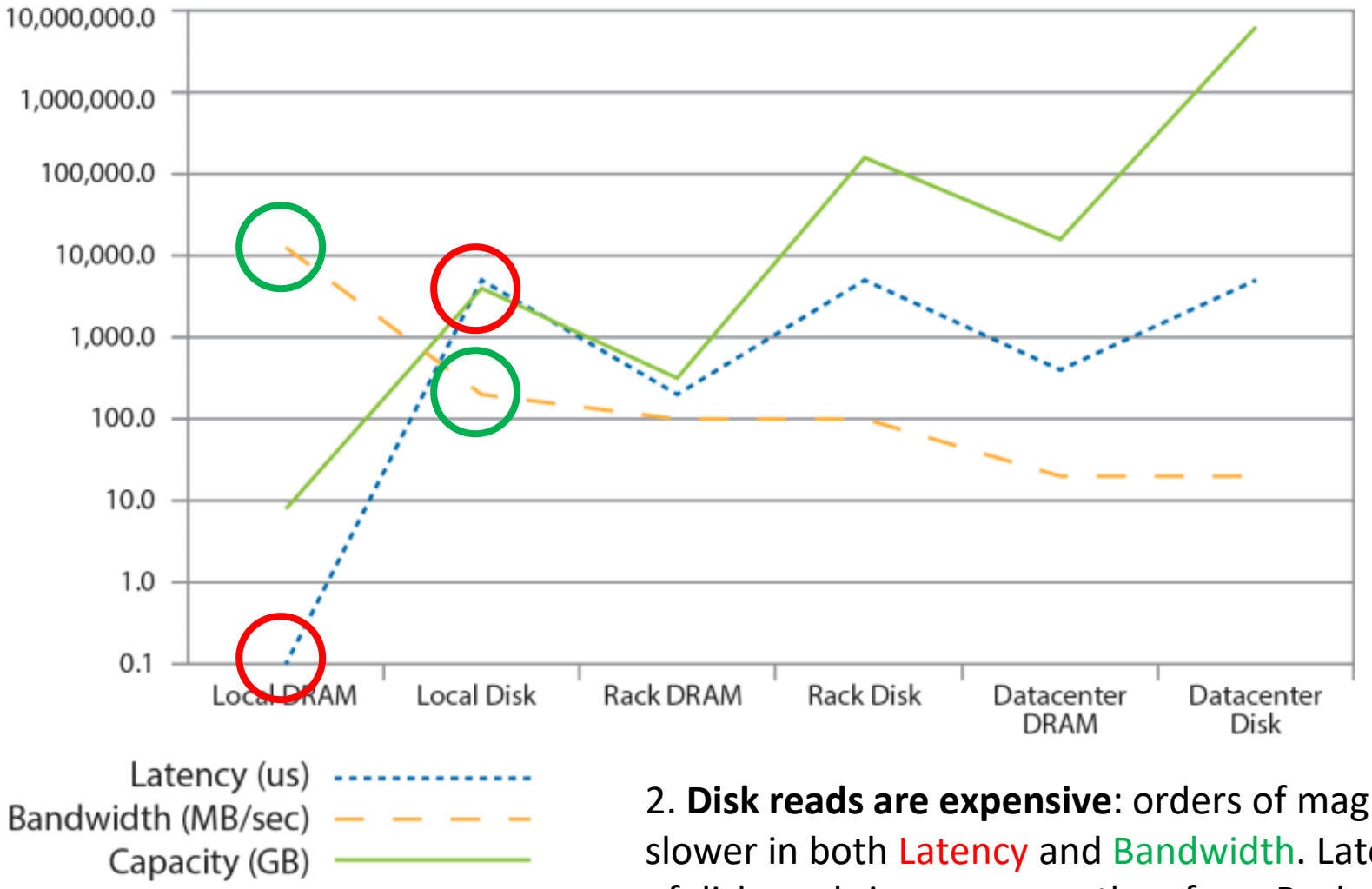
The Cost of Moving Data Around Data Center



1. **Storage hierarchy**: *capacity* increases as we go from Local Server to Rack to Datacenter. But, communication costs increase, so *latency* increases and *bandwidth* decreases. Takeaway: sending data further on a network is costly!

Source: Barroso and Urs Hölzle (2013)

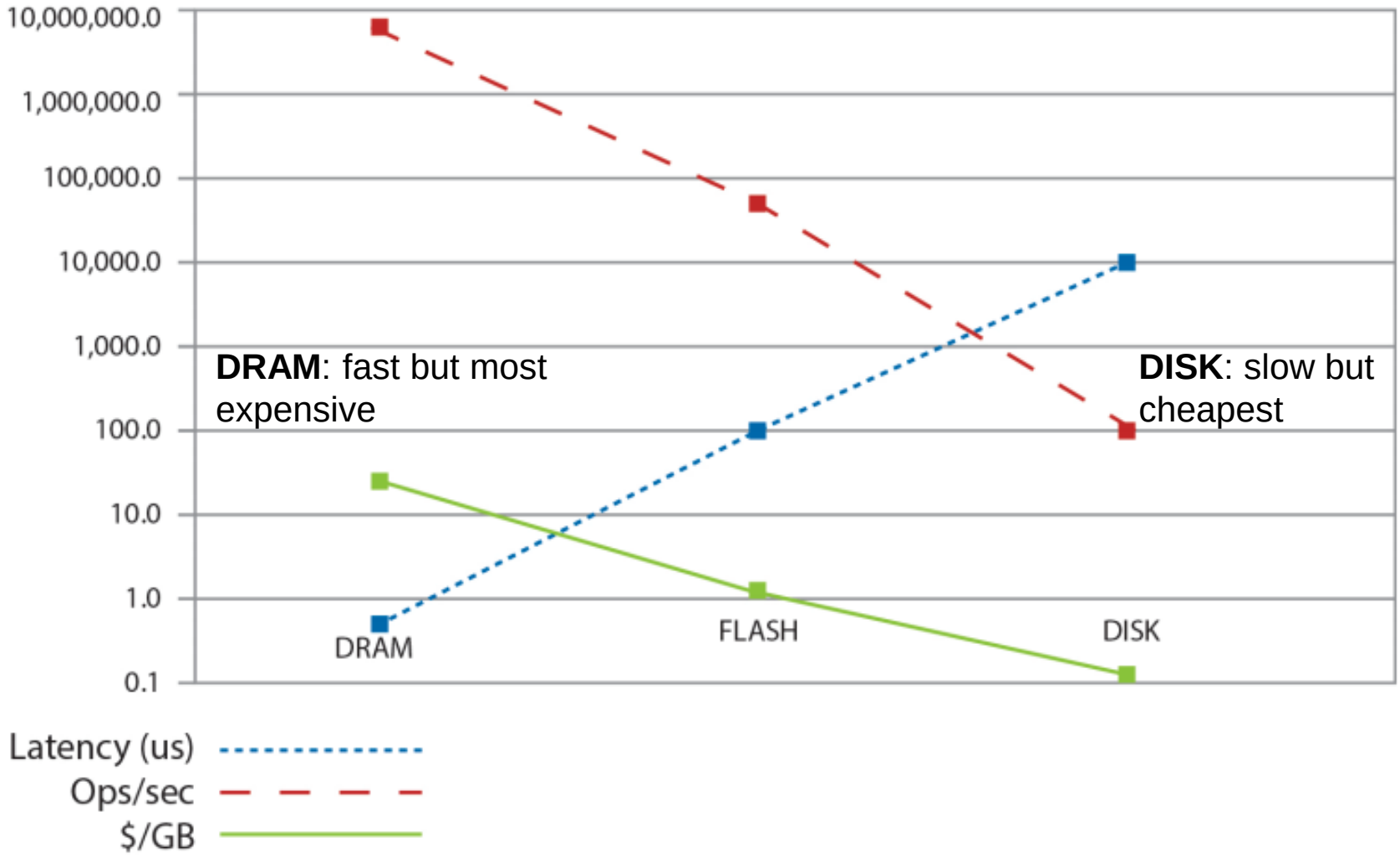
The Cost of Moving Data Around Data Center



2. **Disk reads are expensive**: orders of magnitude slower in both Latency and Bandwidth. Latency of disk reads is even more than from Rack / Datacenter DRAM.

Source: Barroso and Urs Hölzle (2013)

The Cost of Moving Data Within a Server



Source: Barroso and Urs Hölzle (2013)

MCQ Quiz

Sensor readings have precision issues or errors. Such scenario shows _____ of big data.

- (A) Volume
- (B) Velocity
- (C) Veracity
- (D) Variety
- (E) None of the above answers

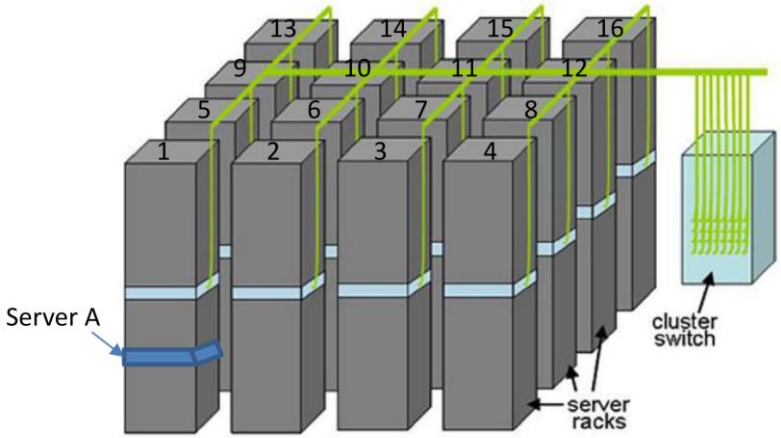
Answer: C, Veracity means the accuracy of the data.

MCQ Quiz

The architecture of a commercial data center is illustrated in the below figure. The number on the top of each rack is the identifier of each rack. Users can run Hadoop or Spark jobs in the data center.

Suppose a program P is running on Server A in Rack 1. Denote the latency of P accessing a byte in the hard disk of Server A, a byte in the hard disk of the other servers in Rack 1, and a byte in the hard disk of the other server in Rack 16 is L1, L2 and L3, respectively. Which of the following statements are True?

- S1. L3 can be ten times larger than L2.
- S2. L2 can be ten times larger than L1.
- S3. L3 is roughly the same as L2.
- S4. L2 is roughly the same as L1.



Answer: S3 and S4 (hard disk is the slowest)

Outline

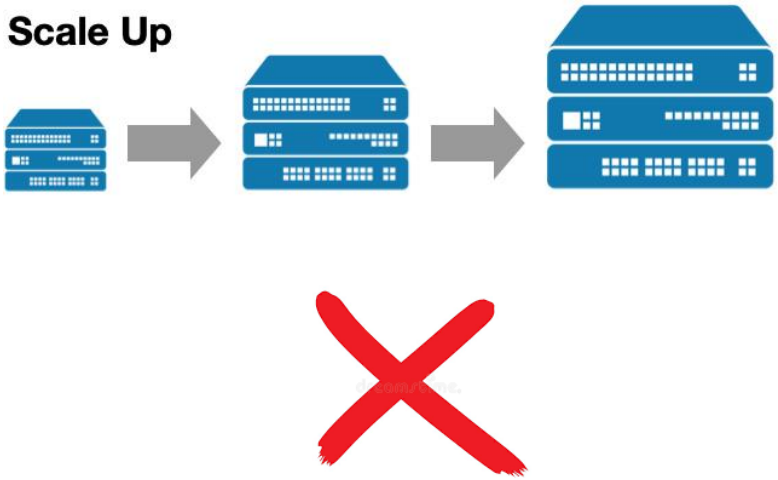
- Data center architecture
- **The four “Big Ideas”**
- Abstractions for big data systems

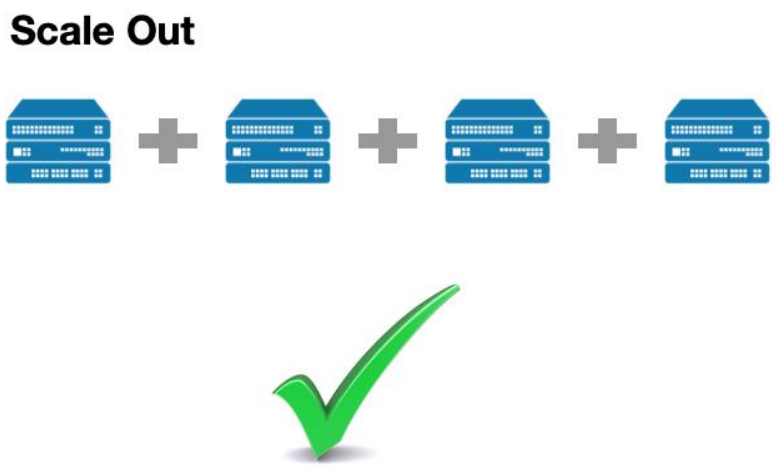
“Big Ideas” of Massive Data Processing in Data Centers

- Scale “out”, not “up”
  - scale ‘out’ = combining many cheaper machines; scale ‘up’= increasing the power of each individual machine
  - Also called ’horizontal’ vs ‘vertical’ scaling
- Seamless scalability
  - E.g. if processing a certain dataset takes 100 machine hours, ideal scalability is to use a cluster of 10 machines to do it in about 10 hours.
- Move processing to the data
  - Clusters have limited bandwidth: we should move the task to the machine where the data is stored
- Process data sequentially, avoid random access
  - Seeks are expensive, disk throughput is reasonable

Scale “out”, not “up”

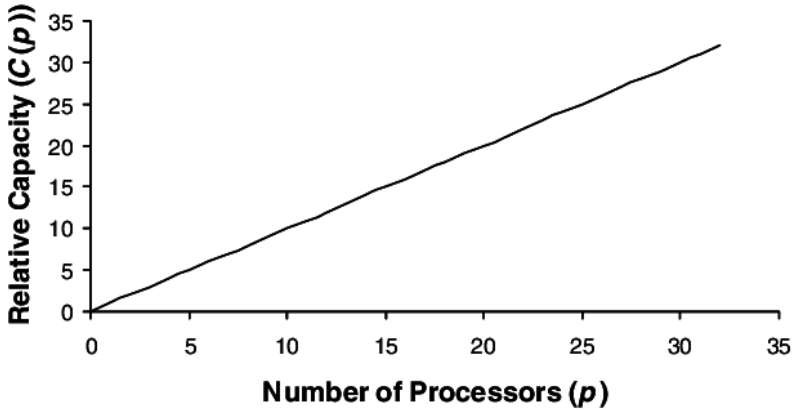
- Scaling up = adding further resources, like hard drives and memory, to increase the computing capacity of physical servers.
- Scaling out = adding more servers to your architecture to spread the workload across more machines.



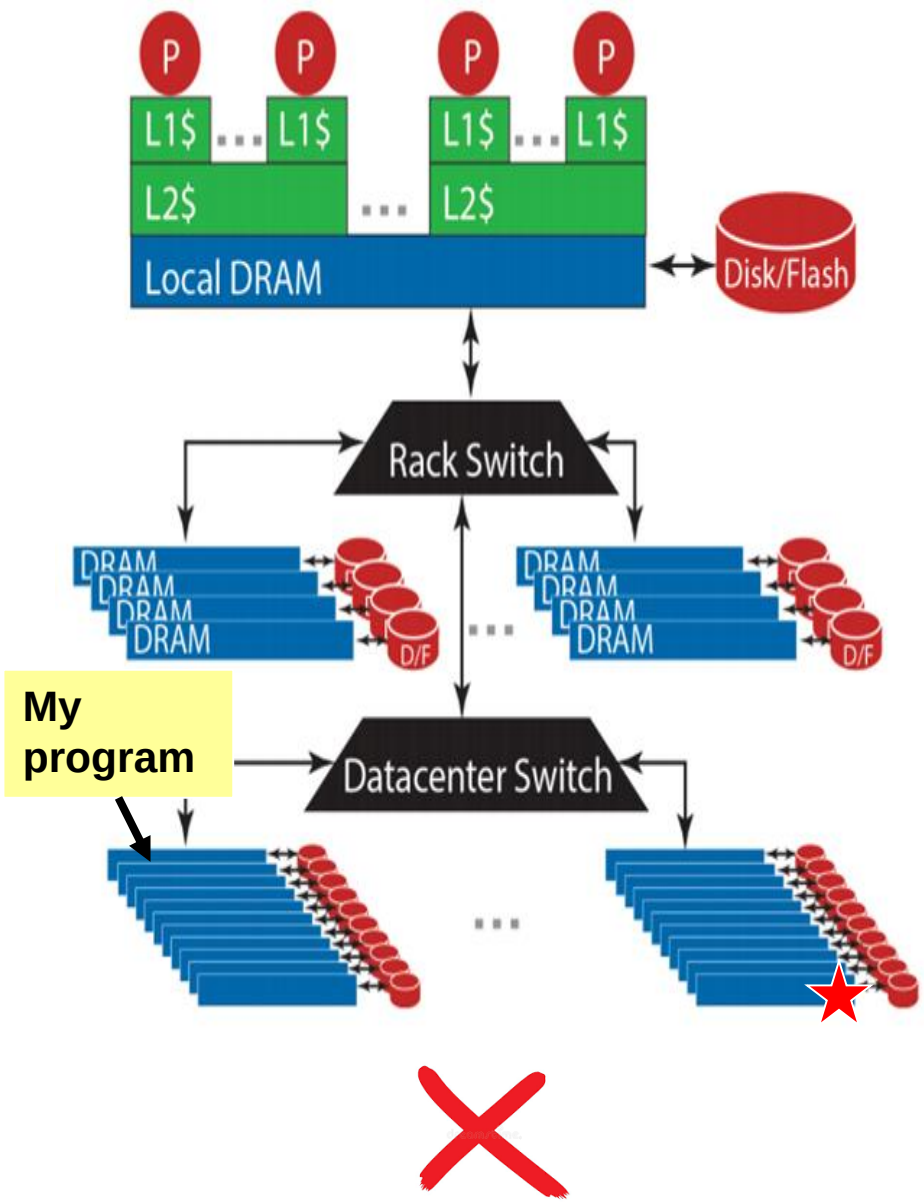


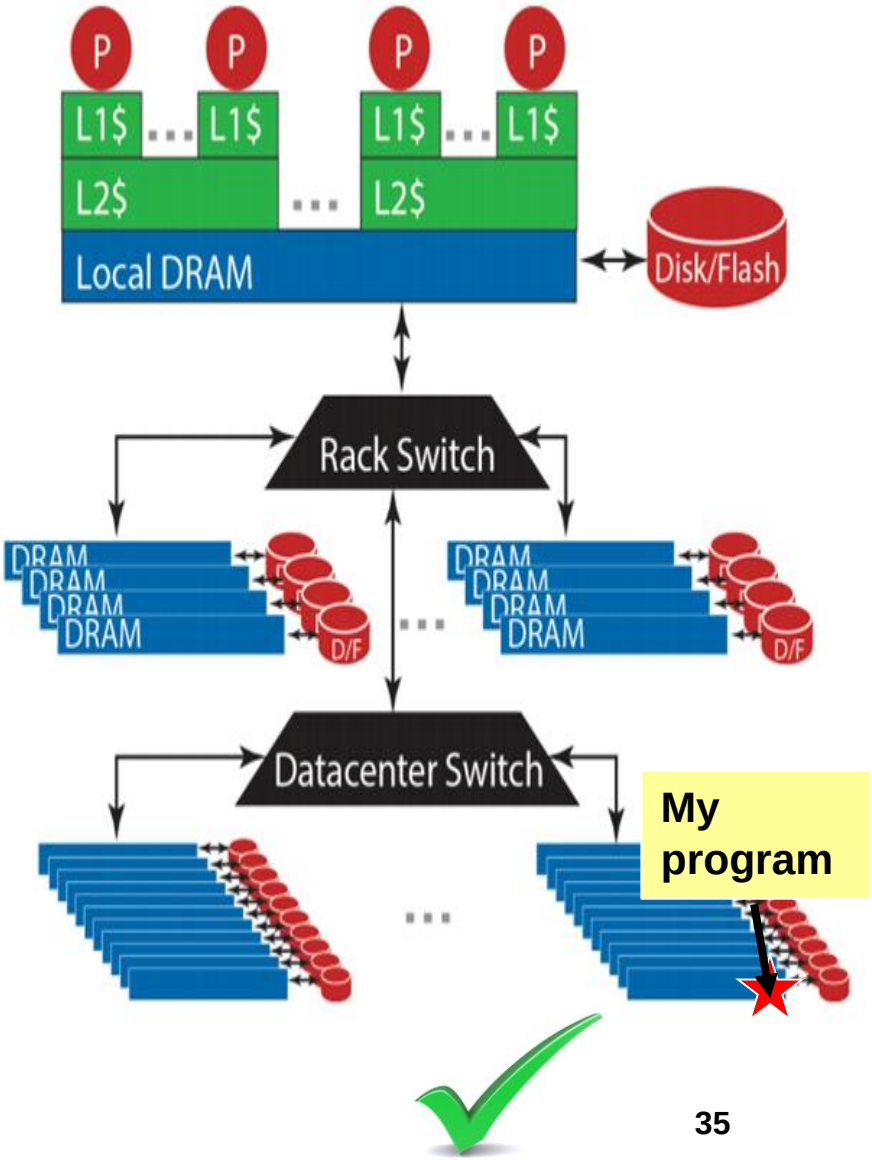
Seamless Scalability

- Think about aggregated bandwidth
  - 1 machine: 200MB/sec disk, 20GB/sec DRAM
  - 10 machine: 2 GB/sec disk, 200 GB/sec DRAM
  - 100 machine: 20 GB/sec disk, 2 TB/sec DRAM
- If our design scales linearly to #machine, we can trade more machines with better performance.



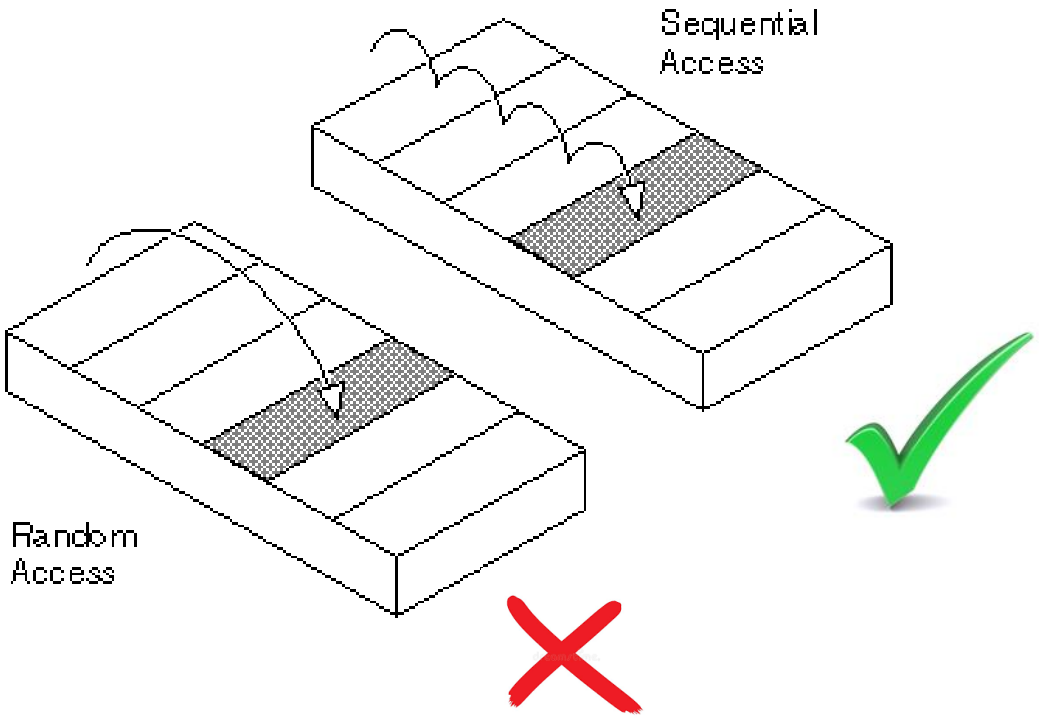
Move processing to the data





Sequential Accesses vs. Random Accesses

- Take hard disk as an example
- Random access: 10ms for 4KB = 400KB/sec
- Sequential access: 200MB/sec



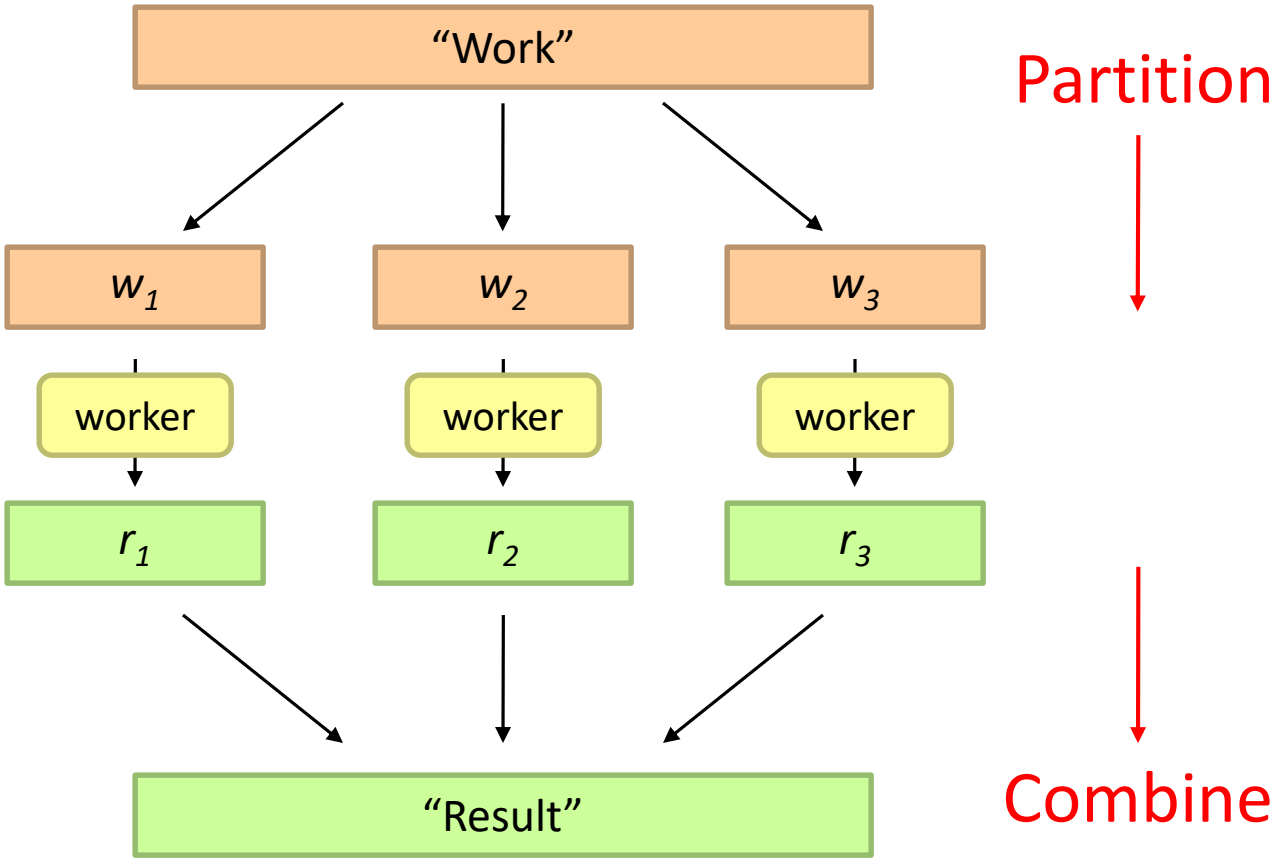
Outline

- Data center architecture
- The four “Big Ideas”
- **Abstractions for big data systems**

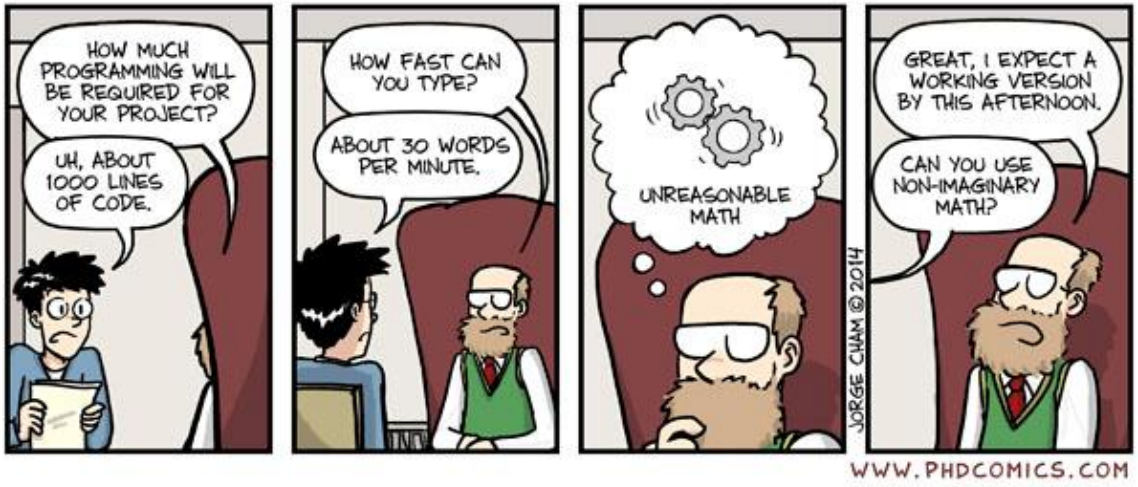
Motivation: Google Example

- 20+ billion web pages x 20KB = 400+ TB
- 1 computer reads 30-35 MB/sec from disk
  - ~4 months to read the web
- ~1,000 hard drives to store the web
- Takes even more to **do** something useful with the data!
- Infrastructure: cloud + data centers
  - Cluster of commodity nodes
  - Commodity network (ethernet) to connect them
- Solution:
  - Parallelization + divide and conquer

Divide and Conquer



Challenges



- How do we assign work units to workers?
- What if we have more work units than workers?
- What if workers need to share partial results?
- How do we aggregate partial results?
- How do we know all the workers have finished?
- What if workers die/fail?

Challenge 1: Machine Failures

- One server may stay up 10 years (~3650 days)
- If you have 3650 servers, expect to loose 1/day
- People estimated Google had ~2.5 million machines in 2016
  - ~700 machines fail every day!

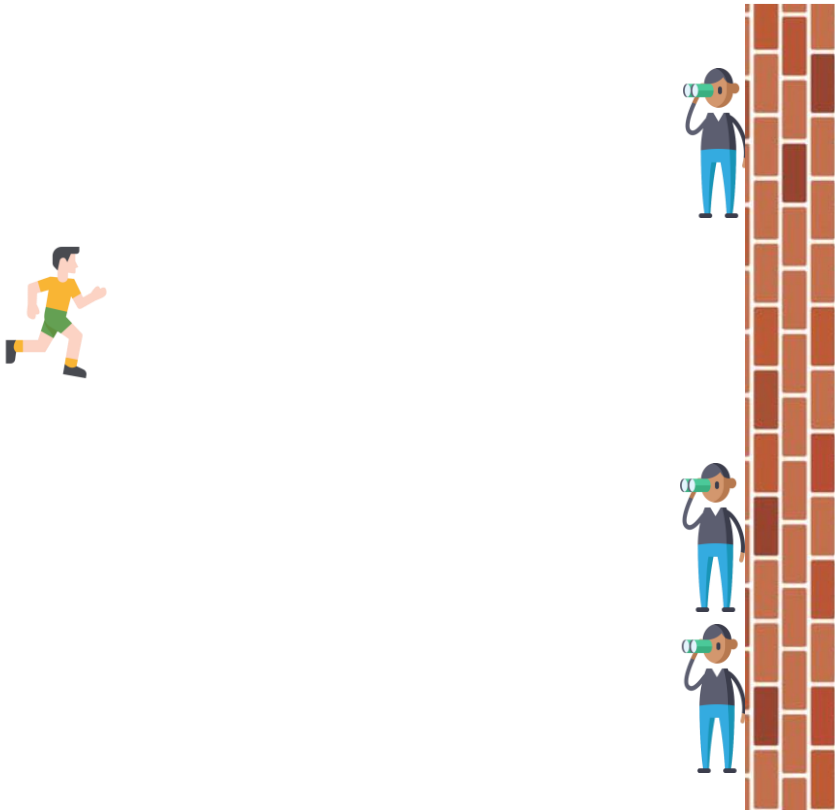
Challenge 2: Synchronization

- Difficult because
  - We don't know the order in which workers run
  - We don't know when workers interrupt each other
  - We don't know when workers need to communicate partial results
  - We don't know the order in which workers access shared data
- Thus, we need (note: not required knowledge for class)
  - Semaphores (lock, unlock)
  - Conditional variables (wait, notify, broadcast)
  - Barriers
- Still, lots of problems:
  - Deadlock, livelock, race conditions...
  - Dining philosophers, sleeping barbers...

Source: Ricardo Guimarães Herrmann

Barrier

- Simple tool used when multiple processes are running at the same time
- It ensures that every process must stop at this point until all processes have reached the barrier



Challenge 3: Programming Difficulty

- Concurrency is difficult to reason about
  - At the scale of datacenters and across datacenters
  - In the presence of failures
  - In terms of multiple interacting services
- Not to mention debugging…
- The reality:
  - Lots of one-off solutions, custom code
  - Write you own dedicated library, then program with it
  - Burden on the programmer to explicitly manage everything

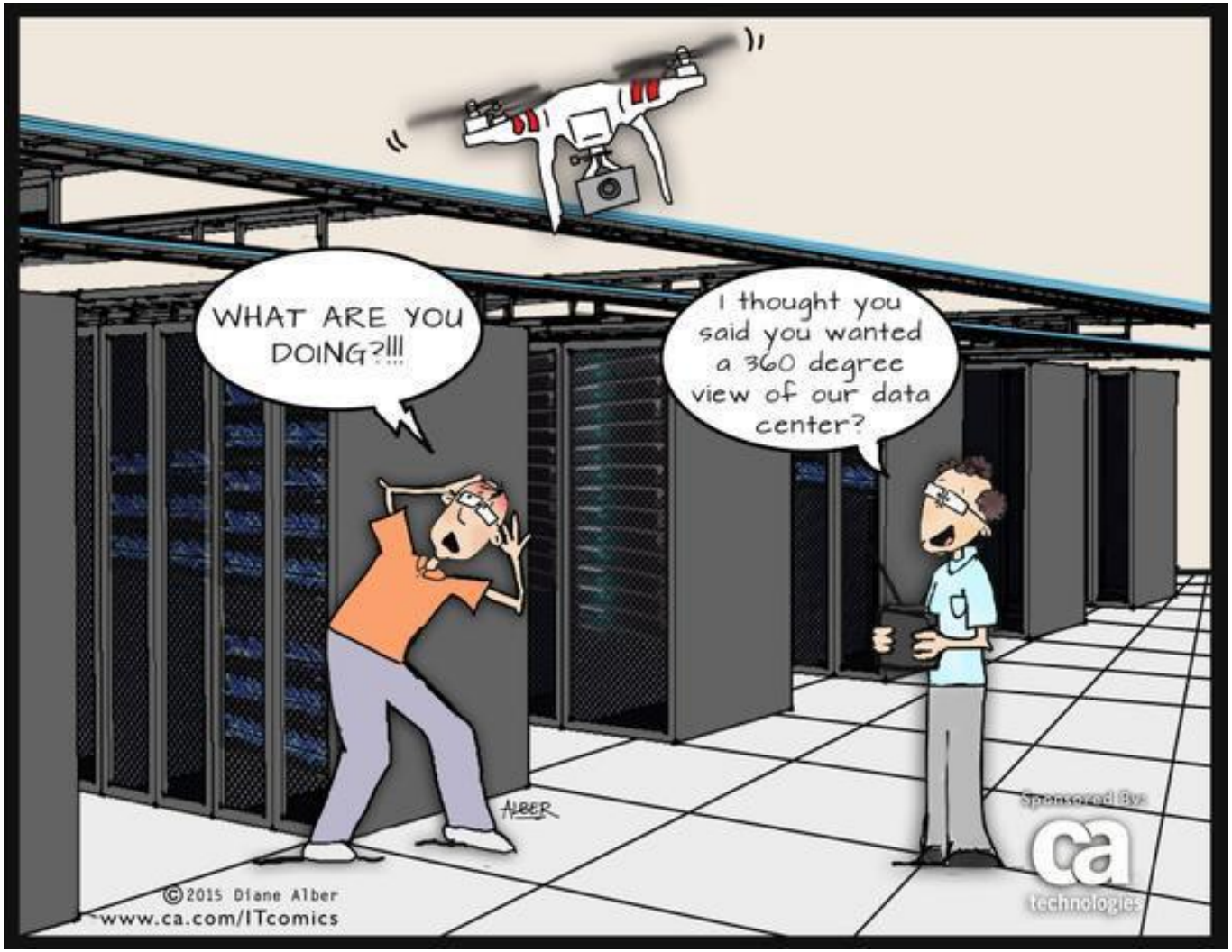
The datacenter *is* the computer

- It's all about the right level of abstraction
  - Moving beyond the single machine architecture
  - What's the "instruction set" (or "API") of the datacenter computer?
- Hide system-level details from the developers
  - No more race conditions, lock contention, etc.
  - No need to explicitly worry about reliability, fault tolerance, etc.
- Separating the *what* from the *how*
  - Developer specifies the computation that needs to be performed
  - Execution framework ("runtime") handles actual execution

Take-away

- “The data centre is the computer”.
- Four “big ideas” are the design principles of big data systems on the current hardware.
- Further readings:
  - Chapter 1. Jimmy Lin and Chris Dyer. 2020. Data-Intensive Text Processing with Mapreduce. Morgan and Claypool Publishers. https://lintool.github.io/MapReduceAlgorithms/MapReduce-book-final.pdf

Questions?


WHAT ARE YOU DOING?!!!
I thought you said you wanted a 360 degree view of our data center?
ALBER
©2015 Diane Alber
www.ca.com/ITcomics
Sponsored By
ca
technologies