



Training Outreach and Education

<http://www.nesc.ac.uk/training>



<http://www.ngs.ac.uk>

Concepts of Condor and Condor-G

Guy Warner

PPARC

<http://www.pparc.ac.uk/>

eGEE

<http://www.eu-egee.org/>

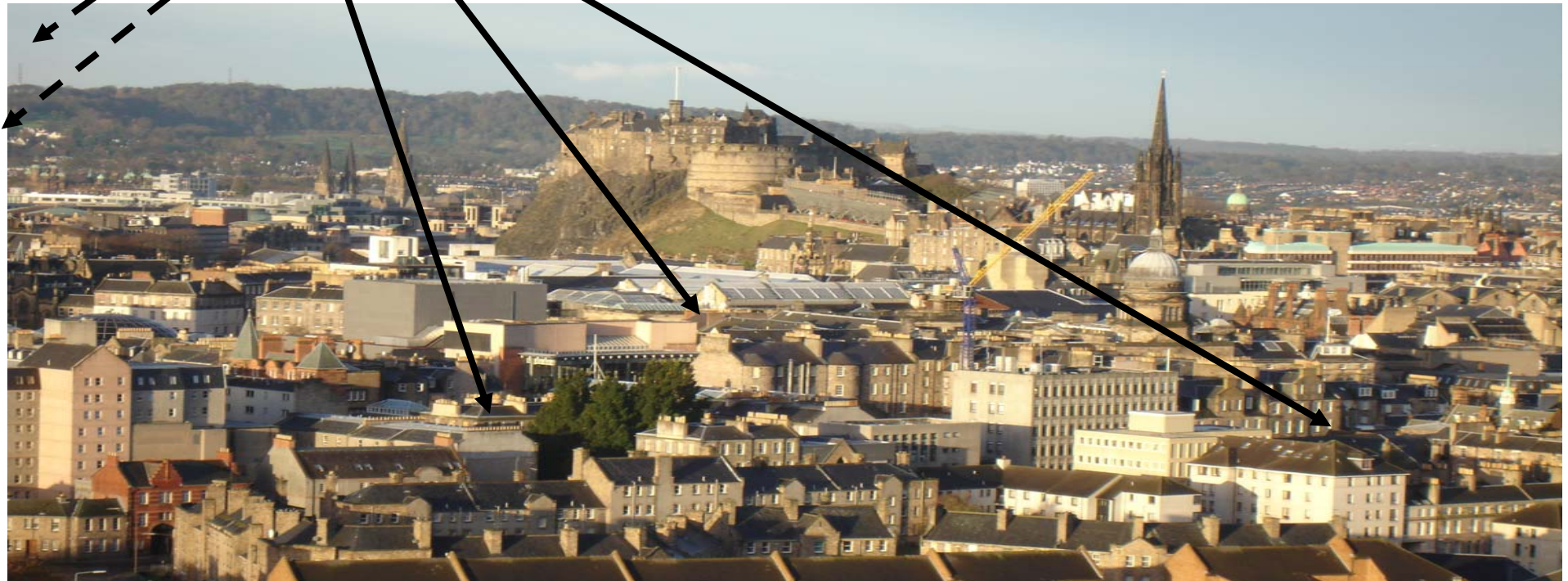


Harvesting CPU time

Often-idle processors!!

Analyses constrained by CPU time!

**Teaching labs. +
Researchers**





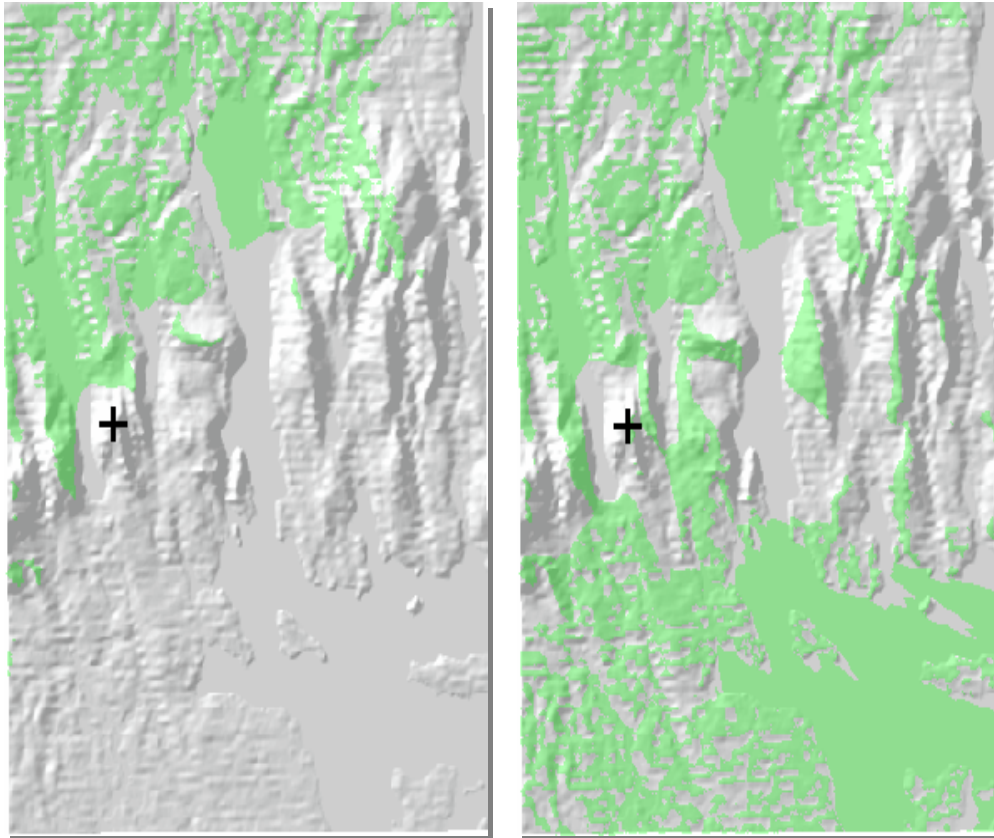
Harvesting CPU time

- Teaching lab machines lie idle for most of the time
- Harvest spare compute cycles to create a low-cost “high throughput computing” (HTC) platform
 - Goal: run many tasks in a week, month, ...
 - Typically: many similar tasks invoked from workflow or a script
 - Monte-Carlo
 - Simulation – parameter sweeps
- Pool processors as a batch processing resource
- Submit jobs that run when a machine is free
- Condor most common approach
 - <http://www.cs.wisc.edu/condor/>



Example: viewshed analyses

Viewsheds: what can be seen from point at “+”



- Derive viewsheds for all points in “digital elevation model” (DEM)
- Build a database to allow
 - Derivation of indices to characterise viewsheds
 - Applications to access pre-calculated viewsheds

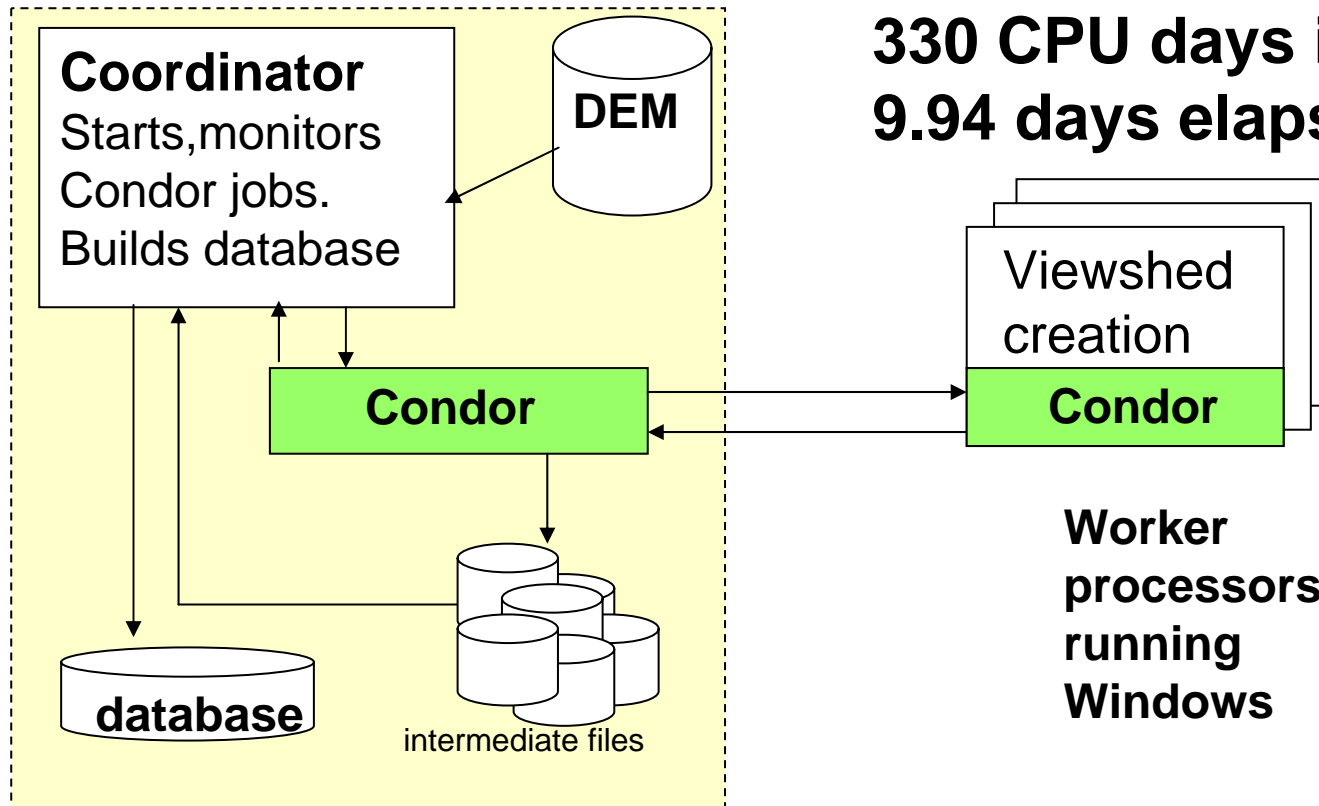
Mineter, Dowers, Caldwell



Example: viewshed analyses

Typical run:

39 PCs (Windows)
330 CPU days in
9.94 days elapsed



Coordinating processor

**Worker
processors
running
Windows**

Mineter, Dowers, Caldwell



Condor

- Converts collections of distributed workstations and dedicated clusters into a high-throughput computing (HTC) facility
 - Condor Pool
- Manages both resources (machines) and resource requests (jobs)
- Cycle scavenging
 - Using a non-dedicated resource when it would otherwise be idle.



Terminology

- Cluster – a dedicated set of computers not for interactive use (definition by Alain Roy)
- Pool – a collection of computers used by Condor
 - May or may not be dedicated
 - Single administrative domain
- Central Manager – one per pool
 - Matches resource requests to resources (Matchmaking)
 - Pool Management
- (Flocking – running jobs submitted from pool A on pool B
 - Sharing resources with administrative domains possibly with user prioritization)



Architecture

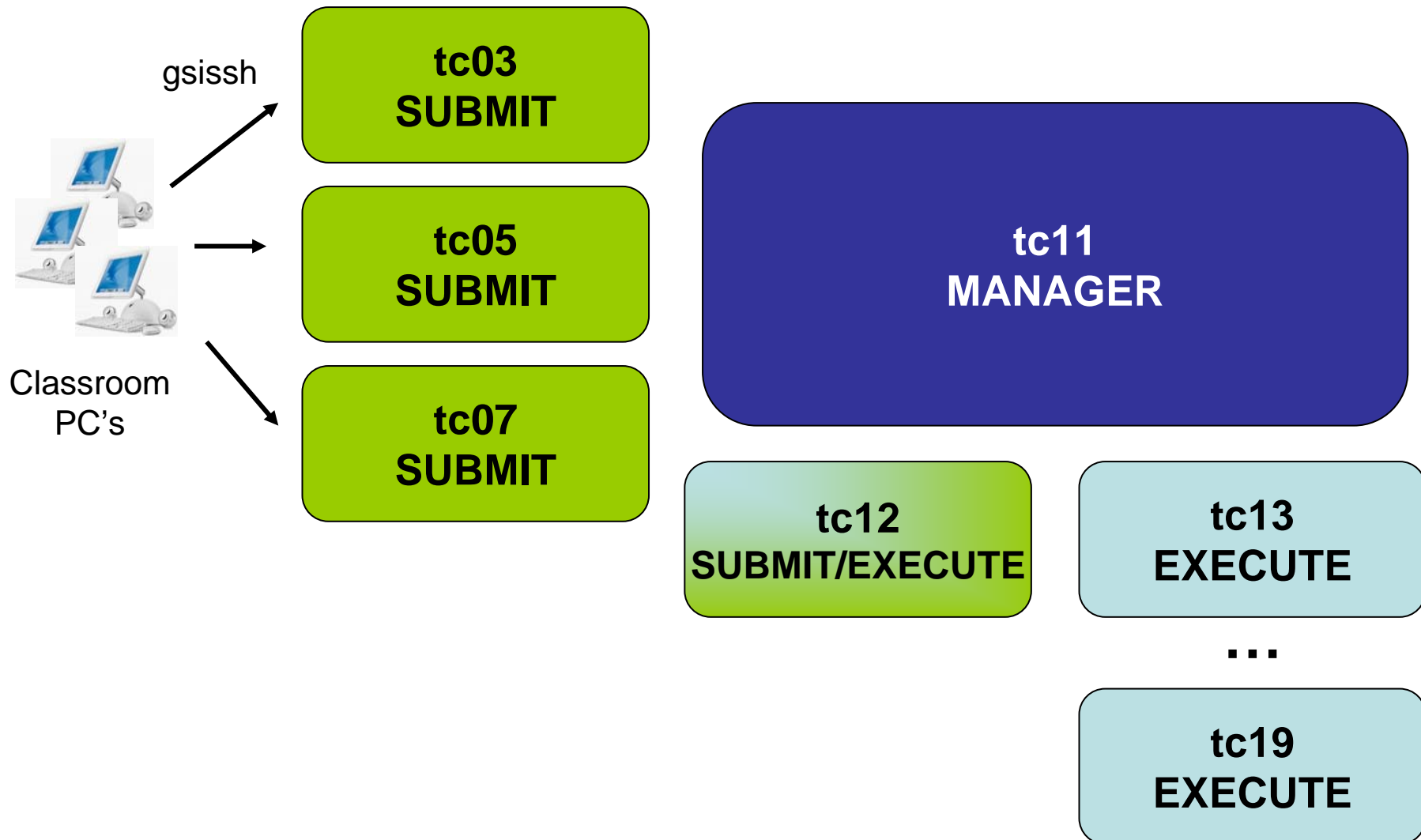
- All nodes run `condor_master`
 - Responsible for control of a node.
- The Central Manager additionally runs `condor_collector` and `condor_negotiator`
 - Responsible for matchmaking
- An Execute Node additionally runs `condor_startd`
 - Responsible for starting a job
- A Submit Node additionally runs `condor_schedd`
 - Responsible for submitting jobs (and allowing user to monitor jobs)
- A Node must be at least one of Manager/Execute/Submit but may be more



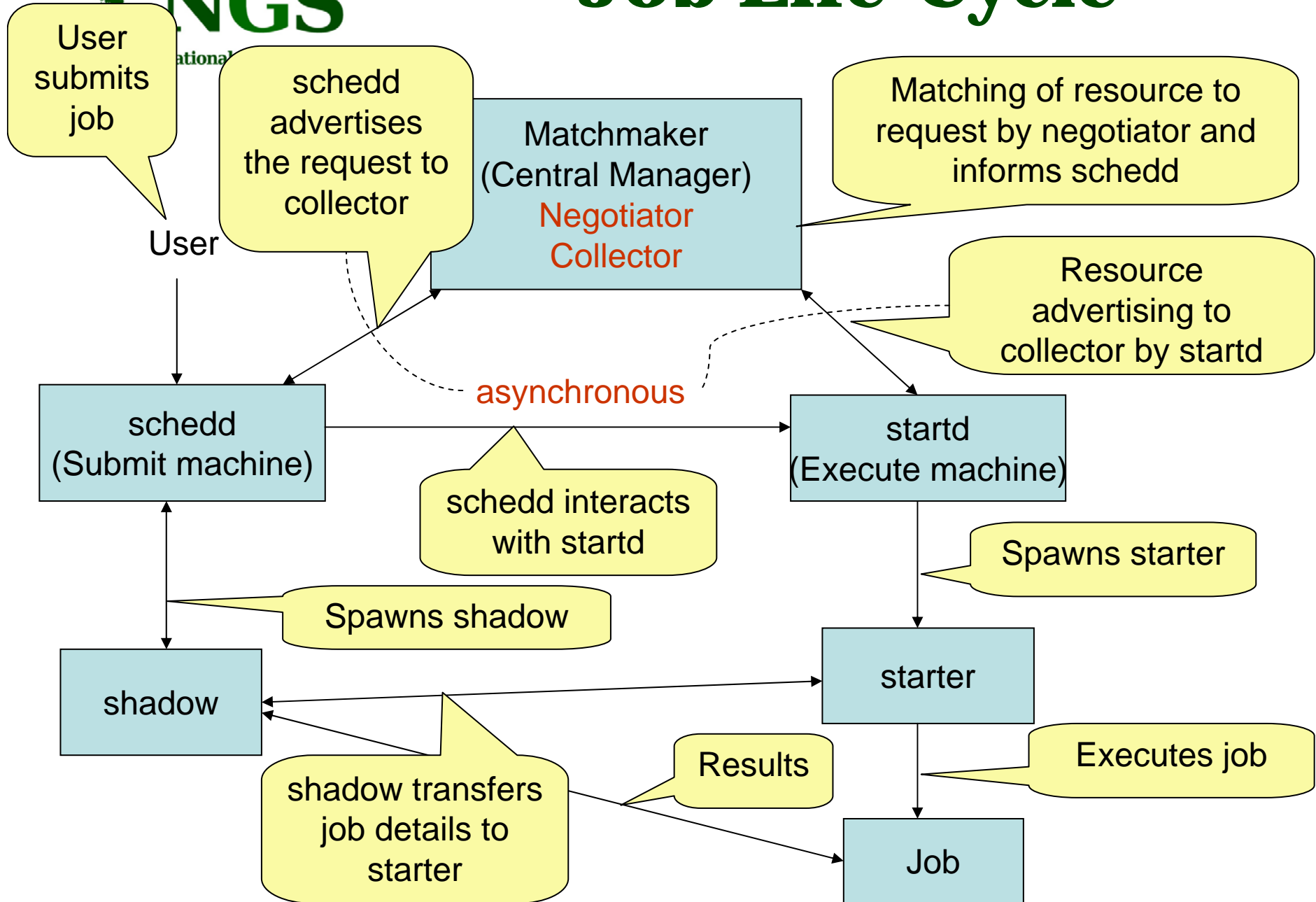
Example Configurations

- Personal Condor – all services on one node
 - Gain benefits of Condors job management
 - E.g. only run jobs on your desktop PC when you are not using it.
- Dedicated Cluster – Manager/Submit on head-node all other nodes are Execute.
 - Users ssh to head-node.
- Shared workstations – one workstation dedicated as Manager, all others as Submit/Execute
 - Submission from any workstation (except Manager)
 - Nodes join the pool as Execute Nodes when idle and leave the pool when Keyboard/Mouse activity detected.

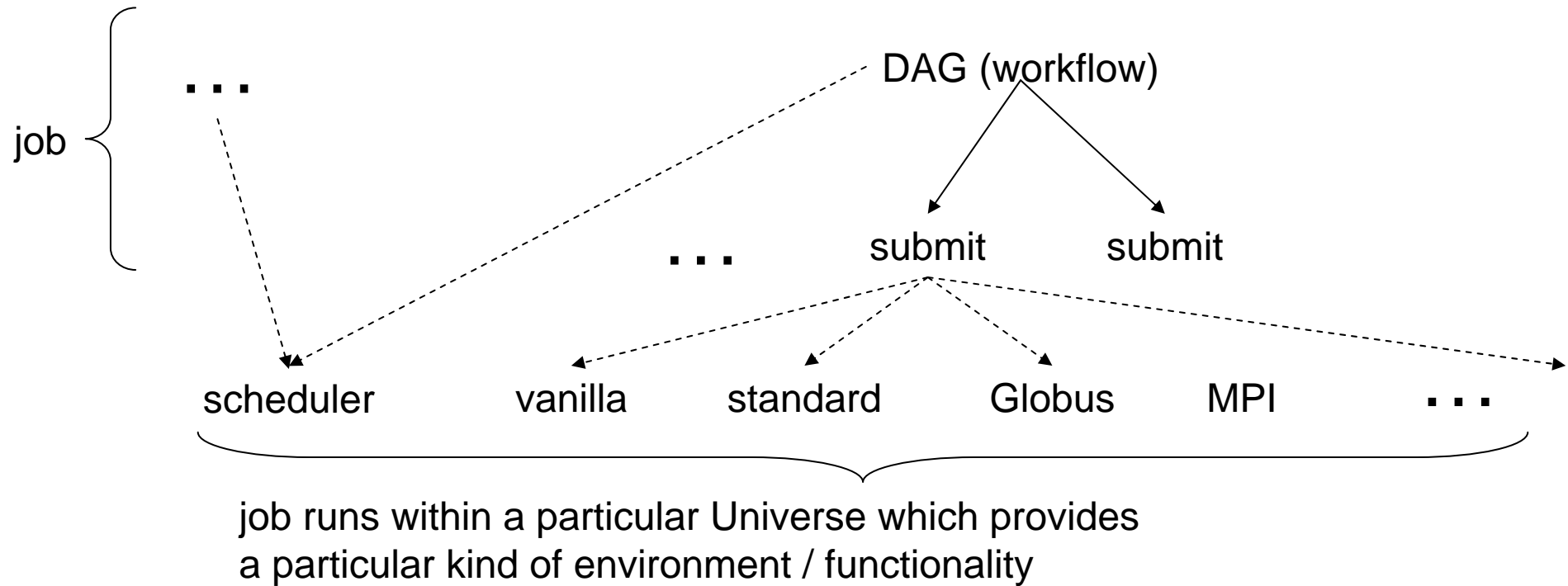
Our setup



Job Life-Cycle



Job types – “Universes”





Some Universes

- Vanilla
 - Any job that does not require the features of the other universes

- Standard
 - goal – to reproduce home Unix environment
 - emulates standard system calls
 - file I/O
 - » remote access to home files
 - signal routing
 - resource management
 - + checkpointing
 - save the entire state into checkpoint file
 - can then restart from there – possibly on a different machine
 - for
 - migration
 - backwards error recovery
 - important for very long running jobs



Some More Universes

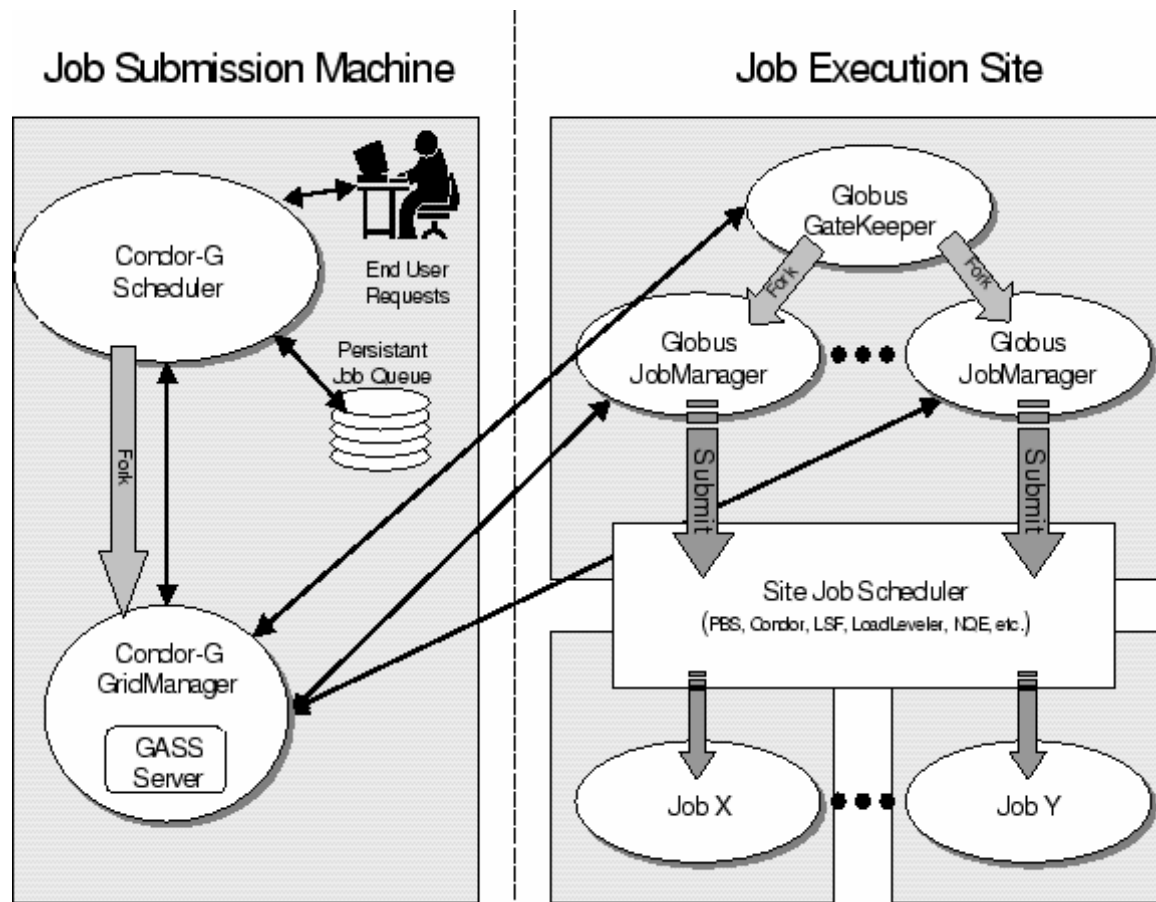
- Java
 - to provide a standard Java environment
- MPI
 - to allow use of message passing interface between component processes
- Scheduler
 - For running a job that schedules other jobs
 - Standard is DAGMAN
- Globus – “Condor-G” –
 - access to Globus Job Manager



Condor-G

- Tool that provides globus universe – Condor-G
- The same Condor tools that access local resources are now able to use the Globus protocols to access resources at multiple sites.
 - One additional line in submit file
- Condor-G
 - manages both a queue of jobs and the resources from one or more sites where those jobs can execute.
 - communicates with these resources and transfers files to and from these resources using Globus mechanisms.
 - more than just a replacement for globusrun

Condor & Globus

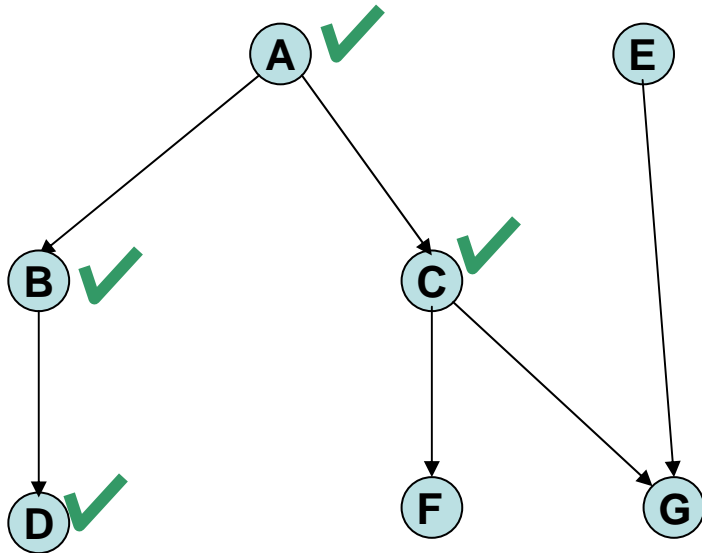


Condor uses the following Globus protocols

- **GSI** – for authentication and authorization
- **GRAM** – protocol that Condor-G uses to talk to remote Globus jobmanagers.
- **GASS** – to transfer the executable, stdin, stdout, and stderr between the machine where a job is submitted and the remote resource.
- **RSL** – to specify job information.

DAGMAN

- Directed Acyclic Graph Manager
 - specify dependencies as a DAG
 - re-start a partially completed DAG
 - records what has been done



**Each vertex (DAG node) is
a normal submit command
For any universe
(most appropriate for a
scheduler universe)**

**Each arc is a sequencing constraint -
Parent must finish before child starts**



Match Making

- Matchmaking is fundamental to Condor
- Matchmaking is two-way
 - Job describes what it requires: I need Linux && 2 GB of RAM
 - Machine describes what it provides: I am a Mac
- Matchmaking allows preferences
 - I need Linux, and I prefer machines with more memory but will run on any machine you provide me
- Condor conceptually divides people into three groups:
 - Job submitters
 - Machine owners
 - Pool (cluster) administrator

} May or may not be the same people
- All three of these groups have preferences



ClassAds

- ClassAds state Facts
 - Submit-side e.g. My job's executable is analysis.exe
 - Execute-side
 - Dynamic e.g. My machine's load average is 5.6
 - Static e.g. My machine's operating system is Linux
- ClassAds state preferences:
 - Job submitter preferences e.g. I require a computer with Linux
 - Machine owner preferences e.g. I prefer jobs from the physics group
 - System Administrator preferences e.g. When can jobs pre-empt other jobs?



The Tutorial

- The tutorial covers:
 - Basic Condor Usage
 - Vanilla universe
 - Standard Universe
 - Condor-G
 - DAGMan
 - Matchmaking

<http://homepages.nesc.ac.uk/~gcw/NGS/AppDev/Condor.html>