Distributed Storage

The YouTube8m Dataset

- 6.5 million videos
- 300 TB of data
- 1 PB when transcoded, normalized
- applications: unsupervised and semi-supervised learning
- assume pre-sharded into 100000 10G shards `chunks-{000000..099999}.tar`

Datasets Larger than Local Storage

- 7-32 TB of local storage
- need to replicate dadtaset across all nodes for local I/O
- 64 GPUs w/DGX-1 = 8 nodes = 56-240 TB of local storage

Limits:

- largest replicated dataset: 32 TB
- largest partitioned dataset: 240 TB
- anything larger → distributed storage

Accessing Sharded Distributed Storage

Easy with WebDataset:

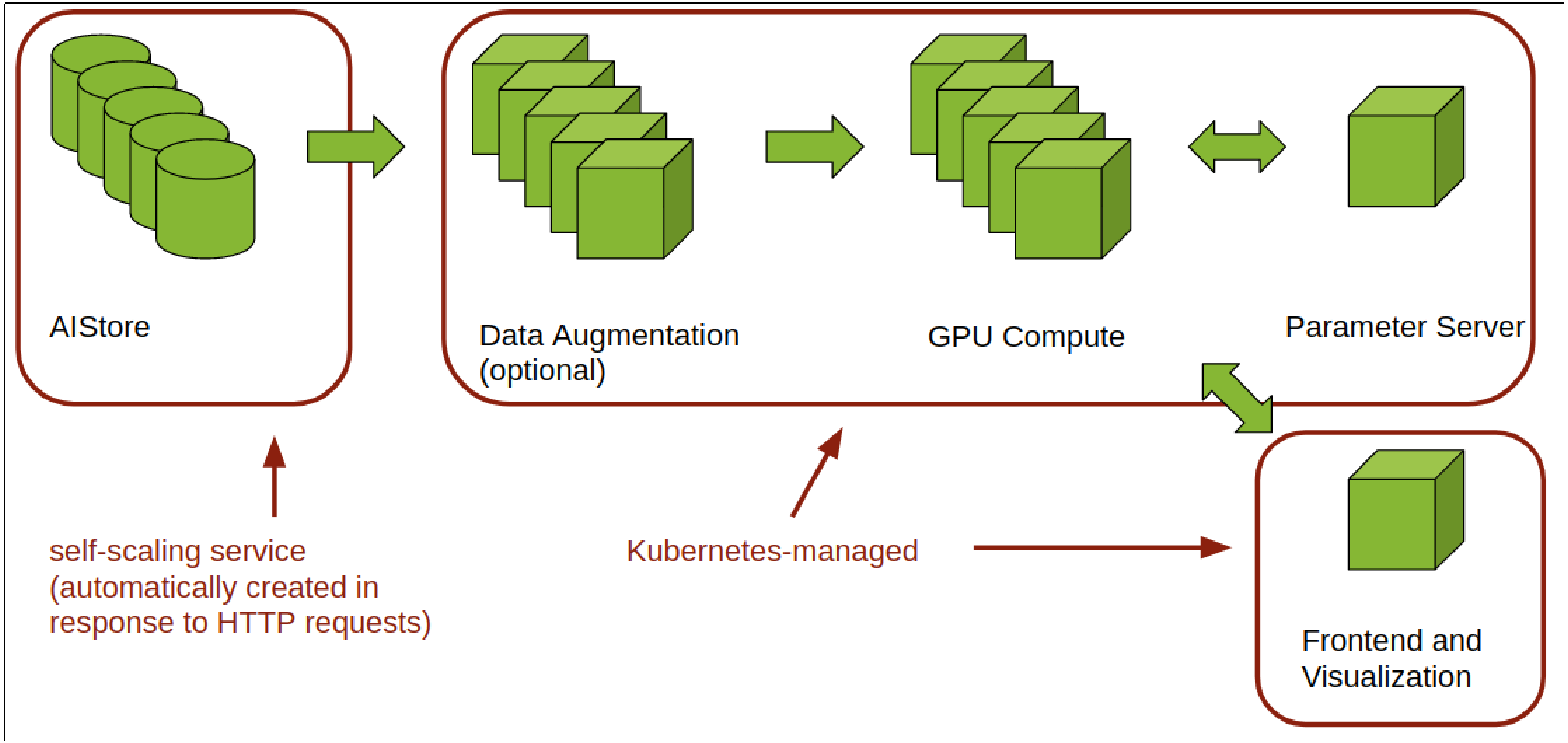
```
samples = WebDataset("http://aistore/yt8m/chunks-{000000..099999}.tar", extensions="mjpeg json)
for clip, json in samples:
    ...
```

- code is otherwise the same as in multi-node training with local replicated data
- performance is as good as local SSD

Kubernetes

- start and stop processes on a cluster of machines
- encapsulate processes in containers
- provide a virtual network overlay
- provide access to hardware resources
- provide job and pod management

Kubernetes for DL


AIStore
Data Augmentation (optional)
GPU Compute
Parameter Server
Frontend and Visualization
self-scaling service (automatically created in response to HTTP requests)
Kubernetes-managed

Kubernetes Pod

- Kubernetes jobs/daemons/services are specified in YAML.
- They are applied with the `kubectl apply` command
- Basic unit is a Pod, a collection of Docker containers ("a pod of whales")

```
apiVersion: v1
kind: Pod
metadata:
name: mypod
spec:
containers:
- name: mypod
    image: gcr.io/research-191823/bigdata19
    command: ["nvidia-smi"]
    resources:
    limits:
        nvidia.com/gpu: "1"
restartPolicy: Never
```

Kubernetes Setup with `kubetpl`

Job specs are complex and contain many repeats; templating helps:

GPU-based Job:

```
$ kubetpl job -G 1 -c nvidia-smi | kubectl apply -f -
```

CPU-based Pod:

```
$ kubetpl pod -l tmbdev/redis-server -c redis-server | kubectl apply -f -
```

Multi-GPU Training

- `DataParallel`, multiple I/O streams
- `DistributedDataParallel`, single node, multiple I/O streams
- `DistributedDataParallel`, multiple nodes, multiple I/O streams

Tools

- Ansible -- general setup and deployment of services
- Helm -- deployment of servers/services
- Kubeflow -- machine learning pipelines
- ... many other workflow systems ...

Distributed Training

(notebook)