

HTPC - High Throughput Parallel Computing (on the OSG)

Dan Fraser, ANL/UChicago
OSG Production Coordinator

Greg Thain, UWisc

Brazil OSG School, Sao Paulo
Dec 10, 2010

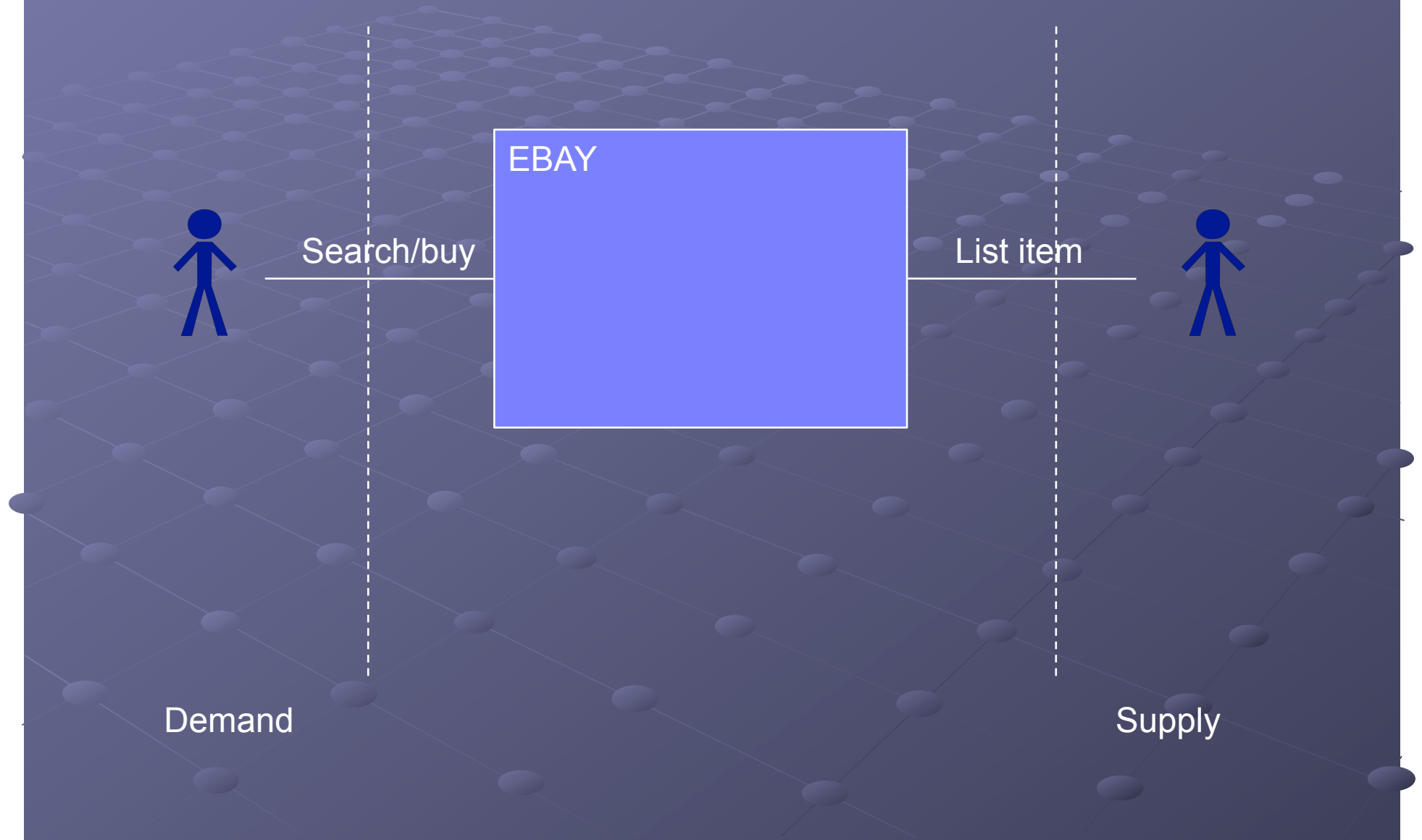
Rough Outline

- What is the OSG? (think ebay)
- HTPC as a new paradigm
- Advantages of HTPC for parallel jobs
- How does HTPC work?
- Who is using it?
- The Future
- Conclusions

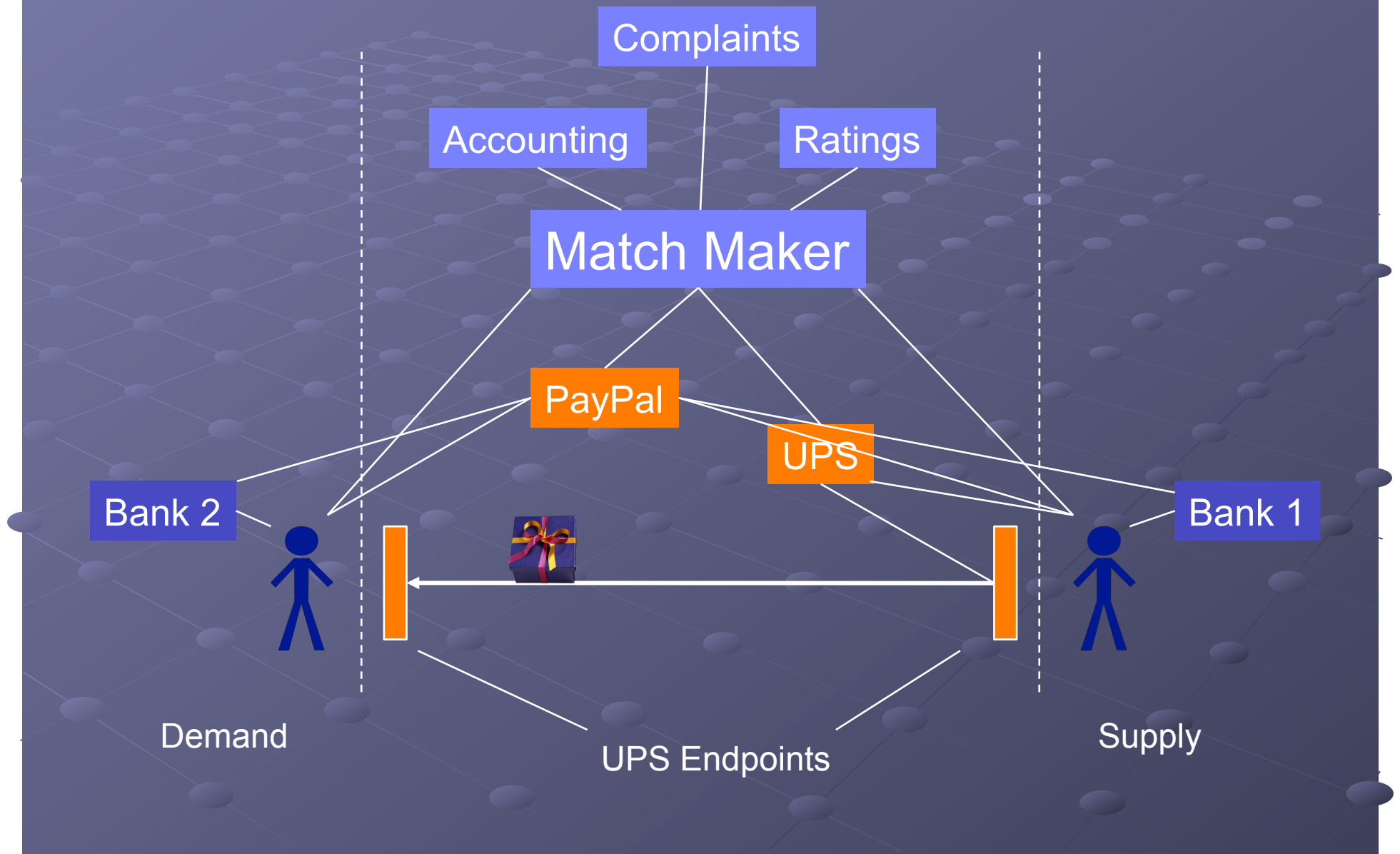
Making sense of the OSG

- **OSG = Technology + Process + Sociology**
- **70+ sites** (& growing) -- Supply
 - contribute resources to the OSG
- Virtual Organizations -- Demand
 - VO's are Multidisciplinary Research Groups
- Sites and VOs often overlap
- OSG Delivers:
 - **>1M CPU hours every day**
 - **1 Pbyte of data transferred every day**

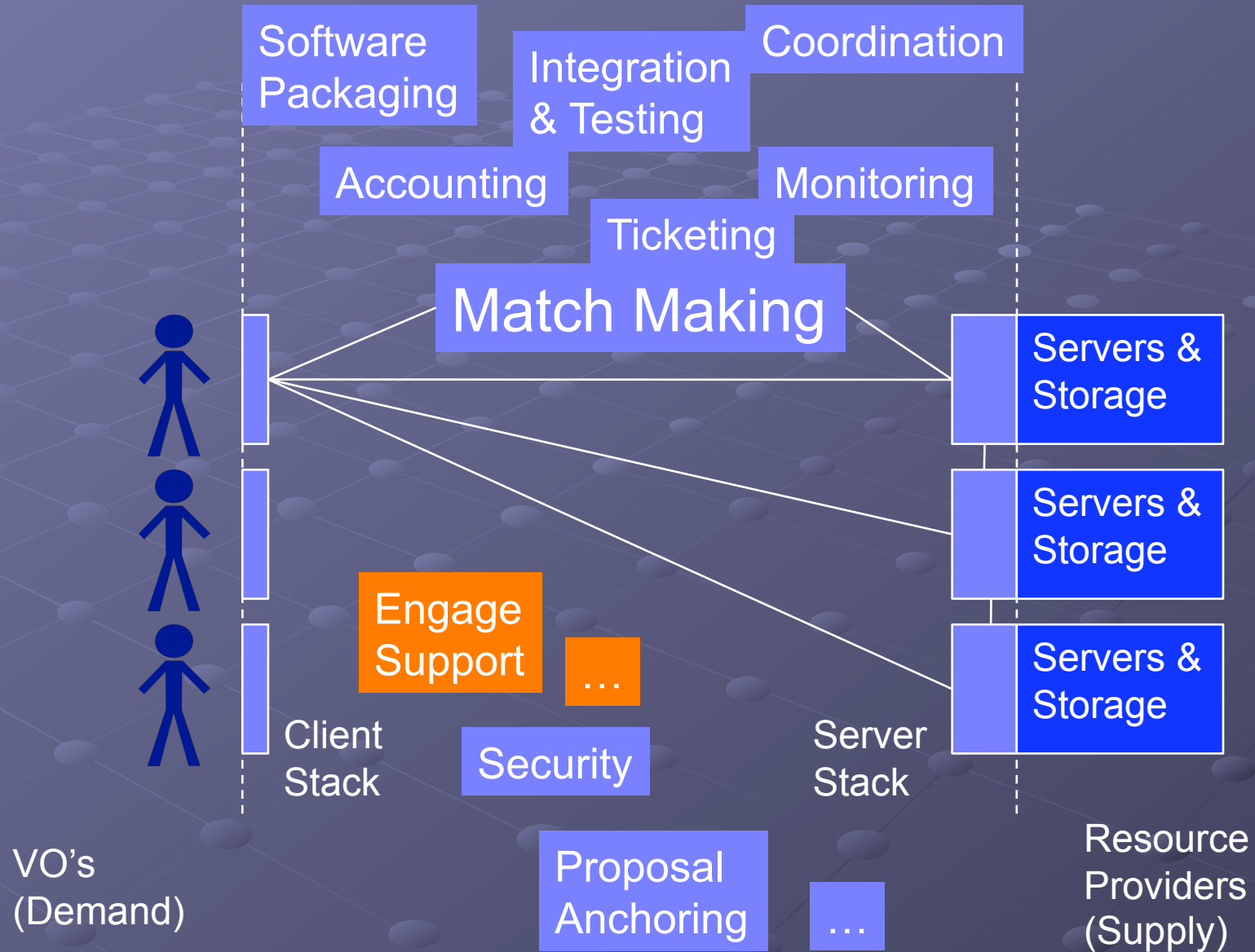
eBay (naïve)

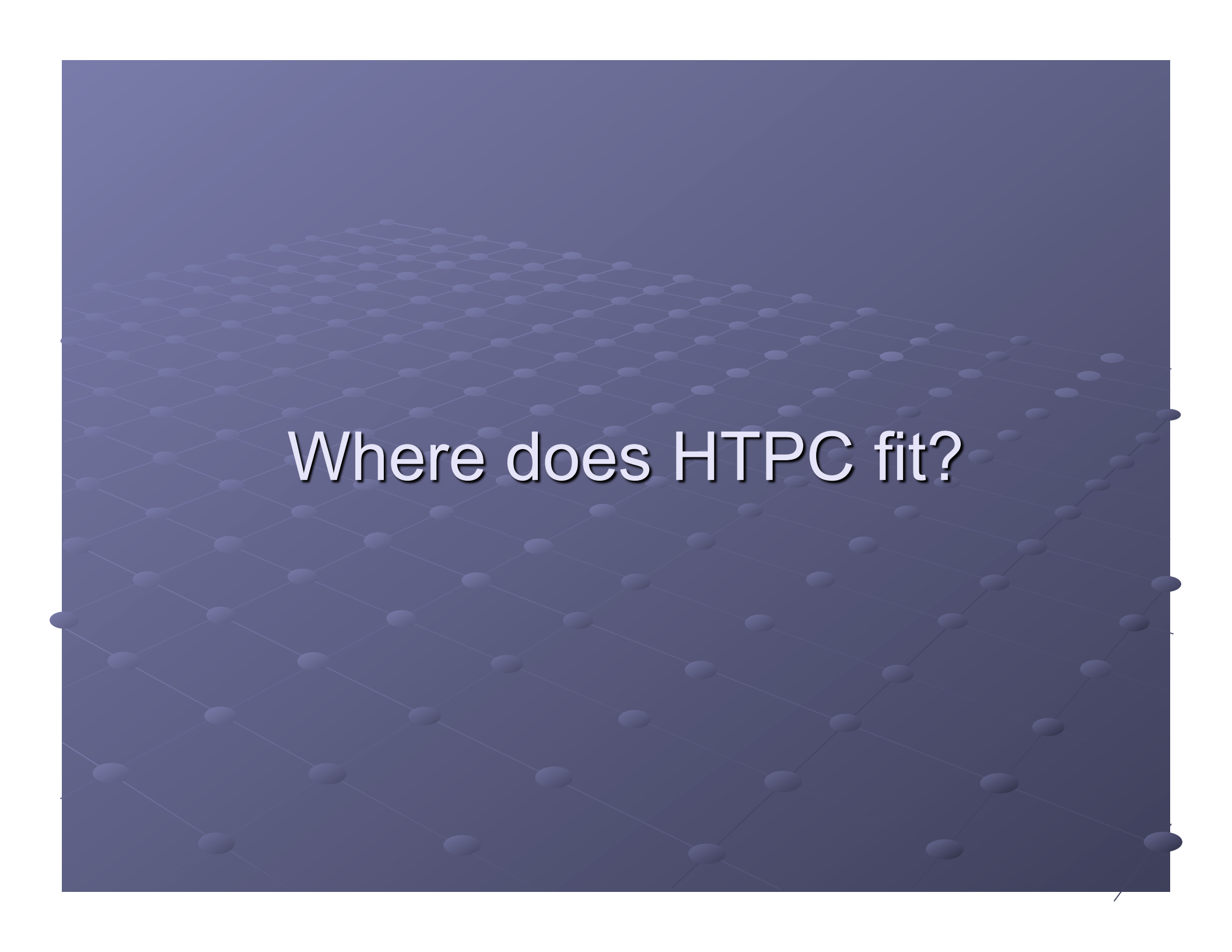


eBay (more realistic)



OSG-Bay





Where does HTPC fit?

The two familiar HPC Models

- High Throughput Computing (e.g. OSG)
 - Run ensembles of single core jobs
- Capability Computing (e.g. TeraGrid)
 - 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)



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
 - 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

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))

Examples of HTPC users:

- 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

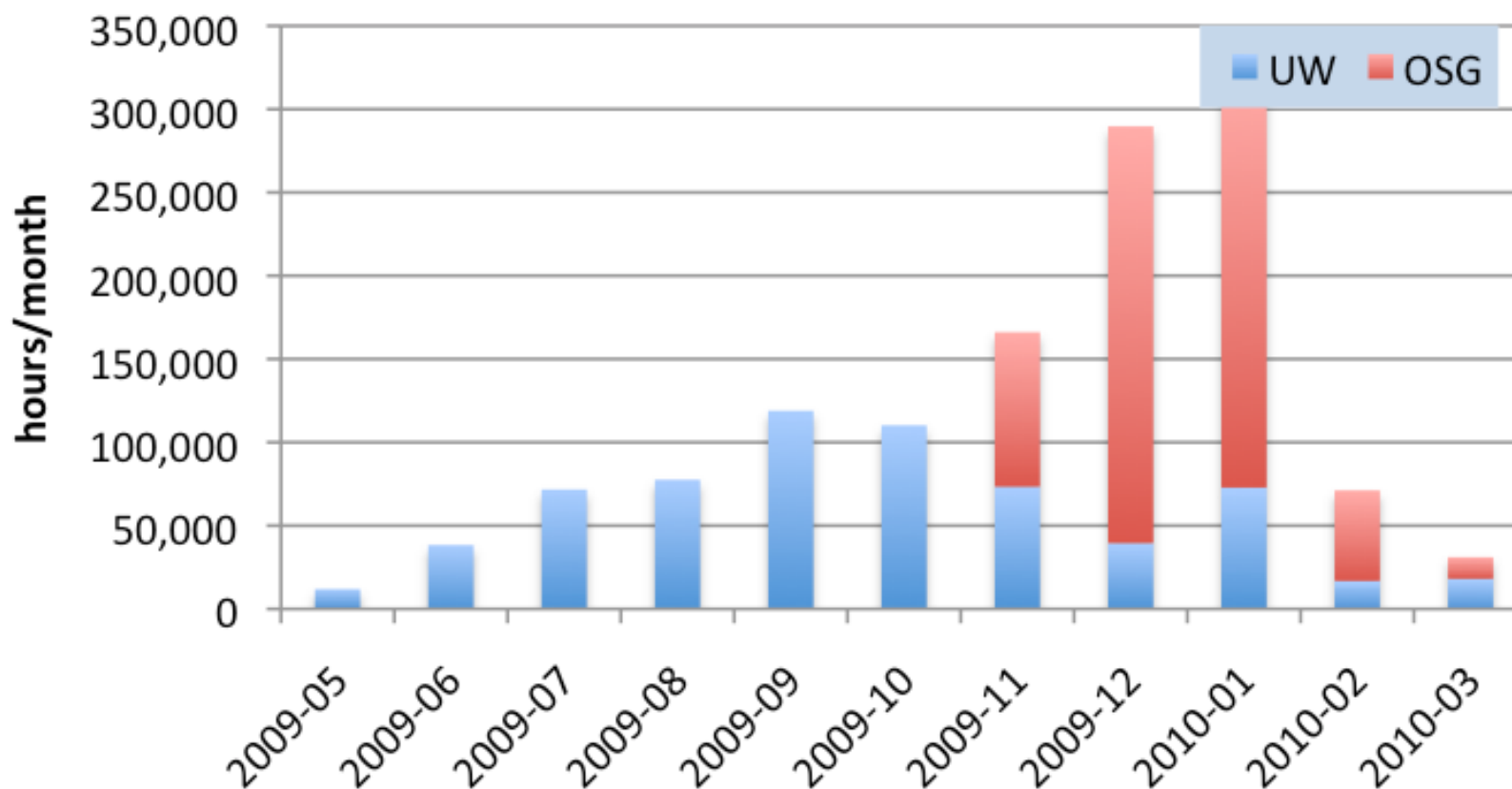
- Each job takes 8 processors, several hours

Examples of HTPC users:

● 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

Chemistry Usage of HTPC



OSG sites that allow HTPC

- OU
 - The first site to run HTPC jobs on the OSG!
- Purdue
- Clemson
- Nebraska
- San Diego, CMS Tier-2

Your site can be on this list!

Future Directions

- More Sites, more cycles!
- More users
 - Working with Atlas (AthenaMP)
 - Working with Amber 9
 - There is room for you...
- Use glide-in to homogenize access

A word about glide-ins

- The OSG really started to shine when the big VOs adopted “pilot” based workflows
- Pilot based models (Atlas & GlideinWMS) are now the recommended access mechanism
 - A pilot job starts on a remote node and requests a user job.
 - Users see a more homogeneous view of the grid
 - Most of the startup problems are solved since a remote job has already started on the node.

Now adopting the glide-in model for Campus Grids

● Campus Grid Factory

- In production use at Nebraska (Weitzel, Bockelman)
- Model makes it transparent whether using local campus resources or the OSG!

● Available for download

- <http://sourceforge.net/apps/trac/campusfactory>
- Includes description and install guide

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

Additional Slides

- <https://twiki.grid.iu.edu/bin/view/Documentation/HighThroughputParallelComputing>
 - HTPC Reference
 - Includes RSL for HTPC pilot jobs

● The players

Dan Fraser

Computation Inst.
University of Chicago

Miron Livny

U Wisconsin

John McGee

RENCI

Greg Thain

U Wisconsin

Key Developer

Funded by NSF-STCI



Configuring Condor for HTPC

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

● <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
```