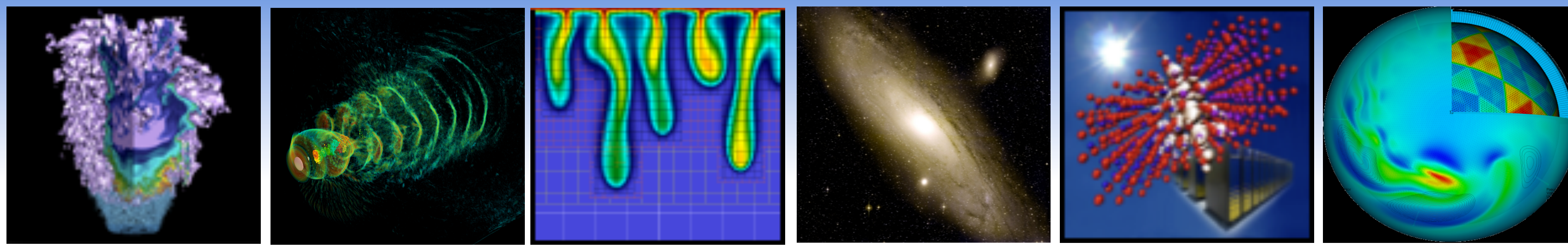


My Cray can do that?

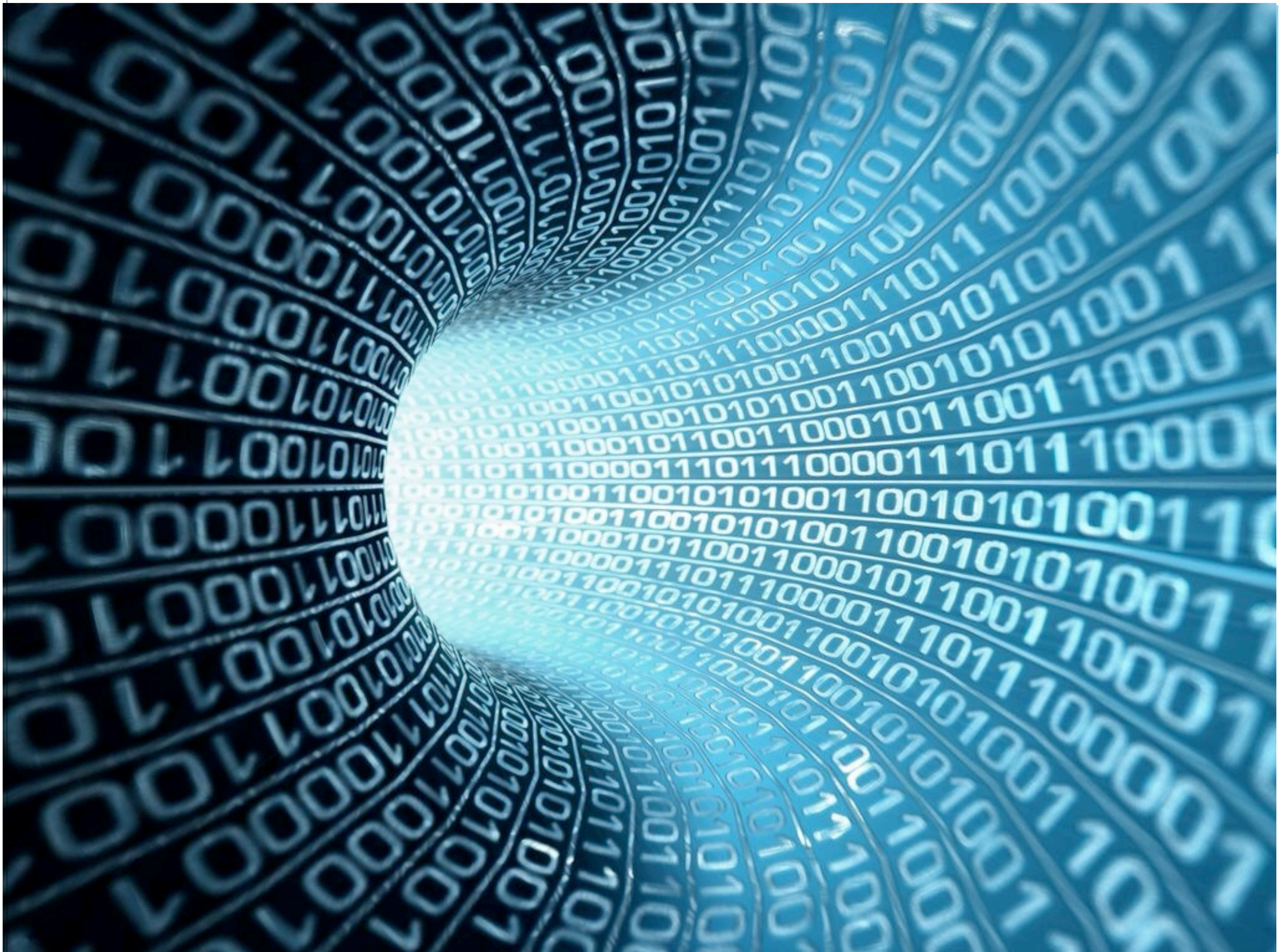
Supporting Diverse Workloads on the Cray XE6

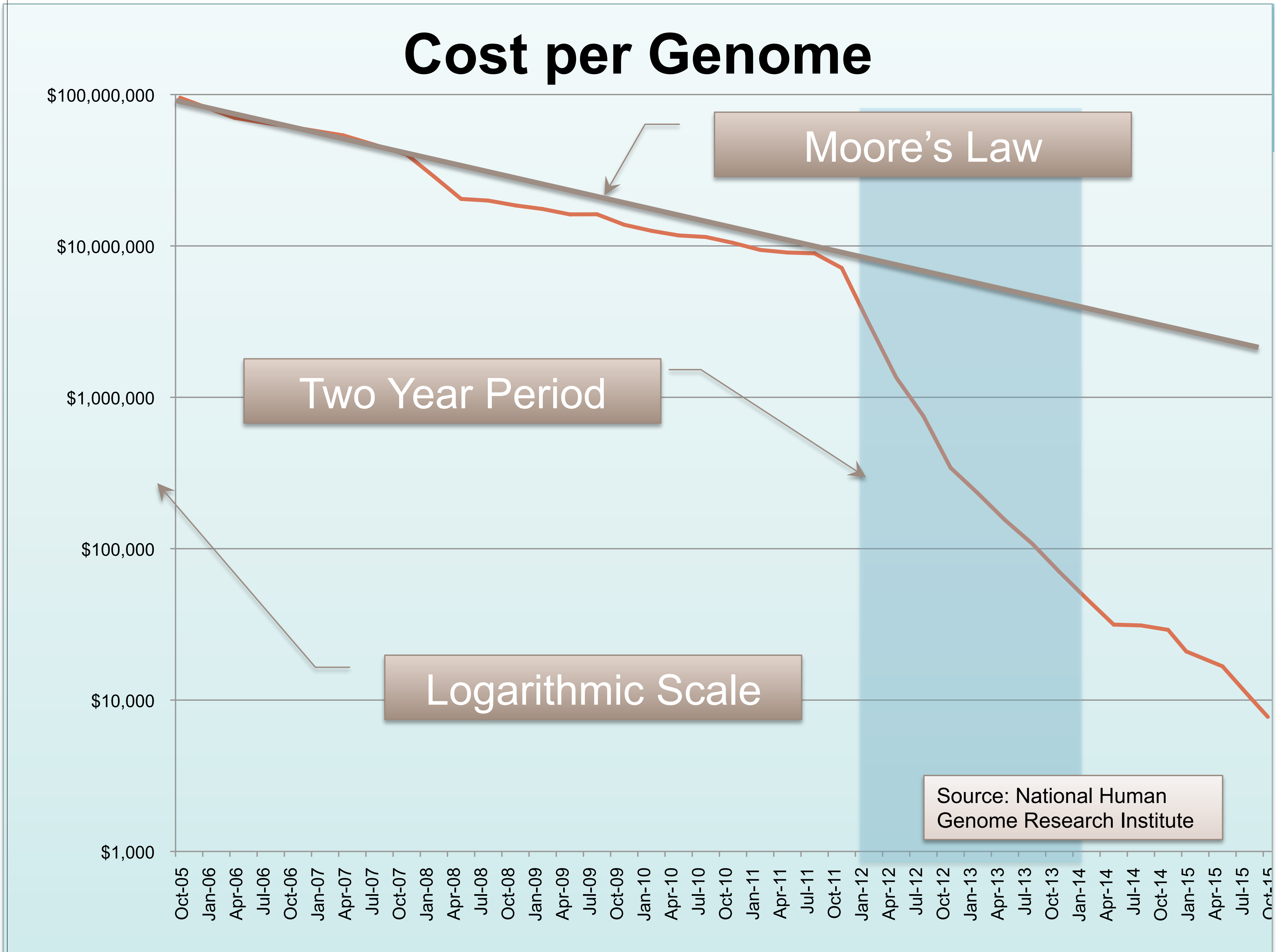
Shane Canon
Lavanya Ramakrishnan
Jay Srinivasan

Lawrence Berkeley National Lab

May 1, 2012

U.S. DEPARTMENT OF ENERGY | Office of Science





Cost per Genome
$100,000,000
$10,000,000
$1,000,000
$100,000
$10,000
$1,000
Moore's Law
Two Year Period
Logarithmic Scale
Source: National Human Genome Research Institute
Oct-05
Jan-06
Apr-06
Jul-06
Oct-06
Jan-07
Apr-07
Jul-07
Oct-07
Jan-08
Apr-08
Jul-08
Oct-08
Jan-09
Apr-09
Jul-09
Oct-09
Jan-10
Apr-10
Jul-10
Oct-10
Jan-11
Apr-11
Jul-11
Oct-11
Jan-12
Apr-12
Jul-12
Oct-12
Jan-13
Apr-13
Jul-13
Oct-13
Jan-14
Apr-14
Jul-14
Oct-14
Jan-15
Apr-15
Jul-15

Home Apps Support About References

Profile for scanon@gmail.com :: Logout

MATERIALS PROJECT

A Materials Genome Approach

Accelerating materials discovery through advanced scientific computing and innovative design tools.

Enter formulas

e.g., Fe_2O_3 Fe_3O_4

Search

Database Statistics

19120 materials

3050 bandstructures

214 intercalation batteries

4158 conversion batteries

Materials Explorer

Search for materials information by chemistry, composition, or property.

Lithium Battery Explorer

Find candidate materials for lithium batteries. Get voltage profiles and oxygen evolution data.

Crystal Toolkit

Convert between CIF and VASP input files. Generate new crystals by substituting or removing species.

Phase Diagram App

Computational phase diagrams for closed and open systems. Find stable phases and study reaction pathways.

Reaction Calculator

Calculate the enthalpy of tens of thousands of reactions and compare with experimental values.

Structure Predictor

Predict new compounds using data-mined substitution algorithms.

Press Highlights

The New York Times

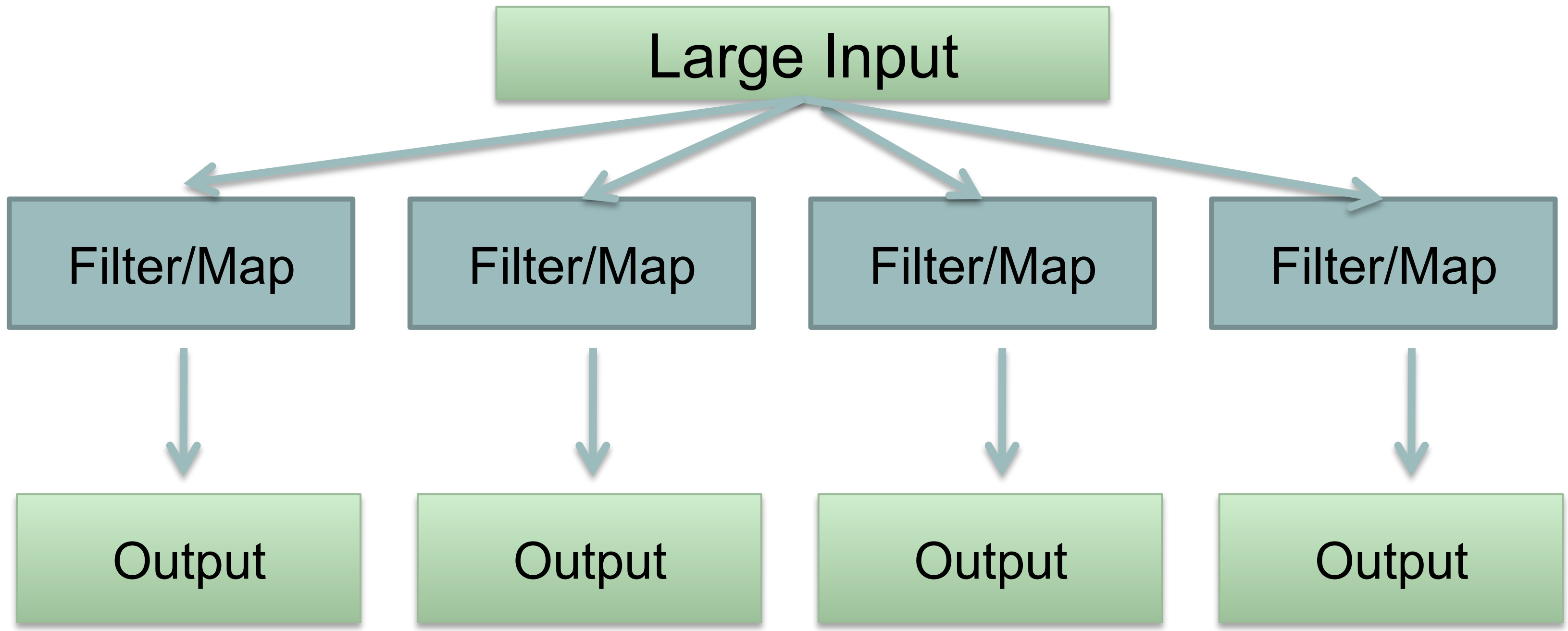
Beyond Fossil Fuels: Finding New Ways to Fill the Tank

Latest News

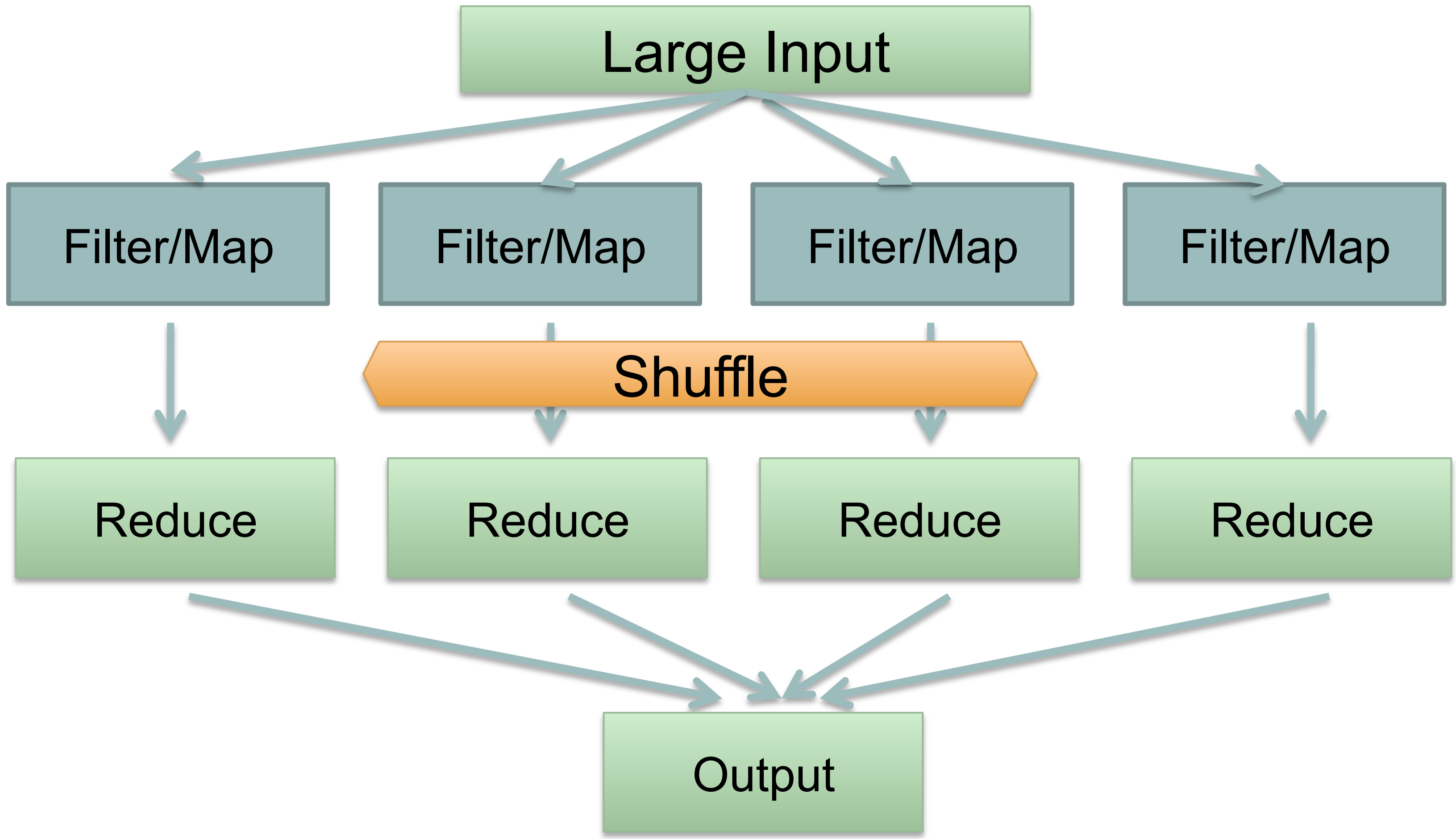
Common Themes

- **Throughput Oriented / Embarrassingly parallel**
- **Rapidly Increasing demand for computation (outpacing Moore's Law)**
- **Often Data Intensive**
- **Scaling from desktop or mid-range scales**

Map/Array Job

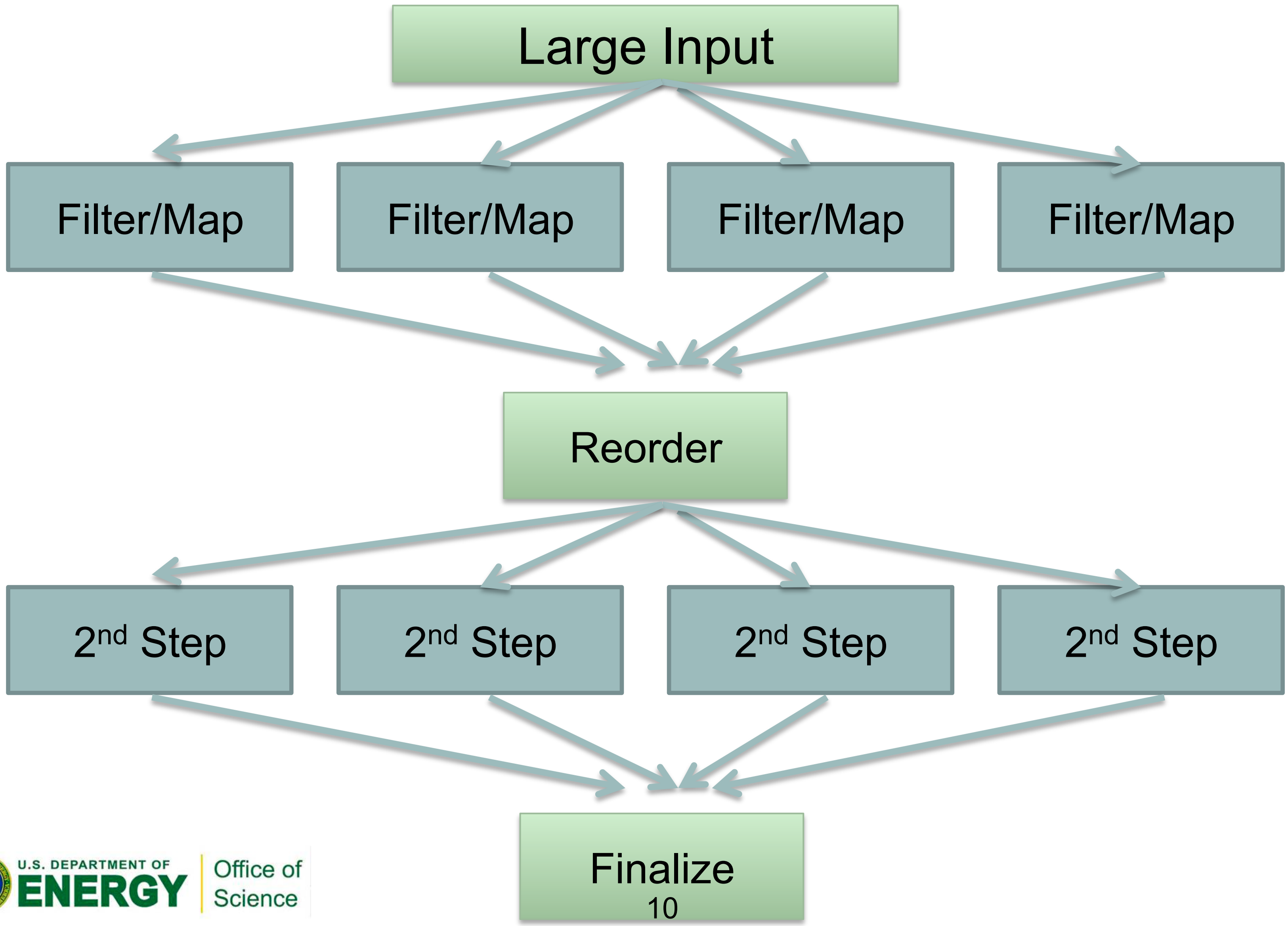


Map/Reduce



Large Input
Filter/Map
Filter/Map
Filter/Map
Filter/Map
Shuffle
Reduce
Reduce
Reduce
Reduce
Output

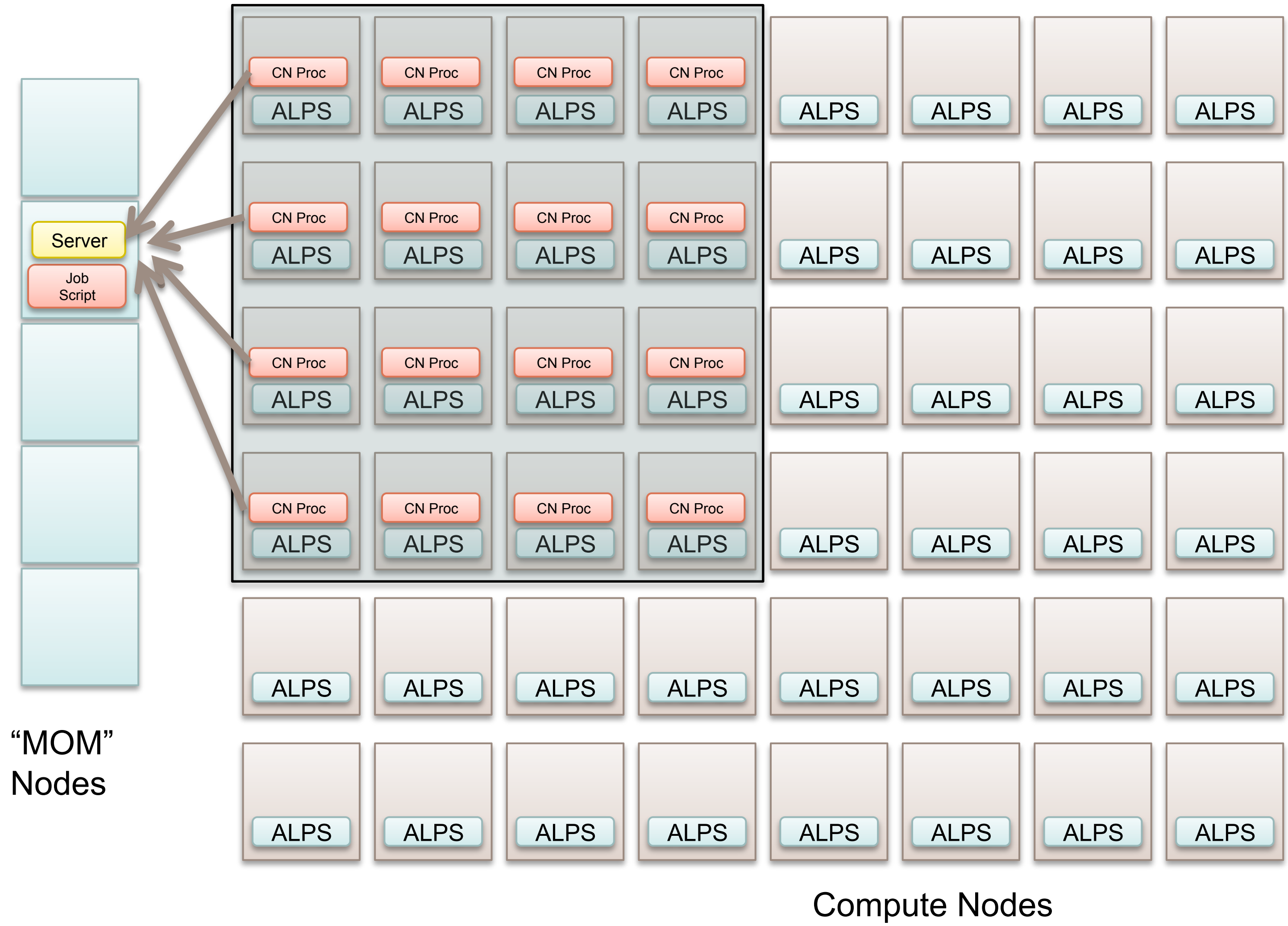
Complex Workflows



Approaches

- Private/User Allocation
  - Task Farmer
  - MySGE
  - MyHadoop
- Shared
  - CCM/Torque

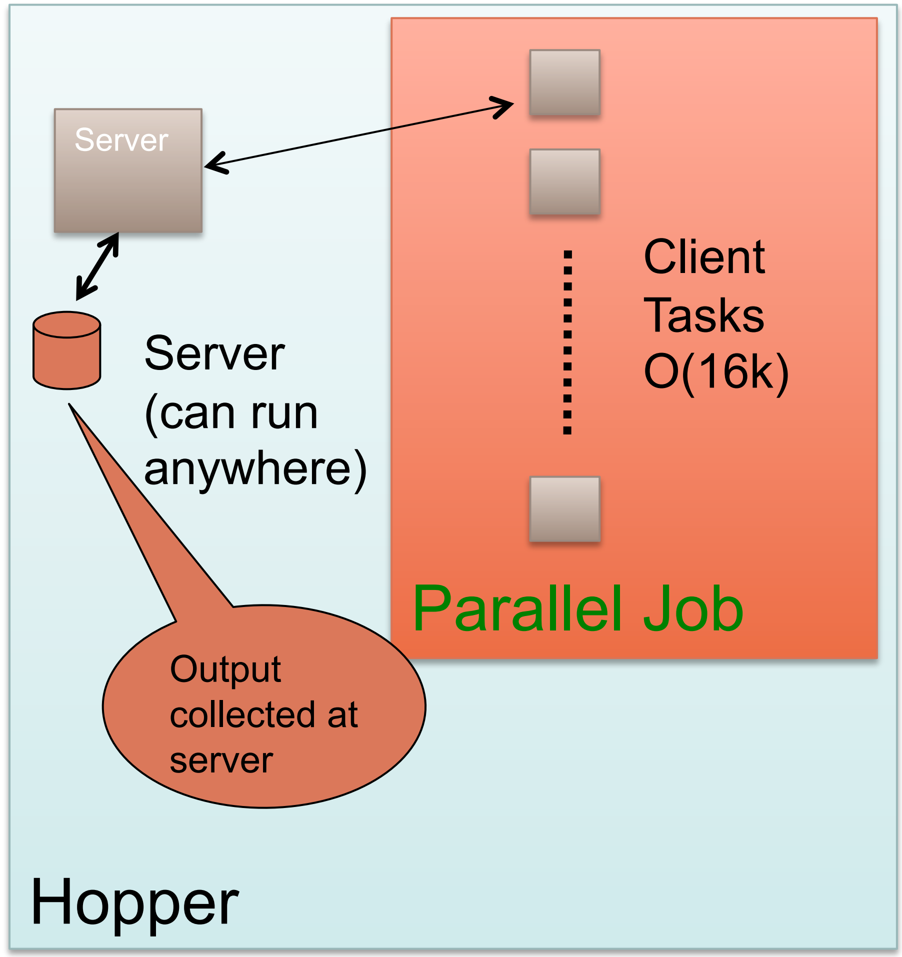
Private Allocation



TaskFarmer

Server
- Portable
- Reads in query genes
- Tracks progress and re-runs failed tasks
- Maintains checkpoint
- Collects output from clients

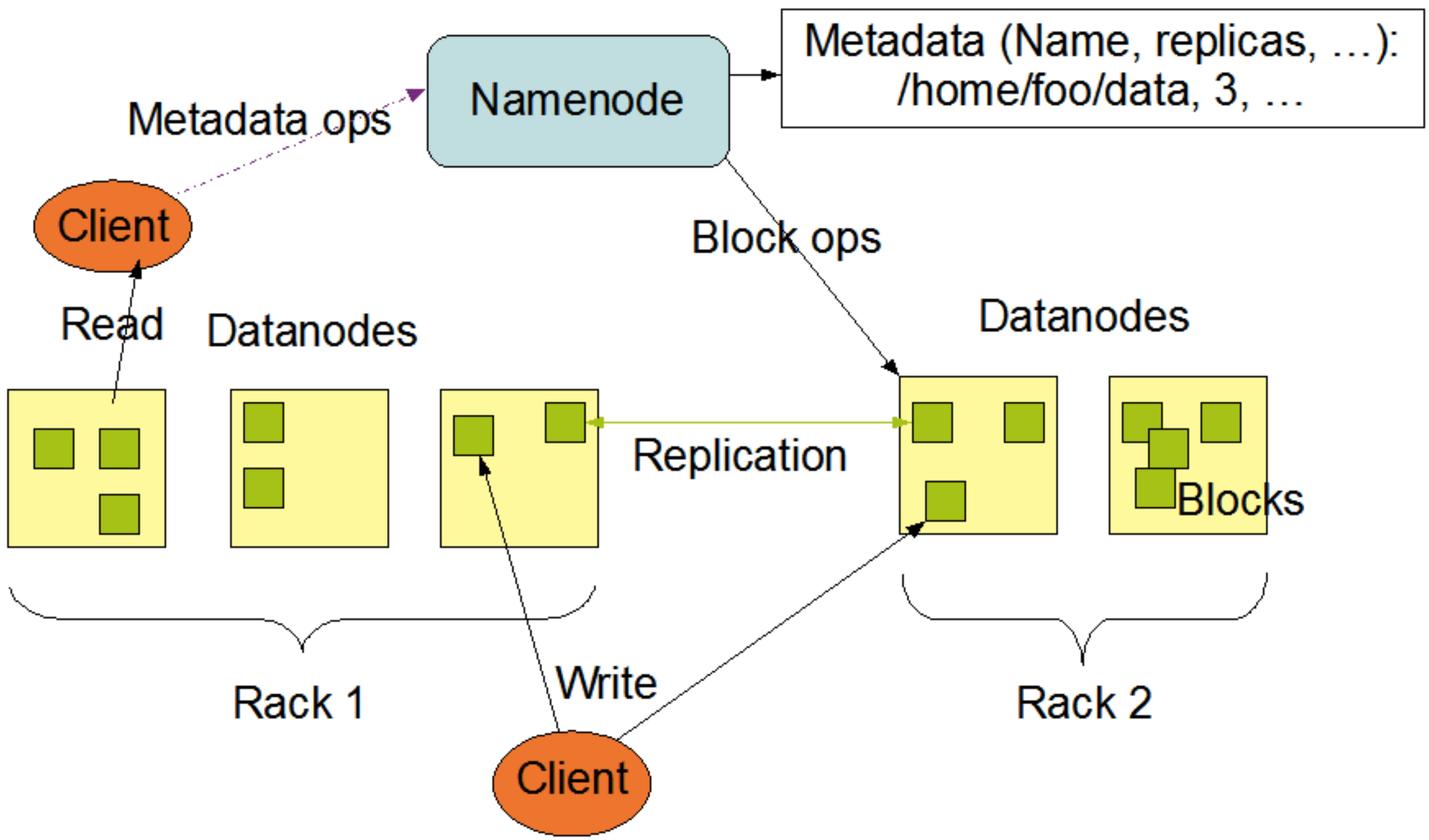
Client
- Can run any executable or script
- Gathers command line arguments from server
- Fetches input from server and pushes back results



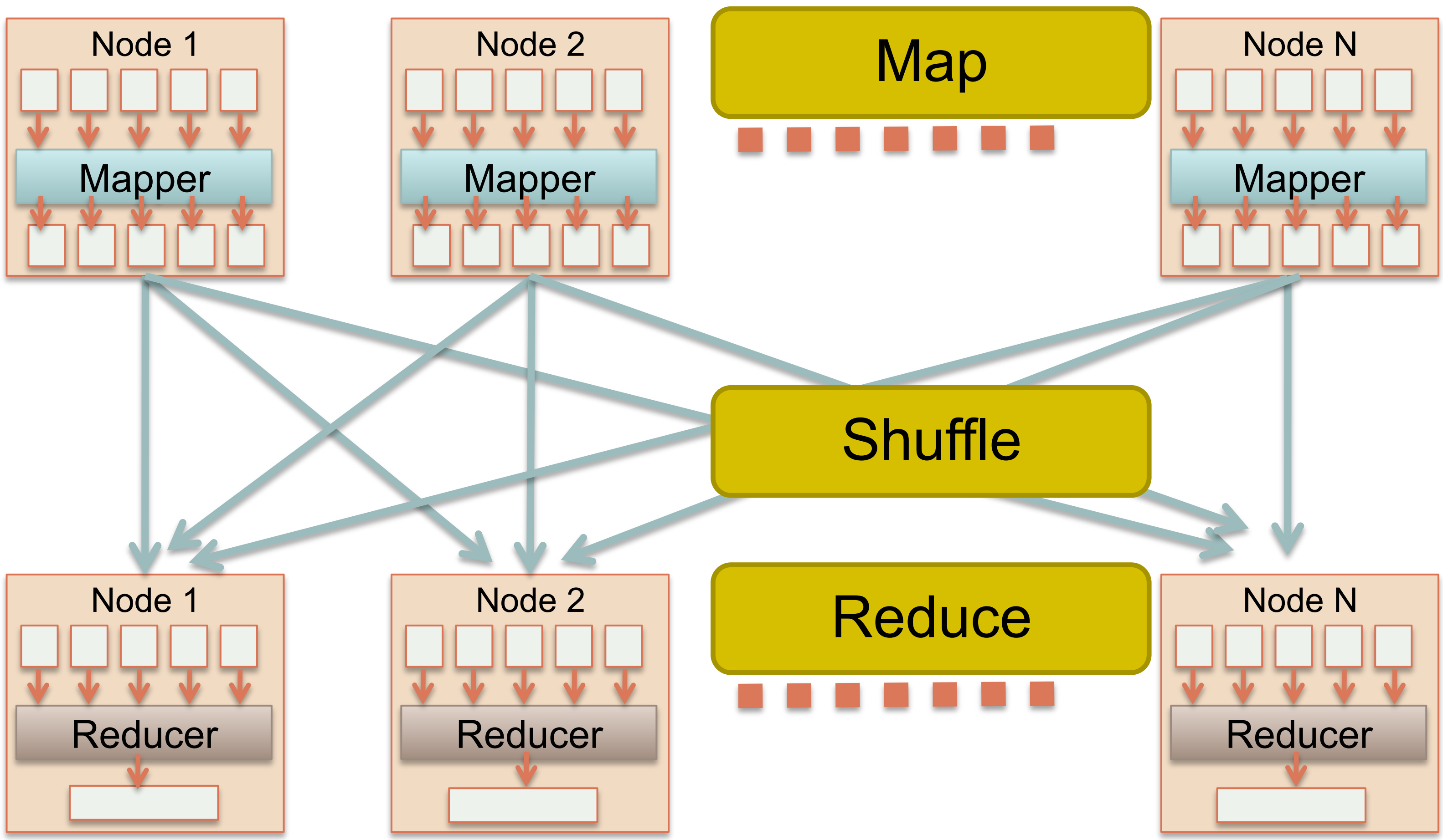
Strengths of MapReduce and Hadoop

- Fault Tolerance Model
- Data Locality
- Simple Programming Model
- Hides Complexity
- Domain Specific Extensions
- Strong Community

HDFS Architecture



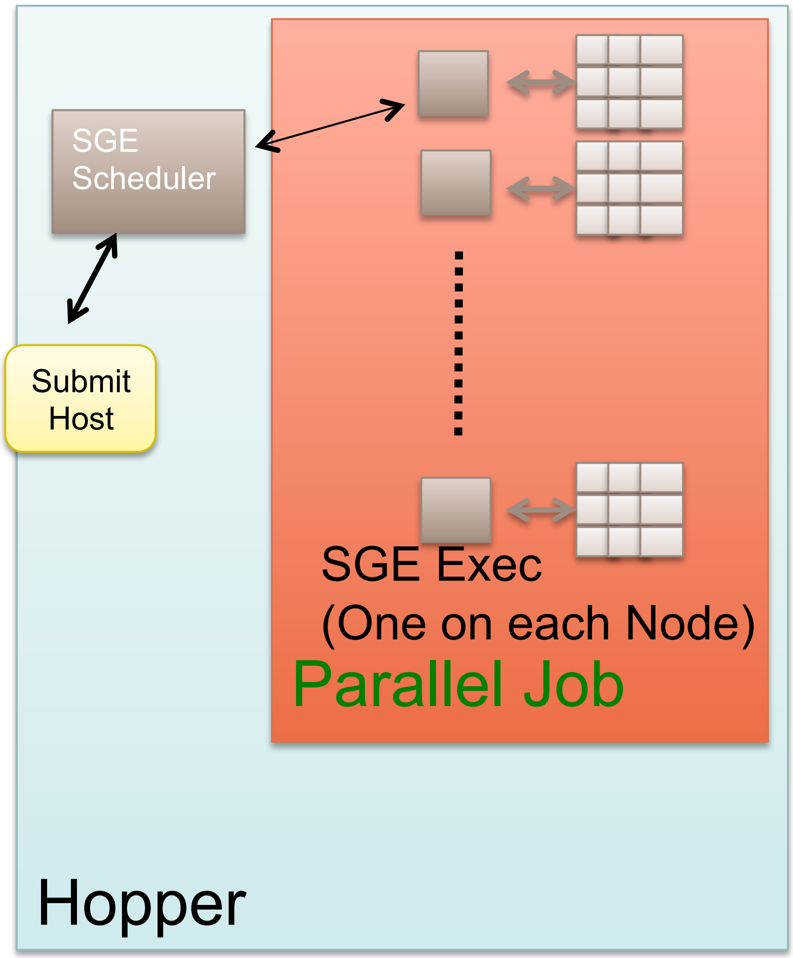
Data Flow in MapReduce



U.S. DEPARTMENT OF ENERGY | Office of Science

MySGE

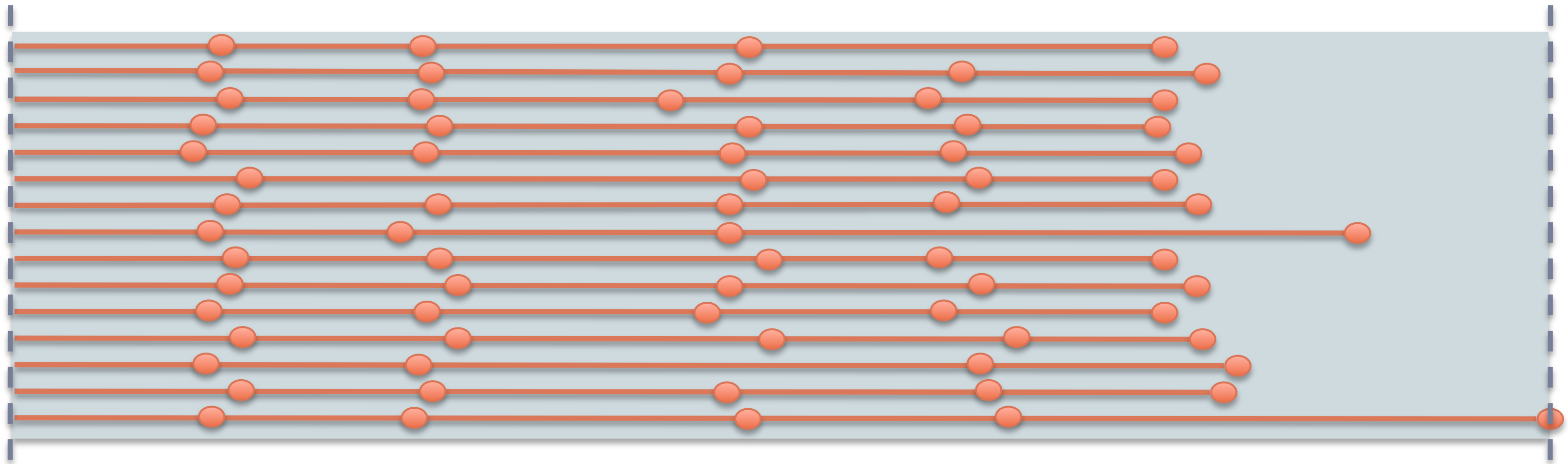
- User submits a single parallel job
- Personnel SGE scheduler is started
- User can submit jobs to SGE without modifications
- User still needs to think about scaling issues



Common Challenges and Solutions

- **Name Services (i.e. passwd/ldap)**
  - PRELOAD library to trap getpw* calls
- **Gathering Hostnames**
  - (Could probably get this from ALPS)
  - Use aprun to gather nids and generate host list
- **Master service runs on “mom” node**

Downside to Private Approach



- Load imbalance can lead to wasted cycles and additional charging
- Other users can't take advantage of idle cores

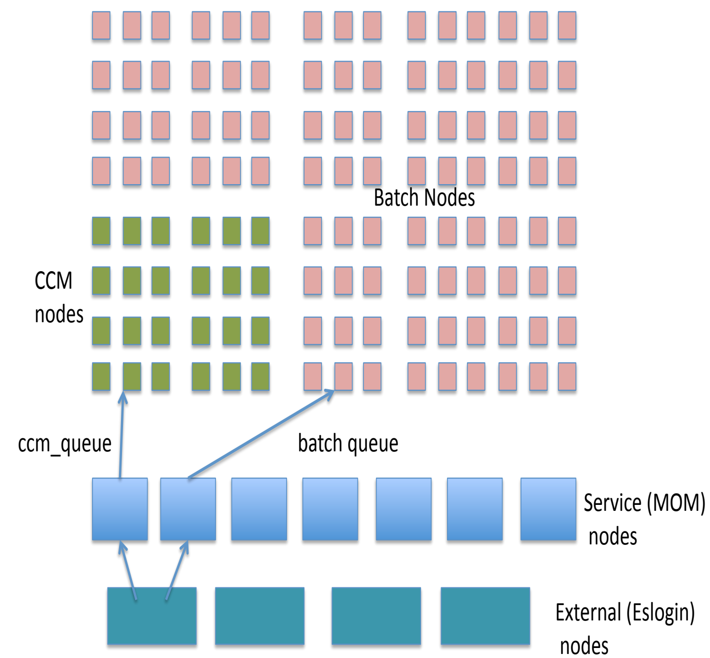
U.S. DEPARTMENT OF ENERGY | Office of Science

Running a shared-node Serial workload on the XE-6 using CCM

U.S. DEPARTMENT OF ENERGY | Office of Science

Using CCM to run a shared-node serial workload

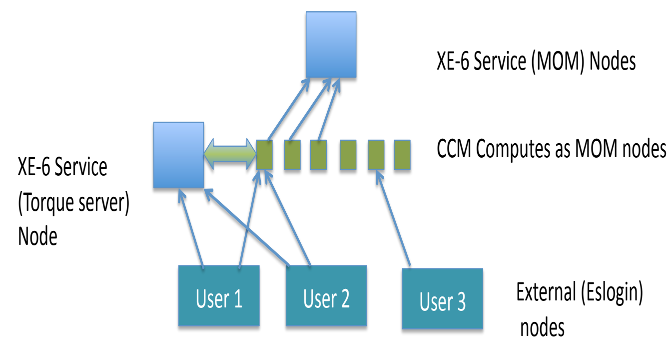
- **CCM can be used to "convert" XE-6 (MPP) compute nodes into standard "cluster-like" nodes with a regular Linux environment.**
- **To run a serial workload on these "CCM nodes" requires they be accessible as regular cluster nodes to the batch system**
  - **This cannot be done using the regular batch system**
  - **This requires starting up a separate batch system instance**
  - **Done using a special CCM "job" which starts up the server and client daemons – the server is started up on the standard XE-6 MOM nodes, and the clients are on the XE-6 CCM compute nodes**



U.S. DEPARTMENT OF ENERGY | Office of Science

Mechanics of running a shared-node serial workload

- **“Special” user submits a job to the ccm_queue, asking for as many nodes as required to handle a serial workload (subject to CCM limits), and for the maximum time allowed.**
- **“Special job” starts up pbs_server on XE-6 MOM node with alternate ports**
- **Job then runs pbs_mom on allocated CCM compute nodes (under alternate ports)**
- **Job starts up scheduler (Maui or pbs_sched) which communicates with the alternate resource manager (RM)**
- **At this point, other users (user1, user2, etc) can submit jobs to the CCM compute nodes (which have now been essentially repurposed as a separate cluster supporting a serial workload)**

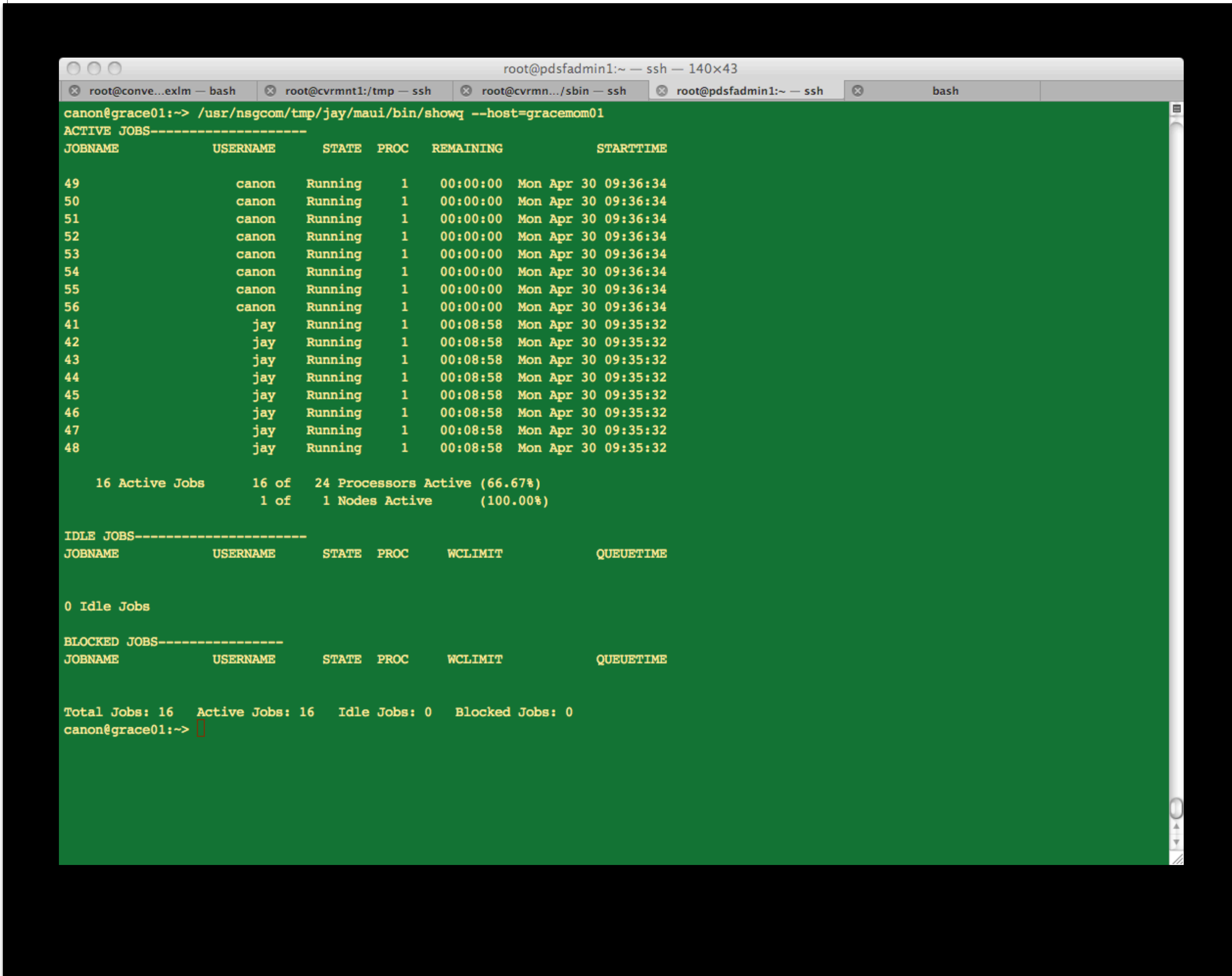


root@pdsfadmin1:~ — ssh — 140×43

root@conve...exlm — bash | root@cvrmnt1:/tmp — ssh | root@cvrmn.../sbin — ssh | root@pdsfadmin1:~ — ssh | bash

```
grace01 j/jay> /usr/nsgcom/tmp/jay/torque/bin/qstat -n @gracemom01:35000

nid00002:35000:
                                                                   Req'd  Req'd   Elap
Job ID               Username Queue    Jobname          SessID NDS   TSK Memory Time  S Time
-------------------- -------- -------- ---------------- ------ ----- --- ------ ----- - -----
41.nid00002          jay      serial   tst.job           22129   --    1    --  00:10 R 00:04
   nid00008/0
42.nid00002          jay      serial   tst.job           22132   --    1    --  00:10 R 00:04
   nid00008/1
43.nid00002          jay      serial   tst.job           22135   --    1    --  00:10 R 00:04
   nid00008/2
44.nid00002          jay      serial   tst.job           22138   --    1    --  00:10 R 00:04
   nid00008/3
45.nid00002          jay      serial   tst.job           22153   --    1    --  00:10 R 00:04
   nid00008/4
46.nid00002          jay      serial   tst.job           22199   --    1    --  00:10 R 00:03
   nid00008/5
47.nid00002          jay      serial   tst.job           22233   --    1    --  00:10 R 00:03
   nid00008/6
48.nid00002          jay      serial   tst.job           22284   --    1    --  00:10 R 00:03
   nid00008/7
57.nid00002          jay      serial   tst.job           26161   --    1    --  00:10 R   --
   nid00008/16
58.nid00002          jay      serial   tst.job           26166   --    1    --  00:10 R   --
   nid00008/17
59.nid00002          jay      serial   tst.job           26186   --    1    --  00:10 R   --
   nid00008/18
60.nid00002          jay      serial   tst.job           26198   --    1    --  00:10 R   --
   nid00008/19
61.nid00002          jay      serial   tst.job           26239   --    1    --  00:10 R   --
   nid00008/20
62.nid00002          jay      serial   tst.job           26288   --    1    --  00:10 R   --
   nid00008/21
63.nid00002          jay      serial   tst.job           26332   --    1    --  00:10 R   --
   nid00008/22
64.nid00002          jay      serial   tst.job           26394   --    1    --  00:10 R   --
   nid00008/23
grace01 j/jay>
```



```
root@pdsfadmin1:~ — ssh — 140×43
root@conve...exlm — bash | root@cvrmnt1:/tmp — ssh | root@cvrmn.../sbin — ssh | root@pdsfadmin1:~ — ssh | bash

canon@grace01:~> /usr/nsgcom/tmp/jay/maui/bin/showq --host=gracemom01
ACTIVE JOBS--------------------
JOBNAME            USERNAME      STATE  PROC   REMAINING            STARTTIME

49                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
50                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
51                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
52                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
53                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
54                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
55                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
56                    canon    Running     1    00:00:00  Mon Apr 30 09:36:34
41                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
42                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
43                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
44                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
45                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
46                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
47                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32
48                      jay    Running     1    00:08:58  Mon Apr 30 09:35:32

    16 Active Jobs      16 of   24 Processors Active (66.67%)
                         1 of    1 Nodes Active      (100.00%)

IDLE JOBS----------------------
JOBNAME            USERNAME      STATE  PROC     WCLIMIT            QUEUETIME


0 Idle Jobs

BLOCKED JOBS----------------
JOBNAME            USERNAME      STATE  PROC     WCLIMIT            QUEUETIME


Total Jobs: 16   Active Jobs: 16   Idle Jobs: 0   Blocked Jobs: 0
canon@grace01:~>
```

Limitations

- **Current approach uses a static assignment of nodes.**
  - **Initial request for CCM nodes needs cannot be changed on the fly, but multiple requests can be made**
- **CCM communication occurs over TCP/IP, so the high-performance network is not available. (Can't share uGNI)**
- **Zhengi Zhao/Helen He's presentation on CCM for other uses of CCM and some of the limitations**

U.S. DEPARTMENT OF ENERGY | Office of Science

Future Work

- **Continue to Improve CCM/Torque Approach**
  - Finish testing and phase into production
  - Dynamically resize serial partition
- **Improve Hadoop Implementation**
  - Optimize shuffle phase for high-bandwidth network

Suggested Cray Optimizations

- Local storage (SSD)
  - Many applications and frameworks rely on local storage
  - Useful for Data Intensive Apps
- Improvements to CCM/DVS
  - Tools to facilitate running Python/Perl at scale
  - Tools to help caching data at scale
- Improvements to ALPS
- Better ways to cleanup after a job

Closing Thoughts

- Increasing demand to support new workloads
  - Driven by improving instruments
  - New classes of modeling and simulation
- NERSC has developed four approaches to supporting new workloads
- The Cray platform is surprisingly flexible:
  - x86/Linux underpinnings help
  - CCM and other extensions have further simplified matters

U.S. DEPARTMENT OF ENERGY | Office of Science

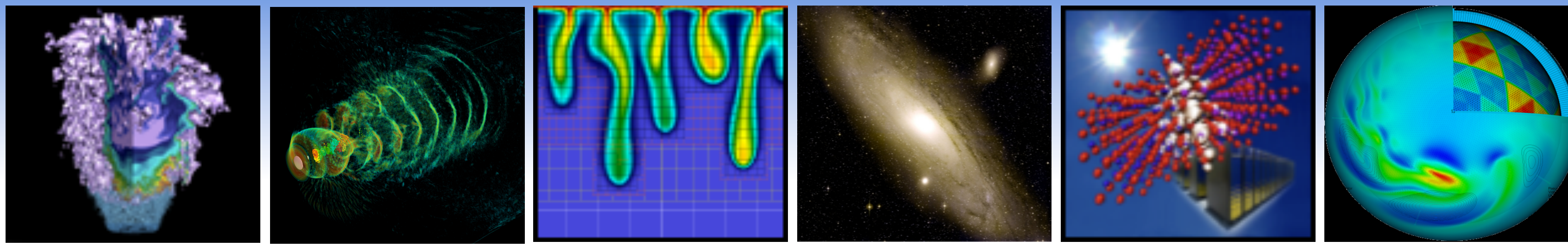
Acknowledgements

Lavanya Ramakrishnan

Jay Srinivasan

This work was supported by the Director, Office of Science, Office of Advanced Scientific Computing Research of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231.

U.S. DEPARTMENT OF ENERGY | Office of Science



Thank you!

Contact Info:
Shane Canon
SCanon@lbl.gov