Storage Server

Many DL Workloads Dominated by I/O

- each GPU easily consumes 1 GByte/s
- 16 GPU machine = 16 GByte/s
- hardware
  - 150 MBytes/s rotation storage
  - 3 Gbytes/s SSD
  - 5 Gbytes/s Ethernet/Infiniband

Distributed Storage

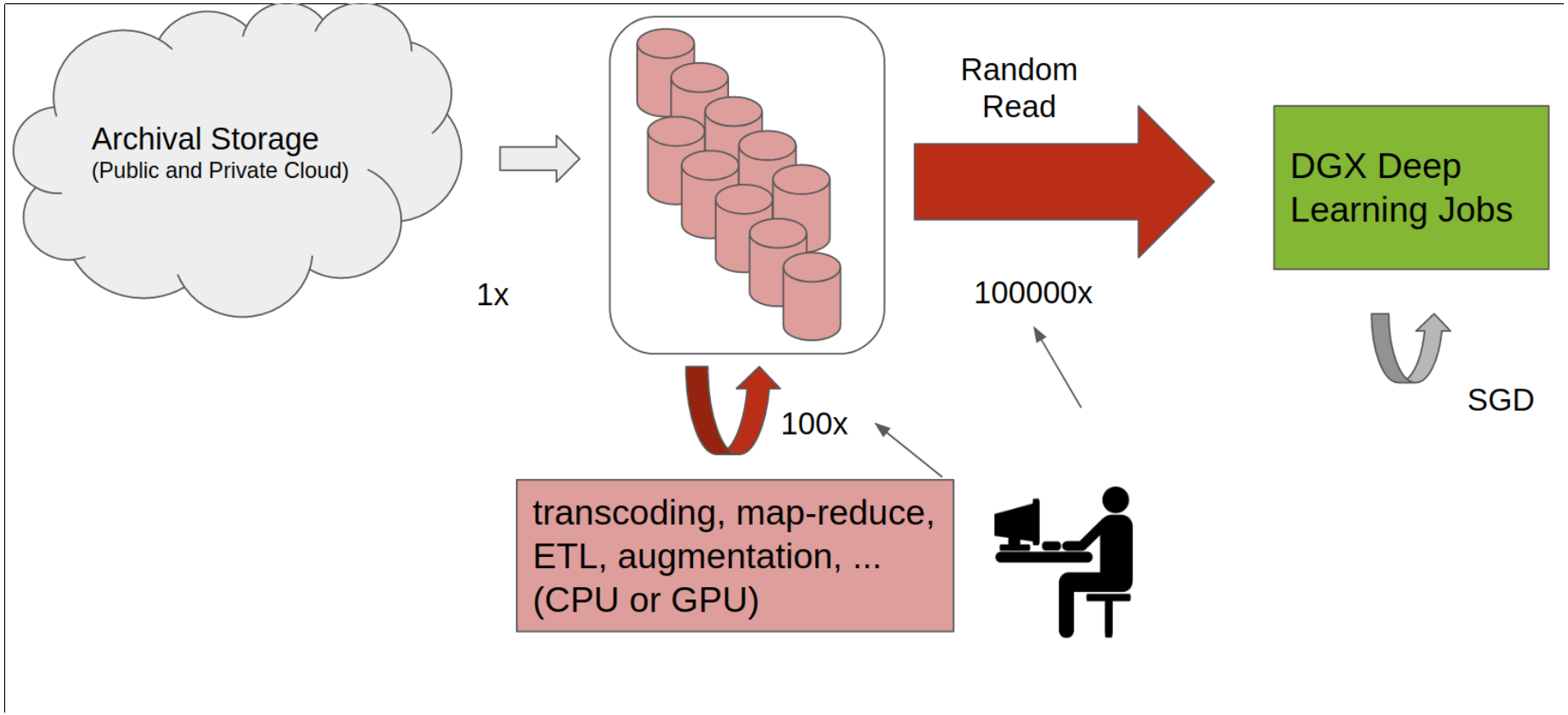
Architecture dependent on application. Classes of technologies:

- full POSIX semantics
  - fully consistent, read, write, seek, partial updates, etc.
- partial POSIX semantics
  - behaves mostly like POSIX but not fully consistent
- object store only
  - get object / put object only

Other Considerations

- loss of drives vs loss of data, recovery
- loss of servers vs loss of service
- access patterns
- effectiveness of caching

Access Patterns for DL/AI



Deep Learning

Status quo:

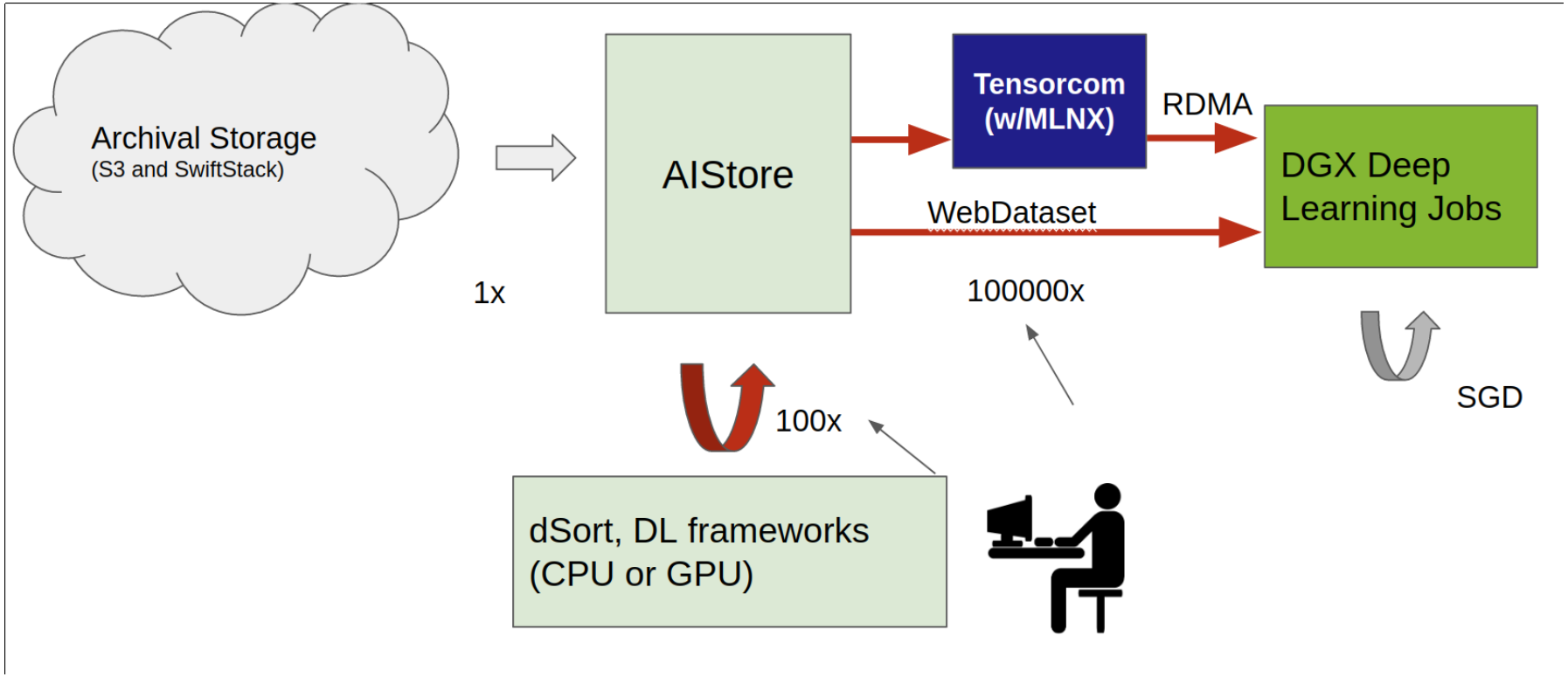
- `torchvision.Imagenet`, `FileDataSet` assume POSIX semantics
- access/caching pattern is unusual
- works fine on local SSD
- very slow on NFS, other distributed filesystems

`WebDataset` only uses whole object sequential reads

Access Patterns for DL/AI with Sequential Storage

- repeatedly read through entire dataset
  - perform ETL, augmentation, map-reduce, sorting
  - perform training
- data divided into 1 Gbyte shards, each read fully sequentially
- local caching not helpful or used

Map of Technologies



- proven technologies, re-implemented with DL focus and using standard formats
- use for production and/or use as didactic example to understand other systems

AIStore

(We'll talk about the client libraries separately.)

- provides an infinitely scalable object store for DL/AI applications
- uses HTTP(s) for all communications
- provides additional server-side support for DL/AI applications
- can be deployed in Kubernetes
- even if you don't use it, illustrates important concepts in high performance storage

AIStore and Failure

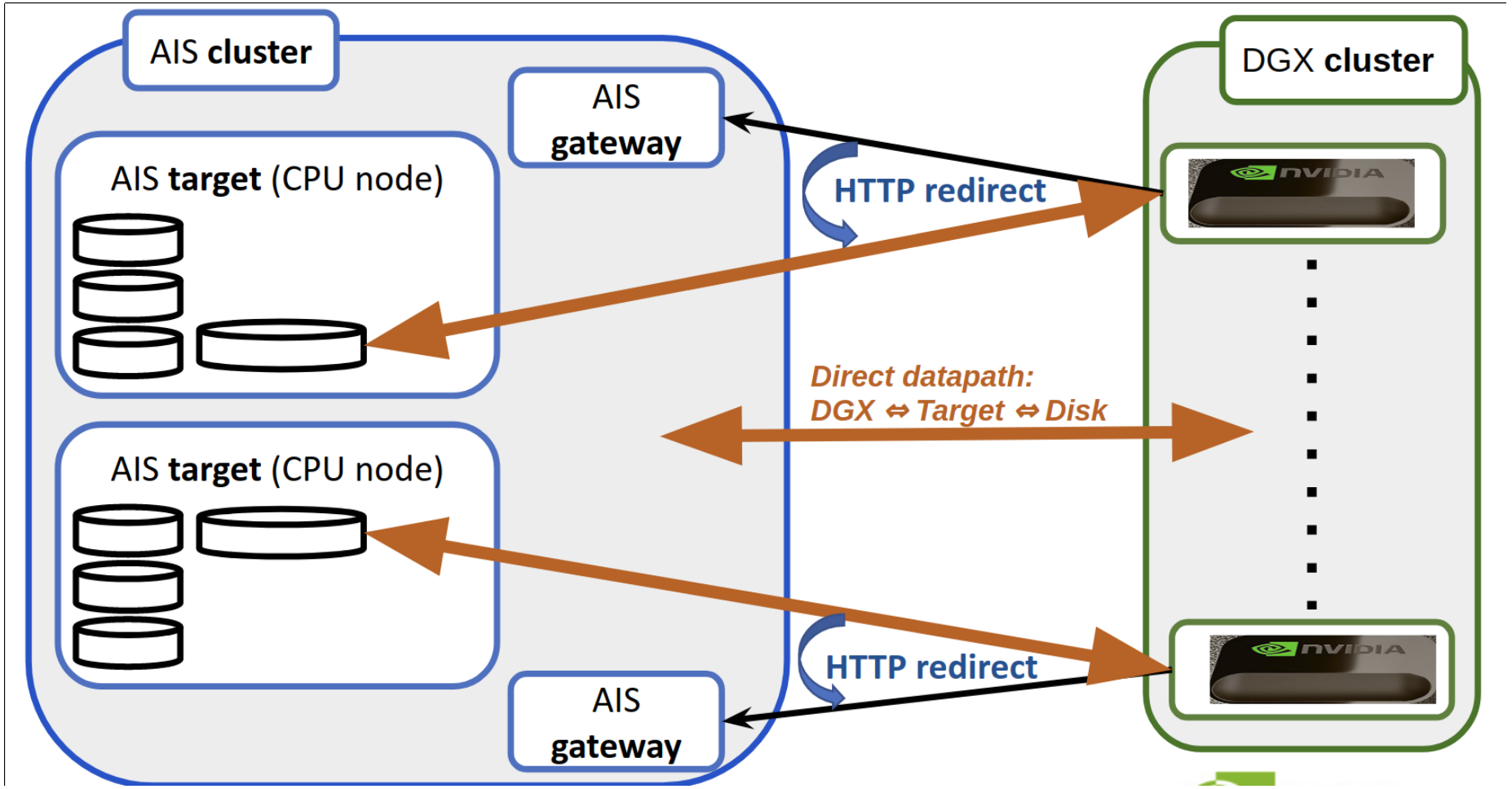
Two modes of recovery from drive failure:

- reconstruct via *erasure coding*
- retrieve from upstream storage (if used as a cache)

Continued operation in the presence of hardware failures:

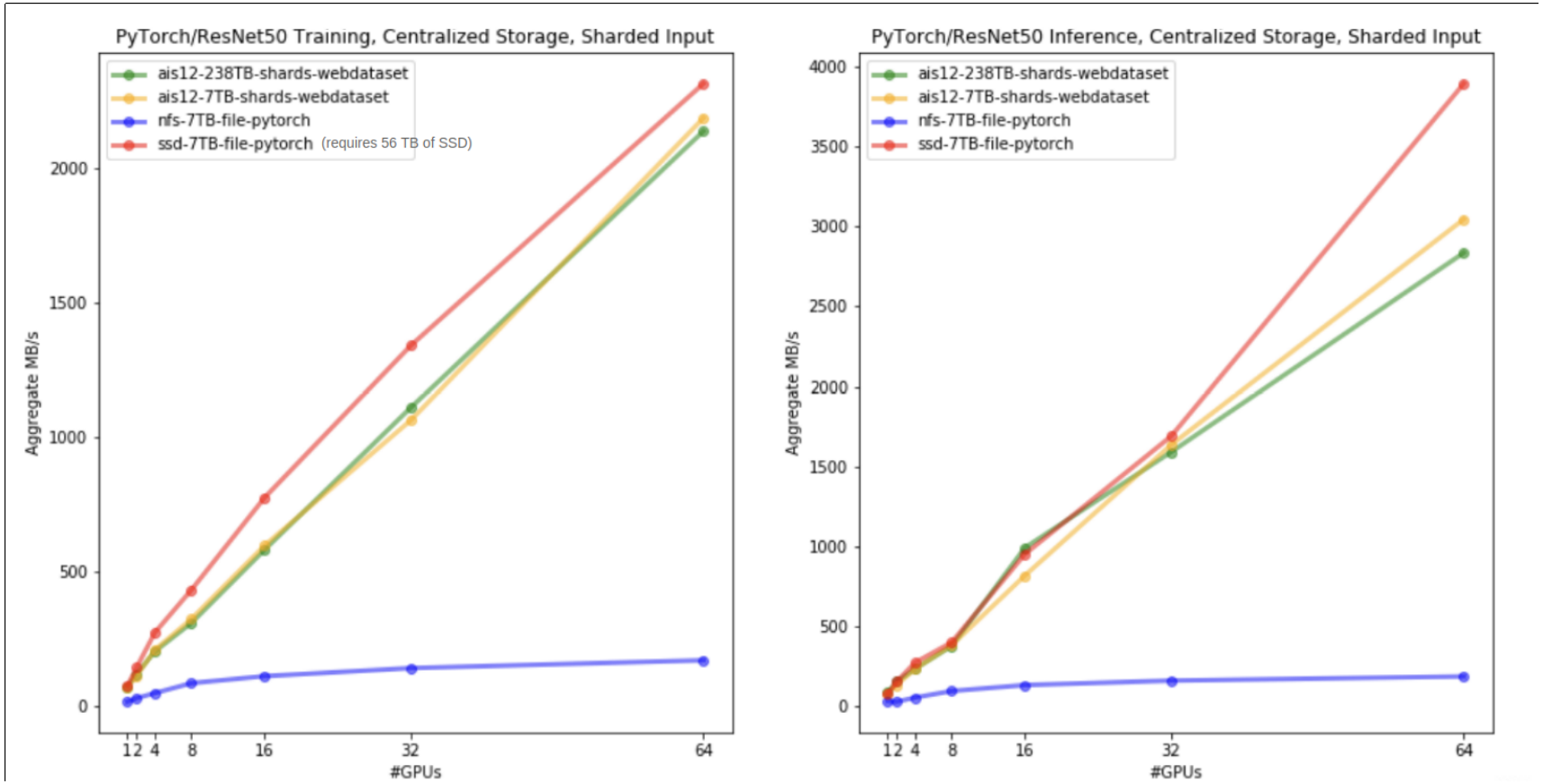
- use *consistent hashing* to redirect to working server
- server initiates recovery from erasure coded parts or upstream server

AIStore Request



- NB: no bottlenecks

AIStore Performance



- AIStore achieves performance comparable to local SSD on large scale datasets
- AIStore can take maximum advantage of all available drive hardware bandwidth

AIStore and WebDataset

- illustrate important concepts in efficient large scale storage
- fully open source, open standards based solutions
- uses cheap commodity hardware, rotational drives (can use SSDs as well)
- easy migration of existing apps & data
- unlimited scalability (both out and down)