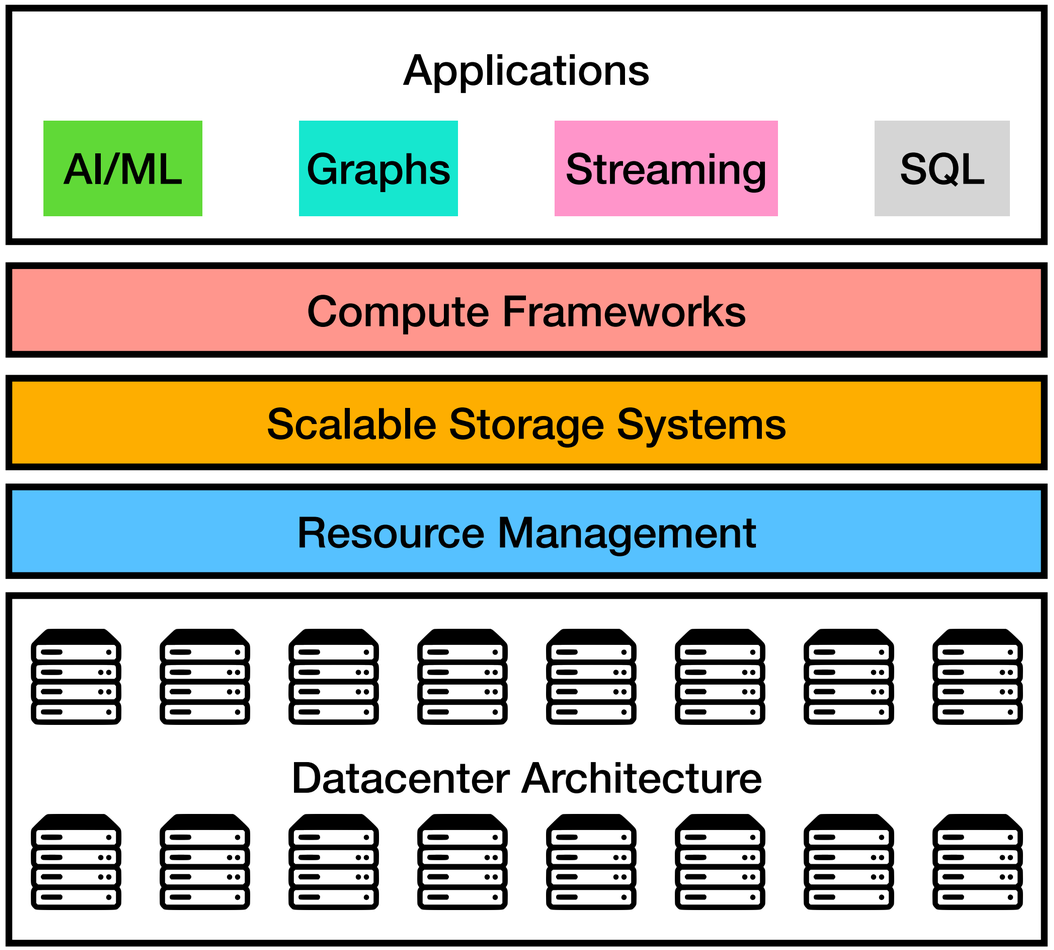
Characterizing Datacenter Architecture & Applications

CPSC 438/538: Big Data Systems

Anurag Khandelwal

Yale

Today's Agenda



Big Data System Stack

What they look like & what are their characteristics

To cater to

Goal: *Understand why they are designed the way they are*

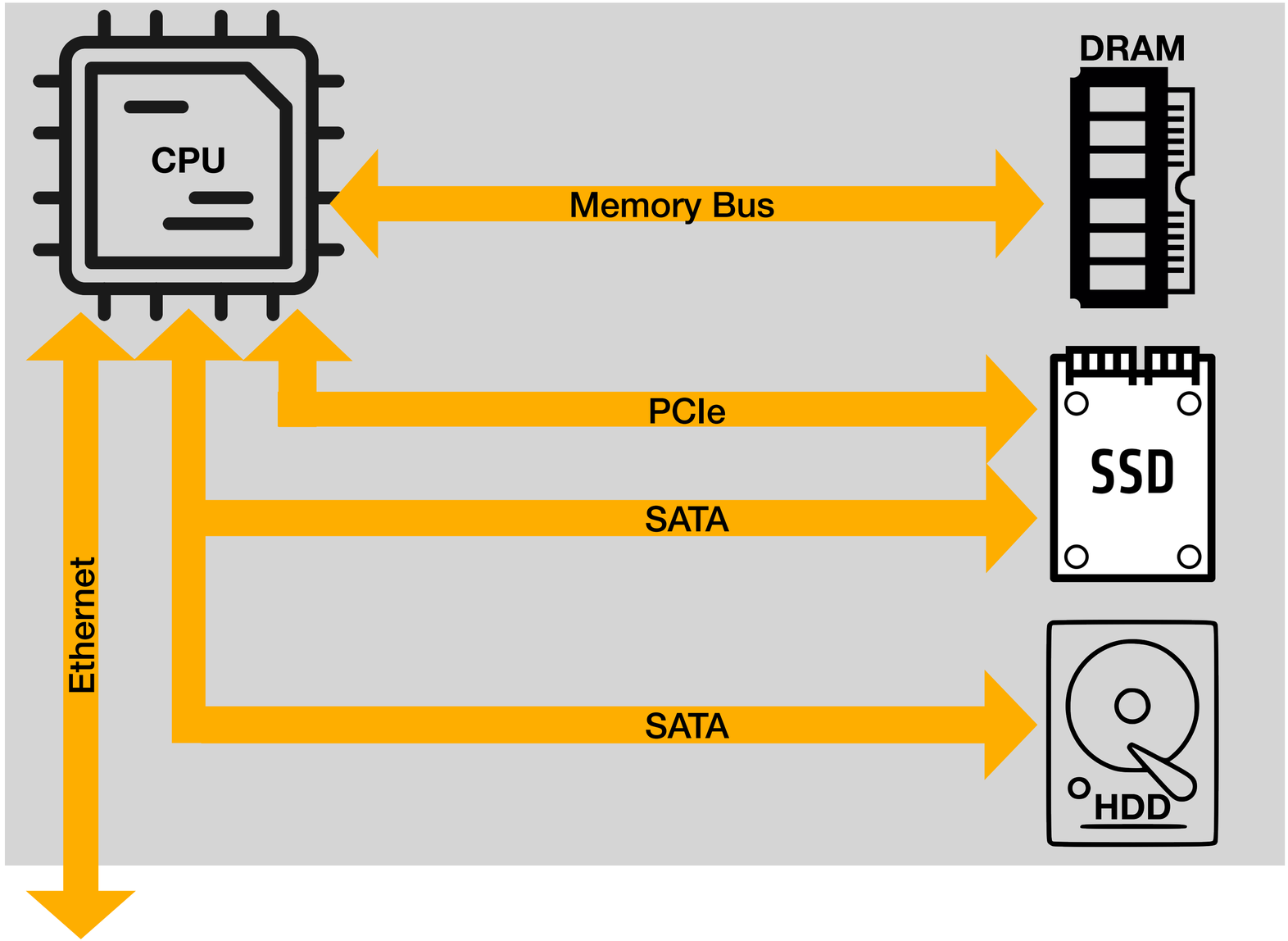
By leveraging

What this looks like & how is hardware evolving

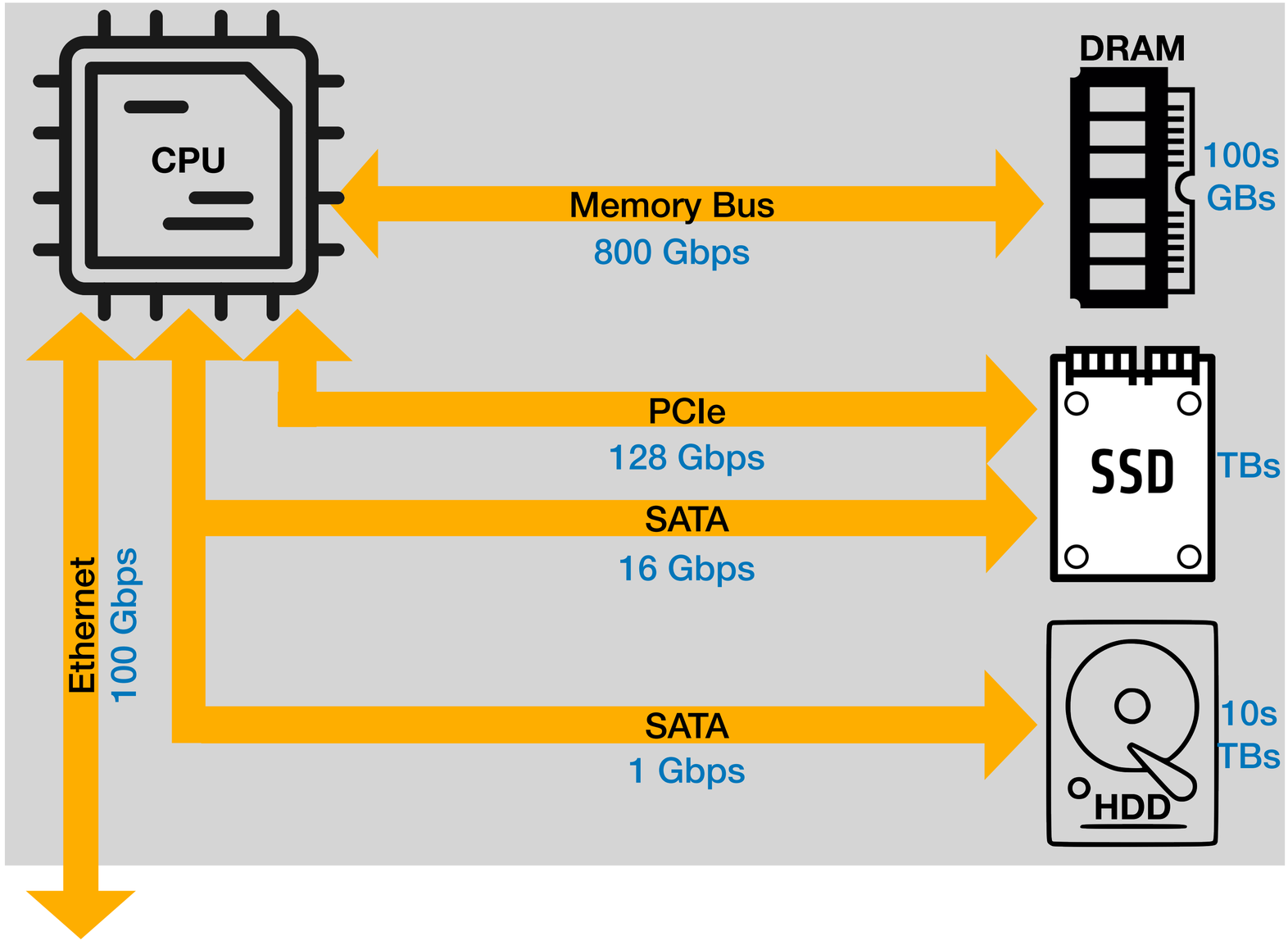
And possible research projects!

Hardware Trends & Datacenter Architecture

Building Block of A Datacenter: Server



Building Block of A Datacenter: Server



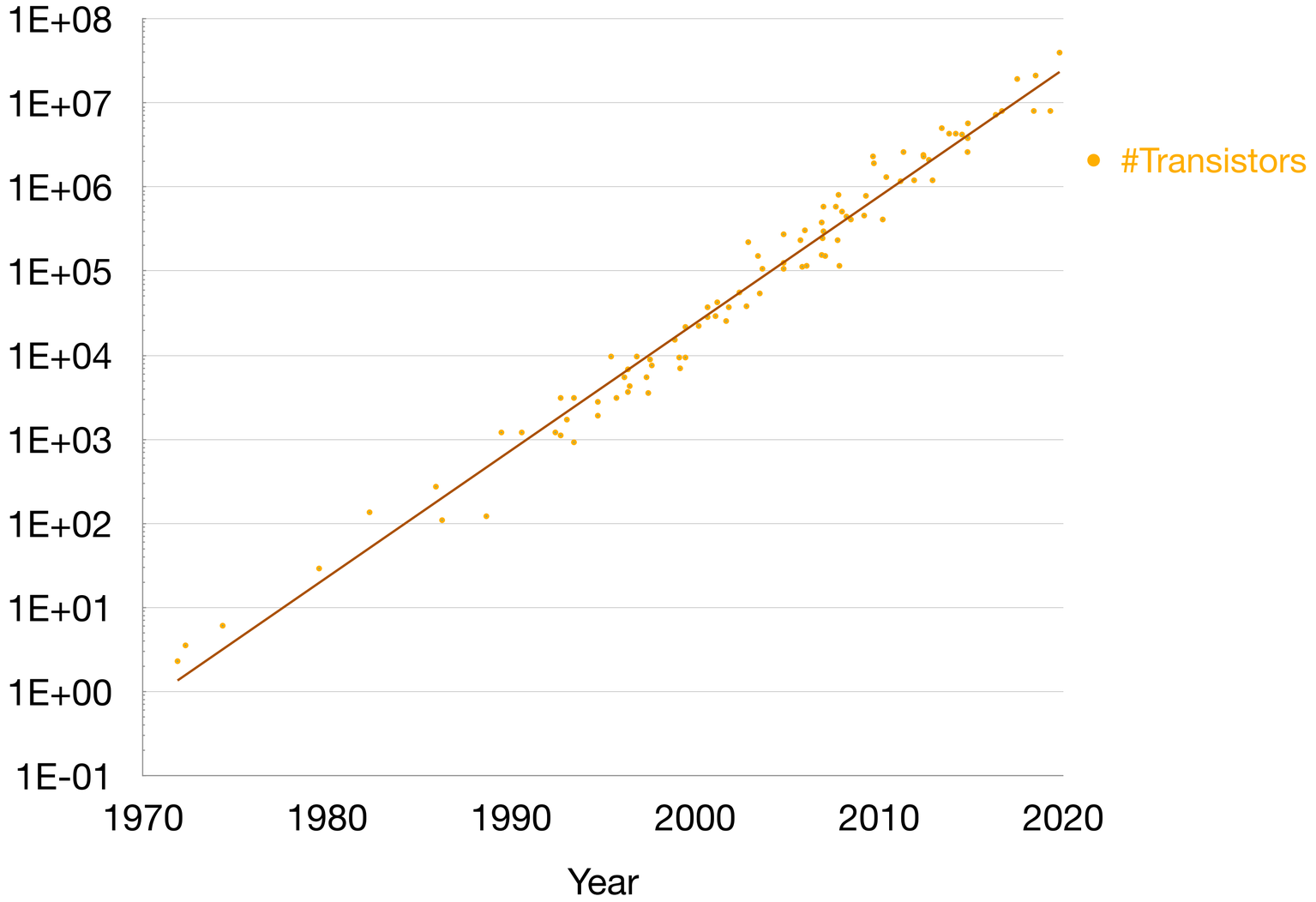
“Skate where the puck is going, not where it’s been.”

- Walter Gretzky

CPU Performance

Conventional Wisdom: Doubles every 2 years

- **Moore's Law (1965)**
  - Intel founder **Gordon Moore**
  - #Transistors on microchip doubles every 2 years
  - **But is ending now…**
  - **Physics strikes:** At 7nm transistors, not much more we can pack…

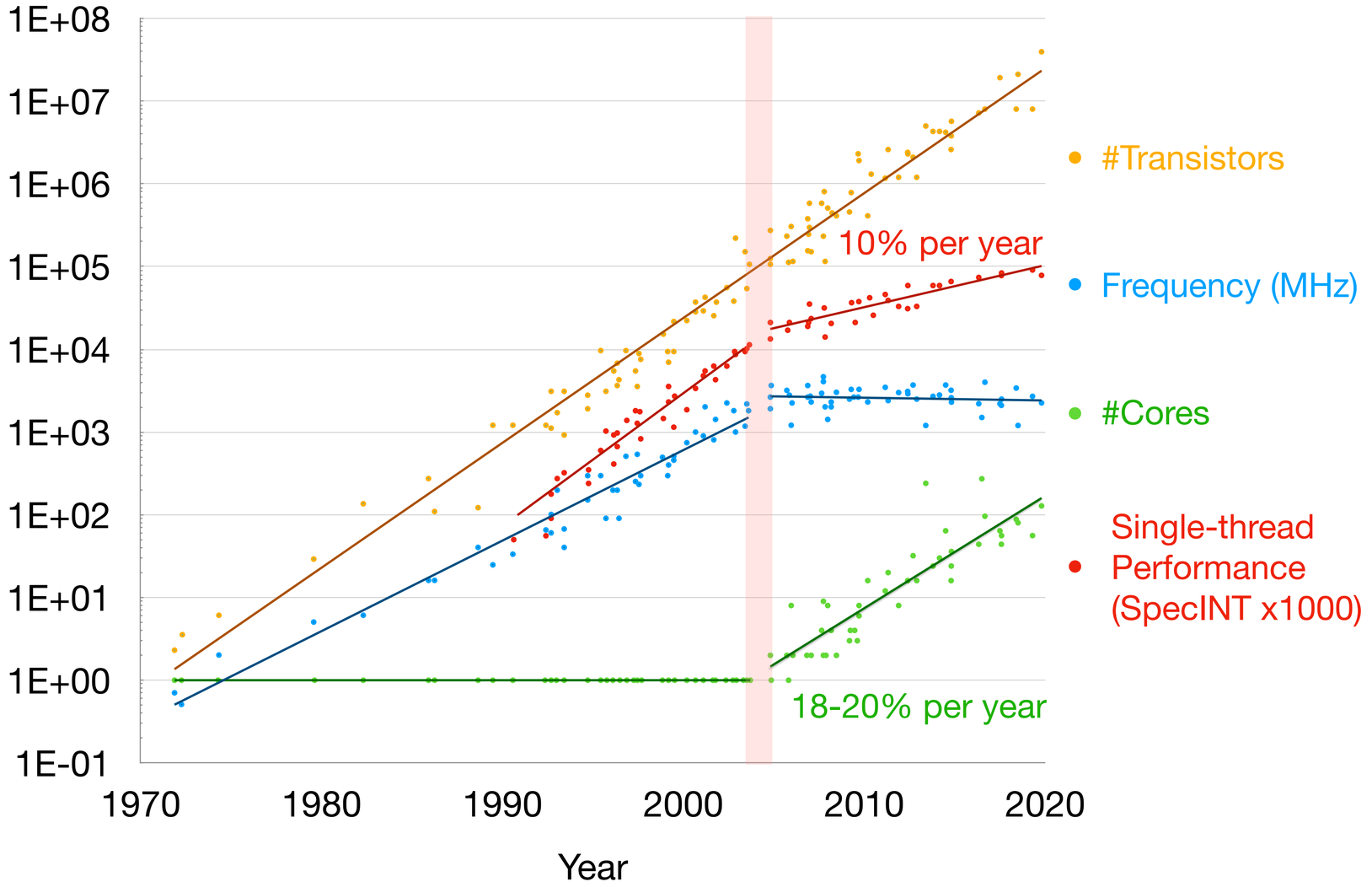


Microprocessor Trend Data, https://github.com/karlrupp/microprocessor-trend-data

CPU Performance

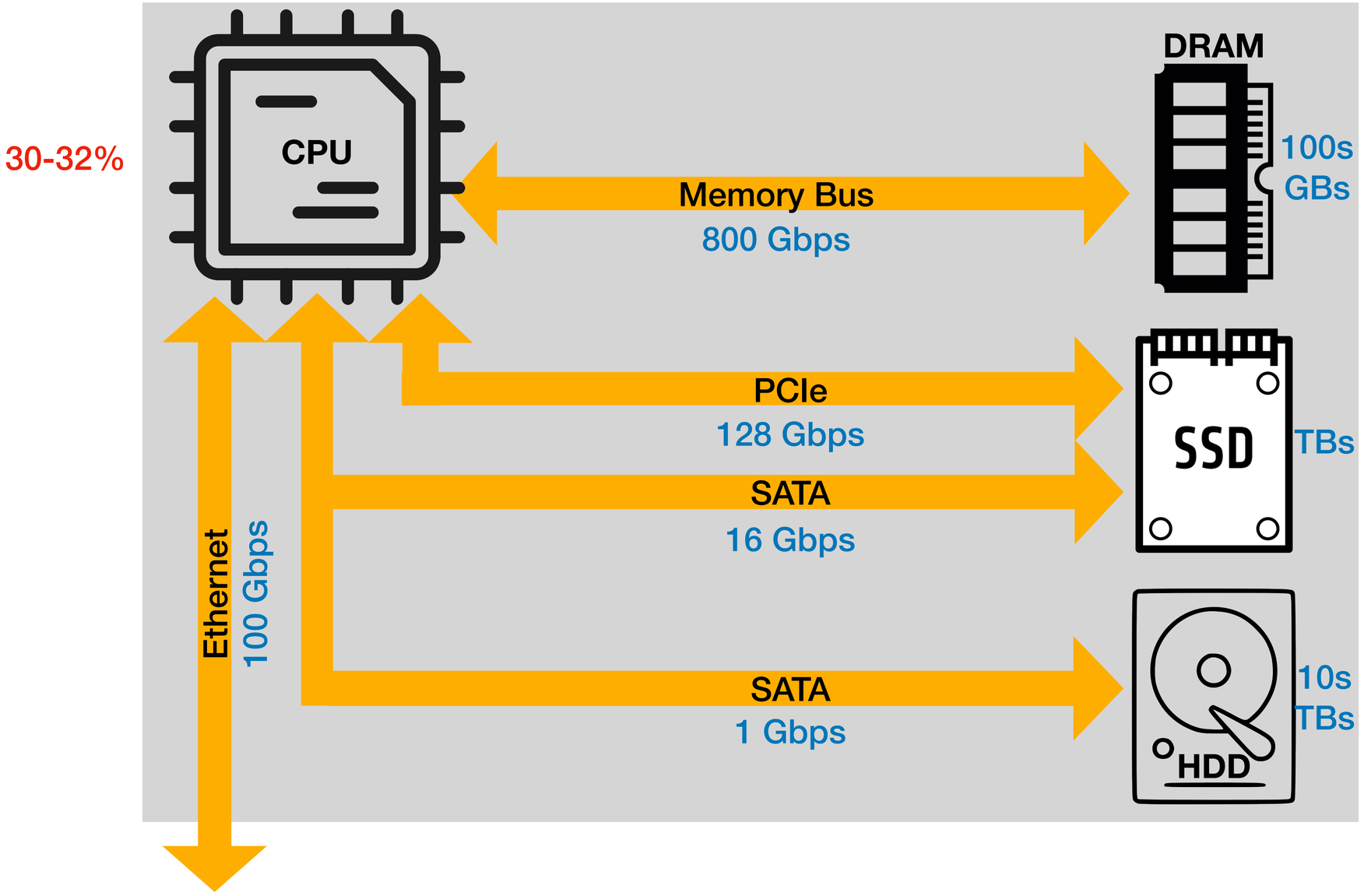
Conventional Wisdom: Doubles every 2 years

- **Dennard Scaling (1974)**
  - DRAM Inventor **Robert Dennard**
  - Power use ∝ transistor area => Smaller transistors use less power
  - Power use ∝ frequency => Smaller transistors using same power can be made faster (higher freq.)
  - At constant power, keep increasing frequency!
  - **Broke down ~2005-07**
  - **Physics strikes again:** Thermal loss…



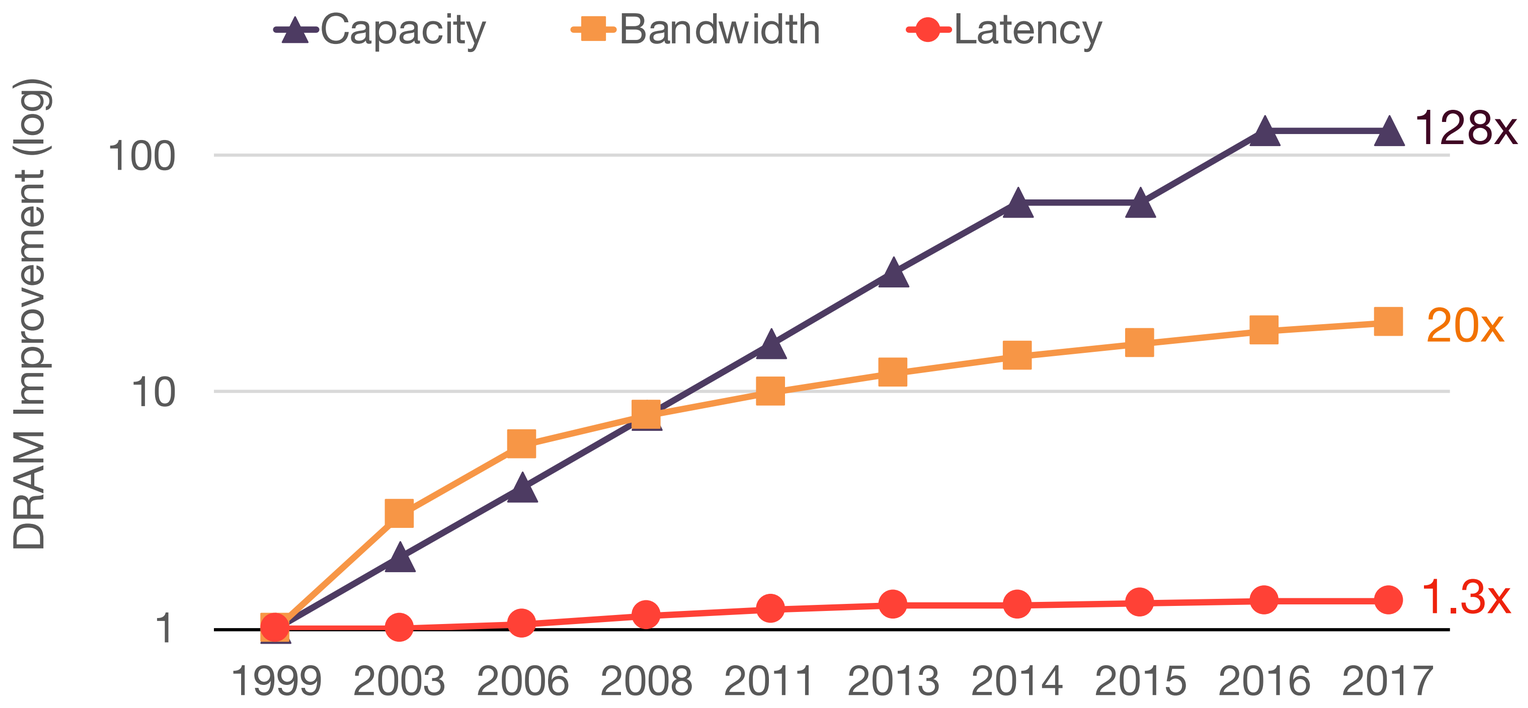
Microprocessor Trend Data, https://github.com/karlrupp/microprocessor-trend-data

Typical Server Node



DRAM

Capacity Growth: 30% per year

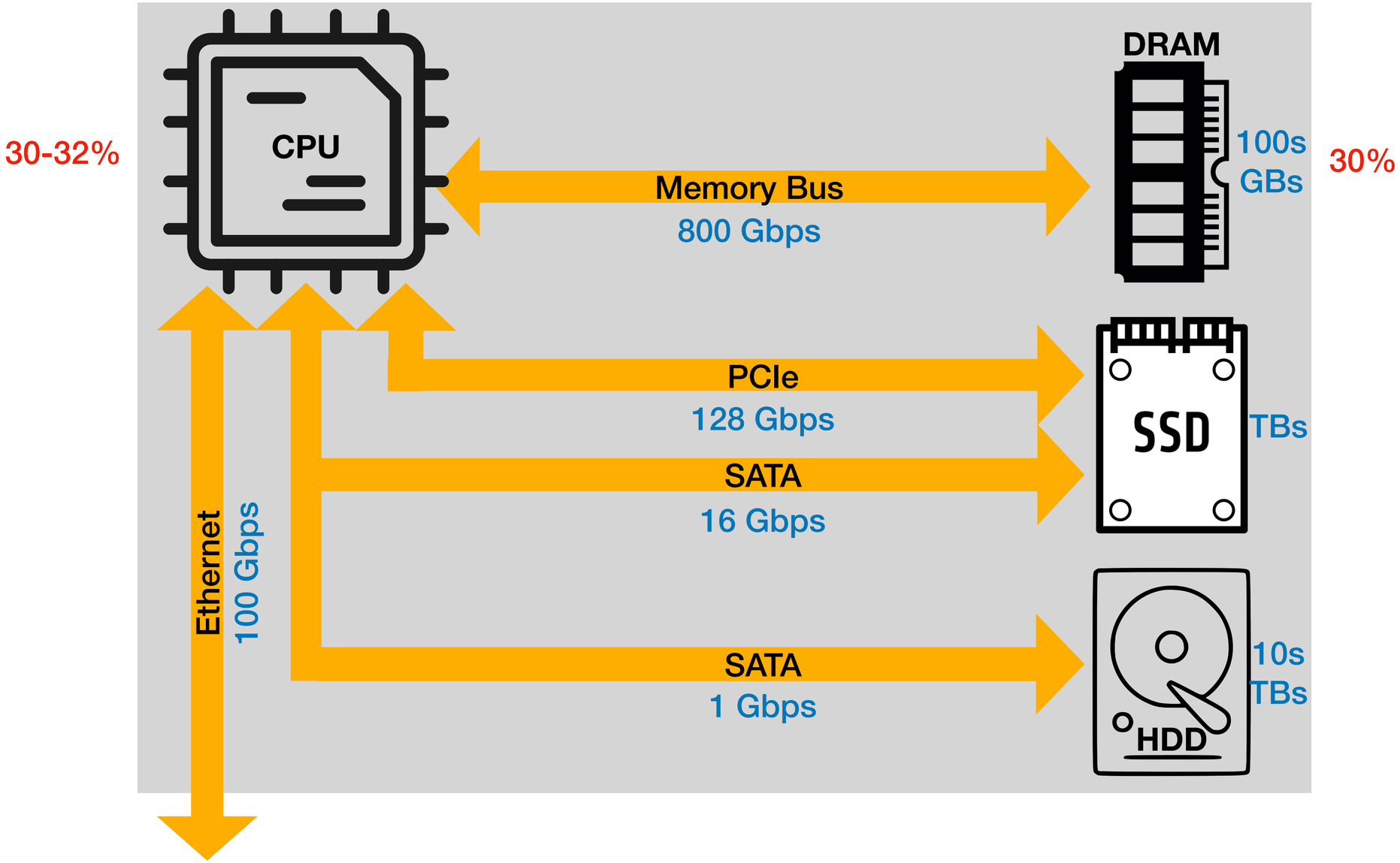


Data access from memory is getting more expensive!

Memory latency isn't improving!

- **What does this mean?**
- In 1999:
  - Memory capacity = M
  - Memory bandwidth = B
  - Time to scan full memory = T = M/B
- In 2017:
  - Memory capacity = 128M
  - Memory bandwidth = 20B
  - Time to scan full memory = 128M/20B = 6.4T

Typical Server Node



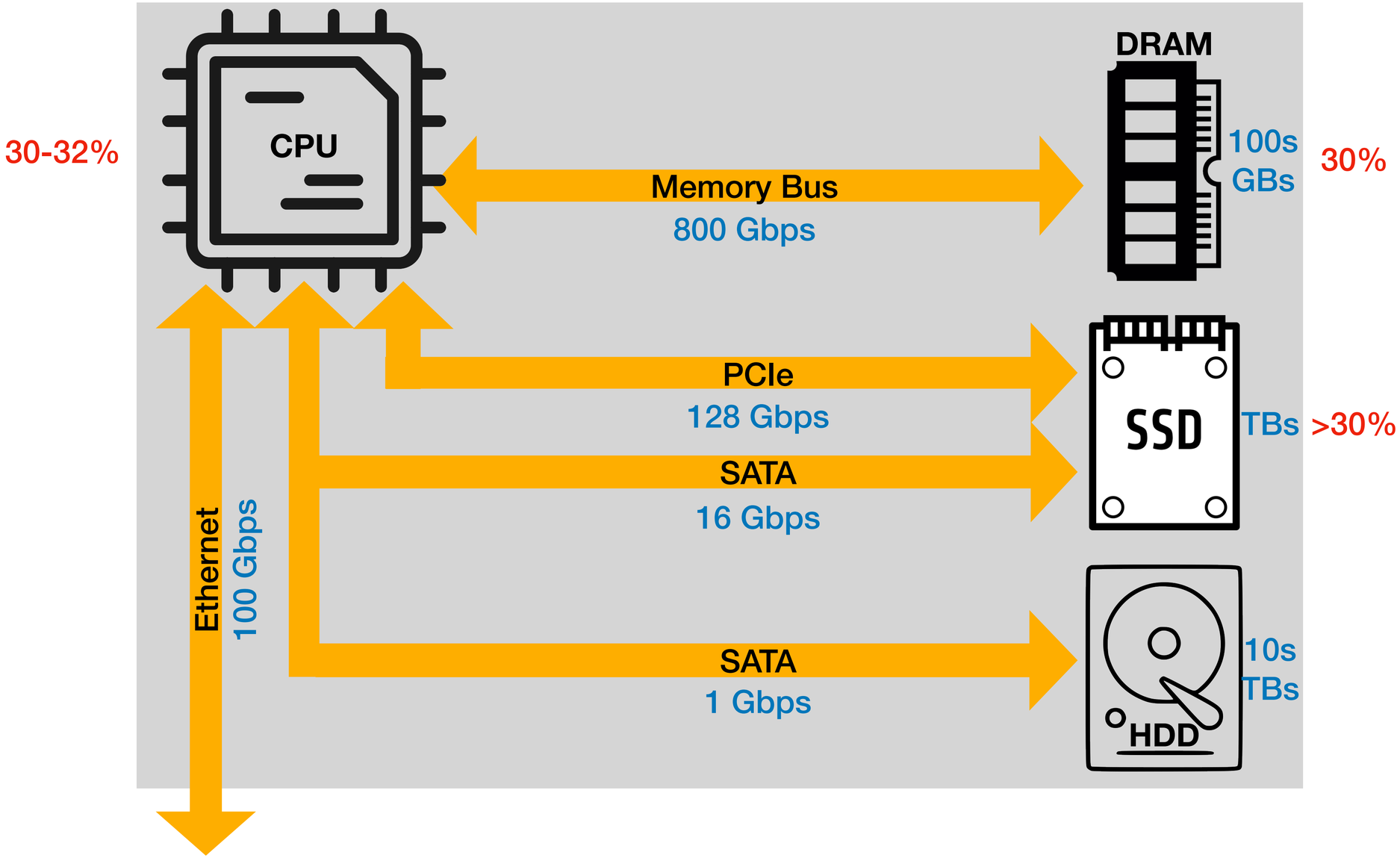
SSD Capacity

Growth: >30% per year

Leverages Moore's Law
i.e., follows CPU scaling trends

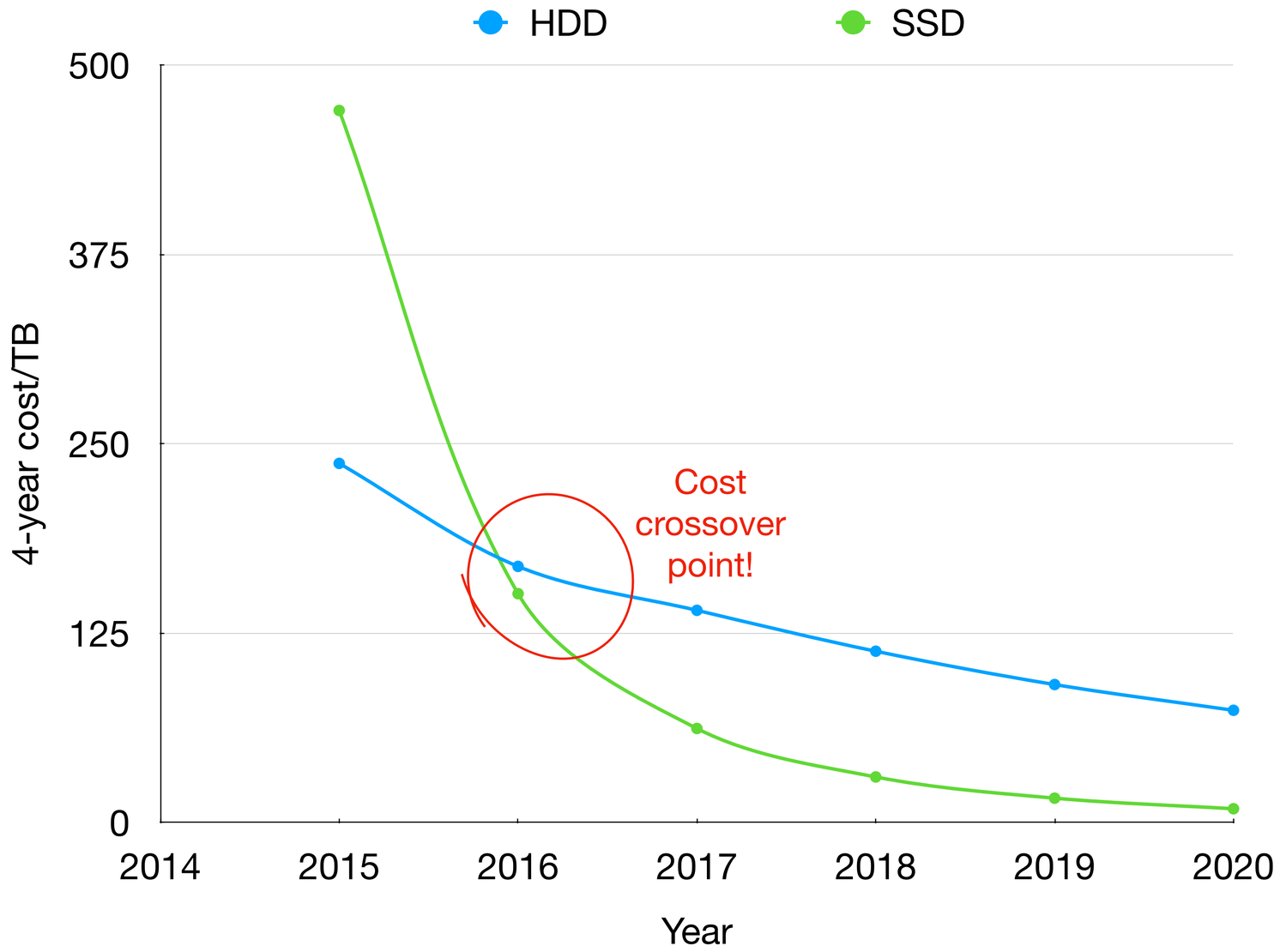
3D-Stacked flash technologies is helping outpace Moore's Law

Typical Server Node

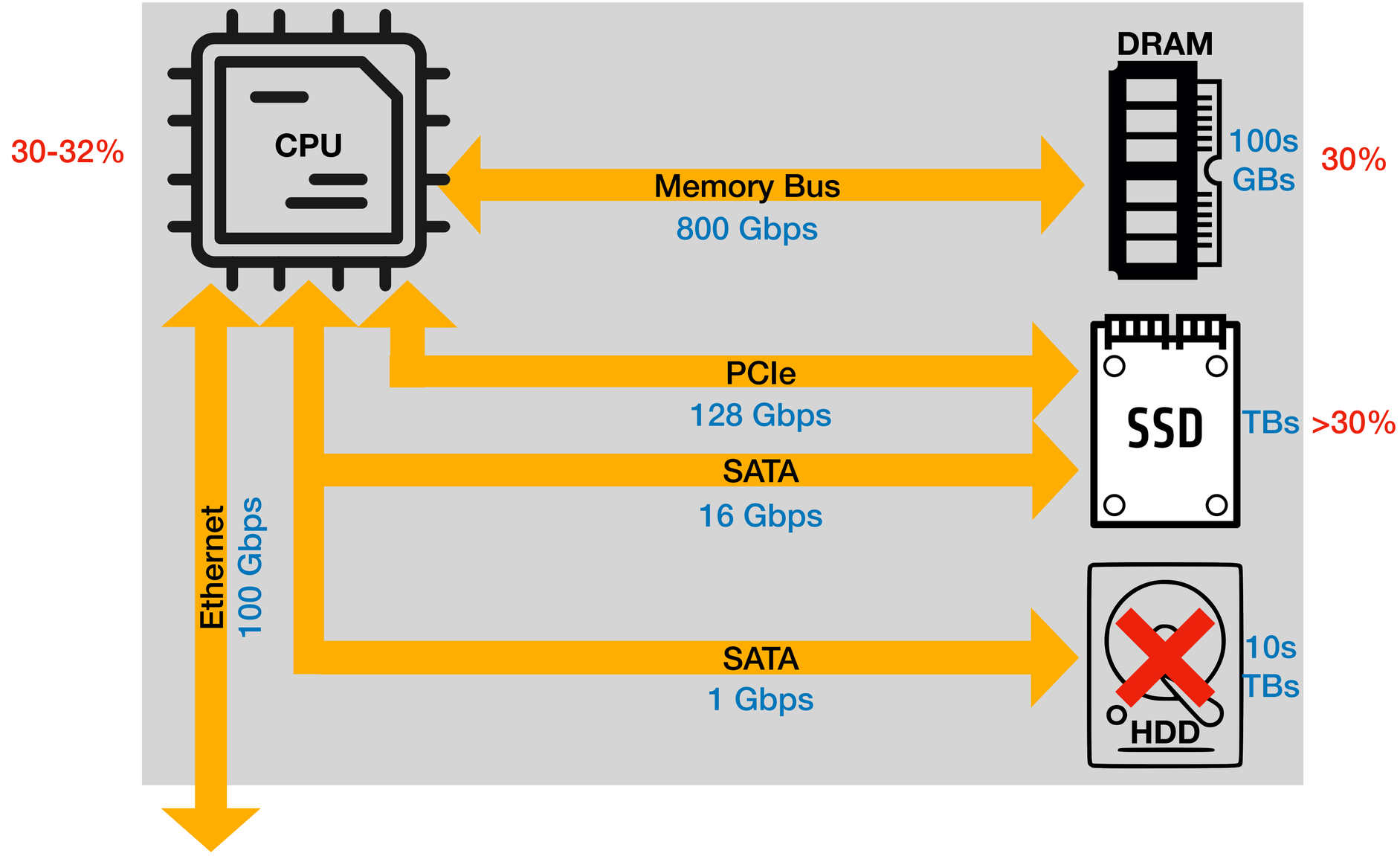


HDD vs SSD

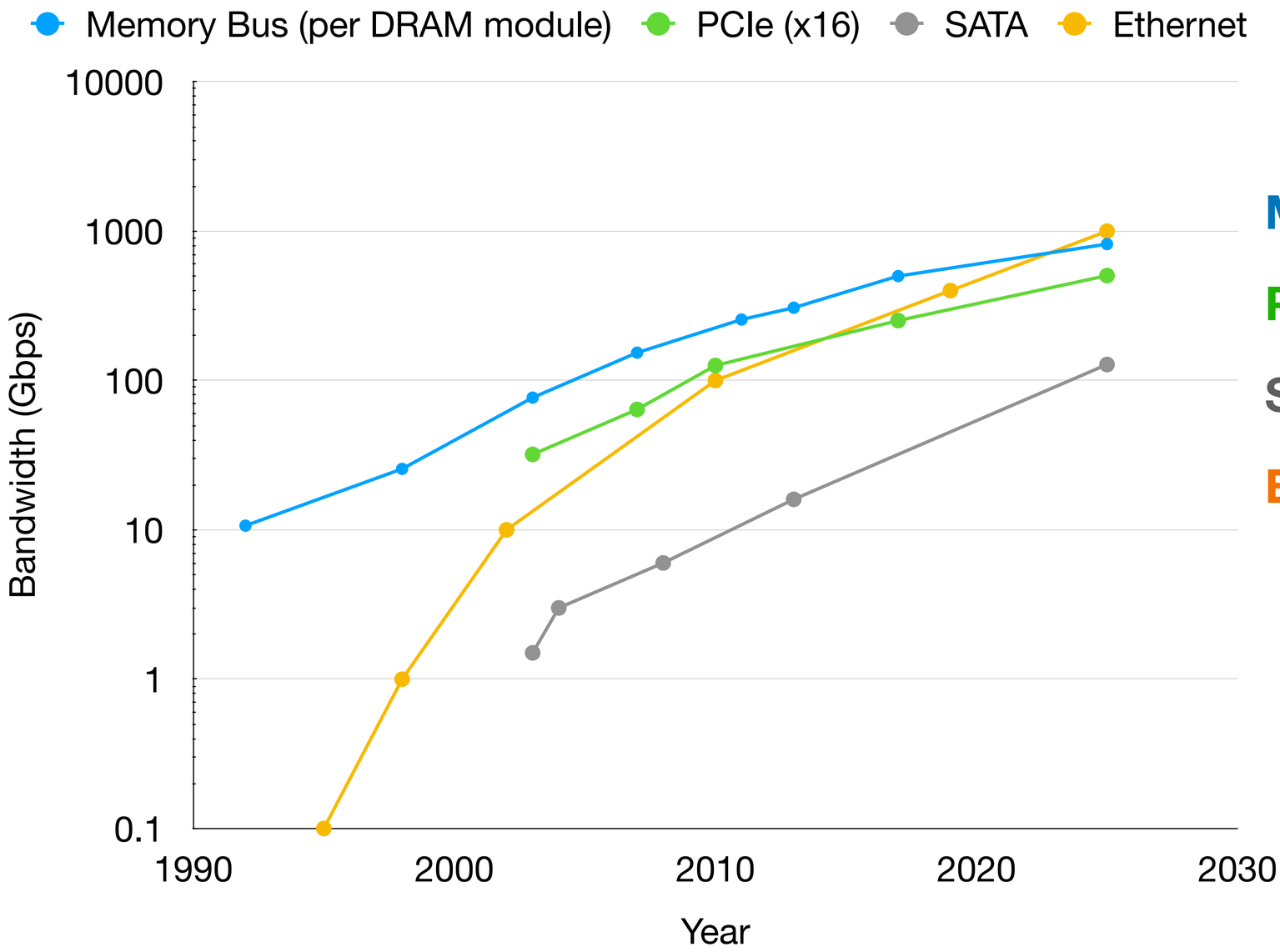
Stagnated growth: Archival only



Typical Server Node



Memory Bus, PCIe, SATA, Ethernet



Memory Bus Growth: 15% per year

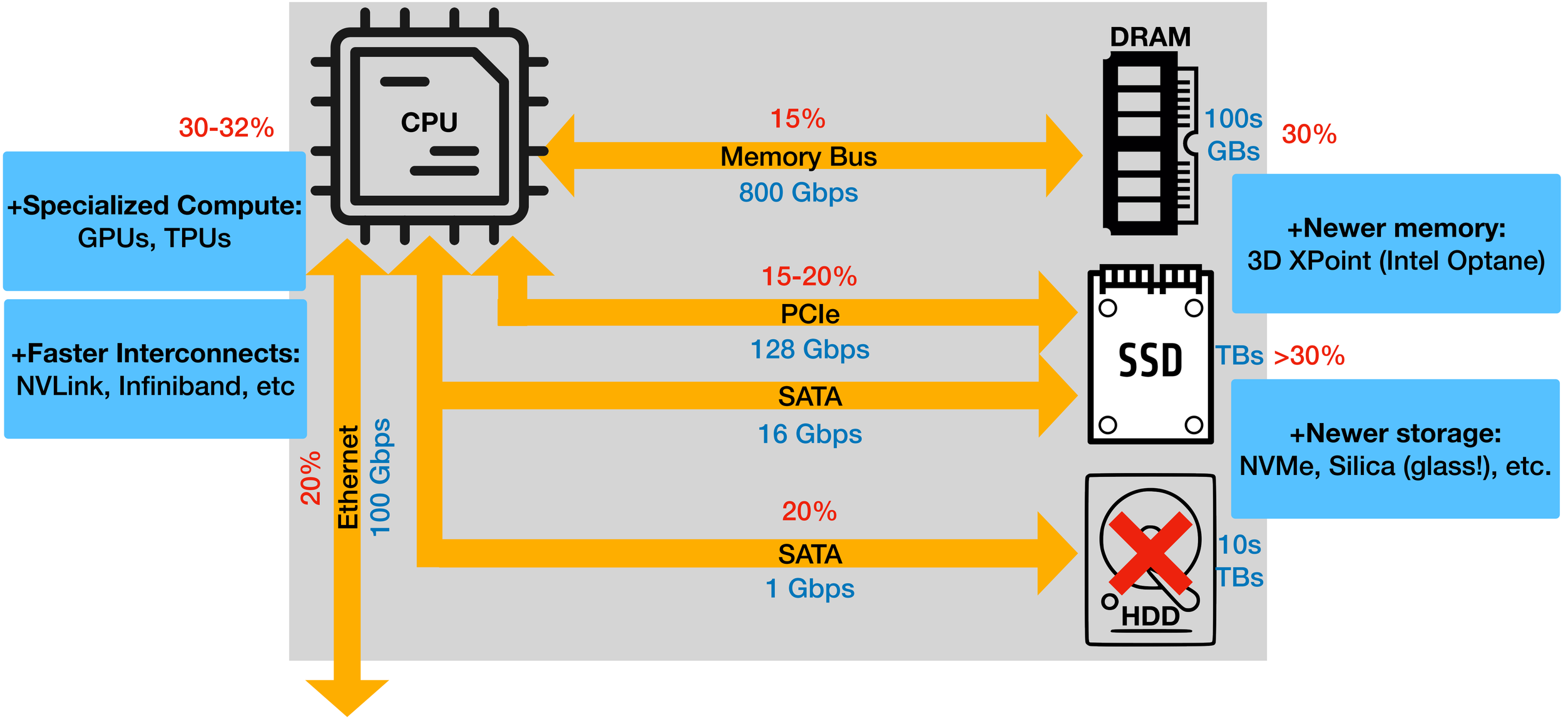
PCIe Growth: 15-20% per year

SATA Growth: 20% per year

Ethernet Growth: 33-40% per year

Network bandwidth growth outpacing the rest!

Typical Server Node



Trends Summary

- CPU per-core speed has stopped improving
- Memory bandwidth growth is slower than capacity
- SSD, NVMe devices replacing HDDs
- Ethernet bandwidth growth is outpacing all other interconnects

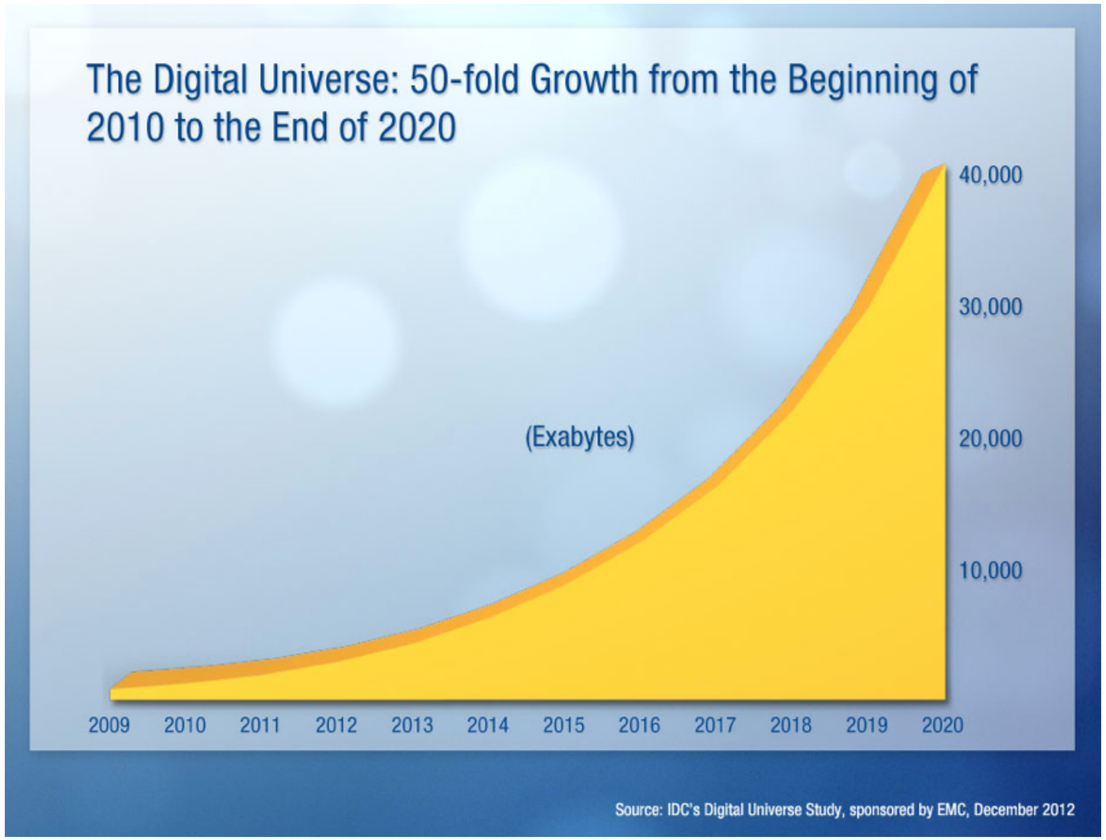
- **Question:** Scale **up** or scale **out**?
- **Question:** Other implications?

N/W stack processing will become a bottleneck

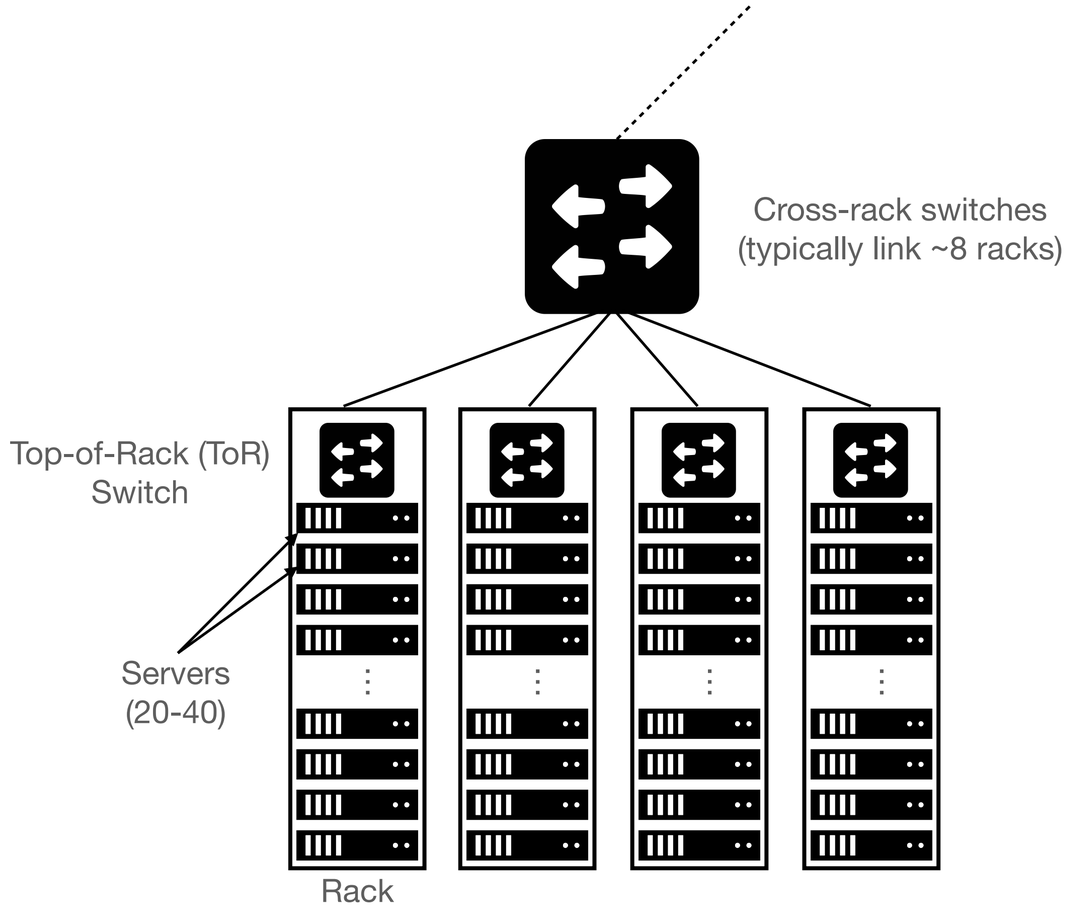
Remote memory will be faster than local SSD

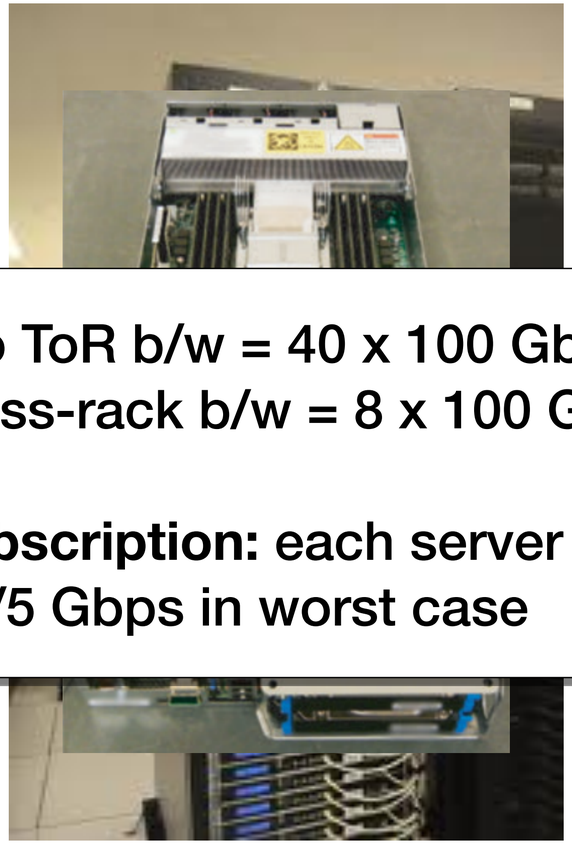
Why is One Server not Enough?

- Way too much data
- Not enough computing capacity
- Not enough storage capacity
- Not enough I/O bandwidth
- What if the server fails?



Datacenter Architecture

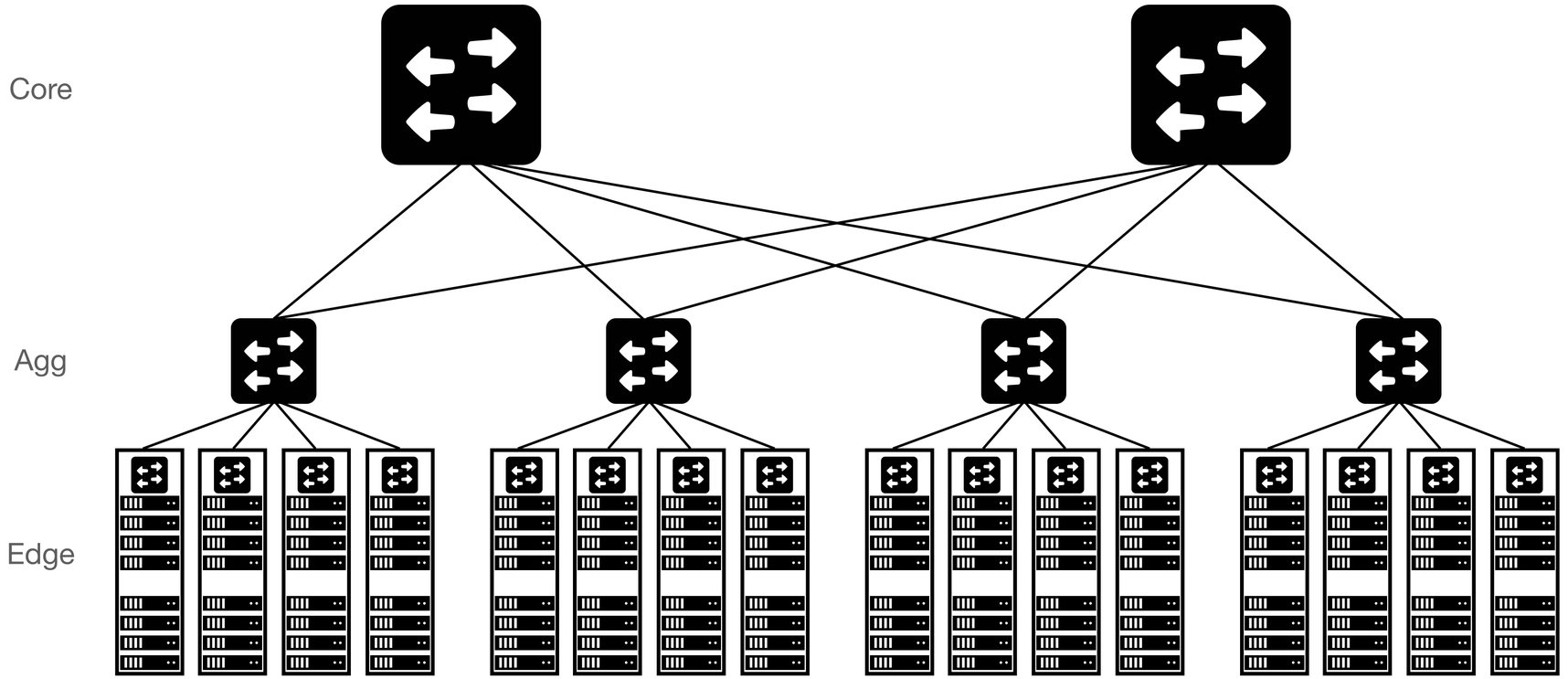




Server to ToR b/w = 40 x 100 Gbps
ToR to cross-rack b/w = 8 x 100 Gbps

5x oversubscription: each server gets 100/5 Gbps in worst case

Datacenter Architecture



- ToR switches connected by **‘Aggregation switches’**
  - 2x redundancy for fault-tolerance
- DCN connected to the internet via **‘Core switches’**
  - **Note:** blurry boundary b/w core and aggregation
- Other (better) topologies:
  - Fat-tree, Clos, Jellyfish, …

Datacenter Architecture for the Future

- How would you redesign datacenter architecture given:
  - Intra-server interconnect bandwidths are stagnating
  - Inter-server interconnect bandwidth continues to grow at rapid pace
  - What are the main challenges?

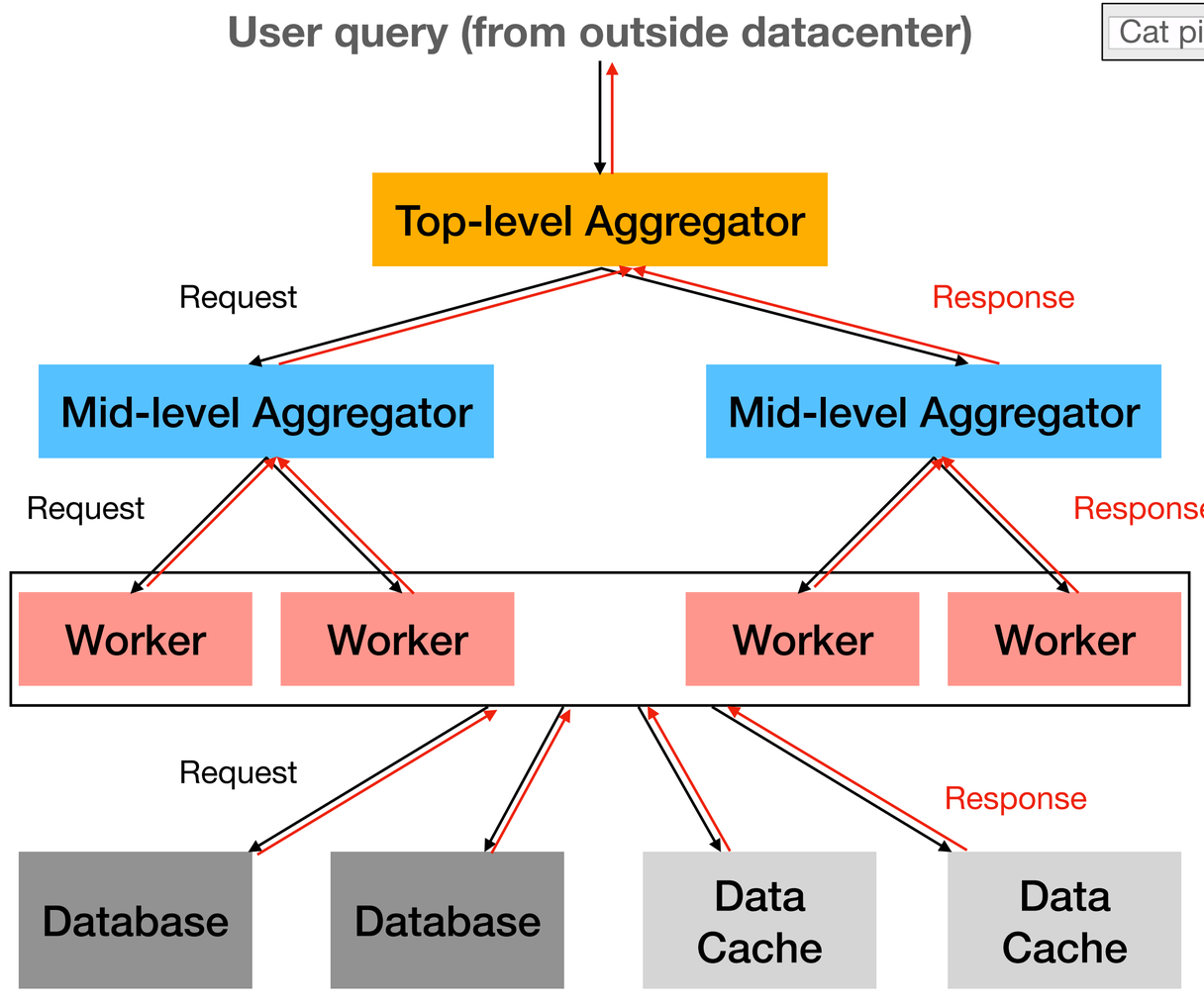
Big Data Application Characteristics

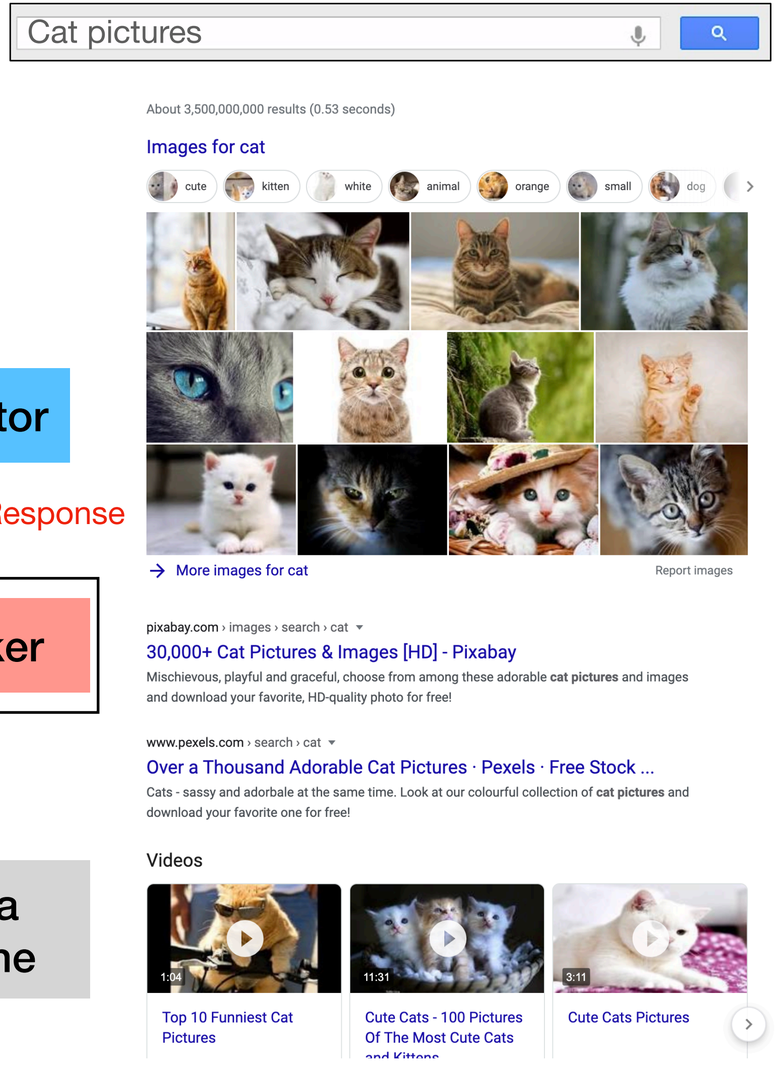
Application Characteristics

- Common Theme: **Parallelism**
  - Application decomposed into **tasks**
  - Running in **parallel** on different servers
- Two common paradigms (not exhaustive)
  - Partition-Aggregate (*Interactive*)
  - Map-Reduce (*Batch Processing*)

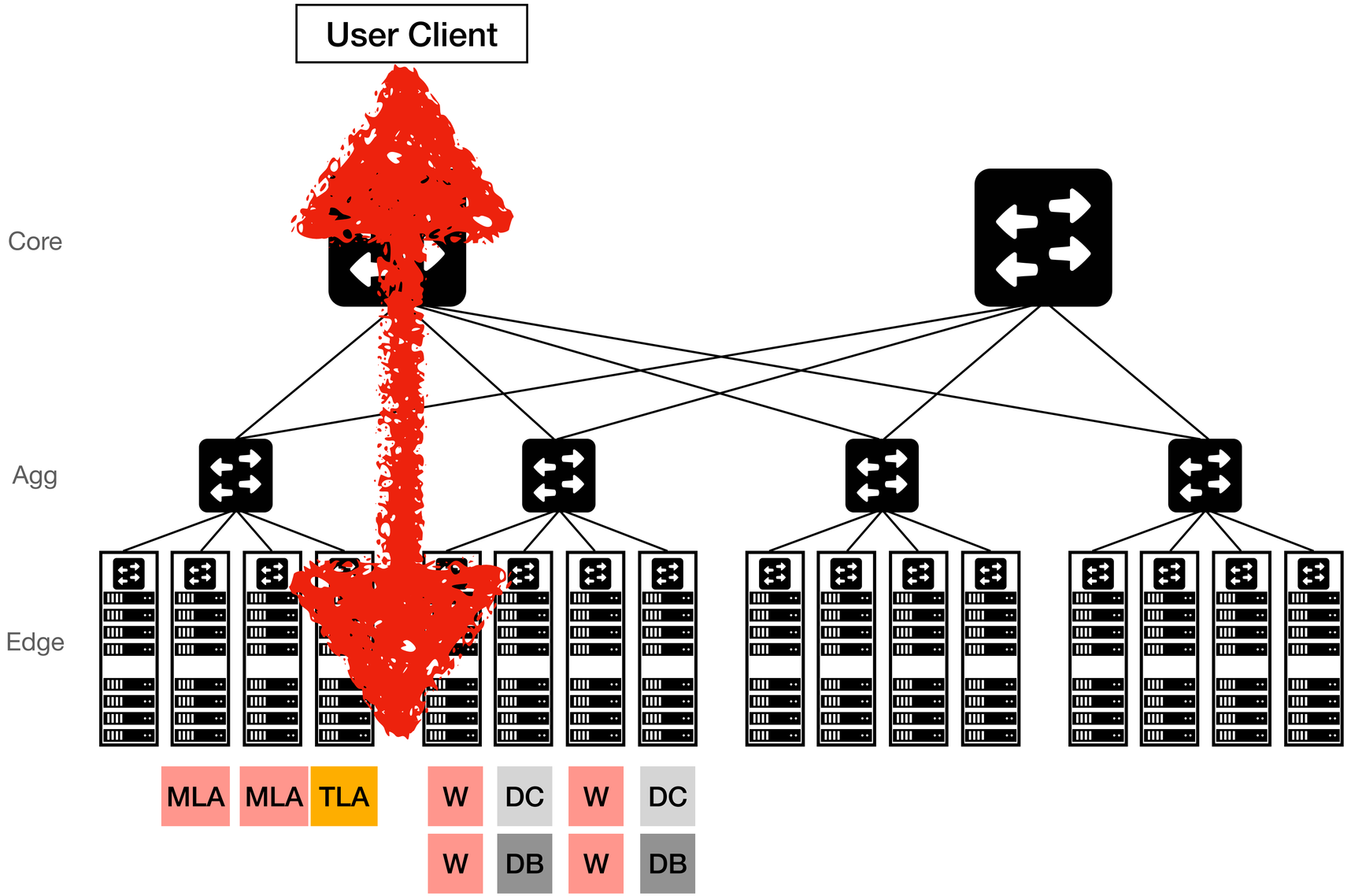
Workload: Partition Aggregate (Interactive)

Most communications over network



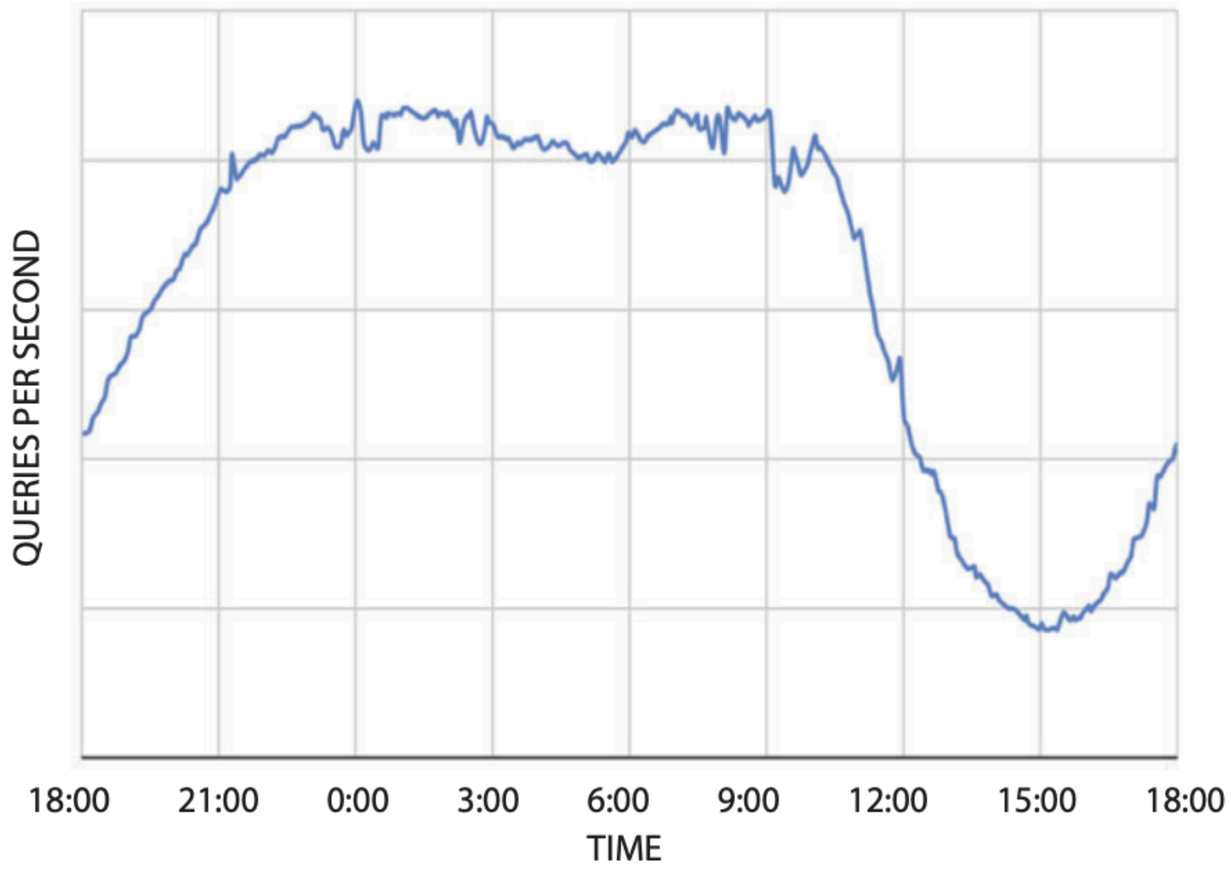


“North-South” Traffic

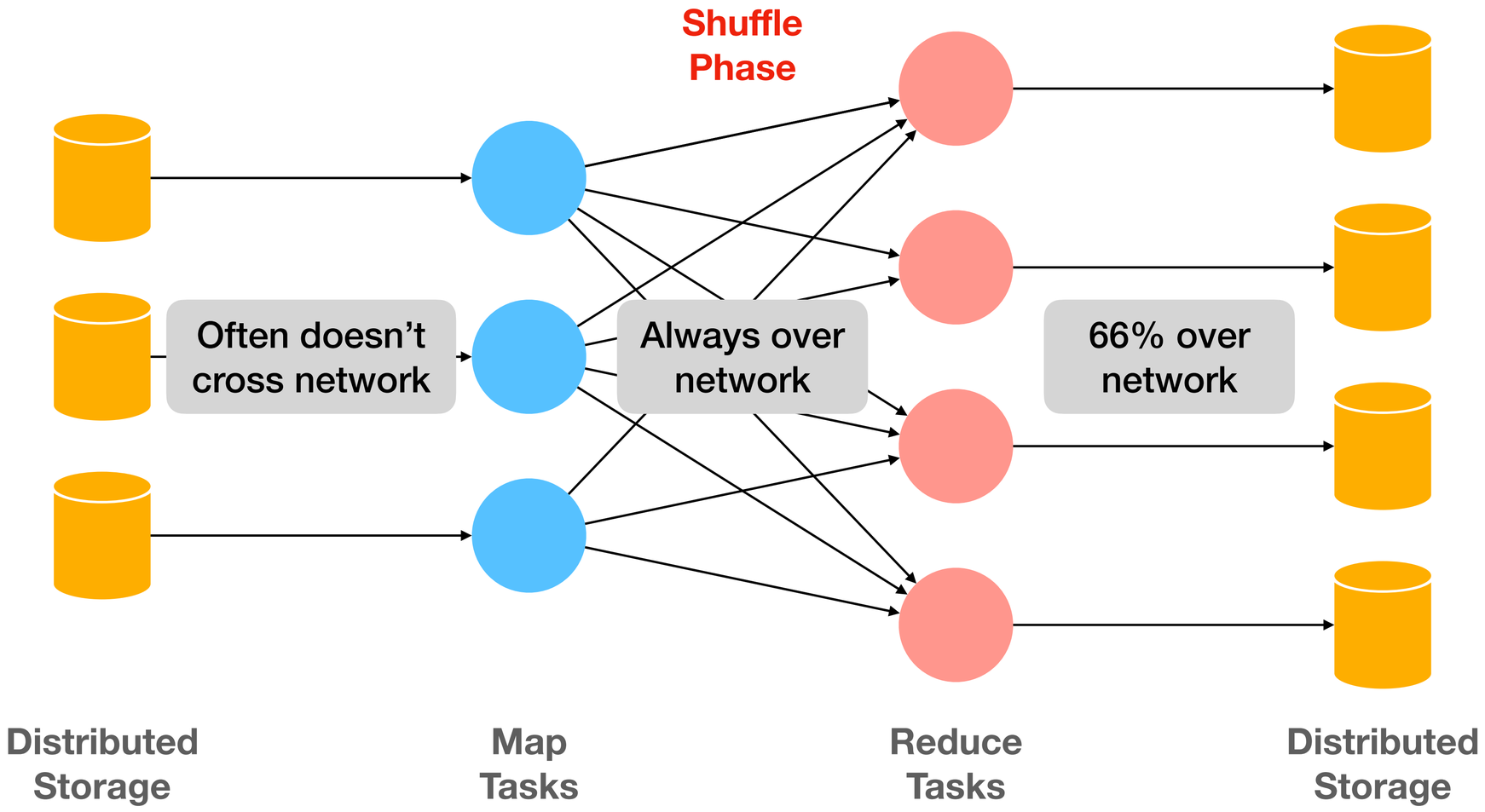


- Interactive query-response exchange b/w external clients & datacenter
  - ***Latency sensitive***
  - ***O(milliseconds)***
- Handled by worker/ aggregator tasks, databases & caches

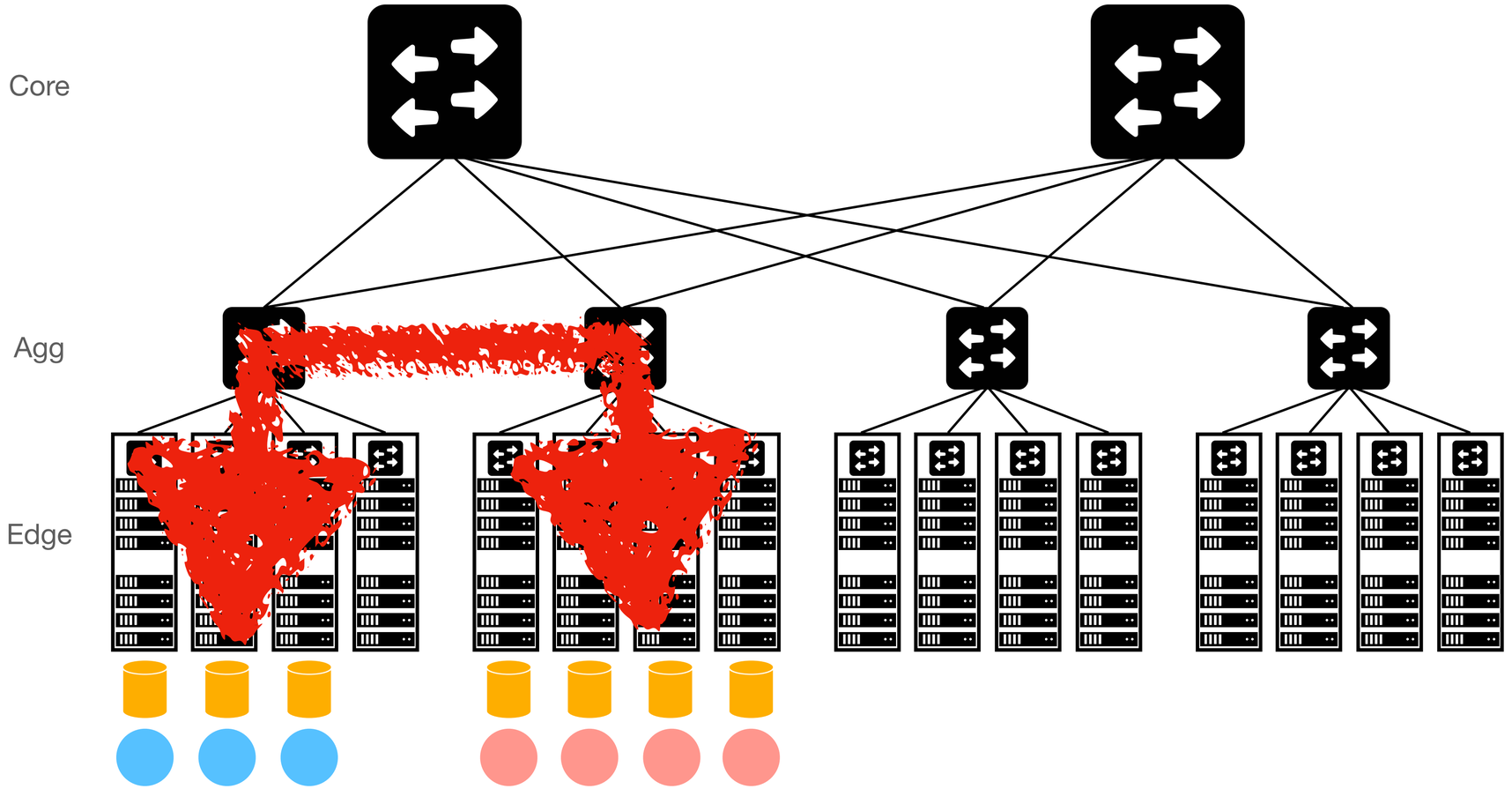
Diurnal Workload Patterns



Workload: Map Reduce (Batch Processing)



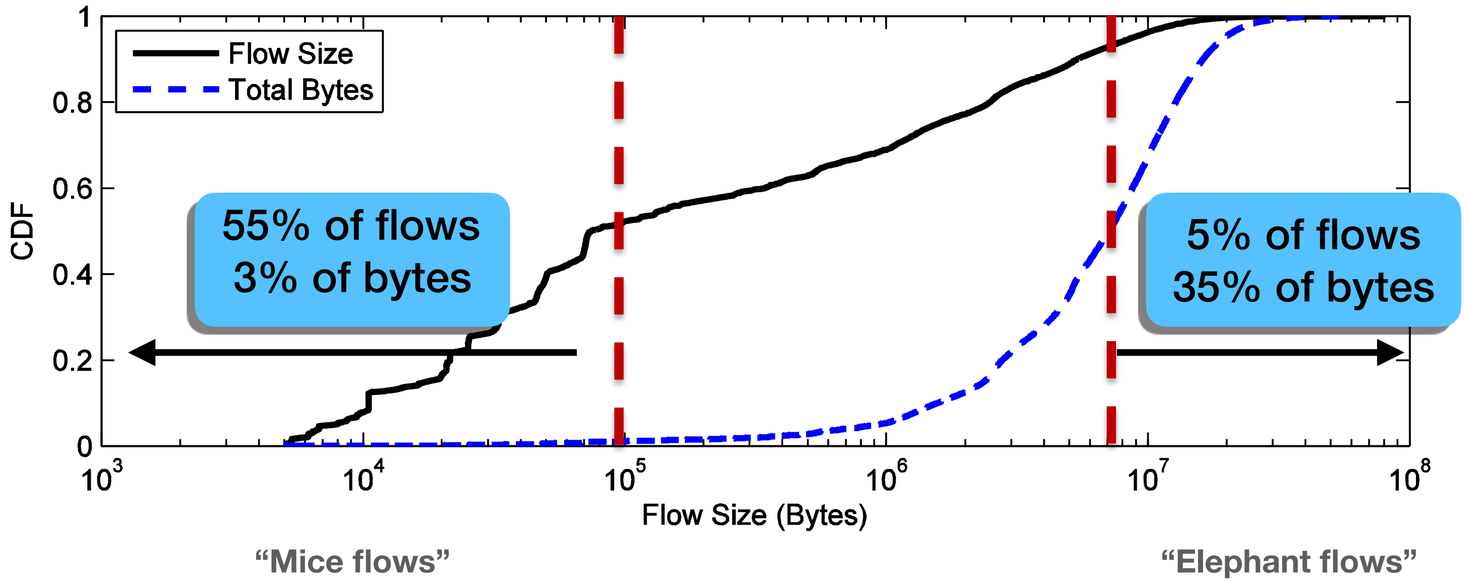
“East-west” Traffic



- Traffic between servers in the datacenter
  - ***Bandwidth intensive***
  - ***O(mins)***
- Handled by map/reduce tasks, distributed filesystem

Characterizing Traffic Pattern: “Elephant” & “Mice” Flows

- **Microsoft [Alizadeh et. al. 2010]**
  - Web-search (north-south), data mining (east-west)



Research: How do you design the network protocols for such traffic?

Big Data Systems

- How are these systems going to be different for:
  - Different Application Characteristics (e.g., Part.-Agg. vs Map-Red.)?
  - Emerging hardware trends?