Introduction to DHTC

Brian Lin
OSG Software Team
University of Wisconsin - Madison
Local High Throughput Computing

UW - Madison

local

compute

resources
Local High Throughput Computing

local

compute

resources

UW - Madison
How do you get more computing resources?
#1: Buy Hardware
#1: Buy Hardware

- Great for specific hardware/privacy requirements
- Costs $$$
  - Initial cost
  - Maintenance
  - Management
  - Power and cooling
- Takes time
- Rack/floor space
- Obsolescence
- Plan for peak loads, pay for all loads
#2: Use the Cloud
#2: Use the Cloud - Pay per cycle

- e.g. Amazon Web Services, Google Compute Engine, Microsoft Azure, Rackspace
- Fast spin-up
- Costs $$$
- Still needs expertise + management
  - Easier than in the past with the condor_annex tool
- Does it fit with your institution’s policies?
#2: Use the Cloud - ‘Managed’ clouds

• e.g. Cycle Computing, Globus Genomics
• Pay someone to manage your cloud resources — still costs $$$
• Researchers and industry have used this to great success
  - Using Docker, HTCondor, and AWS for EDA Model Development
  - Optimizations in running large-scale Genomics workloads in Globus Genomics using HTCondor
  - HTCondor in the enterprise
#3: Share Resources
#3: Share Resources - Distributed HTC

- University of Nebraska - Lincoln
- UW - Madison
- University of Chicago
i. Manual Job Partitioning

Let’s start sharing!
Manual Partitions

- Obtain sharing agreements
- Query each site for idle resources
- Partition and submit jobs based on availability
#3: Share Resources - Distributed HTC

University of Nebraska - Lincoln

UW - Madison

University of Chicago
#3: Share Resources - Distributed HTC

- University of Nebraska - Lincoln
- UW - Madison
- University of Chicago
Manual Partitions - Shortcomings

• Fewer agreements = fewer potential resources
• More agreements = more account management
• Querying and partitioning is tedious and inaccurate
• Are you allowed to share? Do you have anything to share?
• Not all sites use HTCondor — other job schedulers e.g., SLURM, PBS, etc.
• Pools are independent — workflows must be confined to a single pool
ii. Automatic Job Partitioning

Let the computers do the work
Automatic Partitions - Shortcomings

Homer: Kids: there's three ways to do things; the right way, the wrong way and the Max Power way!

Bart: Isn't that the wrong way?

Homer: Yeah, but faster!

Automatic Partitions - Shortcomings

"I SPEND A LOT OF TIME ON THIS TASK. I SHOULD WRITE A PROGRAM AUTOMATING IT!"

THEORY:

WORK

WRITING CODE

WORK ON ORIGINAL TASK

TIME

FREE TIME

AUTOMATION TAKES OVER

REALITY:

WORK

WRITING CODE

DEBUGGING

ONGOING DEVELOPMENT

TIME

RETHINKING

NO TIME FOR ORIGINAL TASK ANYMORE

Source: https://xkcd.com/1319/
#3: Share Resources - Requirements

- Minimal account management
- No manual job partitioning
- DAG workflow functionality
- Don’t have to learn additional job schedulers
- Don’t have to share our own resources
iii.

Overlay Systems

Let the OSG do the heavy lifting
The OSG Model

OSG Submit and CM

OSG

Site
The OSG Model

- OSG Submit
- and CM

OSG

Site
The OSG Model

OSG Submit and CM

OSG

Pilot Jobs

Site
The OSG Model

OSG Submit and CM

Pilot Jobs

Site
On a regular basis, the central manager reviews Job and Machine attributes and matches jobs to slots.
The OSG Model
The OSG Model - Jobs in Jobs

Photo Credit: Shereen M, Untitled, Flickr https://www.flickr.com/photos/shereen84/2511071028/ (CC BY-NC-ND 2.0)
The OSG Model

- Pilot jobs (or pilots) are special jobs
- Pilots are sent to sites with idle resources
- Pilot payload = HTCondor execute node software
- Pilot execute node reports to your OSG pool
- Pilots lease resources:
  - Lease expires after a set amount of time or lack of demand
  - Leases can be revoked!
The OSG Model - Leasing the Cloud

- What if there aren’t enough idle resources?
- Combine overlay system with cloud technology
- Expect some of your OSG jobs to automatically run in the cloud in the next few years
- … but this should be completely transparent to you
The OSG Model - Collection of Pools

- Your OSG pool is just one of many
- Separate pools for each [virtual] organization (VO)
- Your jobs will be running on the OSG VO pool
The OSG Model - Getting Access

• During the school:
  − OSG submit node at UW (exercises)
  − OSG submit node via OSG Connect (Thursday)

• After the school:
  − Both of the above
  − VO-hosted submit nodes
  − Institution integration with the OSG
Thanks!

Questions?