Condor Compatible Tools for Data Intensive Computing

Douglas Thain
University of Notre Dame

Condor Week 2011

The Cooperative Computing Lab

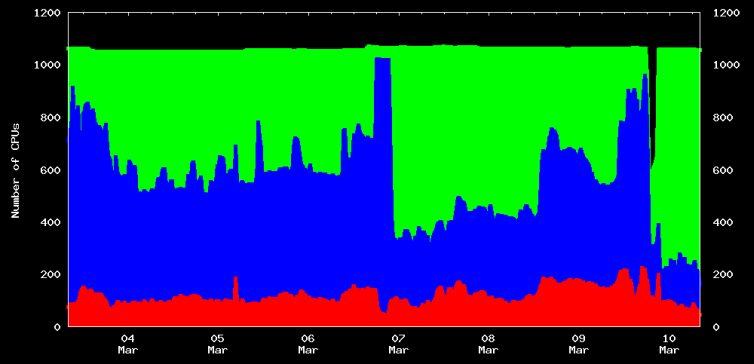
- We collaborate with people who have large scale computing problems in science, engineering, and other fields.
- We operate computer systems on the scale of 1200 cores. (Small)
- We conduct computer science research in the context of real people and problems.
- We publish open source software that captures what we have learned.

http://www.nd.edu/~ccl

What is Condor Compatible?

- Work right out of the box with Condor.
- Respect the execution environment.
- Interoperate with public Condor interfaces.

http://condor.cse.nd.edu

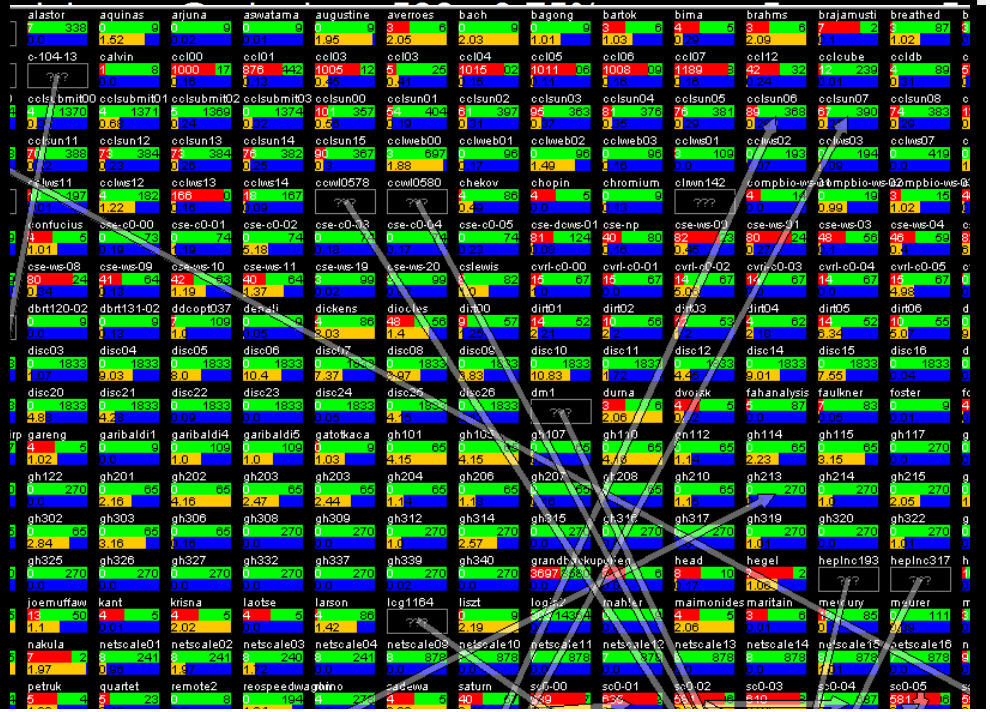


86160 (47%) **CPU-Hours Unused**
76483 (42%) **CPU-Hours Used by Condor**
18839 (10%) **CPU-Hours Used by Owner**
181482 (100%) **CPU-Hours Total**

Top Condor Users for the Last Week

	CPU Hours	Percent Total	Max Jobs Running	Max Jobs Queued
vvijayan@nd.edu	24339	31.18%	450	780
athrash1@nd.edu	16458	21.08%	501	618
pbrenne1@nd.edu	15439	19.78%	104	125
pbui@nd.edu	6423	8.23%	154	162
jthomp11@nd.edu	5494	7.04%	200	200
rcarmich@nd.edu	3652	4.68%	828	2000
lyu2@nd.edu	3398	4.35%	30	30
ccl@nd.edu	2131	2.73%	89	406

http://greencloud.crc.nd.edu



And the “challenging” users…

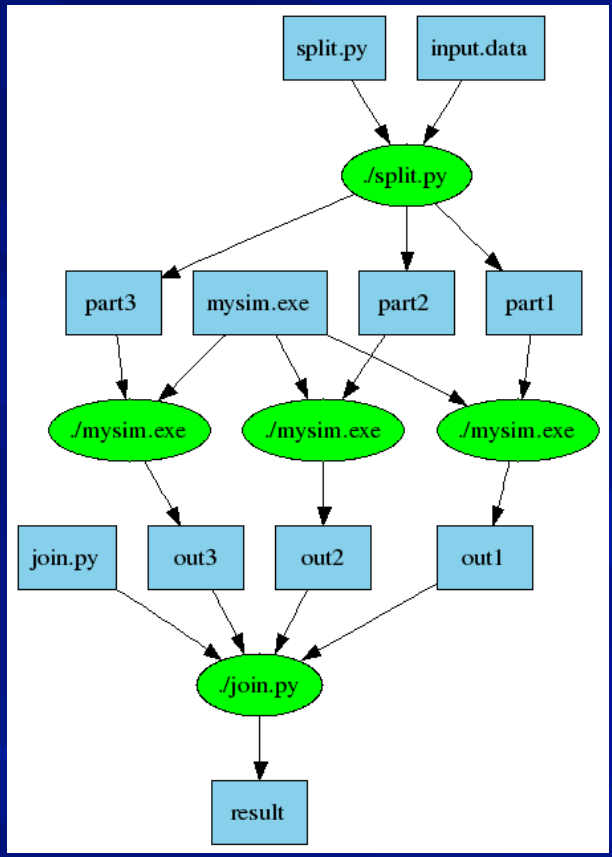
- I submitted 10 jobs yesterday, and that worked, so I submitted 10M this morning!
- And then I write the output into 10,000 files of 1KB each. Per job.
- Did I mention each one reads the same input file of 1TB?
- Sorry, am I reading that file twice?
- What do you mean, sequential access?
- Condor is nice, but I also want to use my cluster, and SGE, and Amazon and…

Idea:

Get the end user into telling us more about their data needs.

In exchange,
give workflow portability
and resource management.

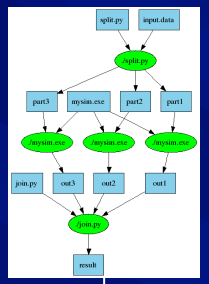
Makeflow




```
part1 part2 part3: input.data split.py
	./split.py input.data

out1: part1 mysim.exe
	./mysim.exe part1 >out1

out2: part2 mysim.exe
	./mysim.exe part2 >out2

out3: part3 mysim.exe
	./mysim.exe part3 >out3

result: out1 out2 out3 join.py
	./join.py out1 out2 out3 > result
```

Douglas Thain and Christopher Moretti, **Abstractions for Cloud Computing with Condor**, Syed Ahson and Mohammad Ilyas, *Cloud Computing and Software Services: Theory and Techniques*, pages 153-171, CRC Press, July, 2010.

Makeflow = Make + Workflow



Transaction Log

Makeflow Core Logic

Abstract System Interface

Local | Condor | SGE | Hadoop | Work Queue

Transaction Log

Makeflow Core Logic

Abstract System Interface

Local | Condor | SGE | Hadoop | Work Queue

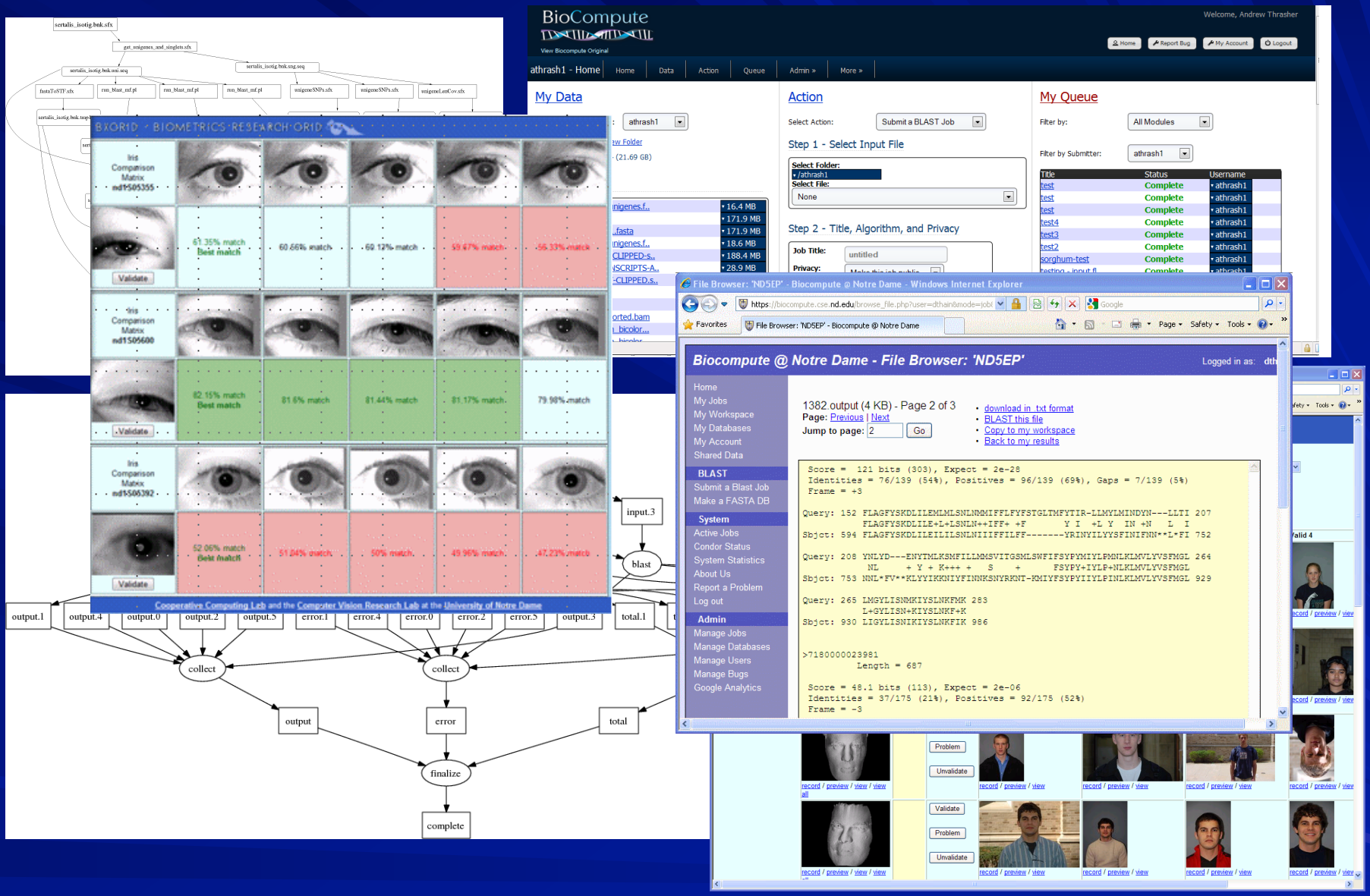
Shared-Nothing Running Condor

Shared-Disk Running SGE

Distributed Disks Running Hadoop

NFS

Michael Albrecht, Patrick Donnelly, Peter Bui, and Douglas Thain,
Makeflow: A Portable Abstraction for Cluster, Cloud, and Grid Computing.



Andrew Thrasher, Rory Carmichael, Peter Bui, Li Yu, Douglas Thain, and Scott Emrich, **Taming Complex Bioinformatics Workflows with Weaver, Makeflow, and Starch**, *Workshop on Workflows in Support of Large Scale Science*, pages 1-6, November, 2010

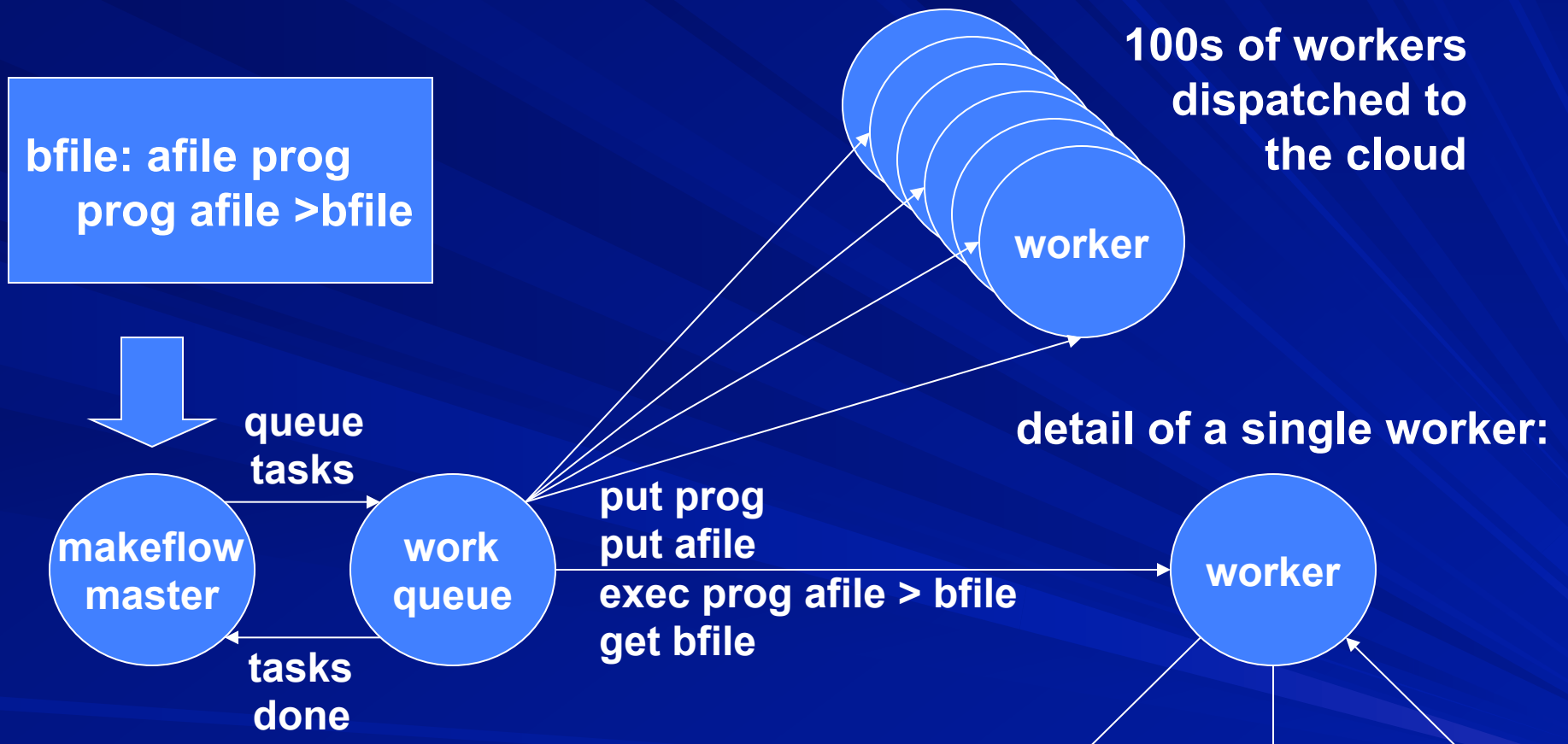
Weaver

```
# Weaver Code
jpgs = [str(i)+'. jpg ' for i in range (1000)]
conv = SimpleFunction('convert',out_suffix ='png ')
pngs = Map(conv,jpgs)
```

```
# Makeflow Code
0.png: 0.jpg /usr/bin/convert
       /usr/bin/convert 0.jpg 0.png
1.png: 1.jpg /usr/bin/convert
       /usr/bin/convert 1.jpg 1.png
...
999.png: 999.jpg /usr/bin
       /usr/bin/convert 999.jpg 999.png
```

Peter Bui, Li Yu and Douglas Thain,
Weaver: Integrating Distributed Computing Abstractions into Scientific Workflows using Python, *CLADE*, June, 2010.

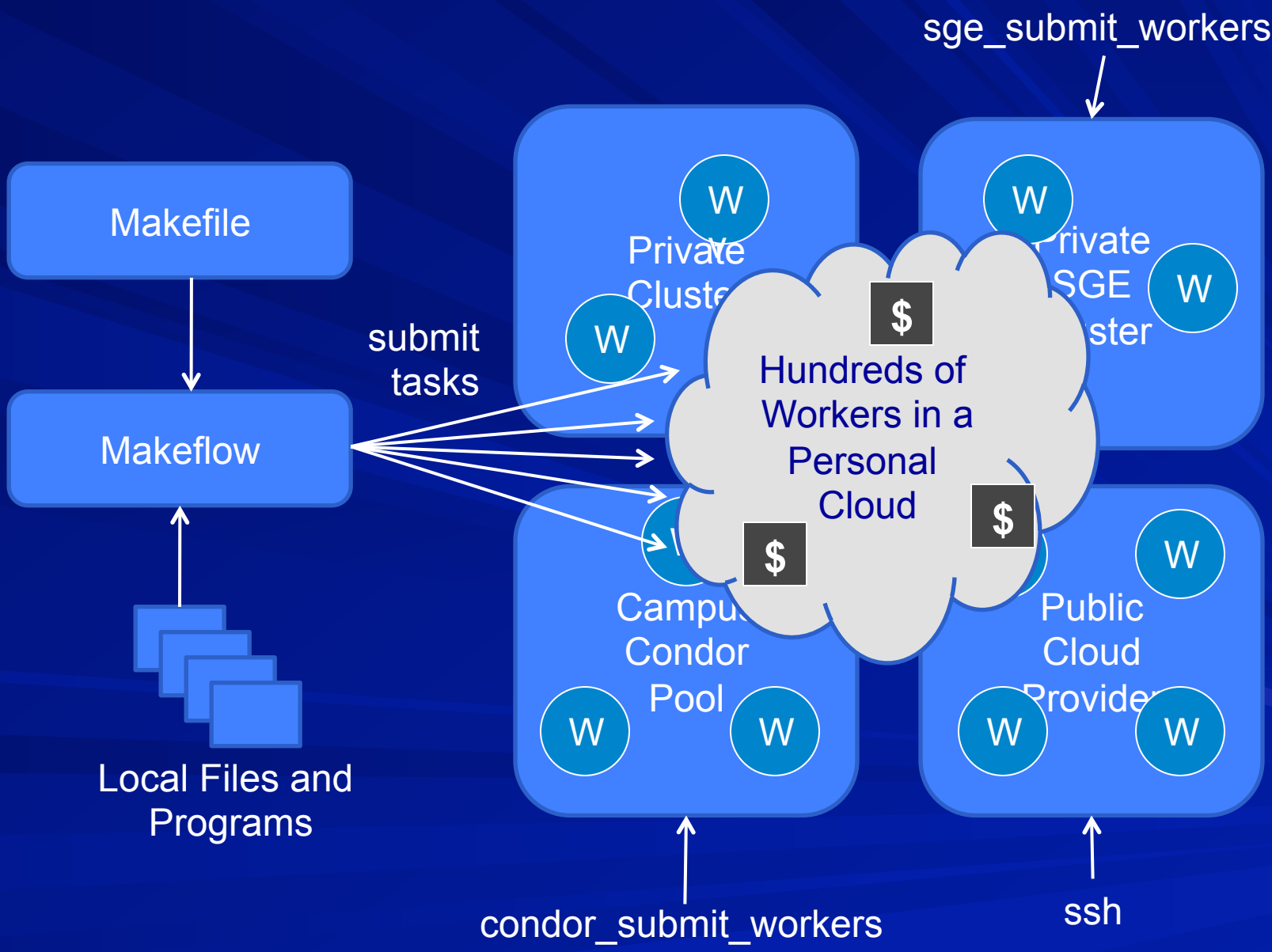
Makeflow and Work Queue



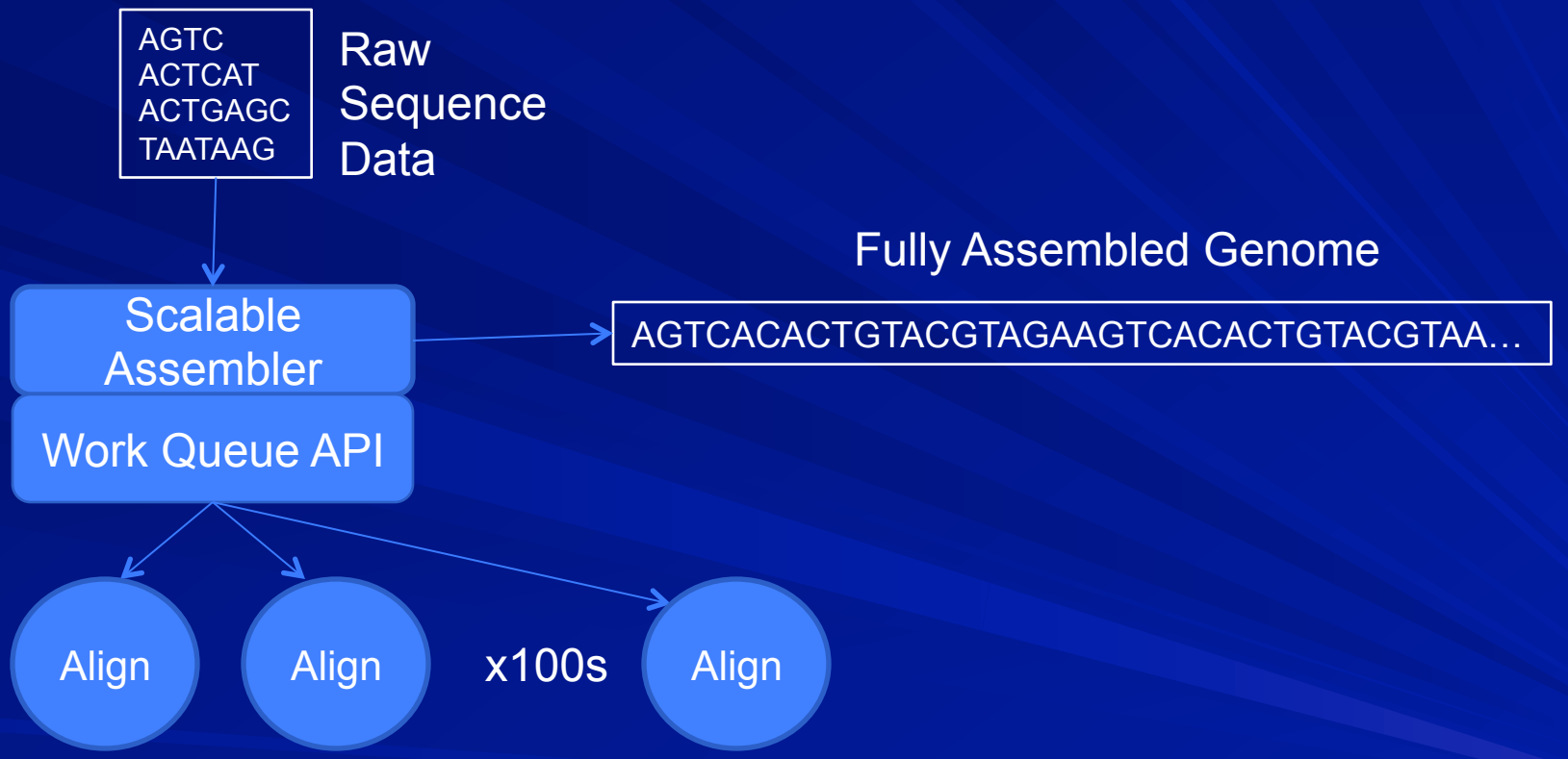
Two optimizations:

- Cache inputs and output.
- Dispatch tasks to nodes with data.

Makeflow and Work Queue



SAND - Scalable Assembler

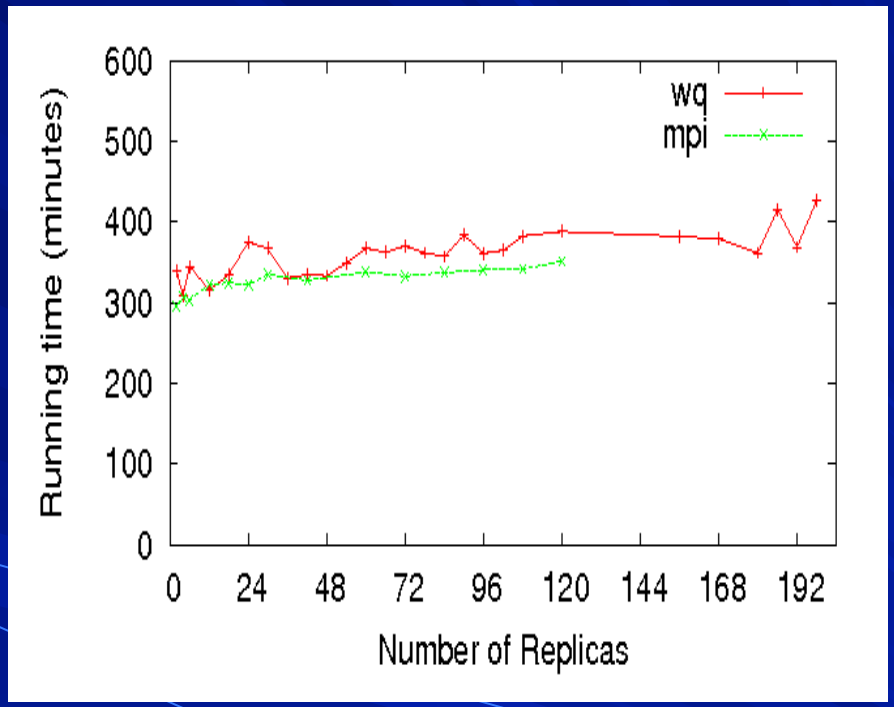


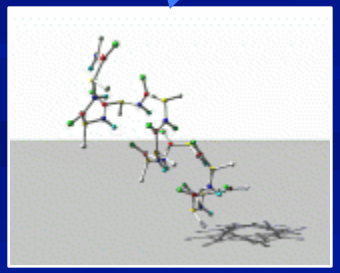
Christopher Moretti, Michael Olson, Scott Emrich, and Douglas Thain,
Highly Scalable Genome Assembly on Campus Grids,
Many-Task Computing on Grids and Supercomputers (MTAGS), November, 2009

Replica Exchange on WQ

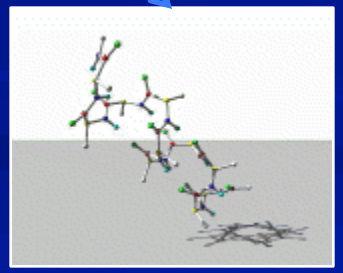
Replica Exchange

Work Queue API

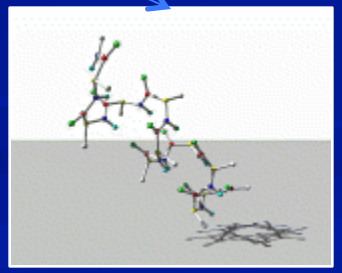




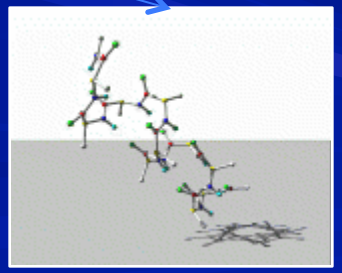
T=10K



T=20K



T=30K

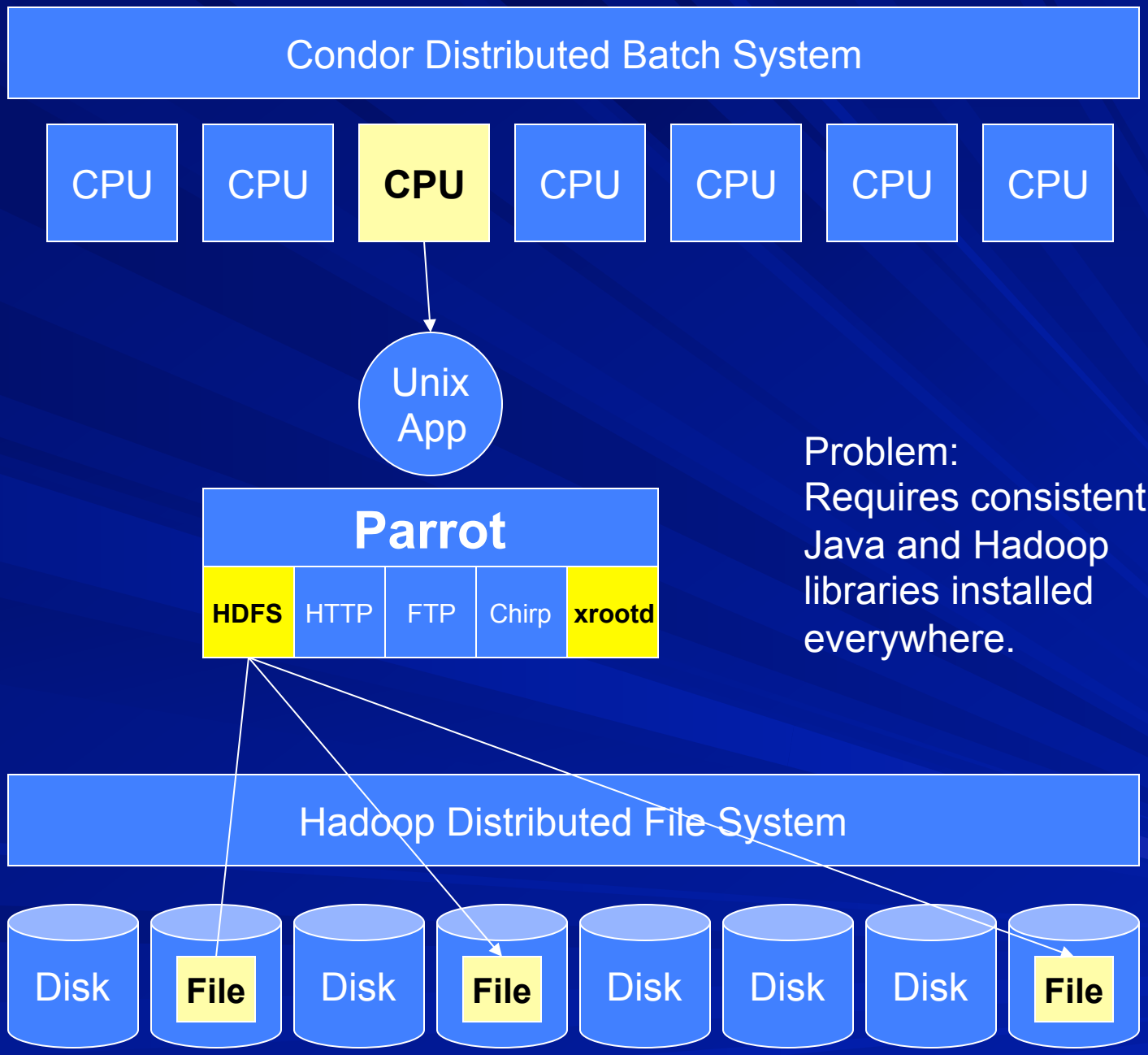


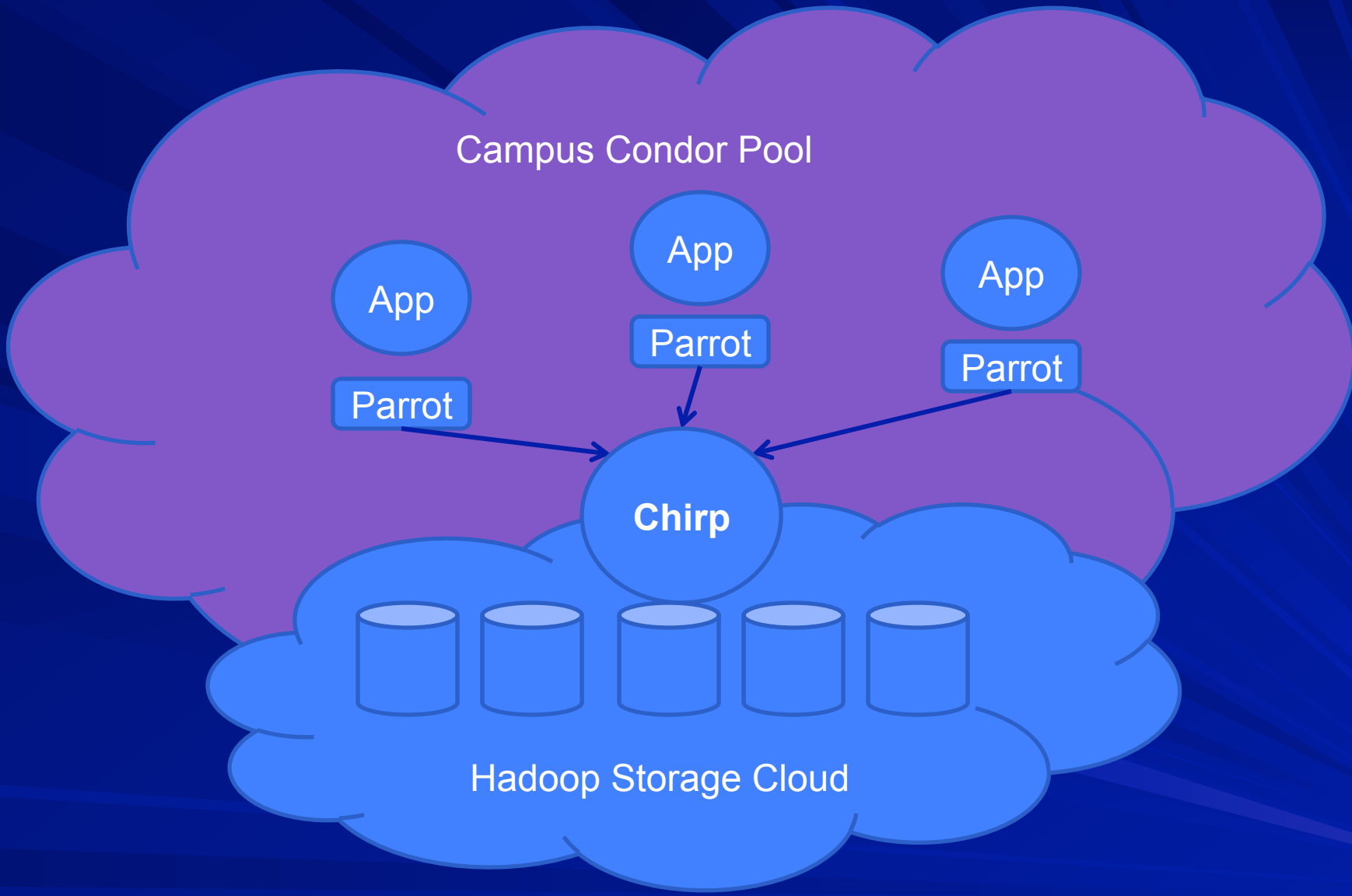
T=40K

Connecting Condor Jobs to Remote Data Storage

Parrot and Chirp

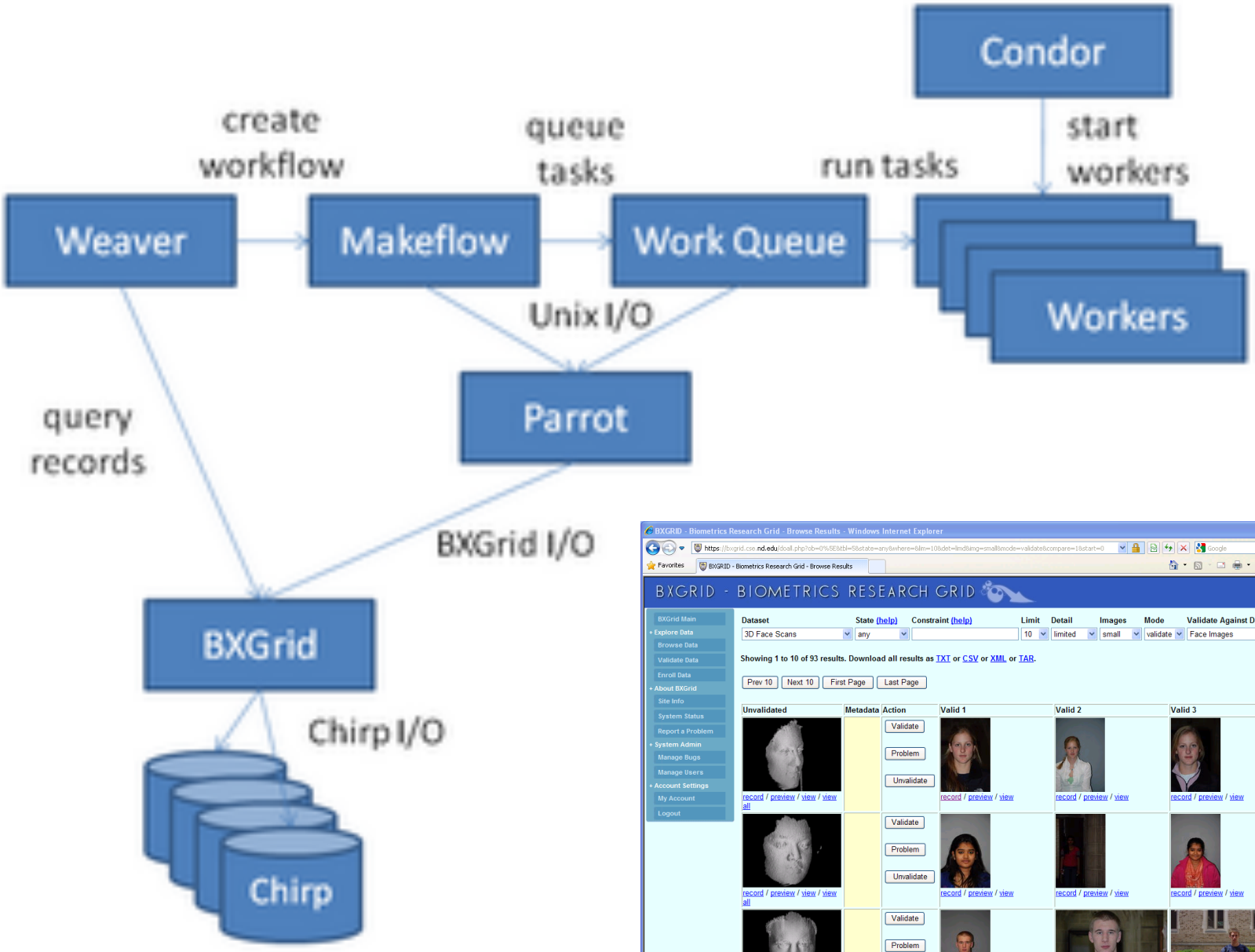
- Parrot – A User Level Virtual File System
  - Connects apps to remote data services:
  - HTTP, FTP, Hadoop, iRODS, XrootD, Chirp
  - No special privileges to install or use.
- Chirp – A Personal File Server
  - Export existing file services beyond the cluster.
    - Local disk, NFS, AFS, HDFS
  - Add rich access control features.
  - No special privileges to install or use.

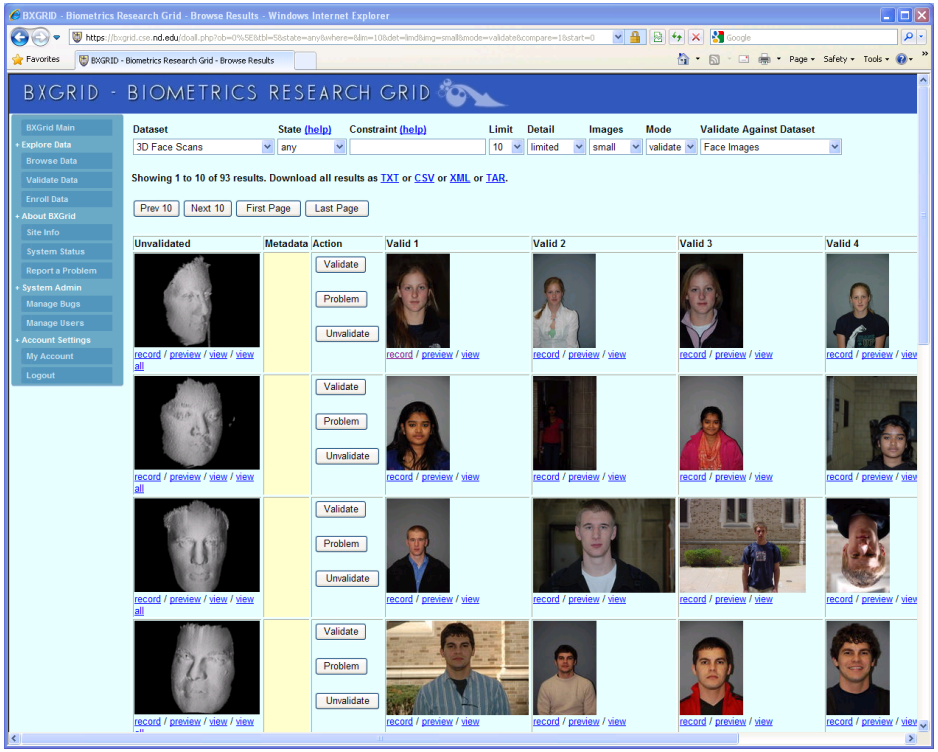

Condor Distributed Batch System
CPU
CPU
CPU
CPU
CPU
CPU
CPU
Unix App
Parrot
HDFS
HTTP
FTP
Chirp
xrootd
Problem:
Requires consistent Java and Hadoop libraries installed everywhere.
Hadoop Distributed File System
Disk
File
Disk
File
Disk
Disk
Disk
File



Patrick Donnelly, Peter Bui, Douglas Thain,
Attaching Cloud Storage to a Campus Grid Using Parrot, Chirp, and Hadoop ,
IEEE Cloud Computing Technology and Science, pages 488-495, November, 2010.

Putting it All Together


Condor
create workflow
queue tasks
run tasks
start workers
Weaver
Makeflow
Work Queue
Workers
Unix I/O
Parrot
query records
BXGrid I/O
BXGrid
Chirp I/O
Chirp


BXGRID - Biometrics Research Grid - Browse Results - Windows Internet Explorer
BXGRID - BIOMETRICS RESEARCH GRID
Dataset
State (help)
Constraint (help)
Limit
Detail
Images
Mode
Validate Against Dataset
3D Face Scans
any
limited
small
validate
Face Images
Showing 1 to 10 of 93 results. Download all results as TXT or CSV or XML or TAR.
Prev 10
Next 10
First Page
Last Page
Unvalidated
Metadata
Action
Valid 1
Valid 2
Valid 3
Valid 4
Validate
Problem
Unvalidate

Computer Science Challenges

- With multicore everywhere, we want to run multiple apps per machine, but the local OS is still very poor at managing resources.
- How many workers does a workload need? Can we even tell when we have too many or too few?
- How to automatically partition a data intensive DAG across multiple multicore machines?
- $$$ is now part of the computing interface. Does it make sense to get it inside the workflow and/or API?

What is Condor Compatible?

- Work right out of the box with Condor.
  - makeflow –T condor
  - condor_submit_workers
- Respect the execution environment.
  - Accept eviction and failure as normal.
  - Put data in the right place so it can be cleaned up automatically by Condor.
- Interoperate with public Condor interfaces.
  - Servers run happily under the condor_master.
  - Compatible with Chirp I/O via the Starter.

A Team Effort

- Faculty:
  - Patrick Flynn
  - Scott Emrich
  - Jesus Izaguirre
  - Nitesh Chawla
  - Kenneth Judd
- Grad Students
  - Hoang Bui
  - Li Yu
  - Peter Bui
  - Michael Albrecht
  - Patrick Donnely
  - Peter Sempolinski
  - Dinesh Rajan
- Undergrads
  - Rachel Witty
  - Thomas Potthast
  - Brenden Kokosza
  - Zach Musgrave
  - Anthony Canino

- NSF Grants CCF-0621434, CNS-0643229, and CNS 08-554087.

For More Information

- The Cooperative Computing Lab
  - http://www.nd.edu/~ccl
- Condor-Compatible Software:
  - Makeflow, Work Queue, Parrot, Chirp, SAND
- Prof. Douglas Thain
  - dthain@nd.edu