Building Blocks in the Cloud:

Scaling LEGO Engineering with AWS High-Performance Computing

Brian Skjerven (He/Him)
Sr. Specialist Solutions Architect, HPC
AWS

Matt Vaughn (He/Him)
Principal Developer Advocate, HPC
AWS

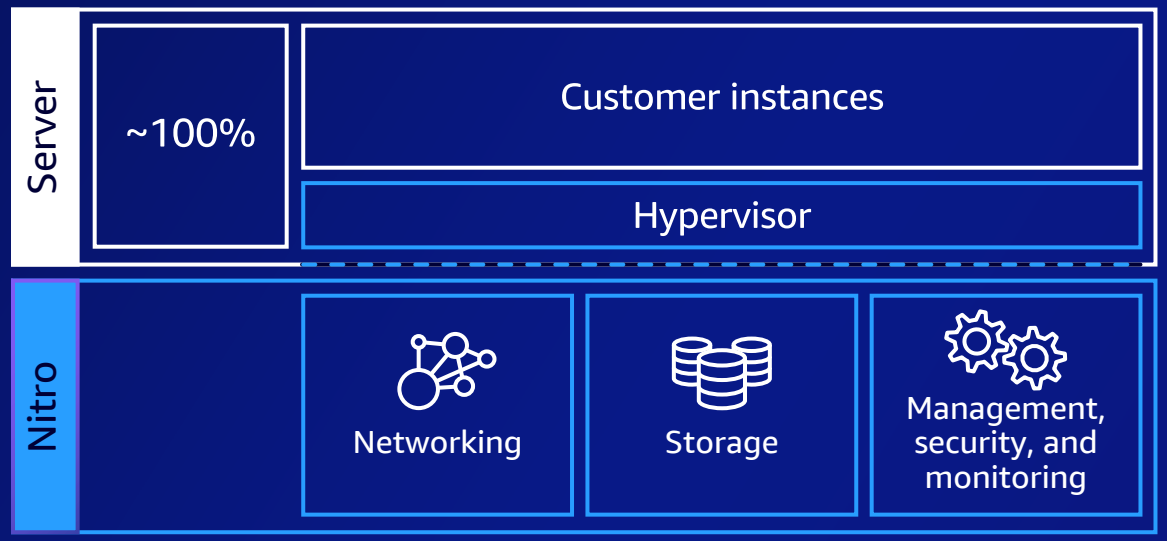
HPC on AWS

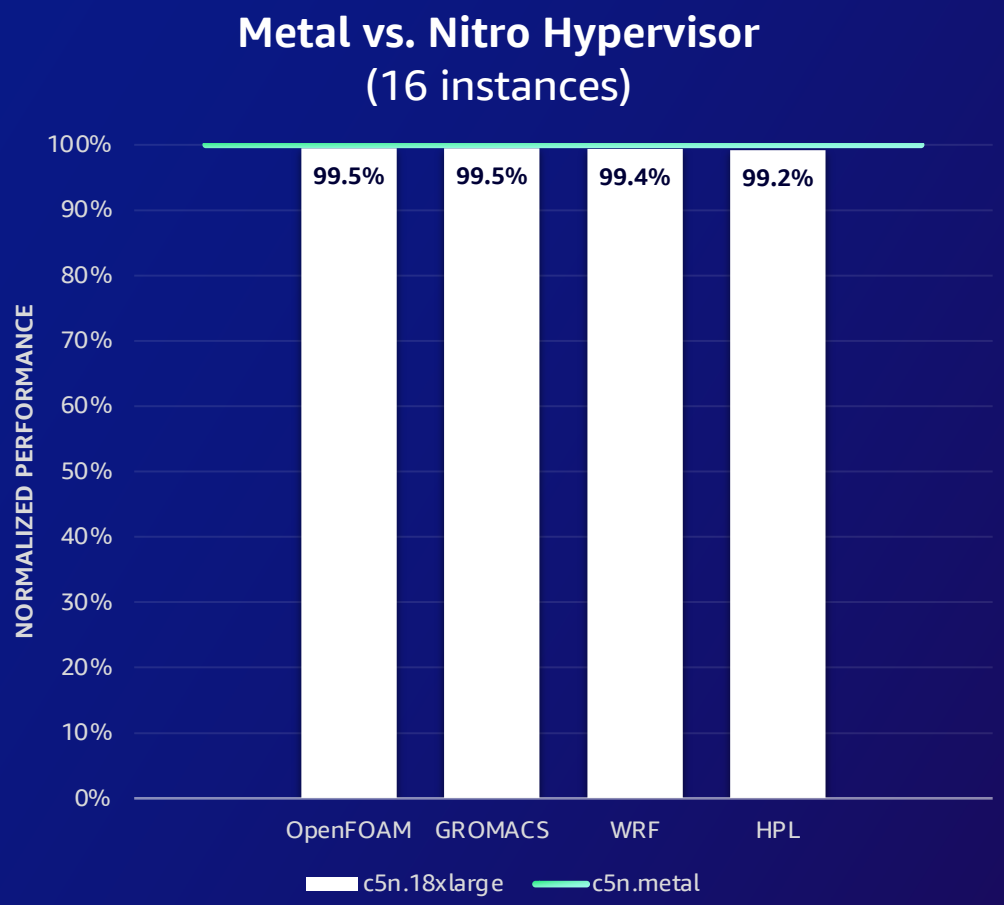
Performance at scale	AWS Nitro System	Amazon EC2	Elastic Fabric Adapter	Amazon FSx
Access and job management	AWS Batch	AWS ParallelCluster	NICE DCV	

The AWS Nitro System

The Nitro System lightweight hypervisor memory and CPU allocation are designed for **performance nearly indistinguishable from bare metal**

Designed using a security chip that monitors, protects, and verifies the instance hardware and firmware





Amazon EC2 | The compute platform for every workload

Workload types


Machine Learning


High-Performance Computing


Media Rendering


Containers


Web-based Apps


Batch Processing


Big Data

Instance types for HPC workloads

HPC Optimized

Hpc7g | Hpc7a | Hpc6id

Compute, Memory, and Networking

C7gn | C7g | C6i | M6i

R6i | X2iezn | C5n | M5zn

Accelerators

P4d | G5 | DL1 | Trn1

F1 | Inf1 | VT1

Scale tightly and loosely-coupled HPC applications

- Choice of processor (e.g., Graviton, Intel, AMD)
- Scale tightly-coupled HPC and ML workloads
- Up to 400 Gbps network bandwidth
- < 15 micro-seconds network latencies
- Accelerators use hardware to perform functions more efficiently than is possible in software running in CPUs

Elastic Fabric Adapter (EFA)

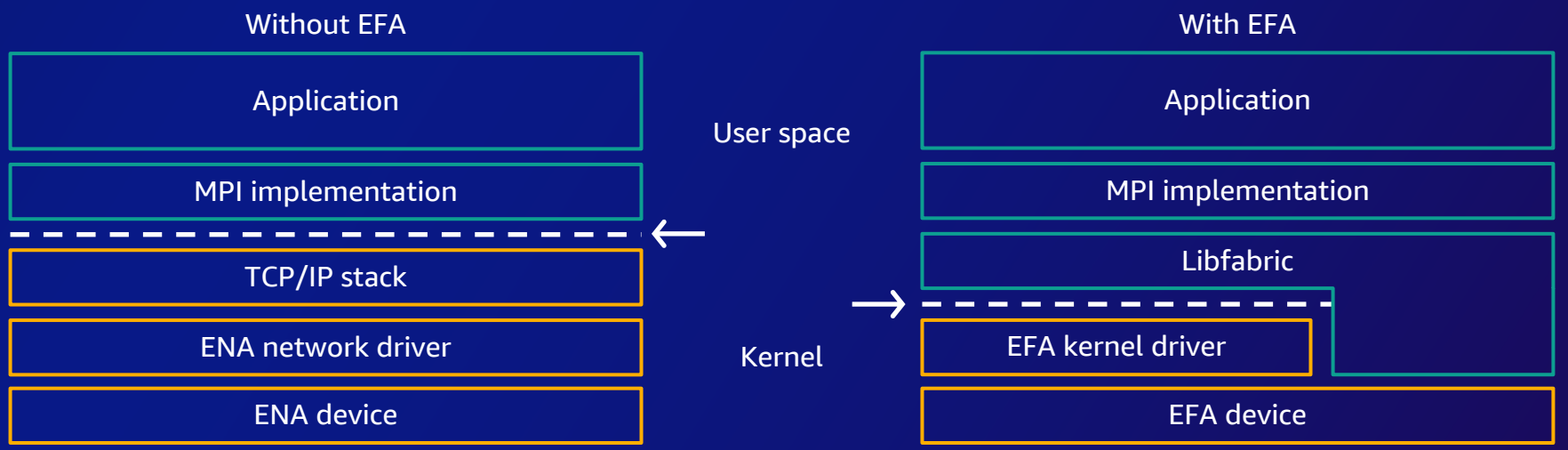
SRD protocol

Proving myths about latency constraints wrong

CFD

Seismic

Weather modeling

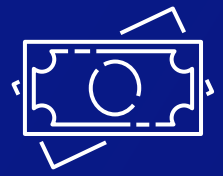


Amazon FSx for Lustre

FULLY MANAGED SHARED STORAGE BUILT ON THE WORLD'S MOST POPULAR HIGH-PERFORMANCE FILE SYSTEM

Sub-ms latencies, **hundreds of GB/s of throughput,** millions of IOPS

Concurrent access for thousands of instances and **100,000s of cores**



Cost-optimized file systems with HDD and SSD storage options

Flexible deployment options for short- and longer-term workloads

Learn more: Amazon FSx for Lustre, https://aws.amazon.com/fsx/lustre/

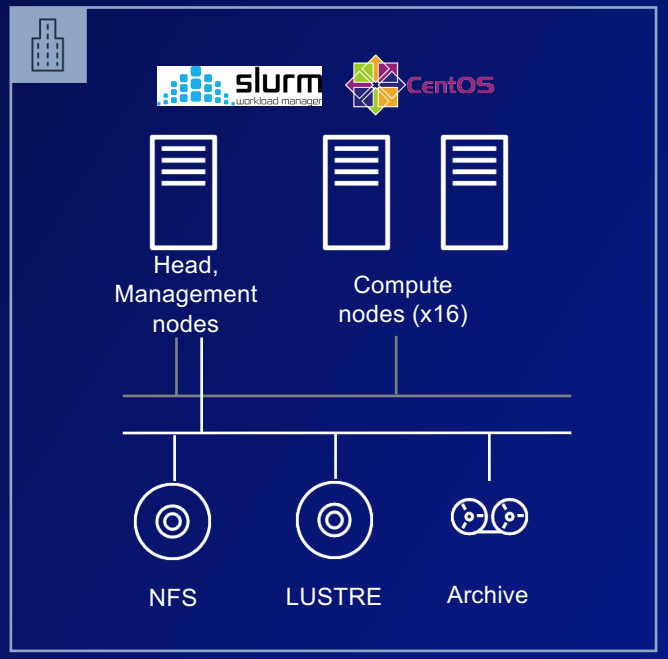
AWS ParallelCluster

One-stop shop to set up your HPC cluster

Integrated with AWS services you need

- Highly-performant file systems
- Amazon EC2 instances
- EFA
- NICE DCV

ON-PREMISES



PARALLELCLUSTER RECIPE (YAML)

```
Image:
  Os: centos7

HeadNode:
  InstanceType: c5.4xlarge
  Networking:
    SubnetId: subnet-b46032ec
  Ssh: KeyName: My_PC3_KeyPair
  AllowedIps: 0.0.0.0/0

Scheduling:
  Scheduler: slurm
  SlurmSettings:
    ScaledownIdletime: 10
    Dns:
      DisableManagedDns: true
  SlurmQueues:
    - Name: q1_ondemand
      ComputeSettings:
        LocalStorage:
          RootVolume:
            Size: 100
        CapacityType: ONDEMAND
        ComputeResources:
          - Name: compute-resource-1
            InstanceType: c5.n18xlarge
            Efa:
              Enabled: true
            MinCount: 0
            MaxCount: 64
            Networking:
              SubnetIds:
              - subnet-a12321bc
              PlacementGroup:
                Enabled:true

SharedStorage:
  - MountDir: /shared
    Name: myebs
    StorageType: Ebs
    EbsSettings:
      VolumeType: gp3
      Size: 100
  - MountDir: /lustre
    Name: myfsx
    StorageType: FsxLustre
    FsxLustreSettings:
      StorageCapacity: 1200
      DeploymentType: SCRATCH_2
      ImportPath: s3://myhpcbucket
```

OS

Head node

Scheduler Settings

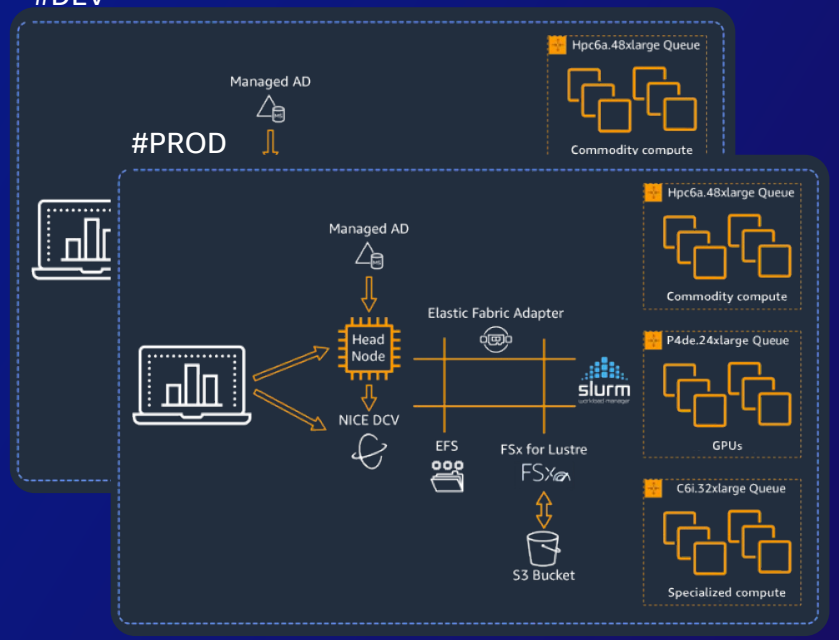
Compute nodes

Network settings

NFS Storage

Lustre storage

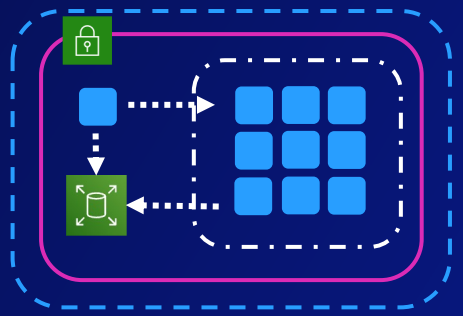
#DEV



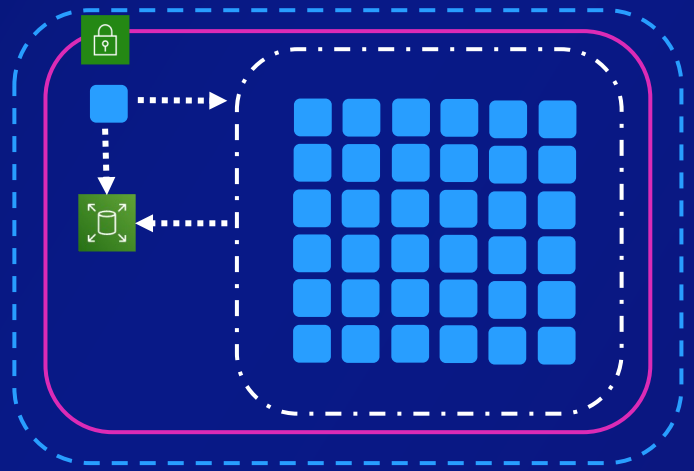
Recipe becomes the manageable asset.

Automatic resource scaling

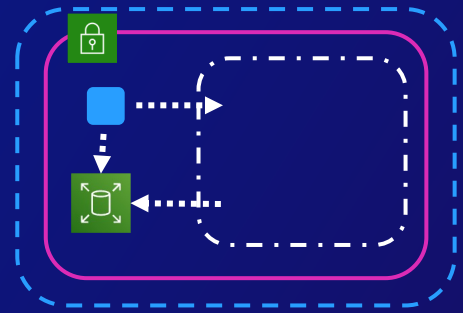
Allocate instance
Run jobs

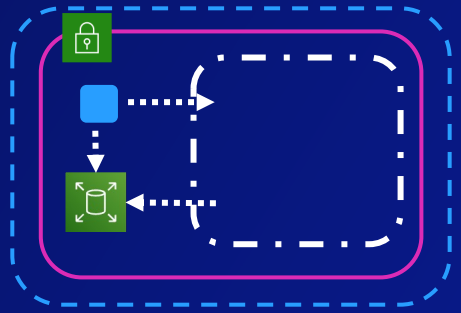


Scale up when jobs
are waiting



Scale down when
the cluster is idle





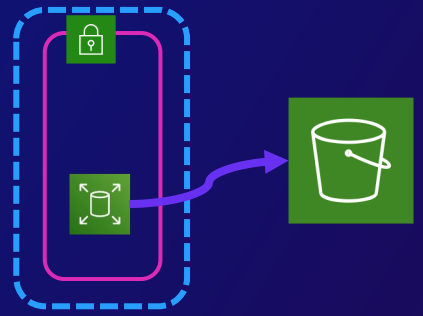
Cluster created

No compute nodes allocated

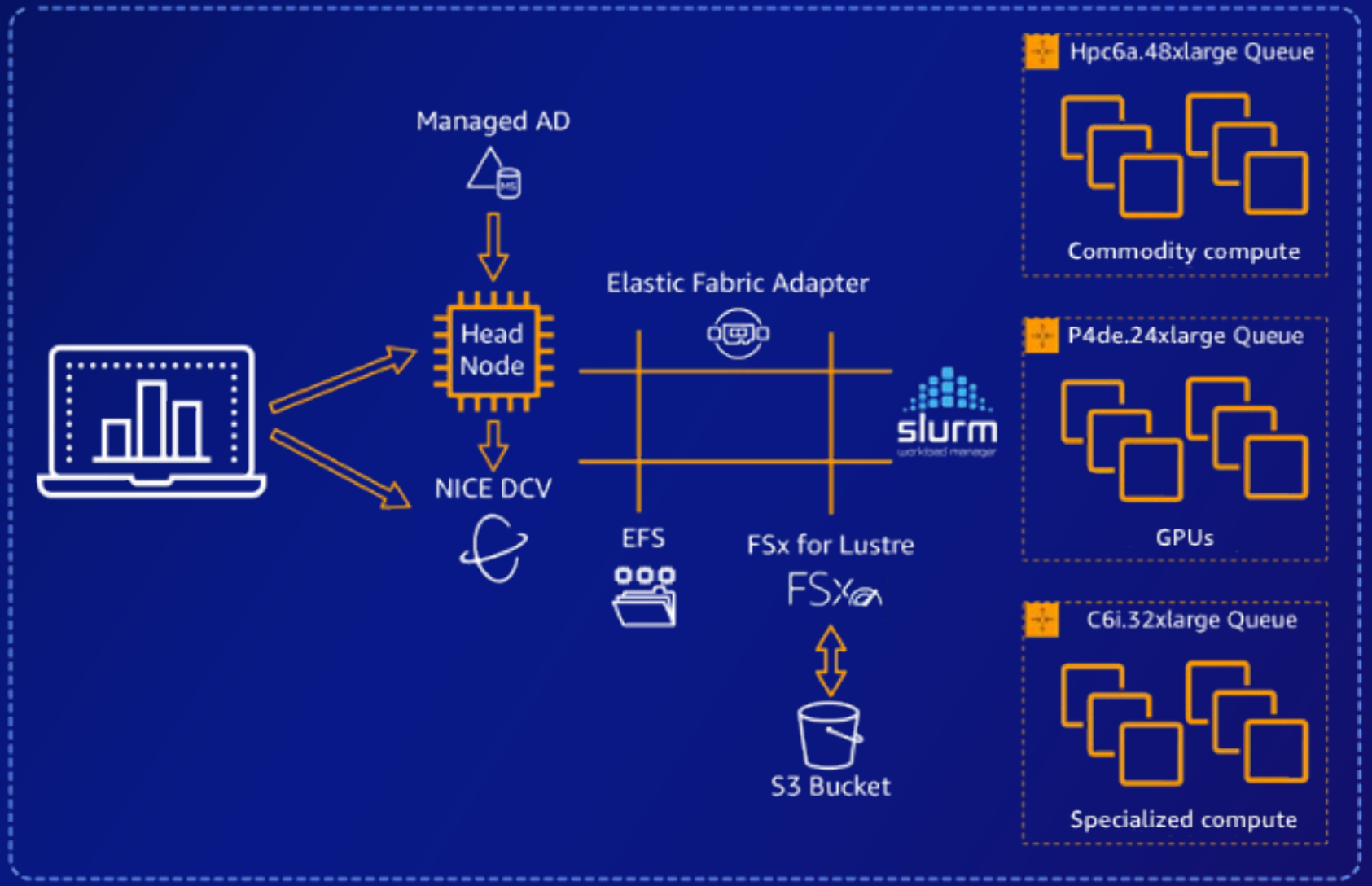
Waiting for jobs …

Optional

Delete the cluster after syncing data to persistent storage



Typical deployment



Formula 1 redesigns car for closer racing and more exciting fan experience

CUSTOMER PROFILE

Formula1 runs its Computational Fluid Dynamics (CFD) platform on AWS HPC

- Reduces downforce loss in wheel-to-wheel racing from 50 percent to 15 percent reducing the impact of the car in front of it
- Better supports strategic priorities for increasing competitiveness and unpredictability on the track
- Cars can drive closer to one another and overtake more easily creating a world-class spectacle for fans

Lowered the cost of CFD simulations by 30%

Reduced CFD simulation time by 80%

Arm accelerates speed to market by migrating EDA workflows to AWS

CHALLENGE

Arm wanted to modernize its offerings for intellectual property design because its on-premises infrastructure could not grow with the pace of its engineering requirements.

SOLUTION

Arm uses AWS Batch and Amazon EC2 Spot Instances to optimize its compute—decreasing the turnaround time for verification jobs, increasing engineer productivity, and accelerating product speed to market.

OUTCOME

- ✓ Can run more than 53 million jobs per week
- ✓ Scaled up to 400,000 virtual CPUs
- ✓ Decreased turnaround time for verification jobs

KEY SERVICE(S): AWS Batch, Amazon EC2, AWS ParallelCluster

Accelerating R&D | 42 days from Sequence to Clinical Batch

- **Challenge**
- When designing mRNA based therapies there can be multiple mRNA sequences and structures that can be produced. Synthesizing and testing each option to determine what is most stable and easiest to chemically develop is costly and time consuming.

- **Solution**
- Moderna utilizes machine learning models to help predict the best mRNA structures for production. The company has achieved rapid learning and insight from vast amounts of data and ever-improving rule sets based on accumulated learning.

- **Benefits**
- Moderna delivered the first clinical batch of its COVID-19 vaccine only 42 days after the sequence of the virus was released with AWS as their preferred cloud provider. Improved turn-around time, increased mRNA quality, and decreased costs.

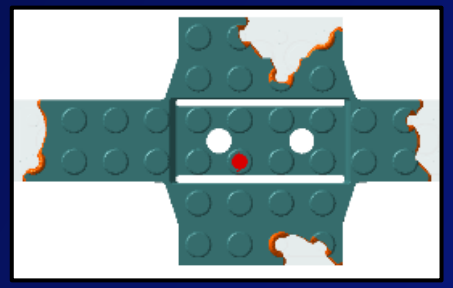
> "Utilizing a neural network, we can predict whether the mRNA sequence will be more or less difficult to produce, and suggest to the scientist changes that help improve the outcome.
>
> Marcello Damiani, Chief Digital and Operational Excellence Officer

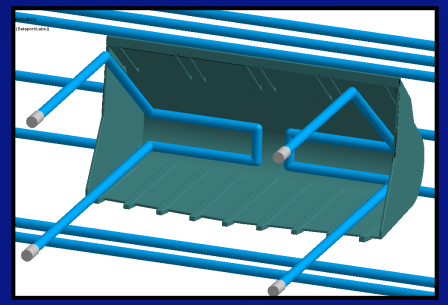
moderna

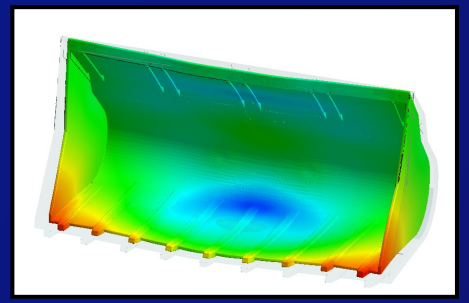
- Company: Moderna Therapeutics
- Country: US
- Employees: 550
- Website: ModernaTX.com

- **About Moderna Therapeutics**
  Moderna Therapeutics, which is based in Cambridge, Massachusetts and employs about 550 people, was founded to deliver on the promise of messenger RNA (mRNA) science to create novel medicines for unmet patient needs.

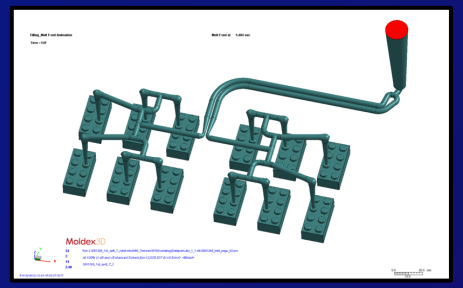
Engineering @ LEGO

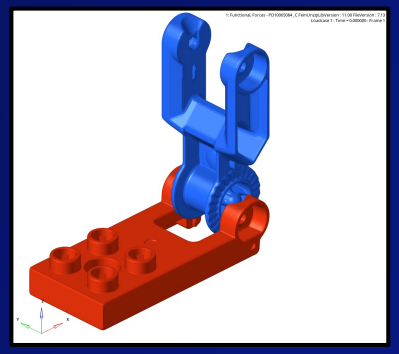


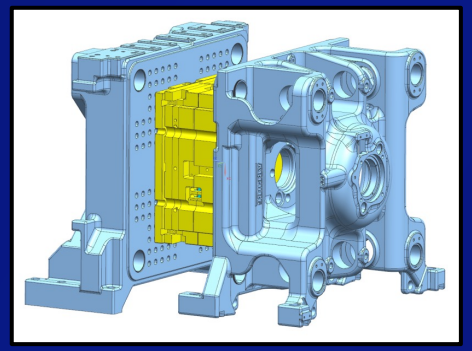


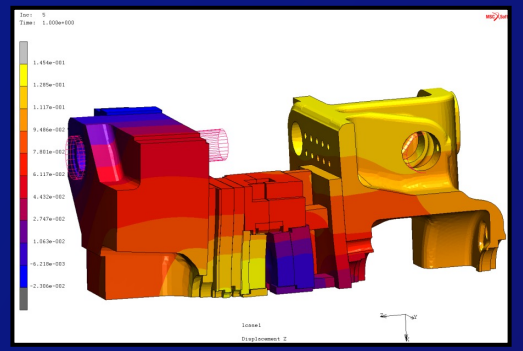


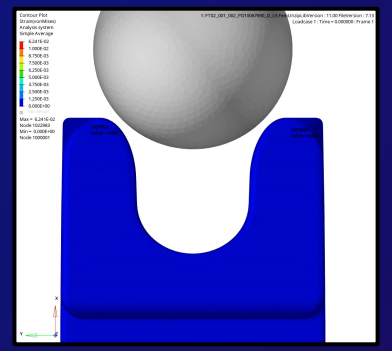
- 17 Engineers
- Injection moulding and structural simulations – 70%/30%
  - Filling, Inlet balancing, Warpage, etc.
  - Product safety and functional simulation
  - Mould parts and assemblies
- Material characterization - Model calibration using material test data
- 3700 simulations ordered in 2022



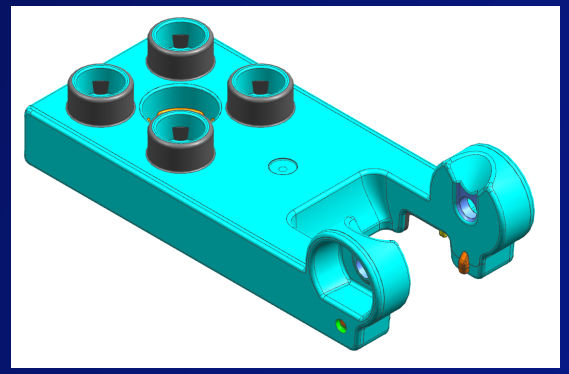




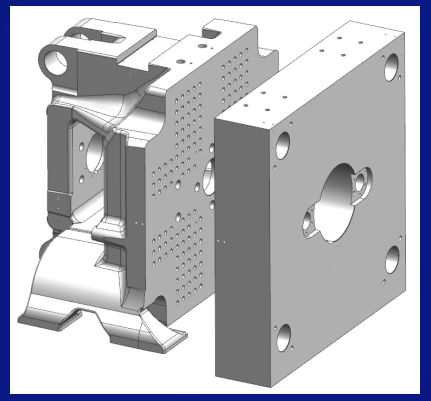




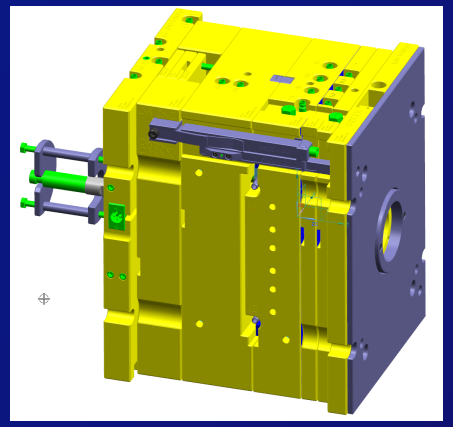
Simulation overview



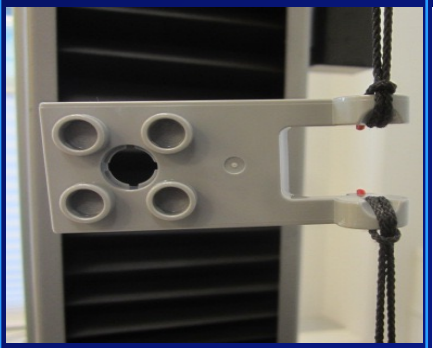
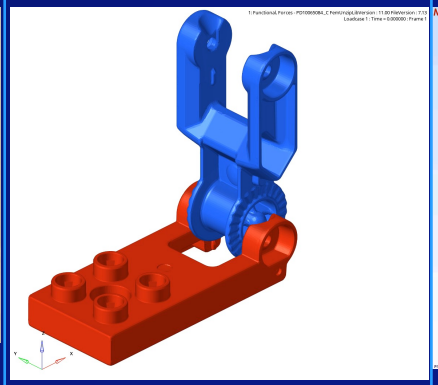
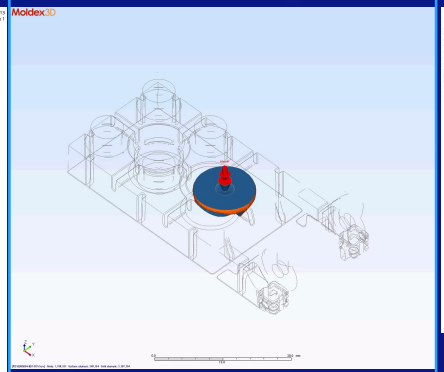
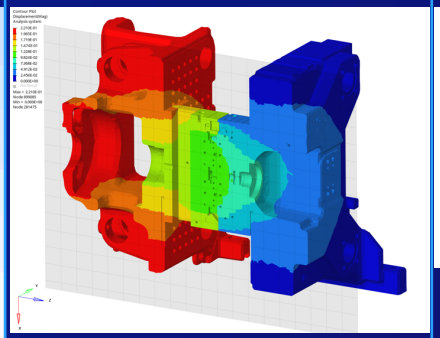
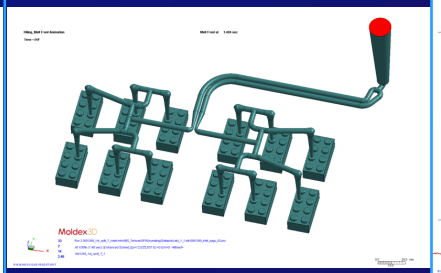
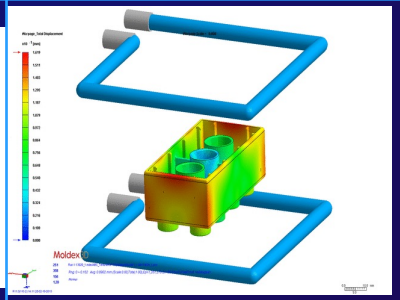
Element



Injection moulding machine

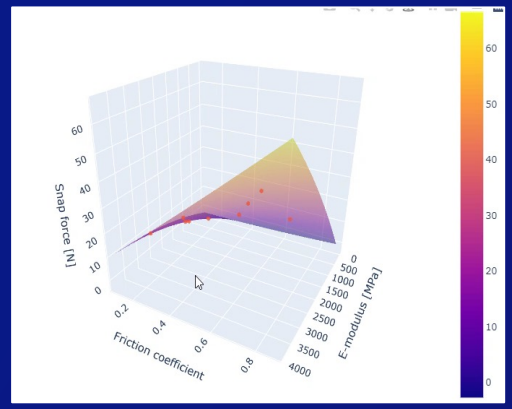


Mould

Product Safety	Function	Cavity Filling	Deflection/Life	Inlet balancing	Warpage
					

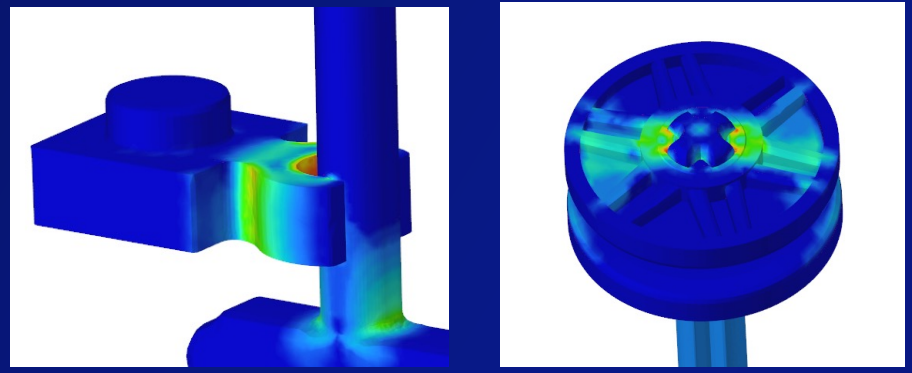
Material connector DOE

Selection of most common connectors



Parametric study to mathematically describe relationship between E-modulus & CoF for each individual connector

Virtual prototype for each individual connector (computer simulation) – 9 different connector types – 37 material combination each = 333 simulations

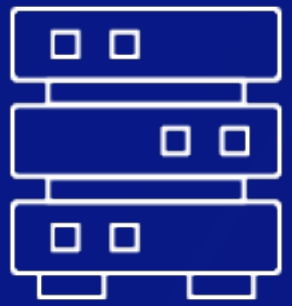


Material Feasibility Tool for Connector Function

Motivations

Consolidate IT

Reduce hardware maintenance



Flexible compute options

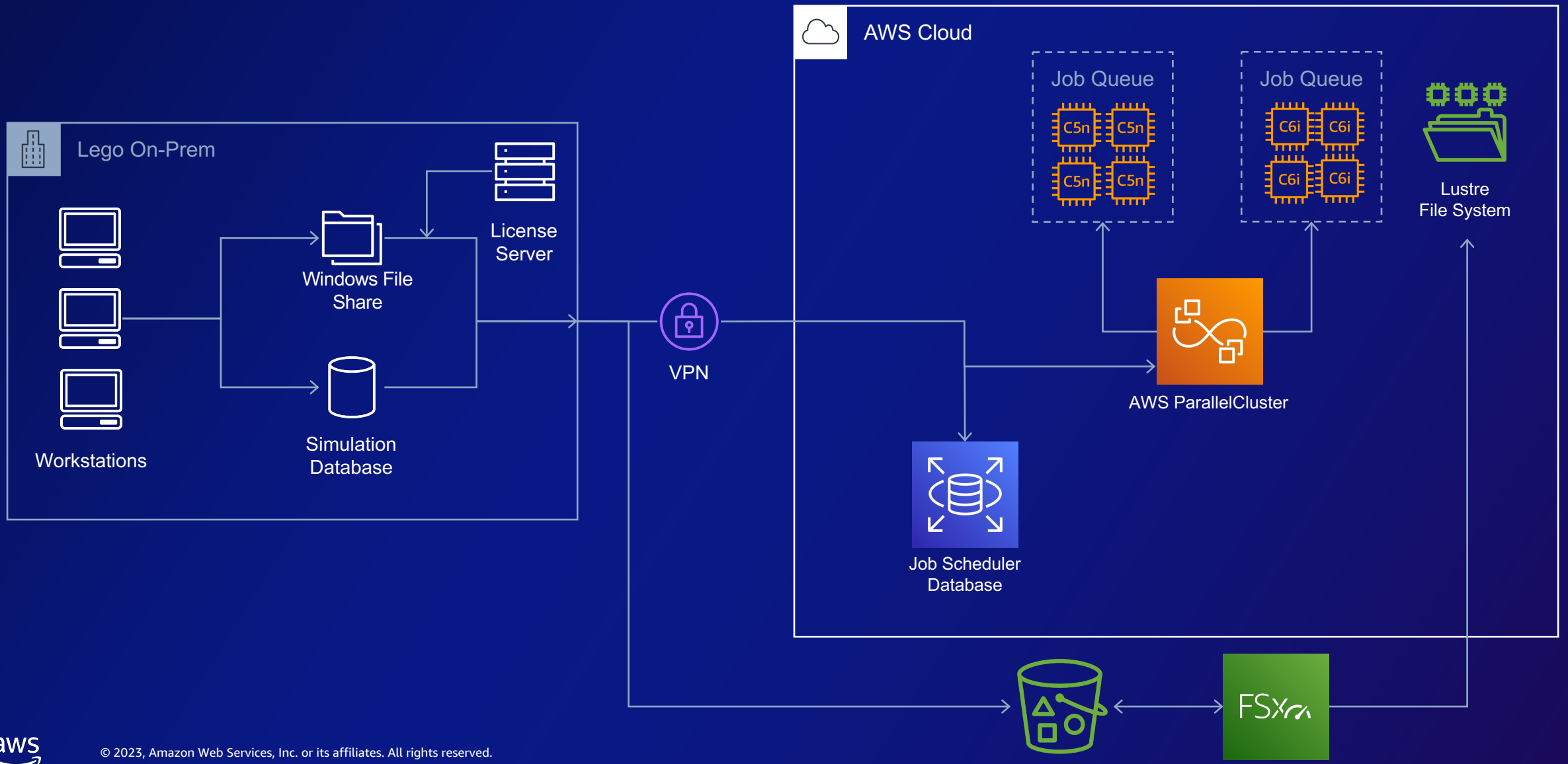
Different CPU requirements for different workloads

Bursty workloads

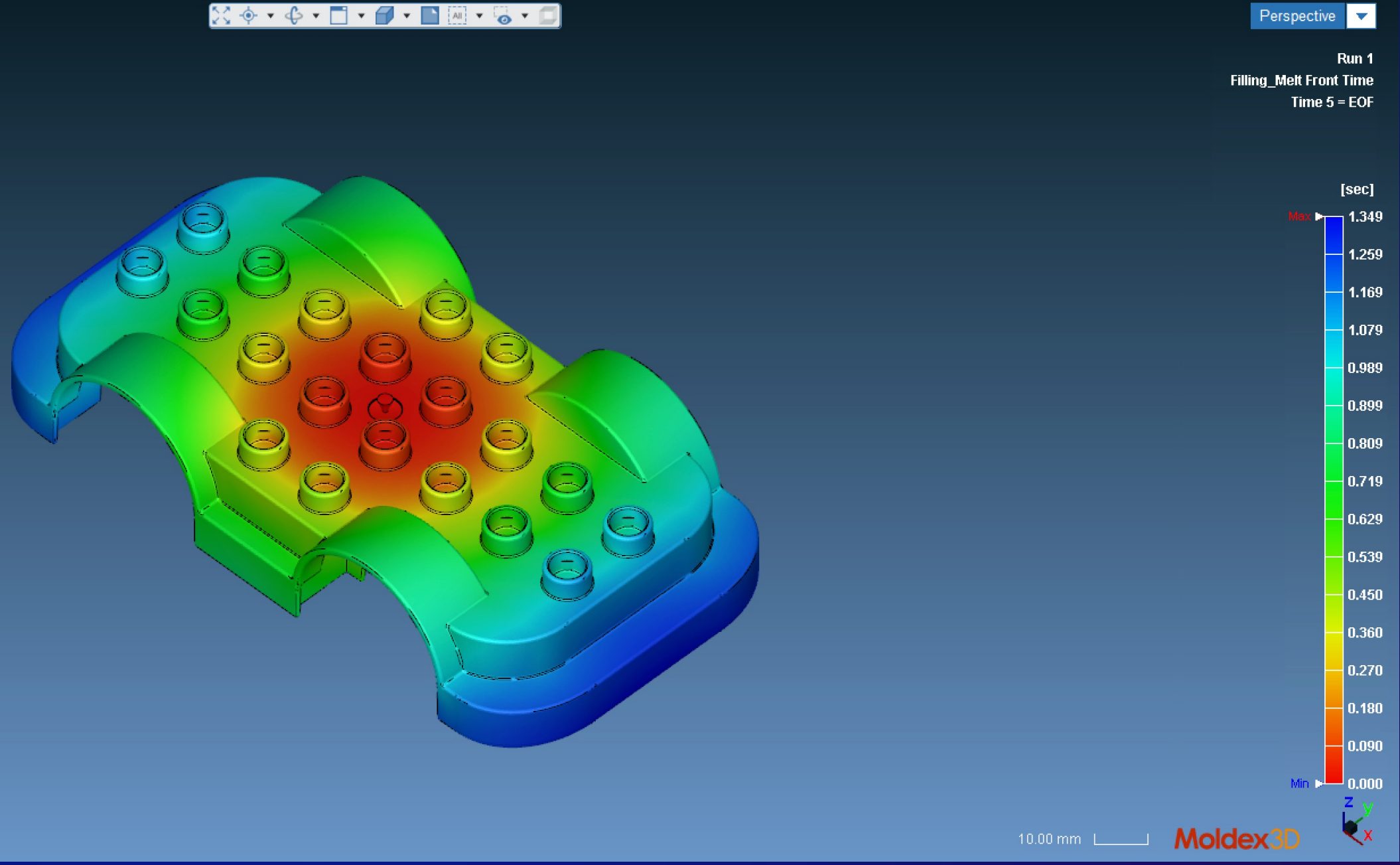
Scalable storage

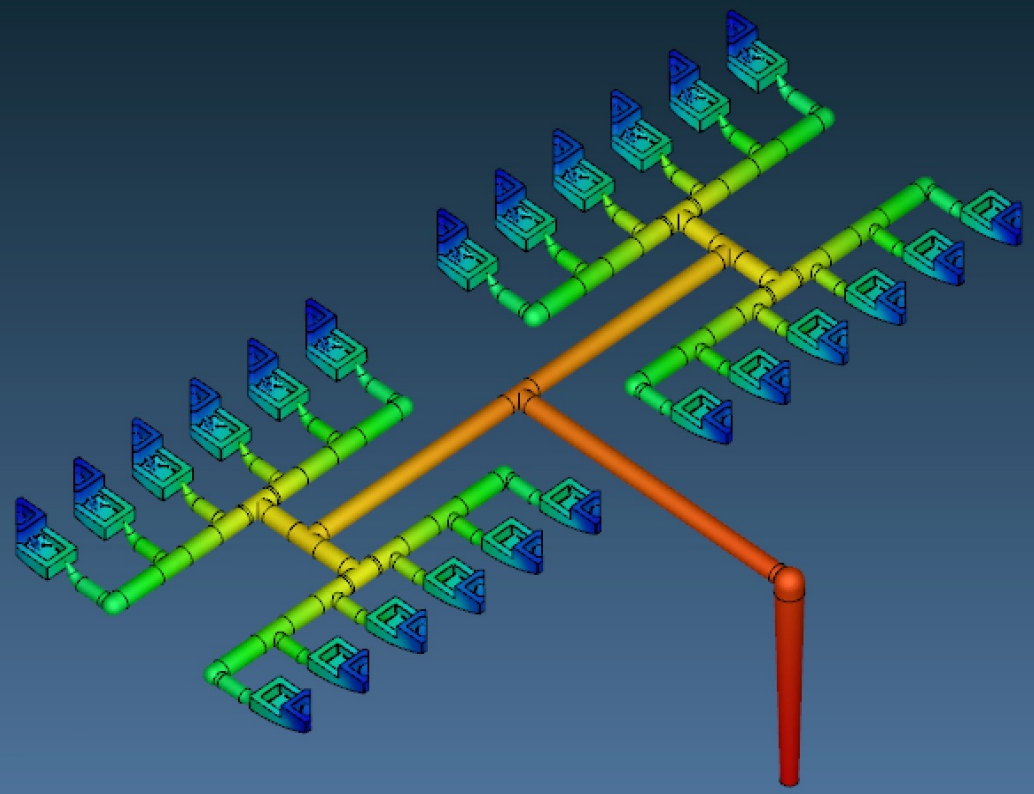
High-performance filesystems

Hybrid HPC Architecture

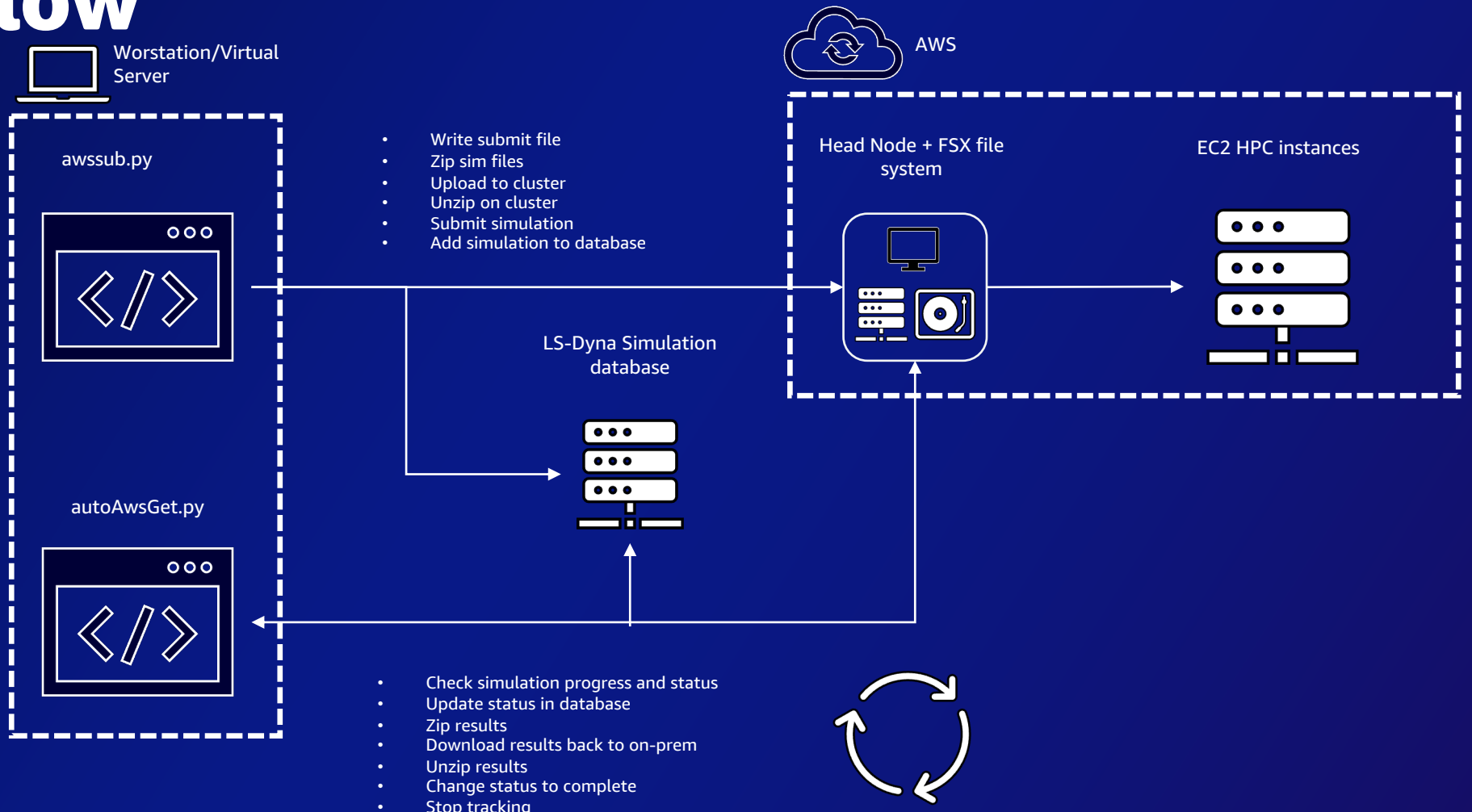


LEGO Simulation Process

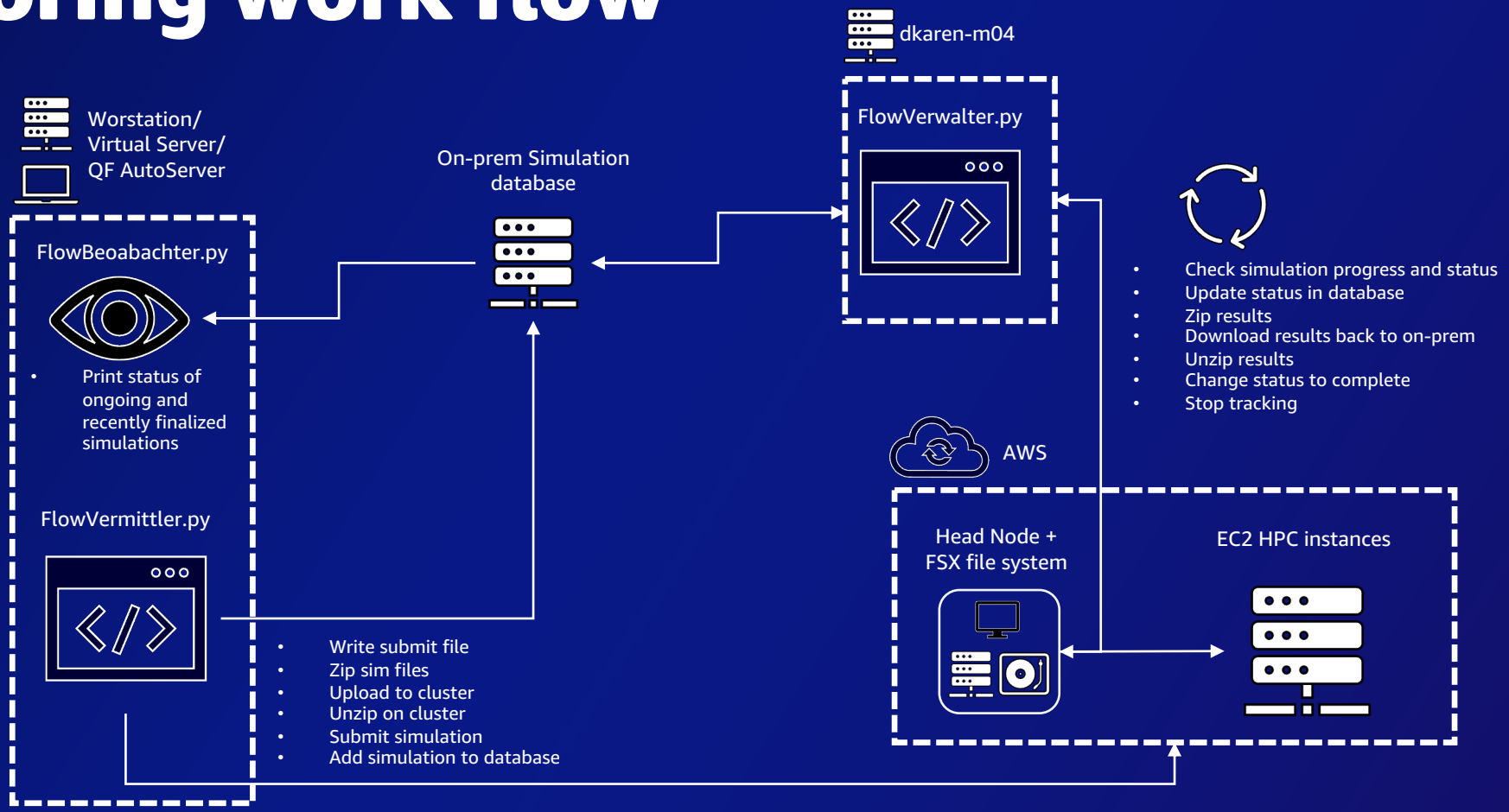




LS-Dyna Simulation Submission and monitoring work flow



Moldex3D Simulation Submission and monitoring work flow



Job Submission

Execution of script awssub.py by user in corresponding simulation folder

awssub.py -c -p -v -e -q -n

- c - number of cores
- p - solver precision - s/d
- v - memory version - smp/mpp
- e - exclusive - y/n
- q - ask interactive question – y/n
- n - instance type – c5.2xlarge/9xlarge/24xlarge

Check if sim exists on server – overwrite y/n

Write submit.sh based on input

Zip simulation input files

Sftp transfer to FSX

SSH to head node

Unzip & submit

Write simulation meta data to LS-Dyna database

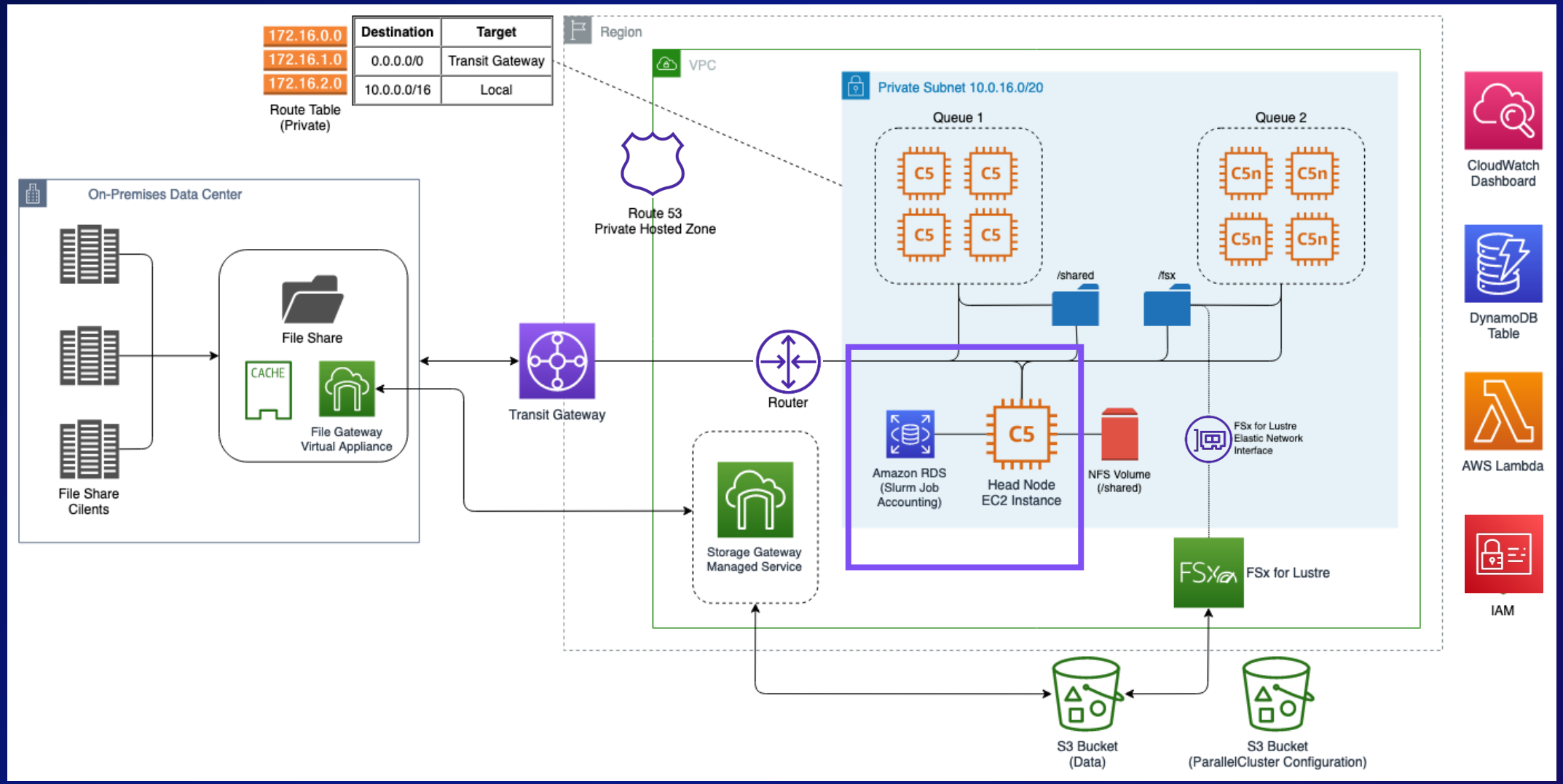
Simulation Monitoring

- Job monitoring procedure
- Start a thread for every simulation that has the status s,p,r or d in the database
- Status is changed from “s” to “p” when thread is started
- In “p” -> check for *.out file in corresponding sim folder
- In “r” -> check *.out file for complete command
- In “d” -> download results zip folder to on-prem and unzip
- In “c” job is complete

```
###################################     STATUS OF YOUR RUNNING JOBS - 10/26/2021, 12:09:36     ###################################

  job_id  user_id    sub_date    sub_time    status    download    sim_progress
--------  ---------  ----------  ----------  --------  ----------  -------------
     362  dkChrMau   12/07/2021  14:03:47    c
     363  dkChrMau   12/07/2021  14:04:27    c
     365  dkChrMau   12/07/2021  14:45:48    c
     366  dkChrMau   12/07/2021  14:46:11    c
     935  dkChrMau   10/08/2021  15:06:39    c         Done

#####################################################     AWS JOB Q STATUS     #####################################################

             JOBID PARTITION                                NAME     USER ST       TIME  NODES     NODELIST(REASON) QOS
              2491   compute FT01_001_000_PD10003005_N_NM_CoF05_   centos PD       0:00      1              (None) normal
              2492   compute FT01_001_000_PD10003005_N_NM_CoF06_   centos PD       0:00      1         (BeginTime) normal
              2493   compute FT01_001_000_PD10003005_N_NM_CoF07_   centos PD       0:00      1              (None) normal
              2494   compute FT01_001_000_PD10003005_N_NM_CoF08_   centos PD       0:00      1         (BeginTime) normal
              2495   compute OP01_001_000_PD10003005_N_NM_CoF05_   centos PD       0:00      1         (BeginTime) normal
              2496   compute OP01_001_000_PD10003005_N_NM_CoF06_   centos PD       0:00      1         (BeginTime) normal
              2497   compute OP01_001_000_PD10003005_N_NM_CoF07_   centos PD       0:00      1 (ReqNodeNotAvail, Un normal
              2498   compute OP01_001_000_PD10003005_N_NM_CoF08_   centos PD       0:00      1              (None) normal


################################################     LS-DYNA LICENSE STATUS     ################################################

No programs running
No programs queued


#####################################################     DISK USAGE     #####################################################

Size Used Avail Use%
=========================
2.2T 669G 1.6T  31%
=========================
Your data occupies 25K
```

License Management

Slurm license management



License Management in Slurm

```
[centos@ip-172-31-20-73 ~]$ scontrol show lic
LicenseName=ls-dyna@flexlm-host
    Total=80 Used=0 Free=80 Reserved=0 Remote=yes
LicenseName=moldex3d@moldexlm
    Total=6 Used=0 Free=6 Reserved=0 Remote=yes
[centos@ip-172-31-20-73 ~]$ _
```

Local: Static licenses, defined in slurm.conf

Remote: "Dynamic", tracked in accounting database

Challenge: Slurm does not natively talk to license managers

```
[centos@ip-172-31-20-73 ~]$ sbatch --partition=cpu --licenses=ls-dyna@flexlm-host job.sh
```

Dynamic Licenses

Query license manager with native tools

```
#!/bin/bash

cnt=0

for i in `lstc_qrun | grep @ | grep -Eo '[0-9]+$'` ; do
    cnt=$(($cnt+$i))
done

echo $cnt
```

- No general approach available
- Automate process
  - CRON
  - Prolog/Epilog
  - Embed in job script

Update Slurm's view of license count

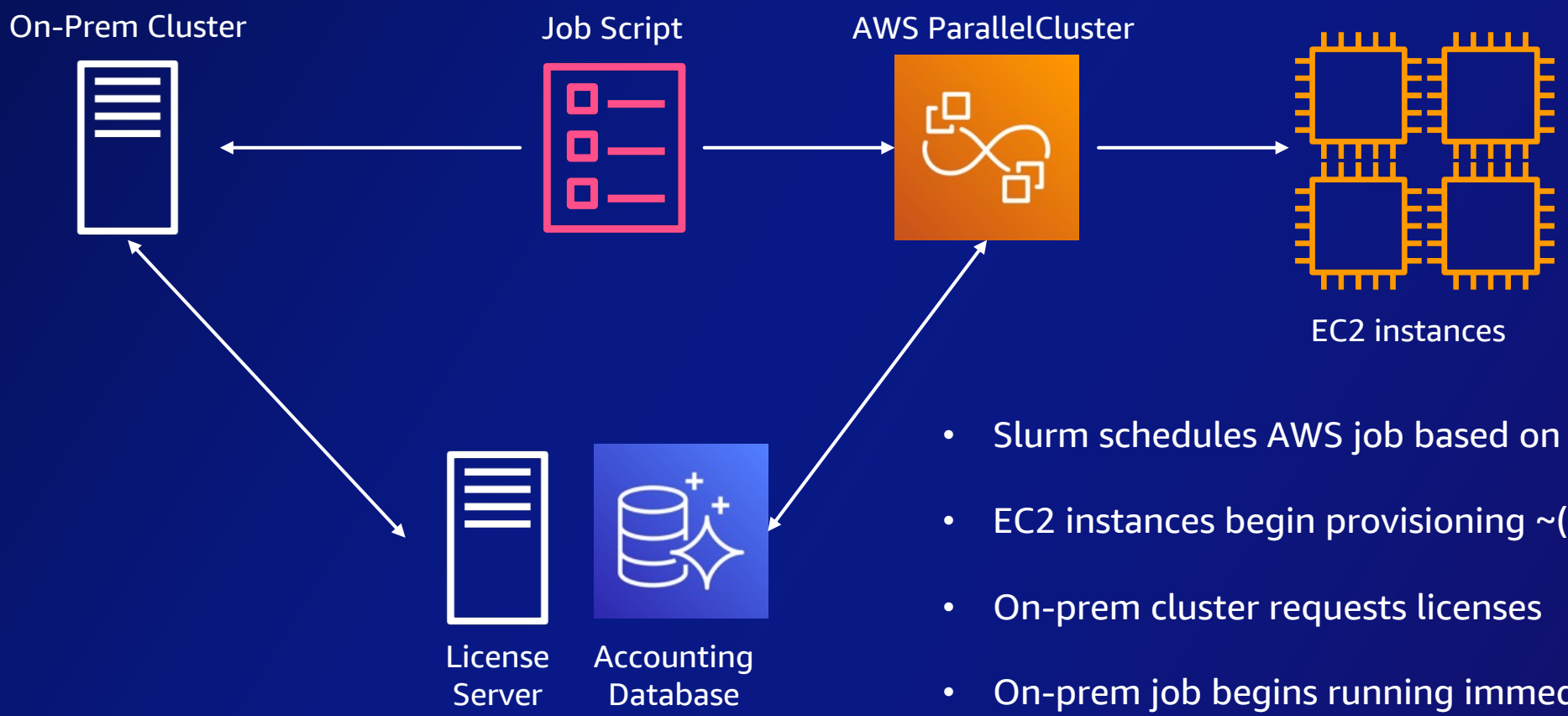
```
#!/bin/bash

AVAIL_LIC=$1

/opt/slurm/bin/sacctmgr -i modify resource name=ls-dyna server=flexlm-host set count=$AVAIL_LIC
```

Dynamic Licenses

```
     473_36       cpu slurm-li   centos CF       0:13      1 cpu-dy-m524xlarge-1
     473_37       cpu slurm-li   centos CF       0:13      1 cpu-dy-m524xlarge-1
     473_38       cpu slurm-li   centos CF       0:13      1 cpu-dy-m524xlarge-1
     473_39       cpu slurm-li   centos CF       0:13      1 cpu-dy-m524xlarge-1
513_[0-39]        cpu slurm-li   centos PD       0:00      1 (Licenses)
514_[0-39]        cpu slurm-li   centos PD       0:00      1 (Licenses)
515_[0-39]        cpu slurm-li   centos PD       0:00      1 (Licenses)
```

Challenges in a Shared License Pool

RACE CONDITIONS



- Slurm schedules AWS job based on available licenses
- EC2 instances begin provisioning ~(3-5 minutes)
- On-prem cluster requests licenses
- On-prem job begins running immediately
- Licenses now locked

Challenges in a Shared License Pool

SOLUTION

- Pre-flight license check
- Minimal cost incurred (startup time)
  - Far less than cost of a locked license
- Node available for other jobs
- **SuspendTime** used to help control scaledown

```
#!/bin/sh

#SBATCH --job-name=Headgear_A_LS_1p2
#SBATCH --ntasks=8
#SBATCH --output=%x_%j.out
#SBATCH --partition=lsdyna-im
#SBATCH --constraint=r5.4xlarge
#SBATCH --licenses lstc@10.10.0.3:8

if [ $(LicenseCheck 8) ]
   mpirun -np 8 foo
   ...
else
   exit
fi
```

Challenges in a Shared License Pool

LICENSE RELEASE

- Moldex3D solver automatically releases licenses at job completion

- Manually cancelling requires SIGINT or SIGTERM
  - Anything else crashes solver
  - Locks the licenses for 24 hours
  - Similar behavior with application failures

- Slurm default behavior
  - SIGCONT, SIGTERM, followed by SIGKILL

Challenges in a Shared License Pool

SOLUTION

- `scancel` options
  - **`scancel --full --signal=TERM <job_id>`**

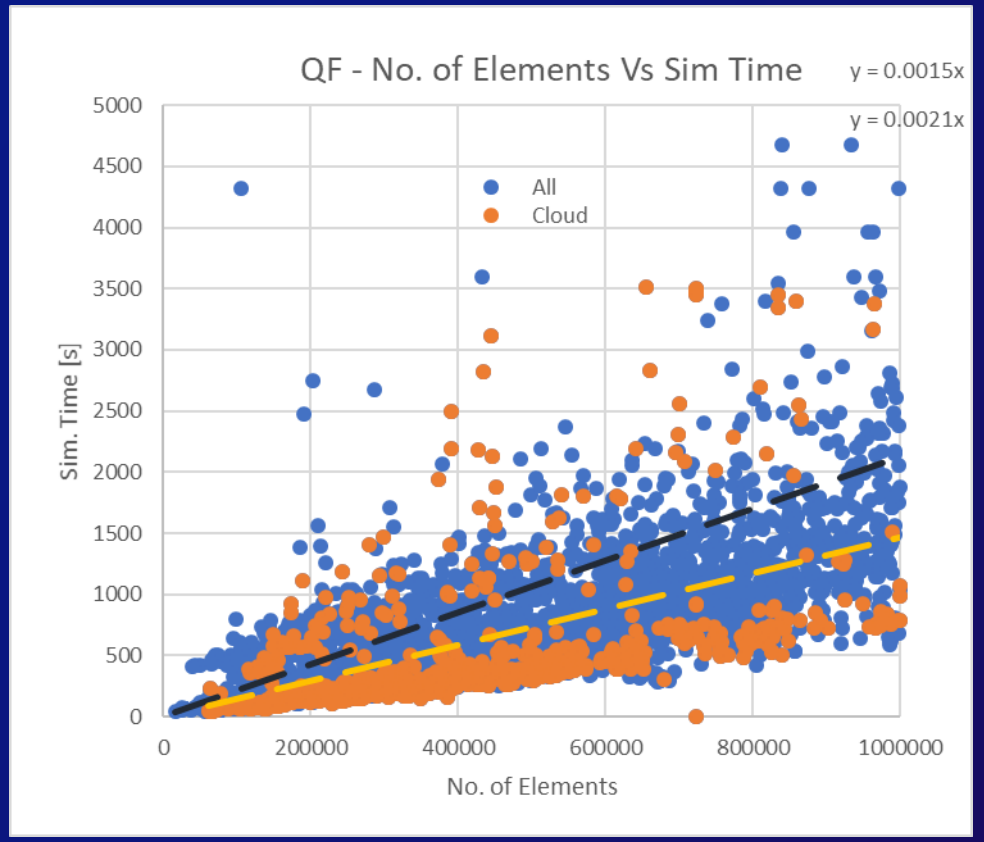
- **"Trap & Kill" function**

```
kill_all()
   trap - SIGINT SIGTERM SIGKILL
   echo "I kill" >> /fsx/kill_trace.log·
   kill -- -$$·

trap kill_all SIGINT SIGTERM SIGKILL
```

Results

- ~40% reduction in simulation time
- Expanded simulation capabilities
  - Easy to expand hardware resources to explore a large design space
  - Ability to increase model details (2 to 4 times larger meshes)

Cost efficient: Cloud costs an order of magnitude cheaper than total simulation license costs



Thank You

Brian Skjerven

bsskjerv@amazon.com