IIT Kharagpur | IIT Madras | IIT Goa | IIT Palakkad

APPLIED ACCELERATED ARTIFICIAL INTELLIGENCE

Scheduling and Resource Management

Introduction to schedulers and orchestration tools

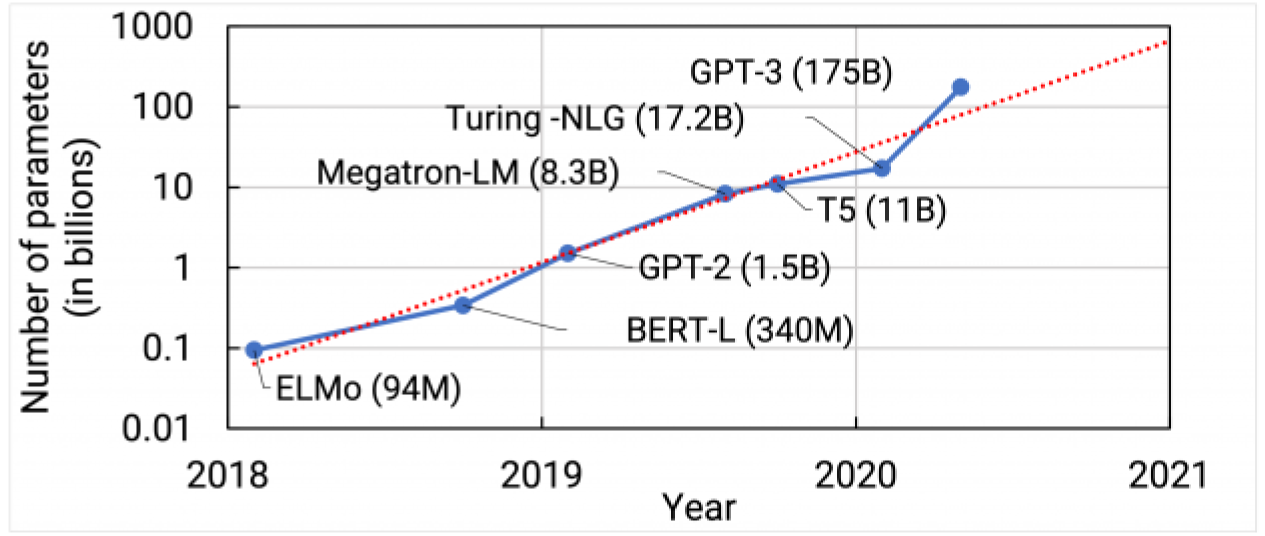
Bharatkumar Sharma
GPU Advocate, NVIDIA

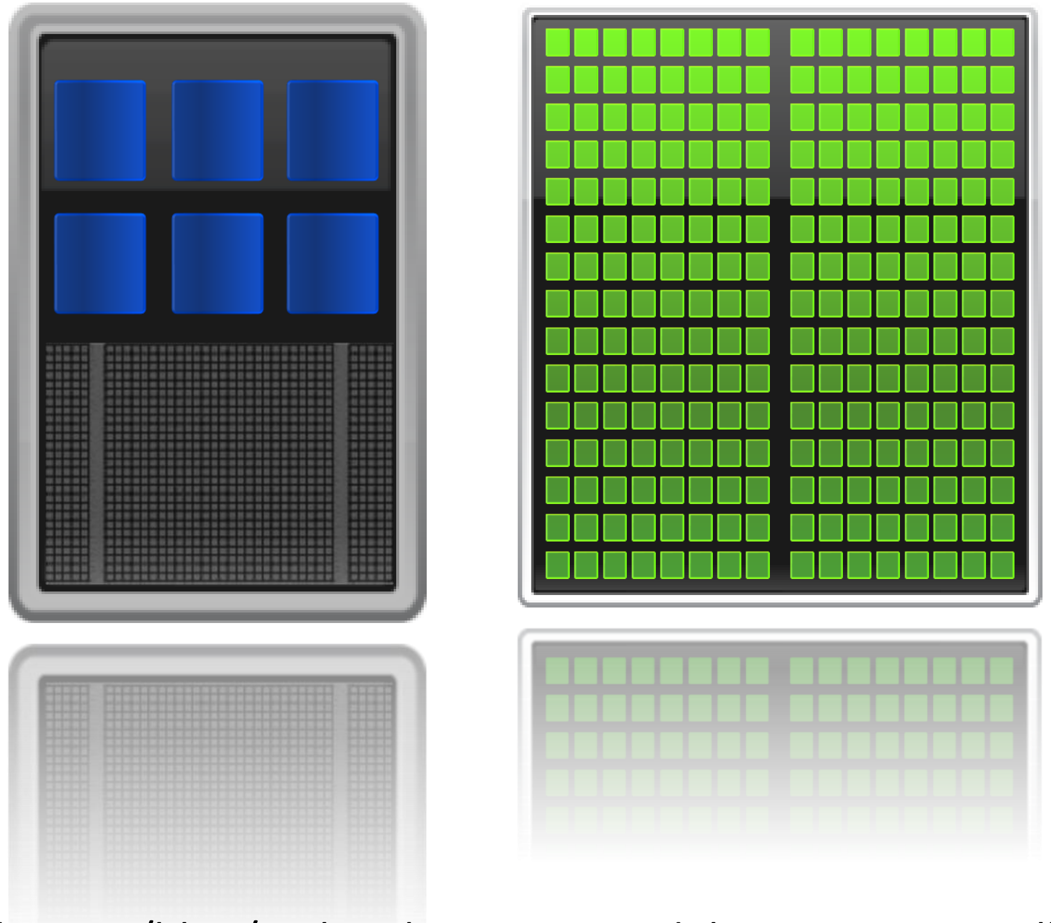
National Supercomputing Mission

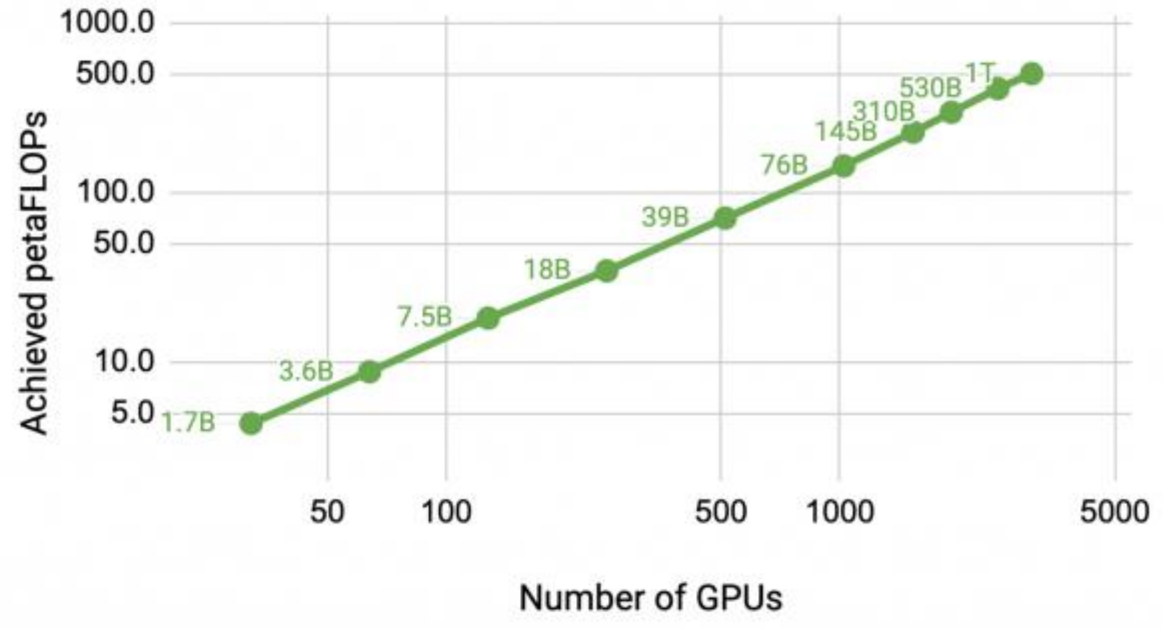
Centre for Development of Advanced Computing

What have we covered so far?

- Need for Speed
  - Accelerated Computing
  - Largest AI model ?
  - What does it require to train these models?



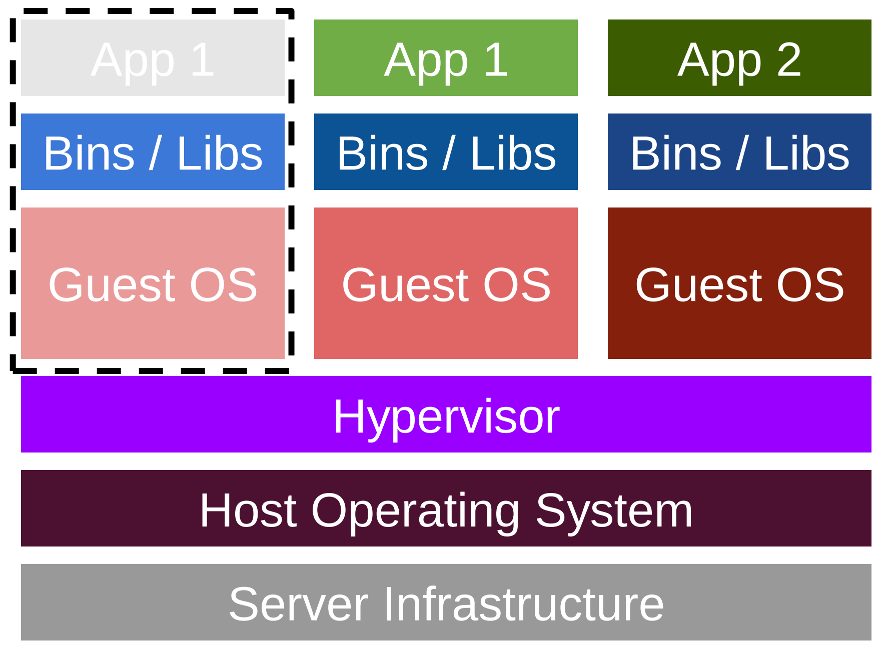




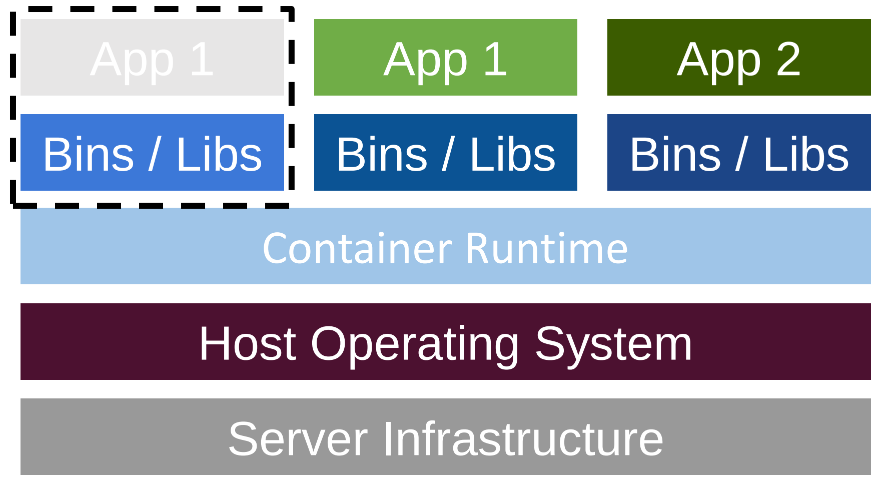
https://developer.nvidia.com/blog/scaling-language-model-training-to-a-trillion-parameters-using-megatron/

What have we covered so far?

- Need for Virtualization and Containers



VIRTUAL MACHINES



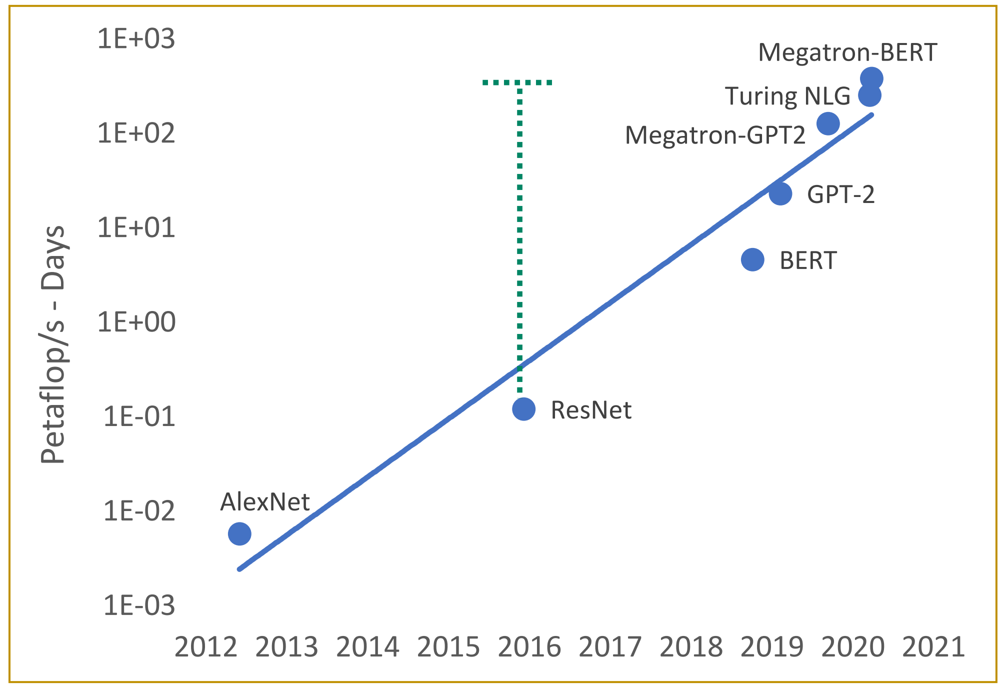
CONTAINERS

Prerequisites

- This session requires the participants to be familiar with concept of computer fundamental, network, containers, basic AI pipeline
- Please follow slack #annoucmenet channel to go through some of these definitions
- The participants will be exposed to concepts related to how the AI infrastructure is setup in real world scenario and how to access the same.

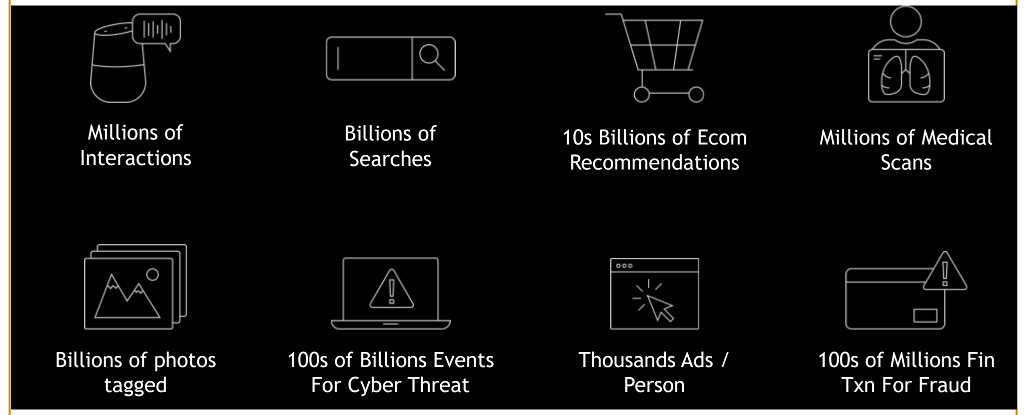
Challenges: Accelerating Big and Small

AI Advances Demand Exponentially Higher Compute



3000X Higher Compute Required to Train Largest Models Since Volta

AI Applications Demand Distributed Pervasive Acceleration

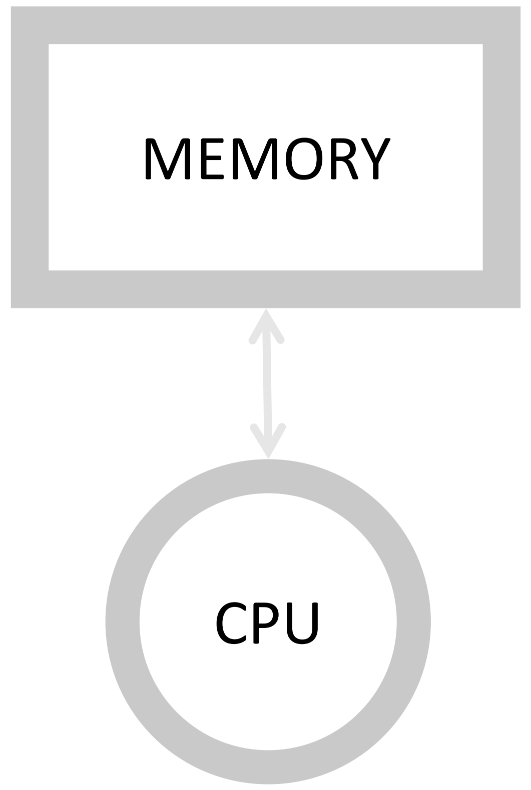


Millions of Interactions
Billions of Searches
10s Billions of Ecom Recommendations
Millions of Medical Scans
Billions of photos tagged
100s of Billions Events For Cyber Threat
Thousands Ads / Person
100s of Millions Fin Txn For Fraud

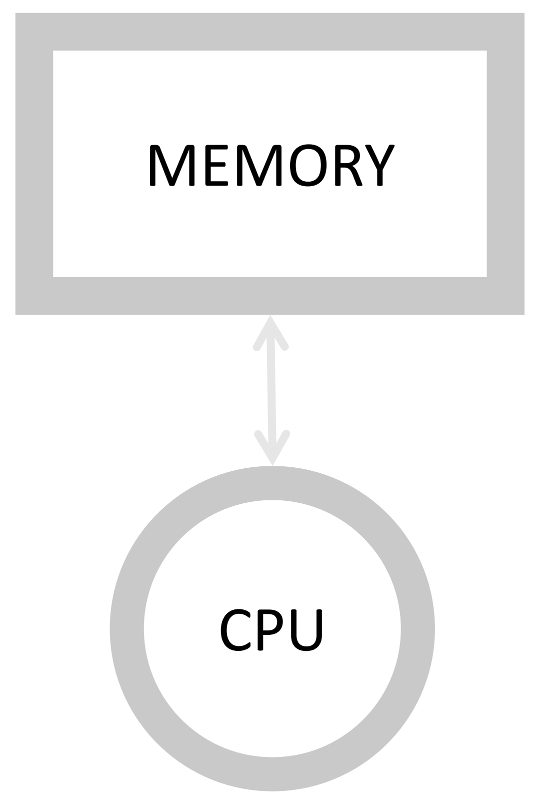
Every AI Powered Interaction Needs Varying Amount of Compute

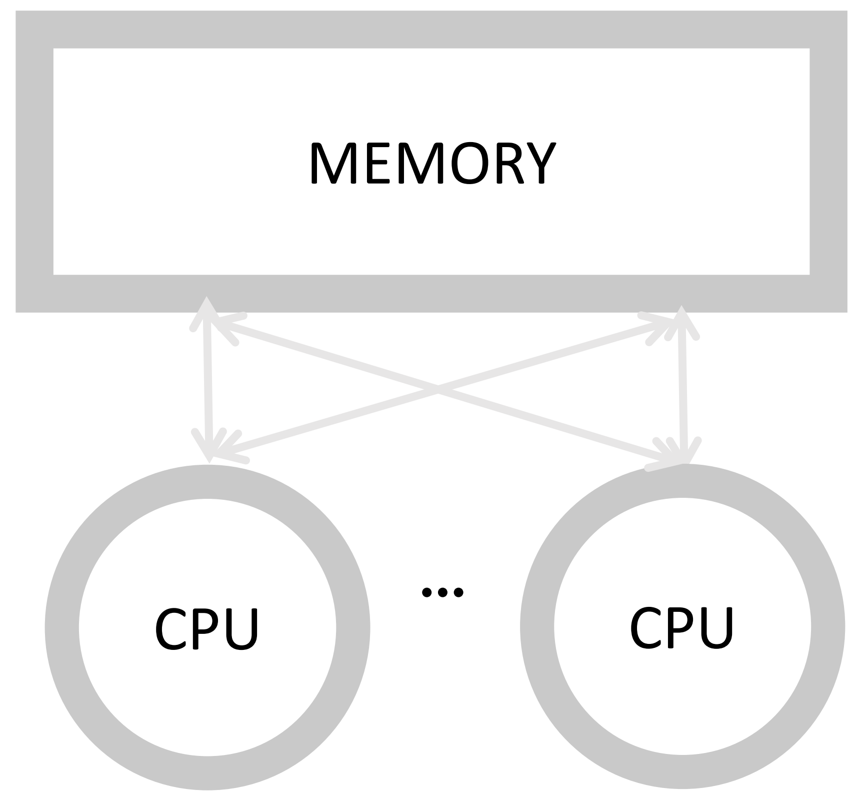
Source: OpenAI, NVIDIA

HPC System evolution

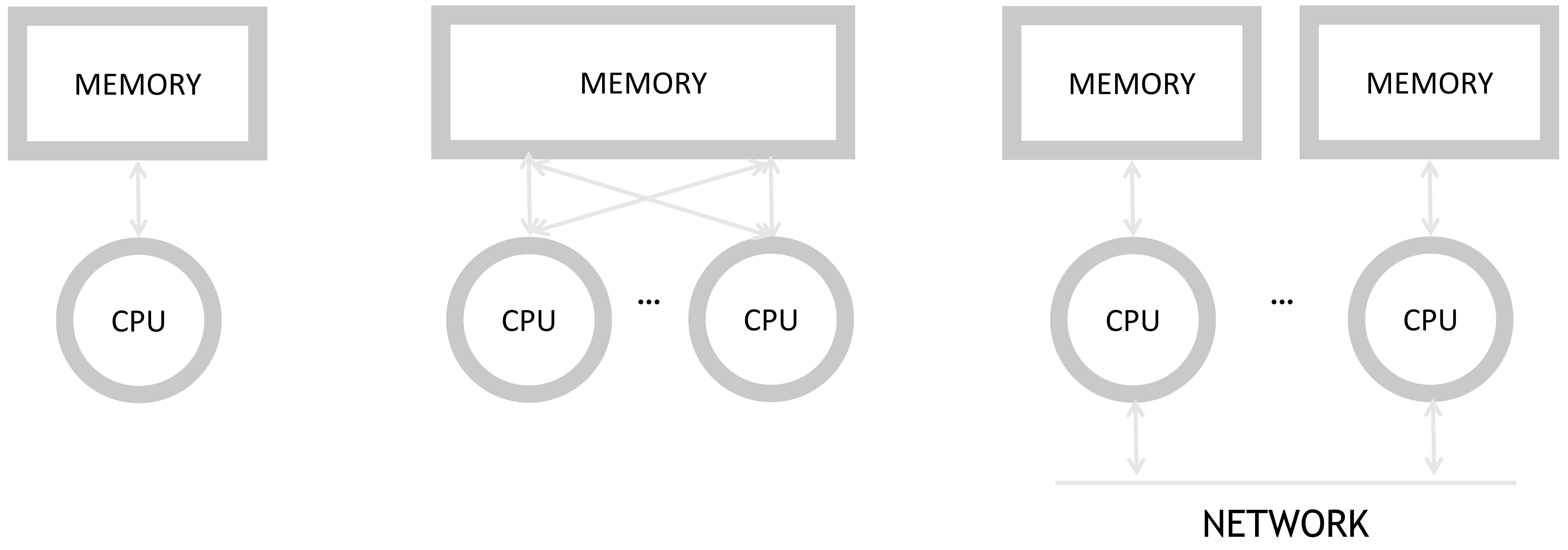


HPC System evolution

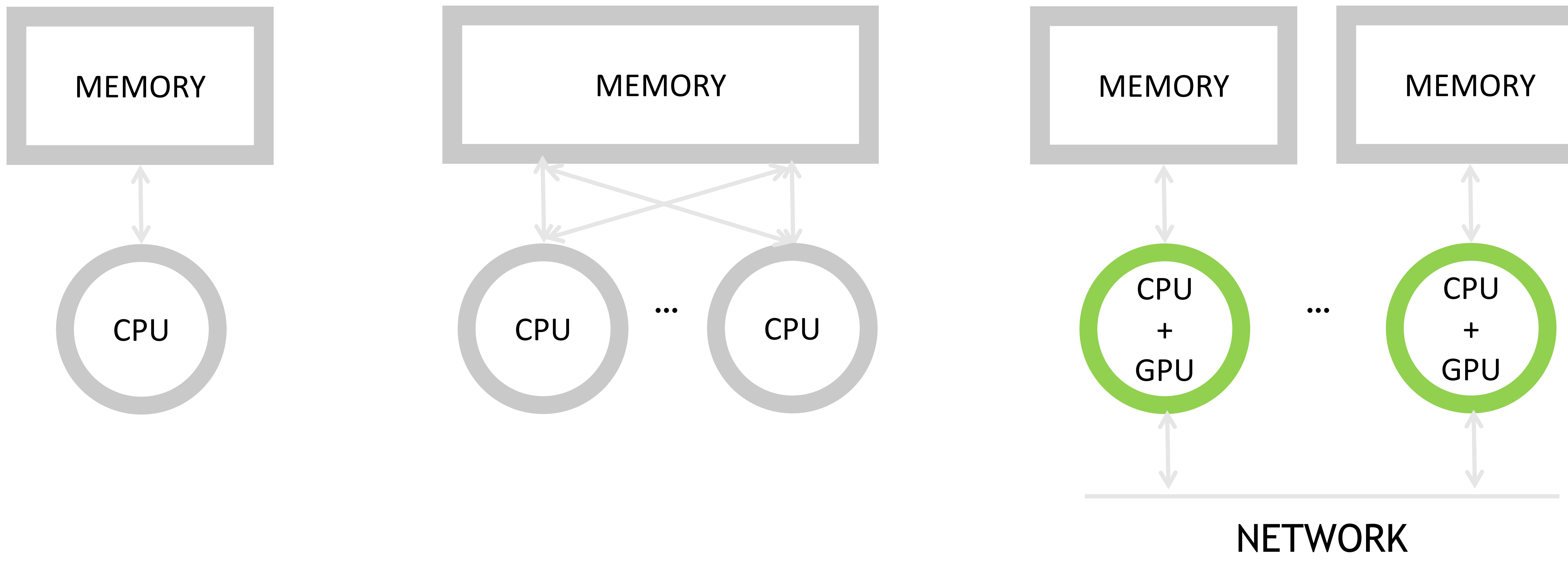




HPC System evolution



HPC System evolution



What is a Cluster?

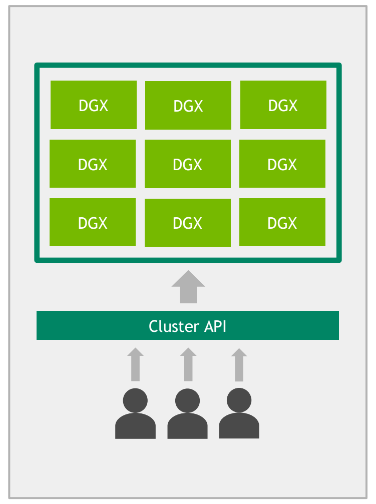
A computer cluster is a set of computers that

work together

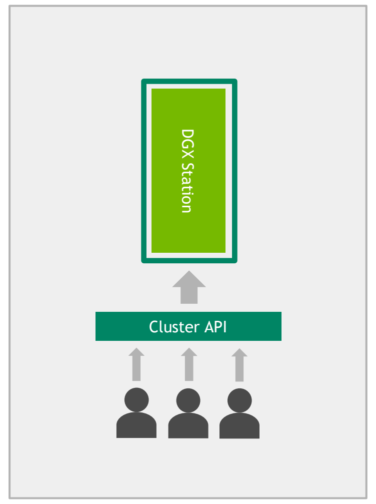
so that they can be viewed as a single system,

controlled and scheduled by software

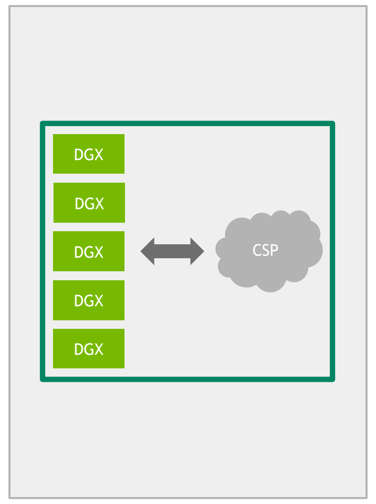
AI : Use Cases



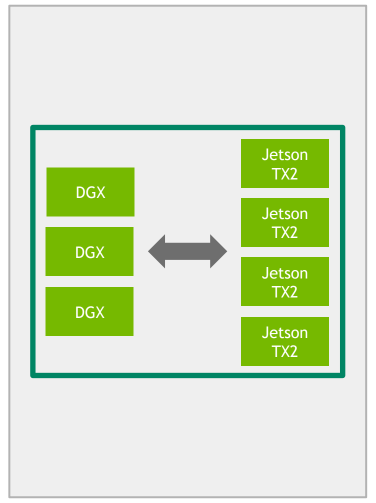
Many users, many nodes
On-prem



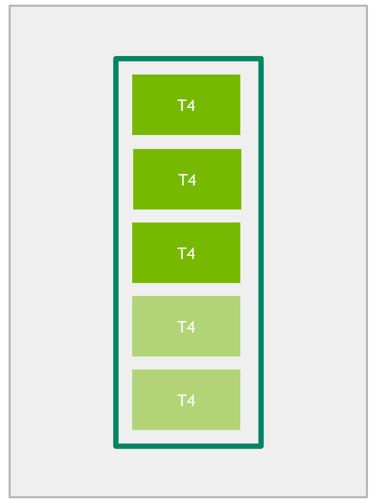
Many users, single node
On-prem



Cloud bursting
Hybrid

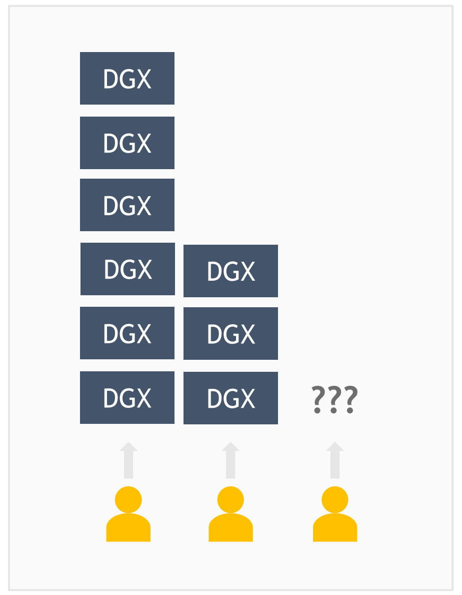


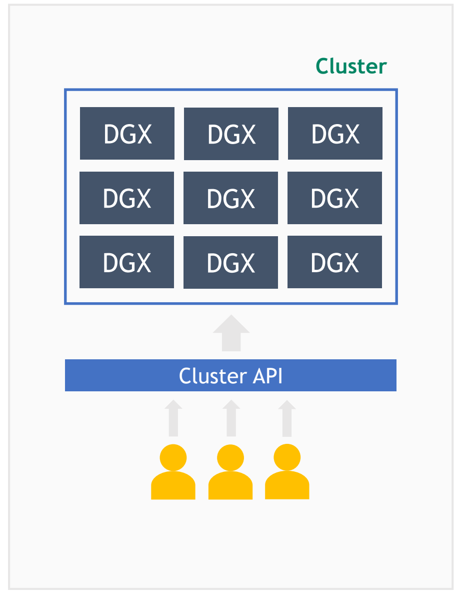
Edge/IoT
Multi-region



Production Inferencing
*

Challenge: Resource Utilization

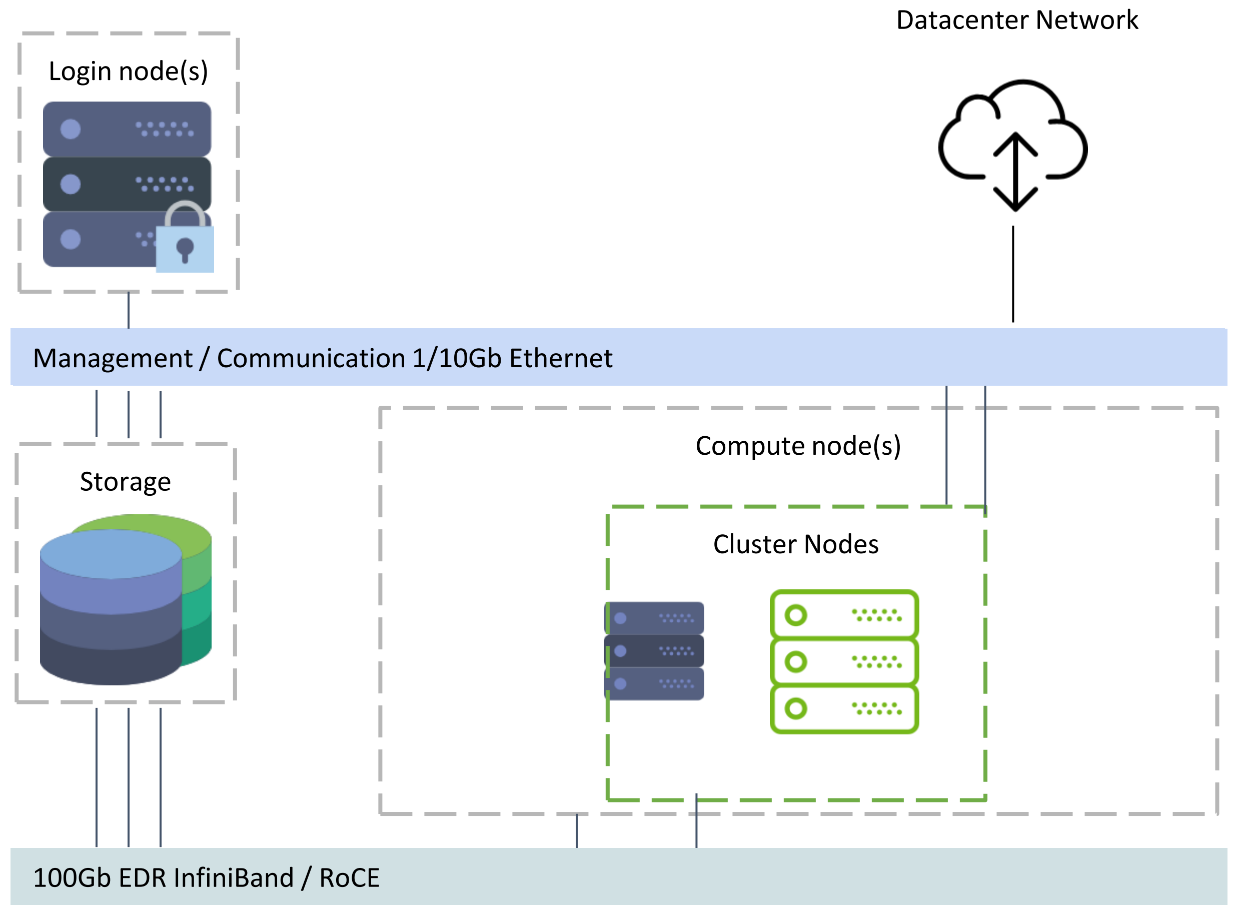




Dedicated Nodes
- User coordination required
- Resources can go unused
- Not scalable
- Difficult to maintain

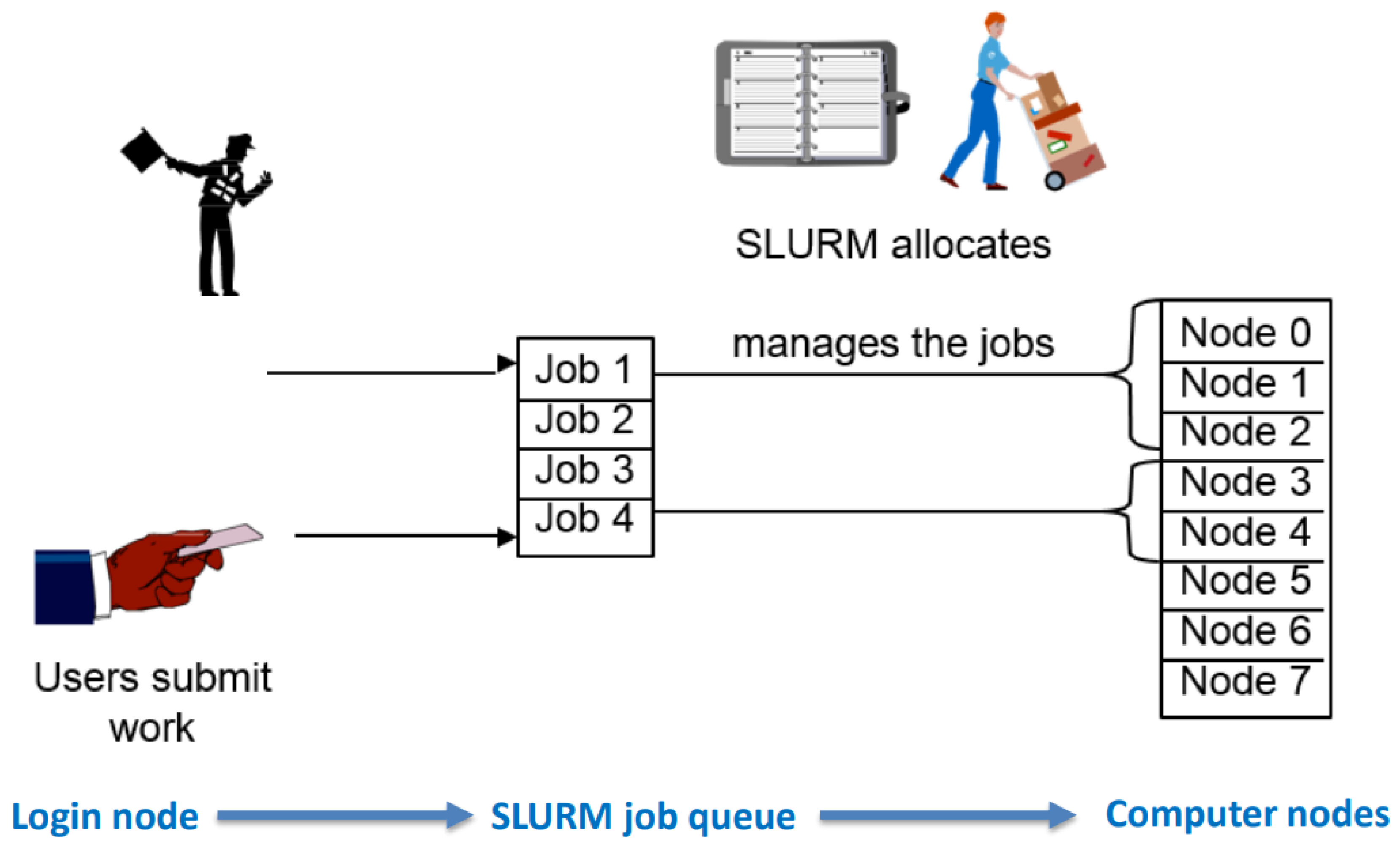
Cluster
- Less user coordination required
- Efficient use of resources
- Highly scalable
- Standardized maintenance

AI Cluster Component


Datacenter Network
Login node(s)
Management / Communication 1/10Gb Ethernet
Storage
Compute node(s)
Cluster Nodes
100Gb EDR InfiniBand / RoCE

http://www.hpc.iitkgp.ac.in/HPCF/systemConfig

Scheduler



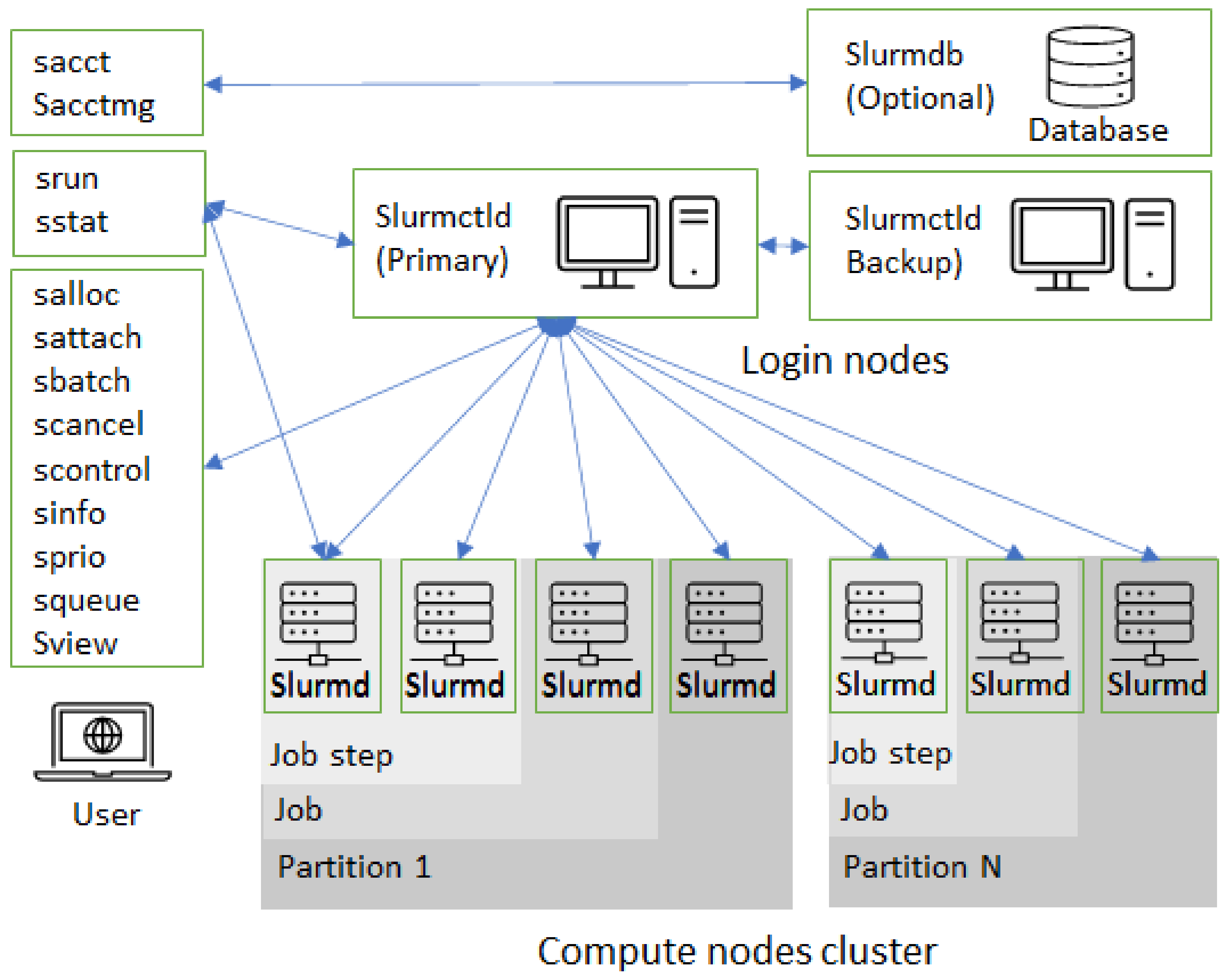
- Simple Linux Utility for Resource Management
  - Started as a simple resource manger for Linux clusters, about 500,000 lines of C code
  - Easy to use (e.g.: run a.out on a PC, run sbatch a.out on a cluster)
  - Fair-share resource allocations
- The “glue” for a parallel computer to execute parallel jobs
  - Make a parallel computer as almost easy to use as a PC

Base Slurm Components

- **Controller:**
  - slurmctld: centralized manager to monitor workloads and available cluster resources

- **Compute Nodes:**
  - slurmd: daemon running on compute nodes; responsible for receiving, executing, and returning results for a job

More information: https://slurm.schedmd.com/overview.html



Job Submission

- To run your job, you will need to specify what resources you need.
- These can be memory, cores, nodes, GPUs, etc.
- There is a lot of flexibility in the scheduler to get specifically the resources you need.

Options	Description
--nodes	The number of nodes for the job (computers)
--mem	The amount of memory per node that your job need
-n	The total number of tasks you job requires
--gres gpu:#	The number of GPUs per node you need in you job
--qos	the QOS you want to run in, currently normal or debug
--mem-per-cpu=	The amount of memory per cpu your job requires
-N	The minimum (and maximum) number of nodes required
--ntasks-per-node=#	tasks per node.
--exclusive	this will get you exclusive use of the node

Slurm Common Commands

- For full list of common Slurm command, follow this link.

Command	Description	Detailed
sinfo	Reports the state of the partitions and nodes managed by Slurm	PARTITION: the name of the partition AVAIL: whether the partition is up or down TIMELIMIT: the maximum length a job will run in the format Days-Hours:Minutes:Seconds NODES: the number of nodes of that configuration STATE: down* if jobs cannot be ran, idle if it is are available for jobs, alloc if all the CPUs in the partition are allocated to jobs, or mix if some CPUs on the nodes are allocated and others are idle. NODELIST: specific nodes associated with that partition.
sacct	Lists the jobs that are running or have been run.	
squeue	Lists the state of all jobs being run or scheduled to run. Use squeue -u username to view only the jobs from a specific user	JOBID: number id associated with the job PARTITION: name of partition running the job NAME: name of the job ran with sbatch or sinteractive USER: who ordered the job to be ran ST: State of the job, PD for pending, R for running TIME: how long the job has been running in the format Days-Hours: Minutes:Seconds NODES: number of nodes allocated to the job NODELIST(REASON): either the name of the node running the job of the reason the job is not running such as JobHeldAdmin (job is prevented from running by the administrator).
scancel	Signals or cancels a job. One or more jobs separated by spaces may be specified.	

Demo

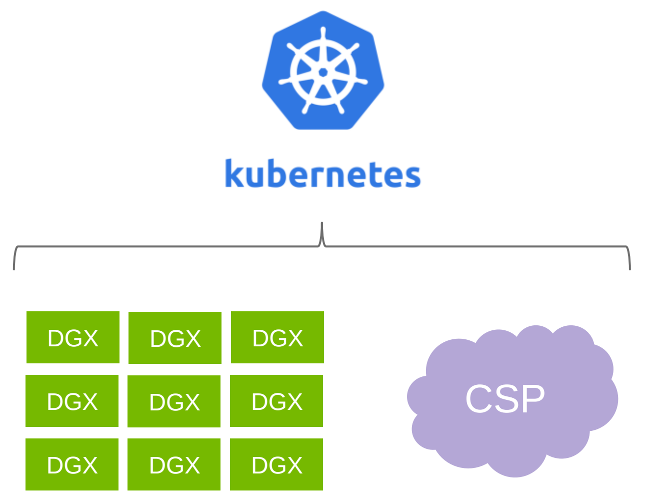
- Check the nodes status
  - `$sinfo`
- Submit an interactive job
  - `$srun --ntasks=5 --nodes=1 --cpus-per-task=2 --partition=batch --time=4:00:00 --gres=gpu:1 --pty /bin/bash`
  - `$nvidia-smi`
  - `$exit`
- Cheque status of the running jobs
  - `$squeue`
- Cancel the submitted job
  - `$scancel <Job Id>`

What Next?

Container orchestration for clusters

- **Resource limit control**. This feature reserves the CPU and memory for a container, which restrains interference among containers and provides information for scheduling decisions;
- **Scheduling**. Determine the policies on how to optimise the placement of containers on specific nodes;
- **Load balancing**. Distribute the load among container instances;
- **Health check**. Verify if a faulty container needs to be destroyed or replaced;
- **Fault tolerance**. Create containers automatically if applications or nodes fail;
- **Auto-scaling**. Add or remove containers automatically.

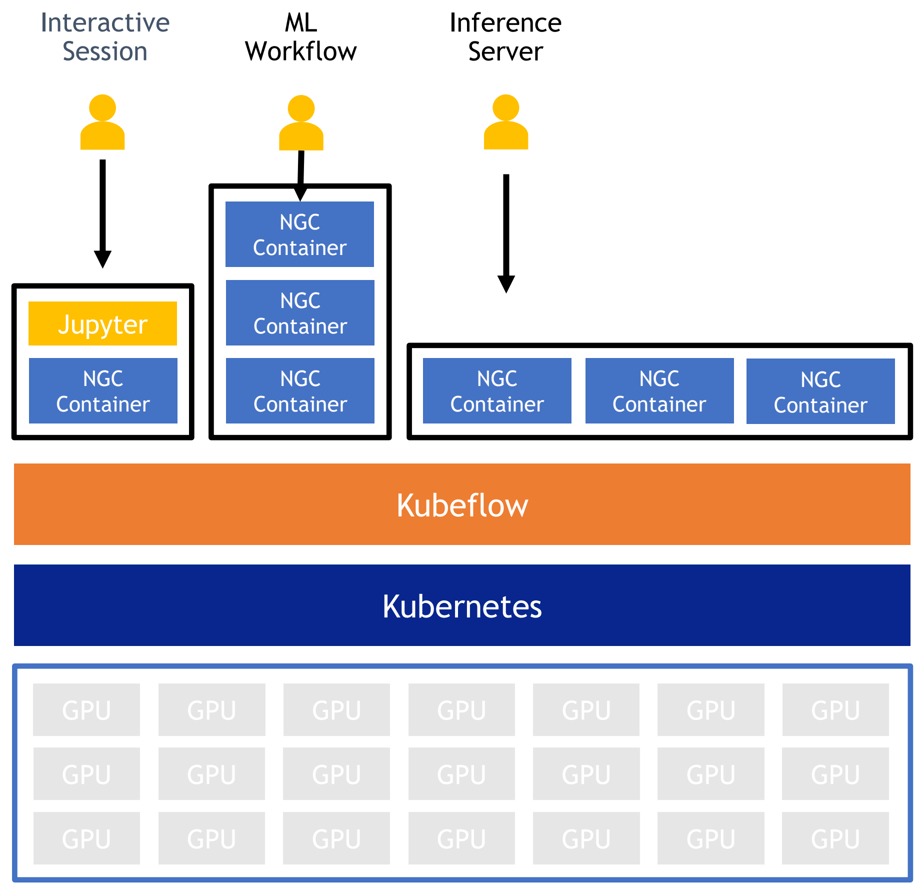
Kubernetes



Think of Kubernetes as an operating system for a cluster

The cluster's servers can be on-prem, in the public cloud, or a mix (hybrid)

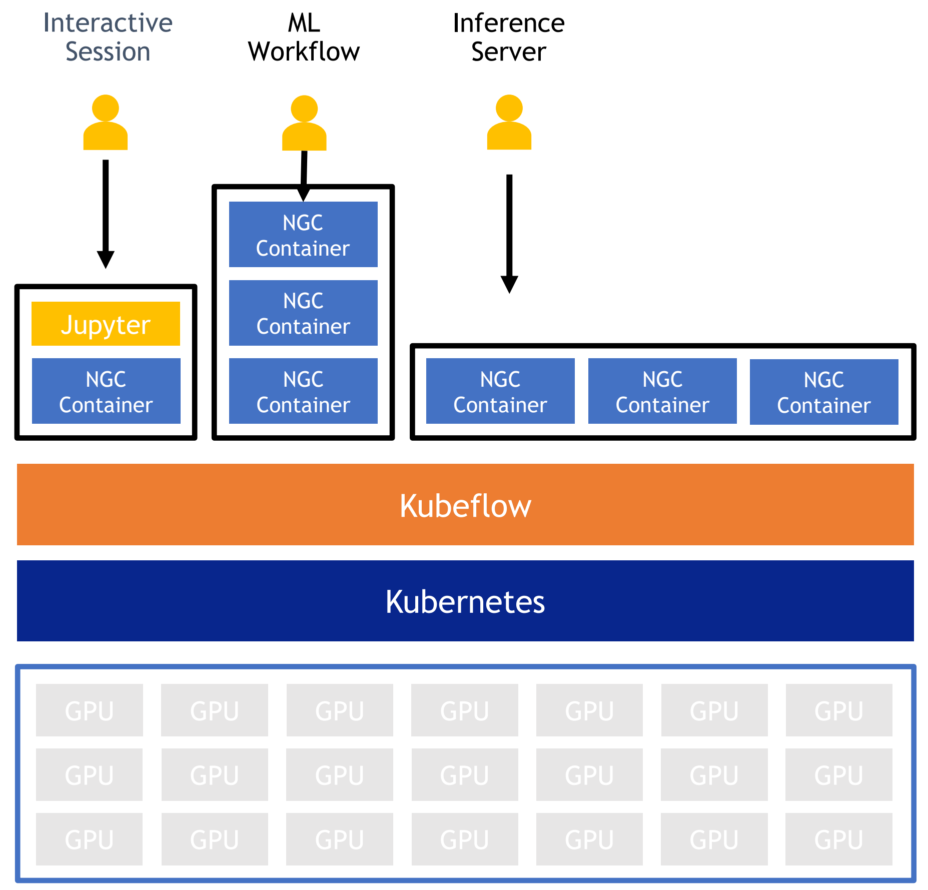
Use Kubernetes to manage pods in the cluster, administer user access, launch jobs as containers, expose running services externally, and more



What is the difference?

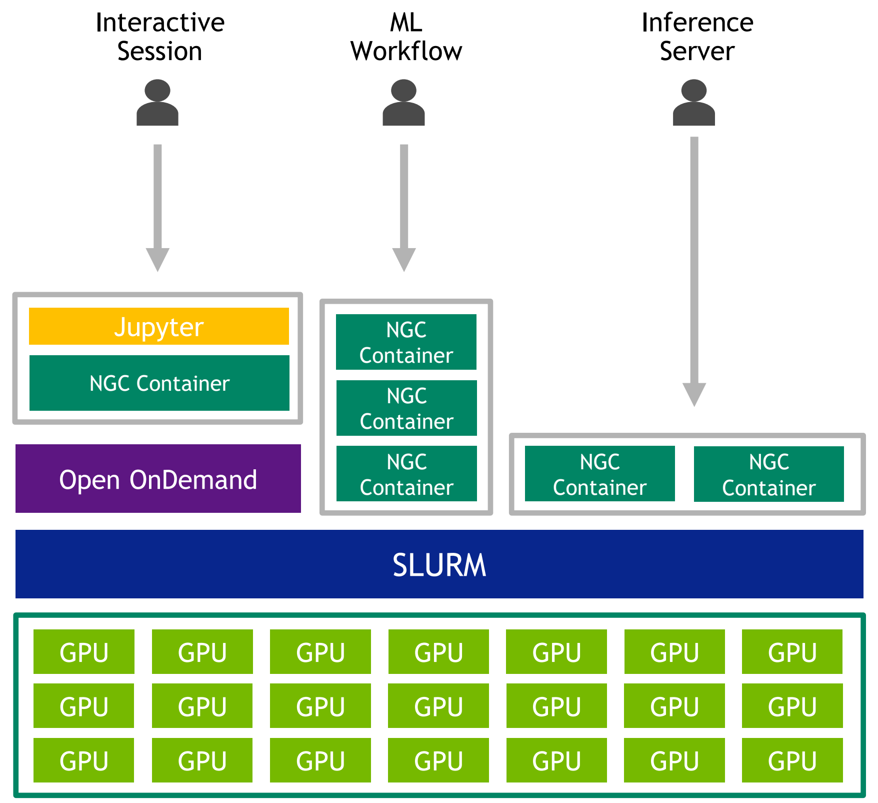
Orchestration

- Designed for micro-services, adapted for AI
- Container based
- Scale up/down
- Requires additional “meta-schedulers” for advanced scheduling features
- Configuring user access/permissions and integrating storage can be complex
- Provides MLOps tools and run complex AI pipelines
- Adopted by enterprise with large community



Job Scheduling

- Designed for HPC
- Bare-metal based, supports containers
- No inferencing capabilities
- Advanced scheduling features built-in (priority queues, pre-emption, …)
- Tied to Linux-based systems with easy integration to security, auth, and storage
- Highly-tuned for performance, leveraging InfiniBand, topology, etc. for multi-node
- Primarily used by researchers for HPC



Thank You