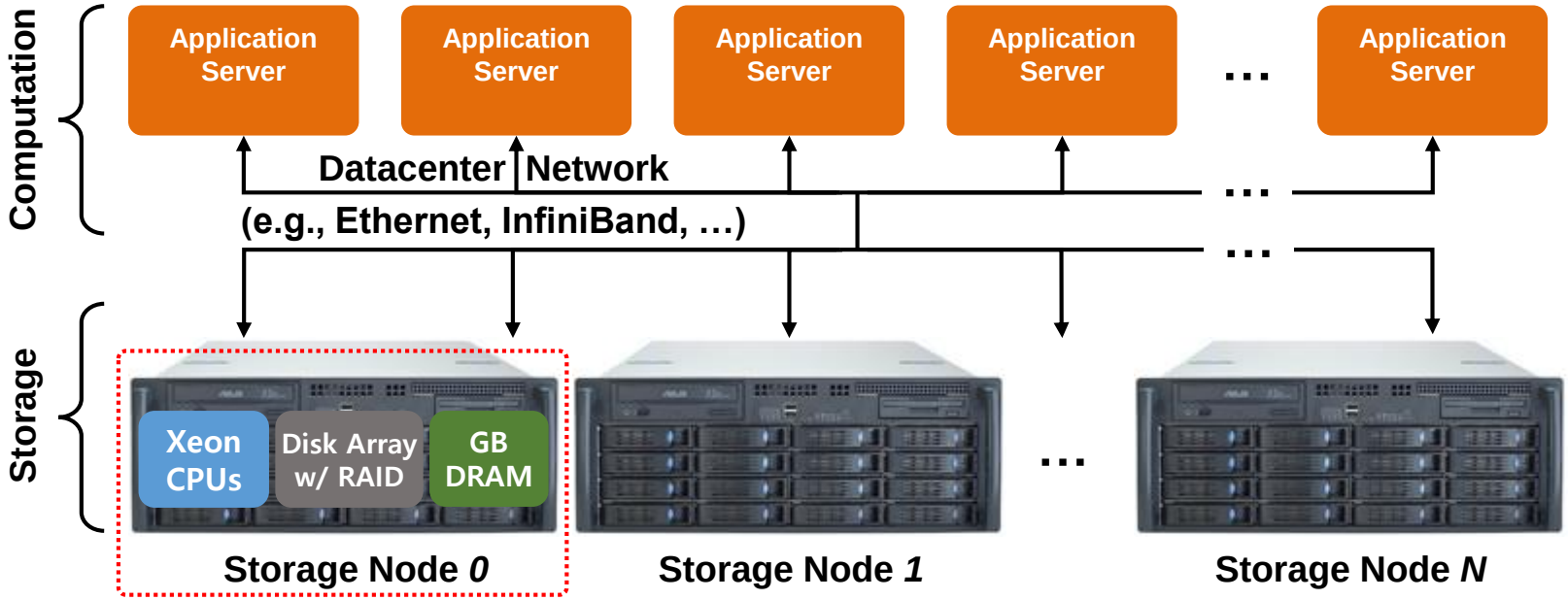
Lightweight KV-based Distributed Store for Datacenters

***Chanwoo Chung*, Jinhyung Koo*, Arvind, and Sungjin Lee†**

Massachusetts Institute of Technology (MIT)
† Daegu Gyeongbuk Institute of Science & Technology (DGIST)
* Inha University

9th USENIX Workshop on Hot Topics in Storage and File Systems
Wild and Crazy Ideas (WACI)
July 10-11, 2017

Existing Distributed Storage Systems



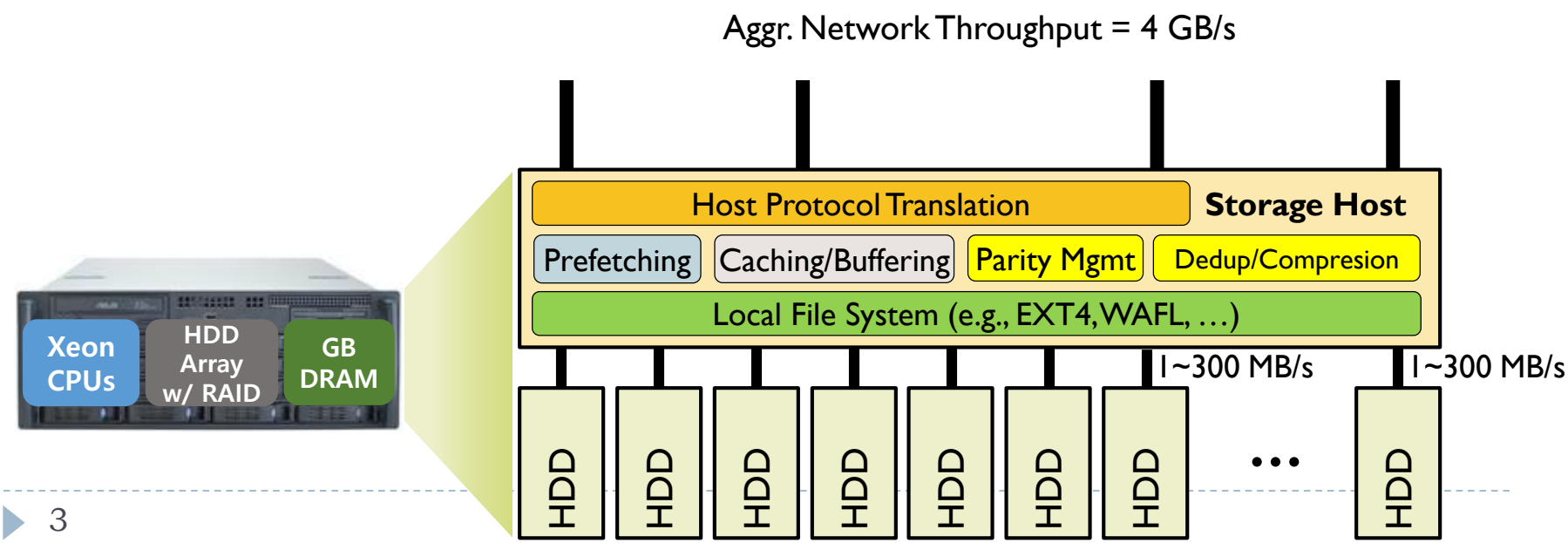
It is not a mere storage – it is **another high-end server**!!!

High-end Xeon CPUs
Several GBs of DRAM
An array of SSDs
Large form-factor
…

Power Hungry (e.g., 800 W)
Expensive (e.g., $42,000 w/o SSDs)
Large Volume (e.g., 2-4 U)
High TCO (e.g., Cooling)
…

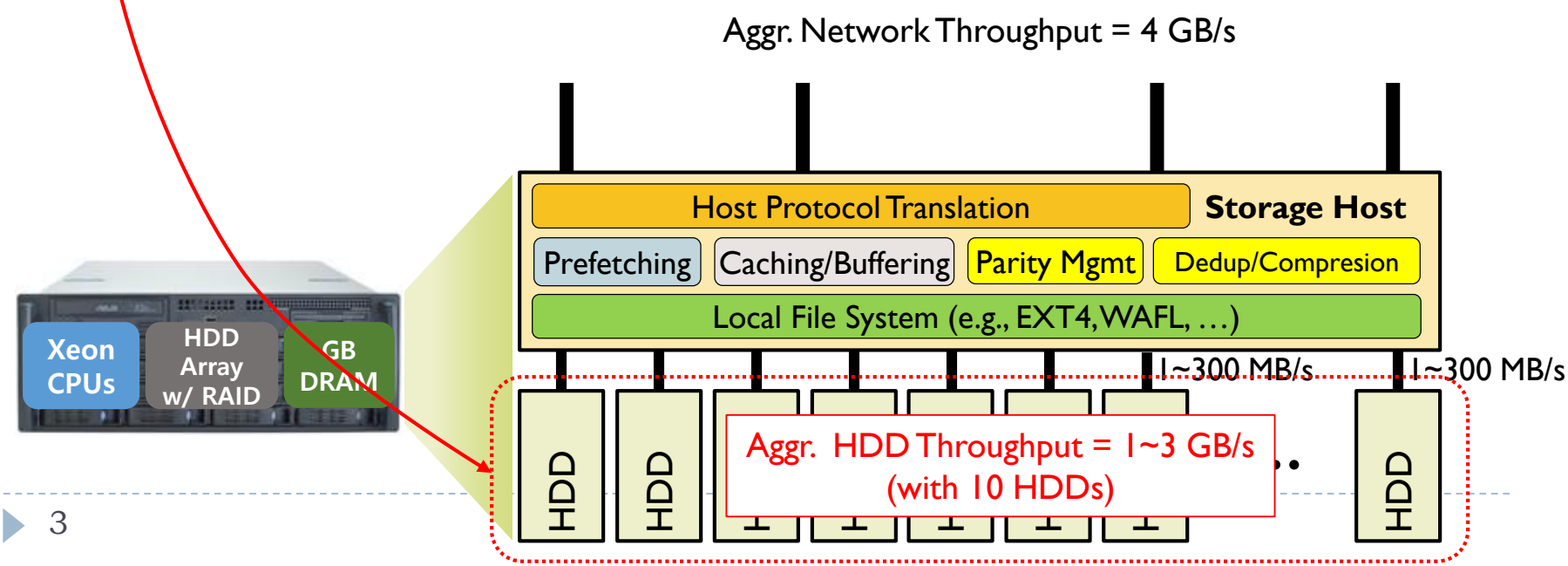
Storage Node is Designed for HDDs

- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput
- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms



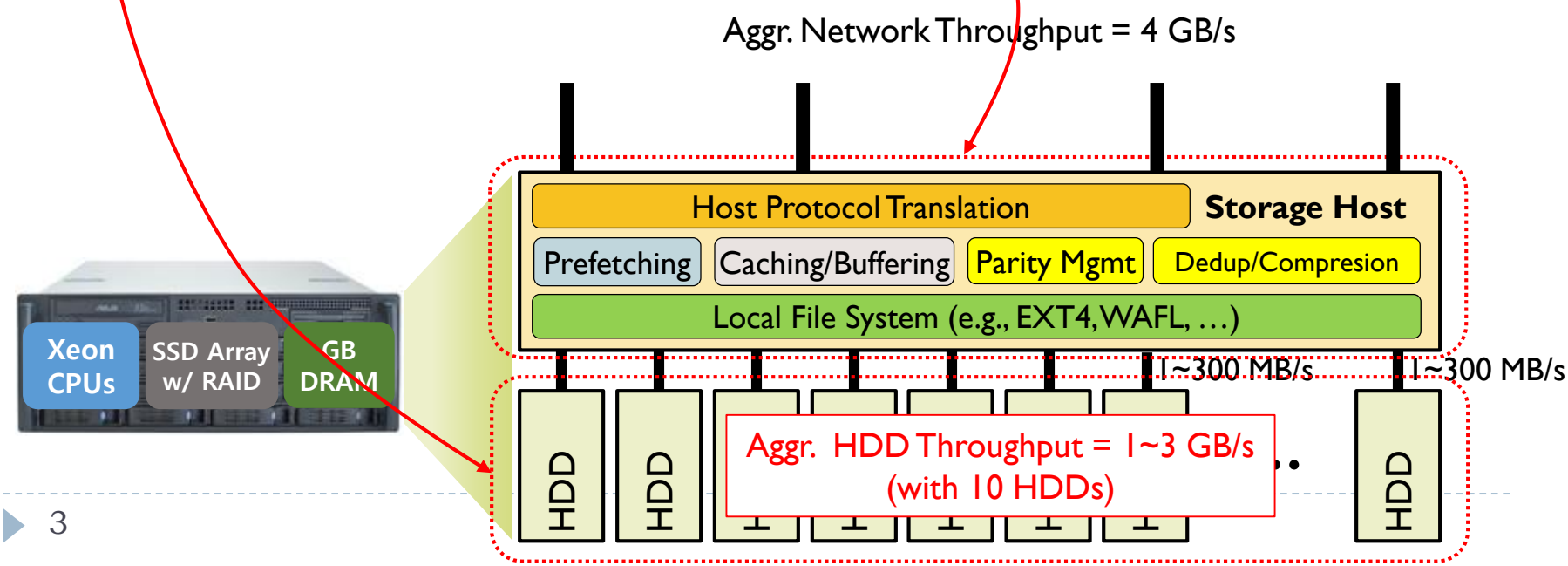
Storage Node is Designed for HDDs

- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput
- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms



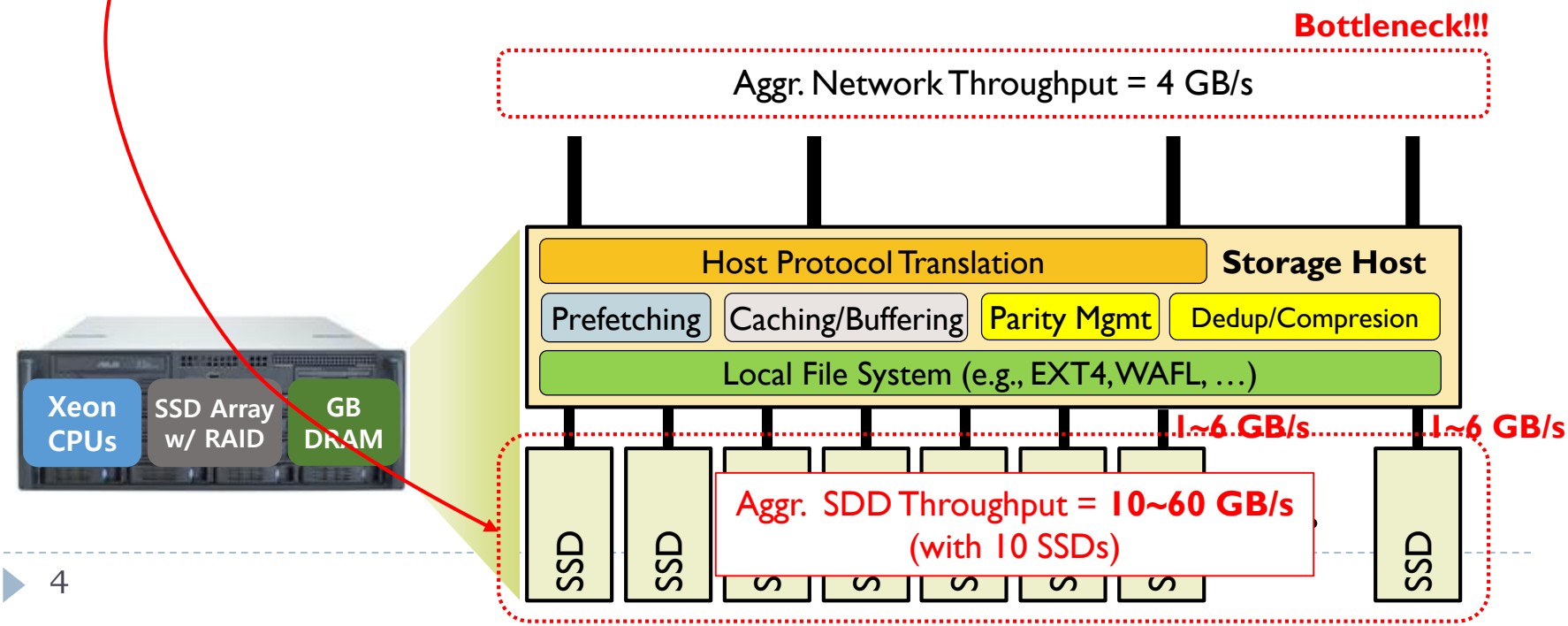
Storage Node is Designed for HDDs

- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput
- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms



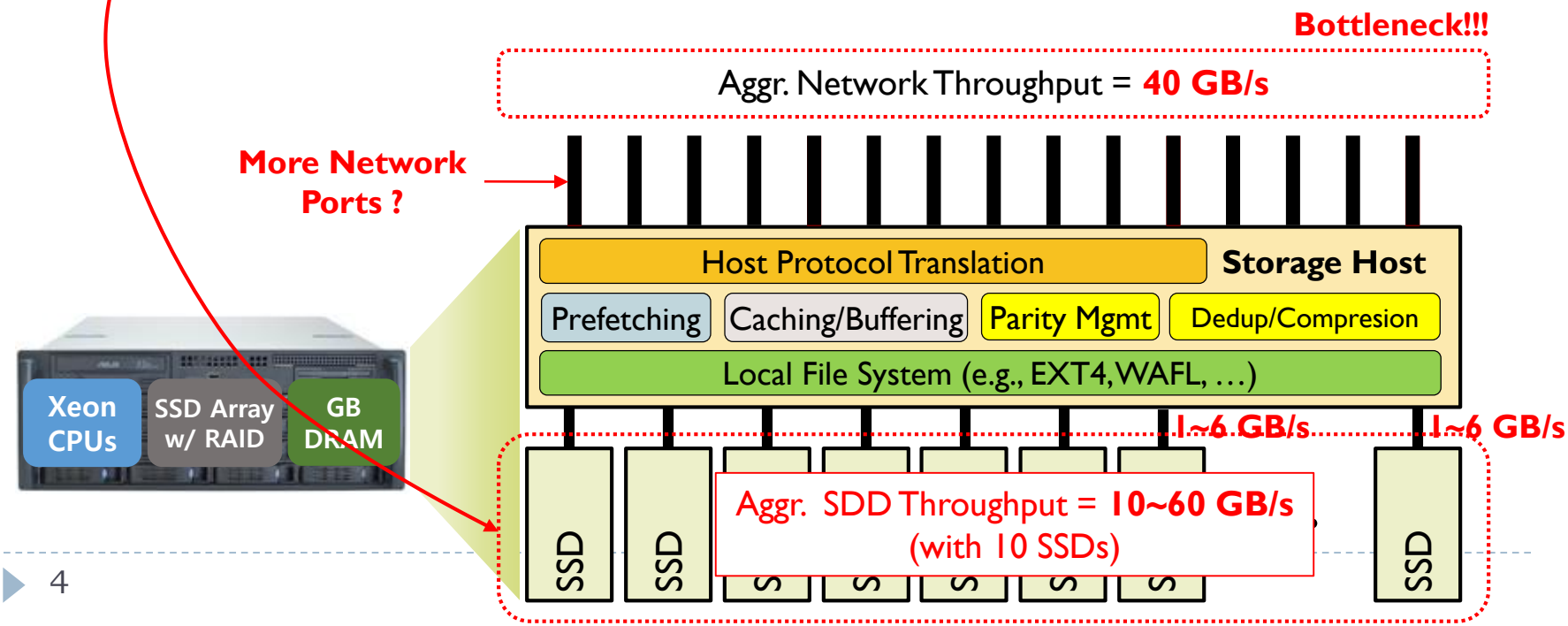
Unfortunately,
Such an Architecture is Invalid with SSDs

- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput
  - ➡ SSDs are not a bottleneck → Network/CPU are new bottlenecks
- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms



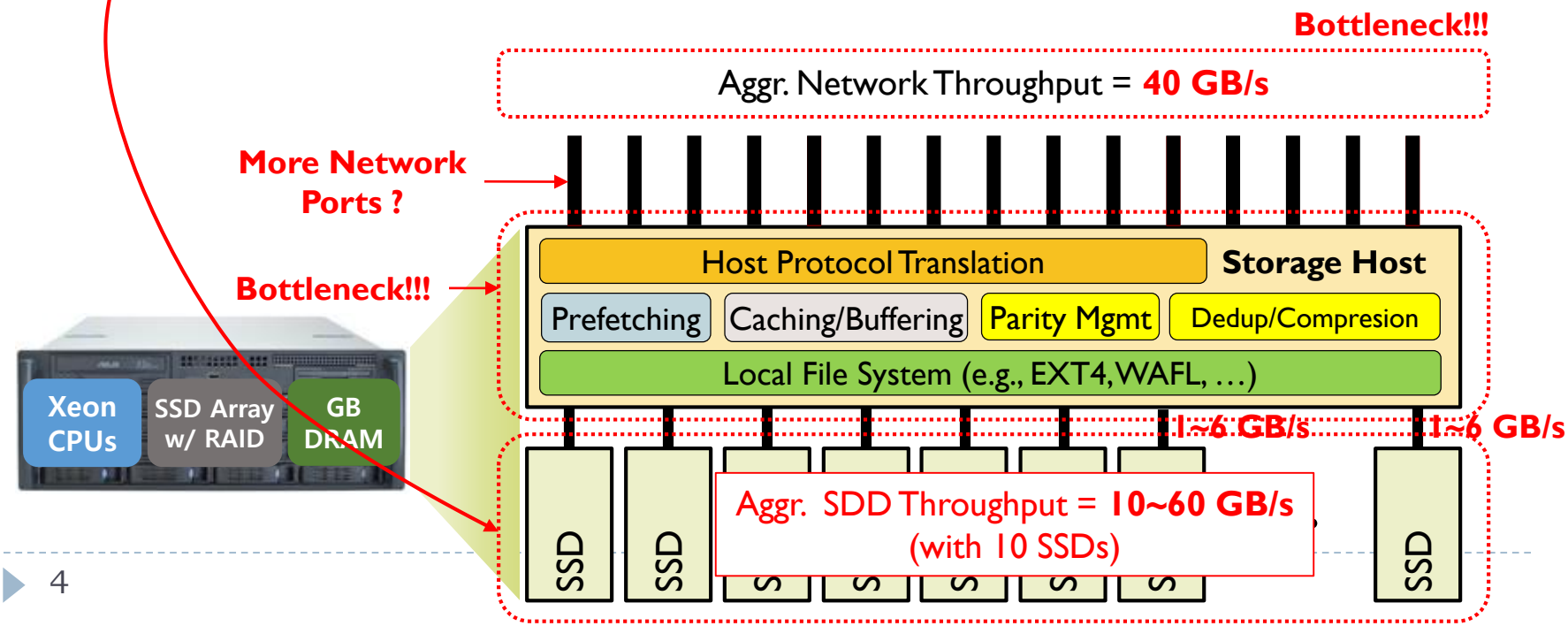
Unfortunately,
Such an Architecture is Invalid with SSDs

- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput
  - ➡ SSDs are not a bottleneck → Network/CPU are new bottlenecks
- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms



Unfortunately,
Such an Architecture is Invalid with SSDs

- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput
  - ➡ SSDs are not a bottleneck → Network/CPU are new bottlenecks
- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms



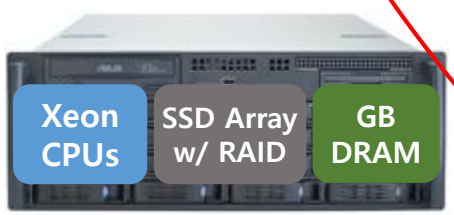
Unfortunately,
Such an Architecture is Invalid with SSDs

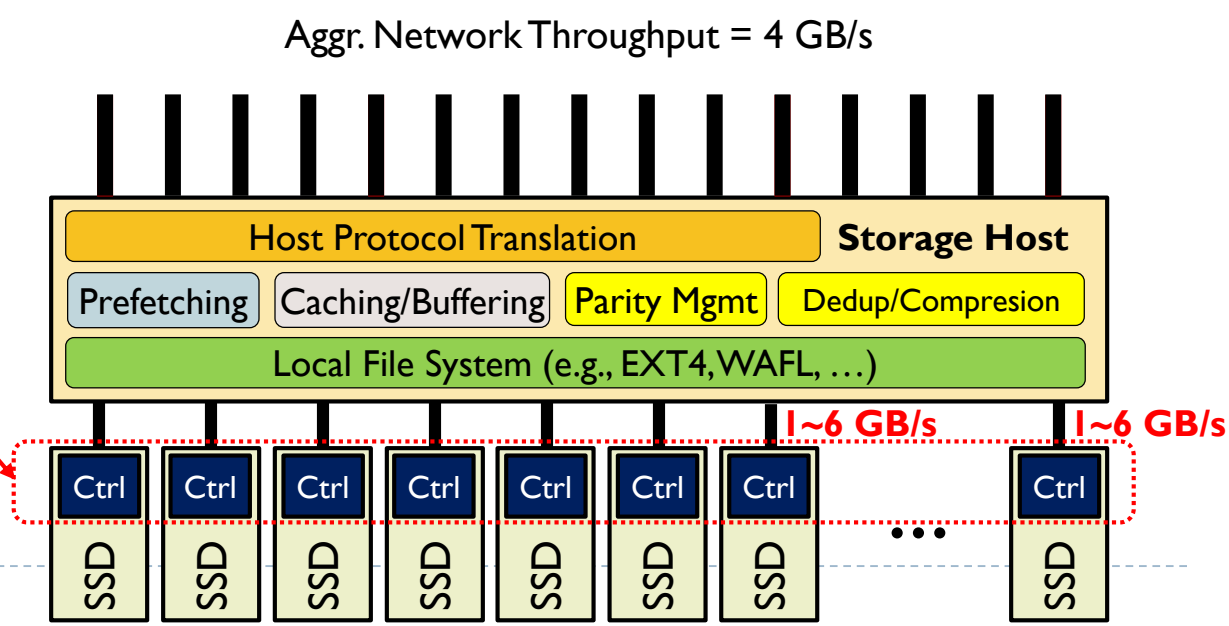
- HDD is ***slow*** – require large DRAM and array of disks
  - 10 *ms* latency & 100~300 MB/s throughput

➡ SSDs are not a bottleneck → Network/CPU are new bottlenecks

- HDD is ***dumb*** – the host system makes it smarter
  - Xeon CPUs with advanced algorithms

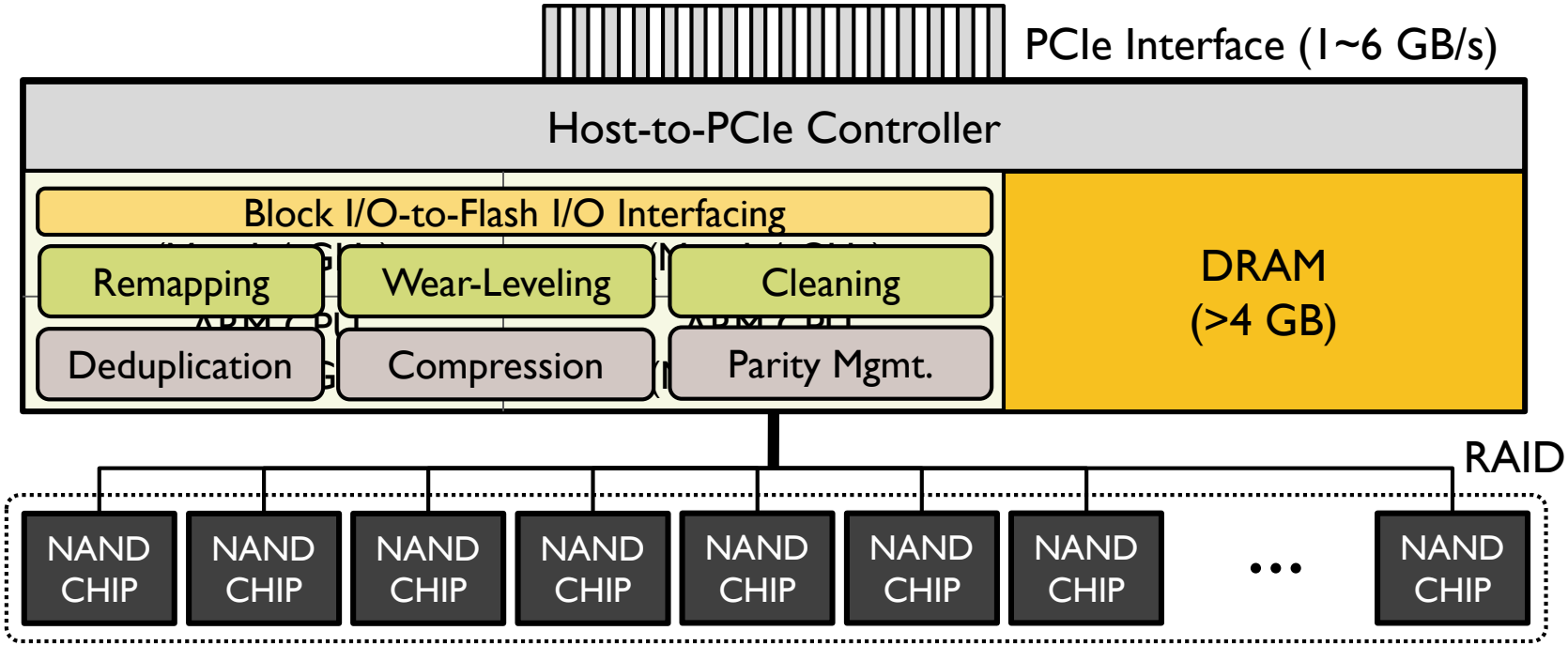
➡ SSDs are smarter than you think





Let's Look into SSDs

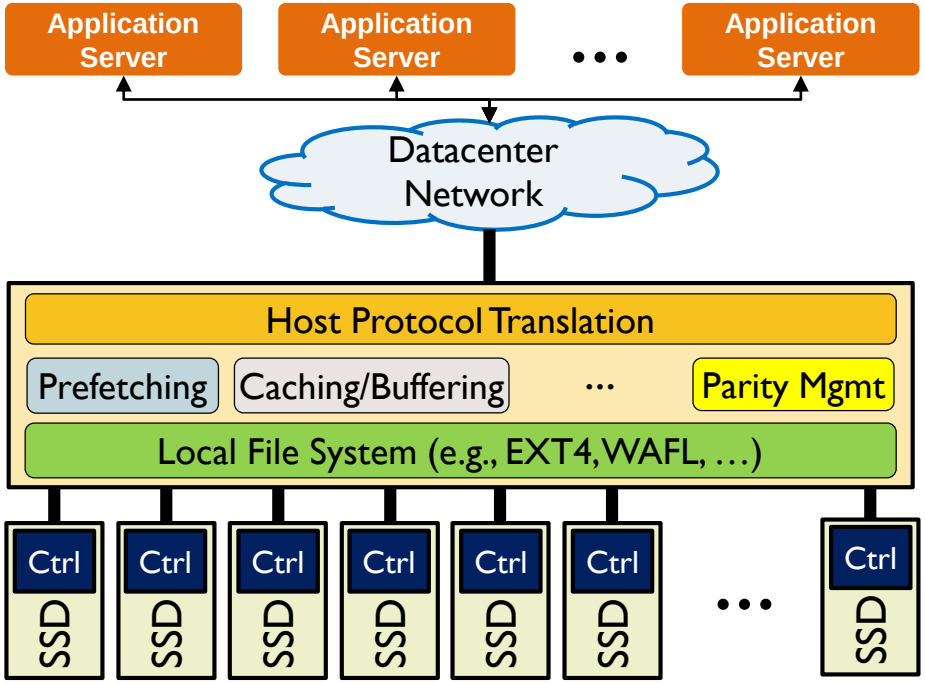
- 4 embedded CPUs running at 700 MHz to 1.4 GHz and >2~4GB DRAM that a desktop system had 10 years ago
- Those resources are required for firmware/FTL algorithms



- But, many of the algorithms are redundant in storage host server

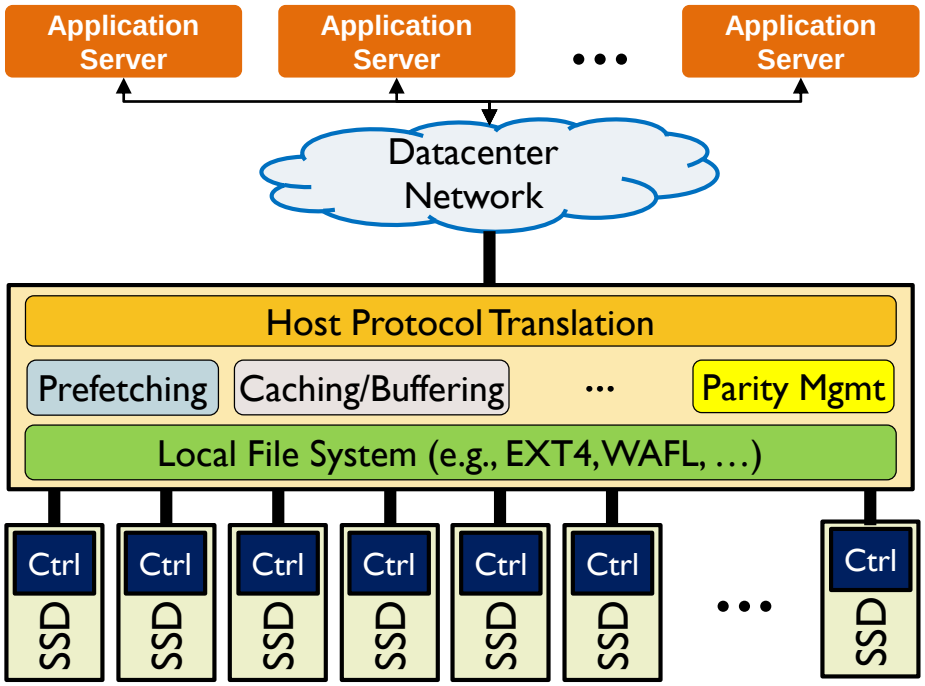
So… What is the Crazy Idea?

1. Get rid of a space-consuming, expensive, power-hungry host server
2. Put and run everything in SSDs
3. Attach SSDs to a datacenter network
4. Let application servers directly talk to SSDs



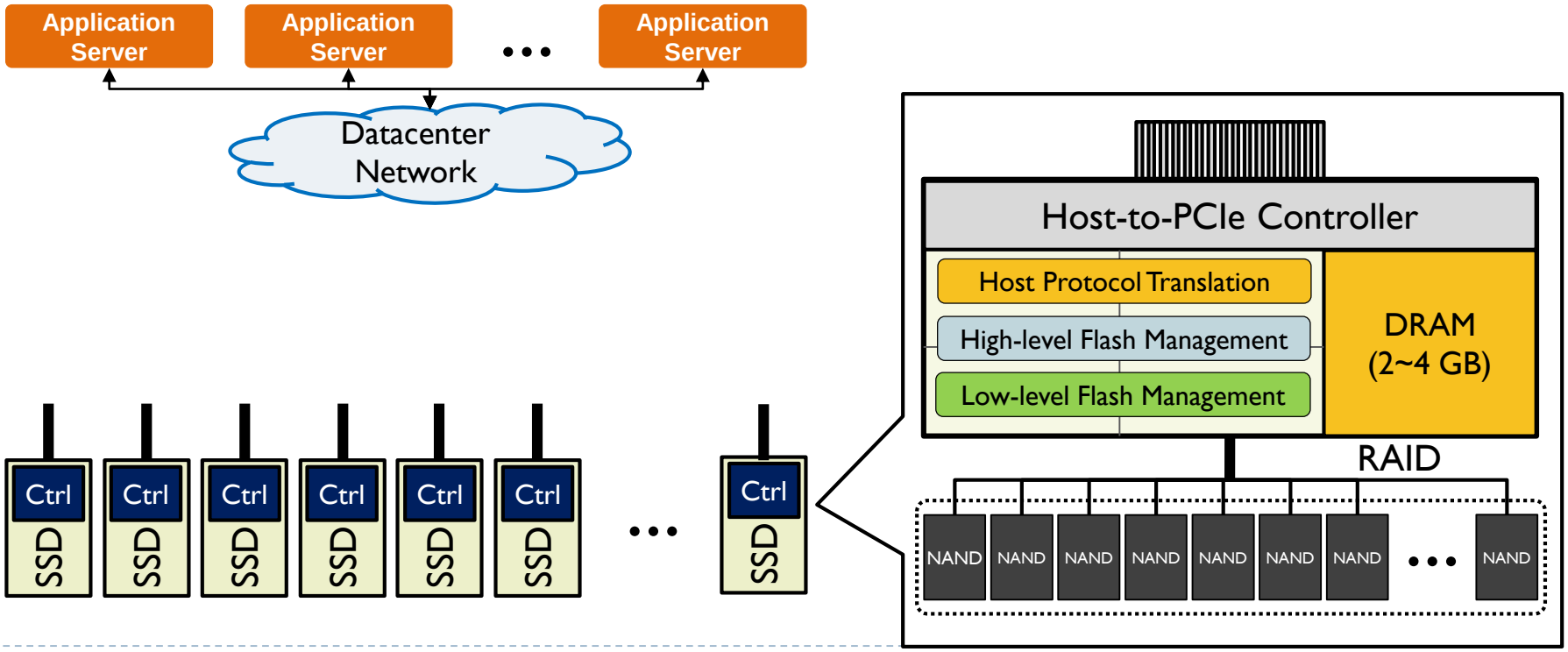
So… What is the Crazy Idea?

1. Get rid of a space-consuming, expensive, power-hungry host server
2. Put and run everything in SSDs
3. Attach SSDs to a datacenter network
4. Let application servers directly talk to SSDs



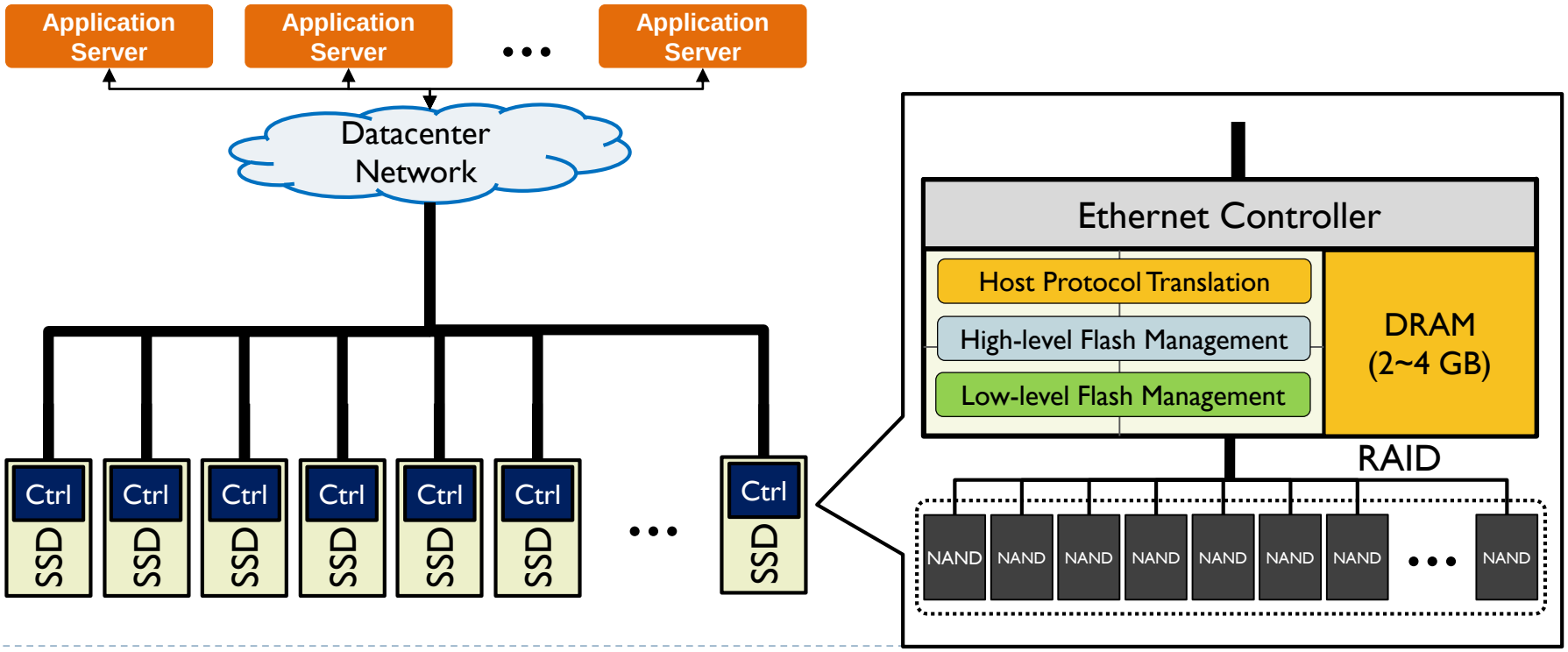
So… What is the Crazy Idea?

1. Get rid of a space-consuming, expensive, power-hungry host server
2. Put and run everything in SSDs
3. Attach SSDs to a datacenter network
4. Let application servers directly talk to SSDs



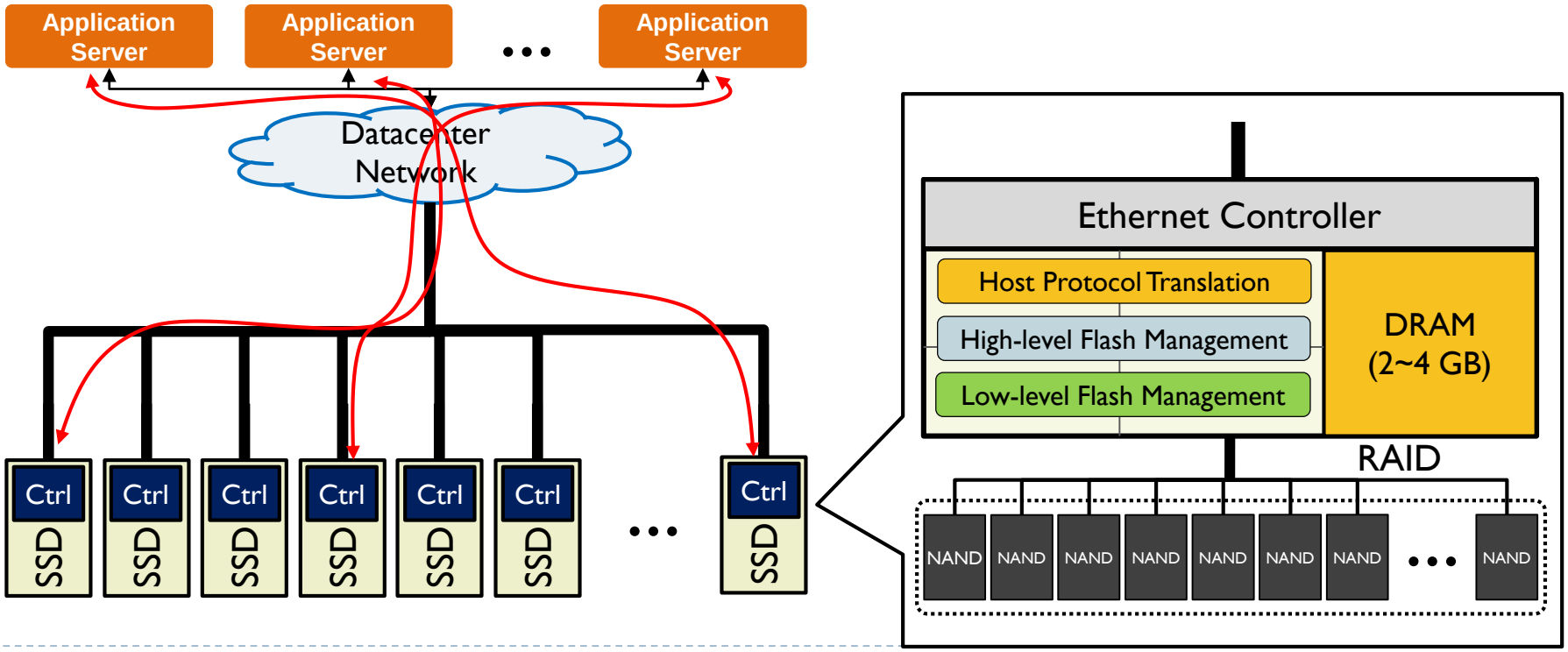
So… What is the Crazy Idea?

1. Get rid of a space-consuming, expensive, power-hungry host server
2. Put and run everything in SSDs
3. Attach SSDs to a datacenter network
4. Let application servers directly talk to SSDs



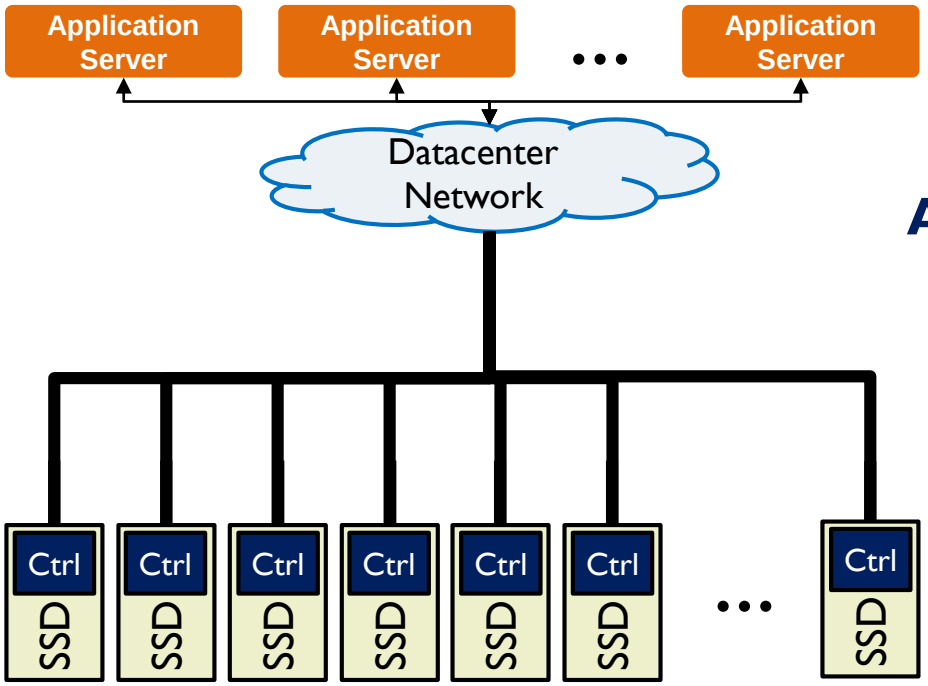
So… What is the Crazy Idea?

1. Get rid of a space-consuming, expensive, power-hungry host server
2. Put and run everything in SSDs
3. Attach SSDs to a datacenter network
4. Let application servers directly talk to SSDs



So… What is the Crazy Idea?

1. Get rid of a space-consuming, expensive, power-hungry host server
2. Put and run everything in SSDs
3. Attach SSDs to a datacenter network
4. Let application servers directly talk to SSDs

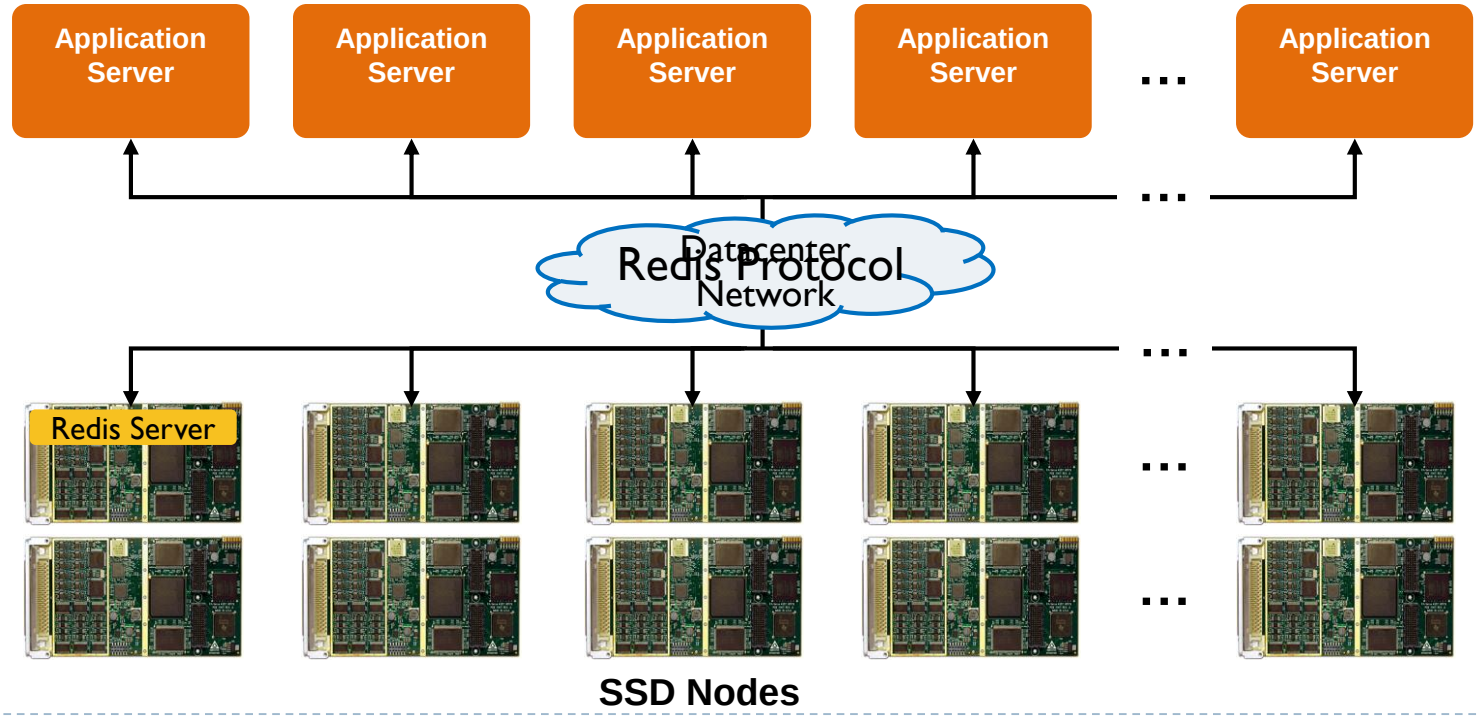


An x86 storage server with *N* SSDs is replaced with *N* SSDs

Low Power (e.g., 100 W)
Cheap (e.g., Zero server cost)
Small Volume (e.g., Less than 1U)
Low TCO (e.g., Less Cooling)
Scalability

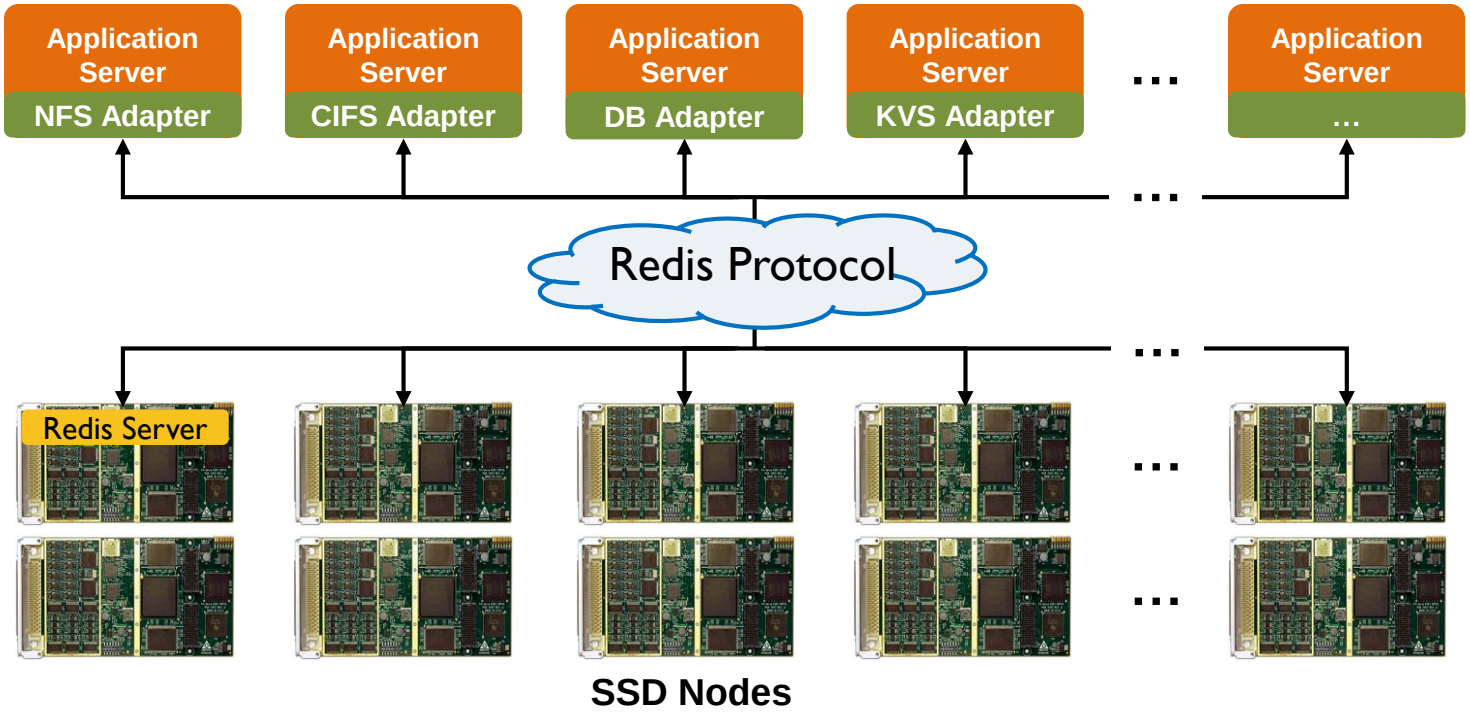
Technical Challenges?

- Which protocol is suitable for host interfacing?
  - **Redis KV Protocol** – Simple, Lightweight, Easy-to-implement, widely used in many applications, …



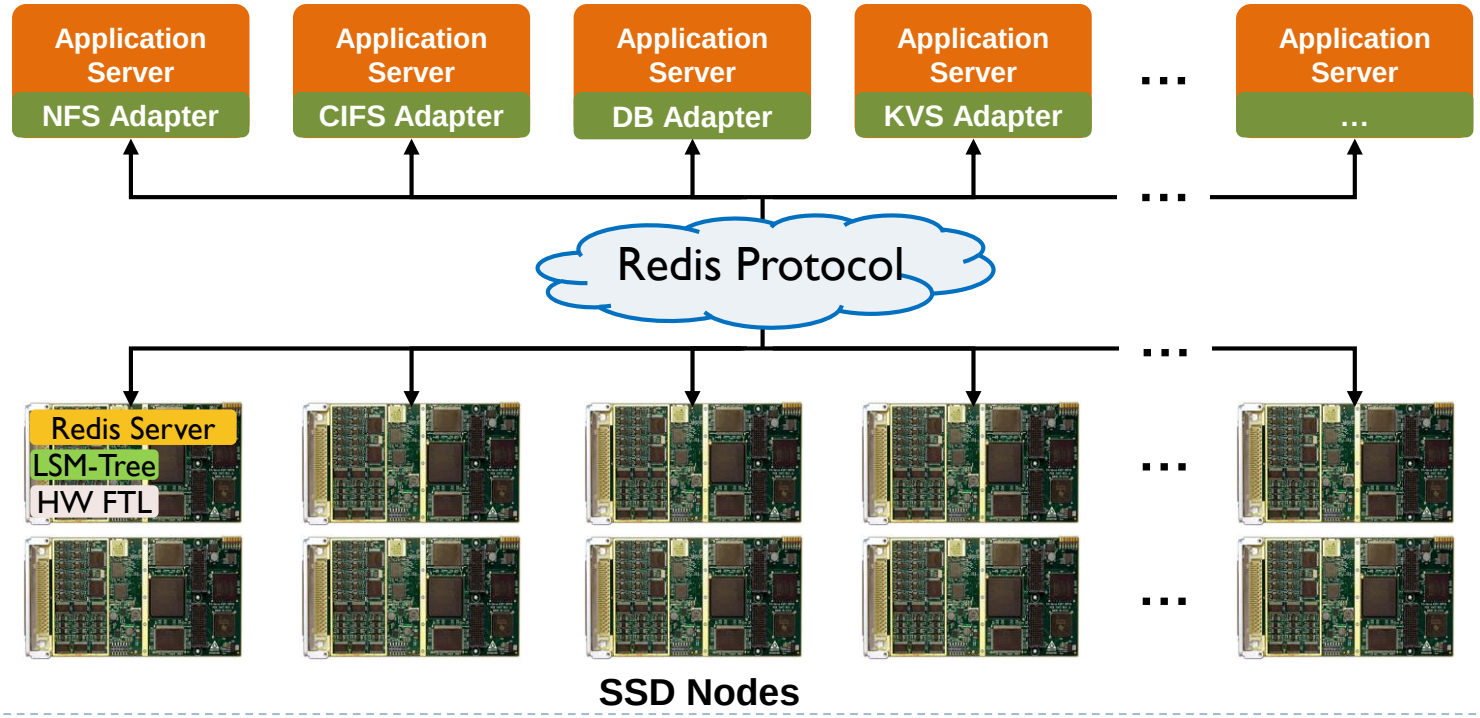
Technical Challenges?

- How to support various applications (e.g., KVS, NFS, SQL)?
  - **Protocol Adaptors** on the application server side
    - For file systems: NFS → Redis
    - For DB servers: DB → Redis



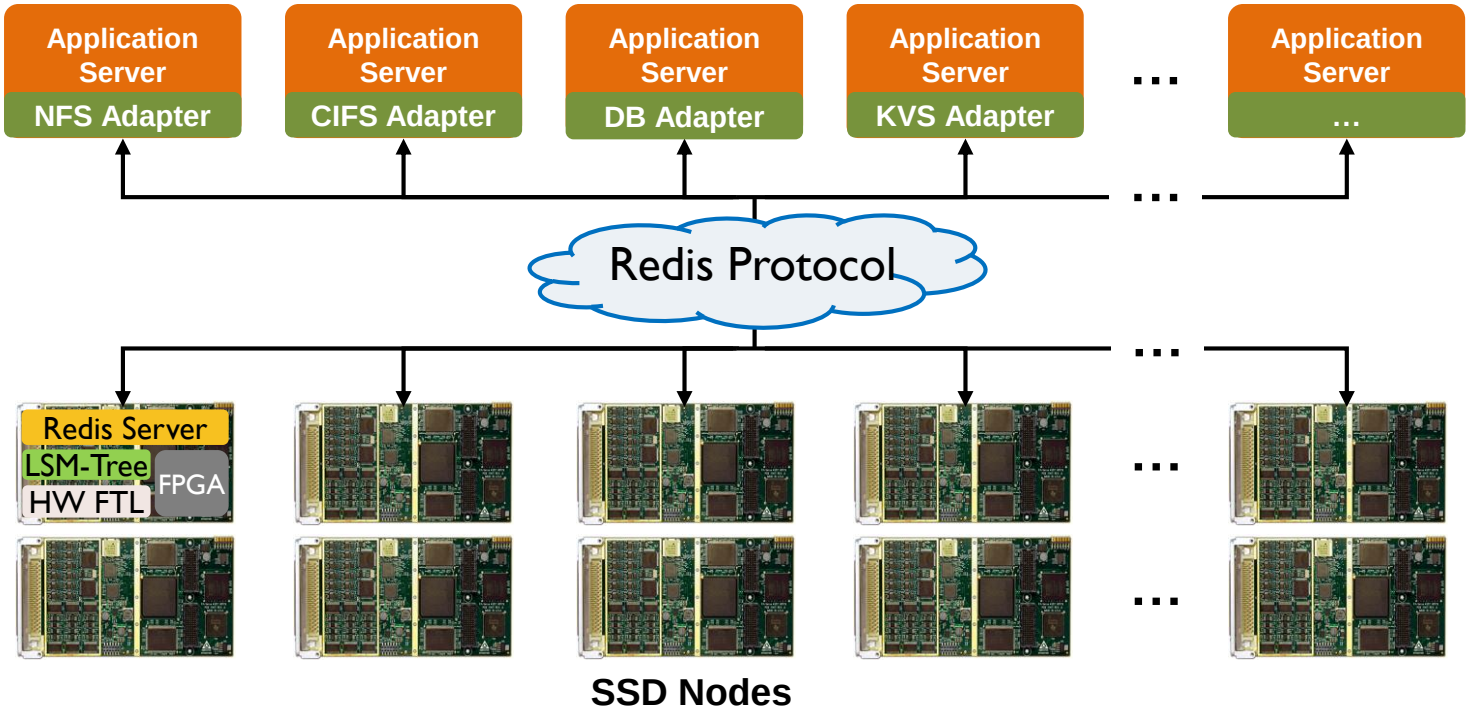
Technical Challenges?

- How to manage flash media
  - **LSM-tree algorithm**: Suitable for Redis-based KV store; Removes almost all of FTL functions
  - **Hardware FTL**: DRAM and CPU-less flash management



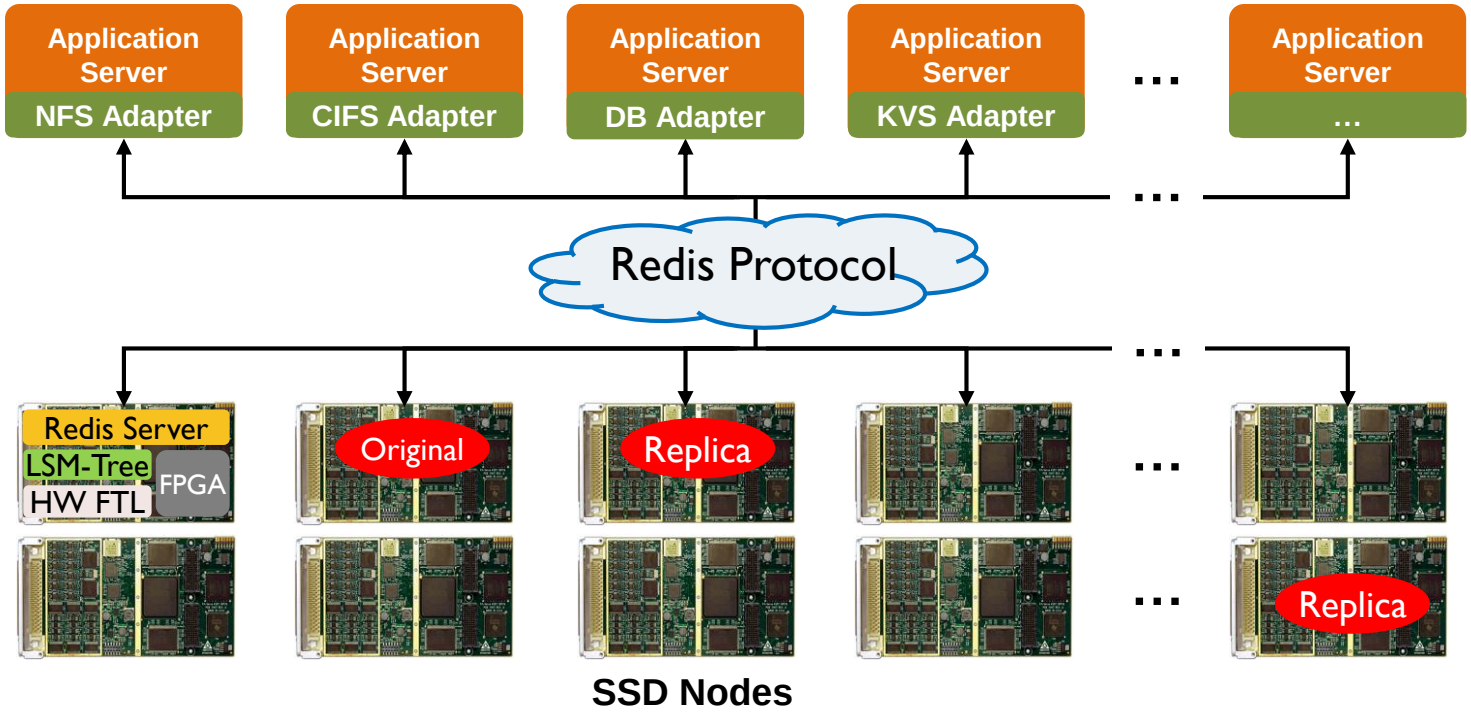
Technical Challenges?

- How to support CPU-intensive operations (e.g., dedup and compression)
  - Use **FPGA accelerator** – Low power & High performance
    - sits between cores and flash chip → data preprocessed in-path



Technical Challenges?

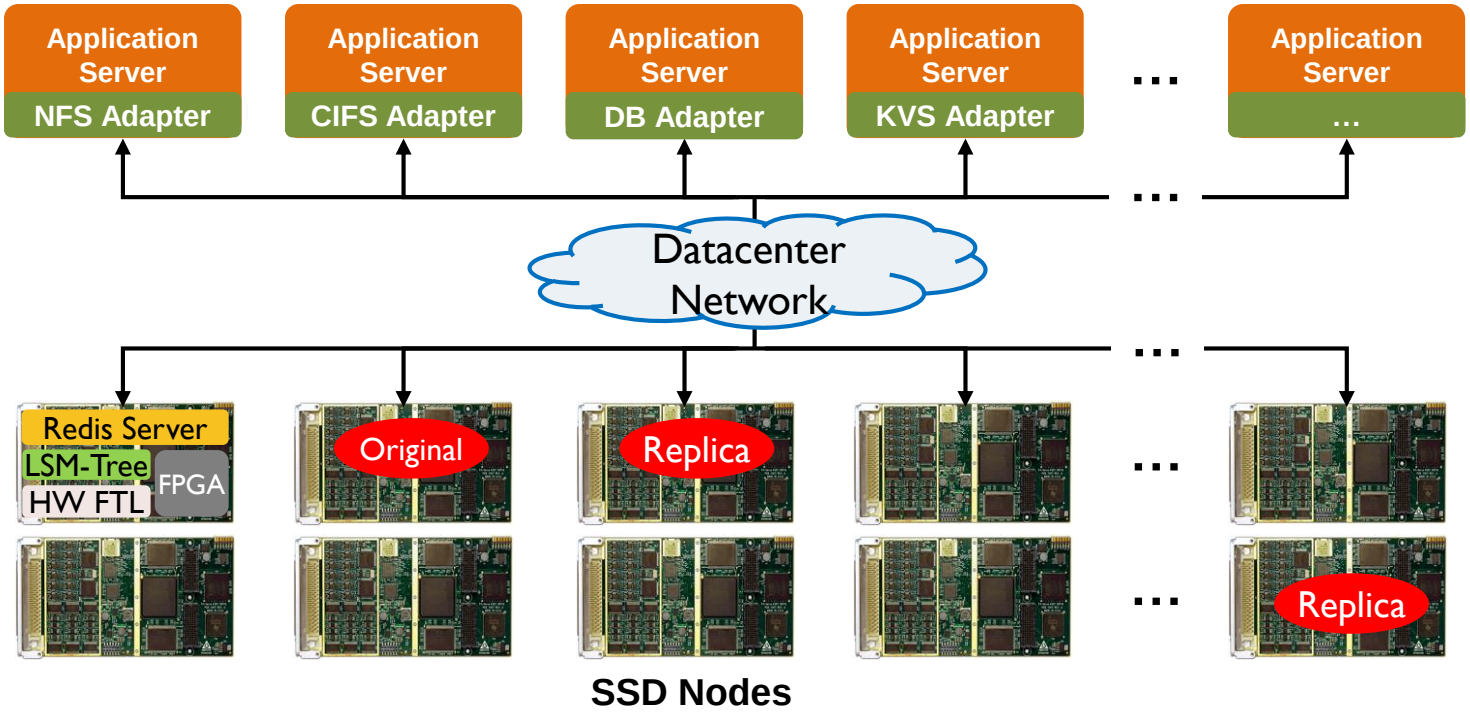
- How about data persistency/data availability?
  - Use a **data replication** feature of Redis, instead of RAID inside nodes



Conclusion

A storage server with *N* SSDs is replaced with *N new* SSD-sized nodes

Low Power (e.g., 100 W)
Cheap (e.g., Zero server cost)
Small Volume (e.g., Less than 1U)
Low TCO (e.g., Less Cooling)



Thank You!