Bringing Cloud Elasticity to High-Throughput Scientific Applications
Cloud Elasticity at Work

HTCondor scheduler

Amazon (60k cores)

Fermi (15k cores)
~60,000 cores from AWS

More than 16 million core-hours in production
Elasticity Steps

1 - Make spending decisions
2 - Prepare image(s)
3 - Provision instances
4 - Run jobs
5 - Monitor
6 - Shut down
Elasticity at UW-Madison

HTCondor scheduler → local → OSG

HTCondor scheduler → local → CHTC
Motivating Example

› Dr. Needs-Moore needs more cycles in less time than she can get even by combining local, campus, and OSG resources.

› She decides she’s willing to spend some of her grant money to make this happen.

› She can’t spend her grant money on other people’s computation, so she needs her own “annex” in the cloud.
Cloud Elasticity at UW-Madison

HTCondor scheduler → local

HTCondor annex daemon → Amazon

Amazon → OSG

HTCondor annex daemon → CHTC
Identify valuable workflows and assign a value and a deadline.

Policy enforcement:

• budget
• number of concurrent jobs

1 - Make spending decisions
2 - Prepare image(s)
3 - Provision instances
4 - Run jobs
5 - Monitor
6 - Shut down
Developers release “canonical” images.
Pool administrator adjusts one to suit.
Image set as default for pool’s users.
HTCondor configures the instances to join the pool and securely shares the required secret at runtime.
5 - Monitor

› How much am I spending?
  › How many instances have we started?
  › How much does each one cost?

› What am I gaining?
  • How many instances have joined the pool?
    • Which ones haven’t?
  • Are those instances running jobs?
    • If not, can we tell why?
  • Are those jobs finishing?
6 - Shutdown

User specifies a lease.
› HTCondor implements lease in the cloud.
› Each instance configured to shut itself off if has no work to do.

- 1 - Make spending decisions
- 2 - Prepare image(s)
- 3 - Provision instances
- 4 - Run jobs
- 5 - Monitor
- 6 - Shut down
Elasticity demonstrated at medium scale.

- (Only 50-60 thousand cores.)

Prototype of end-user tool developed.

- Demonstrated at HTCondor Week 2016.

Developing faster and more scalable mechanism for cloud provisioning.

Designing production tool for campus use.