

Experiences with a HTCondor pool: Prepare to be underwhelmed

C. J. Lingwood, Lancaster University



CCB (The Condor Connection Broker) – Dan Bradley

<http://research.cs.wisc.edu/htcondor/>

Introduction to HTCondor

- High throughput rather than high performance.
- Shares computer power
- Only one dedicated machine
 - Each node is a office PC

This, no doubt, is a condor!

Seagull?



HTCondor
High Throughput Computing



What is it for?

- Dave has a large simulation which takes too long to run on his desktop.
- Dave wants a proper cluster.
- Julie has lots of little simulations which don't take long but take too long to run on her desktop.
- Julie could use a cluster but the Daves would be cross. She could use a **condor pool**.
- Dave asks “**How fast?**”
- Julie asks “**How many?**”

Why bother?

- Why don't we just buy cluster time for Julie?
 - We could, but Julie is poor
- Condor runs on desktop machines when they are idle
- A condor pool is free(ish)
- Condor can manage the power
 - Turn it off if someone isn't using it after a bit
- It's a little friendlier than a cluster.
 - Launch things from your own machine
 - It copies files to and from your machine
 - No login
 - Mostly windows nodes (useful for windows only codes)

Why bother?

Its free

Apart from the electricity

- Poor neglected pool (35ish cores)
 - 30 ish Windows Nodes
 - 5 ish Linux Nodes

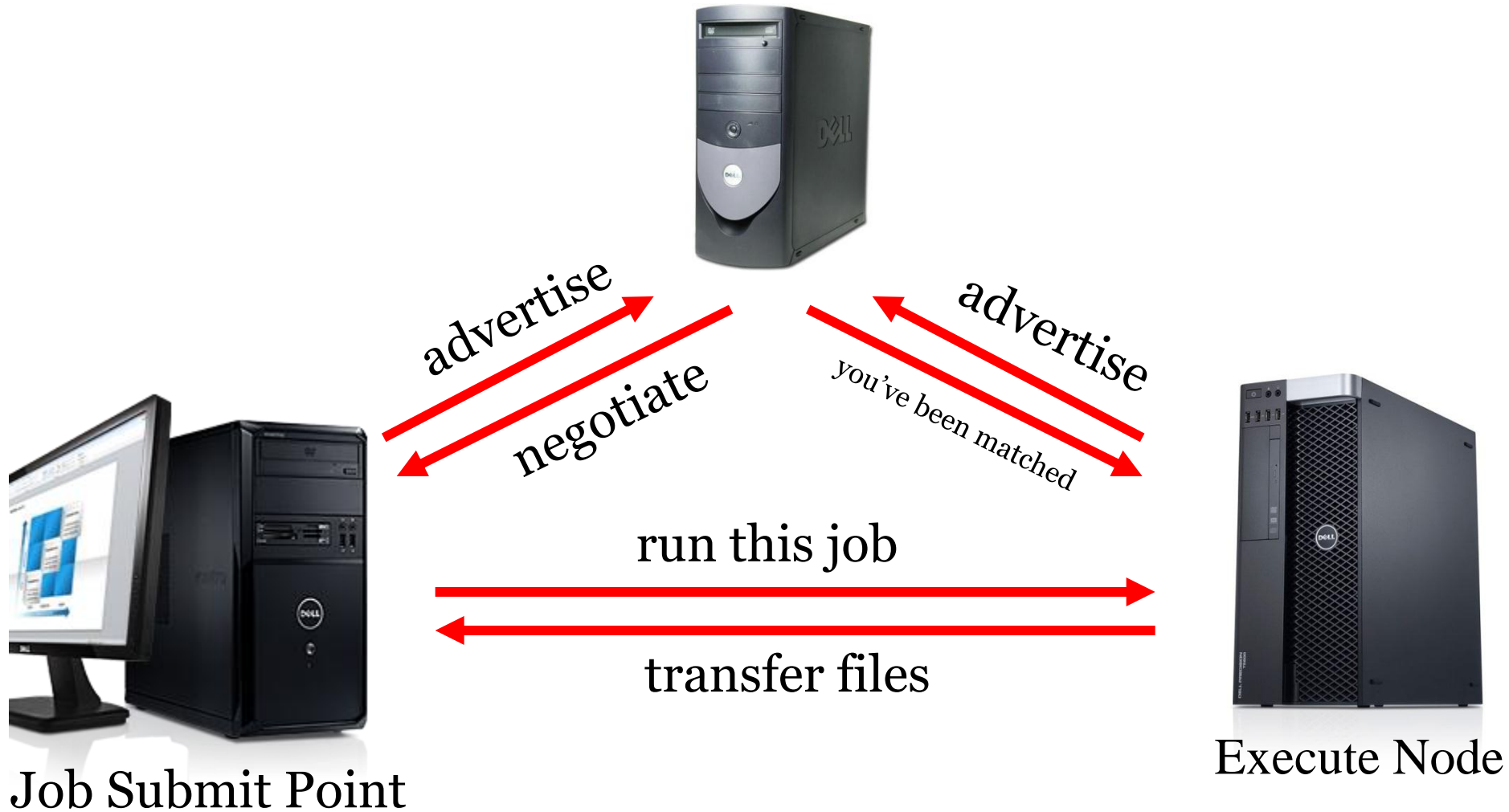


Setup originally by Jonny Smith

Our master server
Powered by coal and steam

Structure of a pool

Central Manager (Match maker)



Handy commands

- `condor_submit` (submit a job)
- `condor_status` (what machines are there)
- `condor_q` (list the jobs you have queued)
- `condor_rm` (remove a job)

Submitting a Job

```
Universe      = vanilla
Executable    = /home/nobody/condor/job.condor
Input         = job.stdin
Output        = job.stdout
Error         = job.stderr
Arguments     = -arg1 -arg2
InitialDir    = /home/nobody/condor/run_1
Queue
```

condor_submit submit_file.sub

Submitting lots of Jobs

```
Universe      = vanilla  
Executable    = this_job  
Arguments     = -arg1 -arg2  
InitialDir   = run_$(Process)  
Queue        600
```

Storage

- No storage as such
- Uses execute nodes storage when running
- Your machine when finished
- Can launch a script to copy from/to a NAS

Case Study: Optimising Klystrons

- The optimisation of a new klystron interaction structure is a many dimensional multi-objective problem.
- Some well defined decisions.
- Some ill defined decisions.
 - Cavity frequencies
 - drift lengths between cavities
 - External coupling
- Objectives
 - High Efficiency, output power, gain
 - Short length of interaction
 - Specification Bandwidth
 - Avoid reflected electrons
 - Etc...

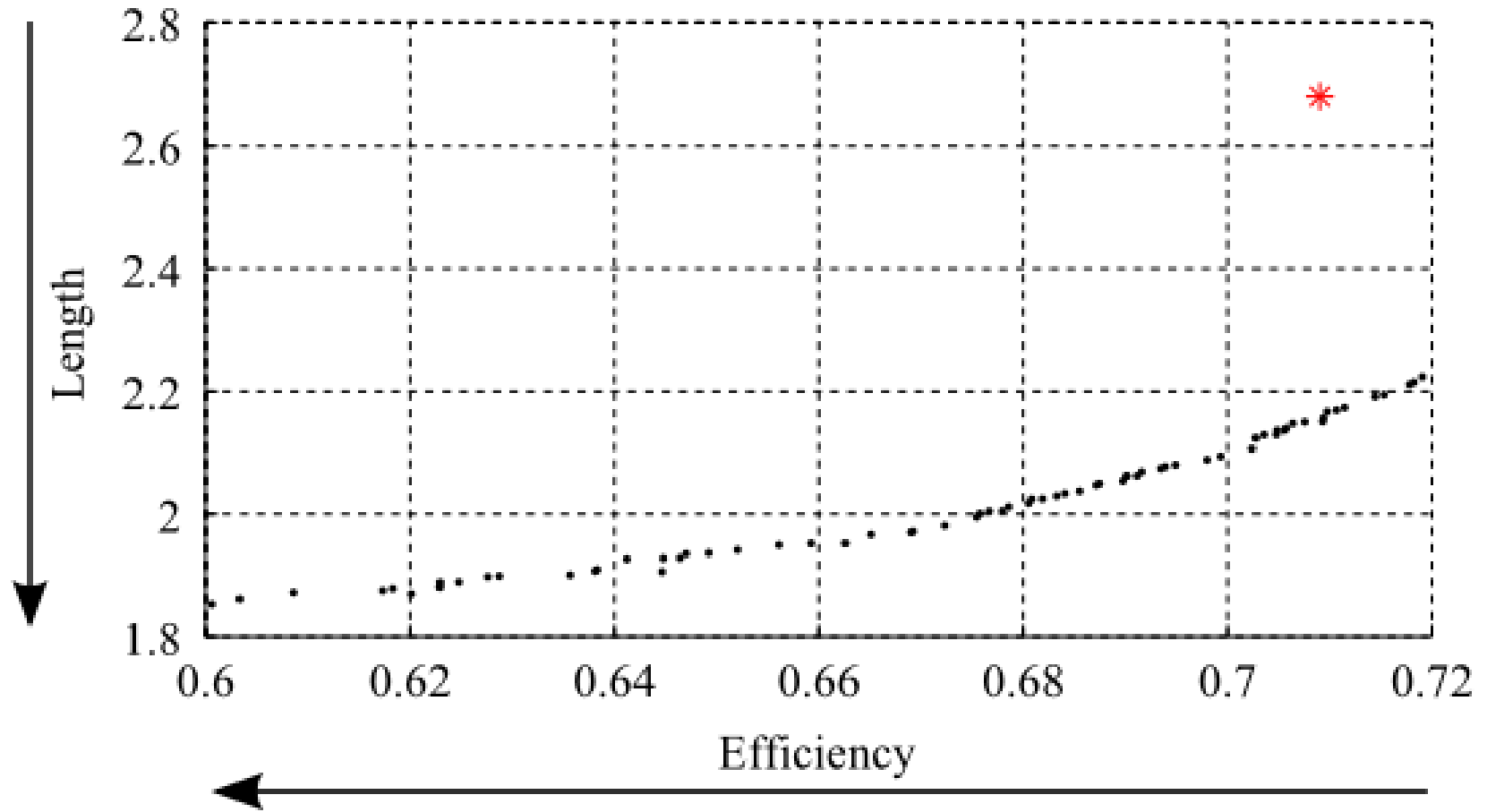
Case Study: Optimising Klystrons

- You could optimise them by hand
 - Tedious
- You could let the computer do all the hard work for you
 - Multi objective optimiser
- Lot's of little simulations (it turns out my name is Julie)
- Admins of clusters like few long large jobs, they don't like short "high frequency" job submissions.
- Data analysis "on the fly" in the EA

Impact of condor

- 10,000 simulations per run (at least 3 different configurations)
- 3 Minutes per simulation (some take 8-10)
- One machine 20 Days
- 60 machines 8 hours

Did we do better?



Demonstration

- Find the transfer curve of a klystron.
 - Vary the input power

Help!

- Clearly we need more execute nodes.
- More users probably wouldn't hurt either.
- email c.lingwood@lancs.ac.uk