



Open Science Grid

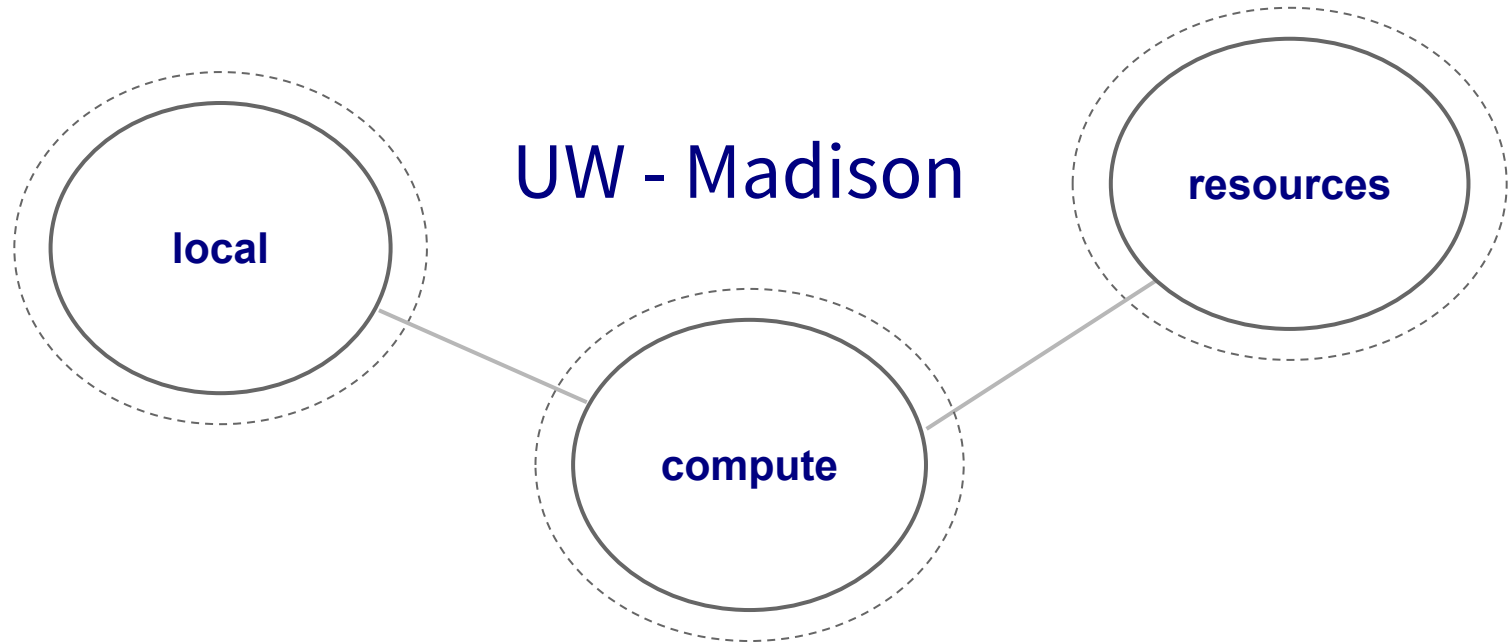
# Introduction to DHTC

Brian Lin

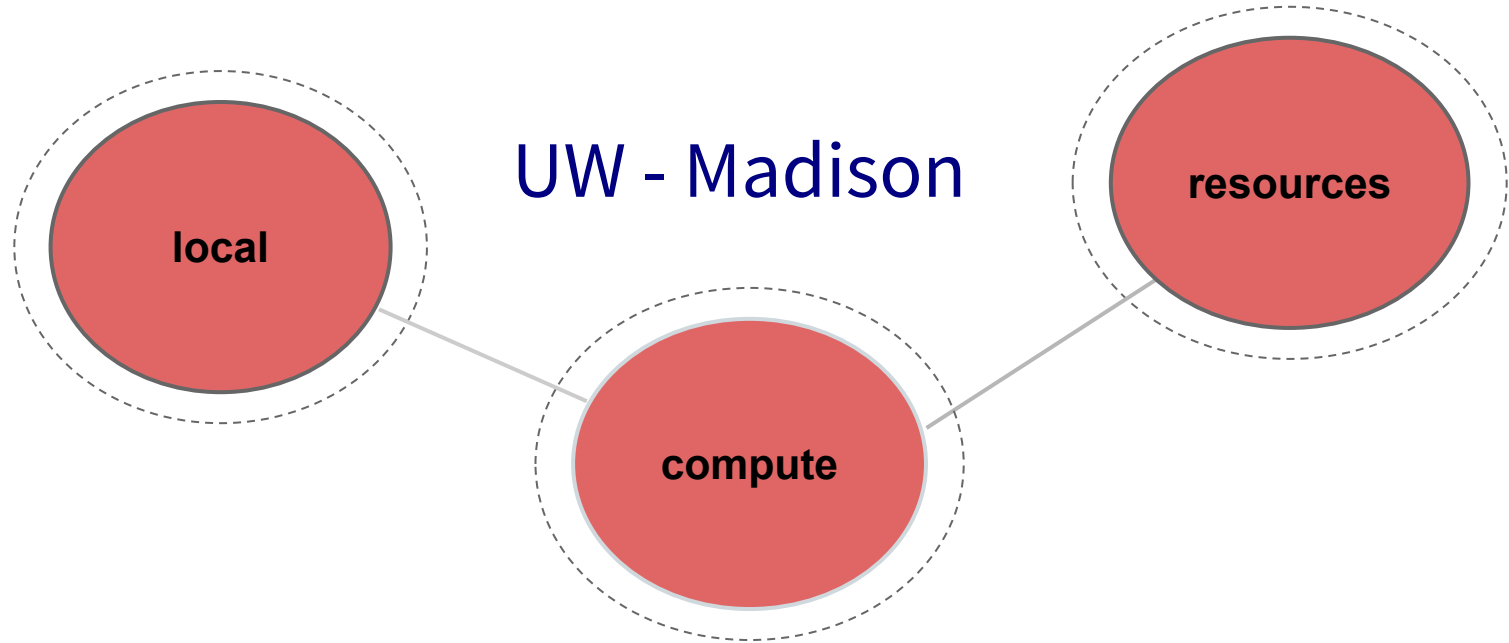
OSG Software Team

University of Wisconsin - Madison

# Local High Throughput Computing



# Local High Throughput Computing





# 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
- Delivery and installation 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 payment fit with your institutional or grant policies?



## #2: Use the Cloud - 'Managed' clouds

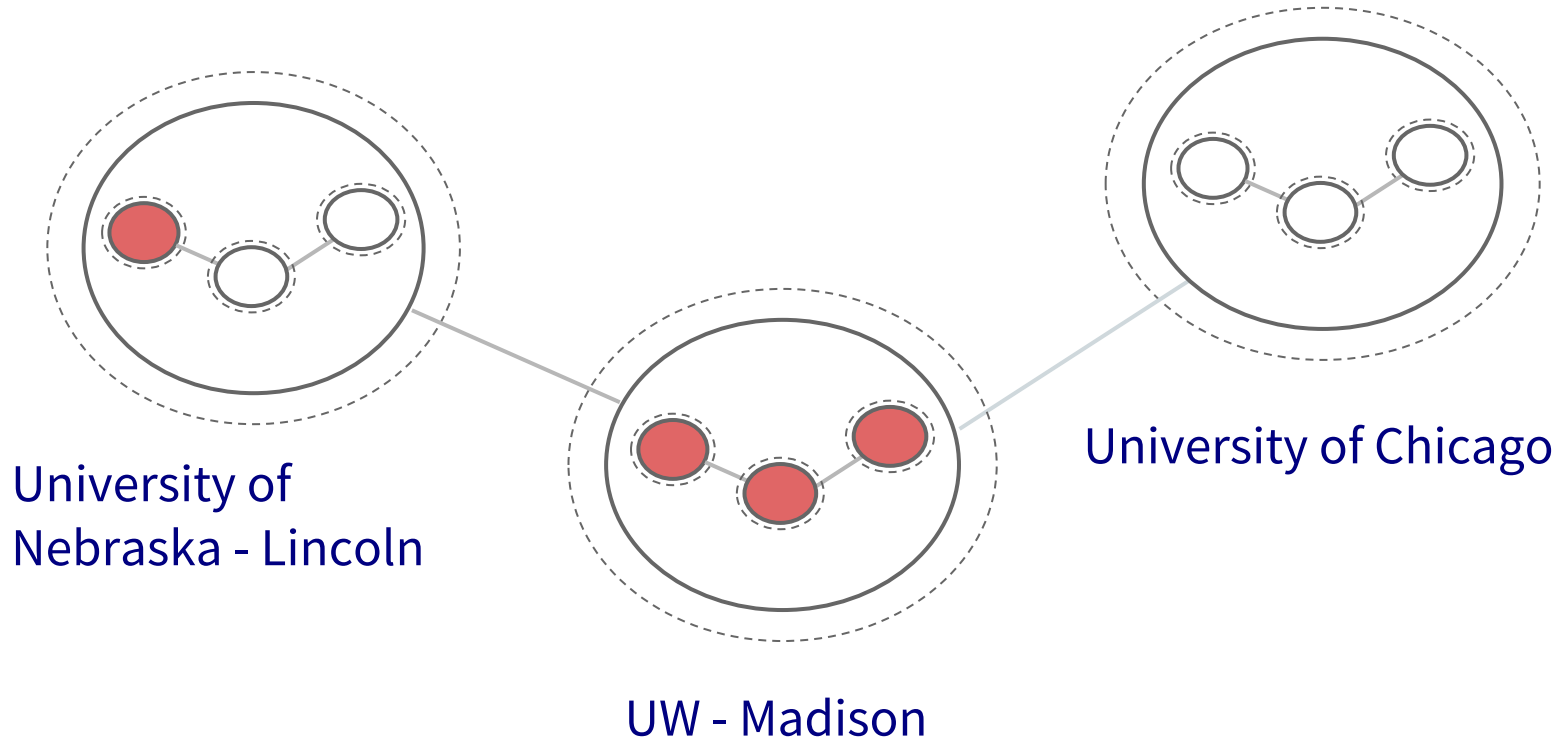
---

- 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
  - HTCondor at Cycle Computing: Better Answers. Faster.



# #3: Share Resources

# #3: Share Resources - Distributed HTC



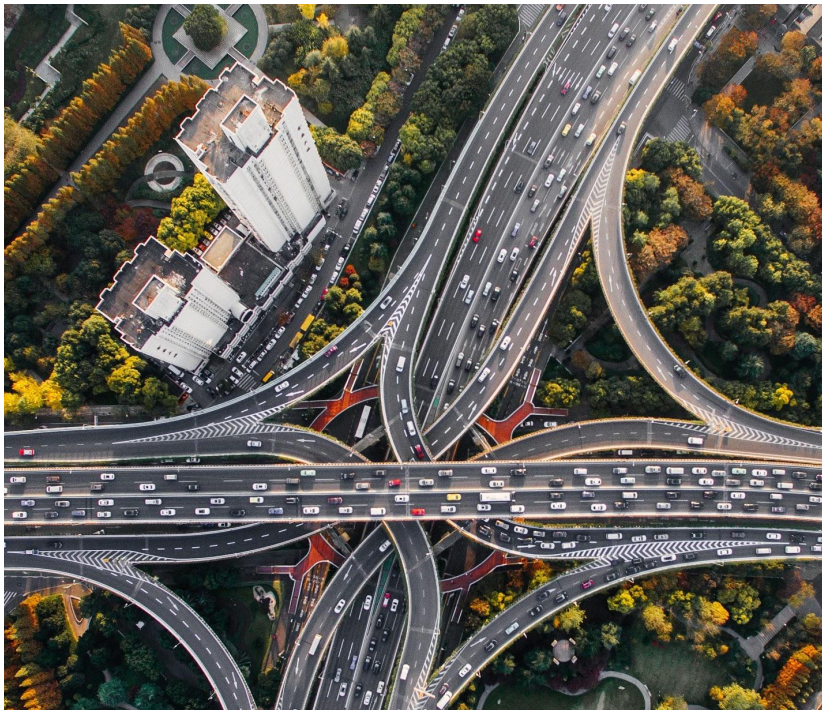


i.

# Split Up Your Jobs Manually

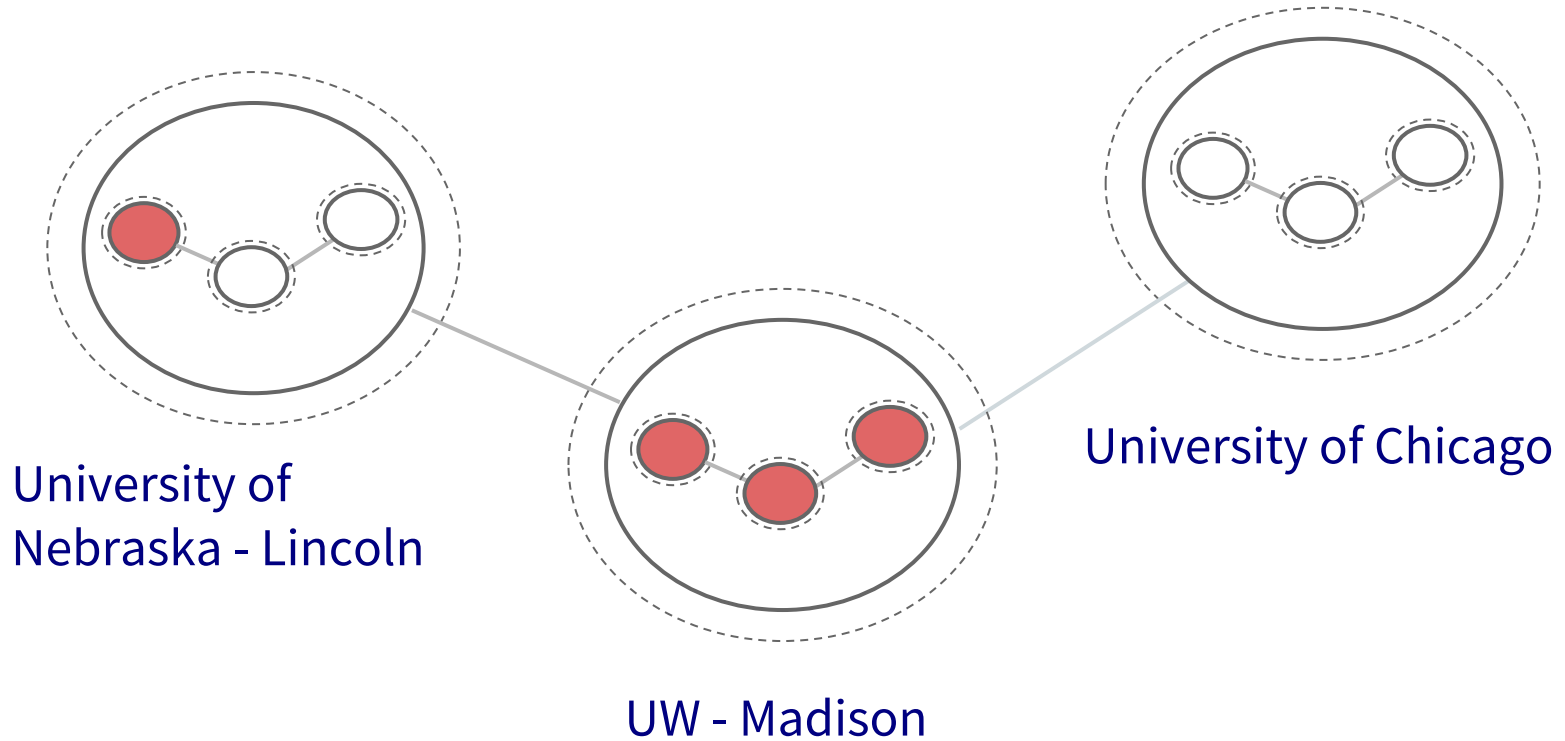
Let's start sharing!

# Manual Job Split

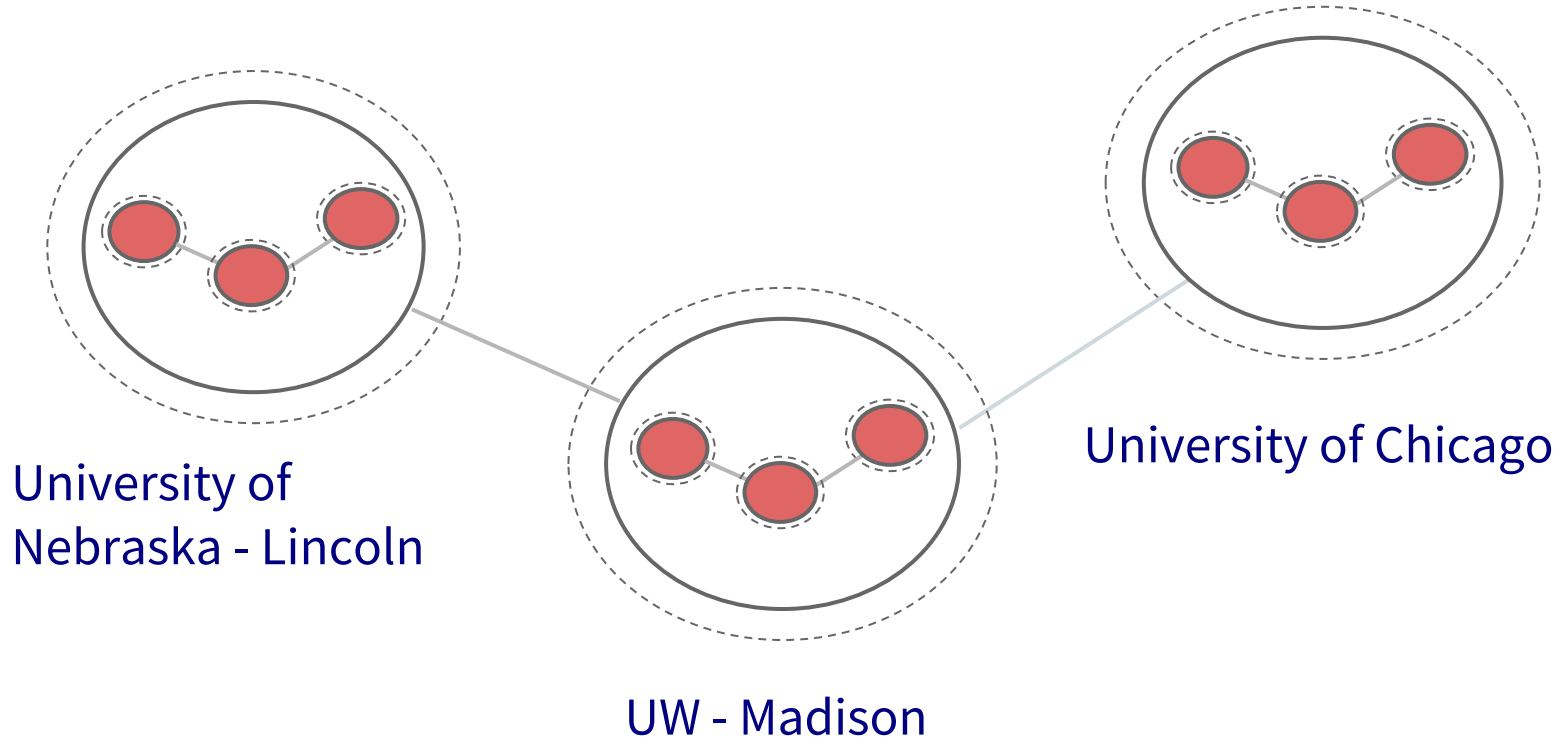


- Obtain login access
- Query each cluster for idle resources
- Split and submit jobs based on resource availability

# #3: Share Resources - Distributed HTC



# #3: Share Resources - Distributed HTC



# Manual Job Split - Shortcomings

---

- Fewer logins = fewer potential resources
- More logins = more account management
- Why would they give you accounts? Are your friends going to want CHTC accounts?
- Querying and splitting jobs is tedious and inaccurate
- Not all clusters use HTCondor — other job schedulers e.g., SLURM, PBS, etc.
- Pools are independent — workflows must be confined to a single pool





**ii.**

# **Split Up Your Jobs Automatically**

Let the computers do the work

# Automatic Job Split - Shortcomings



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

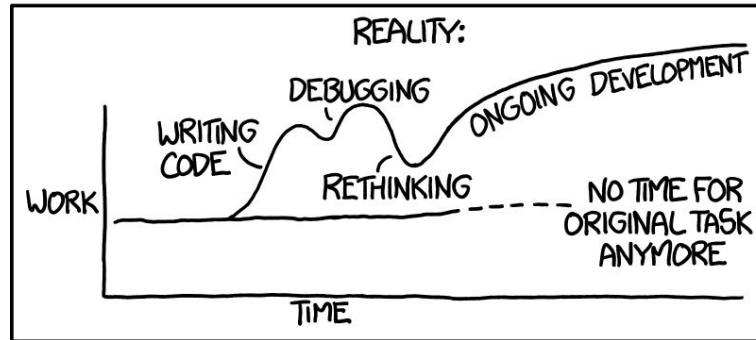
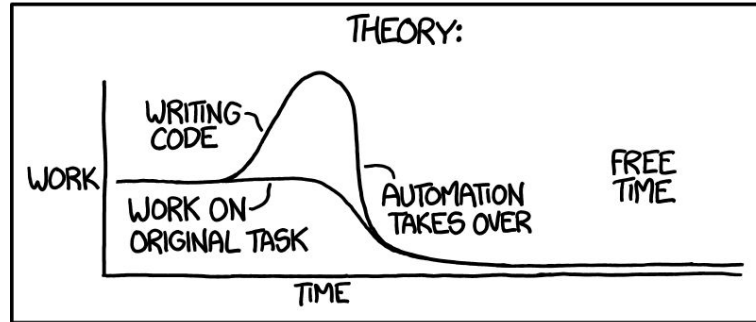
**Bart:** Isn't that the wrong way?

**Homer:** Yeah, but faster!

Groening, M (Writer), Michels, P. (Director) . (1999).  
Homer to the Max [Television Series Episode]. In  
Scully, M. (Executive Producer), *The Simpsons*. Los  
Angeles, CA: Gracie Films

# Automatic Partitions - Shortcomings

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



# #3: Share Resources - Requirements

---

- Minimal account management
- No job splitting
- DAG workflow functionality
- HTCondor only!
- No resource sharing requirements

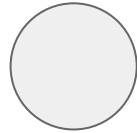
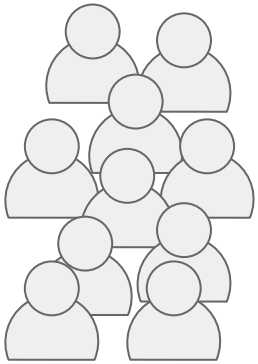


**iii.**

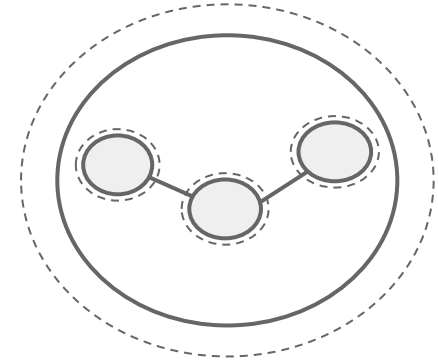
# **Overlay Systems**

Let the OSG do the heavy lifting

# The OSG Model

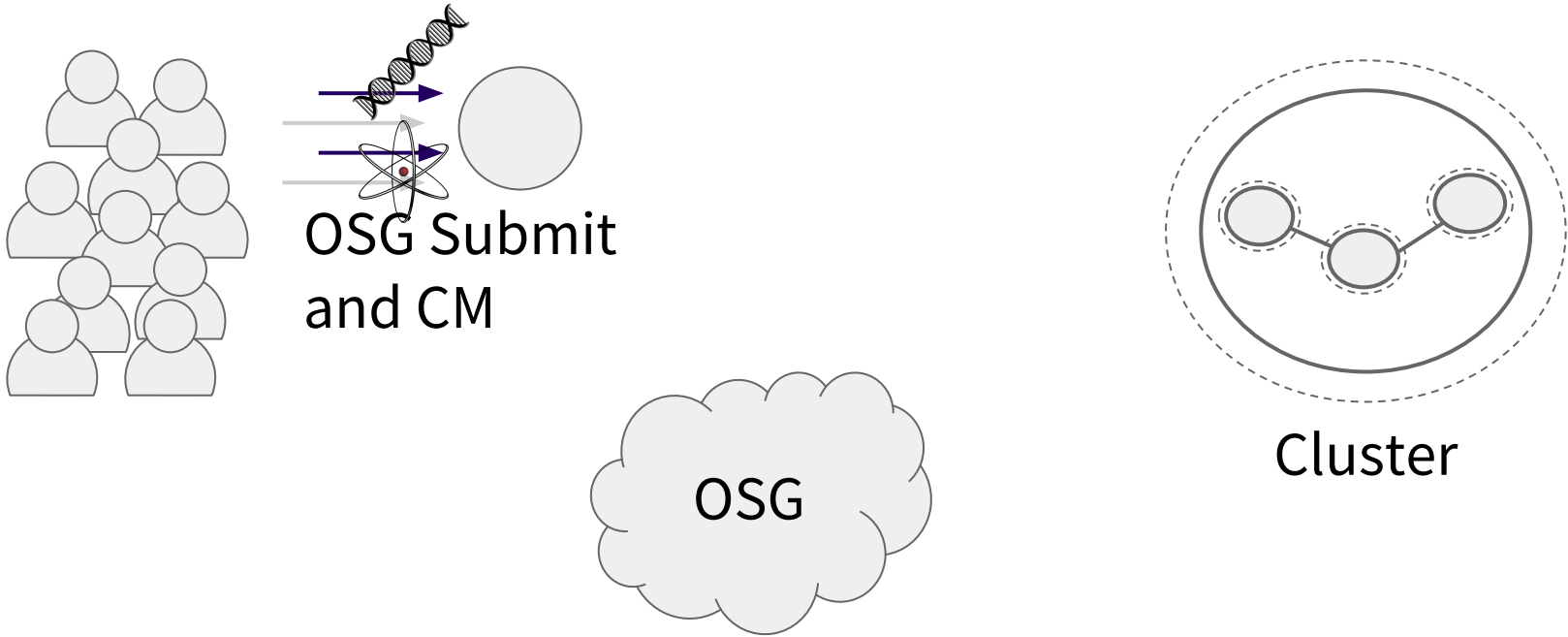


OSG Submit  
and CM

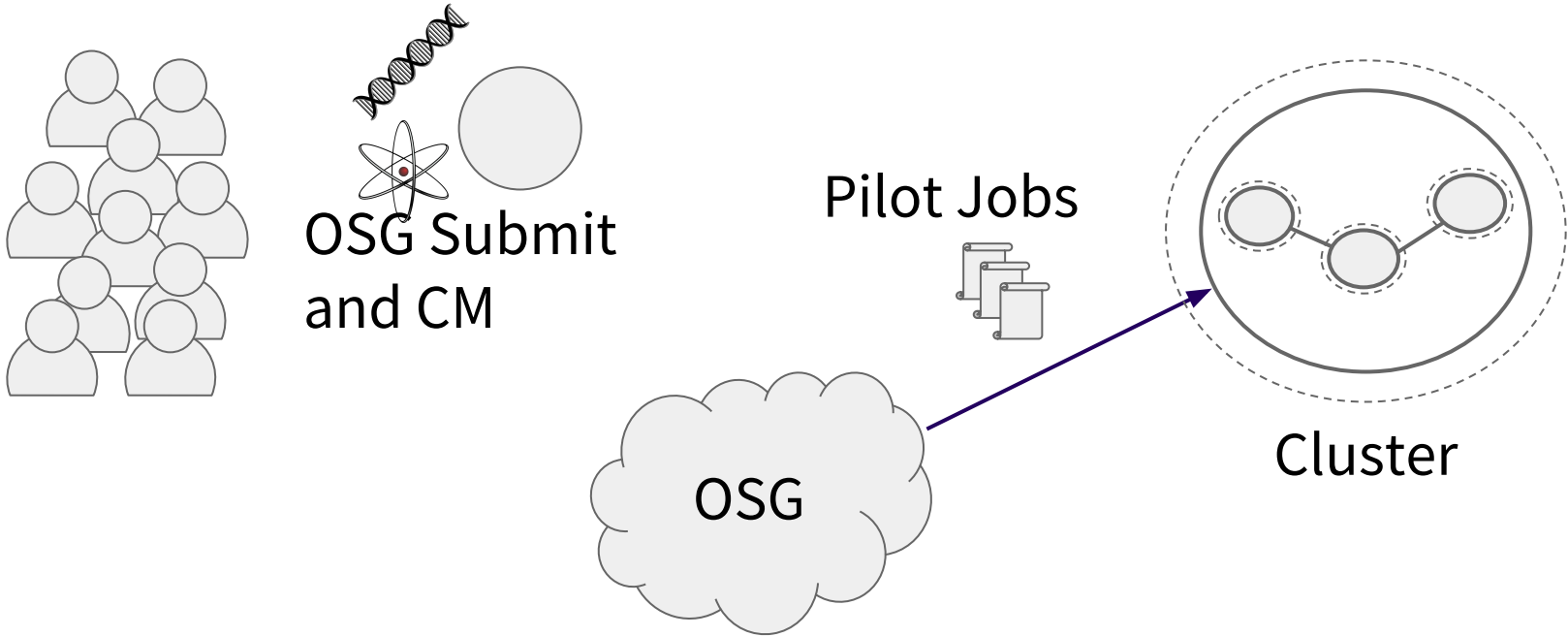


Cluster

# The OSG Model

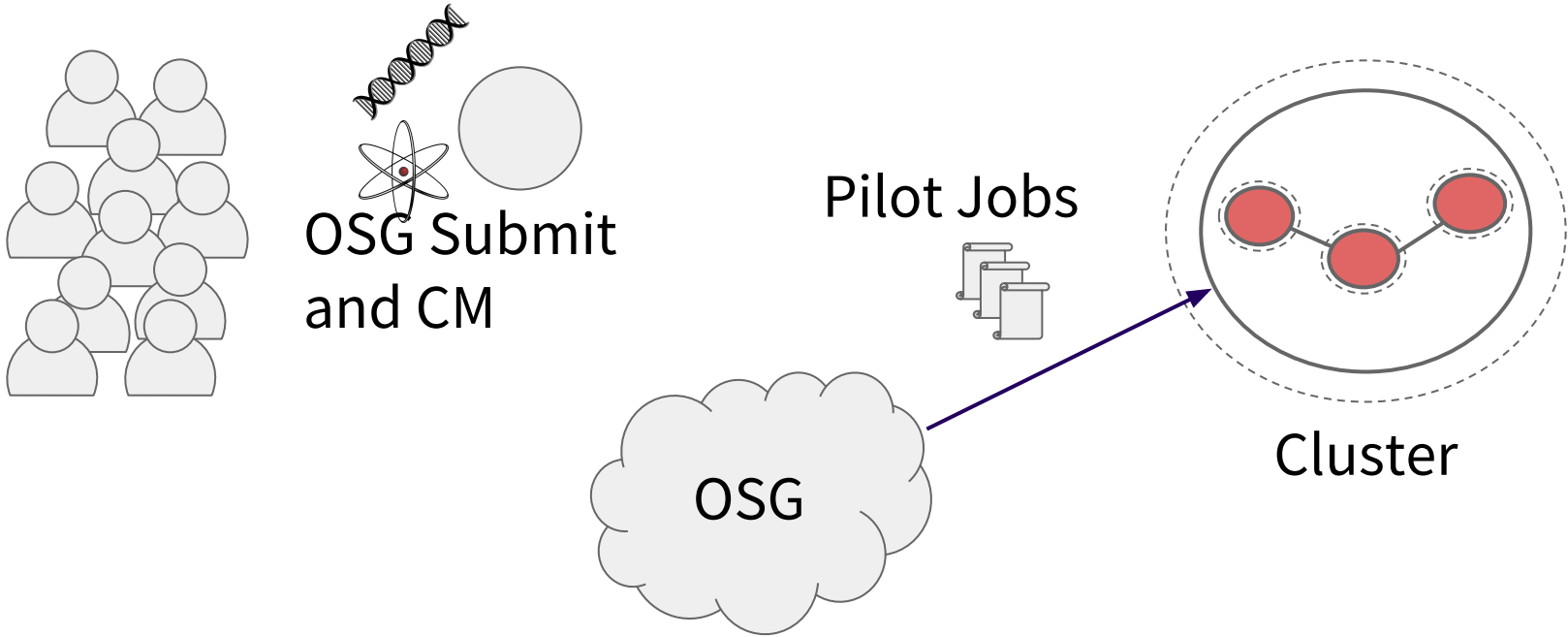


# The OSG Model



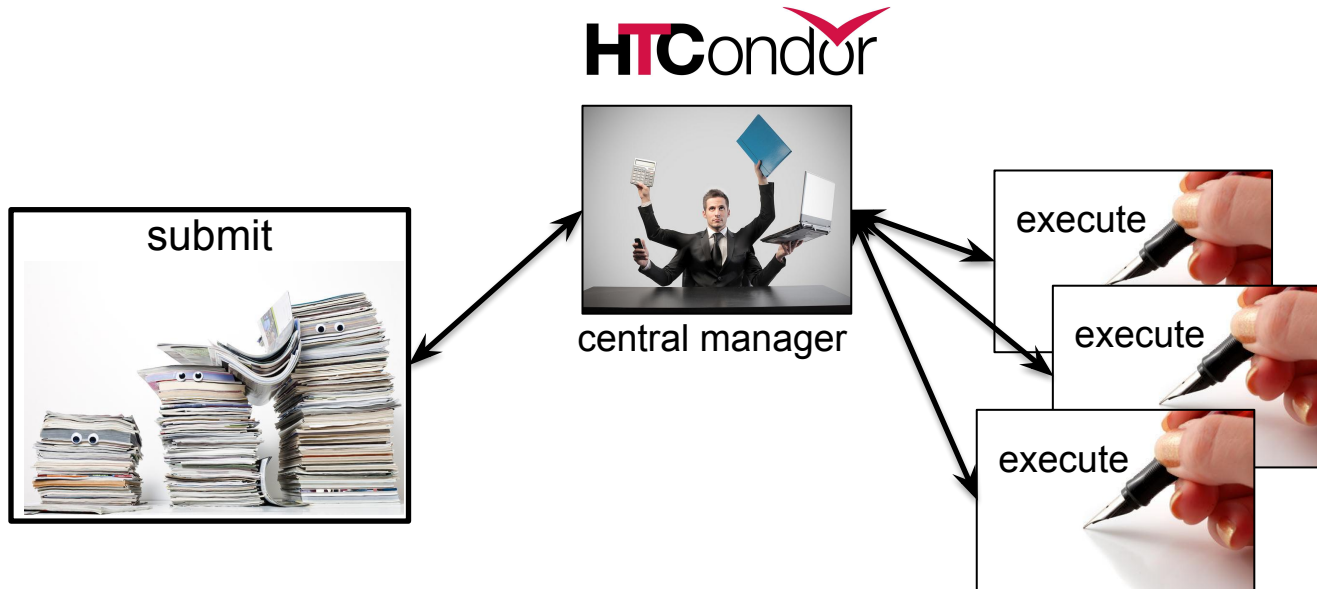


# The OSG Model

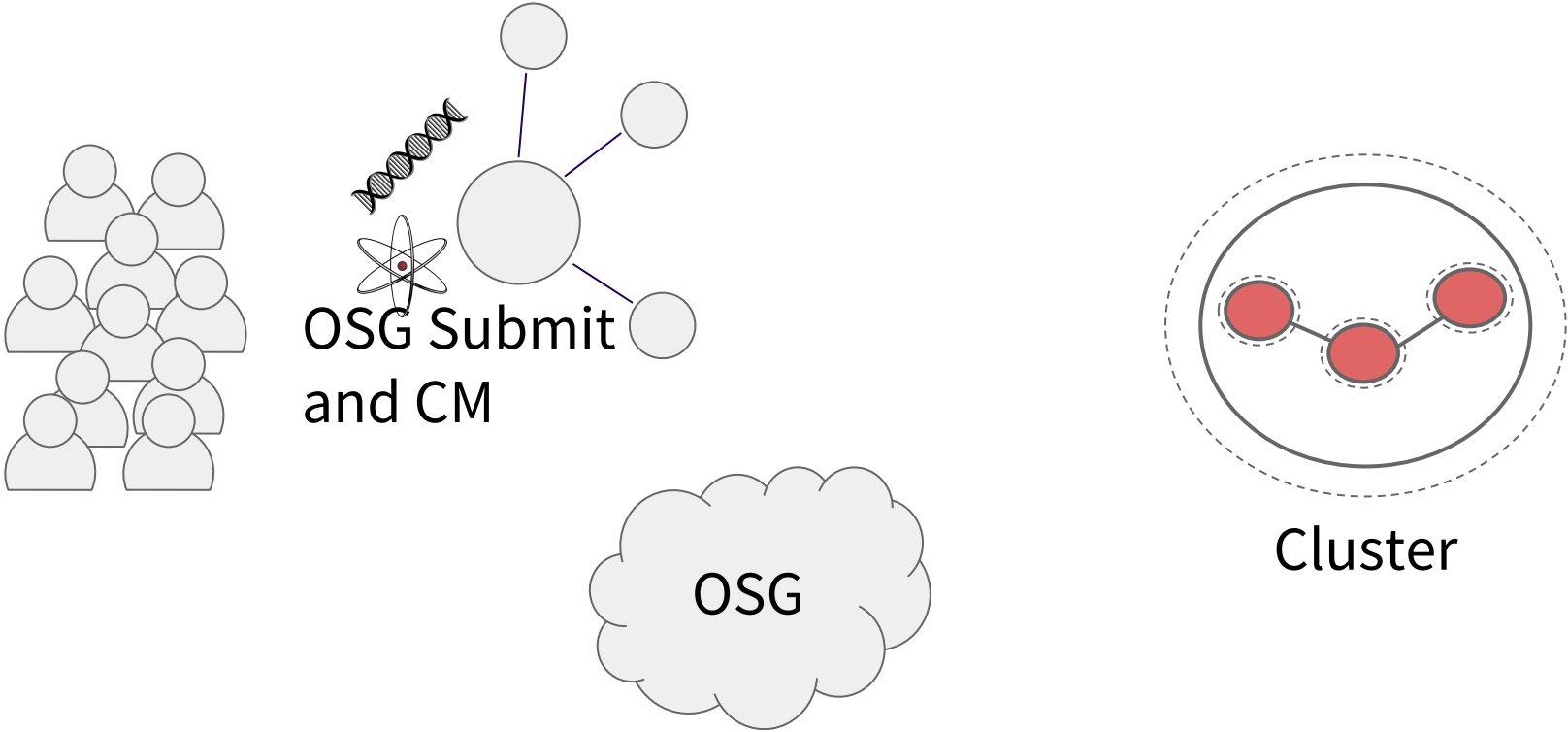


# Job Matching

- 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)

# #3: Share Resources - Requirements

---

- Minimal account management: only one submit server
- No job splitting: only one HTCondor pool
- DAG workflow functionality: Only one HTCondor pool
- HTCondor only: Only one HTCondor pool
- No resource sharing requirements: the OSG doesn't require that users "pay into" the OSG

# The OSG Model - Recap

---

- 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
- Some of your OSG jobs may 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 run 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



# Questions?