The Condor View of

Computing

Miron Livny
Computer Sciences Department
University of Wisconsin-Madison
miron@cs.wisc.edu

Computing power is **everywhere**, how can we make it usable by **anyone?**

The Condor Project (Established '85)

Distributed Computing **research** performed by a team of **~35** faculty, full time staff and students who

- face **software/middleware engineering** challenges in a UNIX/Linux/Windows environment,
- involved in national and international **collaborations**,
- interact with **users** in academia and industry,
- maintain and support a distributed **production** environment (more than 2000 CPUs at UW),
- and educate and train **students**.

Funding – **DoD, DoE, NASA, NIH, NSF,AT&T, INTEL, Micron, Microsoft and the UW Graduate School**

Claims for "benefits" provided by Distributed Processing Systems

P.H. Enslow, *"What is a Distributed Data Processing System?"* Computer, January 1978

- **High Availability and Reliability**
- **High System Performance**
- **Ease of Modular and Incremental Growth**
- **Automatic Load and Resource Sharing**
- **Good Response to Temporary Overloads**
- **Easy Expansion in Capacity and/or Function**

HW is a Commodity

Raw computing power and storage capacity is everywhere - on desk-tops, shelves, and racks. It is

- **cheap,**
- **dynamic,**
- **distributively owned,**
- **heterogeneous and**
- **evolving.**

" … Since the early days of mankind the primary motivation for the establishment of *communities* has been the idea that by being part of an organized group the capabilities of an individual are improved. The great progress in the area of inter-computer communication led to the development of means by which stand-alone processing sub-systems can be integrated into multi-computer *'communities'*. … "

Miron Livny, *" Study of Load Balancing Algorithms for Decentralized Distributed Processing Systems.",* Ph.D thesis, July 1983.

Every community needs a Matchmaker*!

* or a Classified section in the newspaper or an eBay.

We use Matchmakers to build Computing Communities out of Commodity Components

High Throughput Computing

For many experimental scientists, scientific progress and quality of research are strongly linked to computing throughput. In other words, they are less concerned about instantaneous computing power. Instead, what matters to them is the amount of computing they can harness over a month or a year --- they measure computing power in units of scenarios per day, wind patterns per week, instructions sets per month, or crystal configurations per year.

High Throughput Computing is a 24-7-365 activity

FLOPY ≠ *(60*60*24*7*52)*FLOPS*

Master-Worker Paradigm

Many scientific, engineering and commercial applications (Software builds and testing, sensitivity analysis, parameter space exploration, image and movie rendering, High Energy Physics event reconstruction, processing of optical DNA sequencing, training of neural-networks, stochastic optimization, Monte Carlo...) follow the Master-Worker (MW) paradigm where ...

Master-Worker Paradigm

... a heap or a Directed Acyclic Graph (DAG) of tasks is assigned to a master. The master looks for workers who can perform tasks that are “ready to go” and passes them a description (input) of the task. Upon the completion of a task, the worker passes the result (output) of the task back to the master.

- Master may execute some of the tasks.
- Master maybe a worker of another master.
- Worker may require initialization data.

Master-Worker computing is
Naturally Parallel.
It is by no means
Embarrassingly Parallel.
Doing it right is by no means
trivial.

our answer to High Throughput MW Computing on commodity resources

The World of Condors

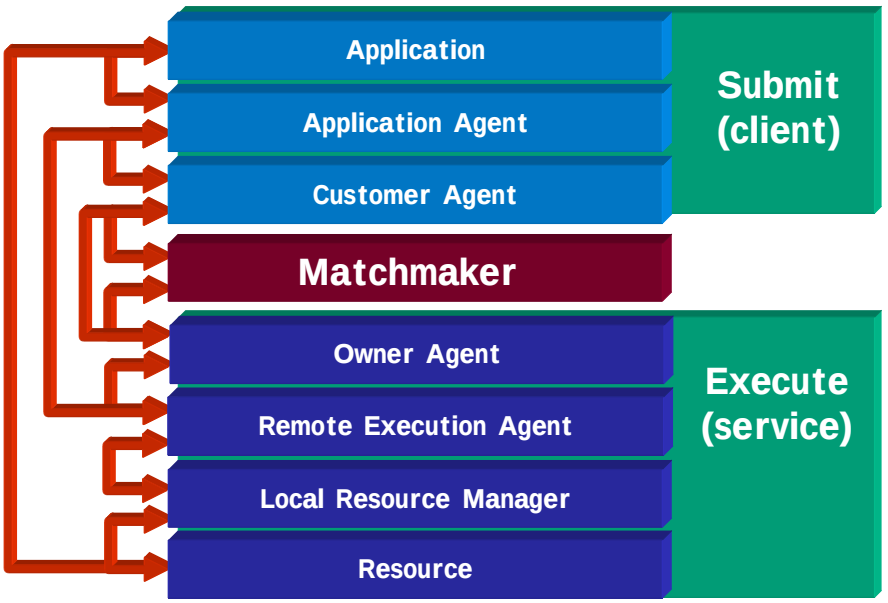
- Available for most Unix and Windows platforms at www.cs.wisc.edu/Condor
- More than 400 Condor pools at commercial and academia sites world wide
- More than 14,000 CPUs world wide
- "Best effort" and "for fee" support available

Some recent *.jp downloads

1. soum.co.jp
2. mikilab.doshisha.ac.jp
3. is.aist-nara.ac.jp
4. mikilab.doshisha.ac.jp
5. unisys.co.jp
6. icrr.u-tokyo.ac.jp
7. sgi.co.jp
8. mxf.nes.nec.co.jp
9. proside.co.jp
10. shimadzu.co.jp
11. ais.cmc.osaka-u.ac.jp
12. suri.co.jp

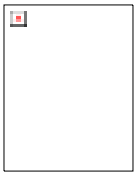
1. nakl.t.u-tokyo.ac.jp
2. hydra.mki.co.jp jp
3. kk.anritsu.co.
4. nakl.t.u-tokyo.ac.jp
5. apr.jaeri.go.jp
6. is.aist-nara.ac.jp
7. infonet.cse.kyutech.ac.jp
8. mi-2.mech.kobe-u.ac.jp
9. pu-toyama.ac.jp
10. cbo.mss.co.jp
11. soum.co.jp
12. crl.hitachi.co.jp

The Layers of Condor



The Grid: Blueprint for a New Computing Infrastructure

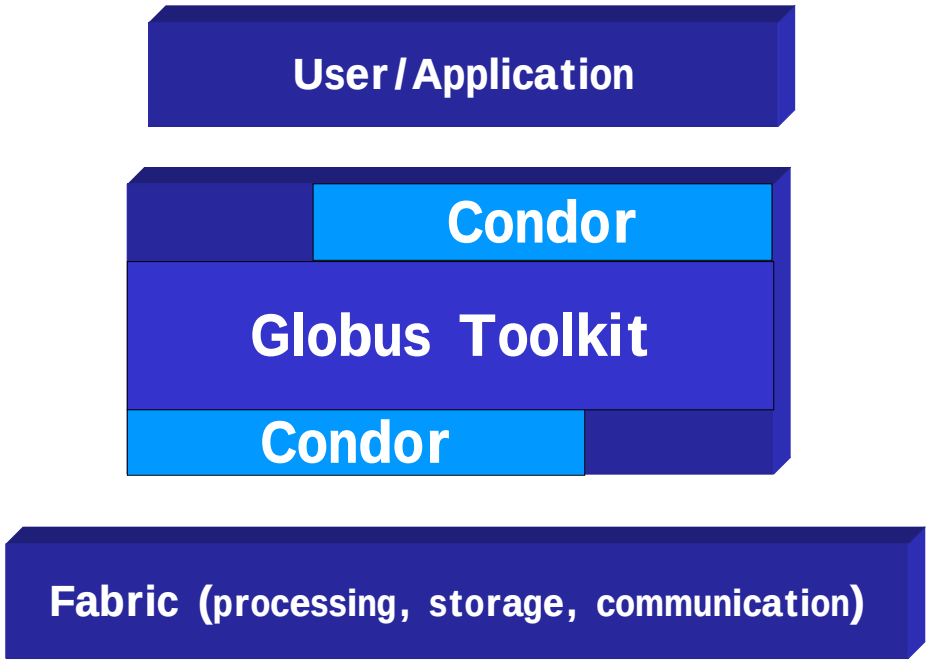
Edited by Ian Foster and Carl Kesselman
July 1998, 701 pages, $62.95



The grid promises to fundamentally change the way we think about and use computing. This infrastructure will connect multiple regional and national computational grids, creating a universal source of **pervasive and dependable** computing power that supports dramatically new classes of applications. The Grid provides a clear vision of what computational grids are, why we need them, who will use them, and how they will be programmed.

"We have provided in this article a concise statement of the "Grid problem," which we define as **controlled resource sharing and coordinated resource use in dynamic, scalable virtual organizations**. We have also presented both requirements and a framework for a Grid architecture, identifying the principal functions required to enable sharing within **VOs** and defining key relationships among these different functions."

"The Anatomy of the Grid - Enabling Scalable Virtual Organizations" Ian Foster, Carl Kesselman and Steven Tuecke 2001.

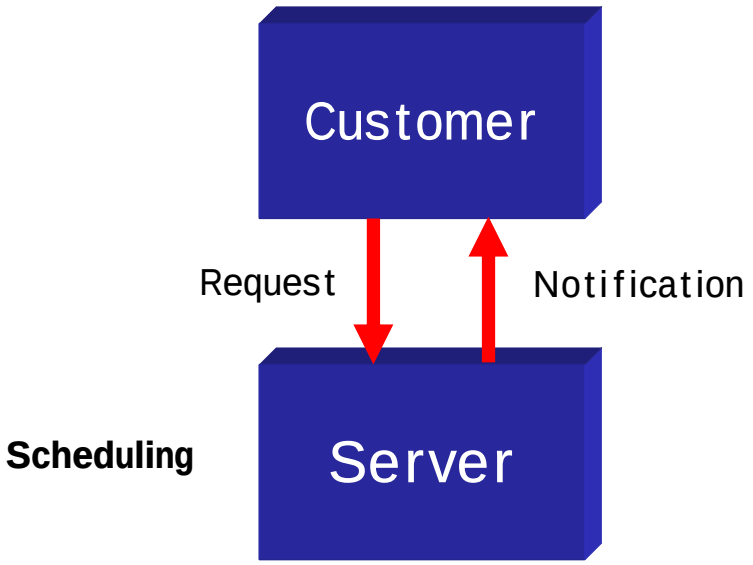


Customer orders:

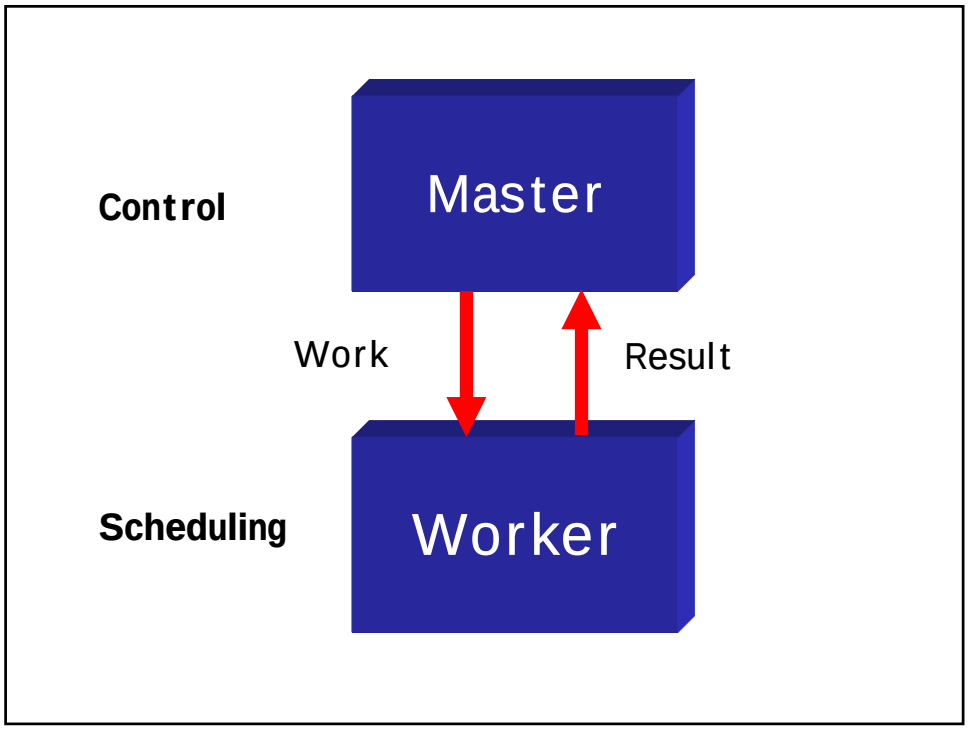
Run Job F

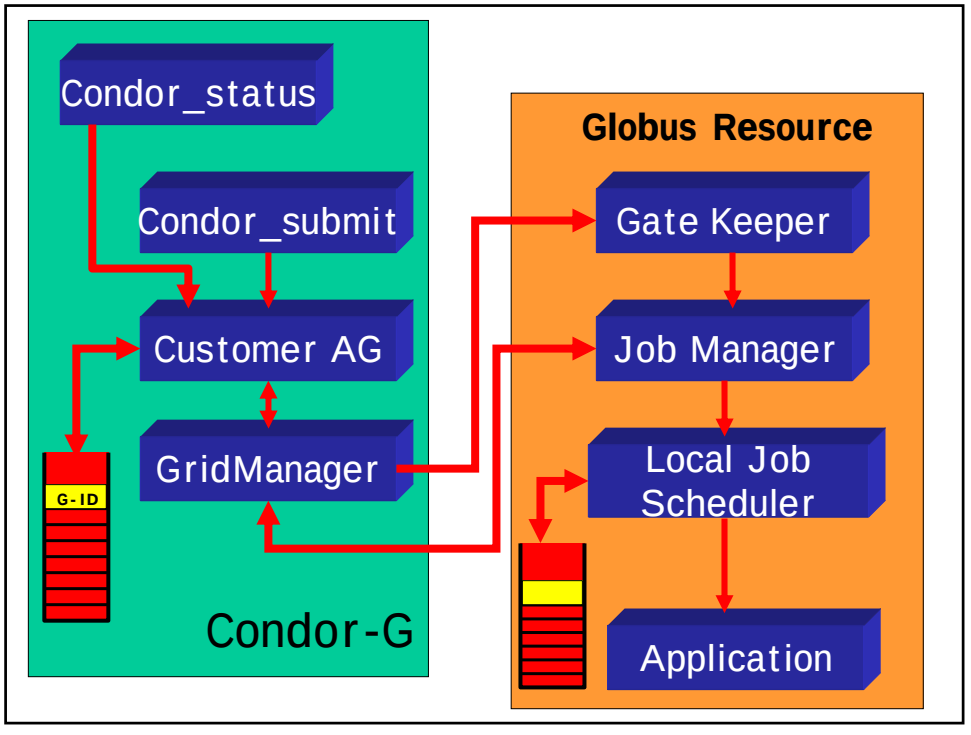
Server delivers.

www.cs.wisc.edu/condor



Scheduling


Control
Master
Work
Result
Scheduling
Worker

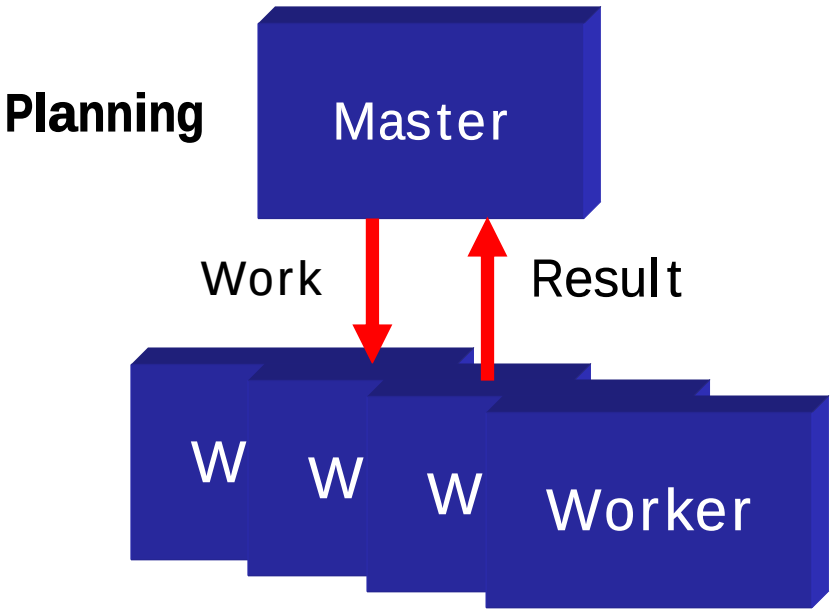

Condor_status
Condor_submit
Customer AG
GridManager
G-ID
Condor-G
Globus Resource
Gate Keeper
Job Manager
Local Job Scheduler
Application

Customer orders:

Run Job F on the best CE

Server delivers.

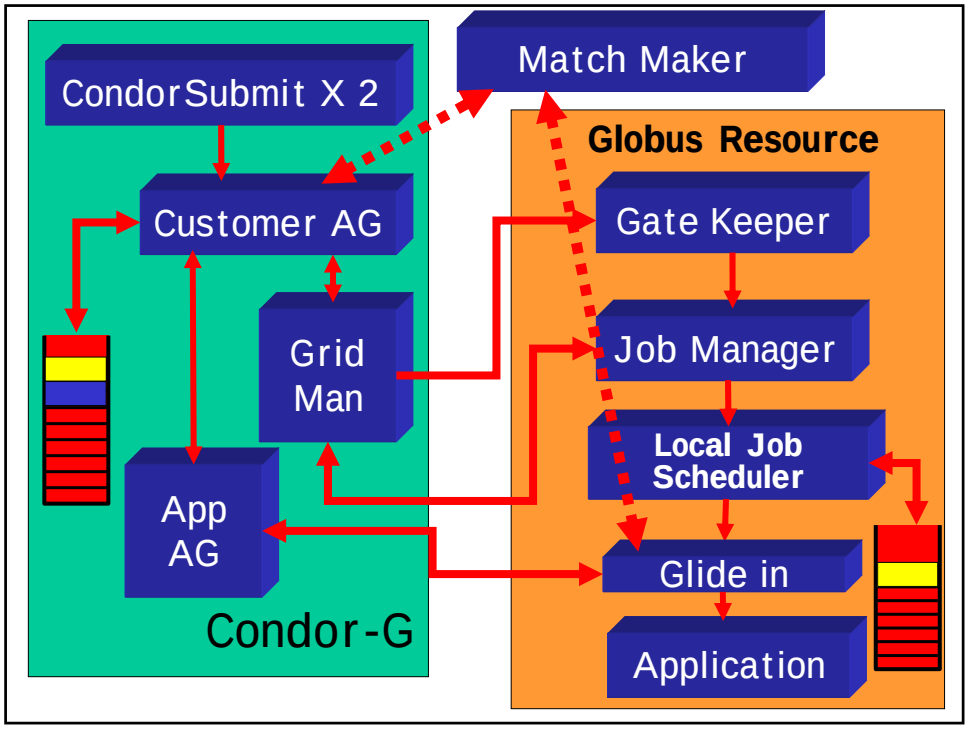
www.cs.wisc.edu/condor

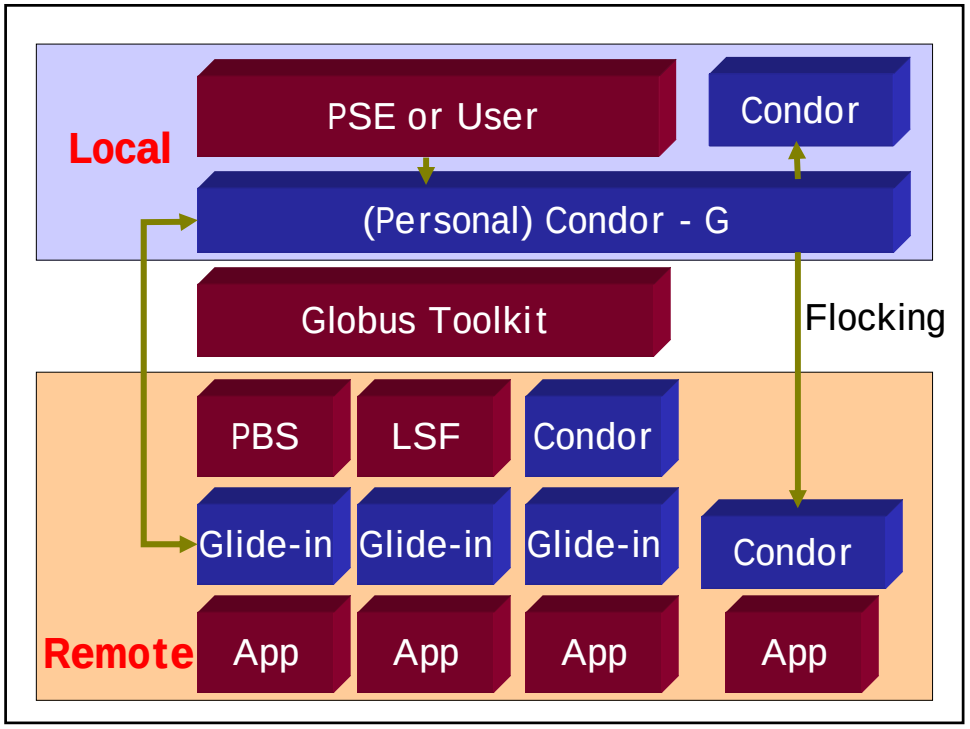


Condor Glide-in:

Expending your Condor pool "on the fly" and executing your jobs on the remote resources in a "friendly" environment.

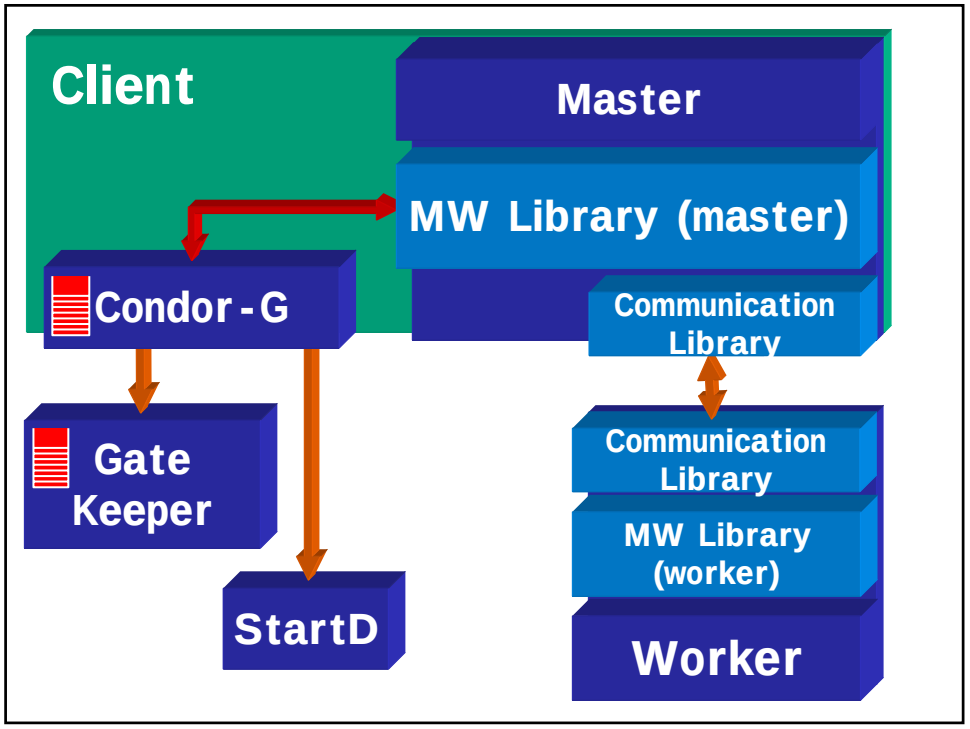
www.cs.wisc.edu/condor





It Works!!!!

Optimization



Master-Worker (MW) library

- Manages **workers** – locates resources, lunches workers, monitors health of workers, …
- Manages **work** – moves work and results between master and worker via files, PVM or TCP/IP sockets

The NUG*n* *Quadratic Assignment Problem (QAP)*

$$\min_{p \in \Pi} \sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} b_{p(i)p(j)}$$

Despite its simple statement - *minimize the assignment cost of n facilities to n locations* - it is extremely difficult to solve even modest instances of this problem. Problems with n>20 are difficult; problems with n>30 have not even been attempted yet.

We currently hold the world record to solve NUG25 in 6.7 hours (previous record : 56 days !!!). *Our goal now is to solve NUG30, an unsolved problem formulated 30 years ago*.

NUGn	Date	Sites	Wall Clock	Workers (avg)	CPU hours
25		1	6.7	94	630
27	02/23/00	1	24	136	3,264
28	04/13/00	3	104	200	20,800
30	06/15/00				100,000

NUG30 Personal Grid ...

Flocking:

-- the main Condor pool at Wisconsin (500 processors)

-- the Condor pool at Georgia Tech (284 Linux boxes)

-- the Condor pool at UNM (40 processors)

-- the Condor pool at Columbia (16 processors)

-- the Condor pool at Northwestern (12 processors)

-- the Condor pool at NCSA (65 processors)

-- the Condor pool at INFN Italy (54 processors)

Glide-in:

-- Origin 2000 (through LSF) at NCSA. (512 processors)

-- Origin 2000 (through LSF) at Argonne (96 processors)

Hobble-in:

-- Chiba City Linux cluster (through PBS) at Argonne (414 processors).

NUG30 - Solved!!!

Date: Thu, 15 Jun 2000 21:26:19 -0500
Sender: goux@dantec.ece.nwu.edu
Subject: Re: Let the festivities begin.

Hi dear Condor Team,

you all have been amazing. NUG30 required 10.9 years of Condor Time. In just seven days !

More stats tomorrow !!! We are off celebrating !

condor rules !

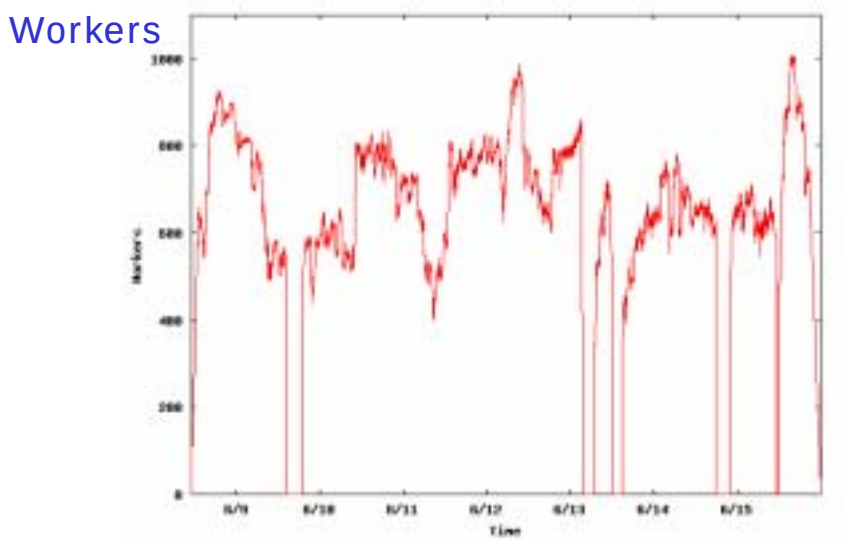
cheers,

JP.

Solution Characteristics.

Wall Clock Time	6:22:04:31
Avg. # Machines	653
Max. # Machines	1007
CPU Time	Approx. 11 years
Nodes	11,892,208,412
LAPs	574,254,156,532
Parallel Efficiency	92%

The Workforce



11 CPU years
in less than a week,
How did they do it?

Effective management
of their workforce!

(www.mcs.anl.gov/metaneos/nug30)

You do not
have to be a
super-person
in order to do
super-computing

www.cs.wisc.edu/condor

It Works!!!

Condor-XW / XtremWeb-C:
Global Computing on Condor Pools

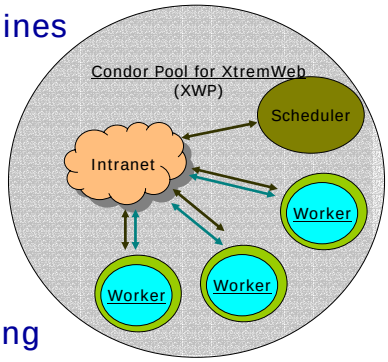
Franck Cappello, Oleg Lodygensky, Vincent Neri
LRI - Université Paris sud

Actions Concertées Incitatives [ACI]
Globalisation des Ressources Informatiques
et des Données [GRID]

XtremWeb-C (XW in Condor)
Deploying XW Workers with Condor

Merge Condor flexibility and XtremWeb connectivity.

- Use Condor to :
  - ➢ manage a pool of machines
  - ➢ dispatch XtremWeb workers as Condor tasks



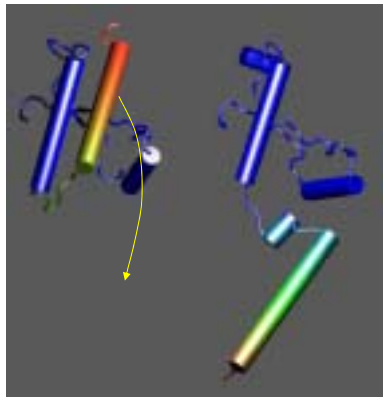
Enable Pull mode task dispatching in a Condor pool.

Exploration of conformational transitions in proteins

- Molecular Dynamics is great for simulating random thermal deformations of a protein...
  - ***but*** unlikely to reach a particular conformation of the protein, even if you *really* want to
- Vibrational Modes is great for identifying preferred deformations towards "interesting" conformations
  - ***but*** strictly applicable to small deformations only
- Combined approach: we force molecular dynamics to explore "interesting" deformations identified by vibrational modes

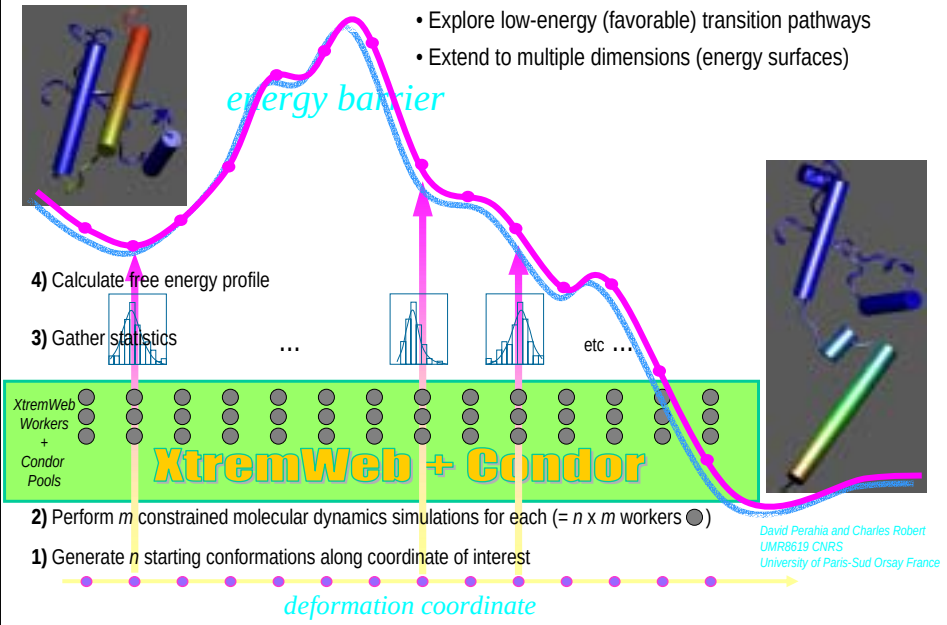
e.g., normal prion protein

"interesting" conformation (amyloid?)



David Perahia and Charles Robert
UMR8619 CNRS
University of Paris-Sud Orsay France

Obtain free-energy profiles



David Perahia and Charles Robert
UMR8619 CNRS
University of Paris-Sud Orsay France

"The Grid"
is not just a Grid of
resources
it is a Grid of
technologies

Customer orders:

Place y = F(x) at L!

Grid delivers.

www.cs.wisc.edu/condor

Logical Request

Planning, scheduling,
execution,
error recovery,
monitoring …

Physical Resources

A simple plan for y=F(x) -> L

1. Allocate size(x)+size(y) at SE(i)
2. Move x from SE(j) to SE(i)
3. Place F on CE(k)
4. Compute F(x) at CE(k)
5. Move y to L
6. Release allocated space

Storage Element (SE); Compute Element (CE)

What we have here is a simple six-nodes Directed Acyclic Graph (DAG)

Execution of DAG must be Controlled by client

Data Placement* (DaP)
is an integral part of
end-to-end
functionality

* Space management and Data transfer

DAGMan

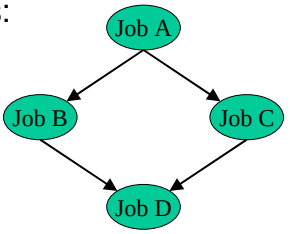
Directed Acyclic Graph Manager

DAGMan allows you to specify the *dependencies* between your jobs (processing and DaP), so it can *manage* them automatically for you.

Defining a DAG

- A DAG is defined by a *.dag file*, listing each of its nodes and their dependencies:

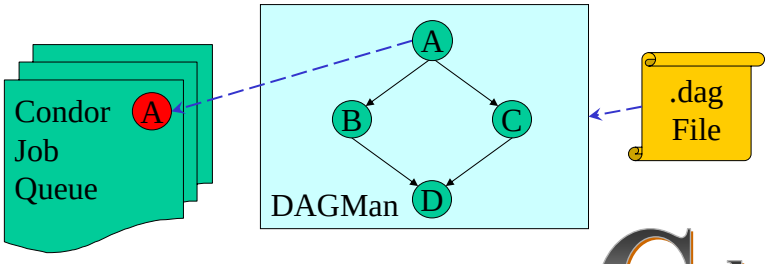
```
# diamond.dag
Job A a.sub
Job B b.sub
Job C c.sub
Job D d.sub
Parent A Child B C
Parent B C Child D
```



- each node will run the job specified by its accompanying Condor submit file

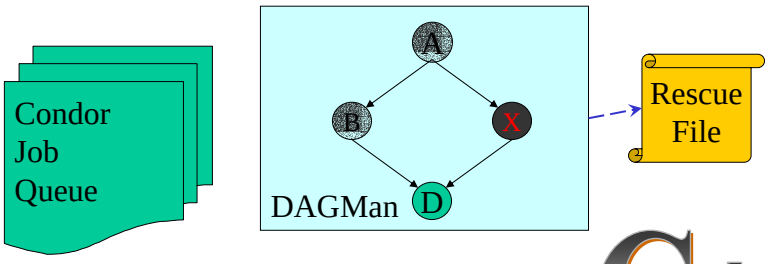
Running a DAG

- DAGMan acts as a "meta-scheduler", managing the submission of your jobs to Condor-G based on the DAG dependencies.



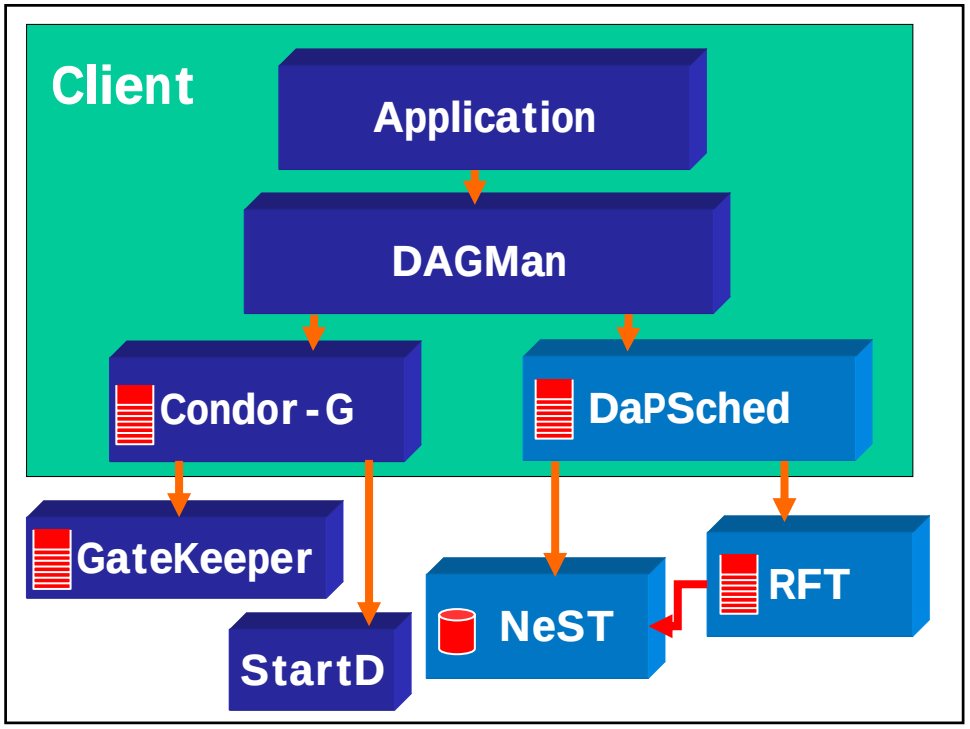
Running a DAG (cont'd)

- In case of a job failure, DAGMan continues until it can no longer make progress, and then creates a *"rescue" file* with the current state of the DAG.

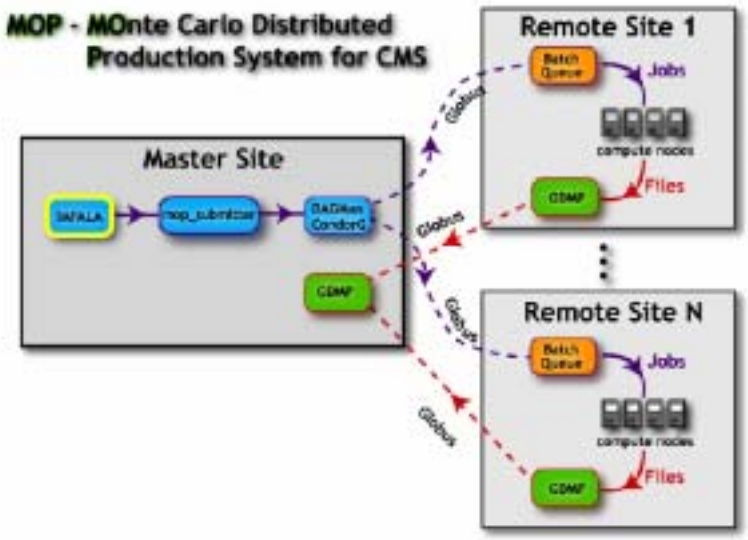


It Works!!!!

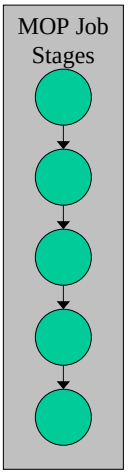
High Energy Physics



PPDG-MOP US-CMS Test bed

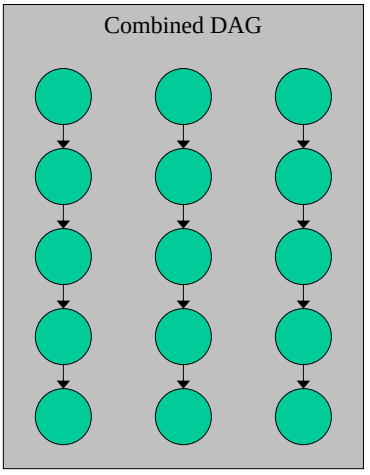


MOP Job Stages

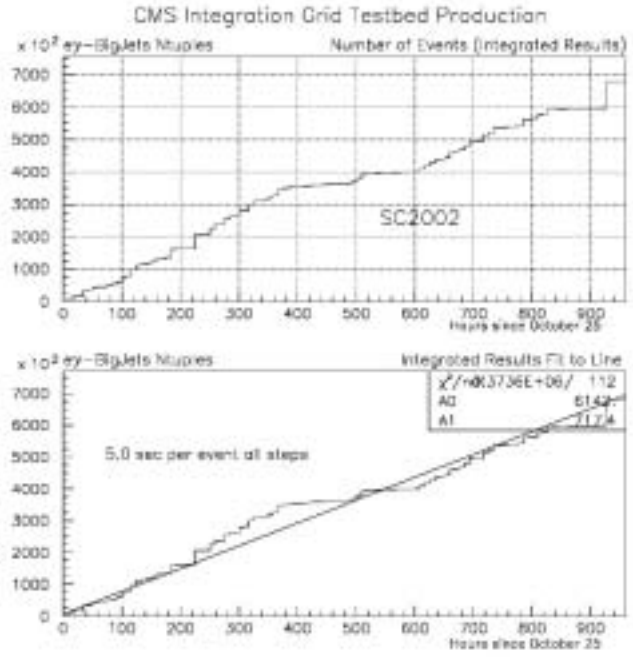


- Stage-in – get the program and its data to a remote site
- Run – run the job at the remote site
- Stage-back – get the program logs back from the remote site
- Publish – advertise the results so they will be sent to sites that want it
- Cleanup – clean up remote site

MOP Job Stages

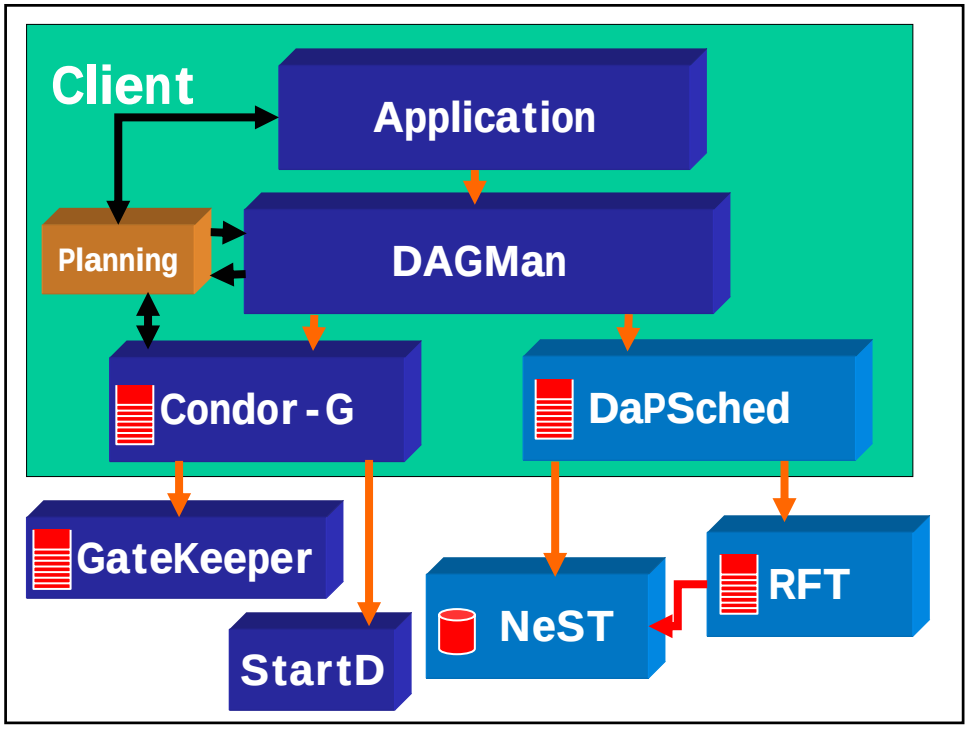


- MOP combines the five-stage DAG for each IMPALA job into one giant DAG, and submits it to DAGMan.



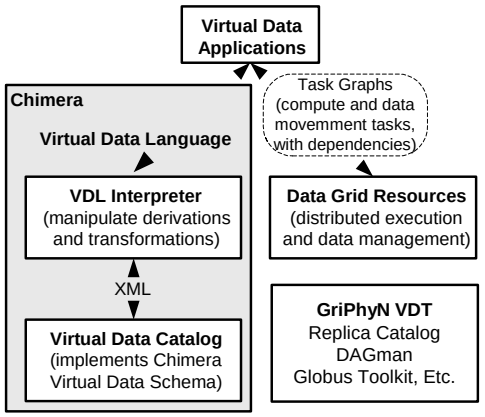
It Works!!!!

Sloan Digital Sky Survey

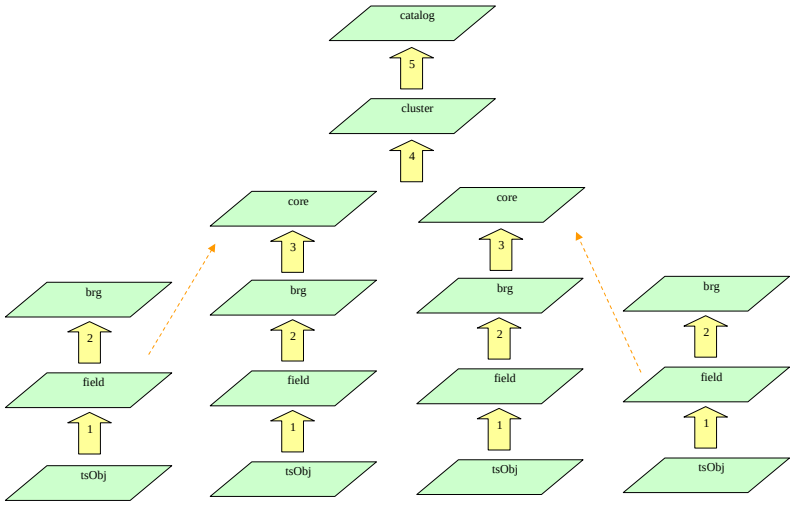


Chimera Virtual Data System

- Virtual data catalog
  - Transformations, derivations, data
- Virtual data language
  - Catalog Definitions
- Query Tool
- Applications include browsers and data analysis applications

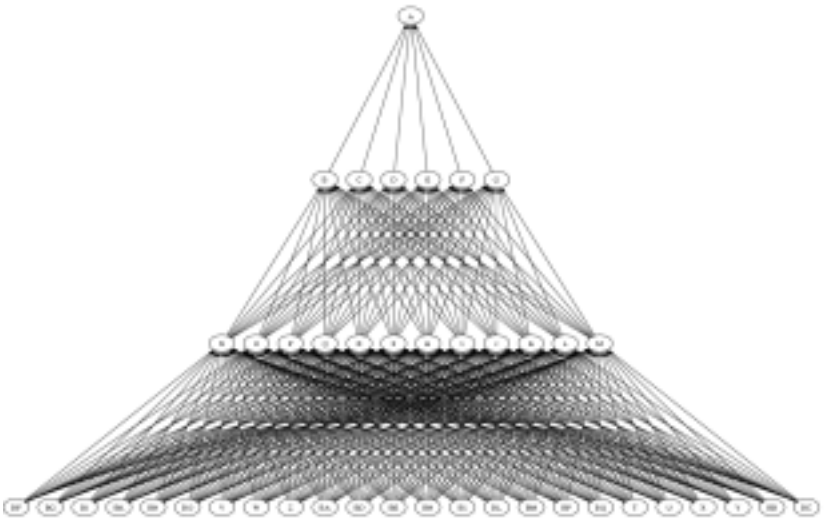


Cluster-finding Data Pipeline



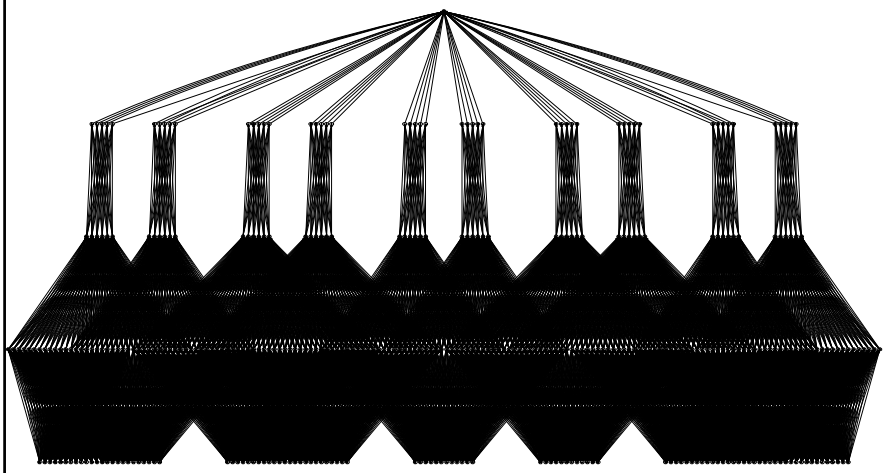
Argonne National Laboratory

Small SDSS Cluster-Finding DAG



Argonne National Laboratory

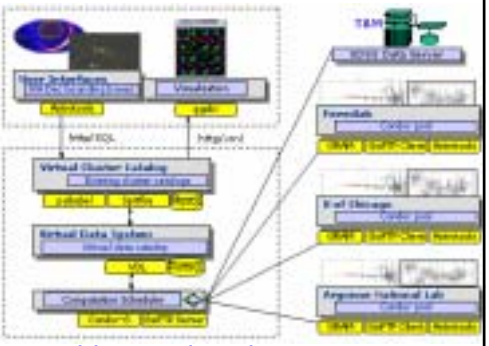
And Even Bigger: 744 Files, 387 Nodes



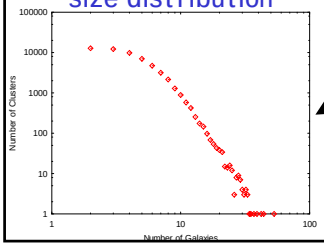
Argonne National Laboratory

Cluster Finding Grid

Size distribution of galaxy clusters?



Galaxy cluster size distribution



Chimera Virtual Data System
+ GriPhyN Virtual Data Toolkit
+ iVDGL Data Grid (many CPUs)

Joint work with Jim Annis, Steve Kent, FNAL

Condor BLASTs Through Jobs

The Science

Biologists compare protein sequences from well-understood organisms with sequences in less well-understood organisms. If they find similar sequences, it may be an indication that the proteins work similarly.

Looking for an exact or similar match among the tens of thousands of already sequenced proteins is computationally intensive. BLAST is a program commonly used by biologists for this task.

One local group we have been working with, BMRB, uses Condor technology to submit and track 18,000 searches every week. Each search takes about five minutes.

The Technology

Two Directed Acyclic Graphics (DAGs) are used to run 18,000 searches.

1)The first DAG transfers BLAST and the data needed for the jobs, then creates the DAG for the 18,000 searchers.

2)The second DAG tracks and throttles the 18,000 searches.

- **Condor DAGMan:** Fault-tolerant scheduler that tracks dependencies between jobs
- **Condor-G:** Fault-tolerant job submission engine for Grid jobs
- **Globus:** Grid toolkit for job submission, data transfer, and information services
- **Condor:** Fault-tolerant, high-throughput batch job system