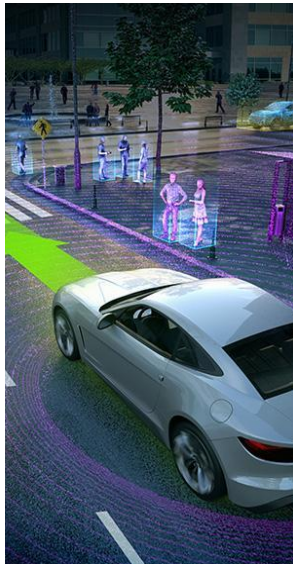
Production GPU Clusters with K8s

Madhukar Korupolu
NGC Team

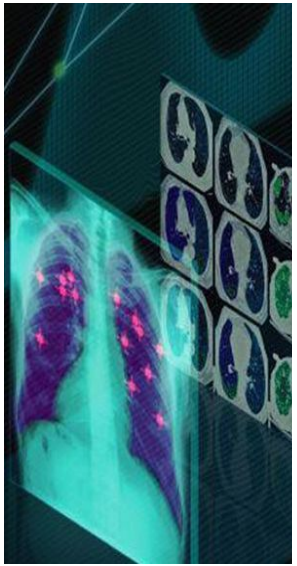
Massively Growing Computing Needs



AUTONOMOUS DRIVING

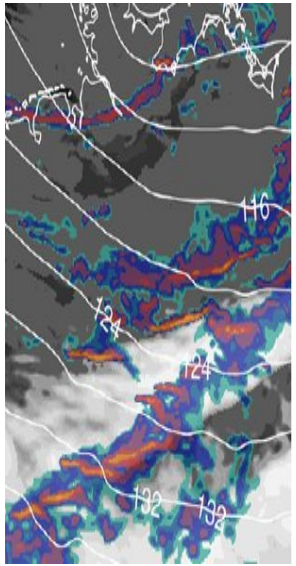
ASTROPHYSICS

GENOMICS



MEDICAL IMAGING

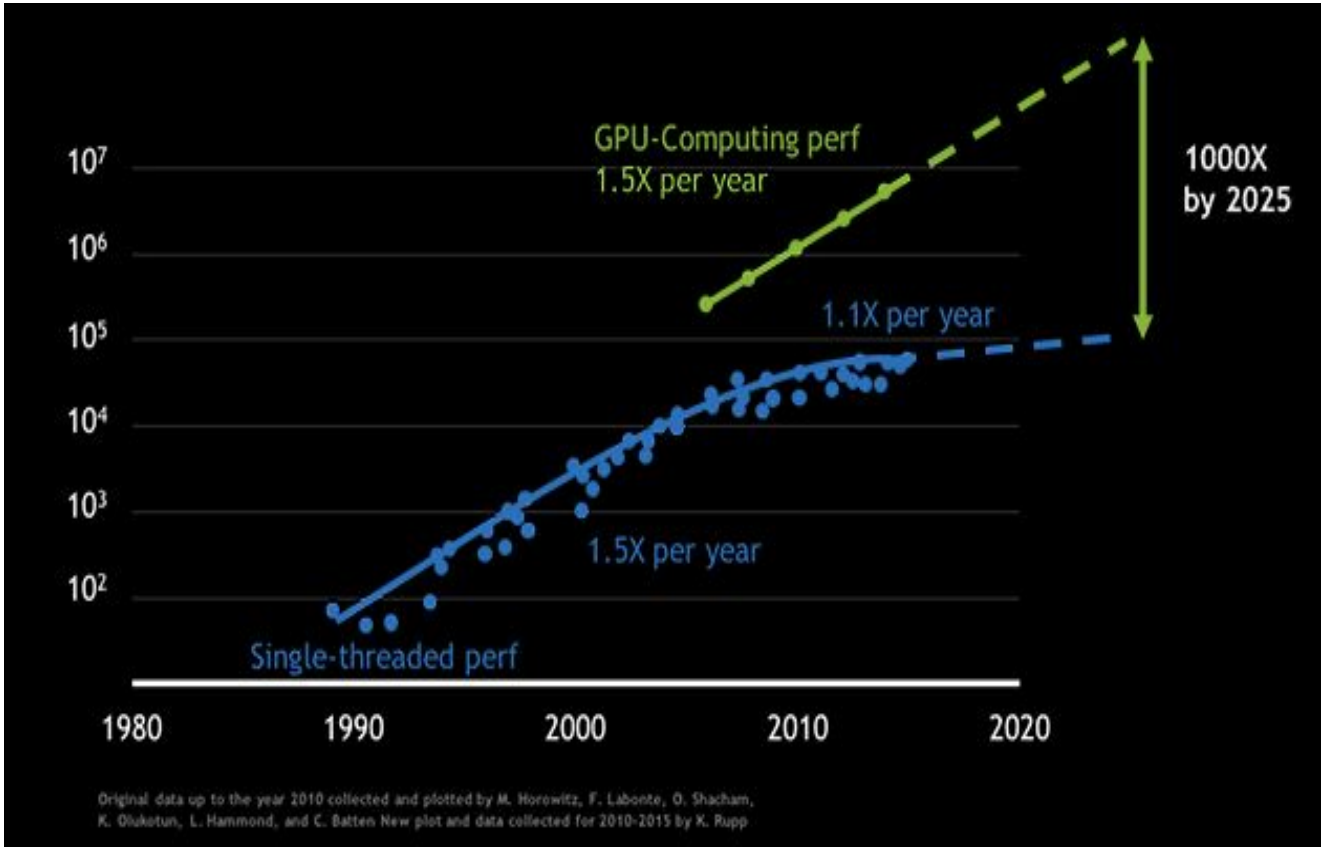
NUCLEAR FUSION



WEATHER

Big Data, IoT, AI, Deep Learning, Machine Learning, HPC

GPU Acceleration and Trends



Parallel computing & CUDA

GPUs speed up DL and HPC tasks from weeks to hours or less [2012+]

GPU-accelerated Containers

Containers & GPU-accelerated Applications

Complex s/w dependencies (e.g., DL)

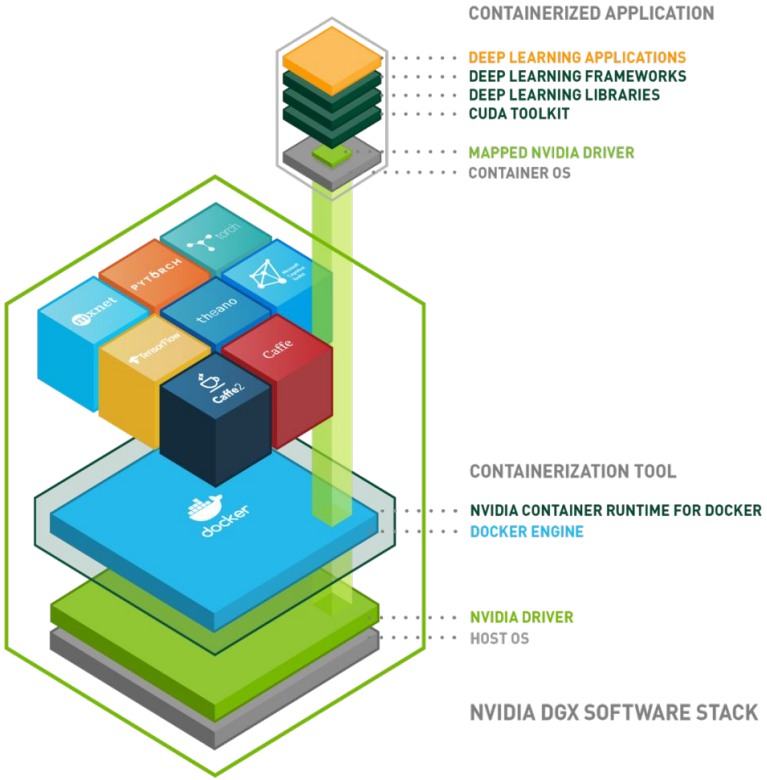
- CUDA toolkit, DL libraries (cuDNN, cuBLAS..)
- DL frameworks: TF, PyTorch, Caffe etc
- NCCL, Open MPI, Horovod etc

Containers to the rescue!

- Pre-packaged dependencies

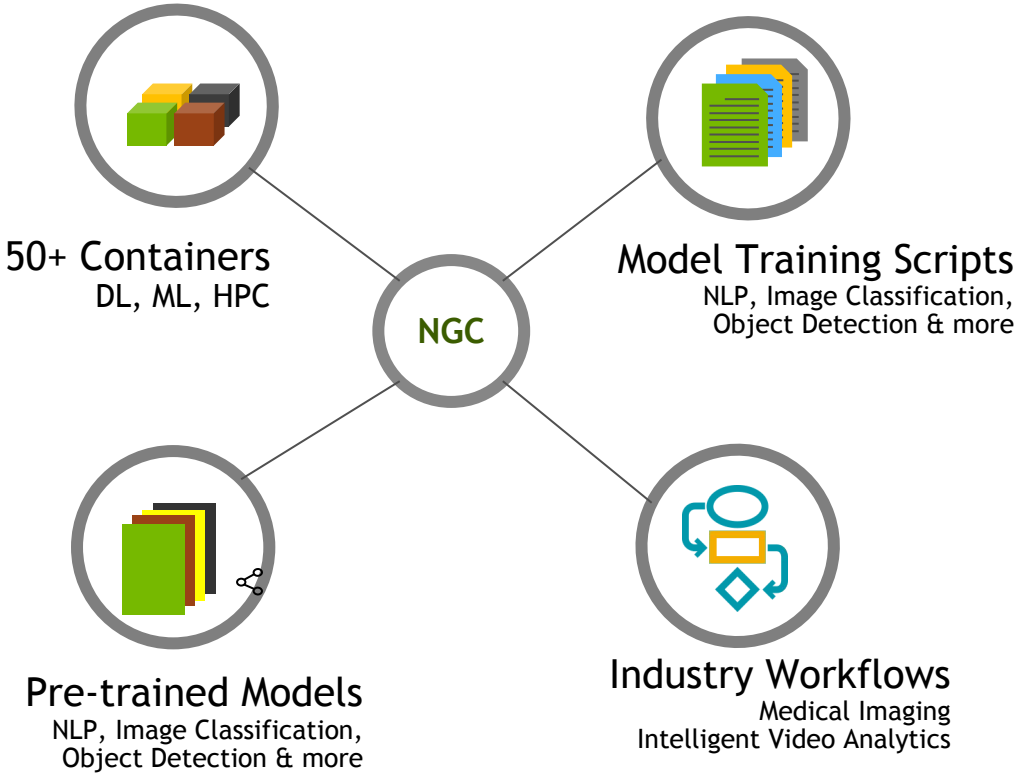
All-in on containers

- Simplify deployments, Quick start, Portable



NGC Registry: GPU-accelerated Software Hub

Containers and models for AI and HPC



- https://ngc.nvidia.com
- Tuned, tested, certified to run on Nvidia GPUs
- Monthly releases
- Software perf improvements

NGC: GPU-accelerated Containers for AI & HPC

Deep Learning	HPC	Machine Learning	Inference	Visualization
Caffe2	BigDFT	Dotscience	DeepStream	CUDA GL
Chainer	CANDLE	H2O Driverless AI	DeepStream 360d	Index*
CT Organ Segmentation	CHROMA*	Kinetica	TensorRT	ParaView*
CUDA	GAMESS*	MapR	TensorRT Infr Server	ParaView Holodeck
Deep Cognition Studio	GROMACS	MATLAB		ParaView Index*
DeepStream 360d	HOOMD-blue*	OmniSci (MapD)		ParaView Optix*
DIGITS	LAMMPS*	RAPIDS		Render server
Kaldi	Lattice Microbes			
Microsoft CNTK	Microvolution			
MXNet*	MILC*			
NVCaffe	NAMD*			
PaddlePaddle	Parabricks			
PyTorch*	PGI Compilers			
TensorFlow*	PIConGPU*			
Theano	QMCPACK*			
Torch	RELION			
TLT Stream Analytics IVA				

- Docker and Singularity, HPC-CM
- *Multi-node enabled, NCCL/MPI
- **Monthly releases, perf updates**

Simplify Deployments

Innovate Faster

Deploy Anywhere

Atos · cisco · CRAY · DELL EMC · aws · Azure · Hewlett Packard Enterprise · Lenovo · NVIDIA · SUPERMICRO · Google Cloud · ORACLE CLOUD PLATFORM

https://ngc.nvidia.com

NVIDIA

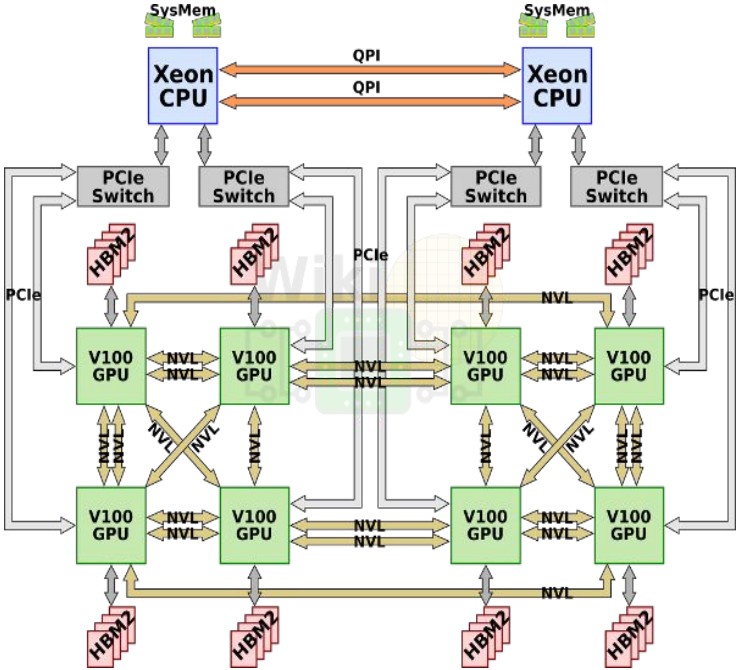
Overview

- ✓ Motivation: AI / DL / ML / HPC
- ✓ NGC registry: GPU-accelerated container hub
- Internal Production GPU clusters
- Orchestration using K8s: challenges and approaches
- Learnings, best practices and next steps

Internal GPU Clusters

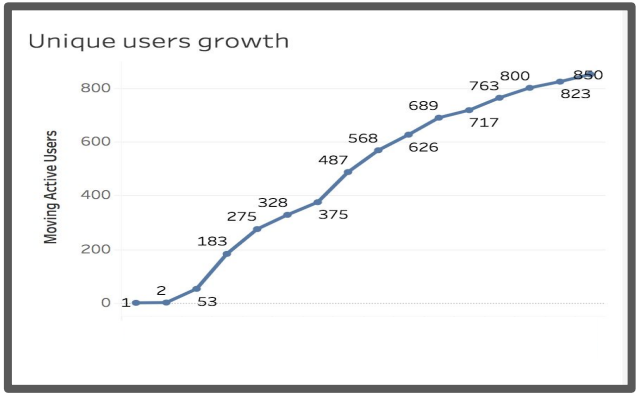
Nvidia Saturn-V: Internal GPU Clusters

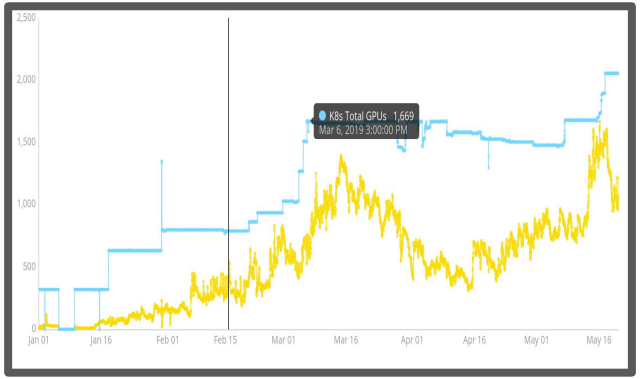
- **Approx 14,000 GPUs total**
- DGX-1 and DGX-2 GPU nodes
- Racks, pods, CPU, storage nodes
- Few different data centers, clusters
  - Mix of dedicated and shared
- Bare-metal / Xen VM (legacy)
- Slurm / Mesos (legacy) / K8s (NEW)

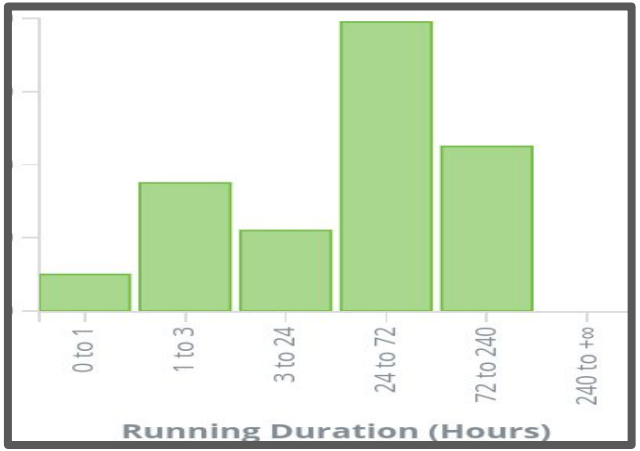


DGX-1 node with 8 GPUs

Cluster Users and Workloads







Internal AI / DL / ML / HPC / Viz teams

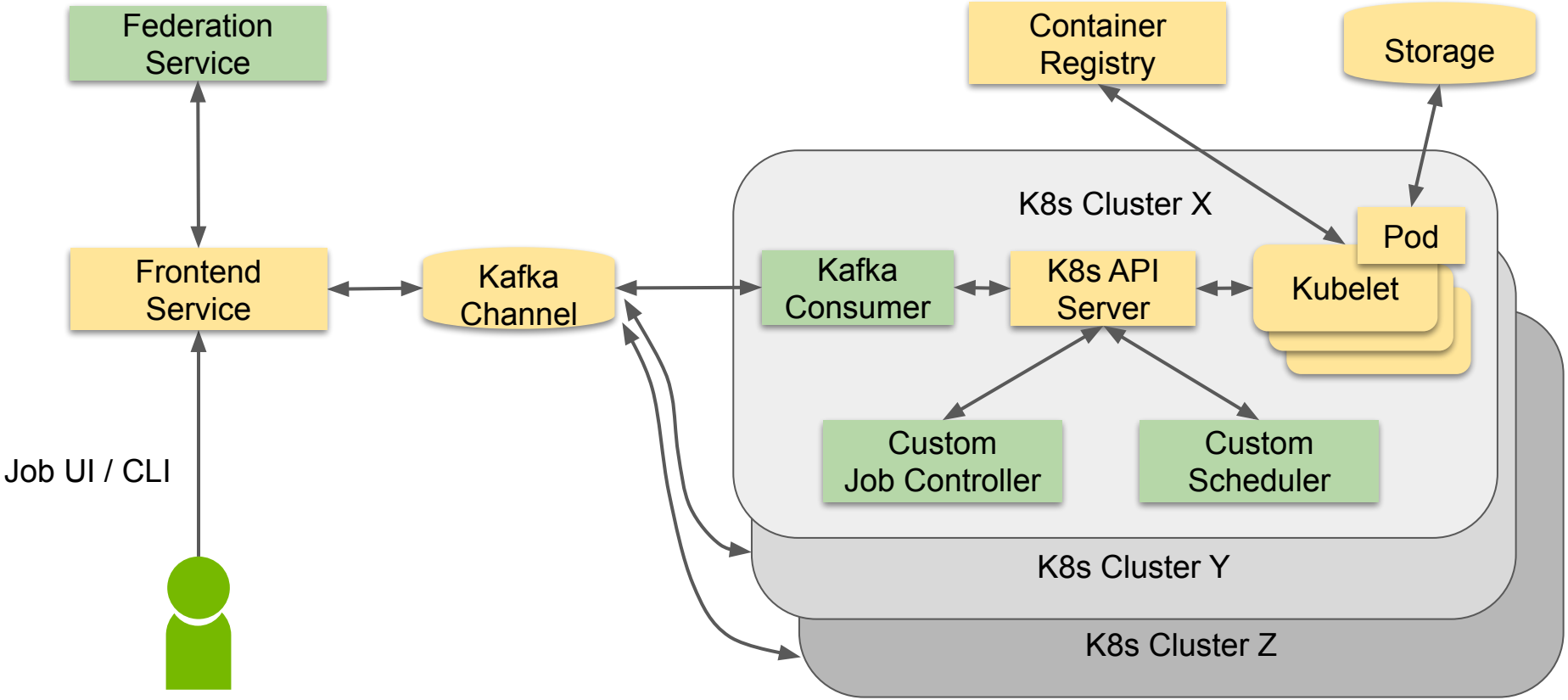
Data scientists, Container engrs, Verticals

Batch jobs: long running (days, weeks)

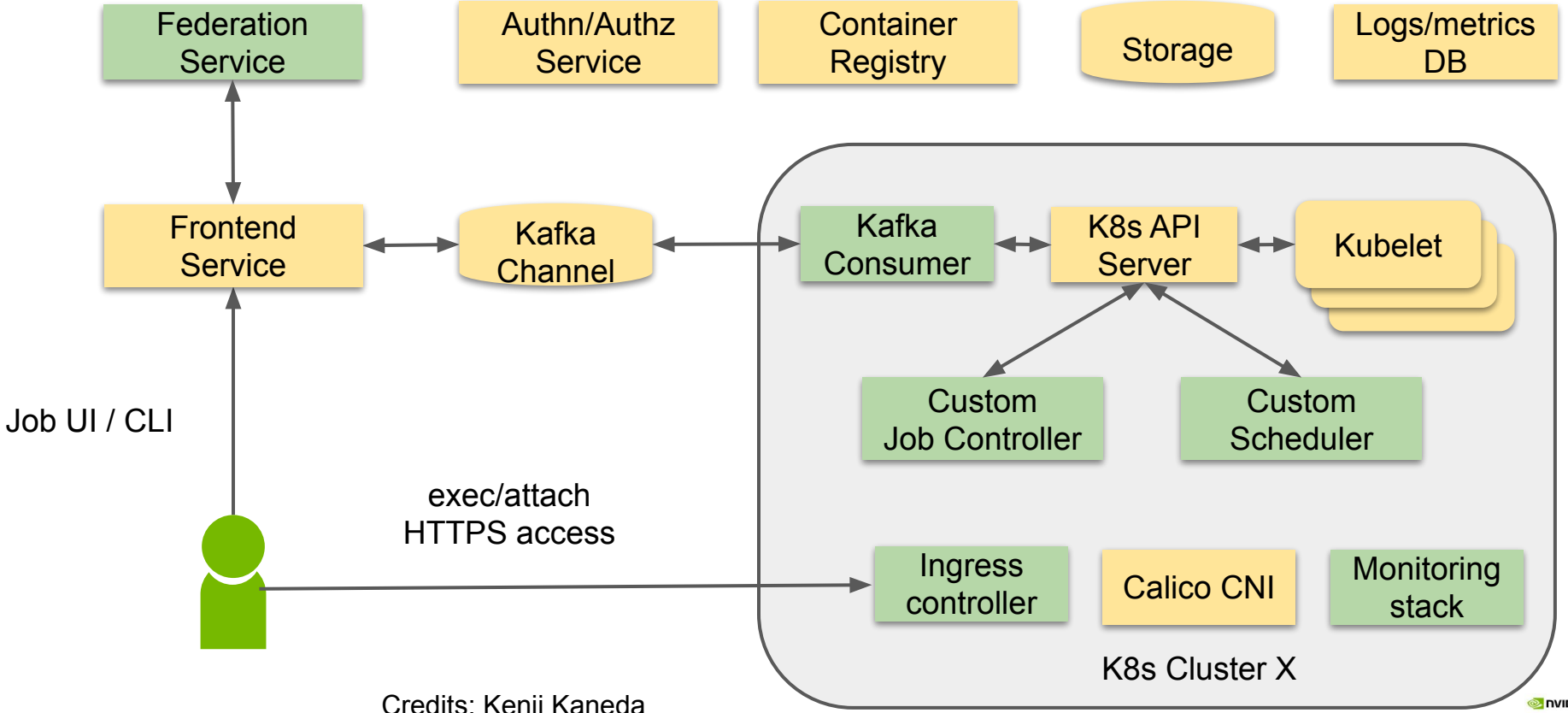
Size: {1, 2, 4, 8} gpus per job; multi-node in some

Mission-critical: SRE team, reliability, yield, perf

Job Submission Flow



Architecture Overview



Credits: Kenji Kaneda

Challenge 1:

HPC / Batch Management
Long running Jobs

K8s and HPC/Batch Limitations

- K8s originally designed for web service workloads

- Multi-tenant + HPC / Batch requires:
  - Time limits and slices
  - Queues, quota, fair share
  - Node topology / affinity
  - Multi-node job
  - Checkpointing, preemption
  - **w/ GPU support**

Upstream activity (WIP / early)

- K8s/issue-68357: “Bring Batch capability to K8s”
- Kube-batch (v0.4), ...
- K8s-default scheduling framework, ...
- Slurm, LSF K8s operators, ...


Bring Batch Capability into Kubernetes #68357

Open k82cn opened this issue on Sep 6, 2018 · 14 comments

k82cn commented on Sep 6, 2018 • edited Member

During the implementation of kube-batch and the discussion of Coscheduling, there're several topics/requirements to support batch workload. I list them here; some items already has related discussion, e.g. Topology, some items are not. I'd like to open related issues (by specific use case) one by one for discussion and review, e.g. Coscheduling. If anyone want to contribute (based on the cases) or have other case about batch workload, please feel free to comments :)

Batch: Queues, Quotas and Fair share

K8s default scheduler:

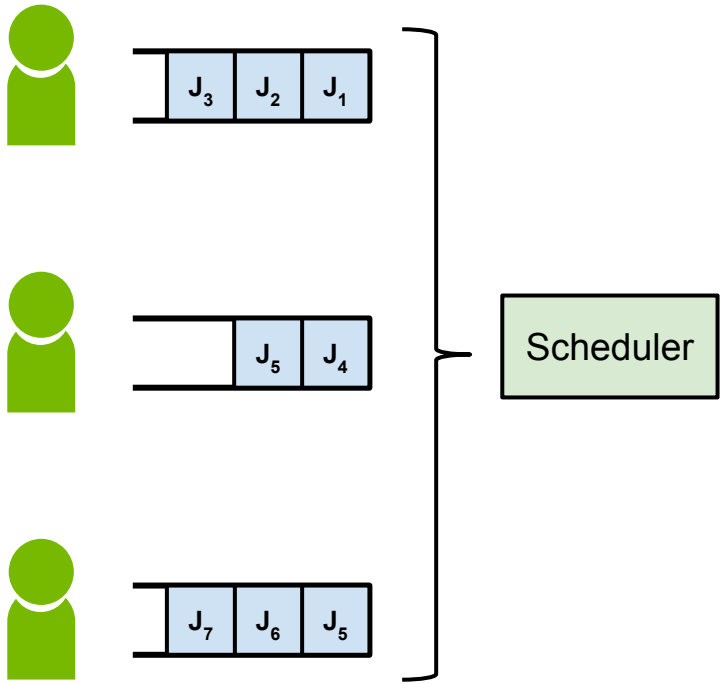
- Single queue, FIFO priority w/o blocking
- Quota per namespace & at pod admission time

Custom scheduler + controller supports:

- Queue per user (or team)
- Quota per user (or team) at scheduling time
- DRF fairness, FIFO w/o blocking

Target:

- Hierarchical quotas, Dynamic fair share



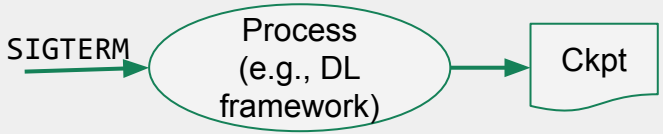
Batch: Handling long running jobs

Timeslices, checkpointing, preemption

Challenges of long running jobs

- Failures / lost work / yield
- Rolling updates take time
- Enforcing fair share

Solution: Checkpointing



Challenges: Long cmdline jobs, Entrypoints

Smaller time slices

Job specifies:

Total run time: e.g, 7 days
Min time slice: e.g., 1 hours

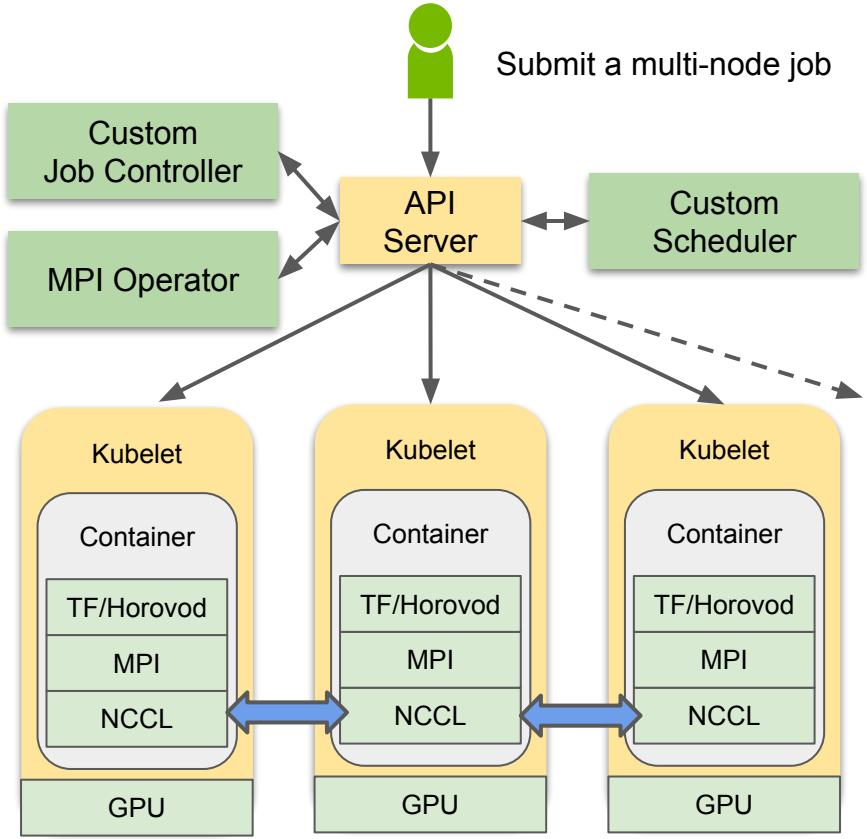
Preemption

Preempt (a) lower priority
(b) over time slice
(c) over fair usage

To help Starving jobs, Fairness, De-frag

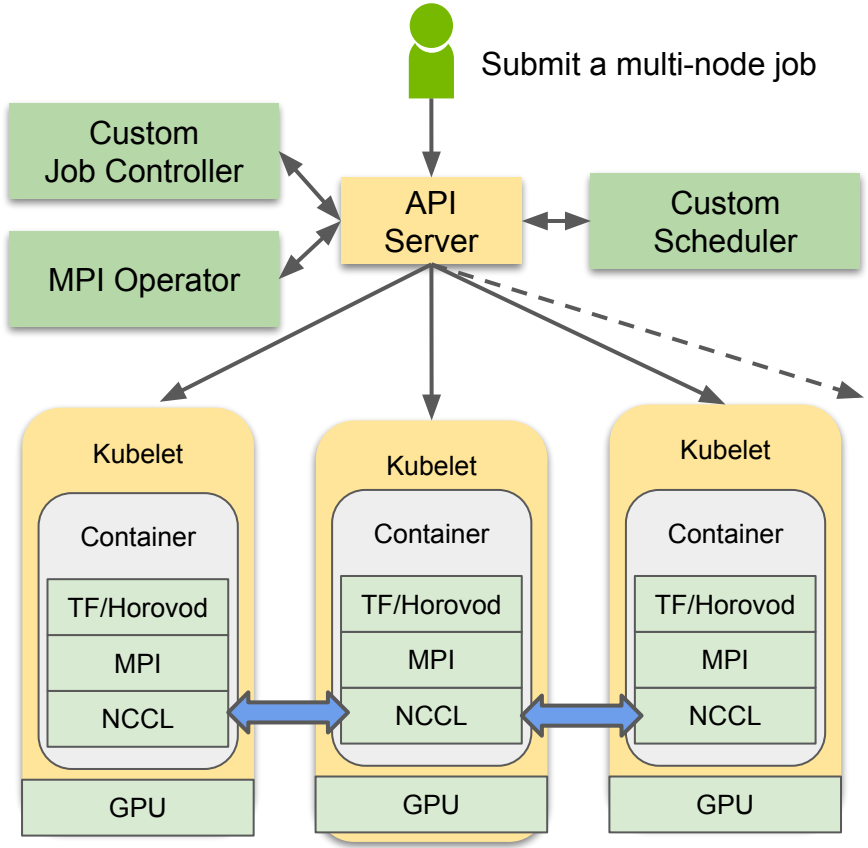
Multi-node Jobs with MPI

- Strong demand for multi-node GPU jobs
  - E.g., 128-GPUs per job
- Two approaches:
  - Async SGD: Parameter-server
  - **Sync SGD: MPI, NCCL, Horovod**
- **Upstreamed**: MPI Operator for Kubeflow



Multi-node Jobs with MPI in K8s

- **Multi-node networking (WIP)**
  - CNI with RDMA / Infiniband / Mellanox
- **Gang/Co-scheduling with MPI operator**
  - Quotas, queues, time limits
  - Full-node, reservation, backfilling
  - Dynamic priority
- **Status**: WIP



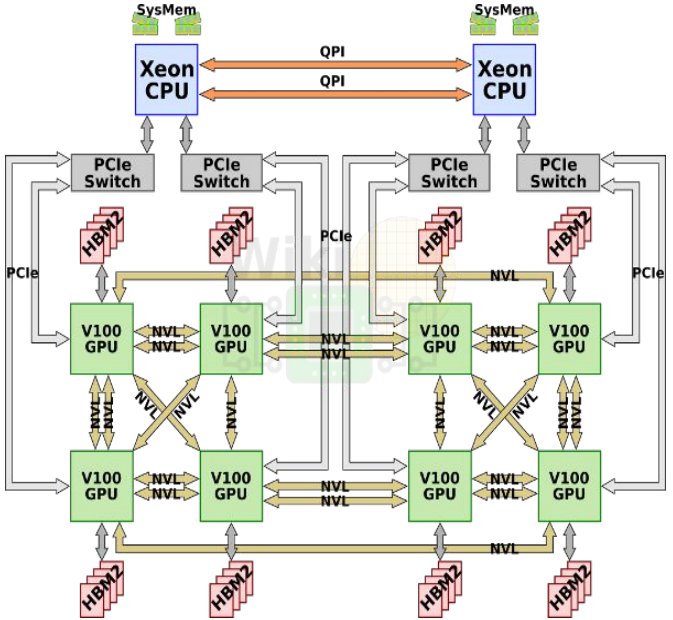
Challenge-2:

Efficient GPU Management

Telemetry, health checks, affinity, upgrades

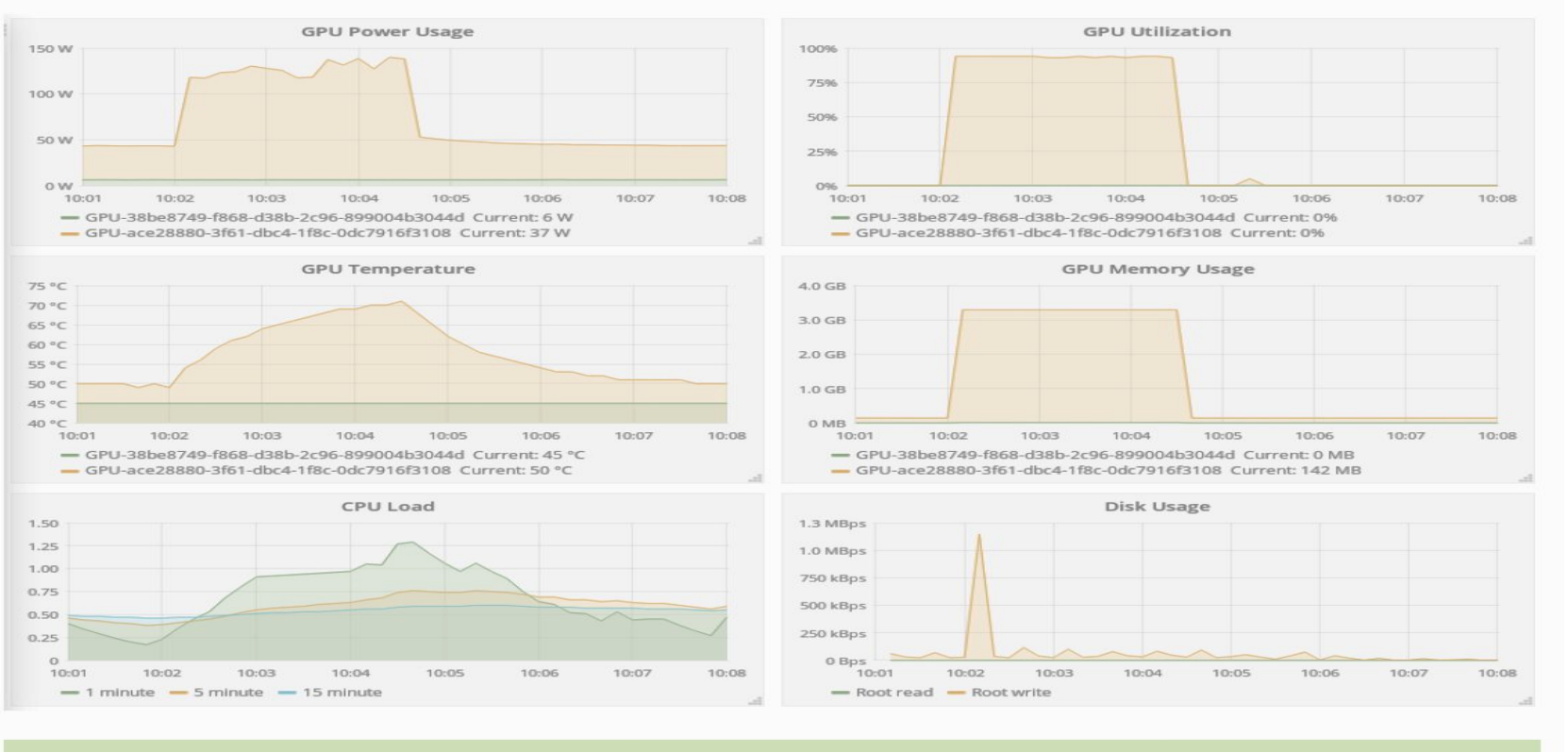
GPU Telemetry and Efficiency

- Utilization, visibility (NVML, nvidia-smi..)
- Data Center GPU Manager (DCGM)
  - Health monitoring, diagnostics
  - Metrics: GPU, memory, power util, temp, NVLink/PCIe, ..
- **Upstreamed**: DCGM exporter for Prometheus
- Per-pod metrics via “`KubeletPodResources`”
- https://github.com/NVIDIA/gpu-monitoring-tools

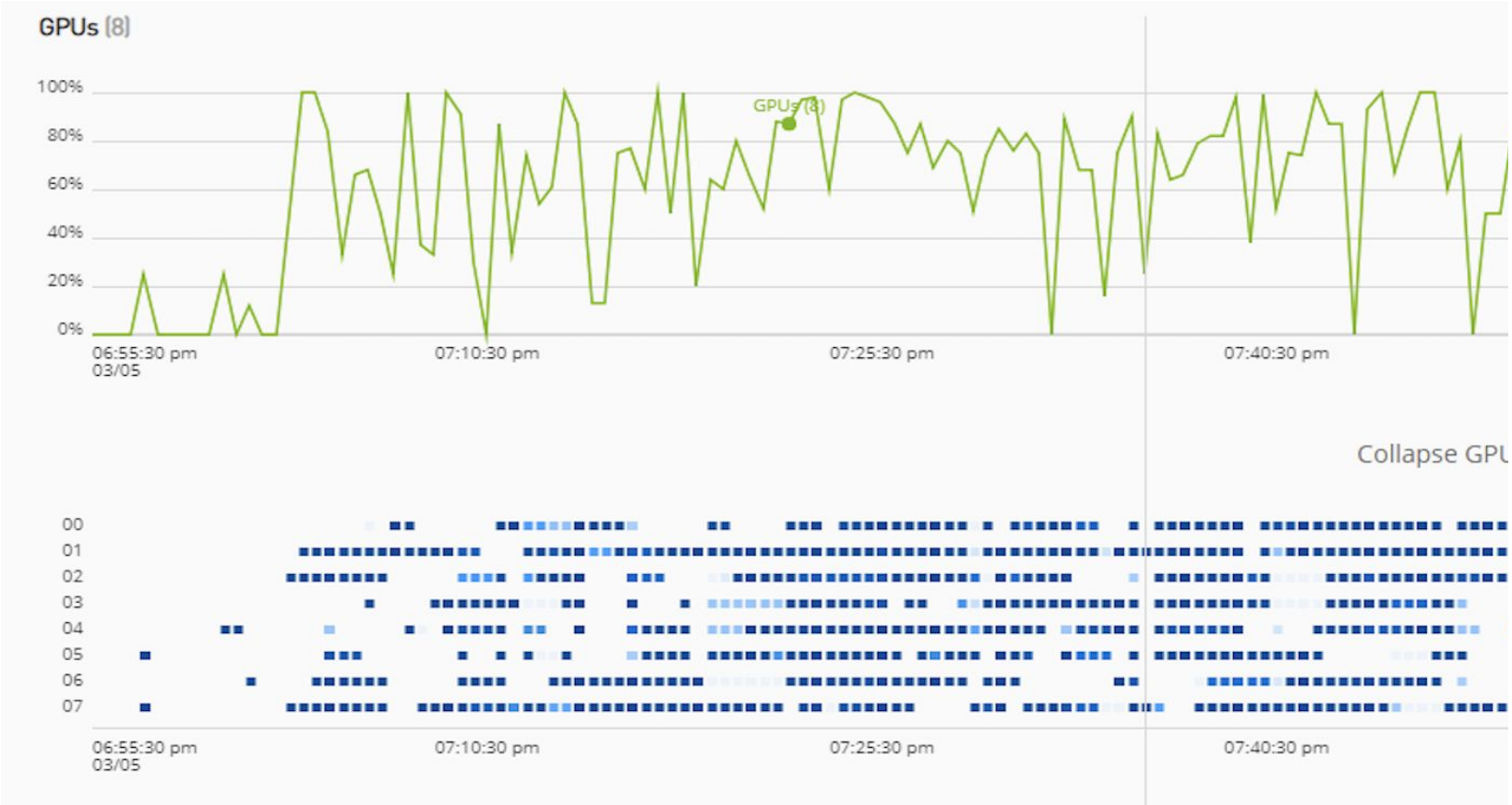


DGX-1 node with 8 Volta GPUs

Telemetry data in Grafana

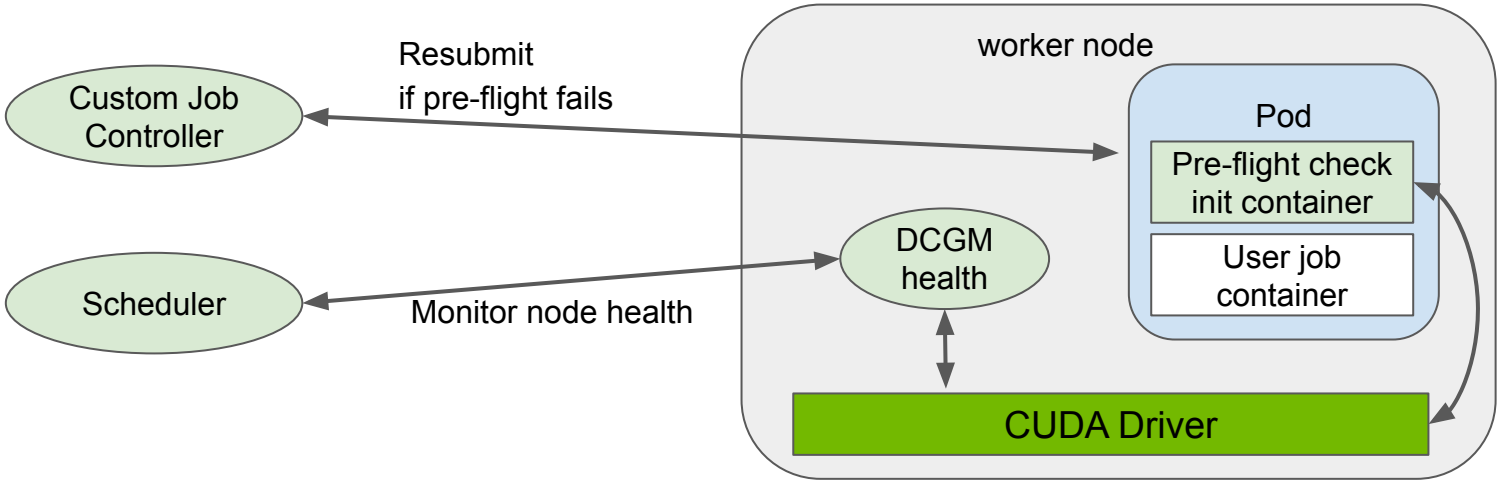


Telemetry: Utilization of a 8-GPU job



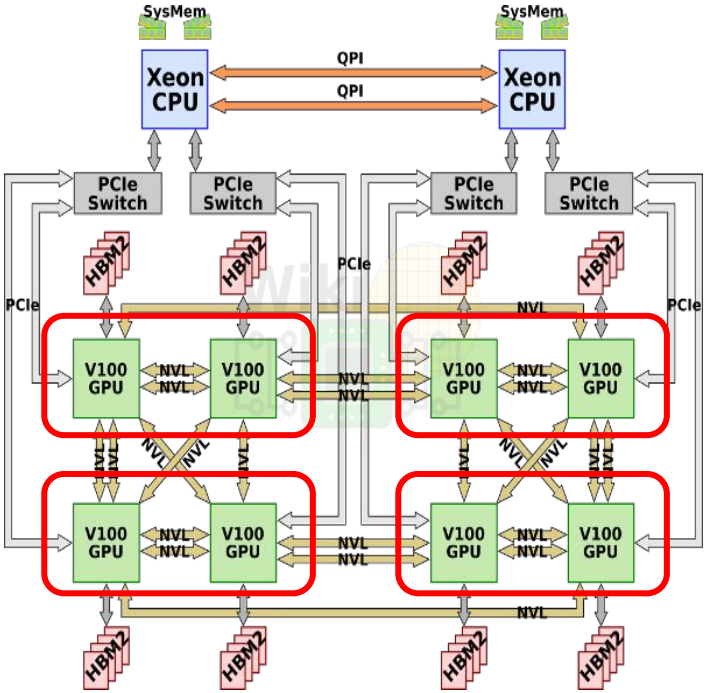
GPU Health and Pre-flight checks

- **DCGM: Periodic non-invasive GPU health checks**
  - Scheduler avoids scheduling jobs on unhealthy nodes
- **Custom: Pre-flight checks (GPU + end-end) to improve Yield**
  - Job re-submitted automatically if pre-flight check fails



GPU Affinity

- **Socket affinity and CPU-GPU allocation**
- Reserving CPUsets for system jobs
- Kubelet / `TopologyManager` modifications
- Custom scheduler enhancements
  - Optimal GPU selection w/o K8s core API mod
- **Status**: Testing / upstream discussion in progress

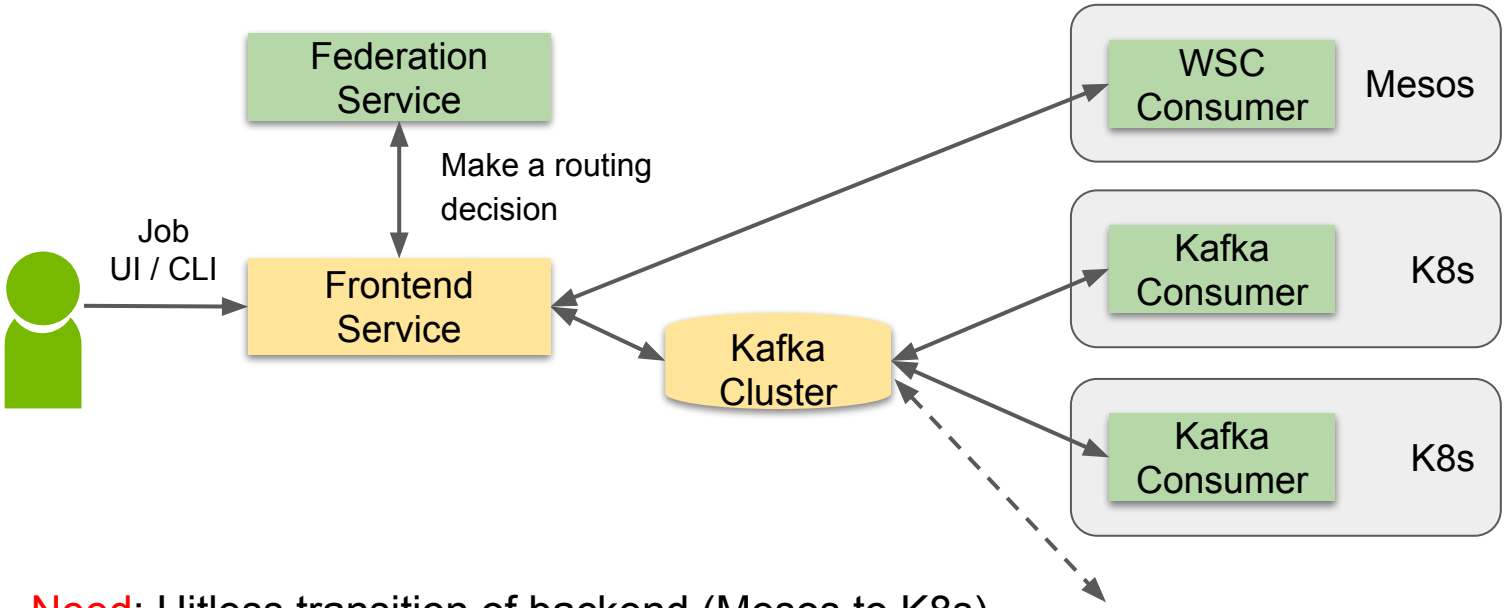


DGX-1 node: 40% perf difference (closeby vs across-socket)

Prod Challenges:

Hitless transition, yield, upgrades

Hitless Transition and Batch Federation

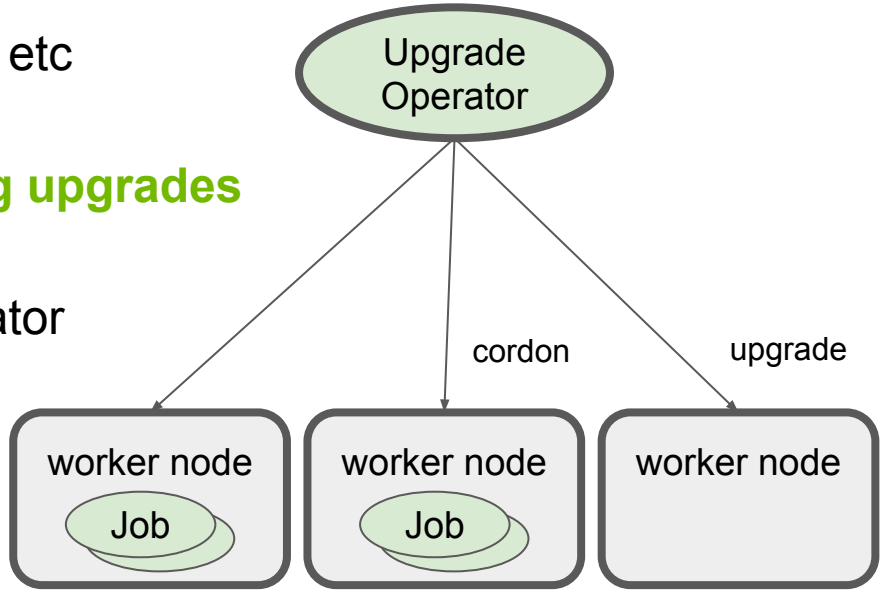


- Need: Hitless transition of backend (Mesos to K8s)
- **Approach: Batch Federation and Job UI/CLI abstraction**
  - Route jobs to cluster based on user, team, job type etc
  - Similarity to Kubefed / multi-cluster but for batch workloads
- **Possibly later**: Queues and quotas at Federation layer

Hitless Rolling Upgrade Operator

Upgrade worker nodes without affecting long running jobs

- GPU drivers, device plugins, Kubelet etc
- **Custom operator to manage rolling upgrades**
- Related: Driver container, GPU operator



Misc: Long Standing K8s Issues

- Node-level user namespace remapping
  - Root inside to non-root outside (kubernetes/KEP/127, since 2016)
- Enforcing size limit on /dev/shm or emptyDir
  - Hinders some Pytorch jobs sharing a node (kubernetes/KEP/63126)
- Bare-metal and private VM service environments
  - Kops, kubespray modifications
- Scalability aspects with Kubelet missing some API server updates
  - Adjusted `MAX_CONCURRENT_STREAMS` and the pod GC Threshold

Summary

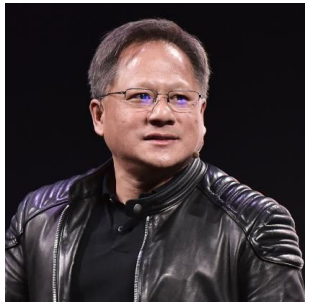
Summary

- **NGC registry**: GPU-accelerated container hub
  - https://ngc.nvidia.com
- **Internal Prod GPU clusters using K8s**
  - Successful transition, Yield improvements
- HPC/batch challenges and enhancements:
  - Time limits/slices, queues, fair share, checkpointing, preemption, multi-node
- GPU management: Telemetry, Health check, GPU affinity, Upgrades etc

Ongoing Work

- Spark, ML/ETL, Inference with GPUs
- HPC/Batch in upstream K8s, Multi-node, GPU affinity, Sharing
- Continue contribution / collaboration with community

⇒ **More GPU empowered workloads in K8s!**



Buy More GPUs,
Save Time & Money