Principles for HTC

2014 OSG User School – Thursday afternoon

Tim Cartwright
cat@cs.wisc.edu

University of Wisconsin–Madison

OSG Software Team Manager
OSG Education Coordinator

The Hope of HTC

Open Science Grid

Reminder: What is HTC?

"… the use of ***many computing resources*** over ***long periods of time*** to accomplish a computational task" (Wikipedia, retr. 25 June 2013)

- Try to use all resources, reliably, all the time
- Maximize operations per ***year***
- Use any computers, even old or slow ones

Open Science Grid

What System Is This?

- ***Mystery System X***:
  - Provides a ***lot*** of computing
  - Has high availability and reliability
  - Degrades gracefully
  - Spreads the workload automatically
  - Grows (and shrinks) easily when needed
  - Responds well to temporary overloads
  - Adapts easily to new uses

- HTCondor? OSG? Amazon EC2? Other Clouds?

Actually…

- Those were all ***promised*** features!
- … of distributed data processing systems
- … from the 1970s!!!

  (Adapted from: Enslow, P. H., Jr. (1978). What is a "distributed" data processing system? *Computer, 11*(1), 13–21. doi:10.1109/C-M.1978.217901)

- Sound like promises of today: HTC, grid, cloud

The *Hype* of HTC?

Open Science Grid

Miron Livny

- Founder and leader of Condor Project since mid-1980s
- PI and Technical Director of OSG
- Coined term “high throughput computing”
- Has principled approach to HTC

Miron's Reminder

What has been
is what will be,
and what has been done
is what will be done,
and ***there is nothing new under the sun***.

— Ecclesiastes 1:9 (ESV)

Attributed to Koheleth, who was Ecclesiastes or its author, often taken to be Solomon, son of David, king in Jerusalem, ~950 BCE

Ecclesiastes, (קֹהֶלֶת, Koheleth, "son of David, and king in Jerusalem," alias Solomon, wood engraving, Gustave Doré (1832–1883)

Open Science Grid

Nothing New Under the Sun

- New terms, new hype

  Distributed data processing, cluster computing, grid computing, cloud computing, virtualization, peer-to-peer, client-server, cyberinfrastructure

- But, the underlying problems are the same
- Principles to address the problems are the same

Open Science Grid

But, But … Aren't Clouds New?

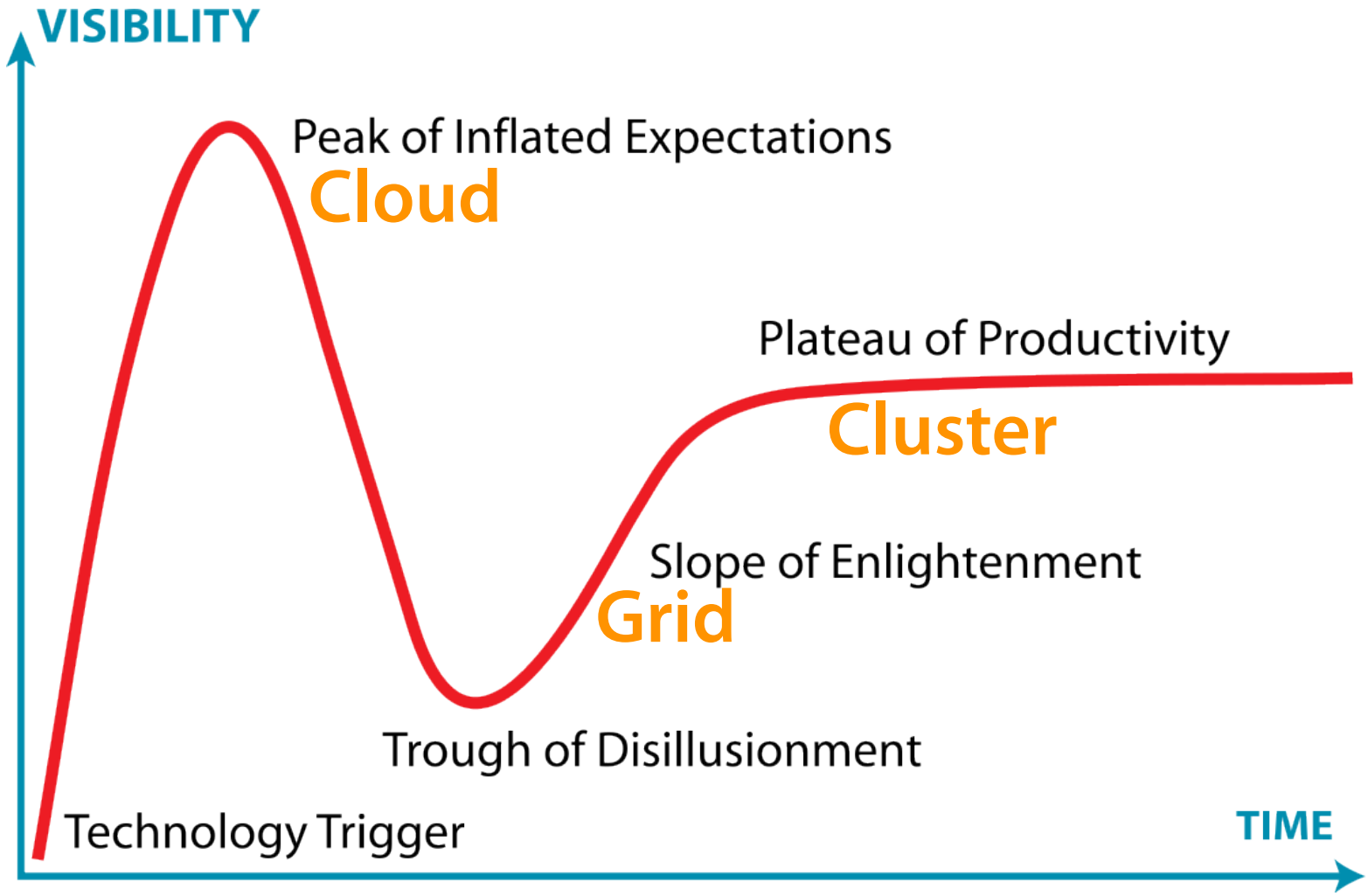
- Amazon EC2 makes it easy to create virtual machines on the fly. That's new, right?
- ***HTCondor:*** Allocate a computer + Start a job
- ***Amazon EC2:*** Allocate a VM + Start it
- Conceptually, a virtual machine is just another kind of job: You start it, you stop it, you have some control over it

Open Science Grid

What *Is* New About Clouds?

- **OSG/XSEDE:** User cost for existing resources: **$0**
  - OSG: Submit jobs, wait for available slots
  - XSEDE: Submit proposal, get free allocation
- **Amazon EC2:** Bring a credit card, ≥**$0.05**/hour
  - Whether you use it or not
- Really, the difference is between gov't-funded and commercial resources, ***not*** technology

Gartner Hype Cycle



http://www.gartner.com/technology/research/methodologies/hype-cycle.jsp

Instead

- Know the principles
- Understand how the principles apply

- And you will be able to use the next technology in the next hype cycle to help your science

- So let's look understand the principles

Distributed Computing

Open Science Grid

Is This Distributed Computing???

"You know you have a distributed system when the crash of a computer you've never heard of stops you from getting any work done."

— Leslie Lamport

Definition of Distributed Computing

1. Multiplicity of resources
2. Component interconnection
3. Unity of control
4. System transparency
5. Component autonomy

Enslow, P. H., Jr., & Saponas, T. G. (**1981**). *Distributed and decentralized control in fully distributed processing systems: A survey of applicable models* (GIT-ICS-81/02). Georgia Institute of Technology.

Open Science Grid

1. Multiplicity of Resources

- Use many general purpose components
  - Physical or logical
  - Do not need to be the same (homogenous)
  - But should have same/similar capabilities
  - I.e., want to assign tasks anywhere
- Improves reliability and throughput

2. Component Interconnection

- Thus, physically and/or logically distributed
- Connected via network (these days)
  - Use a two-way cooperative protocol
  - Requests can be refused based on local knowledge
- Example: HTTP

Open Science Grid

3. Unity of Control

- ***Not*** centralized control
  - Creates single point of failure
  - Avoid critical paths, critical components
- Unify around a common goal
  - Via shared policy and, maybe, software
  - Multiple centers of control
- Improves overall reliability and availability

Open Science Grid

4. System Transparency

- Users should feel like the whole system is one giant virtual machine
- Should not need to know:
  - Which physical component will perform task
  - Where the component is located
- May be the hardest part to achieve

Open Science Grid

5. Component Autonomy

- ***Autonomy*** (act locally)
  - Resources make their own decisions
  - Accept or refuse requests
  - Based on local policy
- ***Cooperative*** (think globally)
  - I.e., common policies and goals

Open Science Grid

Laptops: Distributed System?

Multiplicity?	Sort of: multiple cores
Interconnection?	**No:** not physically distributed
Unity of control?	Yes
Transparency?	Yes: ***is*** one system
Autonomous?	**No:** parts under central control

So, your laptop alone is not a distributed system

Open Science Grid

glideinWMS: Distributed System?

Multiplicity?	Yes: many diverse resources
Interconnection?	Yes: distributed and connected
Unity of control?	Yes: one system and goal
Transparency?	Yes: appears to be one pool
Autonomous?	Yes: sites have local control

So, glideinWMS is a distributed system [phew!]

Principles for HTC

Open Science Grid

4 Principles for DHTC

- Enslow's definitions are ~30 years old
- From them, OSG has derived guiding principles
- Copied from the recent OSG funding proposal

Open Science Grid

DHTC Principles

- ***Resource Diversity***
  The system must be flexible enough to accept many types of resources, software, and services
- ***Dependability***
  Throughput must be tolerant to faults: There will always be services or resources that are not available
- ***Autonomy***
  You must allow resource providers from different organizations to share; you must allow them to preserve local autonomy to set policies and select technologies
- ***Mutual Trust***
  You must support complex trust relationships across organizations and software tools

got principles?

Open Science Grid

glideinWMS: Resource Diversity?

- Can use different kinds of Compute Elements:
  GRAM, CREAM, NorduGrid
- Compute Elements work with different clusters:
  HTCondor, Torque/PBS, LSF, SGE, (SLURM soon)
- Supports multiple platforms:
  Intel 32- & 64-bit × RHEL, CentOS, SL 5 & 6

Open Science Grid

glideinWMS: Dependability?

- Uses HTCondor:
  - Reliable queue to manage jobs
  - Resources appear and disappear
  - Pool (often) relies on one Central Manager…
- Accesses multiple sites:
  - Individual sites come and go

Open Science Grid

glideinWMS: Autonomy?

- ***Sites***
  - Choose to accept VOs and users
  - Can limit how many pilot jobs run
  - Can limit how long pilot jobs run
  - Can prefer local users over glide-ins
- ***Glide-ins***
  - Factory decides how many pilots to submit
  - Factory controls policy on running pilots
  - Pool has policy for how users share resources

Open Science Grid

glideinWMS: Mutual Trust?

- Pilot jobs are authenticated and authorized
  - Based on X.509 certificates
  - Not using user certificates, though
- Traceability
  - So how to associate a ***user*** job with the ***pilot*** job that acquired the resource?
  - Chained: VO knows user, resource knows VO
  - Helping to reduce number of certificates … we all know how much fun certificates are!

Open Science Grid

Principles and Scale

- The principles apply:
  - For a cluster
  - For a campus
  - For a grid (like OSG) or cloud (EC2)
  - For … ???

Principles and You

Open Science Grid

How should you apply the principles?

- ***Resource Diversity***

  Can your jobs run on diverse resources? (Location, platform, configuration, storage, etc.) If you get more, can you use them?

- ***Dependability***

  Are your jobs robust? Can they recover from (or at least report) errors? Do you use methods to check results, retry nodes, etc.?

- ***Autonomy***

  Be prepared to deal with the consequences of local decisions; jobs may be rejected, removed, throttled; resources come & go

- ***Mutual Trust***

  Who do you trust with your data? Do you believe your results ("trust but verify")? Must deal with security systems…

Open Science Grid

Be Skeptical!

- When someone wants to sell you a new way of doing distributed computing, ask:
  - Can it handle diverse resources?
  - Is it dependable? (Oh yeah, how so?)
  - Are components autonomous?
  - Does it manage trust relationships?
  - Does it improve on the ability to provide the fundamental principles, or is it merely a shiny new version of what we already have?

There is nothing new under the sun!

Questions?

Next

- Discussion can continue into break
- Coming next:

Now – 2:45	Break
2:45–4:20	HTC Showcase
4:20–4:45	Break + Showcase Q&A
4:45–5:15	Wrap up