Using Docker, HTCondor, and AWS for EDA Model Development

Accelerating Innovation

Andy Howard
Customer Operations
A Proven Path to Cloud Resources

Simple, Managed, Access to Big Compute
Who is Cycle Computing?

• Leader in Big Compute Cloud Management Software
  • Pioneering Cloud Big Compute/HPC for 10 years
  • 370M compute-hours managed
  • Compute hour growth: 7x every 2 years

• CycleCloud Value Proposition
  • Simple Managed Access to Big Compute
  • Accelerating Innovation for the Enterprise
  • Faster time to result, with cost control

• Our customers
  • Fortune 500, startups, and public sector
  • Life sciences & pharma, financial services, manufacturing, insurance, electronics
Limitations of fixed infrastructure

Too small when needed most
Too large the rest of the time...

• Upfront CapEx anchors you to aging servers

• Costly administration

• Miss opportunities to do better risk management, product design, science, engineering
Containers and the Cloud

• Typically your cloud environment is customized to meet the user’s needs
• Since resources are abundant, sharing of resources isn’t usually needed
• Can use machine images to store configuration state and act as a “container” from an app packaging standpoint
  • Images can even be imported from VM disks to make migration to cloud easier
• But system library dependencies and legacy applications are starting to demand more flexibility...
Western Digital – A proven track record

• Initial single-cluster deployment in 2013
• Now up to 10 production clusters
• Mix of schedulers, applications, and operating systems
• Try to stick to a single user per cluster
EDA Team needs

• Developing simulatable model of solid-state storage architecture
• Existing infrastructure only allowed for 32 simulations in parallel
• Currently using Docker for development
• Goal of running 50,000 simulations per hour
• 6 minute runtime per simulation
Why Docker isn’t always so simple

Where it fits:
• Software development
• Workflows with different library dependencies and versions
• Applications run on a variety of host operating systems

Harder workloads:
• MPI applications
• Applications sensitive to disk or network latencies
• Secure multi-tenant environments
Integration with Amazon’s Elastic Container Repository

• Dockerhub-like private service
• Allows users to upload their own Docker containers
• Each user can see each others containers, but can only upload/delete in their own namespace
• Accessible from both corporate workstations and the cloud environment
• Nodes added to InstanceRole to provide Docker authentication
• Cronjob periodically renews Docker authentication
• HTCondor treats ECR like any other Docker repository
Why not just use Amazon’s Elastic Container Service?!

- Finer-grained control of cluster size and architecture
- Cost controls allow IT managers to cap the cluster budget and alert when users are getting close to their limits
- Workflow can run anywhere, not just AWS
- Supports spot instances – 70-80% cost savings
- Deployed the same way as other engineering clusters in the company
Developer submission workflow

- Users push/pull docker images
- Amazon ECR
- Deploy Image
- App Containers on Spot Instances
- Shared Storage
- data I/O
- WDC
- Client
- SSH
- Cluster Master
- Attached
- AWS
- Virtual Private Cloud
- AWS
HTCondor helps abstract away the compute

• Users only have to define input/output datasets and Docker container to use
• Underlying hardware and software are completely abstracted away
• Wrapper scripts allow users to treat workflow the same way they would submitting to local machines
Core type really doesn’t matter...
Time to production

• Project introduced late November 2016
  • Target PoC end date was December 31, 2016
• Small scale workload (128 cores) was up in less than 1 day
• Large scale test (5000 cores) was running by end of the week
• PoC wrapped up the second week of December
Bringing all the pieces together

• Our software handles all of the orchestration, error handling, and scale optimizations
• Users only need to focus on their application and generating results
• Now to address the deluge of data coming back...
Thank you