

Why You Should Consider Grid Computing

Kenny Daily
BIT Presentation

8 January 2007



Outline

- Motivational Story – Electric Fish
 - Grid Computing Overview
 - N1 Sun Grid Engine Software
 - Use of UCI's cluster
 - My Research (why I use the grid)
 - Misc.
-
-

Electric Fish?



Image from <http://www.bio.indiana.edu/facultyresearch/faculty/Smith.html>

Grid Computing Basics

A grid is a collection of computing resources that perform tasks, and appears to users as a **large system** that provides a **single point of access** to **powerful distributed resources**.

Users treat the grid as a single computational resource (One place for everything!)

Why Use a Grid?

Take advantage of available computational resources

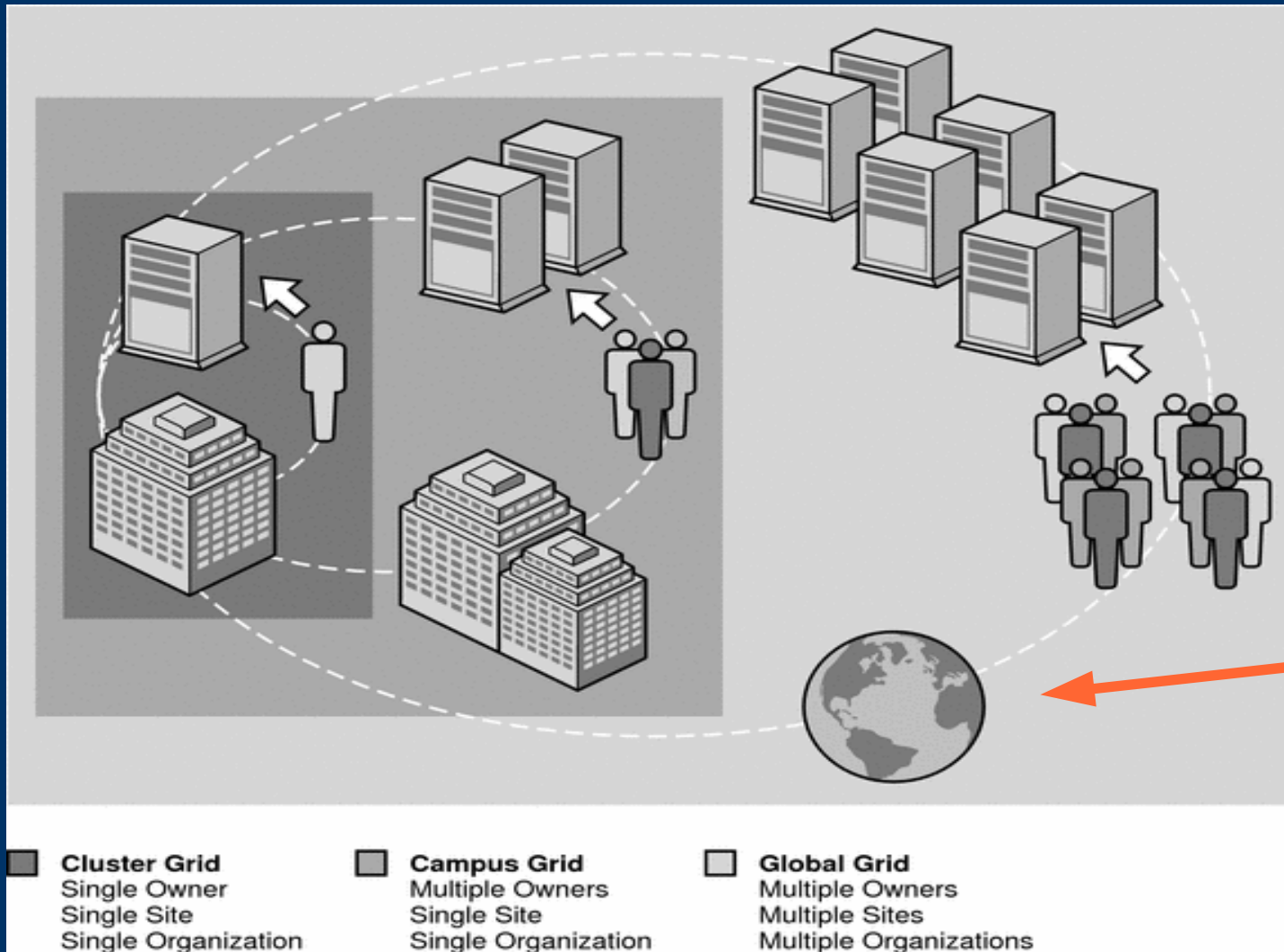
Don't wear out your own computer

Automation of repetitive tasks

Easier to manage!



Grid Computing Basics



TeraGrid has 102 teraflops of computing capability and more than 15 petabytes (quadrillions of bytes) of storage

Image from <http://docs.sun.com/app/docs/doc/817-6117/6mlhdapr5?q=N1GE&a=view>

A couple of definitions

Jobs – Users' requests for resources to run a program.
These resources can be memory, computing speed, amount of time needed, specific computers, architectures, etc.

Queues - provide services for jobs (the computers to run the jobs)

N1 Sun Grid Engine Software (SGE)

A resource management software that:

- Accepts jobs submitted by users and
- Schedules them based on multiple requirements (time, hardware, etc).

It's responsible for the best use of resources based on policies of usage, productivity, etc.

Provides performance monitoring:

Up to the minute access to resource consumption and system load information

When you submit a job

The SGE does all of the following:

1. Accepts jobs (users' requests for computer resources)
 2. Puts jobs in a holding area until the jobs can be run.
 3. Sends jobs from the holding area to an execution device.
 4. Manages running jobs.
 5. Logs the record of job execution when the jobs are finished.
-
-

Types of Jobs

Batch – do the same thing to a bunch of files (also called array jobs)

Interactive – takes input during execution (may be from user or another program)

Parallel – separate parts of a program that don't depend on each other, combine results later (support for multiple interfaces for parallel computing such as MPI)

As A User

Prerequisites:

Access to the system – talk to advisor or sponsor,
send email to helpdesk@uci.edu

Basic familiarity with a UNIX environment
(navigation, running commands, text editing)

More advanced features:

Programming in C, Perl, Java, Python, etc.

As A User

SSH to a host in the cluster:

```
ssh -Y kdaily@dock1.ics.uci.edu
```

Some setup is required, but helpdesk can get you started on that.

Commands To Know

Manage queues (**qhost**, **qconf**, *qmod*)

Submit and delete jobs (**qsub**, *qdel*, *qresub*)

Check job status (**qstat**)

Suspend or enable queues and jobs (*qhold*, *qrls*)

GUI to do most of this: qmon

Each of these commands have hundreds of options,
but the man pages are great!

About The Environment: Queues

Display a list of available queues: `qconf -sql`

```
all.q  
bubs_cluster.q  
casp_cluster.q  
chem_cluster.q  
dna_cluster.q .....
```

Display properties of a queue: `qconf -sq queueName`

Display list of resources you can request for a job:
`qconf -sc` (all of the options available to set)

About The Environment: Hosts

Find hosts to which you can submit jobs: *qconf -ss*

List all the available hosts: *qhost*

Information about a particular host: *qconf -se hostname*



About the Environment: Jobs

What's running now: `qstat`

About a particular user's jobs: `qstat -u username`

About a particular job: `qstat -j jobid`

Only show running jobs: `qstat -s r`

job-ID	prior	name	user	state	submit/start at	queue	slots	ja-task-ID
157709	0.55500	parallel-m	lzhang1	r	01/03/2007 20:02:59	all.q@bubs19.ics.uci.edu	1	3
157707	0.55500	parallel-m	lzhang1	r	01/03/2007 20:02:59	all.q@bubs22.ics.uci.edu	1	1
157694	0.55500	parallel-m	lzhang1	r	01/03/2007 20:02:44	all.q@bubs30.ics.uci.edu	1	1
157708	0.55500	parallel-m	lzhang1	r	01/03/2007 20:02:59	all.q@bubs5.ics.uci.edu	1	1
157697	0.55500	parallel-m	lzhang1	r	01/03/2007 20:02:44	all.q@casp-33.ics.uci.edu	1	1

Submitting a simple job

qsub scriptname

It's really that easy!!!

Let's see some examples.

Examples

1. Simple Jobs
2. Array Jobs
3. Batch Jobs



Why I Use The Grid

I'm studying how different measures of similarity can be used to search for chemical molecules

For example, I have a new chemical structure that I found. What else is like it?

First, what is a “good” measure of similarity?

Looks the same, same properties, same overall shape, etc...

Why I Use The Grid

I have at least 6 different similarity measures with 3 different ways of computing them, and at least 12 different data sets to test them on!

I need to be able to take the pairwise similarity of each dataset to a random dataset of 50K molecules and then analyze the results.

Each experiment will take 12-24 hours of processing time

$6 \times 3 \times 12 \times 12 = 2592$ Hours, or ~100 days!

Why I Use The Grid

Batch processing lets me distribute those experiments over a bunch of computers and run them all at the same time!

Let's look at the code that handles this (warning, kind of ugly!!!)

Not the best way to do this – interfaces exist in Java, Perl, C, and Python to do this within the code

Thank you!

Questions, Comments, Suggestions?



References

- TeraGrid - <http://www.teragrid.org/>
 - Sun Grid Engine Documentation - <http://gridengine.sunsource.net/documentation.html>
 - SGE man pages
-
-