

HPC for the Enterprise Architect

Microsoft in Higher Education 2019

Andy Howard

HPC Software & Services
anhoward@microsoft.com

Topics Today

Introduction into HPC workflows

Scheduling

What are schedulers, and which are supported?

Orchestration

What exactly is orchestration and how is it different from provisioning or scheduling?

Application Install

What are the options for installing and setting up applications in an HPC environment?

Hybrid-Scenarios

What do users mean when they say hybrid?

What is supported?

HPC Compute & Storage Offerings

Example Workflows

What is High Performance Computing

The **coordination of many distinct computational resources** (CPUs, GPUs, Memory, Storage) to **solve** a computational problem.

Two Types of HPC Jobs

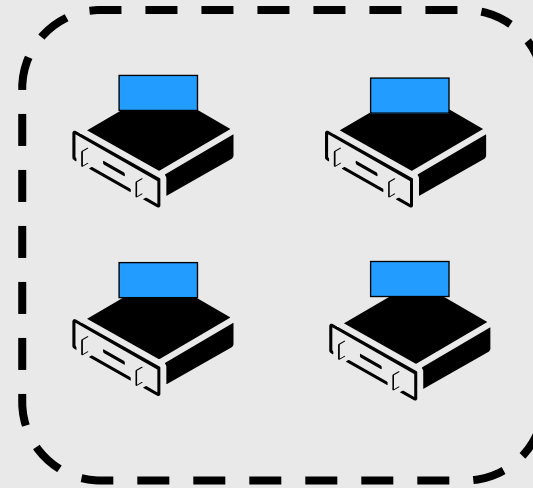
Pleasantly parallel

Applications do not communicate

May share common store & data

May have dependencies

E.g. Monte Carlo simulations, image/video rendering, genetic algorithms, sequence matching

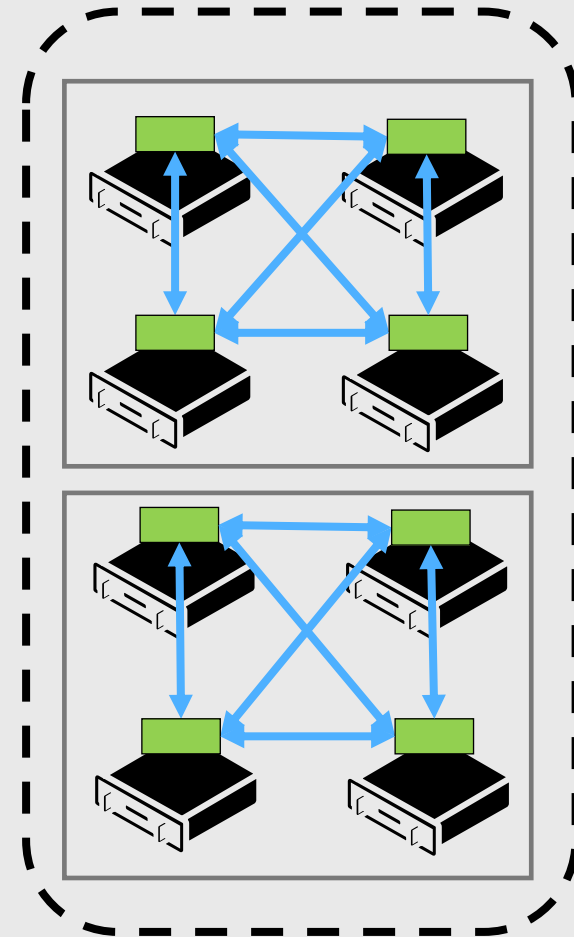
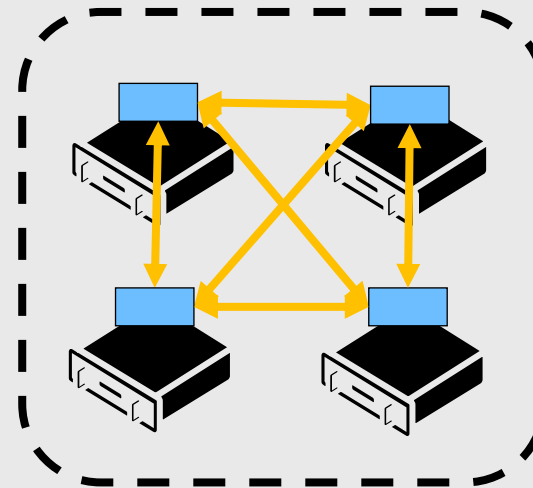


Tightly coupled

Applications communicate; mainly use MPI

Requires low latency, high bandwidth networking for scale

Examples: AI training, car crash simulation, fluid dynamics, climate modeling, reservoir simulation, manufacturing modeling



Schedulers, Orchestrators, and Workflow Managers

Scheduler

- Controls the unattended, background, program execution of jobs
- Matches jobs to available resources
- Often combined with a “Resource Manager”

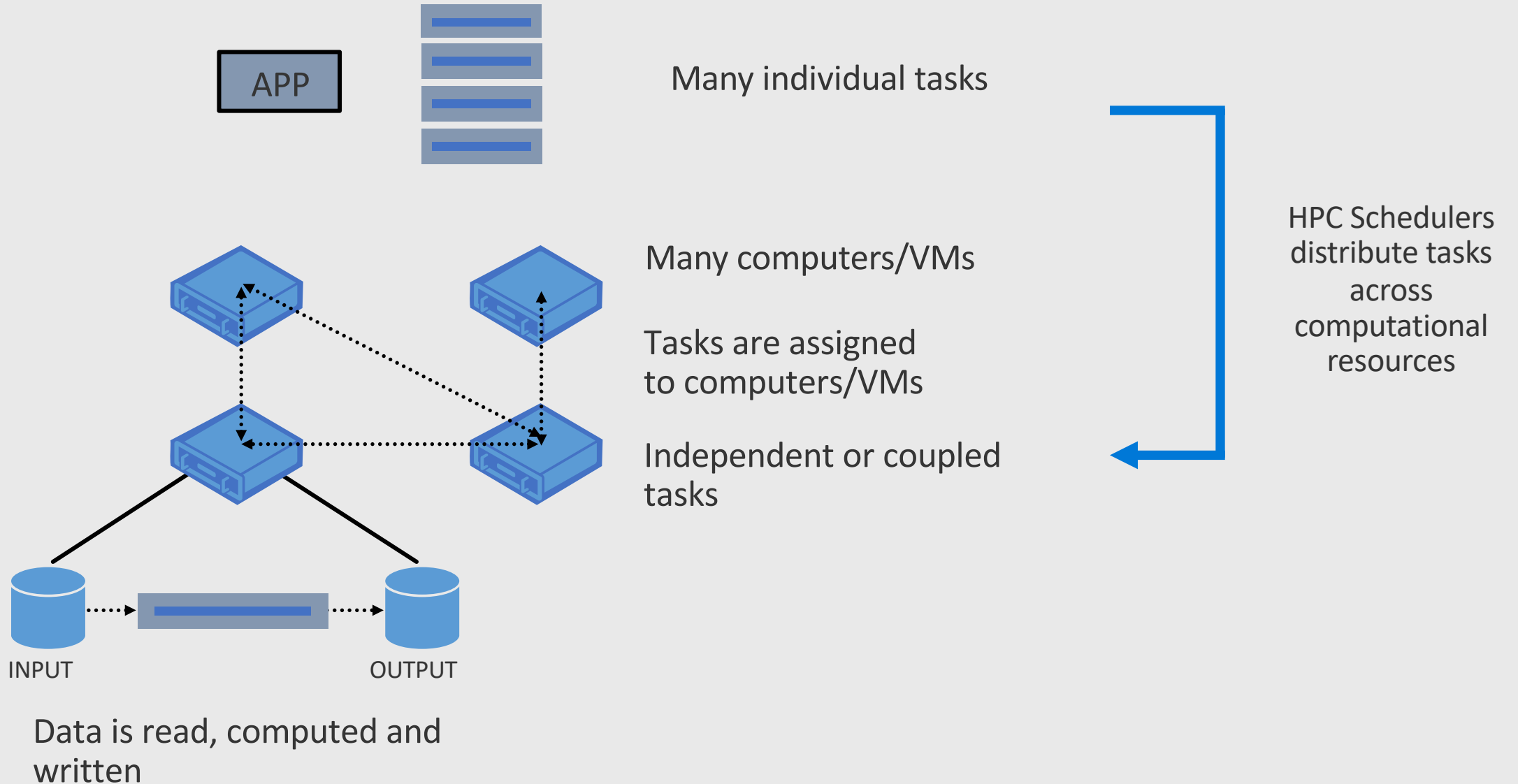
Orchestrator

- Provisions virtual and ephemeral resources, configures them into a working system
- In the context of HPC, creates a organized cluster of worker nodes and the associated services (File Systems etc)

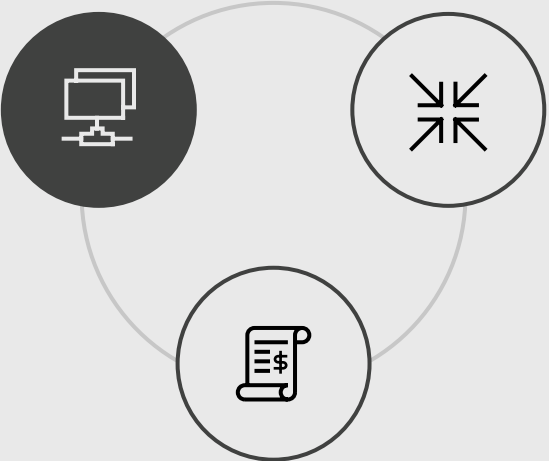
Workflow Manager

- Takes the set and order of tasks that a user wants to run, converts these into jobs that a scheduler understands, and uses an orchestrator to provision the necessary resources

What is Job Scheduling?



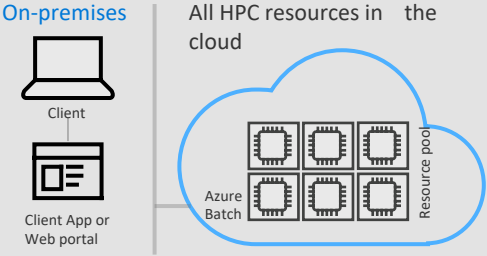
Does Azure provide Job Schedulers?



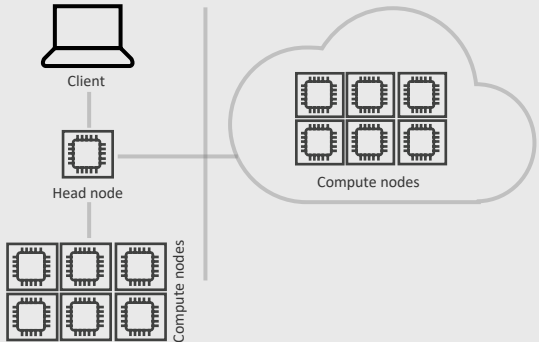
Azure Batch
running jobs

Azure CycleCloud
running clusters

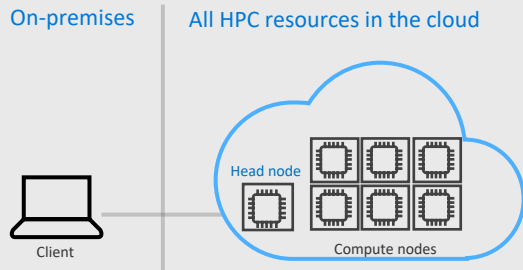
HPC as a service Hybrid/burst Azure cluster



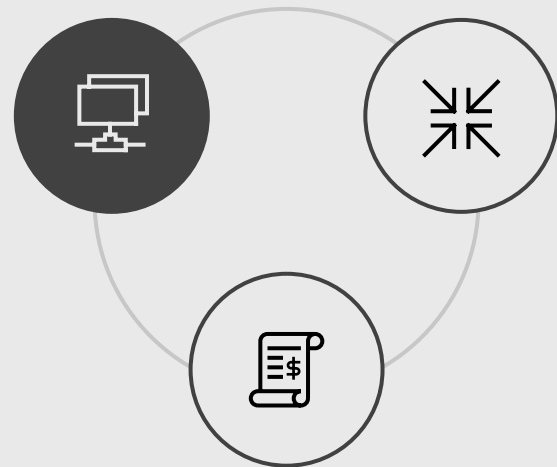
Batch has a native scheduler



CycleCloud orchestrates traditional schedulers



Azure CycleCloud



CycleCloud is NOT a Scheduler

Scheduler Support

Provides autoscaling and orchestration for:

LSF
PBSPro
Grid Engine
Slurm
HTCondor

Azure HPC Platform and Services

App Users

Developers

HPC End-users, IT Staff, Line of Business Mgr

Parallel R

VFX Plug-Ins

SaaS / Client
Solution

Azure ML

Cluster templates to run *existing, on-prem*
HPC schedulers and applications on Azure



Azure Batch

VM Management & Job Scheduling



Azure CycleCloud

Hybrid & Cluster Manager for HPC/AI



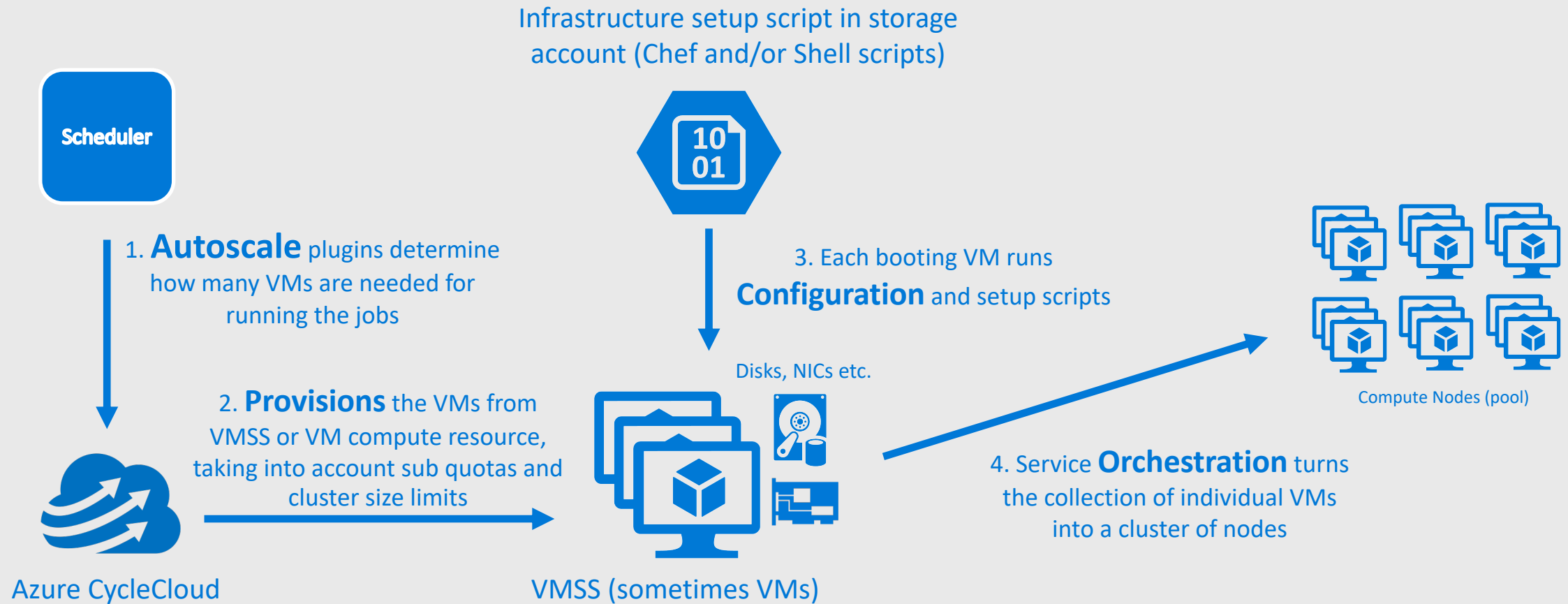
Cloud Services,
VMs, VMSS



Hardware

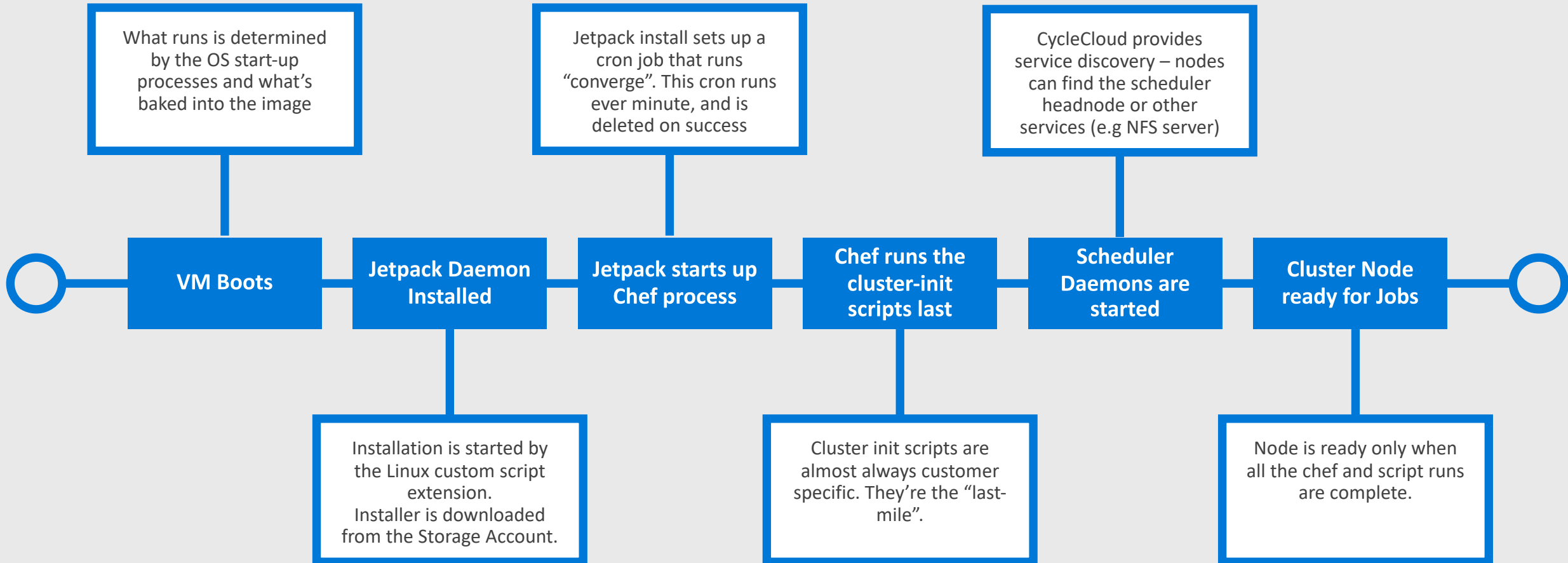
CycleCloud

Orchestration: From autoscale to cluster nodes



CycleCloud Node Preparation Detail

Provision -> Preparation -> Configuration -> Orchestration



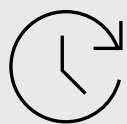
Challenges with on-premises HPC



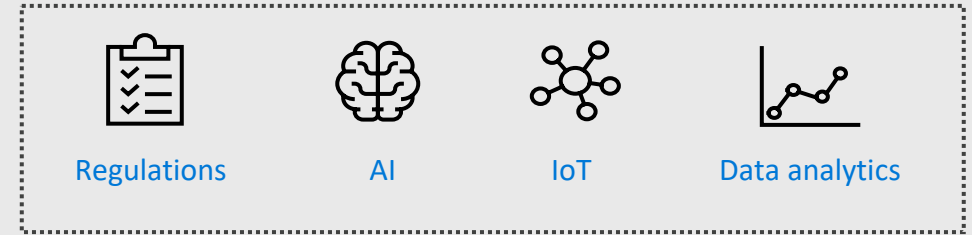
Finite resources do not scale with the business



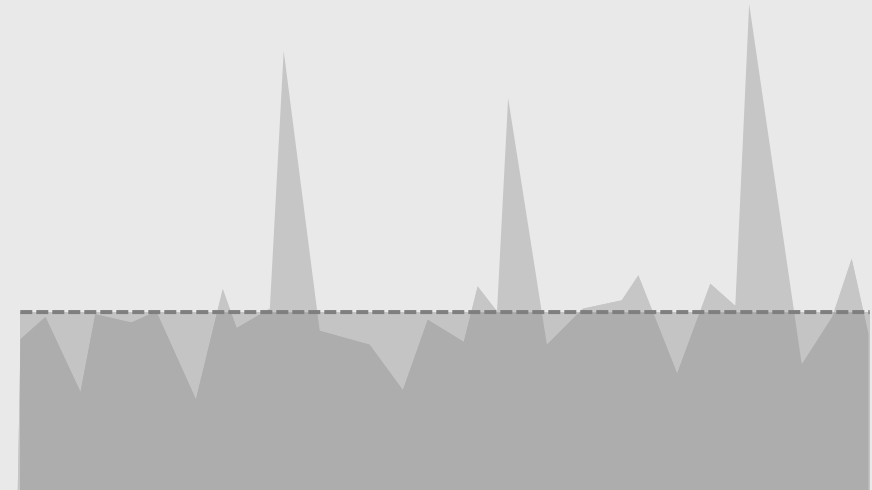
Usage spikes creates inconsistent ROI that is challenging to forecast



Updating HW is time consuming and expensive



Random, unpredictable spikes in demand for HPC can come from any new or existing application workflow



Demand for HPC infrastructure

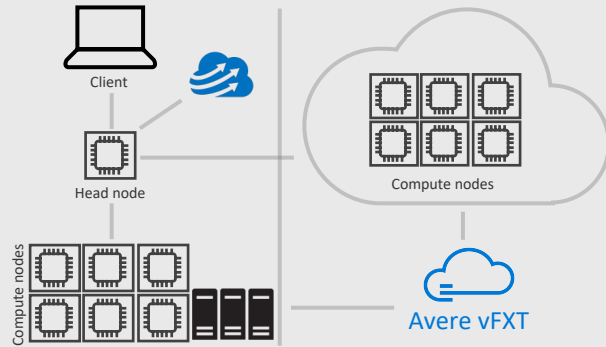


On-premises

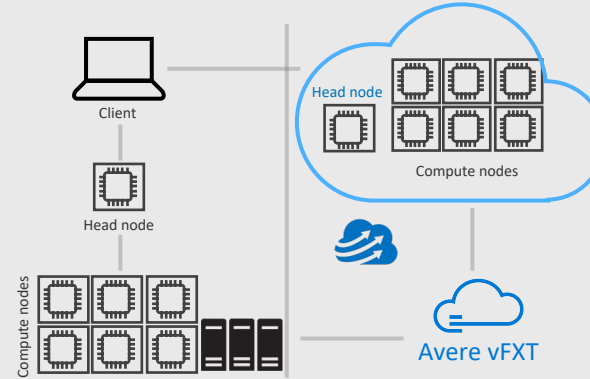
Hybrid Architectures

What do you mean by “Hybrid”?

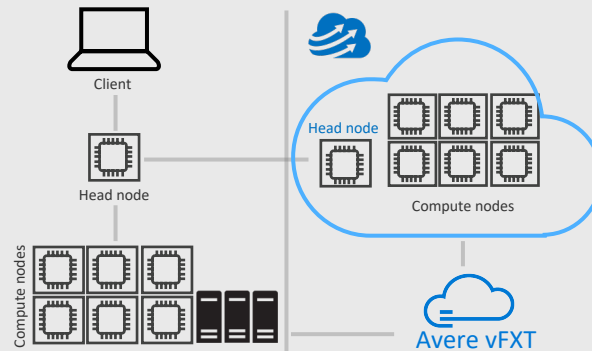
Burst



Hybrid

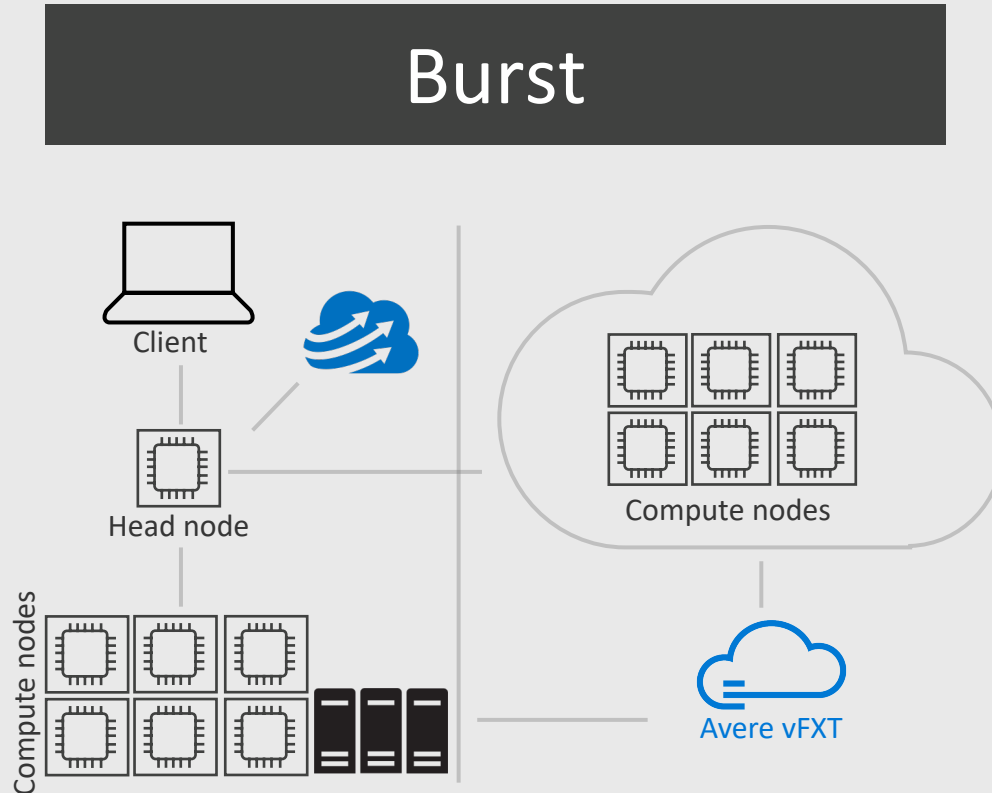


Scheduler Federation



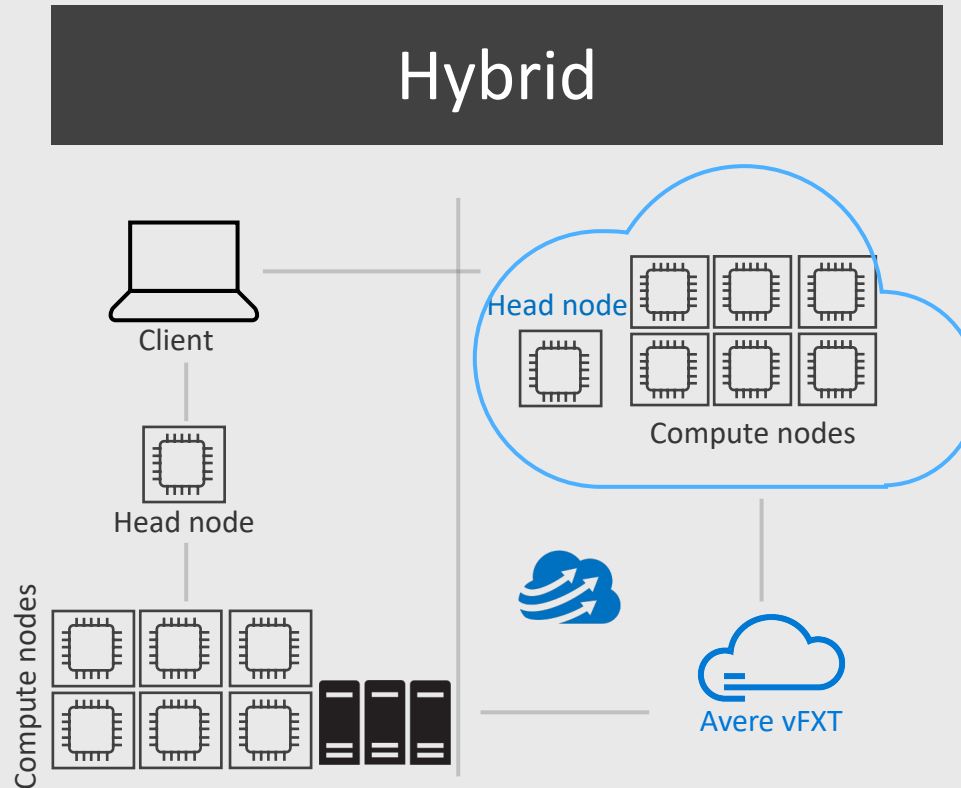
Hybrid HPC Architectures: Burst

1. What most customers think when they say “Hybrid”.
2. Need site-to-site VPN, subnet in Azure VNET is essentially an extension of the internal subnet.
3. DATA Syncing is crucial!!! JIT syncing at job run time is impractical.
4. Possible with almost all schedulers. LSF supports this natively.
 - Other schedulers require an installation of an autoscaling plugin in the internal environment.
 - Tricky to configure with CycleCloud.



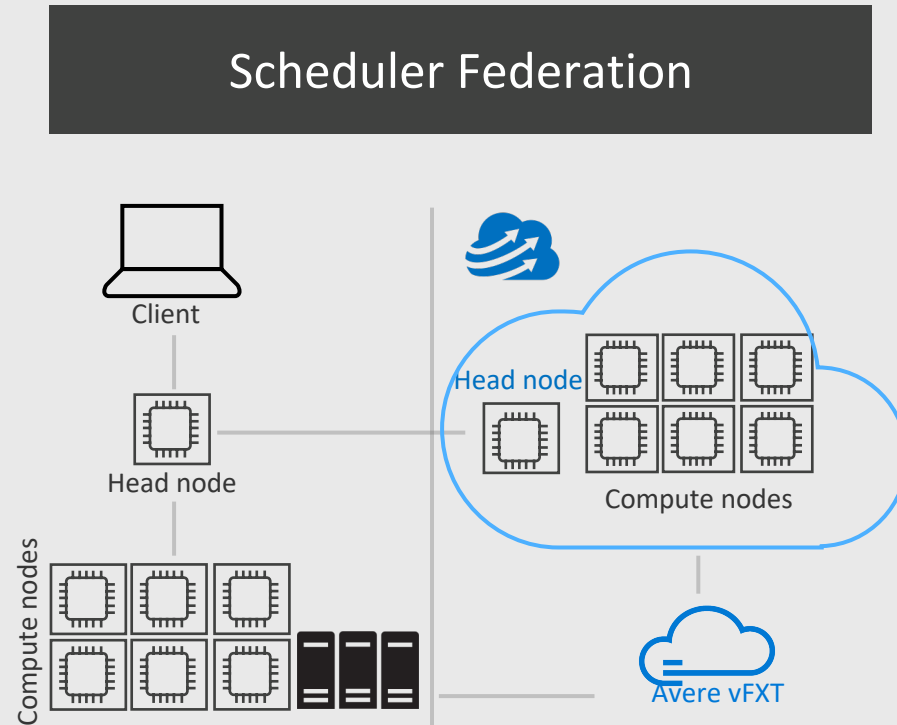
Hybrid HPC Architectures: “Hybrid”

1. Hybrid environment with 2 clusters
2. Trivial to implement! Just direct user to another IP address to submit jobs to.
 - If there's site-to-site VPN, there's no need for public internet access
3. DATA Syncing is crucial!!! JIT syncing at job run time is impractical.
4. Supported in ALL schedulers



Hybrid HPC Architectures: Scheduler Federation

1. The Scheduler software natively supports federation.
2. Scheduler in local cluster is aware of resources and jobs across all clusters in the federation.
3. The clusters coordinate with the "origin" cluster (cluster the job was submitted to) to schedule the job.
4. Trivial to implement if the scheduler supports it. Data sync is sometimes implemented *within* the scheduler's implementation. Otherwise data syncing is required.
5. Supported in SLURM and HTCONDOR



Hybrid Deployments

1. Coordinating compute resources with the scheduler is actually the easy part.

2. The practical implementation is *tedious*

- *How should the customer sync data, such that inputs are in all the clusters? Do outputs get synced back to on-prem too?*
- *Networking and firewalls are tricky to get around. InfoSec teams will need to be involved and takes a while to get everything approved.*
- *How is user management going to be done? It needs to be the same in the internal and external environment*
- *Extremely difficult to succeed in a POC.*

3. CycleCloud's roadmap is to leave the implementation details to the schedulers, and only manage the provisioning and orchestration

- *Provide an autoscaling API for schedulers to implement*
- *Autoscaling and provisioning decisions are the responsibility of the scheduler developer*

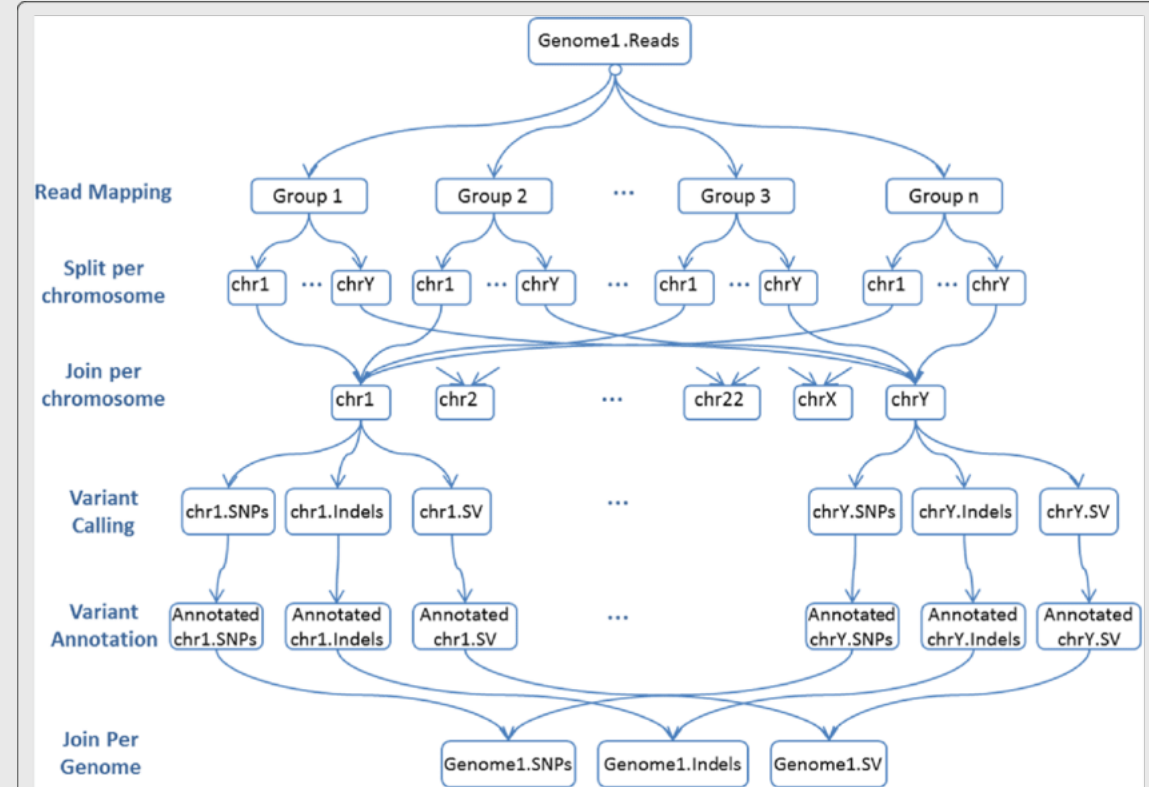
Workflow Management

1. Bioinformatics compute jobs, are typically not a simple, sequential chain:

JobA -> JobB -> JobC -> Finish

2. It usually looks more like a directed-acyclic-graph – jobs split into multiple downstream tasks, then converge in the end.

3. Workflow managers help manage a processing pipeline of job dependencies, and provides portability so that the jobs can be ran across different distributed systems



Workflow Management Frameworks

Examples of Common Frameworks

- Cromwell, Snakemake, NextFlow, TES (GA4GH)

Do these frameworks work on Azure?

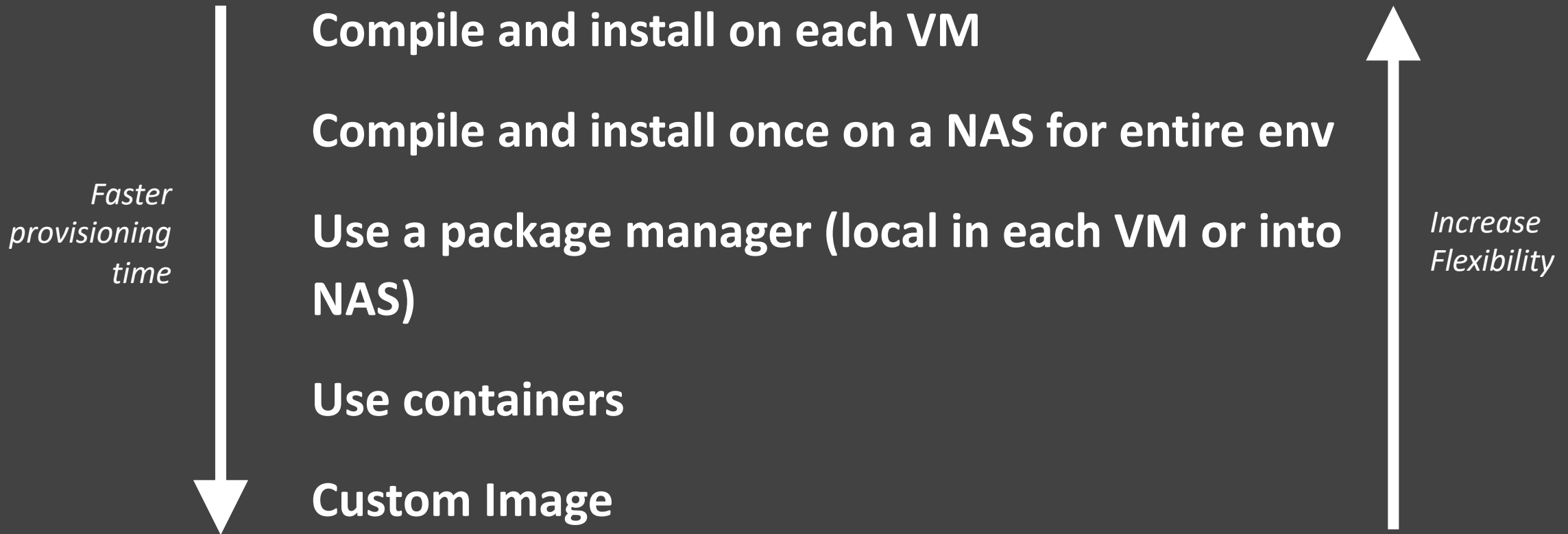
- Yes and No
- Ideally these frameworks work natively with Azure Batch and Blob Storage
 - This way, one can define a set of tasks and docker images, execute these via Batch and have persistent data stored in Blob.
- But aside from TES (allegedly), there's no native integration with Azure.

BUT there's another way:

- Because these frameworks are designed to be portable, they work with traditional HPC schedulers (SGE, PBS, Slurm, LSF etc)
- Start a cluster on Azure using CycleCloud, install the framework -> run job.

Application Management

How to you install applications on VMs for HPC environments?



Application Management

Compile and Install on each VM or into NAS

How to implement this?

- Create a script and run it during the boot-up phase of each node
- For NAS, do it during the boot-up phase of the Filer, or do it manually, once on the Filer.

Pros?

- Agility – can alter the script easily for each node.

Cons?

- SLOW! Adds to the boot-up time of each VM/node

Installing it into the NAS is a really good option though.

- Need a performant read FS for large environments (like an Avere vFXT)
- Often used in conjunction with Modules in Linux systems (<http://modules.sourceforge.net>)

```
$ ./configure
$ make
$ make install

# Stage into NAS
$ cp my_executable
  /shared/apps/bin/
```

Application Management

Use a Package Manager

- Makes compiling and installing scientific applications easy – removes the pain of having to manage dependences and version conflicts.
- Can be done on each VM using a script, or on a NAS
- The two common ones in HPC are Conda and Spack

<https://conda.io>

<https://spack.io>

Pros?

- Agility, very easy to manage

Cons?

- Can be slow but setting it up into the NAS is a really good option.

Application Management

Use a Package Manager

```
# Spack:
```

```
$ git clone https://github.com/spack/spack.git
```

```
$ . spack/share/spack/setup-env.sh
```

```
$ spack install hdf5
```

```
# Conda (Miniconda):
```

```
$ wget https://repo.anaconda.com/miniconda/Miniconda2-latest-Linux-x86_64.sh
```

```
$ bash Miniconda2-latest-Linux-x86_64.sh -b
```

```
$ source ~/miniconda2/etc/profile.d/conda.sh
```

```
$ conda install -y bowtie2 samtools bcftools htop glances nextflow
```

Application Management

Using Containers

Containers

- Essentially running applications in complete process isolation
- Utilizes from Linux namespace and control groups primitives
- Docker and Singularity are the most common technologies
 - Singularity was started in LLNL and really designed for running multi-user, untrusted applications on HPC clusters
- Vibrant communities create and publish images (e.g Docker Hub), makes it easy to distribute and share workloads
 - Replicability and reproducibility is big here
- Azure Container Registry provides users repositories to publish and deploy container images



Application Management

Using Containers

VM Image needs to have the container runtime

- You don't want to install the runtime at boot on each VM
- Use a prepared docker/singularity image

Pros?

- Don't have to manage applications. Just find a container image or create one.

Cons?

- Running and managing HPC jobs that run containers isn't that trivial

Use a workflow manager that has container support

```
# Docker example for bowtie2

$ docker pull biocontainers/bowtie2:latest

$ docker run -u $(id -u ${USER}):$(id -g ${USER})
-v /shared/data:/data -itd
biocontainers/bowtie2:latest bowtie2 -x
bowtie2_index -u read1,read2
```

Application Management

Use a custom image

How to implement this?

- Start a single VM with the desired OS. Compile and install the application(s)
- Capture the image of the VM
- Use new image in the HPC environment

Pros?

- Fastest way to go from provision to running job. Simplest to use in a job.

Cons?

- Very inflexible. Need to build custom image for every change or version update.

```
# install application and dependencies
$ ./configure
$ make
$ make install

# Capture Image
$ sudo waagent -deprovision+user.

# Capture image in Azure Portal or CLI
```

Application Management

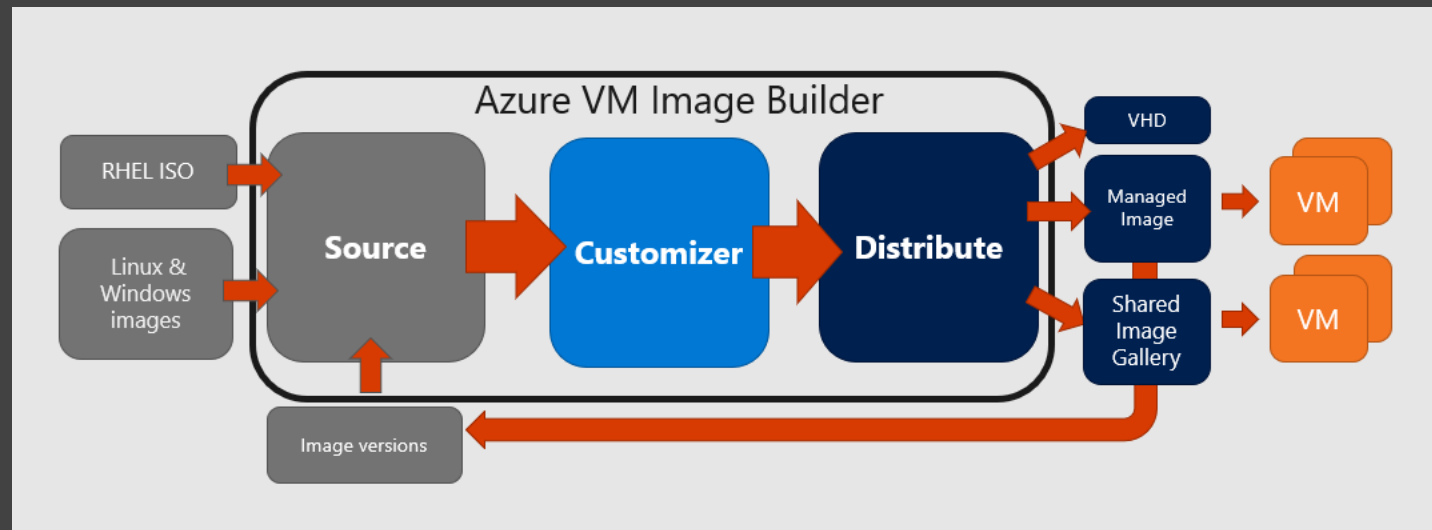
Use a custom image (automated)

Packer (<https://packer.io>)

- From Hashicorp, provides a templated way of automating image creation.

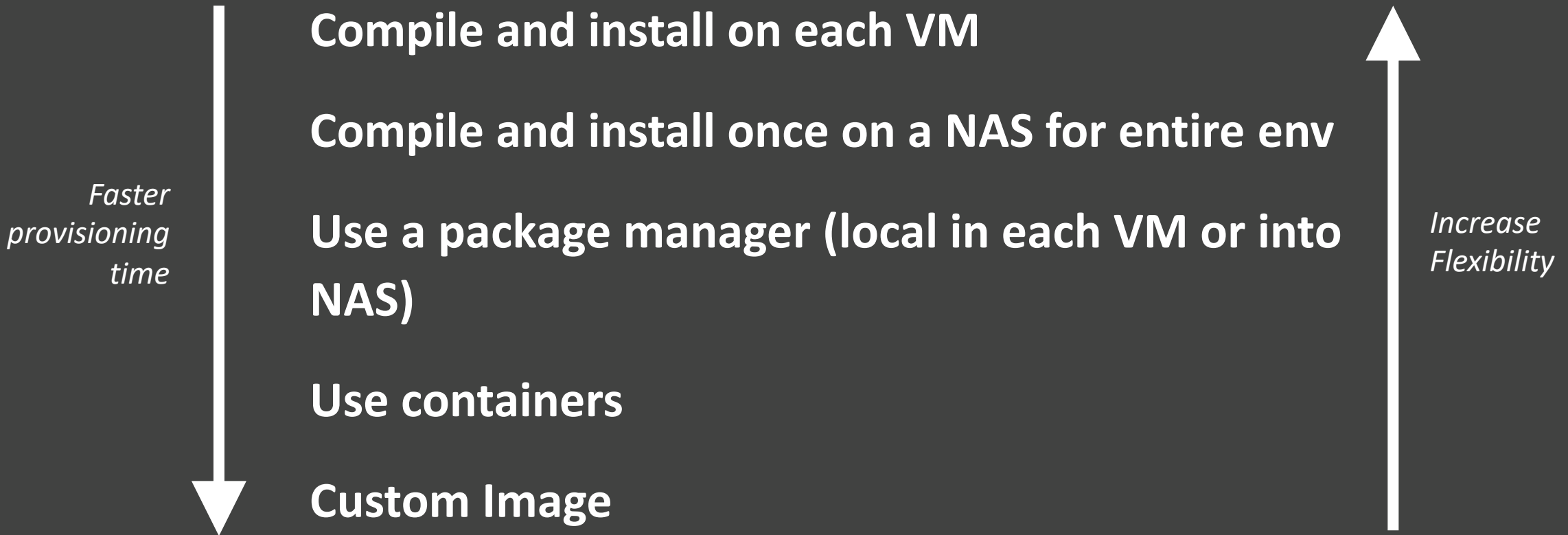
Azure Image Builder Service

- Essentially Packer as a Service on Azure
- <https://docs.microsoft.com/en-us/azure/virtual-machines/linux/image-builder-overview>



Application Management

How to you install applications on VMs for HPC environments?



User Management for Linux VMs on Azure

Azure Active Directory

- Integration via an VM extension.
- Essentially a PAM module that forwards the authentication request to the Azure portal.
- The interaction is similar to the az-cli login, or the Cloud Shell
- Can not be used in HPC jobs where there is no interactive login



Linux VM

Azure Active Directory Domain Service

- Essentially Active Directory Servers as a managed service
- Linux VM needs to be configured to join the AD domain, and configure the relevant modules
- Easiest path is to create a custom image from the configuration



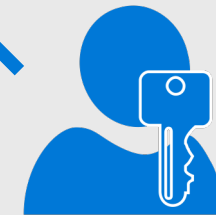
External or Customer Managed AD/LDAP server

- For example, connecting to an on-prem AD or LDAP server.
- Linux VM needs to be configured to join domain, configure PAM modules etc.
- Sometime Centrify is used.
- Create custom image from the config



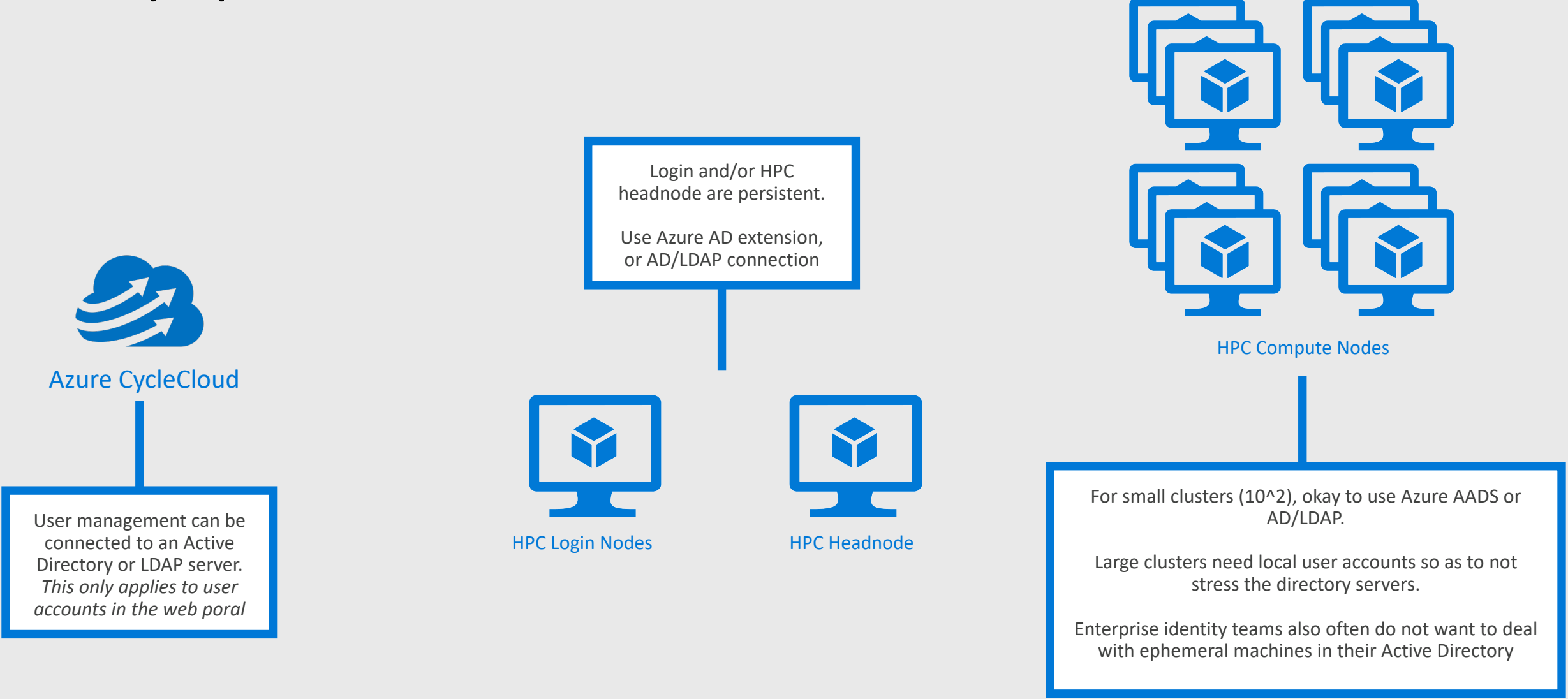
Local User Accounts on each VM

- Managed local on each VM.
- Possible to use scripts or cloud-init to create these users.
- Azure Batch supports this
- CycleCloud 7.8.0 provides a central mechanism to do this, using accounts from the CycleCloud UI.



User Management for Linux Clusters in Practice

Probably requires a mixture of services



HPC VMs on Azure

No-compromise CPU and GPU based resources



- Up to 16 cores, **Intel Xeon E5-2667 V3 processor**
- Up to 224 GB DDR4 memory, 14GB per core
- FDR InfiniBand @ 56 Gbps
- 2 TB local SSD



- Up to 44 cores, **Intel Xeon Platinum 8168 processor**
- 352 GB DDR4 memory, 8GB per core
- EDR InfiniBand @ 100 Gbps
- 700 GB local NVMe SSD



- 60 cores, **AMD EPYC 7551 processor**
- 240 GB DDR4 memory, 4GB per core
- EDR InfiniBand @ 100 Gbps
- 700 GB local NVMe SSD

H-Series:
Most powerful CPU
virtual machines
with InfiniBand

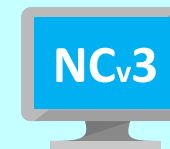
N-Series:
GPU virtual machines
specialized for graphic-
intensive workloads



- Up to 4 NVIDIA Tesla K80 GPUs
- Up to 24 cores
- Up to 224 GiB memory
- Up to 1440 GiB of local SSD
- FDR InfiniBand



- Up to 4 NVIDIA Pascal P100 GPUs
- Up to 24 cores
- Up to 448 GiB memory
- Up to 3 TB of local SSD
- FDR InfiniBand



- Up to 4 NVIDIA Volta V100 GPUs
- Up to 24 cores
- Up to 448 GiB memory
- Up to 3 TB of local SSD
- FDR InfiniBand



- Up to 4 NVIDIA Pascal P40 GPUs
- Up to 24 cores
- Up to 448 GiB memory
- Up to 3 TB of local SSD
- FDR InfiniBand



- Up to 4 NVIDIA Tesla M60 GPUs
- Up to 24 cores
- Up to 224 GiB memory
- Up to 1440 GiB of local SSD



- Up to 4 NVIDIA Tesla M60 GPUs
- Up to 24 cores
- Up to 448 GiB memory
- Up to 2,948 GiB of local SSD

High-Performance Computing VMs (H)

Powered by **AMD EPYC** and **Intel Xeon Platinum**



	HBv2*	HB	HC	H
CPU	AMD EPYC	AMD EPYC	Intel Xeon Platinum	Intel Xeon E5 v3
Cores/VM	120	60	44	16
Cores/cluster	80,000+	38,000+	28,000+	7,000+
Clock Speed**	2.8GHz	2.5 GHz	3.4 GHz	3.3 GHz
Memory Bandwidth	350 GB/sec	263 GB/sec	190 GB/sec	80 GB/sec
Memory	4GB/core 480 total	4 GB/core 240 total	8 GB/core 352 GB	14 GB/core 224 GB
Local Disk	1.6 TB NVMe	700 GB NVMe		2 TB SATA
Infiniband	200 Gb HDR	100 Gb EDR w/ ConnectX-5		56 Gb FDR
Network	40 Gb Ethernet			12.5 Gb Ethernet

In Preview today, Q4 2019 release

**All cores, non-AVX, peak Boost/Turbo frequencies

High-Performance Computing VMs (H)

Powered by **AMD EPYC** and **Intel Xeon Platinum**

	HB	HC	H
Targets	CFD, Seismic, Weather,	FEA, MD, Chemistry	Genomics, HPDA, Single-Threaded
Workload Driver	Memory Bandwidth	Raw Compute	Large RAM/core
Max MPI Job Size	18,000 cores	13,200 cores	3,200 cores*
MPI Support	All	All	Intel MPI 5.x
Azure Storage Support	Premium	Premium	Standard
OS Support for RDMA	CentOS/RHEL 7.5+ SLES 12 SP4+ WinServer 2016+	CentOS/RHEL 7.5+ SLES 12 SP4+ WinServer 2016+	CentOS/RHEL 6.x+ SLES 12+ WinServer 2012+










*Platform issues with H-v1 constrains MPI job scaling to ~512 cores with high efficiency

Sizing Azure VMs for the job

Steps to find the right VM:

1. Determine if 1 job for the HPC application runs on less than 1 machine or uses MPI across multiple machines
2. Determine the job's RAM per core usage (e.g. 6GB of RAM per core), and benchmark the instances that fit
3. If the workload requires Physical Cores, use the VM families in Purple
4. Check on availability of preferred machines
5. Cray is applicable for any workloads:
 - (a) Where a workable VM type isn't regionally available
 - (b) Which are high utilization, servers are used 80+%
 - (c) Where we have no working VM configuration

	Single VM Job						Multiple VM Job					
Job RAM Per Core	2	4	8	16	32	64	2	4	8	16	32	64
 AMD HB	Good for Memory bound						Good for Memory bound					
 Intel HC	Good for CPU bound						Good for CPU bound					
 Hv1	Custom Configured (i.e. no VM fit) Or High Utilization, Bare metal						Custom Configured (i.e. no VM fit) Or High Utilization, Bare metal					
 CRAY	Custom Configured (i.e. no VM fit) Or High Utilization, Bare metal						Custom Configured (i.e. no VM fit) Or High Utilization, Bare metal					
 F	Good for CPU Bound						Up to 3 VMs					
 D/E	Good for CPU Bound						Good for CPU Bound					
 Mv2	Good for CPU Bound						Good for CPU Bound					

Physical Cores
Hyper-threaded

Azure Storage



Disk Storage

Premium
Ultra SSD

Reliable, persistent, high performing storage for Virtual Machines



Object Storage

Azure Blobs

Secure, Scalable storage for unstructured data



File Storage

Azure NetApp Files
Cray ClusterStor
Azure Premium Files

Lift and shift of legacy applications that require file shares to the cloud

101010
010101
101010

Data Transport

Azure Import/Export
Azure DataBox

Move or migrate data into Azure



Hybrid Storage

Azure FXT Edge Filer (Preview)
Avere vFXT for Azure

Secure, intelligent data tiering between on-premises and cloud storage

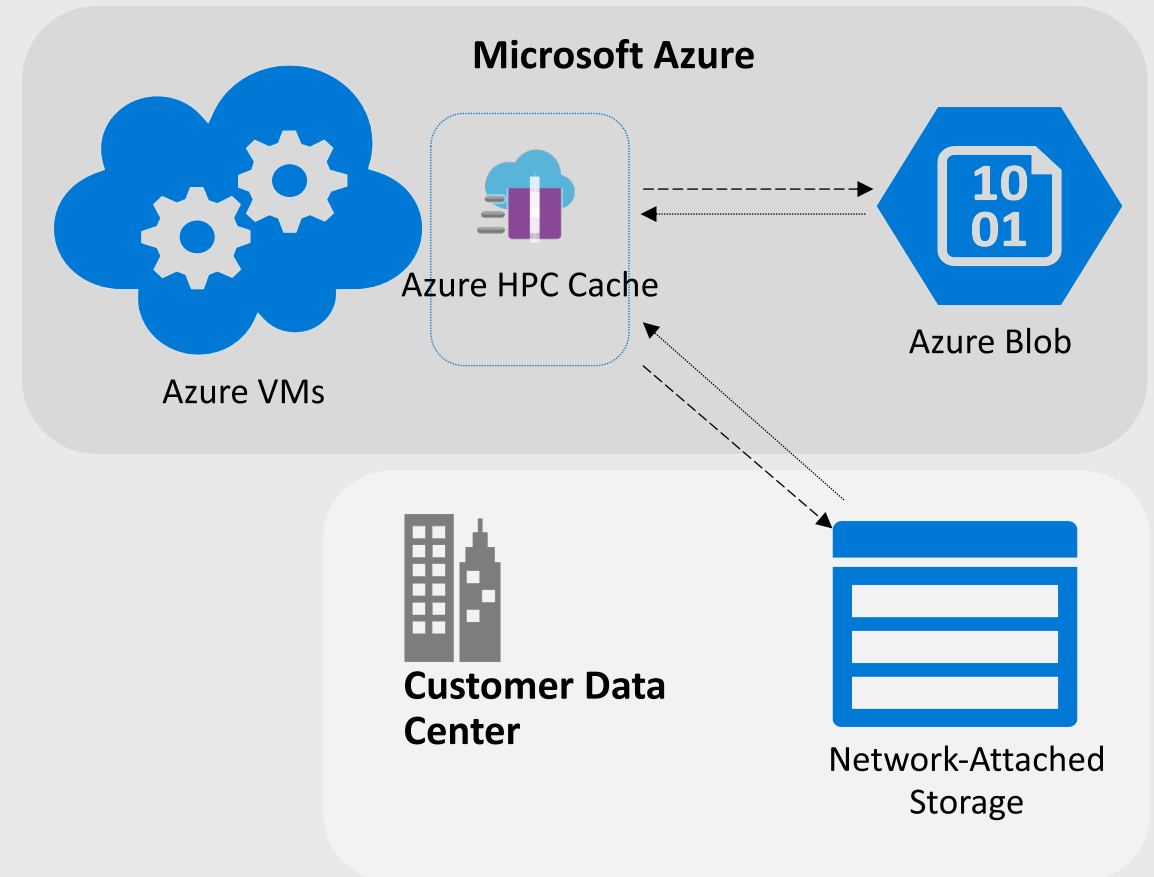
Azure HPC Cache

Storage Caching for High-Performance Computing

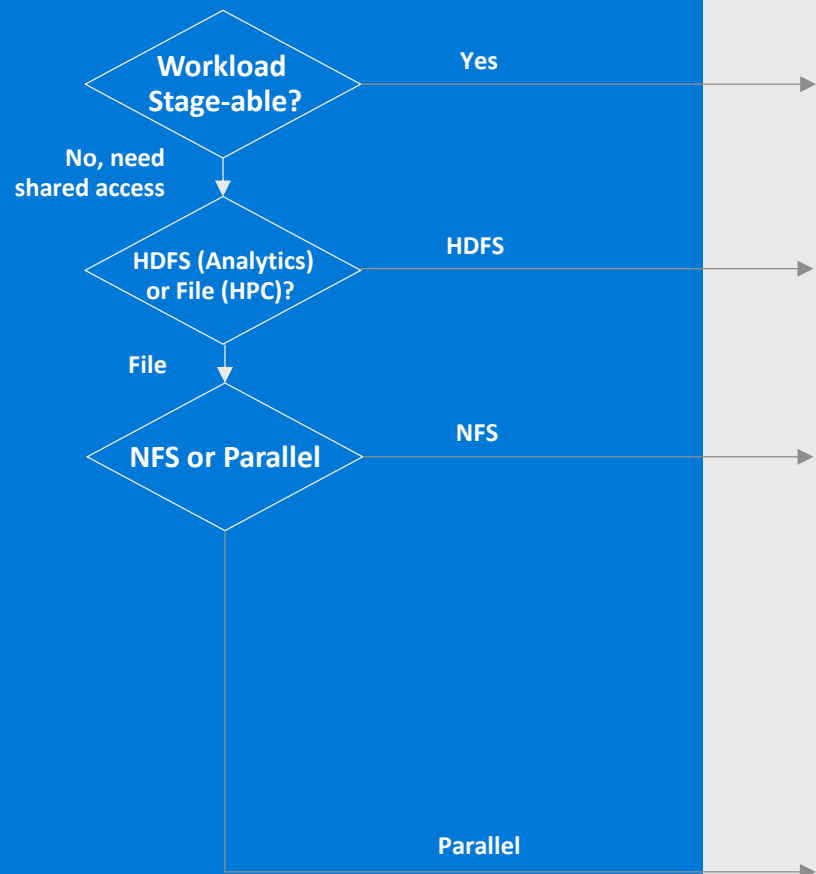
Better performance for computationally-intensive workloads

- **Performance.** Caching of hot data needed for demanding workloads, minimizing storage latency for heavy-read environments.
- **File Access.** File-based applications can access Azure compute resources from either on-premises network-attached storage (NFS) or Azure Blob (REST)
- **Productivity.** Easy to set-up and manage from the Azure portal.




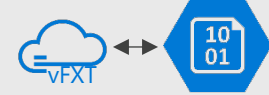



Suitable for many verticals including media & entertainment, life sciences, EDA, manufacturing, oil & gas, financial services and government applications.



File System Triage



Start at top and work down to find a HPC File System Solution

Solution:	Workload Fit:
 Stage Files in Blob to & from Local Disk *	Stage-able Workload: Pleasantly parallel, No multi-node or parallel I/O, no multi-user/ACL support
 Azure DataLake Storage Gen2	HDFS Or Analytics Workload: Pleasantly parallel, No multi-node or parallel I/O, no multi-user/ACL support
 NFS on VM using Lsv2, Premium, Ultra SSD *	Low Scale Workload: < 1.5 GiBps, < 19TB (Lsv2), 100TB (SSD) < 500 cores
 Avere vFXT for Azure (6 to 24 nodes)	Medium to Large Scale, Read-Heavy: <2 GiBps Write, <14 GiBps Read, <192TB Cache, < 2PB FS, < 50000 cores
 Azure NetApp Files	Medium Scale, Balanced or Write-heavy: < 6.5 GiBps, < 100TB/volume, < 4000 cores 12TB max file size
 Orchestrated Parallel FS* (option: Avere vFXT for Azure** for reads)	Large Scale: < 50GiBps Write, <500 TB < 50,000 cores
 Managed Cray ClusterStor Bare-metal HPC storage	Bare-metal, Large Scale: > 30GiBps Write, >500 TB > 50,000 cores

Increasing Scale/Perf Needs

* For scratch use Ephemeral Local Disks, for persistent use Premium Disks or Ultra SSD

Genomics Analysis Workflow

Primary Analysis

Secondary Analysis

Tertiary Analysis



Base Calling



Sequence Reads & Quality Scores



Filter Reads



Read alignment, sorting, duplicates



BAM File merging sorting



Variant Calling
Read quantification



Annotation, Curation, Classification, Interpretation



Clinical Reporting



What are the basepairs in the DNA Sequence?

Where do these DNA sequences come from?
What are the genetic variations?

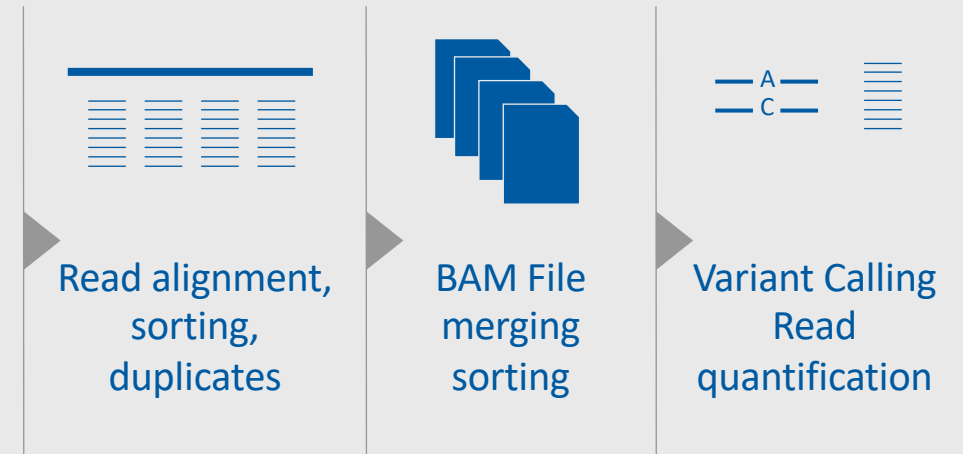
What does this DNA sample tell us?

Genomics Analysis Workflow

Secondary Analysis: Architecture Considerations

1. For every DNA sequence fragment, figure out where it lies in the genome or what gene it represents.
2. Processes are almost always memory bound:
 - Chose the VM SKU that provides the necessary memory/core
 - Memory/core requirement differs from application to application
3. IO load is directly proportional to the number of parallel tasks (or samples) being ran.
 - Each task needs to read a reference data (~4GB) into memory, stream input data, and write output in small blocks into a persistent store
 - Pick the correct storage solution based on the number of parallel tasks

Secondary Analysis

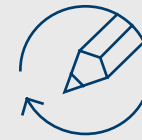


Genomics Analysis Workflow

Tertiary Analysis: Architecture Considerations

1. A catch-all stage for a very varied set of downstream analysis
 - Primarily data-science or machine-learning driven
2. Azure's platform services shine here:
 - Databricks if the customer is using Spark
 - Azure ML if the customer is doing training or inference
 - Azure Notebooks for data exploration
3. Azure Blob with ADLSgen2 for HDFS protocol is useful

Tertiary Analysis



Annotation,
Curation,
Classification,
Interpretation



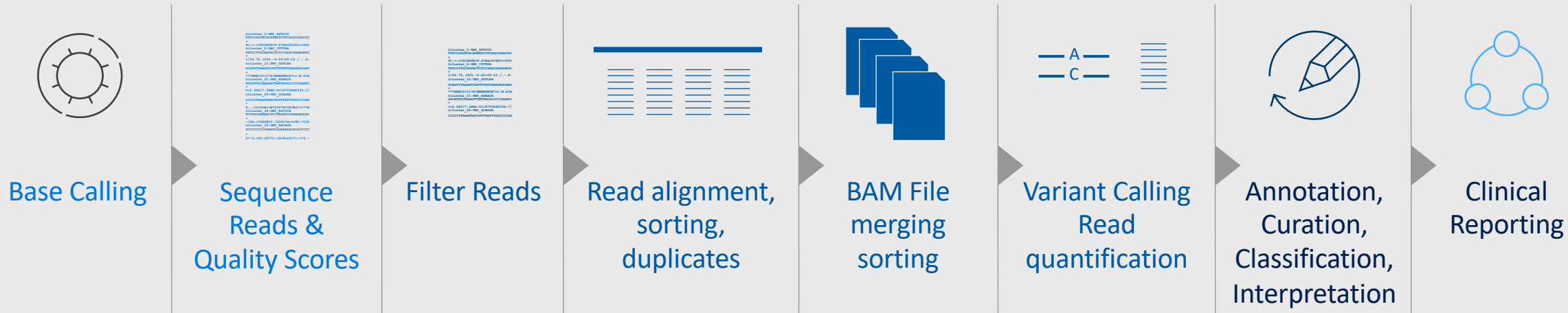
Clinical
Reporting

Genomics Analysis Workflow

Primary Analysis

Secondary Analysis

Tertiary Analysis



Azure compute mapping

HB-Series 4:1 Mem to Core, AMD EPYC

HC-Series 8:1 Mem to Core, Intel SkyLake

F-Series 2:1 Mem to Core, up to 12 Gbps bandwidth

H1-Series: 14 Mem to Core, Intel Haswells

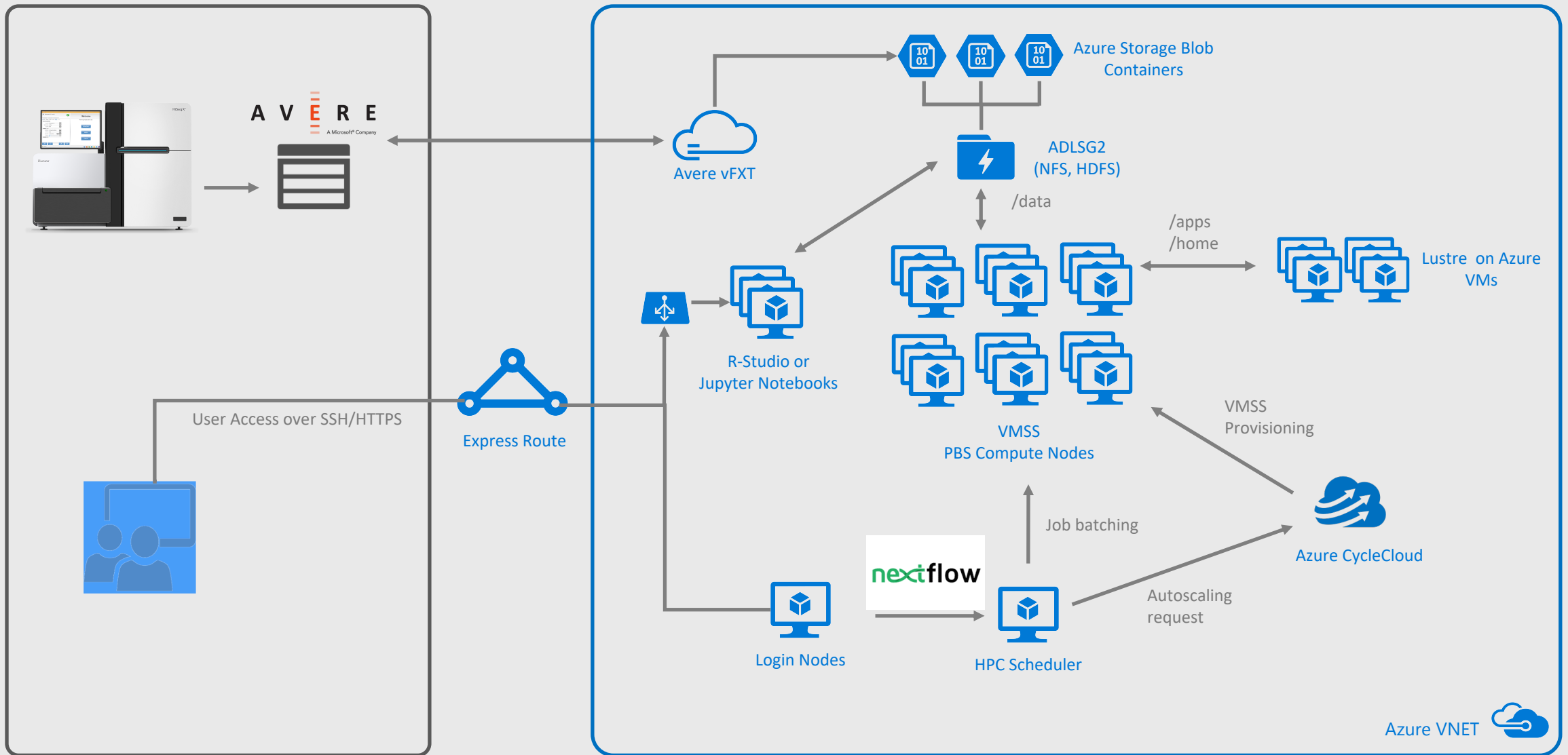
HB-Series 4:1 Mem to Core, AMD EPYC

M-Series up to 28:1 Mem to Core, up to 4 TiB, up to 30 Gbps

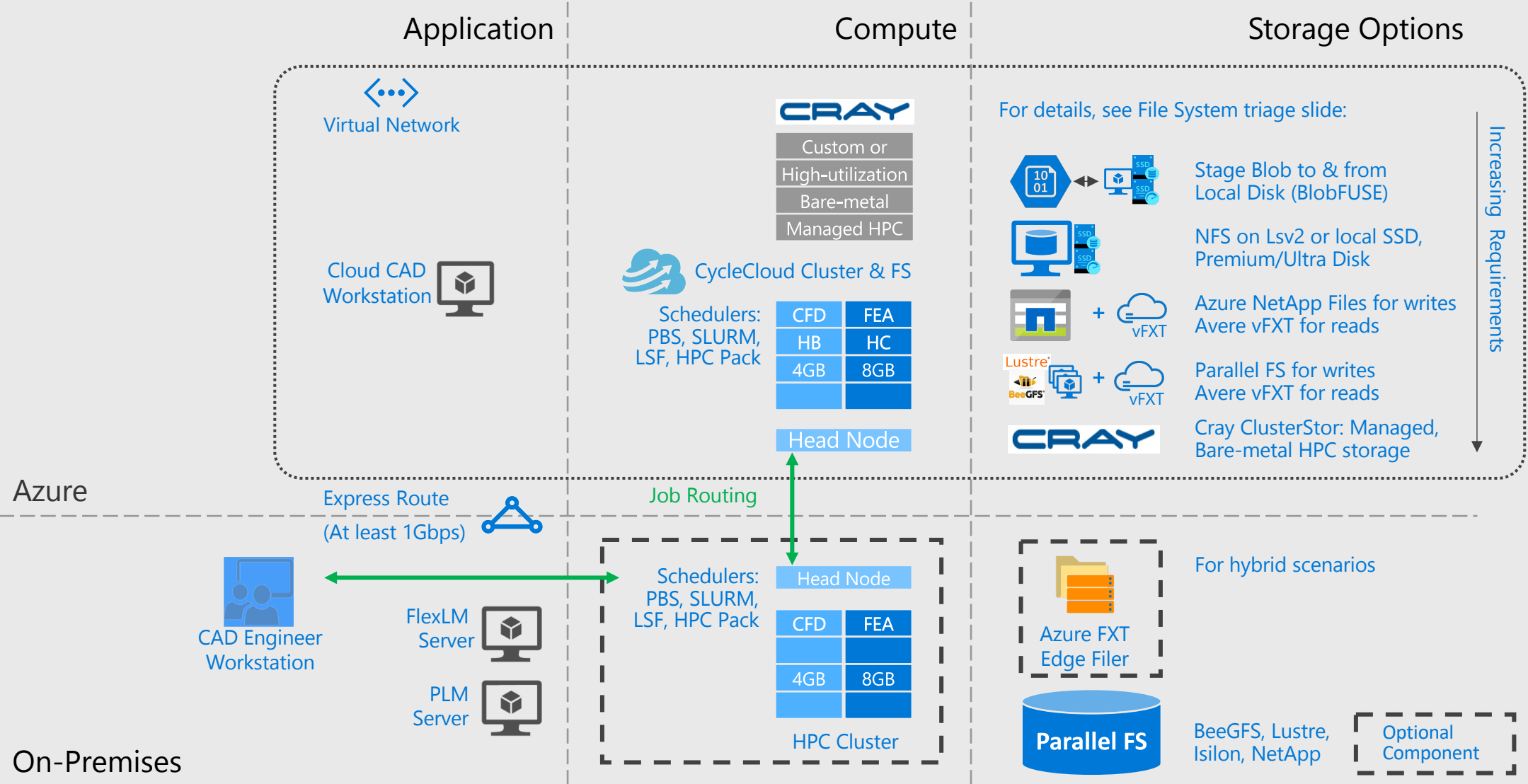
NC/ND-Series: GPUs

High Performance Resource Management and Provisioning – Azure CycleCloud, VM Scale Sets, Azure Batch

Genomics Architecture – How it comes together



CFD & FEA Architecture



Additional Resources

Azure CycleCloud: <https://docs.microsoft.com/en-us/azure/cyclecloud/quickstart-install-cyclecloud>

Azure Batch: <https://docs.microsoft.com/en-us/azure/batch/>

Azure HPC Cache: <https://azure.microsoft.com/en-us/services/hpc-cache/>

HPC for the Enterprise Architect

Microsoft in Higher Education 2019

Andy Howard

HPC Software & Services
anhoward@microsoft.com