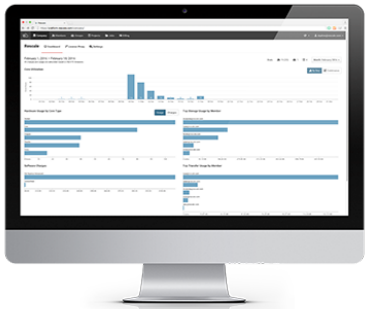
Challenges of deploying your HPC application to the cloud

November 12, 2016

Mulyanto Poort, VP Engineering
mulyanto@rescale.com

Overview

- **Overview of Rescale**
- **Challenges of deploying software on Rescale**
- **How we install and deploy software**
- **Examples**
- **Future developments: ScaleX Developer**
- **Conclusions**



Rescale - Company Profile Overview

Global Growth

San Francisco HQ, Tokyo and Munich offices, further EMEA expansion

300%+ annual growth

Technology

Global cloud-based HPC platform

57+ data centers, 180+ simulation software packages

Customers

ABB Formula 1 SIEMENS Sikorsky

100+ leading Global 2000 enterprises

Investors

Jeff Bezos

Richard Branson

Peter Thiel

TRANSLINK CAPITAL

Microsoft

TRG

ITV ITOCHU TECHNOLOGY VENTURES

JumpCapital

Software on Rescale

180 Applications

26 On demand **- 80** Bring your own license **- 74** Free

Rescale's Cloud Infrastructure

9 **Infrastructure Providers**

58+ Datacenters - 5 regional platforms

Google Cloud Platform

Microsoft Azure

amazon web services™

SYSTEMS

Ohio Supercomputer Center

An OH·TECH Consortium Member

Hardware on Rescale

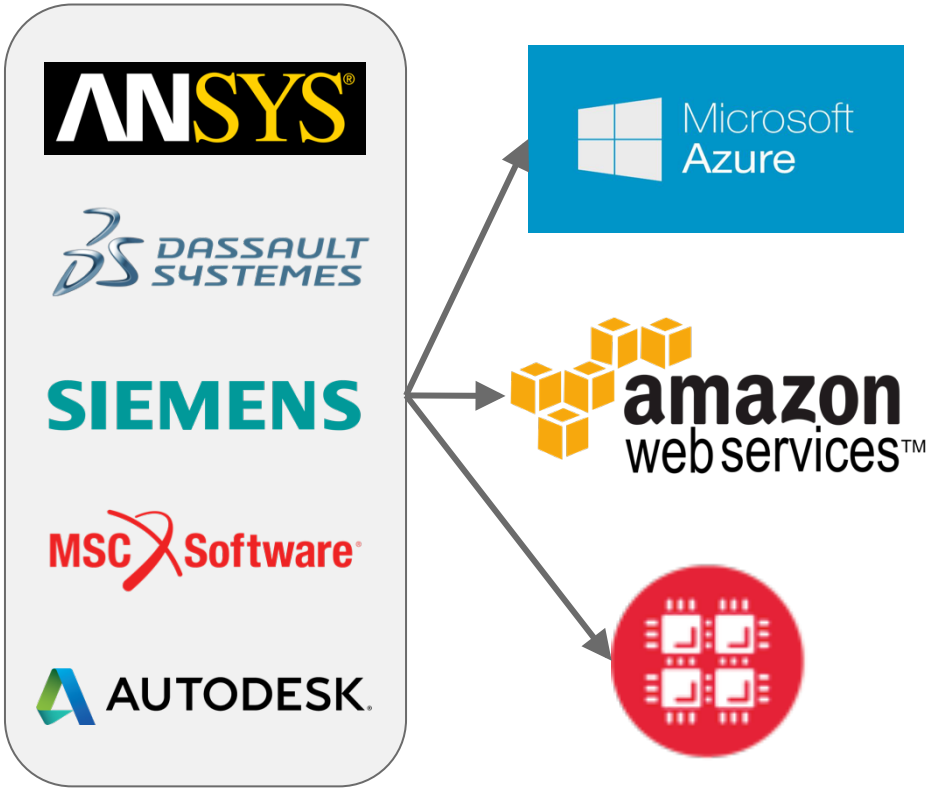
33 **Hardware Configurations**

Intel Xeon

Sandy Bridge **Ivy Bridge**
Haswell **Broadwell**
Phi

Up to 64 cores per node
Up to 2TB of RAM per node
Up to 100Gbps EDR Infiniband

The Challenge: Multiple providers



- RDMA works only with Intel MPI
- RDMA hardware not supported in all regions

- All hardware not supported in all regions
- Amazon Linux OS of choice
- Hard to distinguish hyperthreads from physical cores

- MVAPICH MPI Flavor of choice
- No root access to compute nodes
- Uses rsh instead of ssh
- Bare metal EDR infiniband

The Challenge: Virtual vs Bare metal systems

Virtual	Bare Metal
 	 Ohio Supercomputer Center An OH·TECH Consortium Member
Pros: ● Abstraction of resources ● Configurable environment ● Better user isolation ● Faster hardware refresh cycles Cons: ● Harder to tune hardware/software ● Provisioning time may be slow	Pros: ● Performance ● More familiar environment for traditional HPC users Cons: ● Queued systems ● No root access to compute nodes

The Challenge: Multiple regions

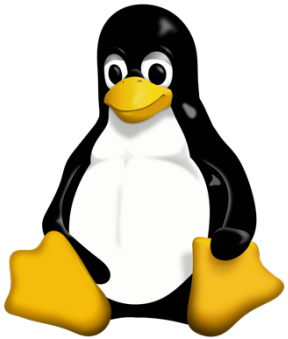
Rescale Platforms

- **platform.rescale.com**
- **eu.rescale.com**
- **kr.rescale.com**
- **itar.rescale.com**
- **platform.rescale.jp**

Provider regions and clusters

- **azure:** westus, westeurope, … 38 regions
- **aws**: us-east, ap-northeast, … 18 regions
- **osc**: Owens, Ruby, Oakley

The Challenge: Multiple OSes and software types



- Linux vs Windows
- Batch vs Virtual Desktop
- Bash vs Powershell
- Intel MPI, OpenMPI, MPICH, Platform MPI, MVAPICH, Microsoft MPI, Microsoft HPC PACK, charm++
- Support workflows with multiple applications / types of applications (e.g. co-simulation, MDO, etc.)

Software installation principles on Rescale

Customer should not have to worry about where the software runs. In other words, the execution should be the same whether it runs on AWS or Azure or another provider

Abstraction of MPI

- Provide common command interface - independent of hardware
- Automatically select optimal MPI for hardware
- Automatically set MPI options like affinity, binding, distribution and interconnect options

Maximize performance

Maximize compatibility with hardware and optimally utilize hardware

- Co-processors, GPUs, AVX2

Global install process

Initial Install

- Install by hand
- Create automated installation script
- Run automated install
- Create JSON definition for installation
- Deploy to provider
- Create regression test

Time ~ 16 hours

Provider Install

- Take base install put it in provider repository
- Create provider specific environment
- Pull down install from repository and install on new provider.

Time ~ 4 hours

Add version to install script

- versions=[10.0, 10.1]
- Run automated install
- Deploy to provider

Time ~ 1 hour

Defining the installation: <software>.json

<software>.json

Description of software, list of versions, environment, license information, etc.

```
{
      “software”: “ansys”,
      “description: “Ansys Software”,
      “versions”: [
            {
                  “version”: “17.0”,
                  “environment_variables”: [...],
                  “installations”: [...],
                  ...
            },
            {
                  “version”: “16.2”,
                  “environment_variables”: [...],
                  “installations”: [...],
                  ...
            }
      ],
      ...
}
```

Defining the installation: <software>.json

<software>.json > environment variables

Defines the environment the software runs in

```
{
      “software”: “ansys”,
      “versions”: [
            {
                  “version”: “17.0”,
                  “environment_variables”: [
                        {
                              “name”: “VERSION”,
                              “value”: “17.0”,
                              “Sort_order”: 1
                        },
                        {
                              “name”: “PATH”,
                              “value”: “$INSTALL_ROOT/$VERSION/bin:$INSTALL_ROOT/$VERSION/mpi/bin”,
                              “Sort_order”: 2
                        }
                  ]
            }
      ]
}
```

Defining the installation: <software>.json

<software>.json > installations

These are references to the locations of the installations

```
{
      “software”: “ansys”,
      “versions”: [
            {
                  “version”: “17.0”,
                  “installations”: [
                        {
                              “provider”: “azure”,
                              “install_root”: “/rescale/ansys”
                        },
                        {
                              “provider”: “osc”,
                              “install_root”: “/shared/rescale/ansys”
                        }
                  ],
                  ...
            }
      ]
}
```

Defining the installation: `rescale-<software>.json`

`rescale-<software>.json`

Defines the resources related to an install root

```
{
      “install_root”: “/rescale/ansys”,
      “providers”: [
            {
                  “provider”: “aws”,
                  “resources”: [
                        {
                              “region”: “us-east-1”,
                              “resource”: “snap-0123456789abcdef”
                        },
                        {
                              “region”: “us-gov-west-1”,
                              “resource”: “snap-00aa11bb22cc33ee”
                        }
                  ]
            },
            ...
      ]
}
```

The install process

Create base install

- Silently install using shell or powershell script
- Install to snapshot, vhd or shared storage location

Replicate install

- Copy install to different regions, storage accounts and clusters
- Use provider API or shell commands

Update json definition files to reflect new resources

Regression testing

Sync json definition with production databases

- Make sure that running jobs are not affected by changes in database

The install process: Install

Common interface for all providers:

```
rescale-install --install-root /program/ansys_fluids --provider azure
```

For AWS, Azure, etc.

- Provision VM/Instance
- Attach clean volume and mount it to <install-root>

For bare metal providers

- An ssh connection is opened to a login account on the cluster

Execute install script

- Powershell (Windows) or Bash (Linux)
- Pull down bits from blob storage
- Run pre-generated script to install software to <install-root>

For AWS, Azure, etc

- Snapshot or vhd is generated. The ID of the resource and save it to the JSON definition

For bare metal providers

- Install is archived and stored in a repository

The install process: Install > code

Python Code

- **AWS Python SDK (**Boto, https://github.com/boto/boto**)**
- **Azure Python SDK (**https://github.com/Azure/azure-sdk-for-python**)**
- **Google Python SDK (**https://github.com/GoogleCloudPlatform/google-cloud-python**)**
- **Fabric (for ssh)**

Common interface for provisioning install resources and executing commands on those resources:

```
provider = ‘azure’
os = ‘linux’
provision_resource = ProvisionResource(install_settings, provider=provider, os=os)
install_action = InstallAction(install_settings, os=os)
provision_resource.with_action(install_action)
```

The install process: Stage

Common Interface:

```
rescale-copy --from us-east-1 --to ap-northeast-1 --provider aws
```

Features:

- **Use provider API when possible, otherwise rsync between regions**
- **For bare metal, pull installation from repository**
- **Save state. Make sure you don’t copy if it’s not necessary**

The install process: Testing

Regression Testing in Testing Environment

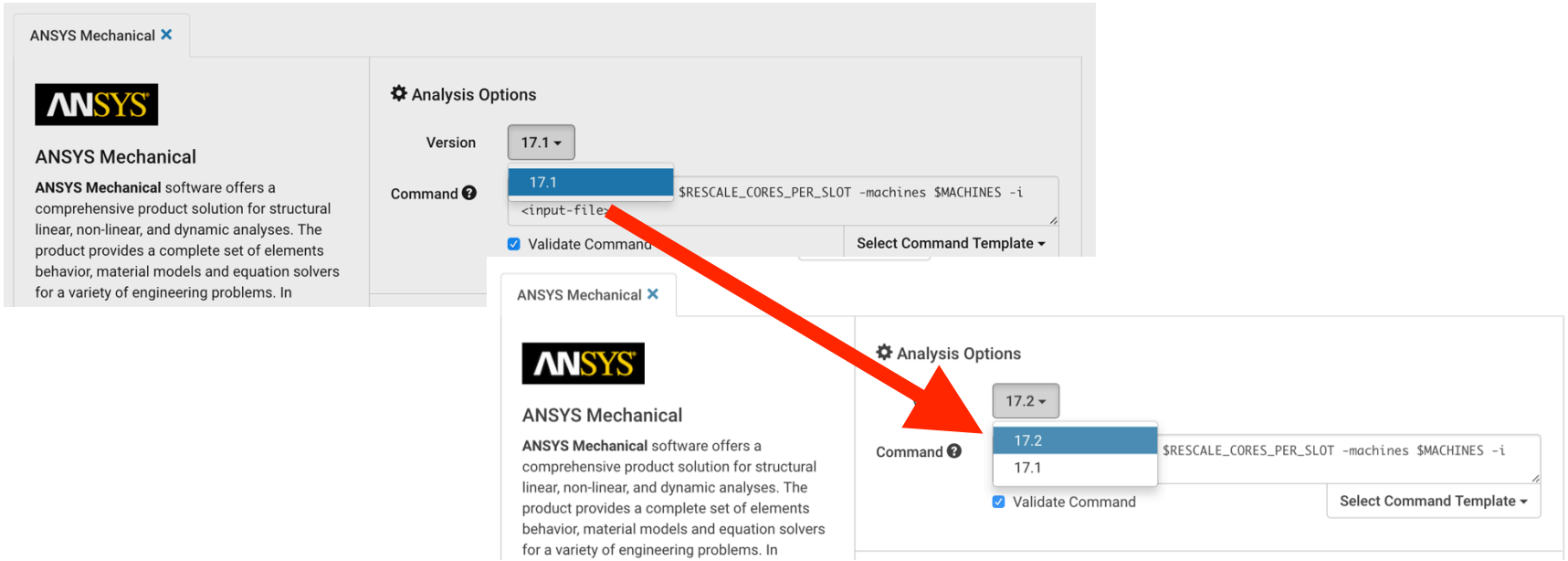
- Use Rescale's API to re-run baseline jobs and compare against expected results
- Integrated with Jenkins build server
- Example definition

```
{
      "environment": "testing",
      "baseline_job_id": "aBcDeF"
      "name": "Ansys MPI Test"
      "tags": [...]
      "tests": [
            {
                  "type": "file_content",
                "file_name": "output.log",
                "Parameters": [
                      {"type": "contains_regex", "value": "^.*[1-2][0-9] seconds.*$"}
                      {"type": "line_count", "value": ">256"}
            }
      ]
}
```

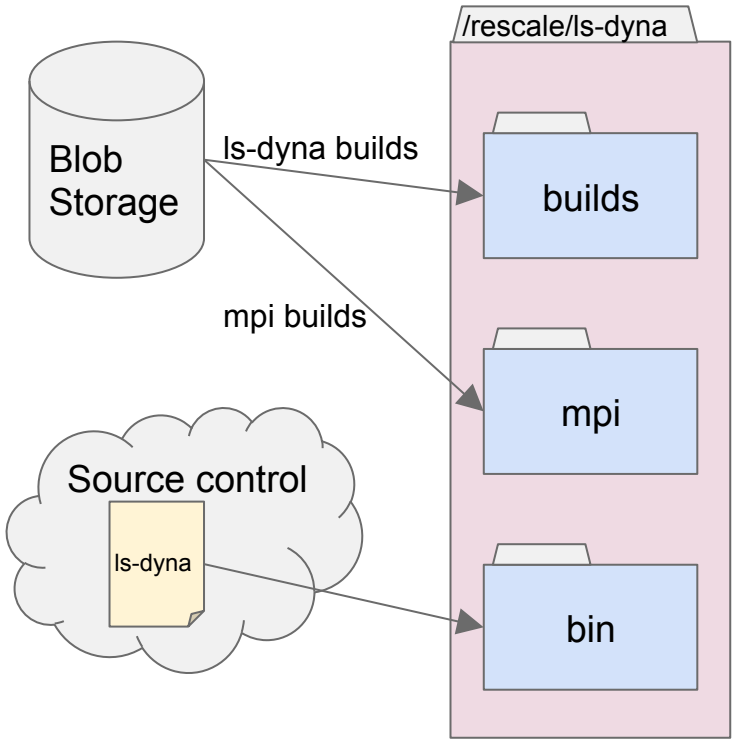
How software is installed on Rescale > Deploy

Definitions are synced to production databases

- Integrated with jenkins build server
- Definitions are pulled from source control and synced with the production database.



Example: LS-DYNA



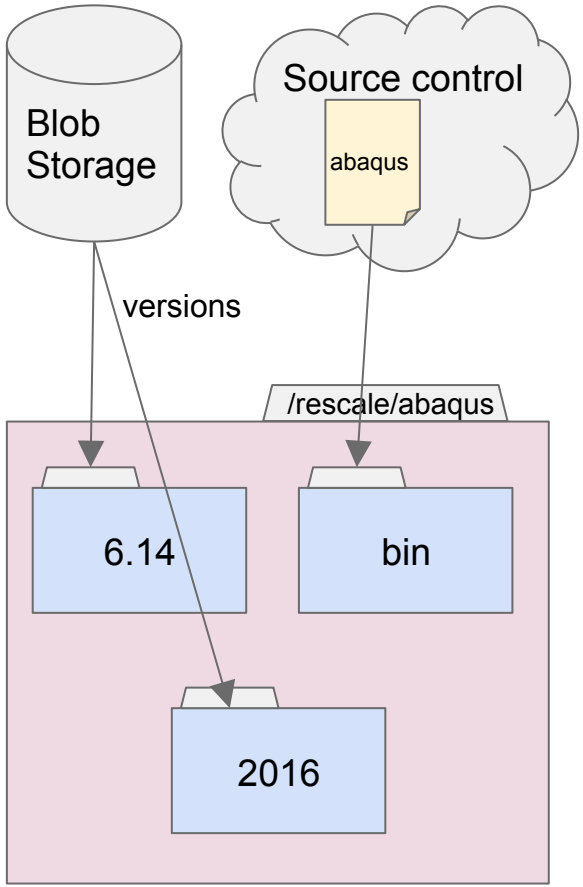
User Command:
```
ls-dyna -i input.k -p double
```

Environment:
```
VERSION=9.0.0
MPI=intelmpi
CORES=32
NODES=2
INSTALL_ROOT=/rescale/ls-dyna
INTERCONNECT=RDMA
PROVIDER=azure
REMSH=ssh
```

ls-dyna

Execution Command:
```
/rescale/ls-dyna/mpi/intelmpi/5.0/bin64/mpirun -np 32 -
machinefile /home/user/machinefile /rescale/ls-dyna/builds/
lsdyna-9.0.0-intelmpi-mpp-double i=input.k
```

Example: Abaqus



User Command:
Abaqus
job=job
cpus=$cpus
mp_mode=mpi
interactive

Environment:
VERSION=2016
INSTALL_ROOT=/rescale/abaqus
ABAQUS_BIN=${INSTALL_ROOT}/${VERSION}/code/bin
ABAQUS_EXE=abaqus
MP_HOSTLIST=['node1',16,'node2',16]
REMSH=ssh

```
#!/bin/bash
# Abaqus wrapper for Azure RDMA
export I_MPI_FABRICS=shm:dapl
export I_MPI_DAPL_PROVIDER=ofa-v2-ib0
export I_MPI_DYNAMIC_CONNECTION=0

echo "mp_mpi_implementation=IMPI" >> ~/abaqus_v6.env
echo "mp_environment_export+=('I_MPI_FABRICS',
'I_MPI_DAPL_PROVIDER', 'I_MPI_DYNAMIC_CONNECTION')" >> ~/
abaqus_v6.env

${ABAQUS_BIN}/${ABAQUS_EXE} "$@"
```

abaqus_v6.env:
mp_hostlist=...
mp_rsh_command=...

ABAQUS

Future plans

ScaleX Developer

- Provide a GUI to our tools to allow ISVs to deploy their software directly to Rescale
- Integrate with Rescale's ISV portal to manage installations and version access
- Integrate with continuous integration systems for testing dev builds and QA testing

ScaleX Open Source

- Integrate with version control systems (github, bitbucket, sourceforge) to allow users to build and deploy their own open source builds at any time
- Create a community for users to share their open source builds with each other

Use these products internally to build, install and deploy software

Conclusions

Lessons learned

- Keep things as abstract as possible to ease integration with new cloud providers
- Understand software limitations and use cases before integrating in the cloud

Unsolved challenges

- A good process for customer provided software
- Continuous integration
- Automatically deploy software when vendor releases new version

Conclusions

Advice for HPC developers to successfully transition to the cloud

- Make your software relocatable (`export SOFWARE_ROOT=/rescale/software`)
- Simple installation process (`tar -xzf install.tar.gz`)
  - Consistent installation process
- Simple batch execution of your software.
- Minimize dependency on user provided libraries (bundle dependencies)
- Have a clear cloud licensing strategy
- Clear separation between Solver and GUI executables.

Become a Rescale Software Partner

Onboarding ISV Package for Intel HPC Dev Con Attendees

What's Included?

- Hosted webinar at launch
- Rescale test credits
- Benchmarking on 3 core types
- Logo on partner page
- Guest blog post
- Beta access to ScaleX Developer
- Case study on Rescale.com
- Dedicated ISV portal*

Email partners@rescale.com

Subject: SW Partner - Intel HPC Dev Con

*For ISV partners with on-demand licensing

Thank You

Questions?

mulyanto@rescale.com - rescale.com