High Throughput Parallel Computing (HTPC)

Dan Fraser, UChicago
Greg Thain, UWisc

Condor Week
April 13, 2010

The players

-Dan Fraser

Computation Inst.

University of Chicago

Miron Livny

U Wisconsin

John McGee

RENCI

Greg Thain

U Wisconsin

Funded by NSF-STCI

The two familiar HPC Models

- High Throughput Computing
  - Run ensembles of single core jobs
- Capability Computing
  - A few jobs parallelized over the whole system
  - Use whatever parallel s/w is on the system

- HTPC – an emerging model

Ensembles of small-way parallel jobs
(10's – 1000's)

Use whatever parallel s/w you want ☺
(It ships with the job)

Who's using HTPC?

- Oceanographers:
  - Brian Blanton, Howard Lander (RENCI)
    - Redrawing flood map boundaries
  - ADCIRC
    - Coastal circulation and storm surge model
    - Runs on 256+ cores, several days
  - Parameter sensitivity studies
    - Determine best settings for large runs
    - 220 jobs to determine optimal mesh size
    - Each job takes 8 processors, several hours

Tackling Four Problems

- Parallel job portability
- Effective use of multi-core technologies
- Identify suitable resources & submit jobs
- Job Management, tracking, accounting, …

Current plan of attack

- Force jobs to consume an entire processor
  - Today 4-8+ cores, tomorrow 32+ cores, …
  - Package jobs with a parallel library (schedd)
    - HTPC jobs as portable as any other job
    - MPI, OpenMP, your own scripts, …
    - Parallel libraries can be optimized for on-board memory access
  - All memory is available for efficient utilization
  - Submit the jobs via OSG (or Condor-G)

Problem areas

- Advertising HTPC capability on OSG
- Adapting OSG job submission/mgmt tools
  - GlideinWMS
- Ensure that Gratia accounting can identify jobs and apply the correct multiplier
- Support more HTPC scientists
- HTPC enable more sites

Configuring Condor for HTPC

- Two strategies:
  - Suspend/drain jobs to open HTPC slots
  - Hold empty cores until HTPC slot is open

Configuring Condor

```
# require that whole-machine jobs only match to Slot1
START = ($(START)) && (TARGET.RequiresWholeMachine =!= TRUE || SlotID
    == 1)
# have the machine advertise when it is running a whole-machine job
STARTD_JOB_EXPRS = $(STARTD_JOB_EXPRS) RequiresWholeMachine

# Export the job expr to all other slots
STARTD_SLOT_EXPRS = RequiresWholeMachine

# require that no single-cpu jobs may start when a whole-machine job is running
START = ($(START)) && (SlotID == 1 || Slot1_RequiresWholeMachine =!= True)

# suspend existing single-cpu jobs when there is a whole-machine job
SUSPEND = ($(SUSPEND)) || (SlotID != 1 && Slot1_RequiresWholeMachine =?=
    True)
CONTINUE = ( $(SUSPEND) =!= True )
```

Get all that?

http://condor-wiki.cs.wisc.edu

How to submit

```
universe = vanilla
requirements = (CAN_RUN_WHOLE_MACHINE =?= TRUE)
+RequiresWholeMachine=true
executable = some job
arguments = arguments
should_transfer_files = yes
when_to_transfer_output = on_exit
transfer_input_files = inputs
queue
```

MPI on Whole machine jobs

Whole machine mpi submit file

```
universe = vanilla
requirements = (CAN_RUN_WHOLE_MACHINE =?= TRUE)
+RequiresWholeMachine=true

executable = mpiexec

arguments = -np 8 real_exe

should_transfer_files = yes
when_to_transfer_output = on_exit

transfer_input_files = real_exe

queue
```

How to submit to OSG

```
universe = grid
GridResource = some_grid_host
GlobusRSL = MagicRSL
executable = wrapper.sh
arguments = arguments
should_transfer_files = yes
when_to_transfer_output = on_exit
transfer_input_files = inputs
transfer_output_files = output
queue
```

What's the magic RSL?

Site Specific

We're working on documents/standards

PBS

(host_xcount=1)(xcount=8)(queue=?)

LSF

(queue=?)(exclusive=1)

Condor

(condorsubmit=('+WholeMachine' true))

What's with the wrapper?

Chmod executable

Create output files

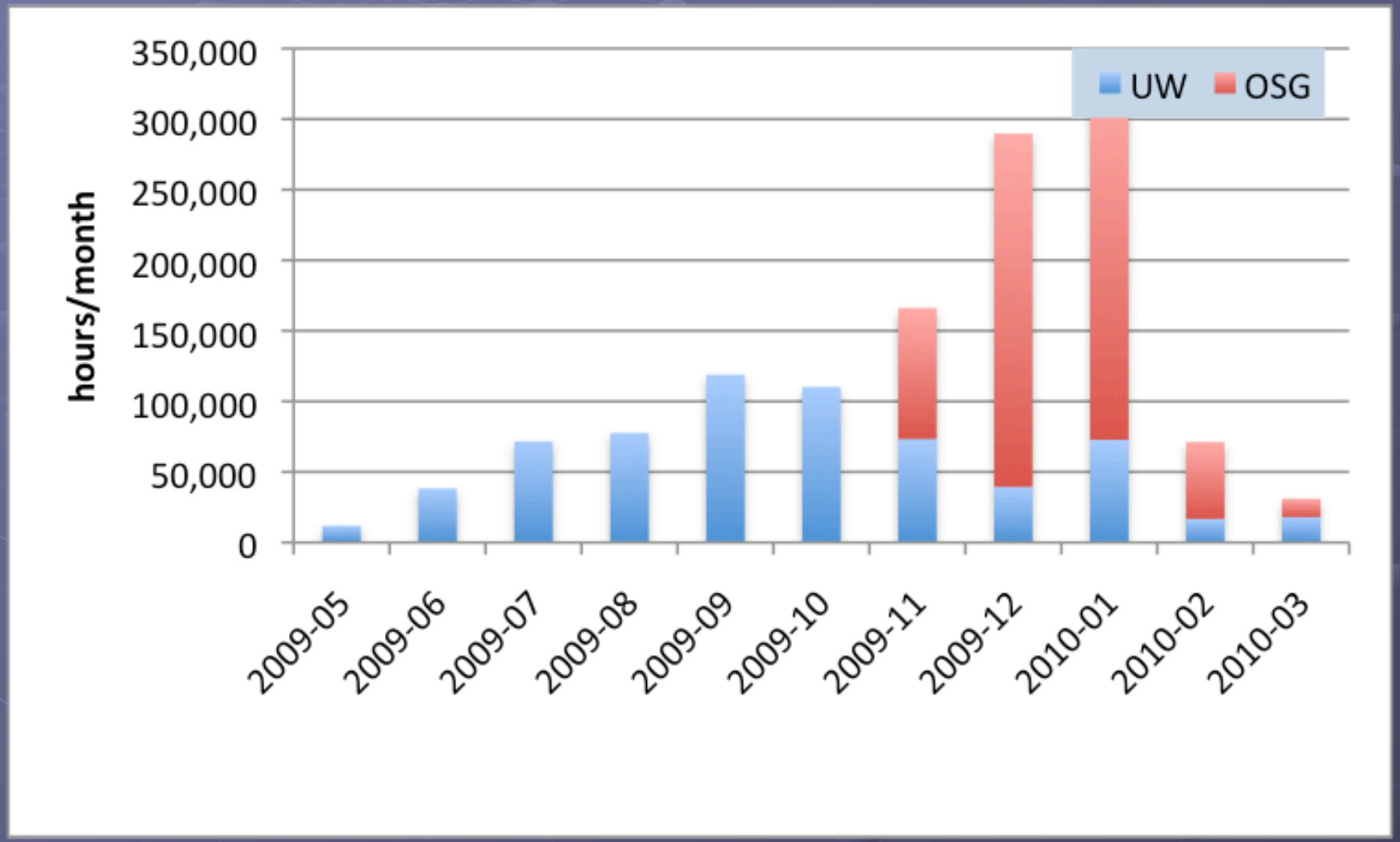
```
#!/bin/sh
chmod 0755 real.ex
touch output
./mpiexec -np 8 real.ex
```

Who's using HTPC?

- Chemists
  - UW Chemistry group
  - Gromacs
  - Jobs take 24 hours on 8 cores
  - Steady stream of 20-40 jobs/day
  - Peak usage is 320,000 hours per month
    - Written 9 papers in 10 months based on this

This could be you!

Chemistry Usage of HTPC



Current operations

- OU, slots based on priority
  - Logged over 2M HTPC hours so far
- Purdue, # of slots
- Clemson, # of slots
- San Diego, CMS T2, 1 slot

Your OSG site can be on this list!

Future Directions

- More Sites, more cycles!
- More users – any takers here?
- Use glide-in to homogenize access

Conclusions

- HTPC adds a new dimension to HPC computing – ensembles of parallel jobs
- This approach minimizes portability issues with parallel codes
- Keep same job submission model
- Not hypothetical – we're already running HTPC jobs
- Thanks to many helping hands