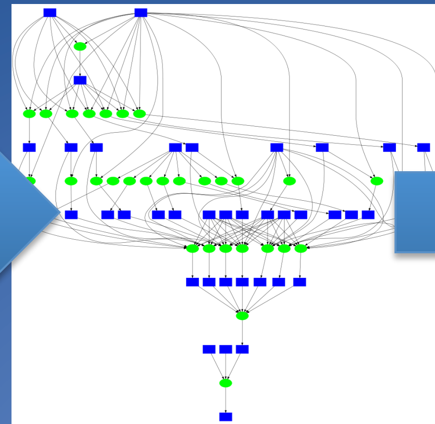




Provisioning Complex Software Environments for Scientific Applications

Prof. Douglas Thain, University of Notre Dame

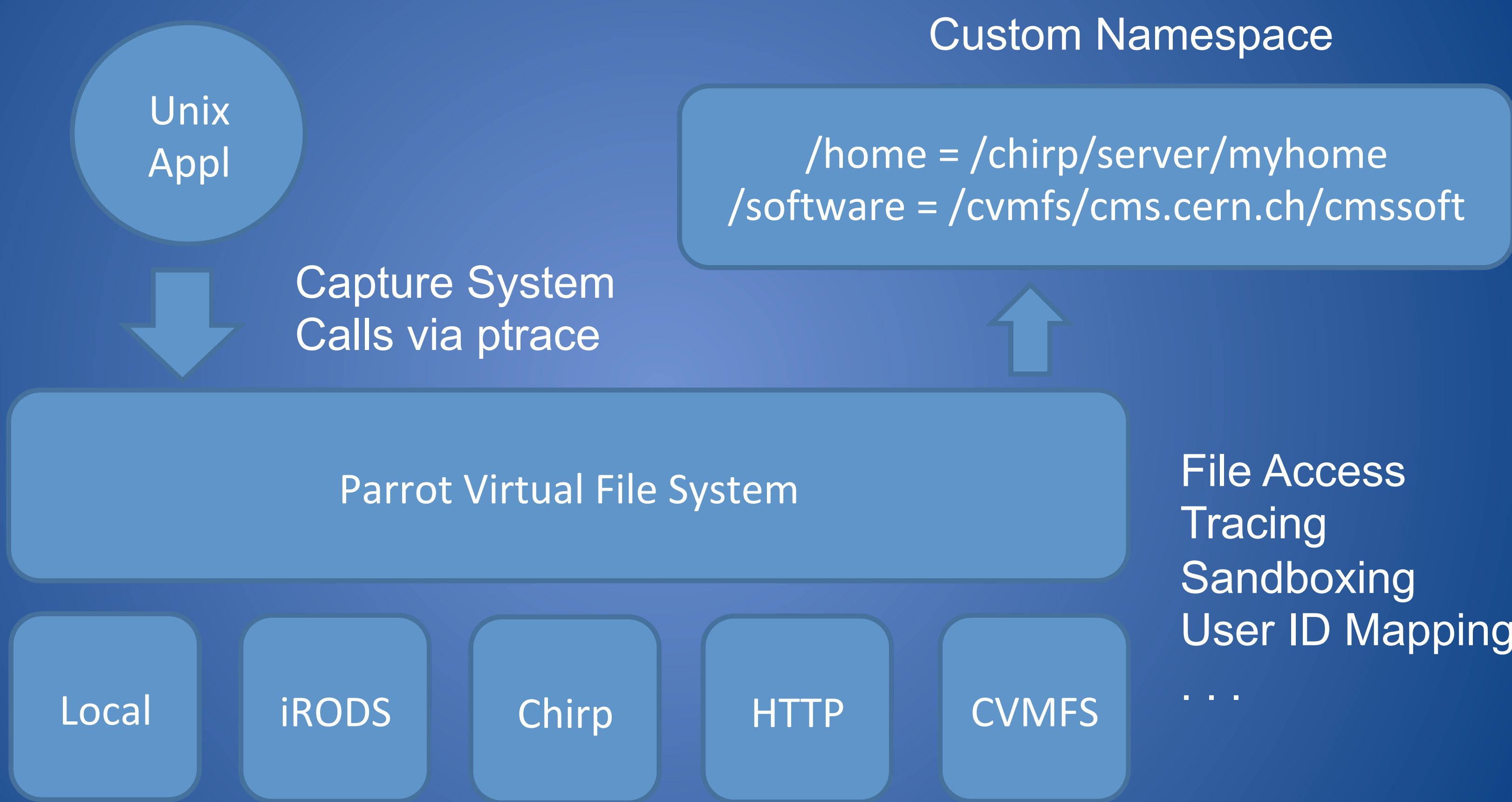
http://www.nd.edu/~dthain
dthain@nd.edu
@ProfThain

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 O(10,000) cores: clusters, clouds, grids.
- We *conduct computer science* research in the context of real people and problems.
- We *develop open source software* for large scale distributed computing.

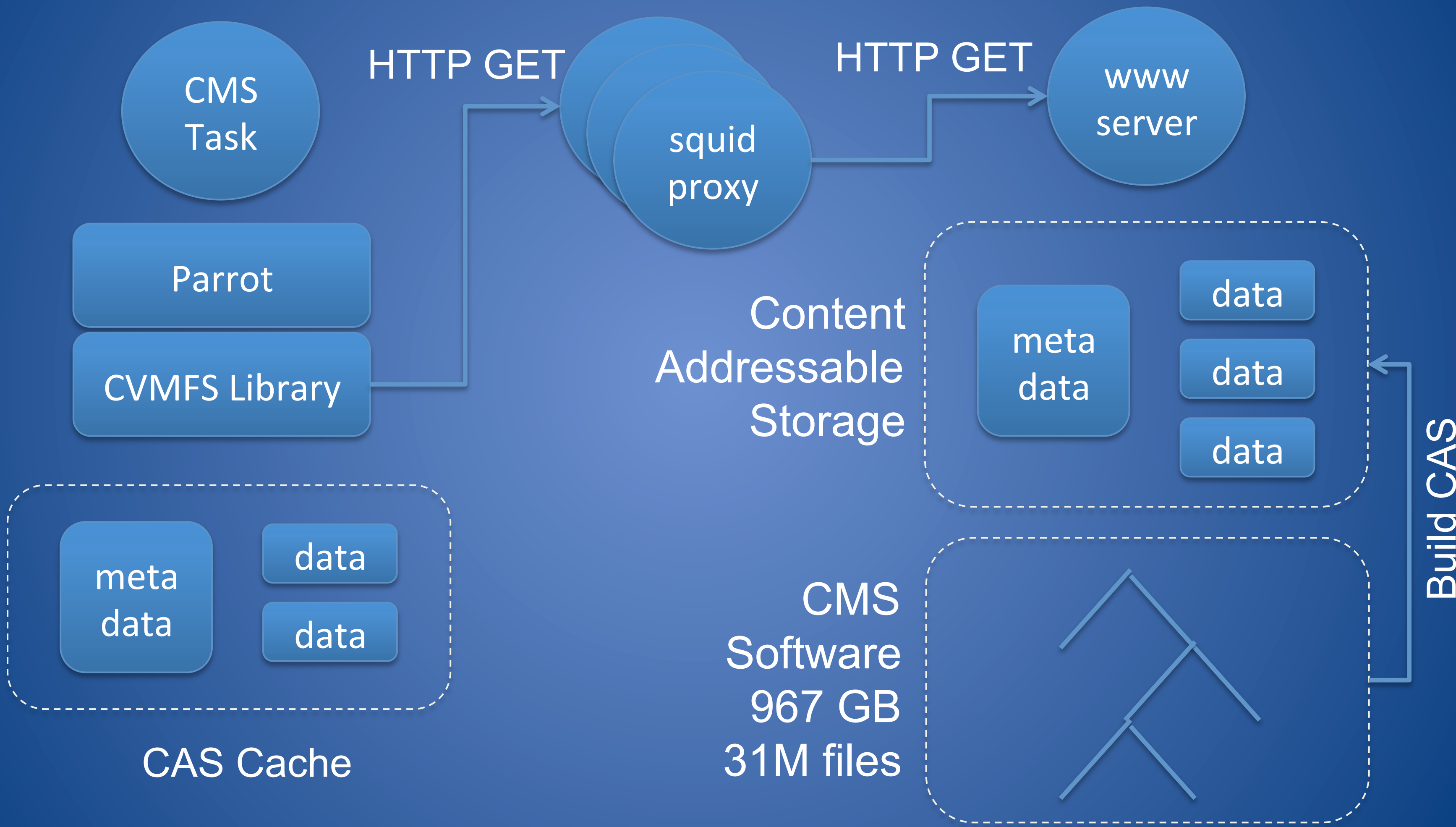
http://ccl.cse.nd.edu

Parrot Virtual File System

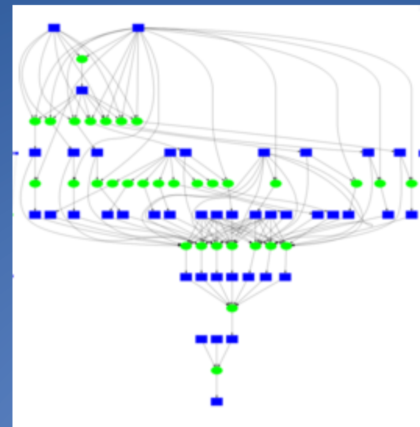


Douglas Thain, Christopher Moretti, and Igor Sfiligoi, **Transparently Distributing CDF Software with Parrot**, *Computing in High Energy Physics*, pages 1-4, February, 2006.

Parrot + CVMFS



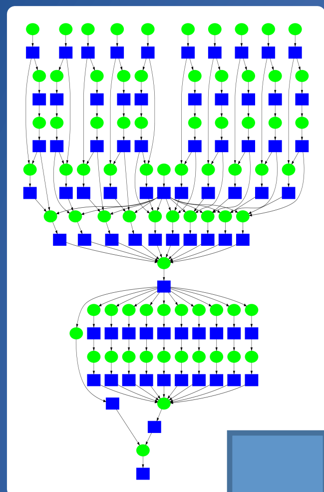
Jakob Blomer, Predrag Buncic, Rene Meusel, Gerardo Ganis, Igor Sfiligoi and Douglas Thain, **The Evolution of Global Scale Filesystems for Scientific Software Distribution**, *IEEE/AIP Computing in Science and Engineering*, **17**(6), pages 61-71, December, 2015. DOI: 10.1109/MCSE.2015.111



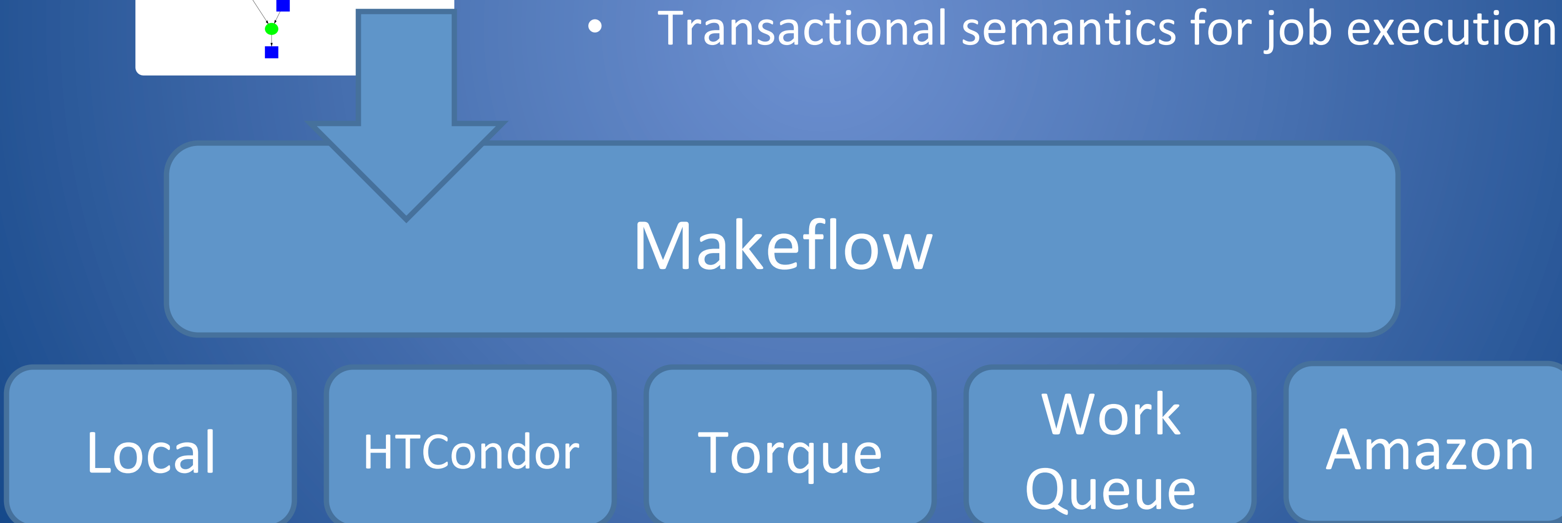
How do we run complex workflows on diverse computing resources?

Open Science Grid

Makeflow = Make + Workflow

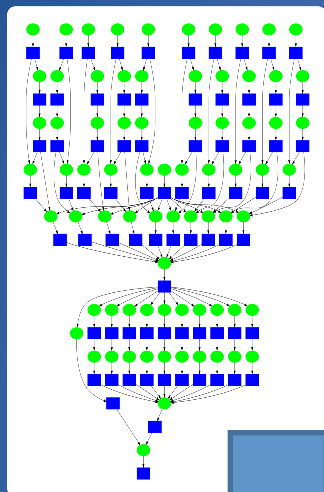


- Provides portability across batch systems.
- Enables parallelism (but not too much!)
- Fault tolerance at multiple scales.
- Data and resource management.
- Transactional semantics for job execution.



http://ccl.cse.nd.edu/software/makeflow

Makeflow = Make + Workflow



- Provides portability across batch systems.
- Enables parallelism (but not too much!)
- Fault tolerance at multiple scales.
- Data and resource management.
- Transactional semantics for job ex

Charles Zheng
(czheng2@nd.edu)

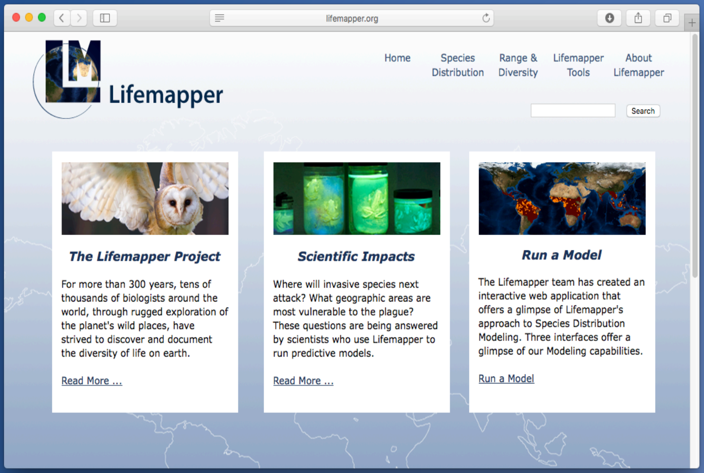
Makeflow

Local | HTCondor | Torque | Mesos | Kuber-netes

http://ccl.cse.nd.edu/software/makeflow

Example: Species Distribution Modeling

www.lifemapper.org



Full Workflow:

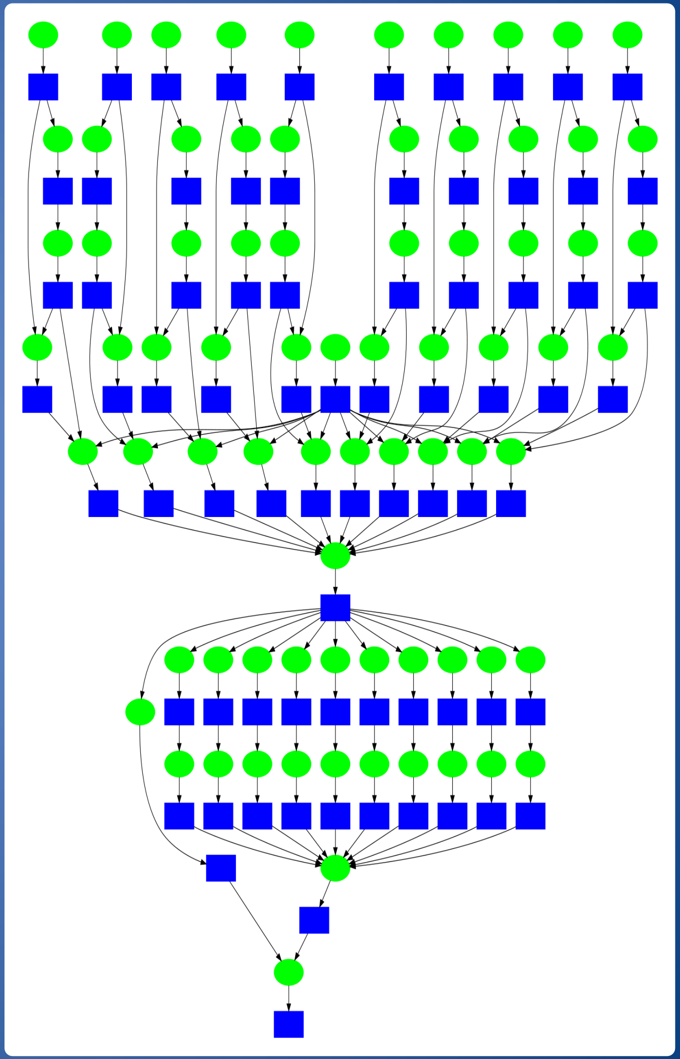
12,500 species

x 15 climate scenarios

x 6 experiments

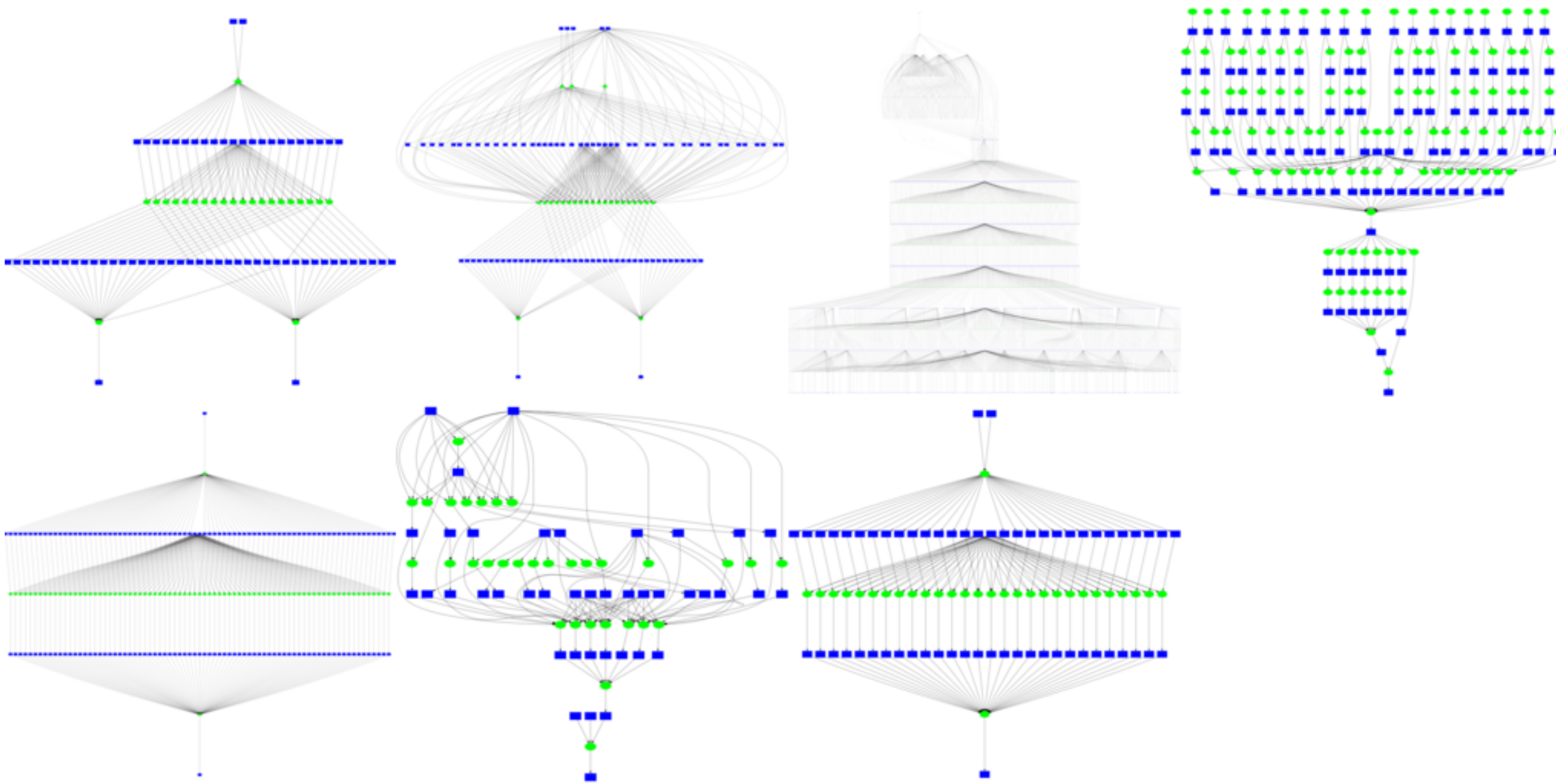
x 500 MB per projection

= 1.1M jobs, 72TB of output



Small Example: 10 species x 10 expts

More Examples



http://github.com/cooperative-computing-lab/makeflow-examples

Workflow Language Evolution

Classic "Make" Representation

```
output.5.txt : input.txt mysim.exe
     mysim.exe –p 10 input.txt > output.5.txt
```

Tim Shaffer
(tshaffe1@nd.edu)

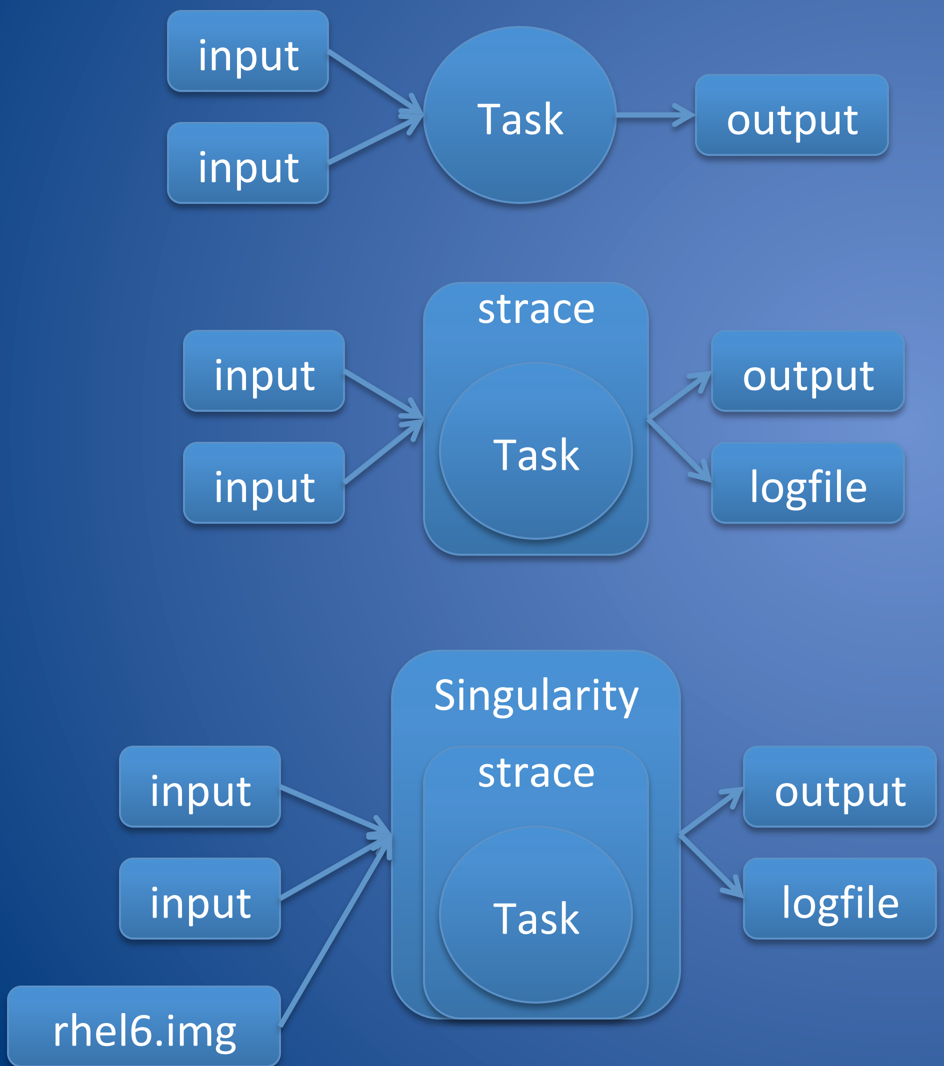
JSON Representation of One Job

```
{
"command"  :  "mysim.exe –p 10 input.txt > output.5.txt",
"outputs" :  [ "output.5.txt" ],
"inputs"  :  [ "input.dat", "mysim.exe" ]
}
```

JX (JSON + Expressions) for Multiple Jobs

```
{
"command"  :  "mysim.exe –p " + x*2 + " input.txt > output." + x + " .txt",
"outputs" :  [ "output" + x + "txt" ],
"inputs"  :  [ "input.dat", "mysim.exe" ]
} for x in [ 1, 2, 3, 4, 5 ]
```

Elaborating Jobs with Wrappers



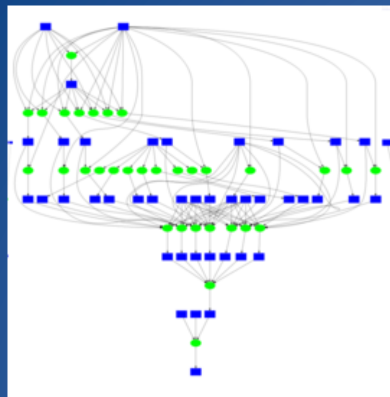
Original Job

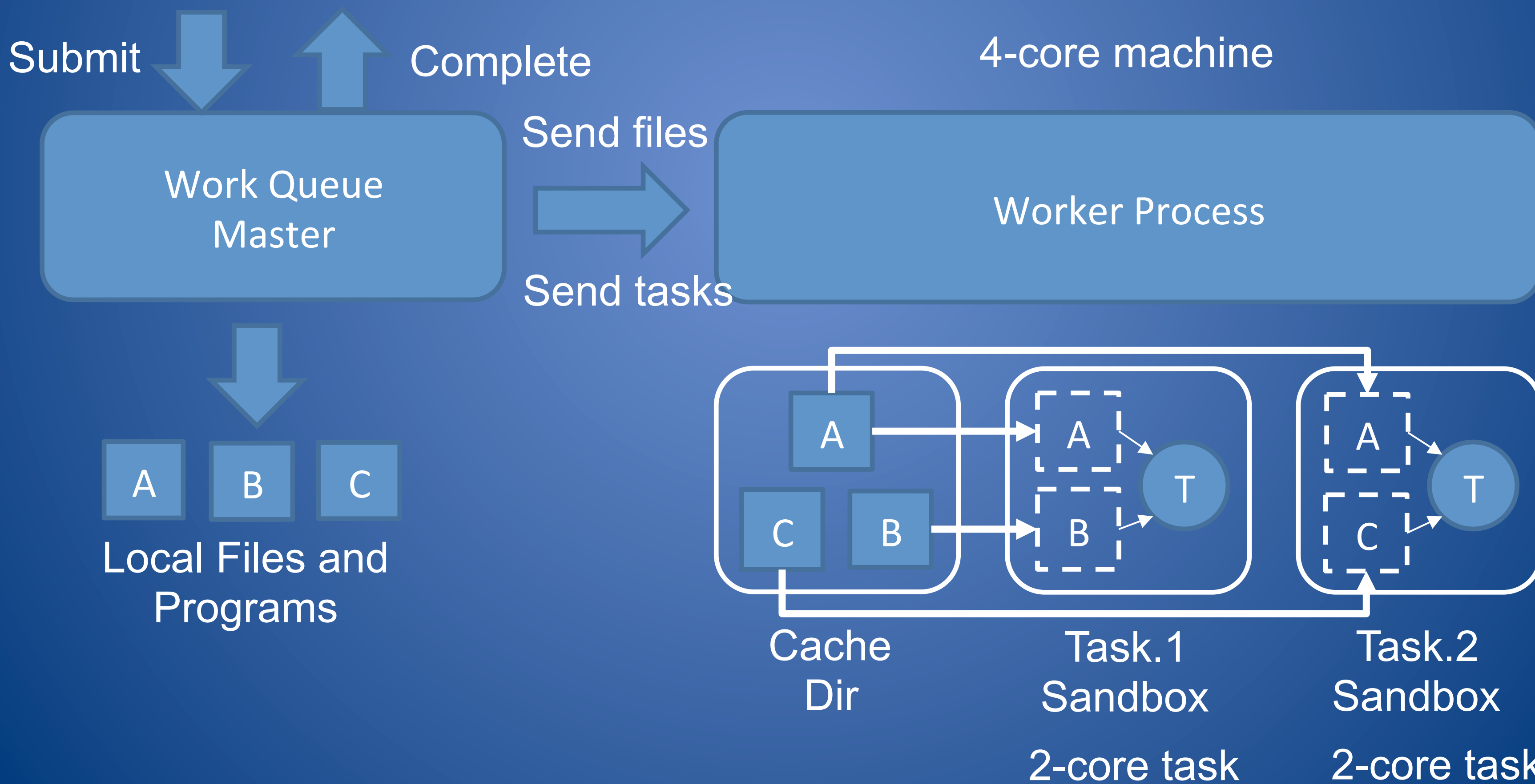
Add Debug Tool

Add Container Environment

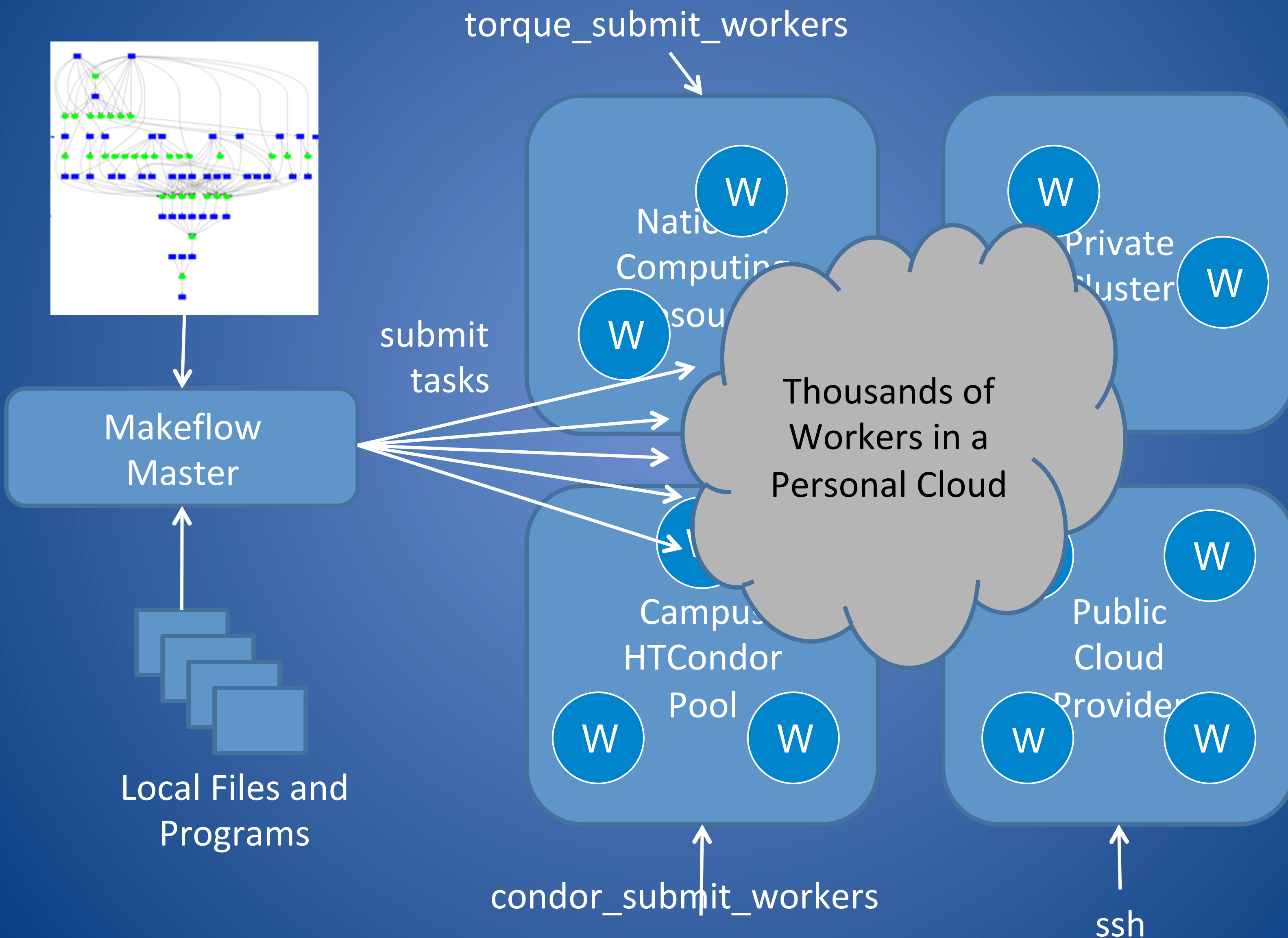
Nick Hazekamp
(nhazekam@nd.edu)

Work Queue Architecture





Makeflow + Work Queue



Problem: Software Deployment

- Getting software installed on a new site is a big pain! The user (probably) knows the top level package, but doesn't know:
  – How they set up the package (sometime last year)
  – Dependencies of the top-level package.
  – Which packages are system default vs optional
  – How to import the package into their environment via PATH, LD_LIBRARY_PATH, etc.
- Many scientific codes are not distributed via rpm, yum, pkg, etc. (and user isn't root)

Even Bigger Differences:

- Hardware Architecture
  - X86-64, KNL, Blue Gene, GPUs, FPGAs, . . .
- Operating System
  - Green Avocado Linux, Blue Dolphin Linux, Red Rash Linux, . . .
- Batch System or Resource Manager
  - HTCondor, PBS, Torque, Cobalt, Mesos, . . .
- Container Technology
  - None, Docker, Singularity, CharlieCloud, Shifter, …
- Running Services
  - FUSE, CVMFS, HTTP Proxy, Frontier, . . .
- Network Configuration
  - Public/Private, Incoming/Outgoing, Firewalls

Our Approach:

- Provide tools that make a flexible mapping between the user's intent and the local site:
  - "I need OS RHEL6"
    - Check if already present, otherwise run in container.
  - "I need container X.img"
    - Try Docker, try Singularity, try CharlieCloud.
  - "I need /cvmfs/repo.cern.ch"
    - Look for /cvmfs; activate FUSE; build/run parrot.
  - "I need software package X"
    - Look for X installed locally, else build from recipe.

Delivering ***Platforms*** with RunOS

"runos slc6 – mysim.exe"

Kyle Sweeney
(ksweene3@nd.edu)

mysim.exe
slc6
Site A

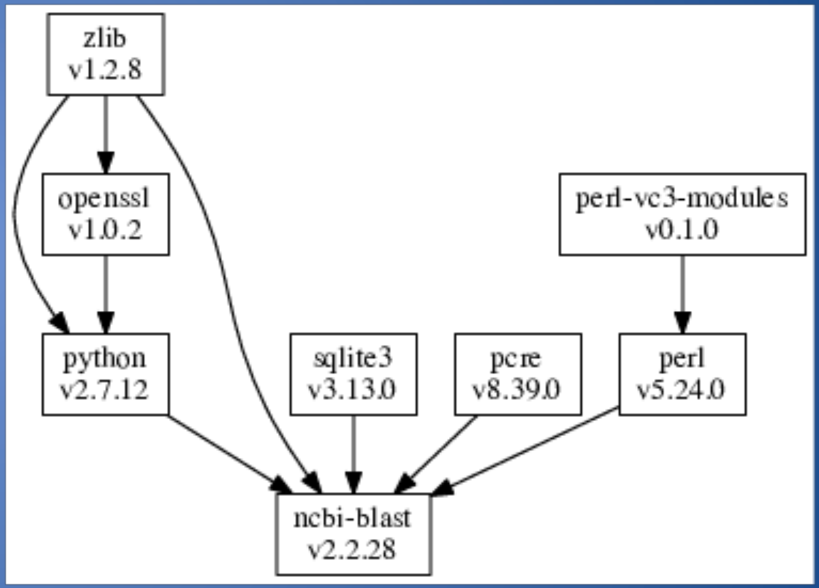
mysim.exe
slc6
singularity
rhel7
Site B

mysim.exe
slc6
charliecloud
debian45
Site C

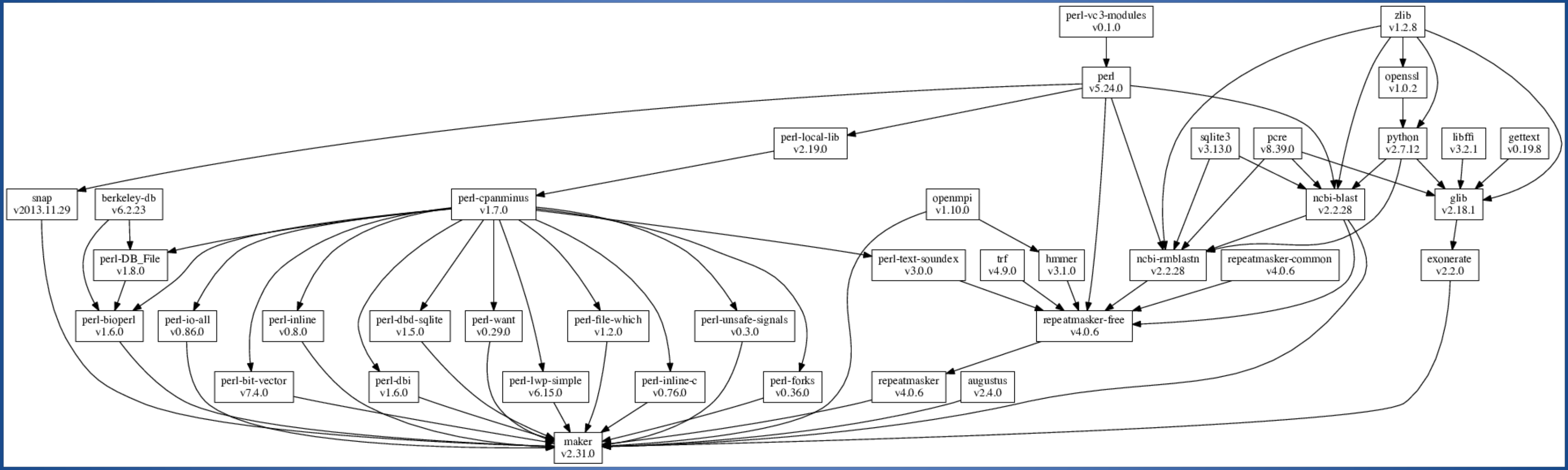
Delivering ***Software*** with VC3-Builder

Typical User Dialog Installing BLAST

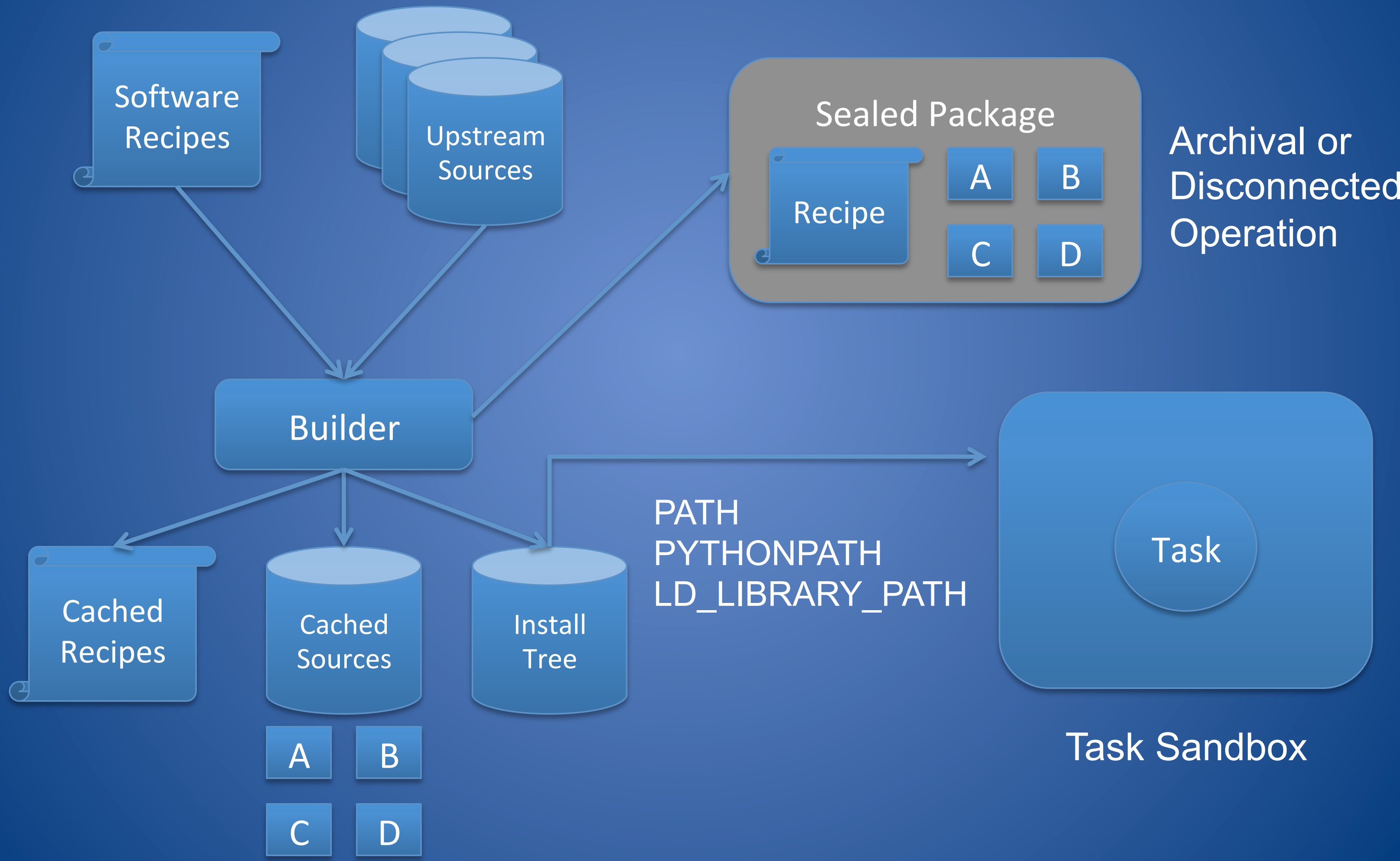
"I just need BLAST."
"Oh wait, I need Python!"
"Sorry, Python 2.7.12"
"Python requires SSL?"
"What on earth is pcre?"
"I give up!"



MAKER Bioinformatics Pipeline



VC3-Builder Architecture



"vc3-builder –require ncbi-blast"

..Plan: ncbi-blast => [,]
..Try: ncbi-blast => v2.2.28
....Plan: pe
....Try: pe
....could not
....Try: pe
....could not
....Try: pe
......Plan: p
......Try: p
......Success
....Success:
....Plan: py
....Try: py
....could not
....Try: py
......Plan: c
……………
Downloading
details: /tmp/test/vc3-root/x86_64/redhat6/python/v2.7.12/python-build-log
processing for ncbi-blast-v2.2.28
preparing 'ncbi-blast' for x86_64/redhat6
Downloading 'ncbi-blast-2.2.28+-x64-linux.tar.gz' from http://download.virtualclusters.org…
details: /tmp/test/vc3-root/x86_64/redhat6/ncbi-blast/v2.2.28/ncbi-blast-build-log

(New Shell with Desired Environment)

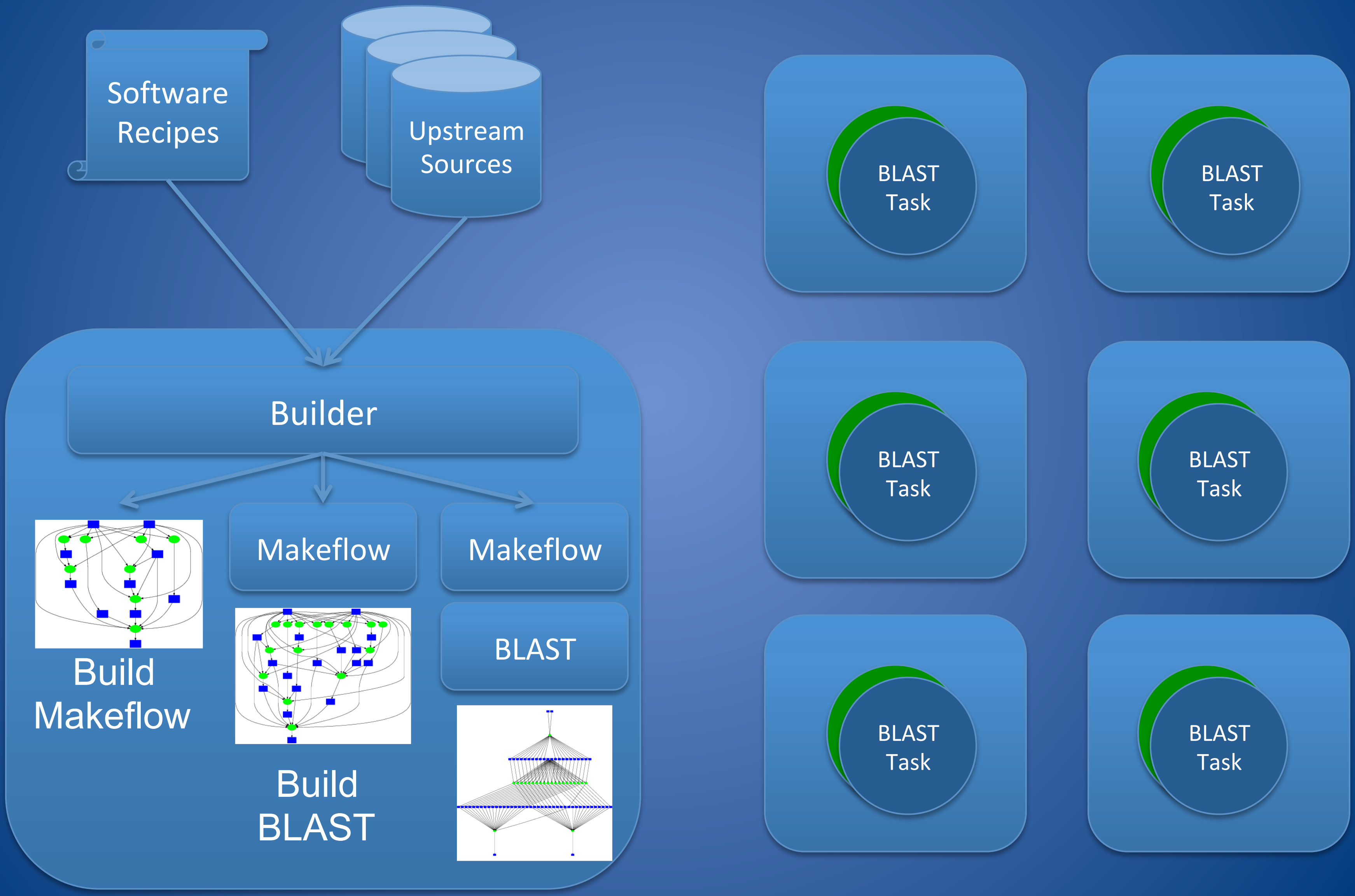
```
bash$  which blastx
/tmp/test/vc3-root/x86_64/redhat6/ncbi-blast/v2.2.28/
bin/blastx

bash$ blastx –help
USAGE
  blastx [-h] [-help] [-import_search_strategy filename]
   . . .

bash$ exit
```

Problem: Long Build on Head Node

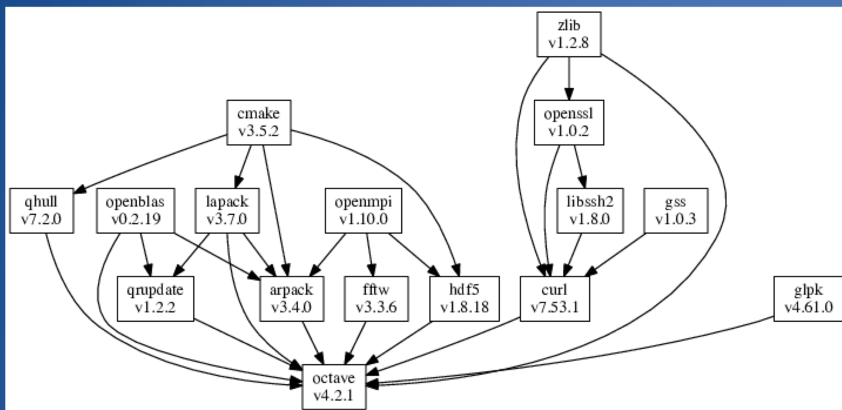
- Many computing sites limit the amount of work that can be done on the head node, so as to maintain quality of service for everyone.
- Solution: Move the build jobs out to the cluster nodes. (Which may not have network connections.)
- Idea: Reduce the problem to something we already know how to do: Workflow!
- But how do we bootstrap the workflow software? With the builder!

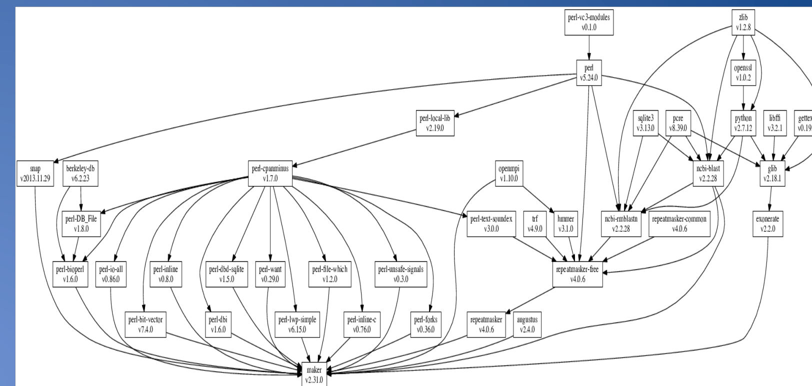
```
vc3-builder
    --require makeflow
    --require ncbi-blast
    --
    makeflow –T condor blast.mf
```

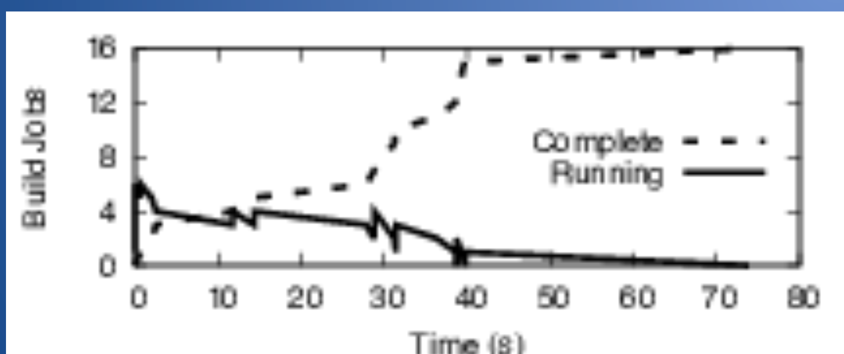
Bootstrapping a Workflow

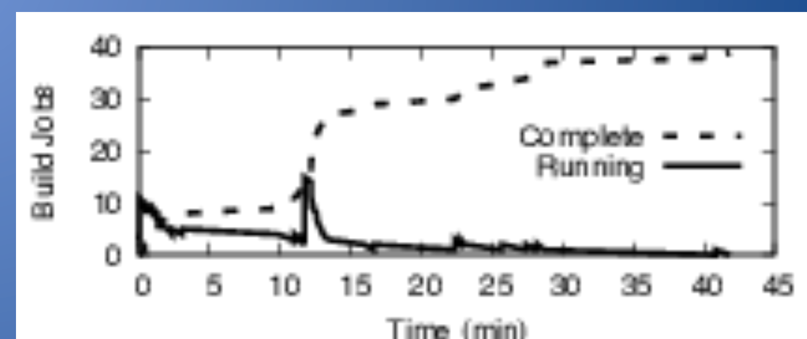


Example Applications









Octave

MAKER

Benjamin Tovar, Nicholas Hazekamp, Nathaniel Kremer-Herman, and Douglas Thain, Automatic Dependency Management for Scientific Applications on Clusters, *IEEE International Conference on Cloud Engineering (IC2E)* , April, 2018.

Delivering ***Services*** with VC3-Builder

"vc3-builder –require cvmfs"

..Plan: cvmfs => [,]
..Try: cvmfs => v2.0.0
....Plan: parrot => [v6.0.16,]
....Try: pa
......Plan: c
......Try: c
........Plan:
........Try:
........Succes
......Fail-prer
........Plan:
........Try:
..........Plan:
..........Try:
..........Succe
........could n
........Try:
..........Plan:
..........Try:
..........Success: perl-vc3-modules v0.1.0 => [v0.1.0,]
........could not add any source for: perl v5.016 => [v5.10.0, v5.10001.0]
........Try: perl => v5.24.0
..........Plan: perl-vc3-modules => [v0.001.000,]
..........Try: perl-vc3-modules => v0.1.0
..........Success: perl-vc3-modules v0.1.0 => [v0.1.0,]
........Success: perl v5.24.0 => [v5.10.0, v5.10001.0]

(New Shell with Desired Environment)

bash$ **ls /cvmfs/oasis.opensciencegrid.org**

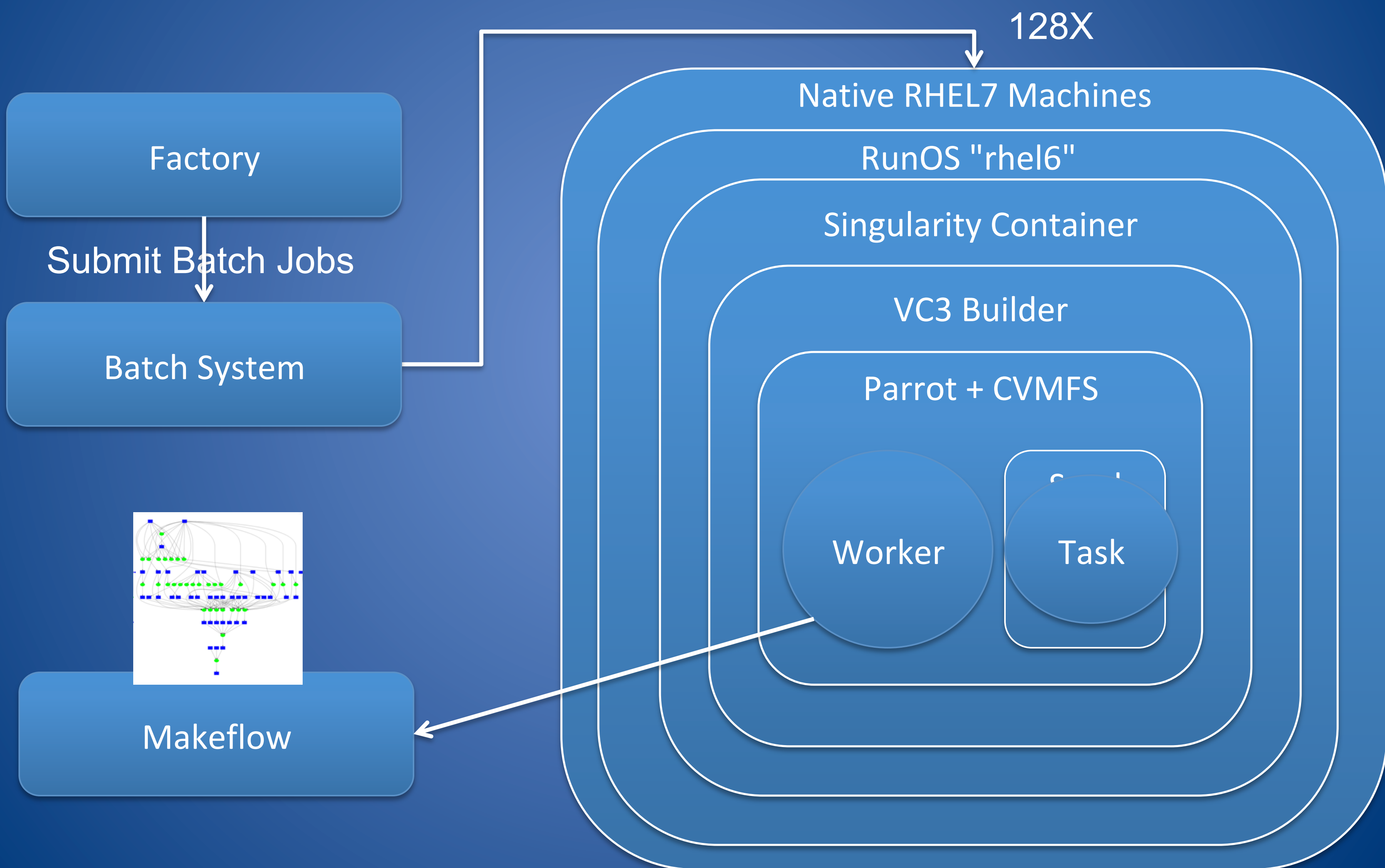
atlas	**csiu**	**geant4**	**ilc**	**nanohub**	**osg-software**
auger	**enmr**	**glow**	**ligo**	**nova**	**sbgrid**
cmssoft	**fermilab**	**gluex**	**mis**	**osg**	

snoplussnolabca
. . .

bash$ exit

Putting it All Together

Request 128 nodes of16 cores, 4G RAM, 16G disk
with RHEL6 operating system, CVMFS and Maker software installed:

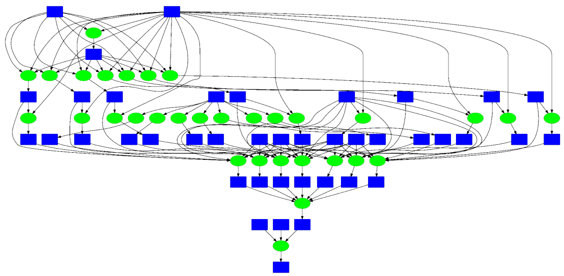


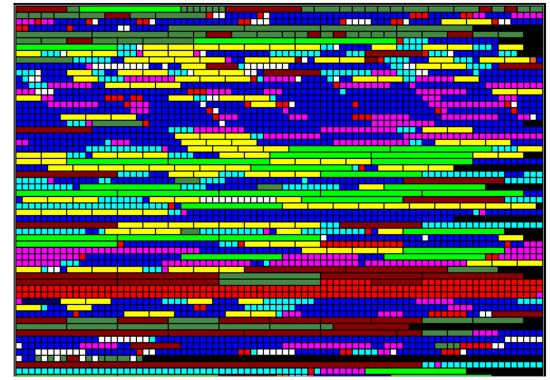
VC3: Virtual Clusters for Community Computation

Douglas Thain, University of Notre Dame
Rob Gardner, University of Chicago
John Hover, Brookhaven National Lab

http://virtualclusters.org

You have developed a large scale workload which runs successfully at a University cluster.





Now, you want to migrate and expand that application to national-scale infrastructure.
(And allow others to easily access and run similar workloads.)

Open Science Grid

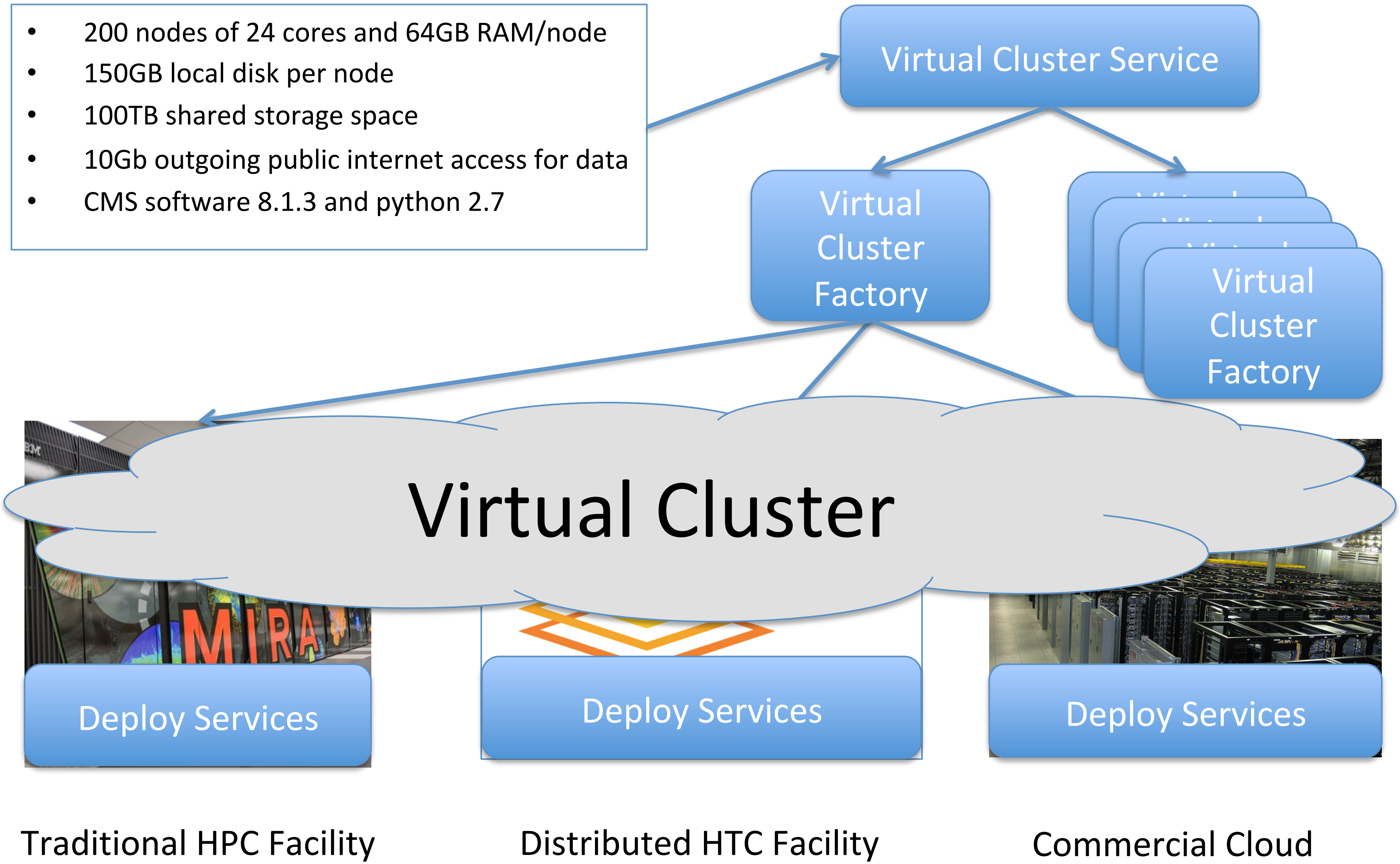
Traditional HPC Facility

Distributed HTC Facility

Commercial Cloud

Concept: Virtual Cluster

- 200 nodes of 24 cores and 64GB RAM/node
- 150GB local disk per node
- 100TB shared storage space
- 10Gb outgoing public internet access for data
- CMS software 8.1.3 and python 2.7



Some Thoughts:

- Make software dependencies more explicit.
  - Proposed: Nothing should be available by default, all software should require an "import" step.
- Layer tools with common abstractions:
  - Factory -> HTCondor -> Singularity -> Builder -> Worker
  - Provision -> Schedule -> Contain -> Build-> Execute
- Need better, portable, ways of expressing:
  - What software environment the user wants.
  - What environment the site provides.
- The ability to nest environments is critical!

Acknowledgements

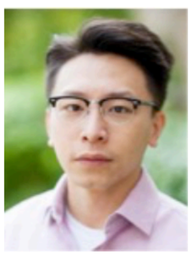
People in the Cooperative Computing Lab

Douglas Thain
Director

Benjamin Tovar
Research Soft. Engineer

Peter Ivie

Nicholas Hazekamp



Charles Zheng

Nathaniel Kremer-Herman

Tim Shaffer

Kyle Sweeney

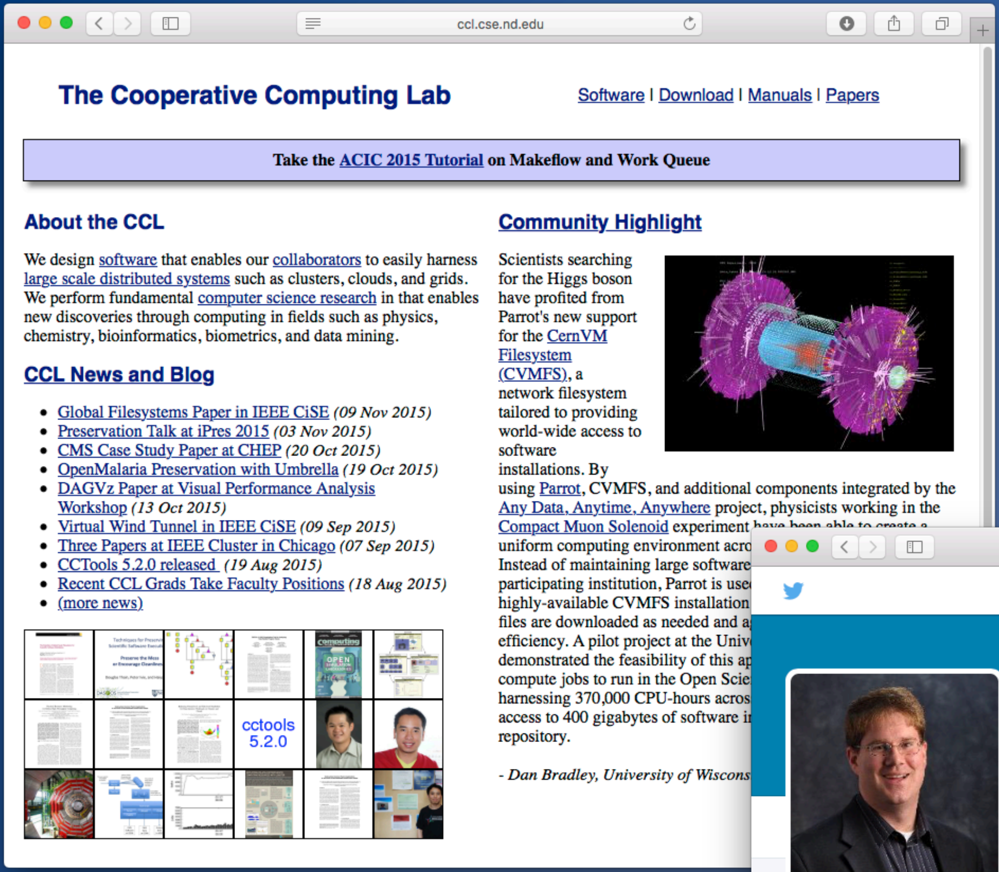
Notre Dame CMS:
Kevin Lannon
Mike Hildreth
Kenyi Hurtado

Univ. Chicago:
Rob Gardner
Lincoln Bryant
Suchandra Thapa
Benedikt Riedel

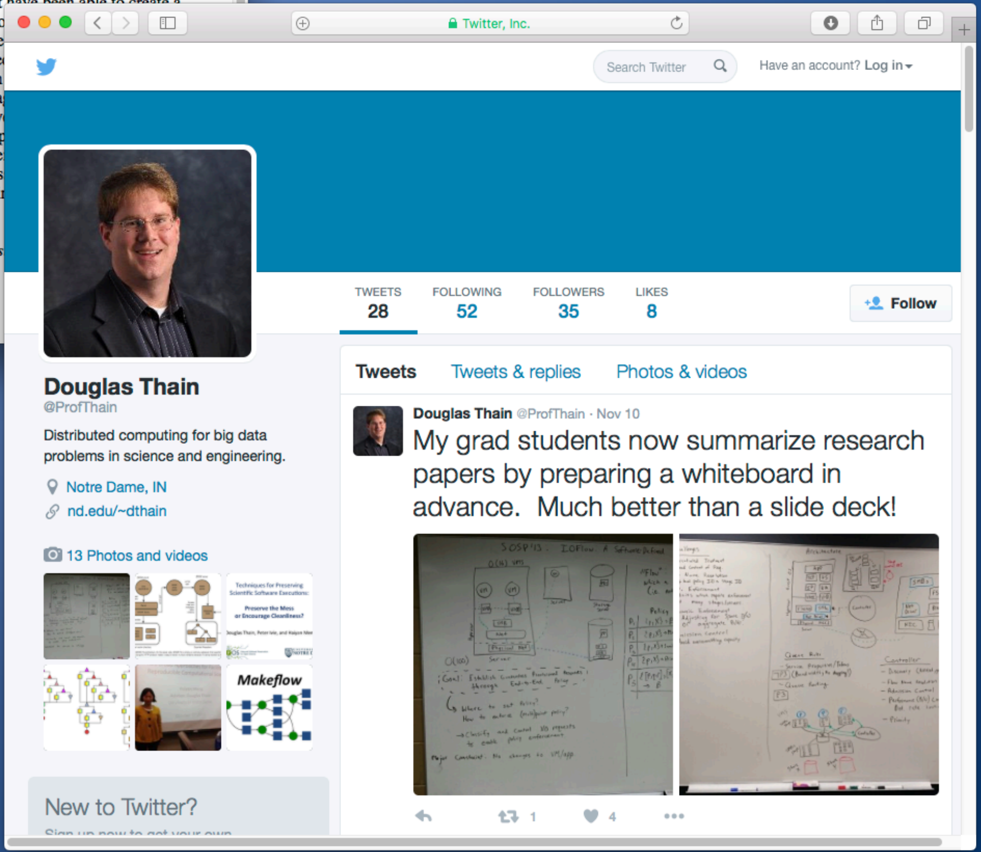
Brookhaven Lab:
John Hover
Jose Caballero

DE-SC0015711
VC3: Virtual Clusters for Community Computation

ACI-1642409
SI2-SSE: Scaling up Science on Cyberinfrastructure with the Cooperative Computing Tools



http://ccl.cse.nd.edu



@ProfThain