Shift to Server Side Computing

- Computing은 shifting to very small / user-side and very big devices / server side
  - Improved UX
  - Advantage for Vendors
    → Simpler to make SW changes and improvement
  - Compute-intensive workloads는 server가 처리하기에 적절하다.

WSC

: Warehouse Scale Computer

- datacenter의 new 버전이다.

Data Center	Warehouse-scale Computer
—co-located machines that share power, security, environment, and network connectivity needs	—computing system designed to run massive internet services — e.g., Facebook, Amazon, …
—applications: a few binaries, each running on a dedicated, isolated HW infrastructure	—applications: tens of binaries running on hundreds to thousands of machines
—heterogeneous hardware and system software	—homogeneous hardware and system software
—partitioned resources, managed and scheduled separately	—common pool of resources that are managed centrally
—facility and computing equipment designed separately	—integrated design of facility and computing machinery

→ Program = executable on single machine

→ Program = an internet service

Addressing Network Bottleneck

WSC는 몇천개의 server로 이루어져있음.

Bandwidth : Cost per gigabyte port

- 48-port (rack) switch : $60 ✓ 선을 최대한 이용하려면!?
- 10,000 port (data center) switch : Priceless

deep hierarchy network 시

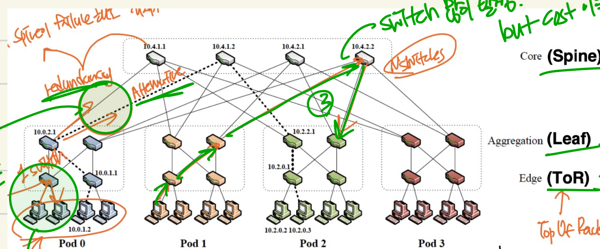
approach for lower-cost

· large-scale networking을 cheap cost로 하려면?) 선으로 여러개 이어진 등

- collection of small commodity switches

→ communication locality · failure concern 고려

switch 많이 발생. but cost 이득일수 있음!



Spine. / leaf / Top of rack

① communication 多 = co-locate … ToR은 failure시 shutdown 가능

② 다른 pod 연결 redundancy 이용해서 shutdown 방지

③ switch

→ network bottleneck 피하기 위해서 redundancy 이용!

Flash (SSDs) in Warehouse-Scale Machines

Disk

Flash value proposition

- DRAM을 대체 하기엔 아직 부족해 보이지만, 많이 향상됨

- 여전히 storage에서 data특성에 따라서 다르게 접근해야한다.
  - Cold / warm / hot data로 type 나눠서
    cold data가 위주로 communicate 하지 않음

Storage hierarchy

- Complexity of storage hierarchy 는 점점 더 커진다 ㅎ

Performance Variability · Faults

→ Rare event가 되더라도, scale 커질수록 많이 걸리는 것

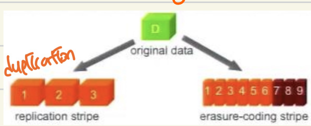
Performance Variability

- large scale에서는 Performance variability가 크다
- Rare slowdown events can hurt
  - ex) Avg server response가 10ms인데, every 10K query마다 1s 걸림 (random performance hiccup)
    그러면 1/10000 확률로 slow down 됨
    → 이 경우, 1 machine = 0.01% slow queries
    5K machine = 50% slow queries ← 최악!
- Minimizing performance idiosyncrasies (특이성)가 중요하다 = rare한 것 잡기!

Service Level Outages (정전, 통신장애 등)

- Causes
  - networking issues
  - Power issues
  - oops issues (automated / manmade mistakes)

- Solution
  - Storage Availability



→ duplication 여러개 두는 것
  - different disk / server / racks

→ distribute data blocks across whole cluster
  - load-balanced read 가능
  - recovery from faults도 easy

Demands of GPU clusters

Demands

- Resource handling / management
  - DL training on GPUs
  - high-speed network
- Failure handling
- Storage · data handling
  - sharing identical data
  - reusing checkpointed models

> locality 등에도 Zoi 중요!

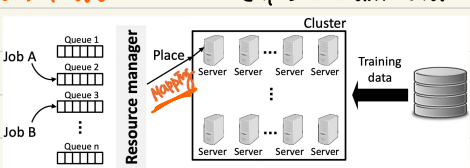
GPU cluster Architecture

- cluster : 100s of servers and thousands of GPUs
- Distributed filesystem : shared storage
- Resource Manager

↓

collective 하고 fairness / efficiency 간의 trade-off.

- Job queue : resource (GPU) amount allowed for each group

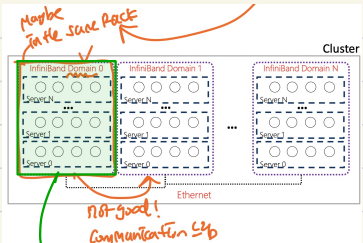


→ idle resource management, load-balancing handling

Demands of GPU clusters (Cont.)

Communication Heterogeneity

- Communication frequency 높은 애들끼리 rack-localized 시켜주는게 좋음

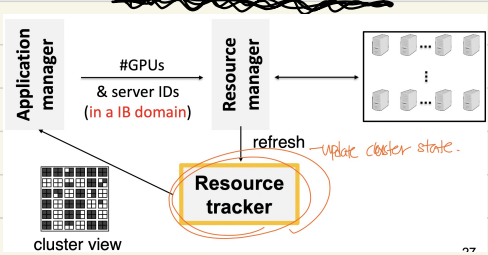


high-bandwidth network … rack localized ✓

Job Placement

- Job's GPU 할당할때도 가까운 곳에 있는 server 사용하도록! → locality 고려
- " within single InfiniBand domain

- Resource Negotiation
  - each Job은 각자가 resource 할당에 대해 application manager가 협력.



  - Intra-server (서버내부) locality에 따라서 ...