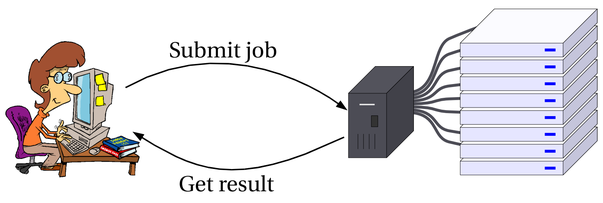
Pseudo-interactive monitoring in distributed computing

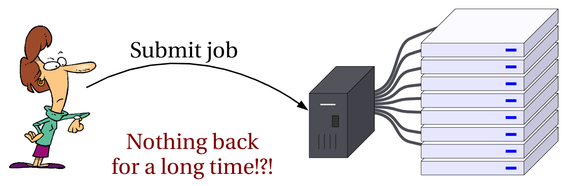
presented by **Dan Bradley**[1], Igor Sfiligoi[2] and Miron Livny[1]
[1]University of Wisconsin, Madison, [2]Fermi National Accelerator Laboratory

Distributed computing from the users' point of view

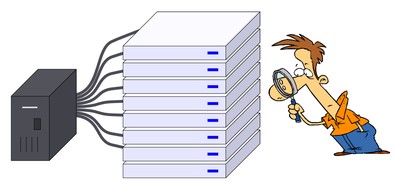
Users like the simplicity of batch submission.



… as long as everything works!



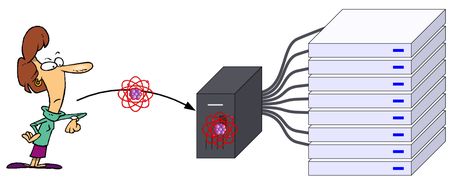
When things don't work, they want to understand why!



Types of monitoring information the users want

Most batch systems don't support this

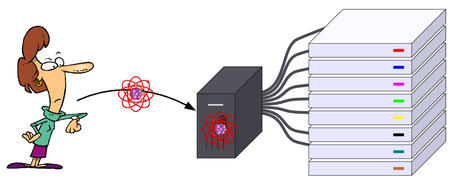
Is it running yet?



Condor: condor_q
PBS: qstat
Condor-G: condor_q

What is it waiting for?

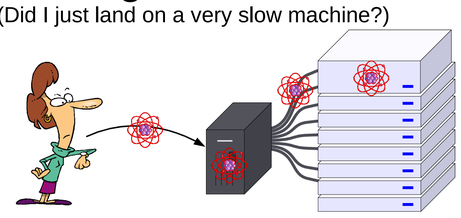
(Just too many people in front of me or did I do something wrong?)



Condor: condor_q -analyze
PBS: not accessible to users
Condor-G: limited info

What kind of node it is running on?

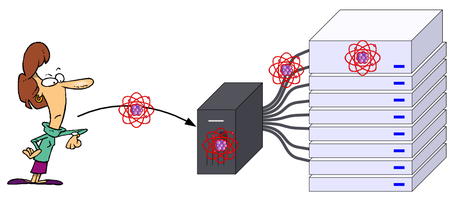
(Did I just land on a very slow machine?)



Condor: condor_status
PBS: pbsnodes
Condor-G: no clue

What it is doing right now?

(Is it using CPU? Is it waiting for a file to be downloaded? What do the log files say?)



Condor: only load and memory use
PBS: only load and memory use
Condor-G: no clue

Adding pseudo-interactive monitoring to batch systems

Users want to know:

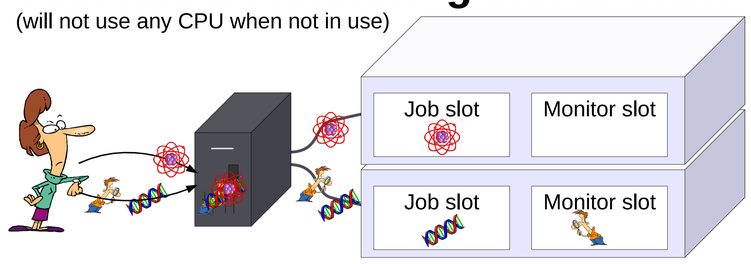
- What processes are runnning (ps)
- Peek at the log files (cat/tail)
- What files have been created (ls)
- Peek at the process stack (gdb bt)
- Is the machine thrashing? (top)

Users would like interactive access.

But all the listed operations can run as batch commands.

Just add a monitoring slot to each CPU

(will not use any CPU when not in use)



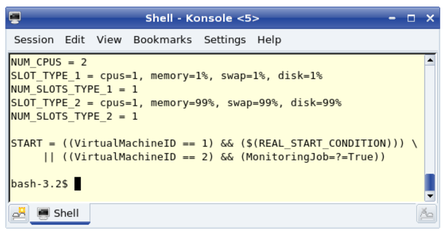
Same authentication and permissions as regular job.

Monitoring command just a targeted batch job.
Will start immediately since no competition.

Condor implementation

Worker node configuration

(assuming single CPU, for multiple CPUs very similar)

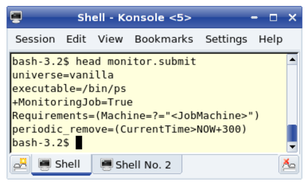


Enable second slot and reserve it for monitoring jobs

Pseudo-interactive command

Multi step process:

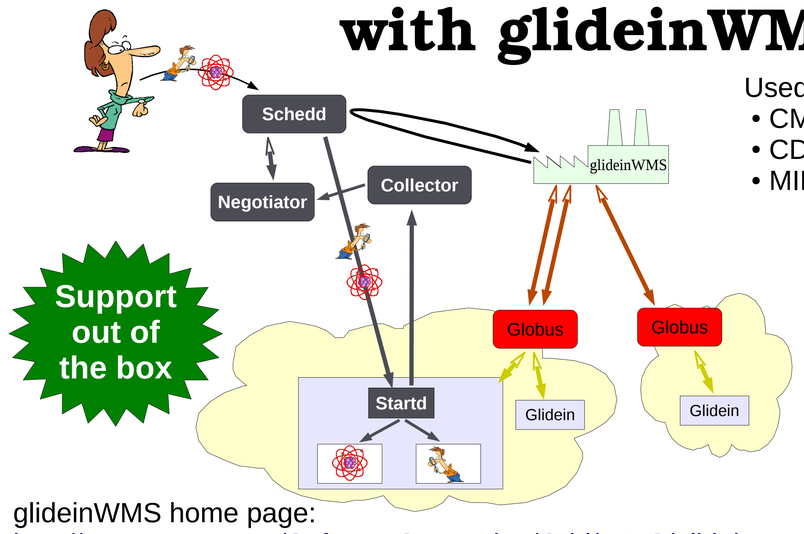
- Get the machine name where the job is running
- Prepare the submit file
  (declare it as a monitoring job and specify the machine name)
- Submit the monitoring job
- Wait for it to finish
- Read the output file



Should use a wrapper

Condor home page: http://www.cs.wisc.edu/condor/

Grid job monitoring with glideinWMS



Used by:

- CMS
- CDF
- MINOS

Support out of the box

glideinWMS home page:
http://www.uscms.org/SoftwareComputing/Grid/WMS/glideinWMS/

Work partially supported by the U.S. DoE under contract No. DE-AC02-07CH11359, and by the U.S. NSF grants PHY-0427113 (RACE) and PHY-0533280 (DISUN).