

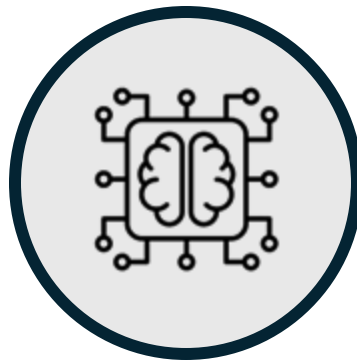
Model Training in Public Clouds: Case for IBM Storage Scale

Vasily Tarasov, Scott Guthridge, Jeremy Cohn,
Marc Eshel, Leo Luan, Travis Janssen, Alex Merenstein,
Frank Schmuck, Lei Pan, Thanh Pham,
Veera Deenadhayalan, Swami Sundararaman,
Seelam Seetharami, Sophia Wen, Talia Gershon
IBM Research - Hybrid Cloud Infrastructure

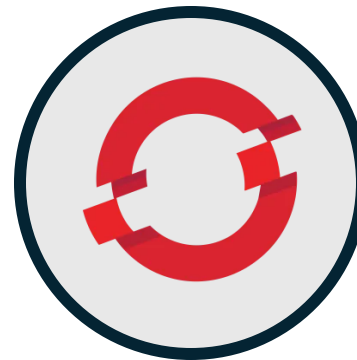
Kevin O'Connor, Abdoulaye Traore,
Chris Laibinis, Brent Wolfe, Carlos Fonseca
IBM Research - Emerging Technology Engineering
Piyush Chowdhary
IBM Cloud – Scale
Brian Reitz, Steve Pritko, Piyush Shivam
IBM Cloud – Block Storage



+



+



+



IBM Storage Scale

Model Training

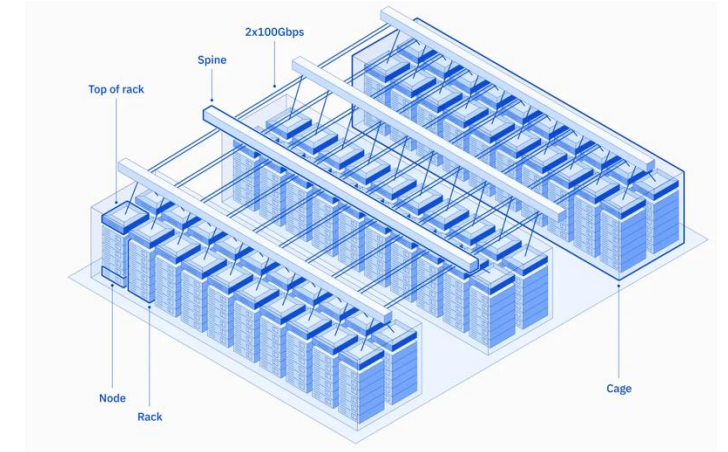
Red Hat Open Shift

IBM Cloud

Disclaimer: Research work

Vela(s): Cloud-native AI Training Cluster(s)

- IBM Research needs infrastructure to train ML models
 - Large Language Models (LLMs) have billions of parameters
- Can we build a **cloud-native** training cluster?
- Of-the-shelf GPU-rich host servers added to IBM Cloud
 - 200 nodes, 8× NVIDIA A100 / 60 nodes, NVIDIA H100 80GB
- KVM-based Virtual Machines as building blocks
- Standard Ethernet networking
 - RoCE for GPU-GPU communication
- Red Hat OpenShift (OCP) for resources and training job management
 - MLBatch / Kueue MLBatch queuing and quota management system
<https://github.com/project-codeflare/mlbatch/blob/main/CODEFLARE.md#mlbatch-for-codeflare-users>
- IBM Granite models
 - <https://huggingface.co/ibm-granite>



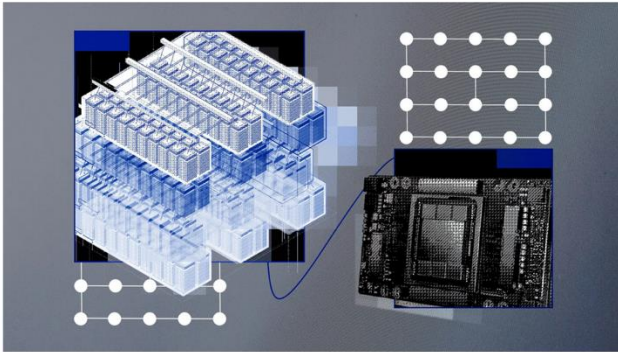
Research Blogs and Papers

Research

7 minute read

Why we built an AI supercomputer in the cloud

Introducing Vela, IBM's first AI-optimized, cloud-native supercomputer.



Vela: A Virtualized LLM Training System With GPU Direct RoCE

Apoorve Mohan* IBM Research Yorktown Heights, USA	Robert Walkup IBM Research Yorktown Heights, USA	Bengi Karacali IBM Research Yorktown Heights, USA
Ming-hung Chen IBM Research Yorktown Heights, USA	Abdullah Kayi IBM Research Bethesda, USA	Liran Schour IBM Research Haifa, Israel
Shweta Salaria IBM Research Yorktown Heights, USA	Sophia Wen IBM Research Yorktown Heights, USA	I-hsin Chung IBM Research Yorktown Heights, USA
Abdul Alim IBM Research Yorktown Heights, USA	Constantinos Evangelinos IBM Research Cambridge, USA	Lixiang Luo IBM Research Yorktown Heights, USA
Marc Dombrowa IBM Research Yorktown Heights, USA	Laurent Schares IBM Research Yorktown Heights, USA	Ali Sydney IBM Research Cambridge, USA
Pavlos Maniotis IBM Research Yorktown Heights, USA	Sandhya Koteswara IBM Research Yorktown Heights, USA	Brent Tang IBM Cloud Rochester, USA
Joel Belog IBM Cloud Lowell, USA	Rei Odaira IBM Cloud Austin, USA	Vasily Tarasov IBM Research Almaden, USA
Eran Gampel IBM Cloud Haifa, Israel	Drew Thorstensen IBM Cloud Durham, USA	Talia Gershon IBM Research Yorktown Heights, USA
	Seetharami Seelam* IBM Research Yorktown Heights, USA	

Abstract

Vela is a cloud-native system designed for LLM training workloads built using off-the-shelf hardware, Linux KVM-based virtualization, and a virtualized RDMA over Converged Ethernet (RoCE) network. Vela virtual machines (VMs) support

*Corresponding Authors: apoorve.mohan@ibm.com, seelam@us.ibm.com



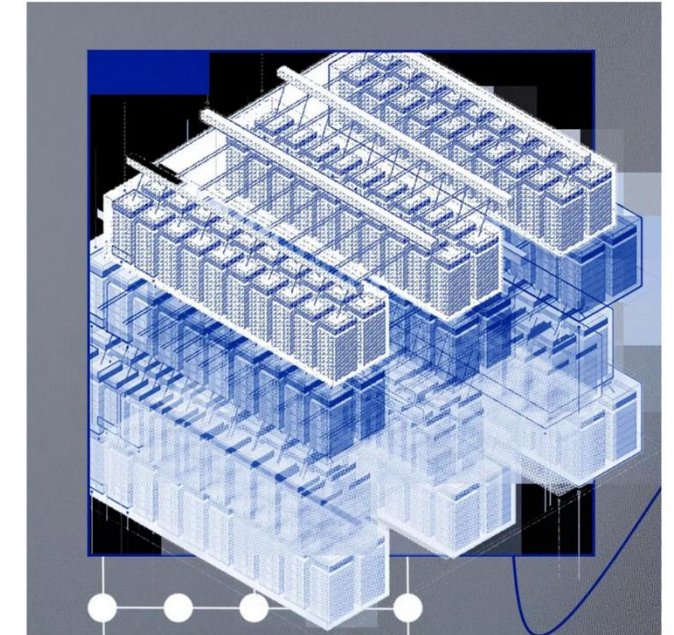
ASPLOS '25, Rotterdam, Netherlands
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1879-7/2025/03

peer-to-peer DMA between the GPUs and SRIOV-based network interface.

In this paper, we share Vela's key architectural aspects with details from an NVIDIA A100 GPU-based deployment in one of the IBM Cloud data centers. Throughout the paper, we share insights and experiences from designing, building, and operating the system over a ~2.5 year timeframe to highlight the capabilities of readily available software and hardware technologies and the improvement opportunities for future AI systems, thereby making AI infrastructure more accessible to a broader community. As we evaluated the system for performance at ~1500 GPU scale, we achieved ~80% of the ideal throughput while training a 50 billion parameter decoder model using model parallelism, and ~70%

IBM uses Storage Scale in its AI model training

By Chris Mellor - August 1, 2024



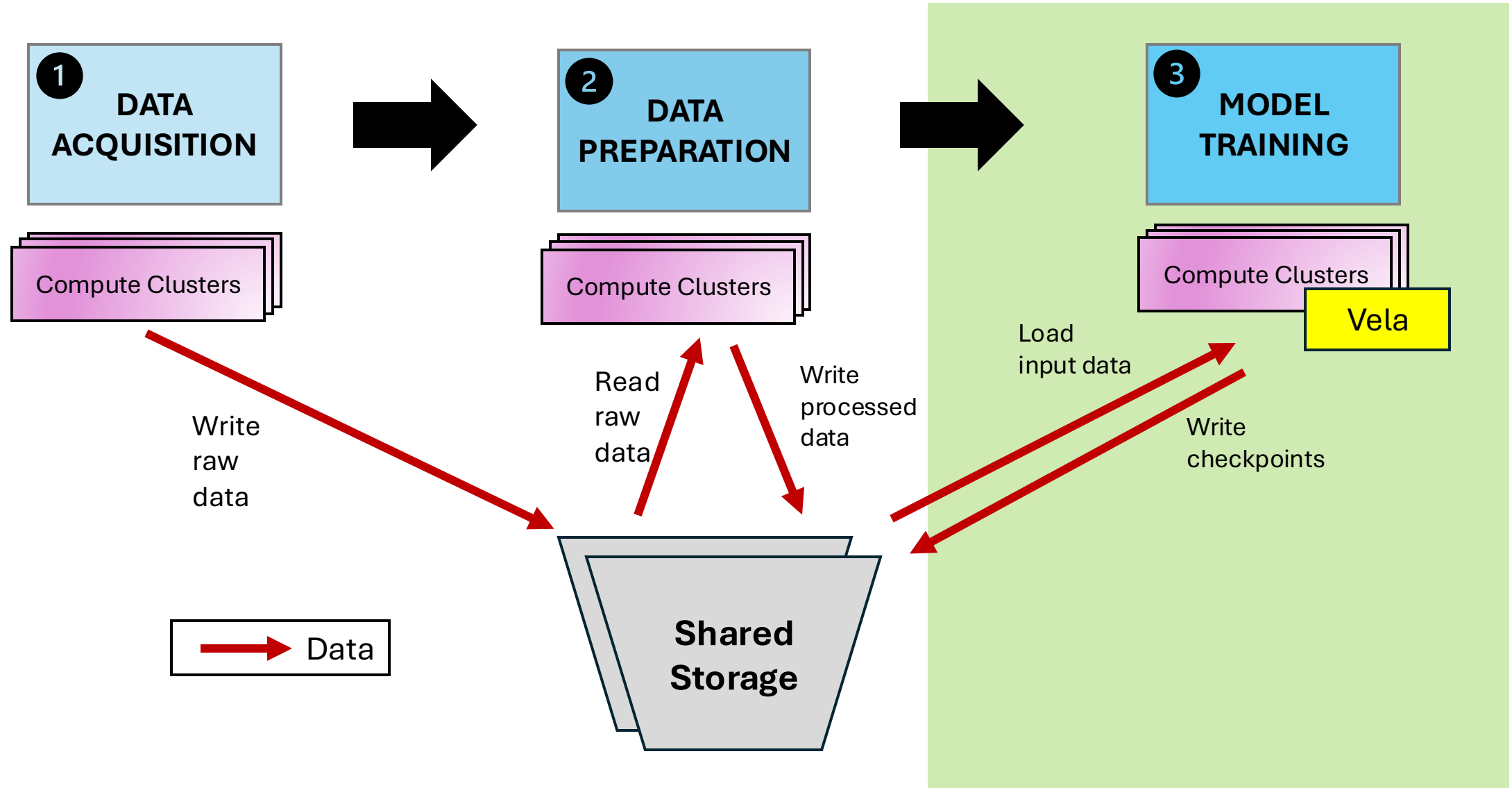
Big Blue developed its own Vela cluster, using [Storage Scale](#), to train its AI models.

<https://blocksandfiles.com/2024/08/01/ibm-uses-storage-scale-in-its-ai-model-training/>

ASPLOS '25: Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems

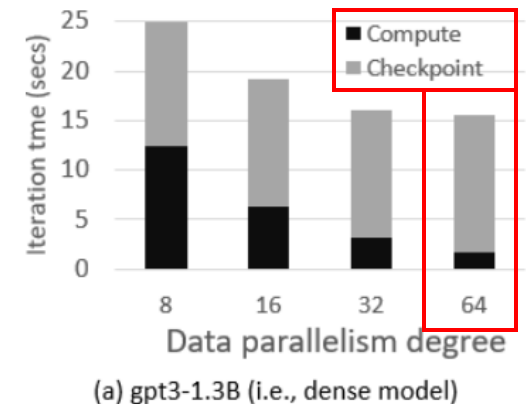
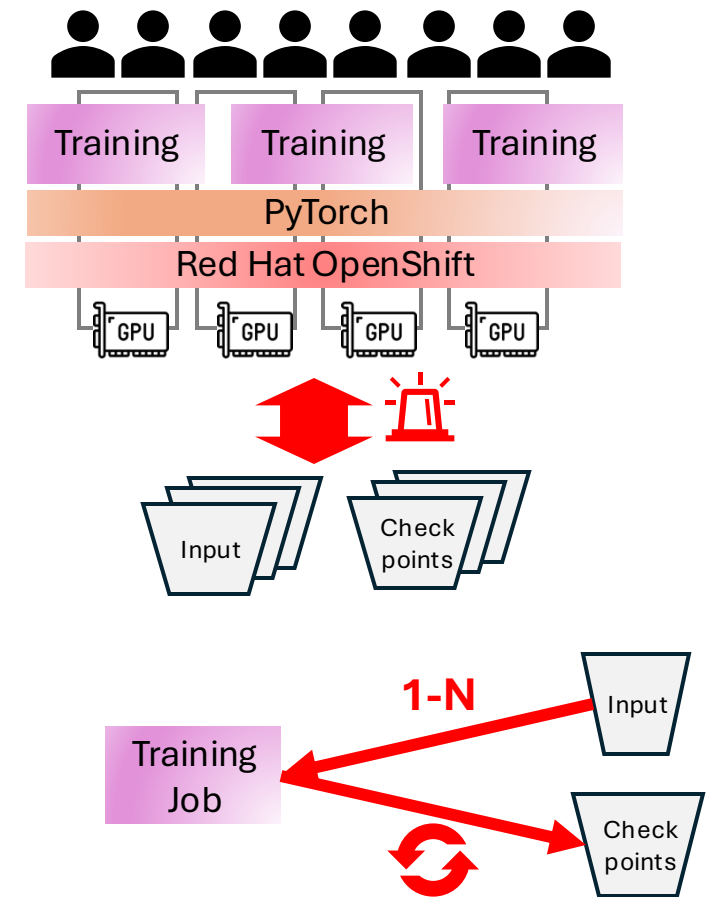
<https://dl.acm.org/doi/pdf/10.1145/3676641.3716280>

Data Access in AI workflows



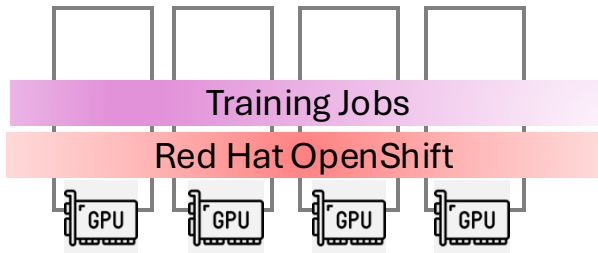
Storage for Model Training

- Multiple users run concurrent training jobs
 - Some jobs for weeks – months
- Native shared storage services in IBM Cloud
 - **IBM Cloud Object Storage (COS)**
 - IBM Cloud File Storage (NFS) – 1 GB/s per share only
 - IBM Cloud Block Storage – no shared namespace
- Typical training job uses
 - 1 bucket for input data – read all input data a few times
 - 1 bucket for checkpoints – many periodic writes, infrequent reads
- Everything gets bigger
 - Input data (more tokens): 1TB → 20TB
 - Checkpoints (larger models): 100GB → 1TB
 - Cluster size
- **Problems**
 - ✗ Slow checkpointing
 - ✗ Slow training data loading
 - ✗ Storage backend overload
 - ✗ Slow restore from checkpointing
 - Extended training times
 - Idling GPUs
 - Can't perform checkpoints as frequently as desired



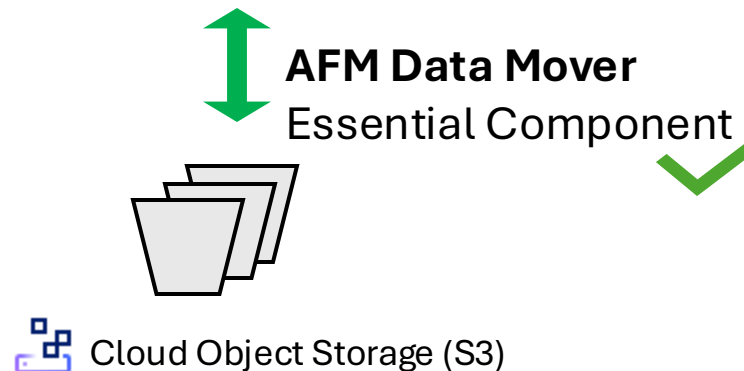
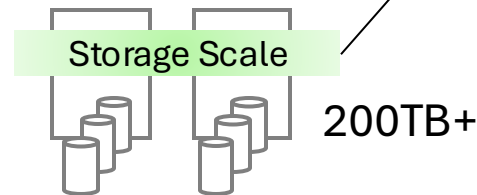
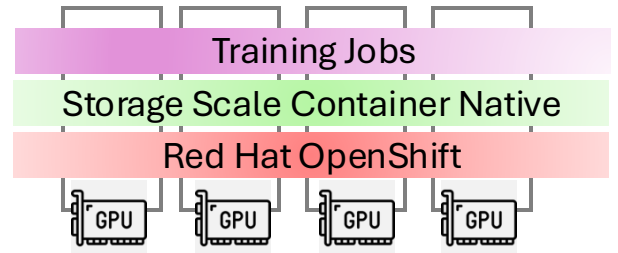
Multi-user Elastic Cache w/ Storage Scale

BEFORE



 Cloud Object Storage (S3)

NOW



 Cloud Object Storage (S3)

**Large, persistent,
and high-performance
cache dedicated
to the AI cluster**

- ✓ 1. Consistently fast checkpoints
- ✓ 2. Fast data load from cache
- ✓ 3. No backend overload

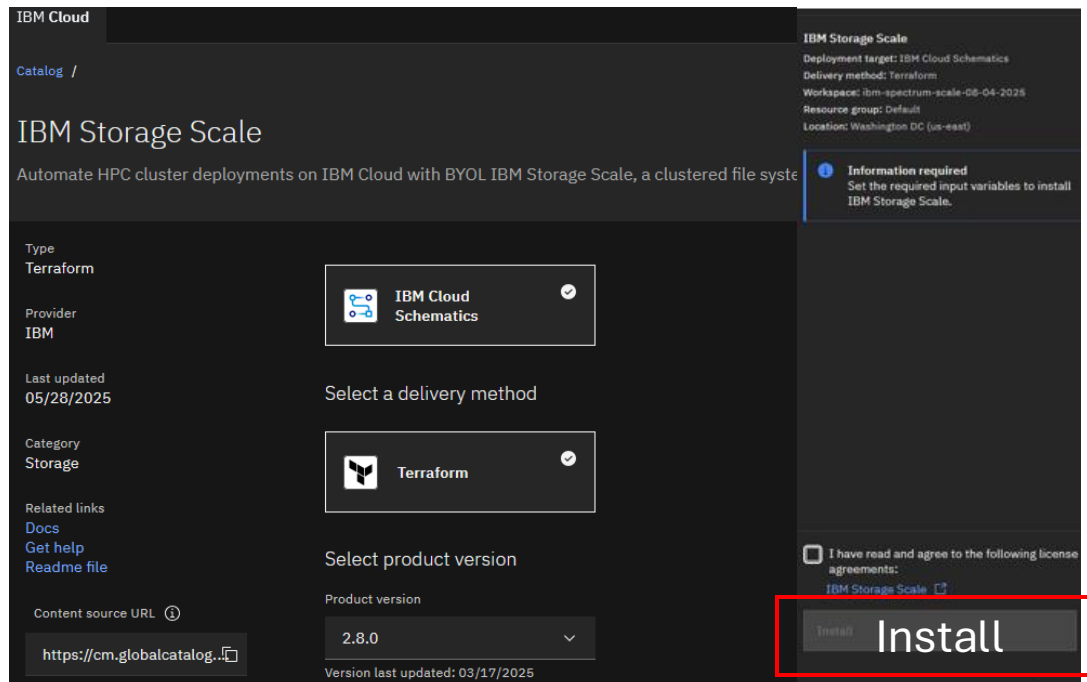
- ✓ 4. Automated data movement

Users now call it
"Infinite file system"

Automation for IBM Storage Scale in IBM Cloud

1. Deploy with IBM Cloud Tile

- IBM Schematics (Terraform) and Ansible based
- UI, CLI, API



2. Use Terraform/Ansible from Github

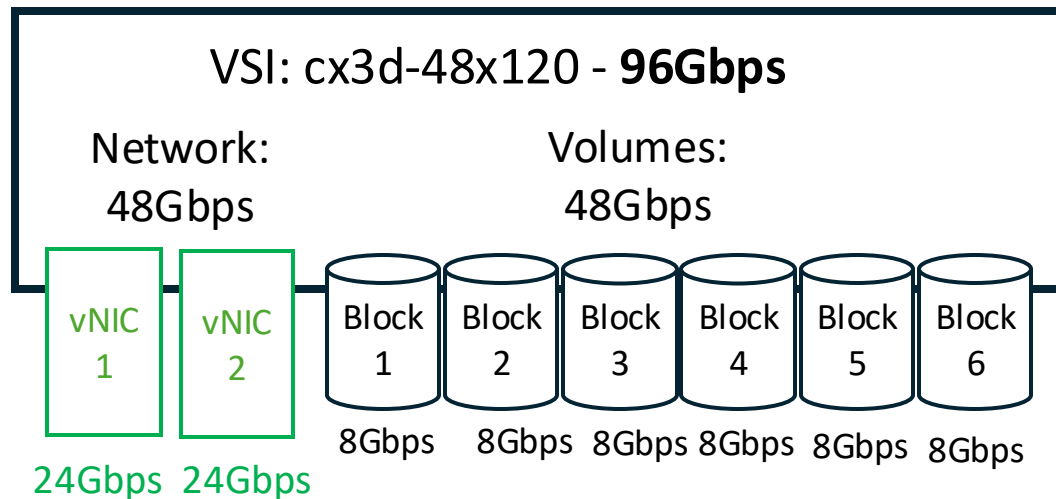
- <https://github.com/IBM/ibm-spectrum-scale-cloud-install>
 - AWS, GCP, Azure, (IBM Cloud)
- <https://github.com/IBM/ibm-spectrum-scale-install-infra>
- More customizable

<https://cloud.ibm.com/catalog/content/ibm-spectrum-scale>

Storage Cluster Example

Every client write results in a networked write to block: split available VSI network 50/50 between block and vNICs

Setup



6GB/s (48Gbps) per server

96GB/s per 16-node cluster

240TB if 2.5TB volumes used

- Use of (new) networked block storage type in IBM Cloud
- Alternatives
 - Baremetal
 - ECE / Replication
- Multiple interfaces support
 - MROT
- Some nodes are AFM gateways
- Scale GUI / REST API Server

Scale setup: No ECE, not data replication, 2-way metadata replication

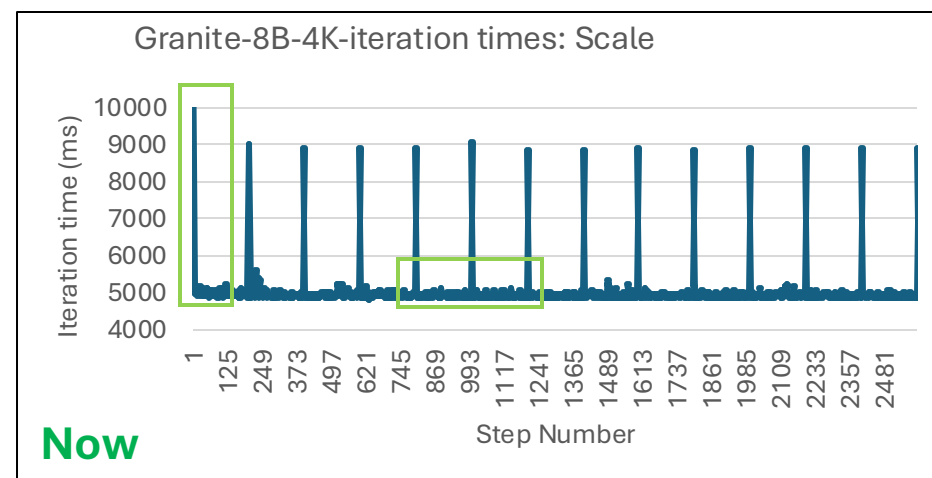
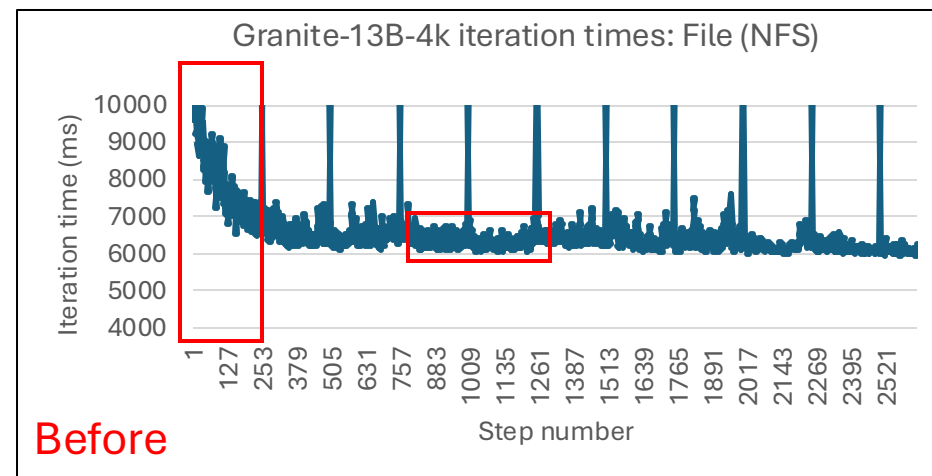
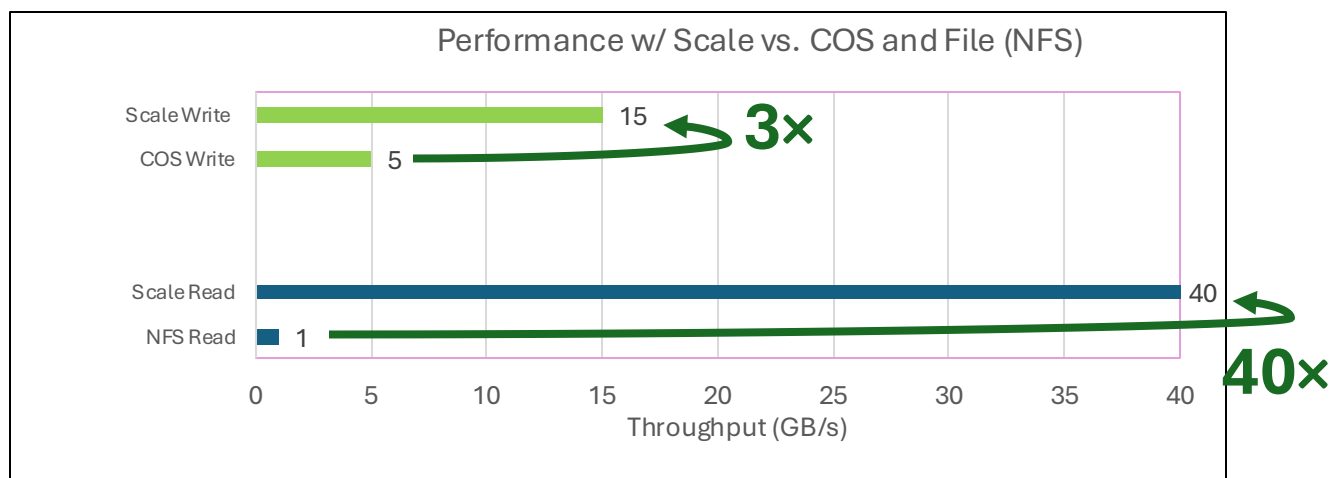
Compute Cluster

- Dynamic OCP clusters
 - Failures, node movement
- Largest Container Native Scale cluster – over 200 nodes
 - Dedicated non-GPU VSIs for quorum nodes
 - Node removal process
 - Kernel module compilation adjustments
- First use of Container Native Scale in IBM Cloud

Production Deployment Results

Summary

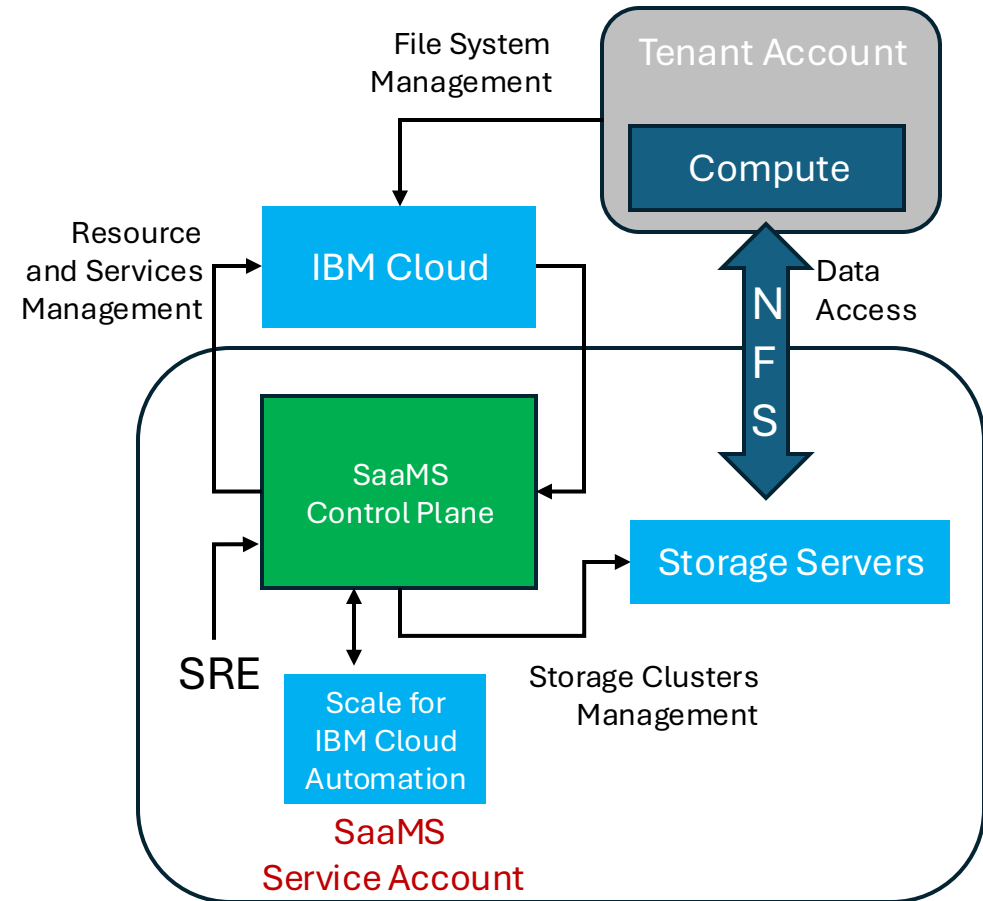
- **3×** faster writes (than COS) and **40×** faster reads (than NFS)
- **10+%** improvement in training speed
- Consistent training step times
 - w/ File – 9-6 seconds, 50% variation
 - w/ Scale – 4.8 – 5.2 seconds, **only 10% variation**
 - No ramp down of the training time



Users demand need for more capacity, efficiency, and operational simplicity

Research: Scale as a Managed Service

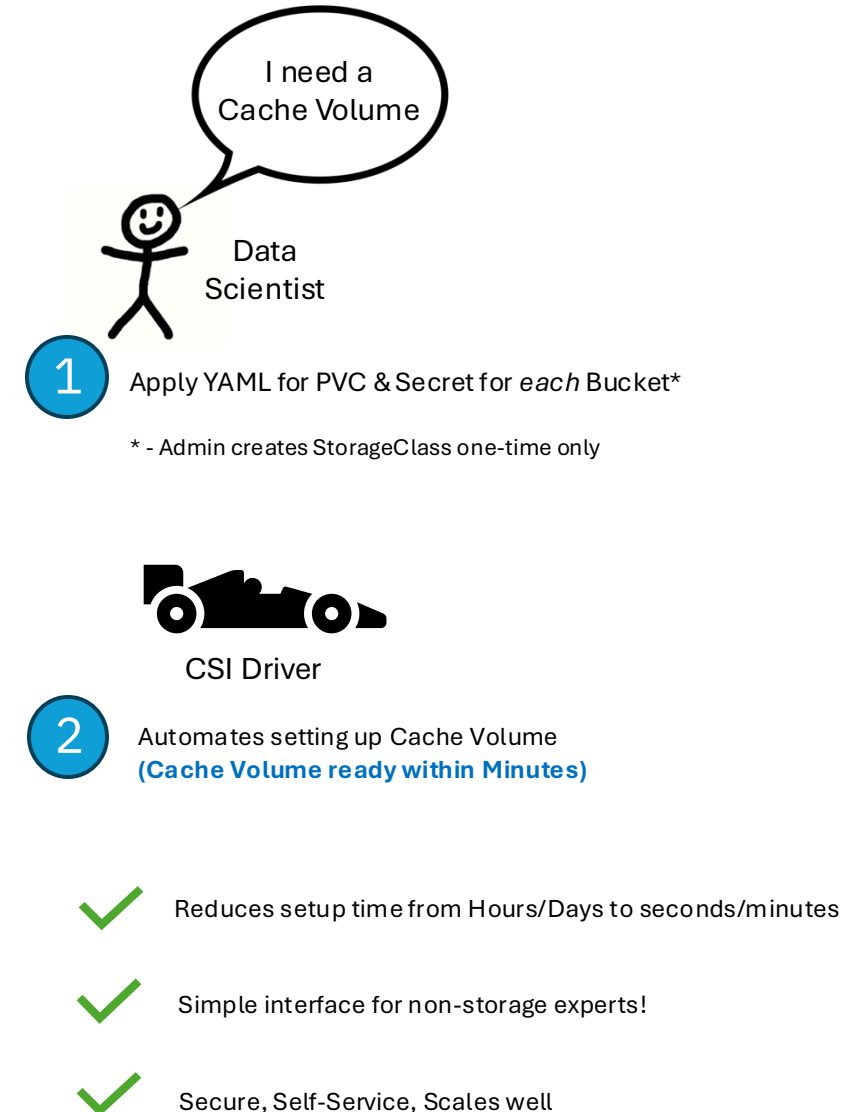
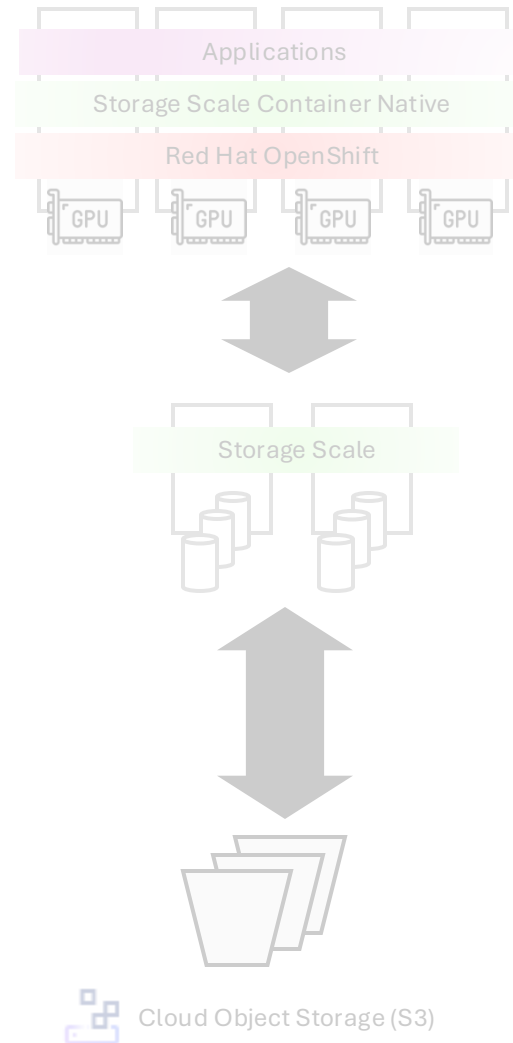
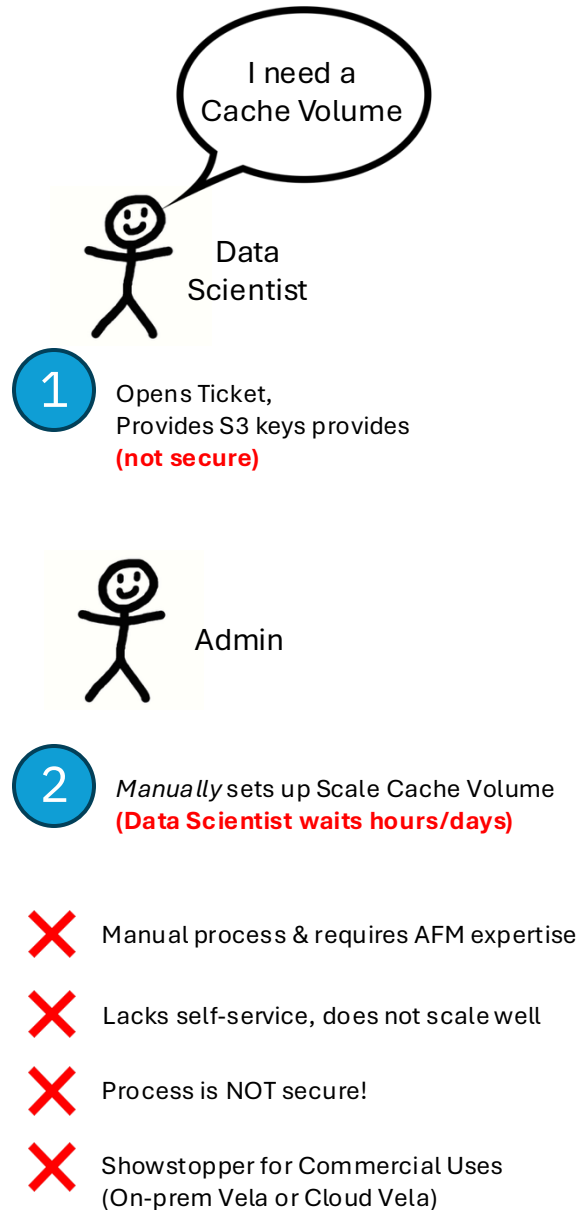
- Managing and operating Scale clusters requires expertise
 - Initial deployment, upgrades, resizing, troubleshooting, etc.
- Cloud users expect more simplicity
 - “I need a highly-available file system volume of size 100TB w/ 100GB/s performance (1GB/s/TB)”
- Storage cluster in service account
 - Managed by IBM Storage SREs
- NFS – so that client clusters don’t run Scale software
- Clusters not shared between tenants
- Zonal service
- Integrated billing, catalog, monitoring, etc.



Users demand need for more capacity, efficiency, and operational simplicity

Scale Cache Volumes

Earlier process



What are Scale Cache Volumes?

1. Release Info

- 5.2.1 (Tech Preview)
- 5.2.3 (General Availability)

2. Maps PV to S3 Bucket

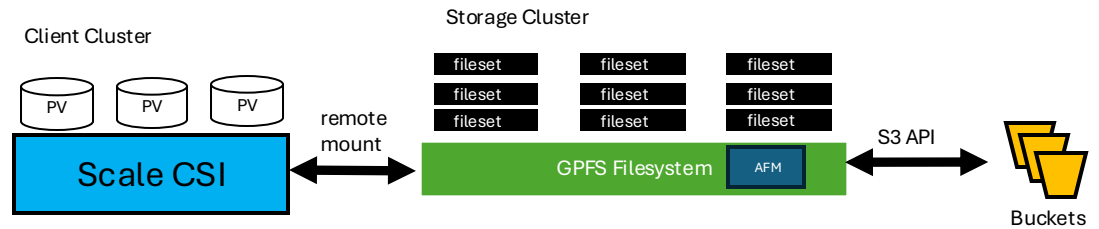
- Uses AFM fileset as a caching layer
- *Most* AFM modes are supported

3. Scale CSI driver supports...

- Dynamic & Static provisioning
- 1:1 or Many:1 (PV → Bucket)

4. Advanced Features (as of release 5.2.3)

- Adjust AFM parameters (fine tuning)
- Manual cache prefetch / eviction



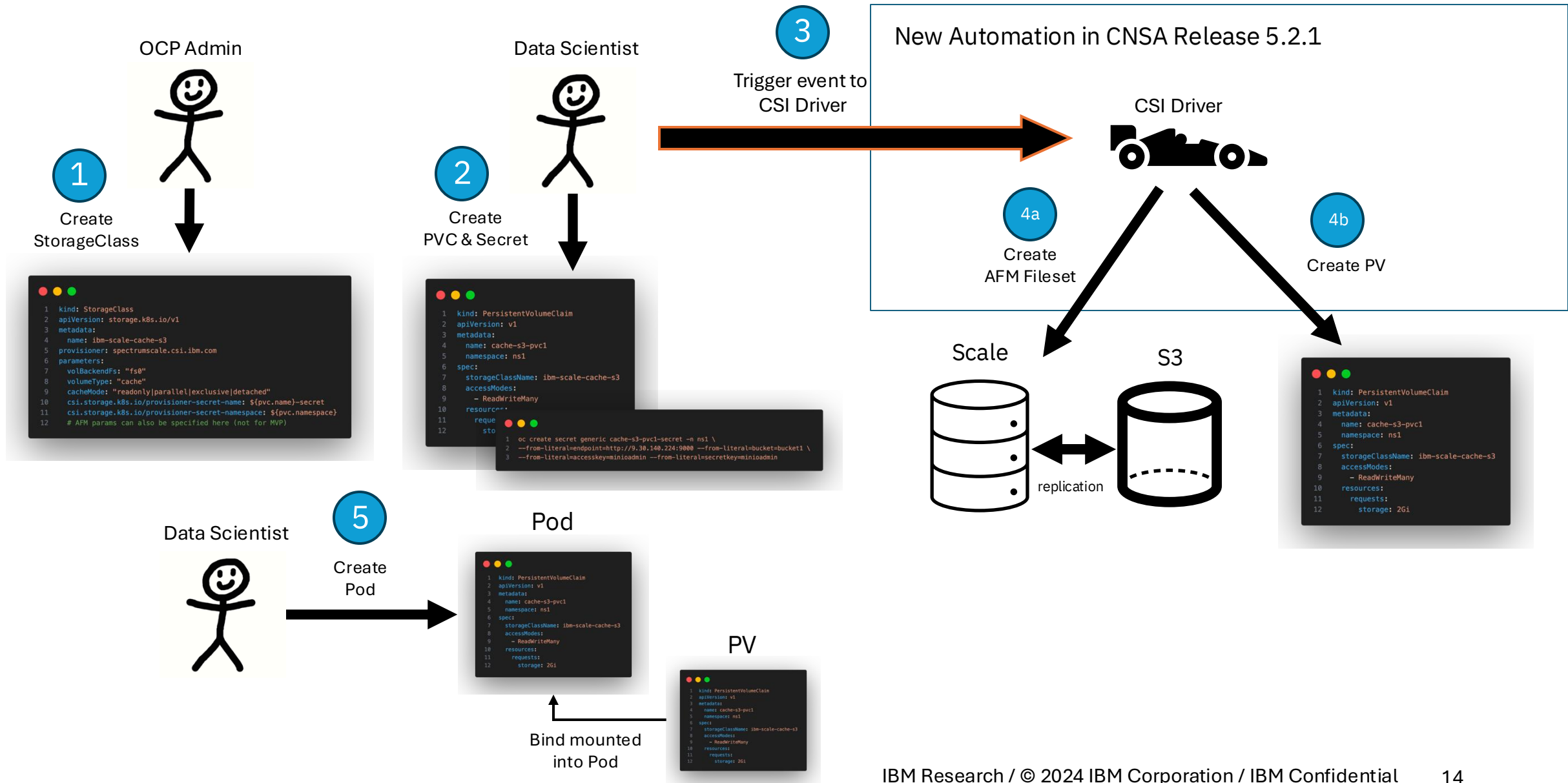
Scale Cache Volume is PV → Bucket mapping

Mapping: PV → AFM fileset → Bucket

AFM does File → Object translation

<https://www.ibm.com/docs/en/scalecontainernative/5.2.3?topic=caching-data-from-object-storage>

Scale Cache Volumes - workflow



Interface Design – Storage Class

OCP Admin



Create

```
1 kind: StorageClass
2 apiVersion: storage.k8s.io/v1
3 metadata:
4   name: ibm-scale-cache-s3
5   provisioner: spectrumscale.csi.ibm.com
6   parameters:
7     volBackendFs: "fs0"
8     1 volumeType: "cache"
9     2 cacheMode: "readonly|parallel|exclusive|detached"
10    csi.storage.k8s.io/provisioner-secret-name: ${pvc.name}-secret 3
11    csi.storage.k8s.io/provisioner-secret-namespace: ${pvc.namespace}
12    # AFM params can also be specified here (not for MVP)
```

1

volumeType: cache

-> Means to create AFM Fileset
(as opposed to non-AFM Fileset)

2

cacheMode set AFM mode

(optionally set)

- Readonly -> RO
- Parallel -> IW
- Exclusive -> SW
- Detached -> LU

If not set, then access mode in PVC sets the AFM mode.

3

▪ **provisioner-secret-name**

- Build-in template
- Specifies where S3 Bucket access & credentials will be stored

Interface Design – Secret & PVC

Data Scientist



Create

```
1 oc create secret generic cache-s3-pvc1-secret -n ns1 \
2 --from-literal=endpoint=http://9.30.140.224:9000 --from-literal=bucket=bucket1 \
3 --from-literal=accesskey=minioadmin --from-literal=secretkey=minioadmin
```

1 S3 Endpoint

2 Secret name
(Matches Secret template
In Storage Class)

3 Bucket Name

4 Bucket Access &
Secret Keys

Data Scientist



Create

```
1 kind: PersistentVolumeClaim
2 apiVersion: v1
3 metadata:
4   name: cache-s3-pvc1
5   namespace: ns1
6 spec:
7   storageClassName: ibm-scale-cache-s3
8   accessModes:
9     - ReadWriteMany 5
10  resources:
11    requests:
12      storage: 2Gi 6
```

5 Controls AFM mode*
RWO, RXW, RXOP -> **IW** mode
ROX -> **RO** mode
* If not set in StorageClass

6 Size of Volume**
CSI driver sets Fileset Quota

** StorageClass architecture allows Admins to set
Storage Resource Quotas (e.g. max storage)

CacheVolume CR

automatically generated by CNSA after Fileset created

Data Scientist



```
1 kind: CacheVolume
2 apiVersion: v1alpha1
3 metadata:
4   name: cacheVolume-<volume name>
5   namespace: ns1
6 spec:
7   Mode: ReadOnly
8   # Other AFM config parameters
9 status:
10  filesystem: "gpfs0"
11  fileset: <AFM fileset>
12  size: 2Gi
13  endpoint: http://9.30.206.86:9000
14  bucket: bucket1
```

1

Users can modify AFM Params

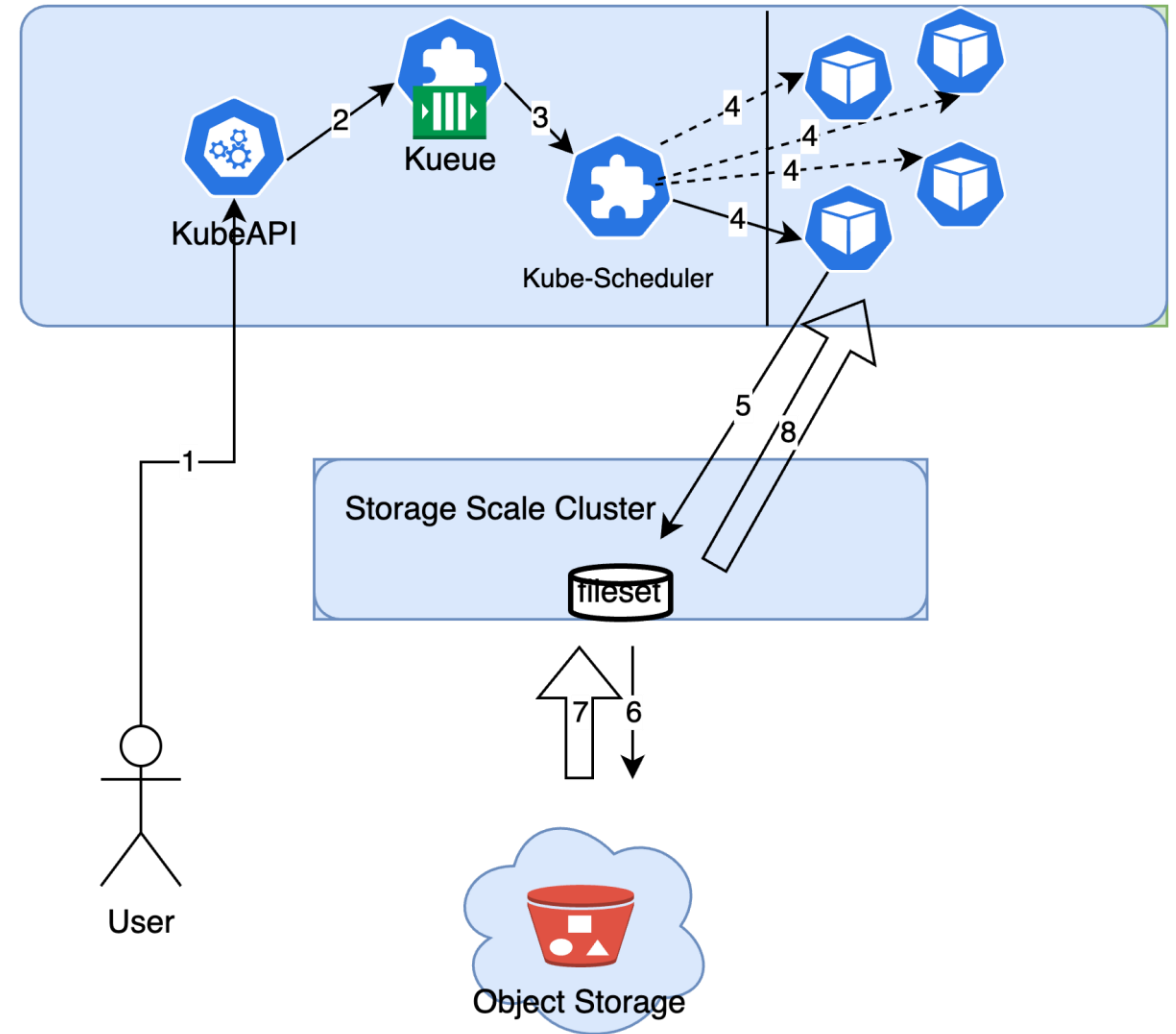
2

Exposes AFM status & errors

Current architect overview with Kueue

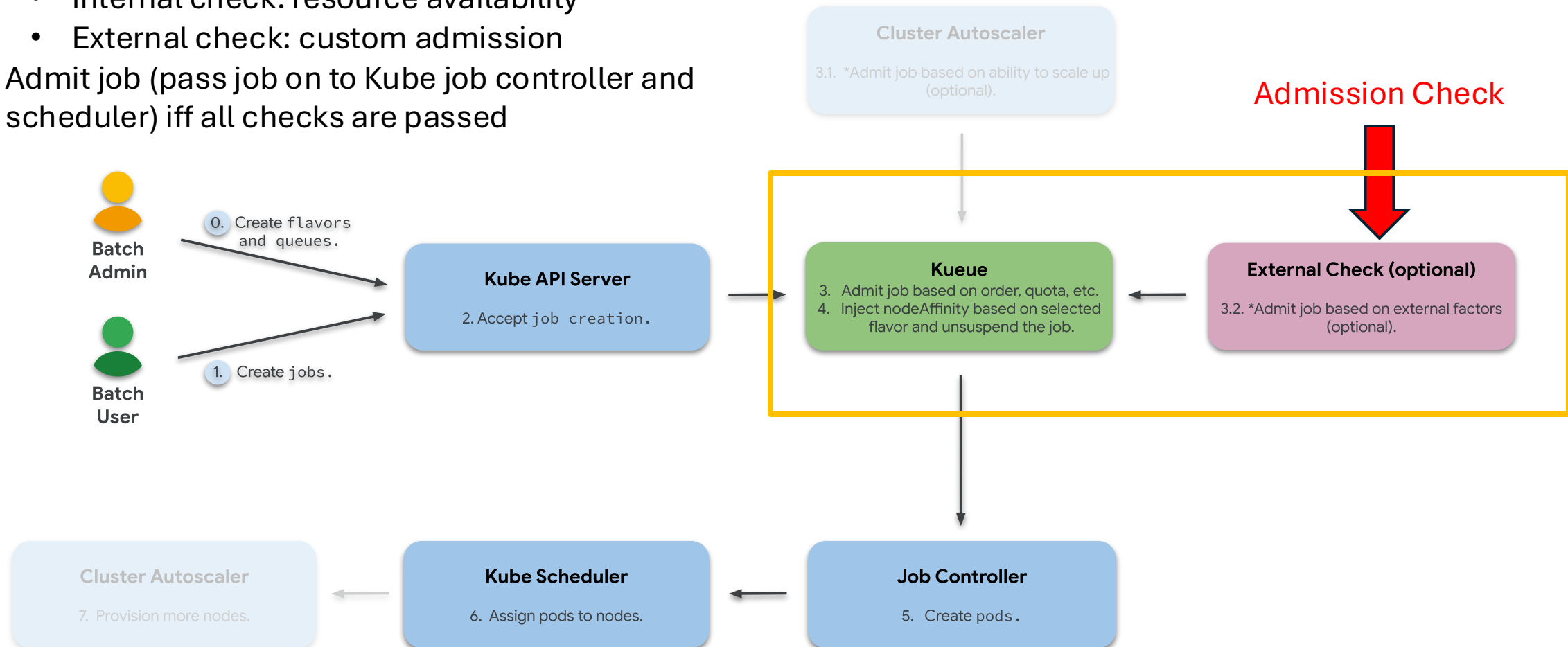
Problem: kueue admits job as soon as GPUs (and other resources) are available, but data it needs is not yet available in the storage cluster (still needs to be fetched from object store).

1. User submits job to the OCP/Kubernetes cluster
2. KubeAPI sends job to Kueue for scheduling
3. Kueue
 1. queues workload
 2. reserves resources
 3. admits workload
4. Kube-scheduler assign node to pod
5. Pod runs training jobs and fetches dataset from storage cluster
6. Storage cluster fetches data from object storage
7. Object storage sends data to storage cluster
8. Storage cluster sends data to training pod



Custom Admission Check in Kueue

- Job scheduling with Kueue
 - Internal check: resource availability
 - External check: custom admission
- Admit job (pass job on to Kube job controller and scheduler) iff all checks are passed



Solution: use external admission check to fetch data from object store before admitting the job.

1. User submits job to the Kubernetes cluster
2. KubeAPI sends job to Kueue for scheduling

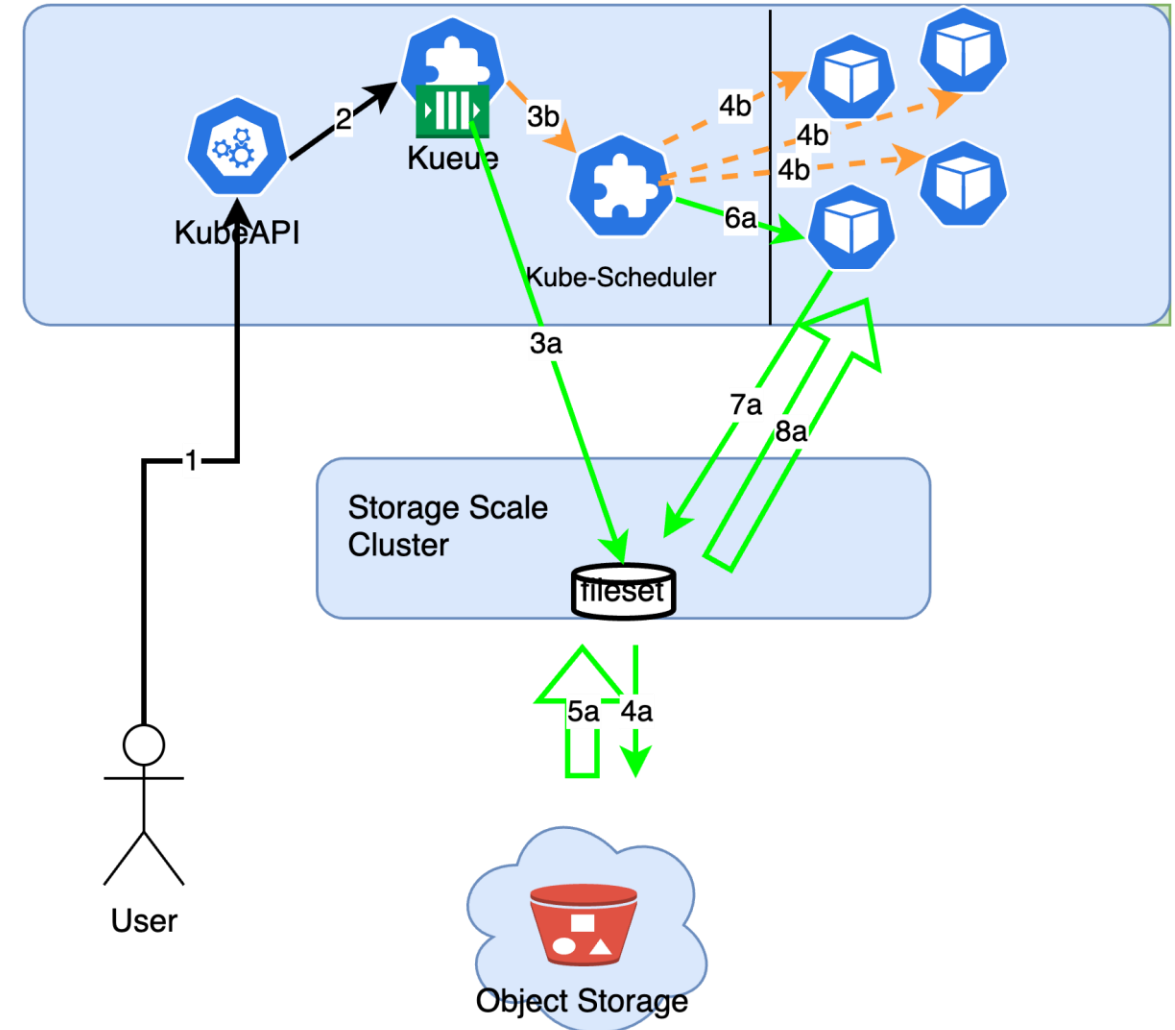
Job 1

- 3a. Kueue triggers prefetch of dataset to storage cluster
- 4a. Storage cluster fetches data from object storage
- 5a. Object storage sends data to storage cluster
- 6a. Kube-scheduler assign node to pod
- 7a. Pod runs training jobs and fetches dataset from storage cluster
- 8a. Storage cluster sends data to training pod

Job 2, 3, 4...

- 3b. Kueue admits workloads that passed admission and dataset is prefetched
- 4b. Kube-scheduler assign node to pod
- ...

Dataset prefetch



Cluster Setup

- ClusterQueue

```
apiVersion: kueue.x-k8s.io/v1beta1
kind: ClusterQueue
metadata:
  name: "cluster-queue"
spec:
  namespaceSelector: {} # match all.
  resourceGroups:
  - coveredResources: ["cpu", "memory"]
    flavors:
    - name: "default-flavor"
      resources:
      - name: "cpu"
        nominalQuota: 1
      - name: "memory"
        nominalQuota: 1Gi
  admissionChecks:
  - custom-ac
```

- Custom Admission Check

```
apiVersion: kueue.x-k8s.io/v1beta1
kind: AdmissionCheck
metadata:
  name: custom-ac
  namespace: teammlb
spec:
  controllerName: kueue.sandbox.com/prefetch-request
```

- Job Specification

```
spec:
  pytorchReplicaSpecs:
    Master:
      replicas: 1
      restartPolicy: Never
      template:
        metadata:
          namespace: teammlb
          annotations:
            kueue.sandbox.com/pvc-name: "prefetch-pvc"
            kueue.sandbox.com/directory: "/dir"
            kueue.sandbox.com/sub-directory: "subdir"
        spec:
          volumes:
            - name: topology-volume
          containers:
            - name: pytorch
              image: ghcr.io/foundation-model-stack/base:pytorch-latest-nightly-20230126
              imagePullPolicy: IfNotPresent
```

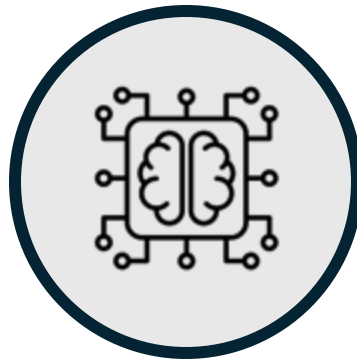
Model Training in Public Clouds: Case for IBM Storage Scale

Vasily Tarasov, Scott Guthridge, Jeremy Cohn,
Marc Eshel, Leo Luan, Travis Janssen, Alex Merenstein,
Frank Schmuck, Lei Pan, Thanh Pham,
Veera Deenadhayalan, Swami Sundararaman,
Seelam Seetharami, Sophia Wen, Talia Gershon
IBM Research - Hybrid Cloud Infrastructure

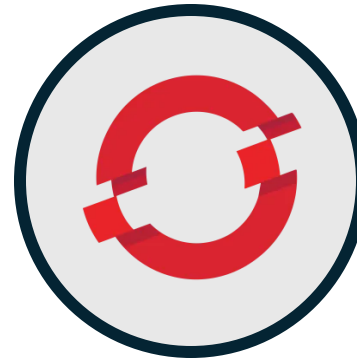
Kevin O'Connor, Abdoulaye Traore,
Chris Laibinis, Brent Wolfe, Carlos Fonseca
IBM Research - Emerging Technology Engineering
Piyush Chowdhary
IBM Cloud – Scale
Brian Reitz, Steve Pritko, Piyush Shivam
IBM Cloud – Block Storage



+



+



+



IBM Storage Scale

Model Training

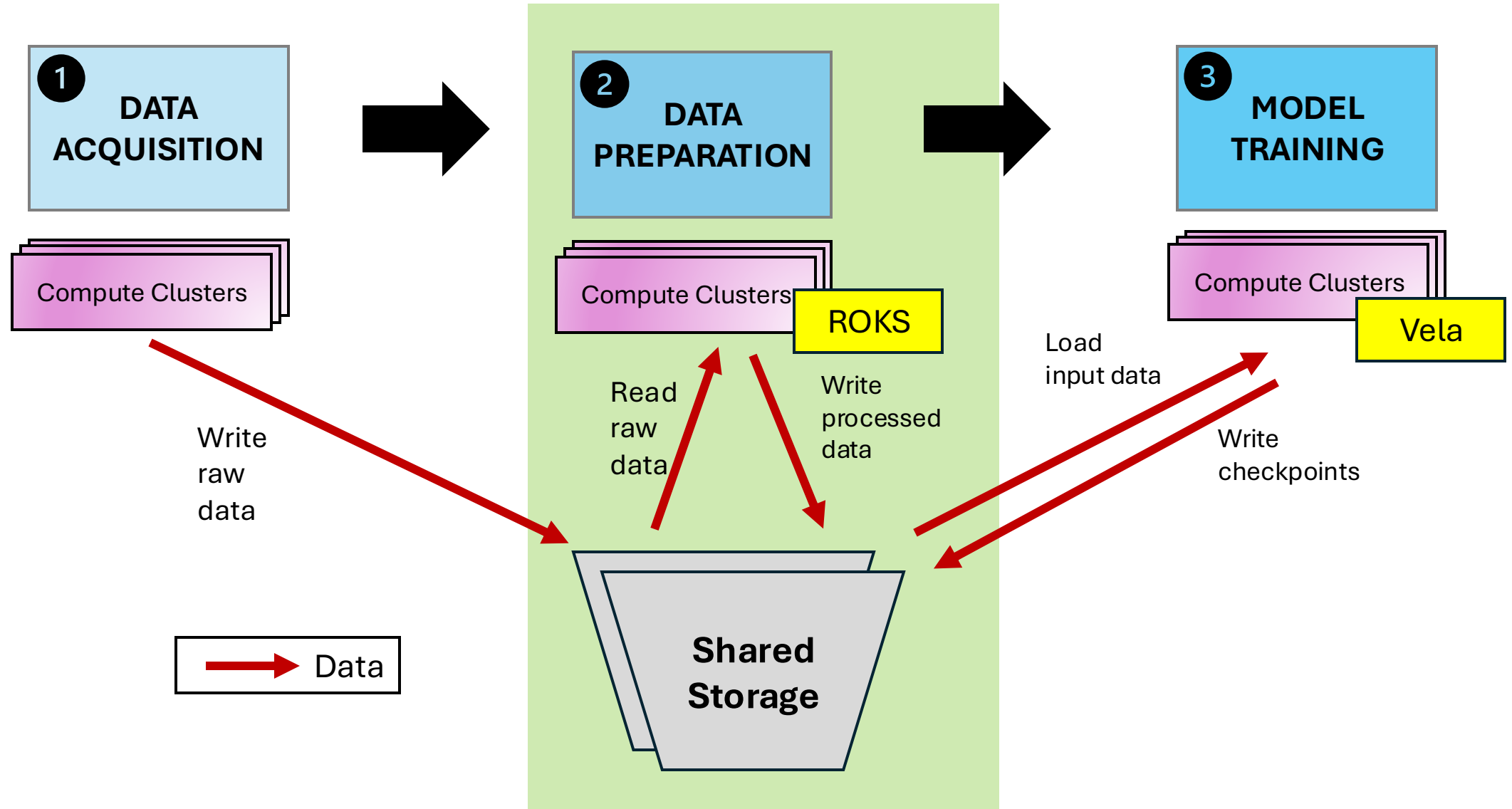
Red Hat Open Shift

IBM Cloud

Disclaimer: Research work

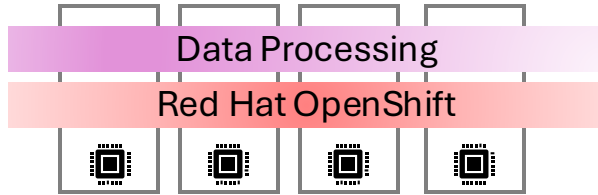
Backup

Data Access in Data and AI workflows

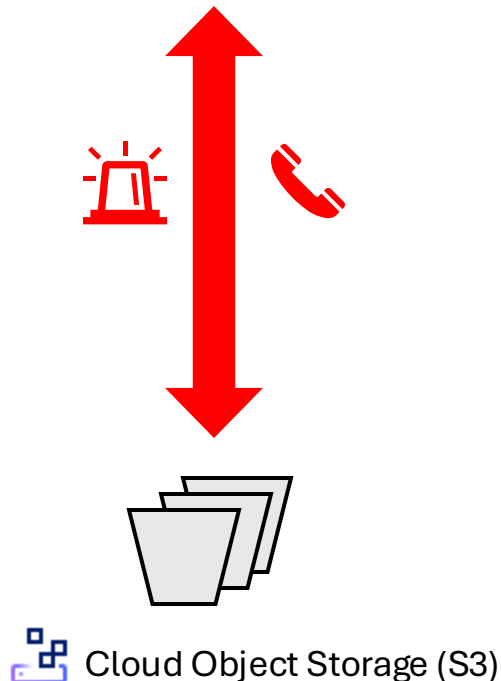
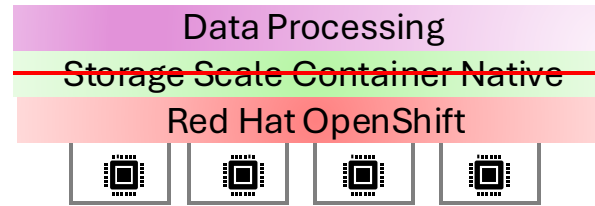


Storage for Data Preparation

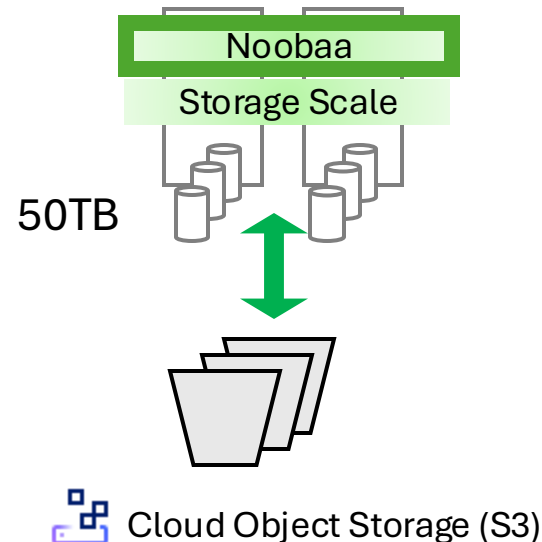
Compute Cluster (ROKS)



Compute Cluster (ROKS)



Storage Cluster



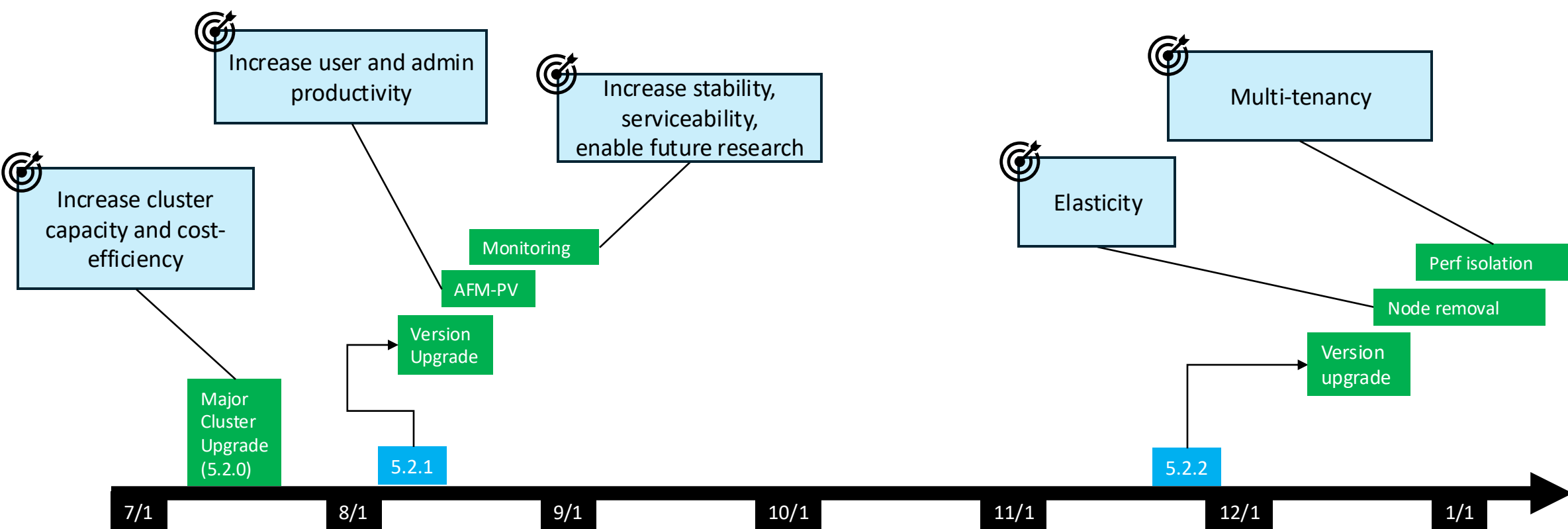
- Embarrassingly parallel workload for which one can deploy larger compute clusters to accelerate data preparation
- **Problem.** When using over 70 compute nodes, data preparation applications overload COS backend.
Limits the speed of data preparation.
- Deploying cache allows to flatten I/O burst, not overload COS, and run data preparation at larger scale – 200 nodes (target).
 - Lots of data reuse
- Data factory applications require object storage interface (S3), not file system. Scale provides object storage (S3) interface through Noobaa technology
 - No need to deploy Scale Cloud Native ☺
- Status
 1. Functional PoC completed
 2. Performance PoC in progress: Larger scale, reduced peak load on COS, higher throughput



- Scale release



- Deployment in Prod Vela



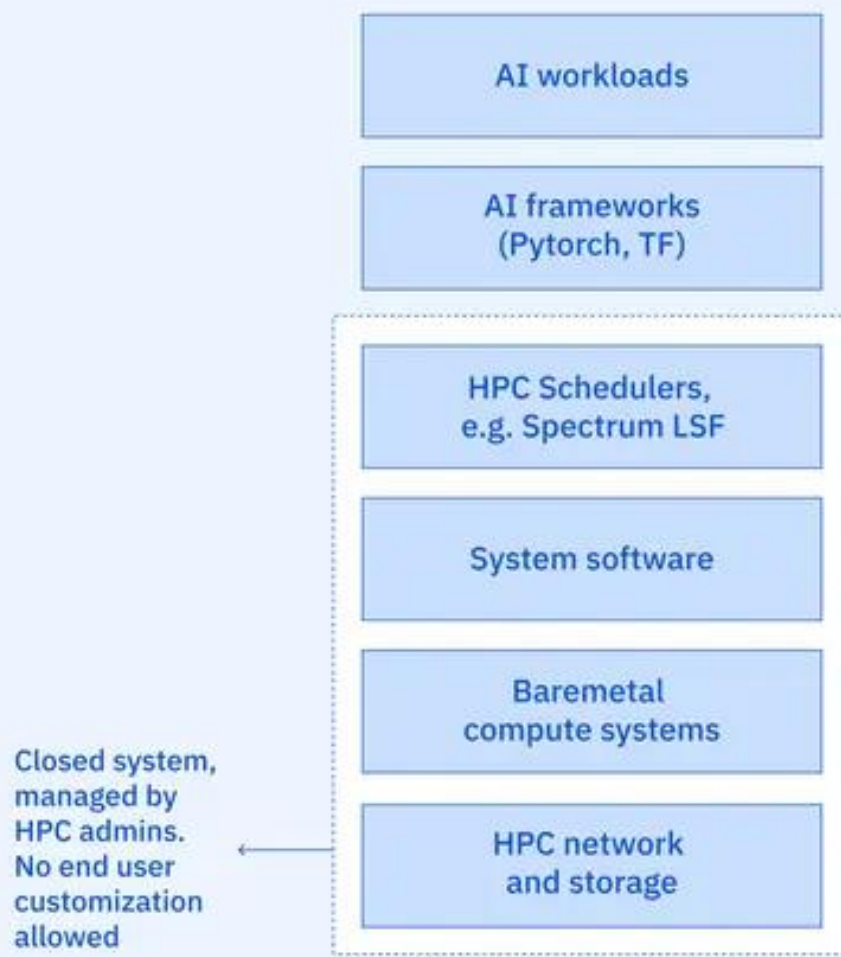
Upcoming Major Upgrade

1. Expand capacity to **250TB** to accommodate more workloads
2. Reduce operational **cost per TB by 50%**
3. Automate user **data access configuration**
4. Improve **system serviceability** by adding performance, health, and workload monitoring

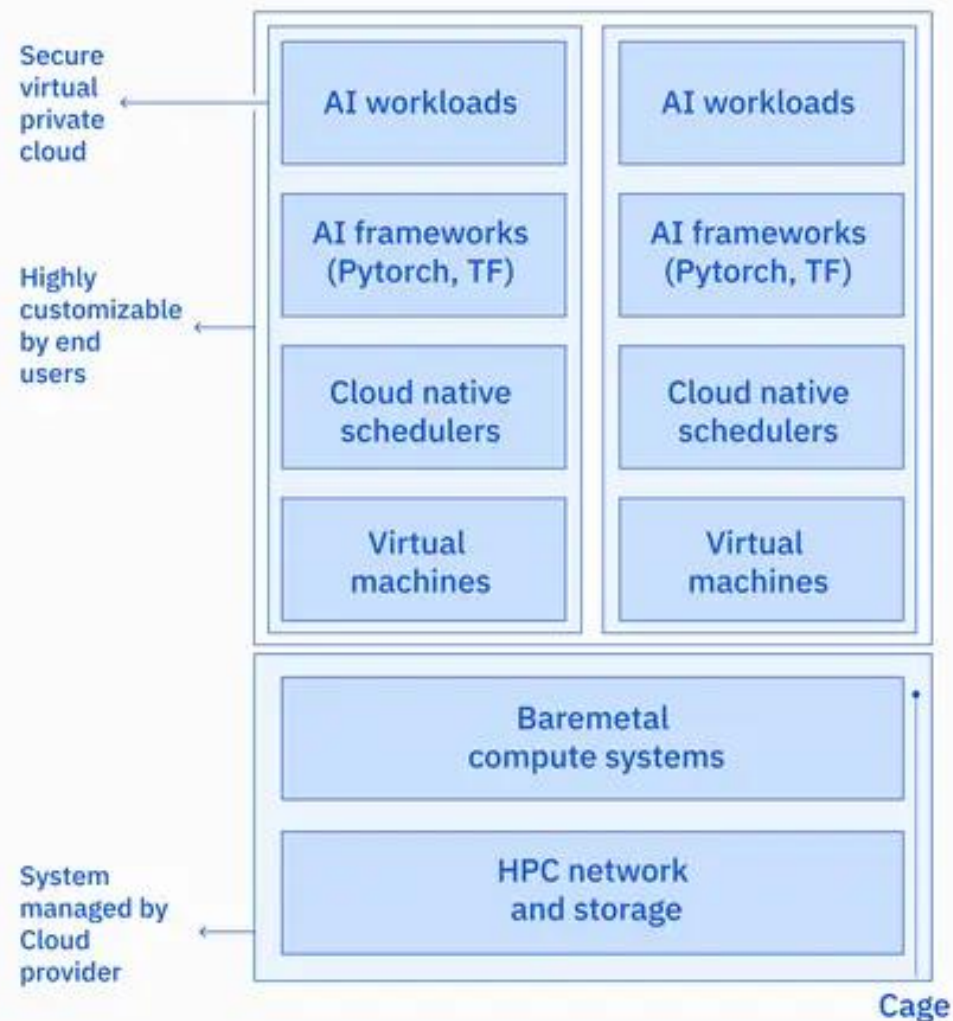
Future

1. Elasticity, performance isolation, operational simplicity
2. Holistic multi-tenancy (security, access, administration)
3. Use of local devices in compute nodes (50× potential improvement!)

HPC AI system stack



Cloud-native AI system stack



Kueue CRDs: admission check and workload

- Workload
 - A job or task to be scheduled and run
 - States: created, suspended, running, succeeded, failed, etc.
- AdmissionCheck
 - Define rules and conditions for a workload to be admitted
 - Configured before taking workloads
- Admission control
 - User submits a workload
 - Kueue check and reserves quota for the workload
 - Admission check evaluates the workload
 - Kueue only admits the workload if all admission checks are passed

References

- <https://github.com/project-codeflare>
 - <https://github.com/project-codeflare/multi-cluster-app-dispatcher>
 - <https://github.com/project-codeflare/multi-cluster-app-dispatcher/blob/main/doc/deploy/deployment.md>
 - <https://github.com/project-codeflare/appwrapper>
- <https://research.ibm.com/blog/AI-supercomputer-Vela-GPU-cluster>
- <https://research.ibm.com/blog/ibm-pytorch-cloud-ai-ethernet>
- <https://research.ibm.com/blog/ibm-artificial-intelligence-unit-aiu>
- How to Deploy a High-performance Distributed AI Training Cluster with NVIDIA GPUs and KVM (Presented by IBM)
<https://www.nvidia.com/en-us/on-demand/session/gtcspring22-s42633/>
Keynote Title: Hardware-Middleware System co-design for flexible training of foundation models in the cloud
<https://middleware-conf.github.io/2022/keynote-speakers/>
- <https://github.com/ibm-granite>