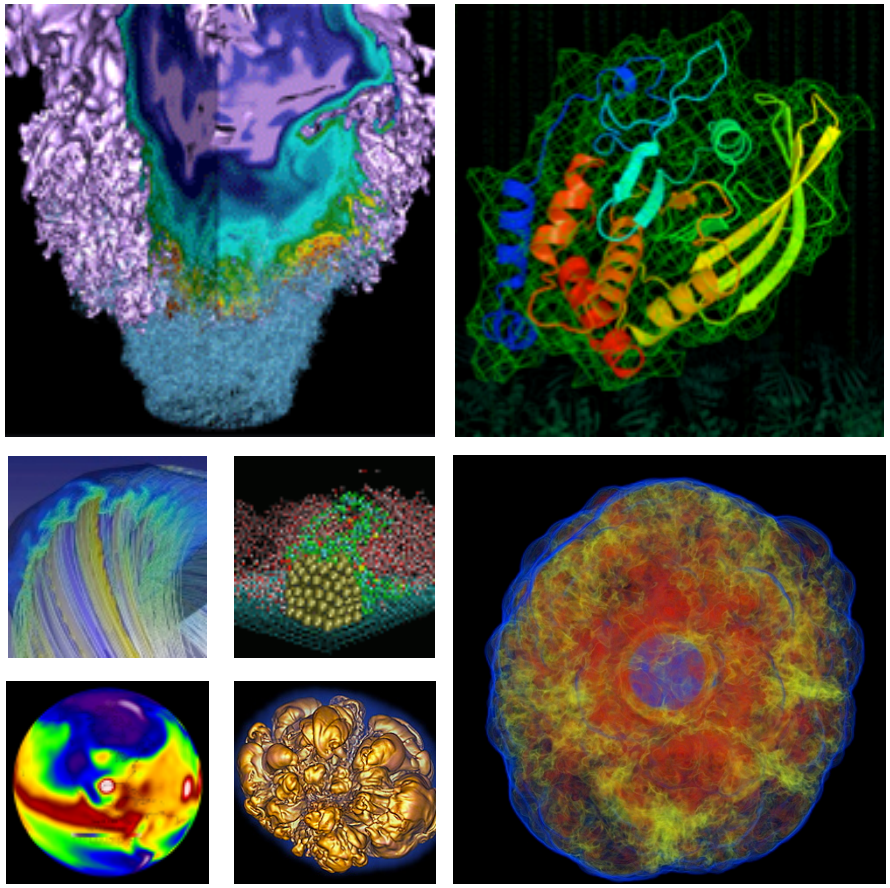
NERSC and HTC



Shane Canon, David Skinner and Jay Srinivasan

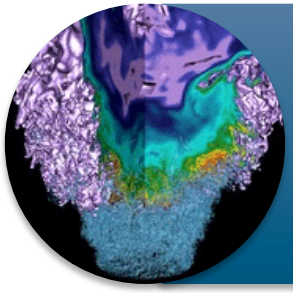
NUG2013

February 12, 2013

U.S. DEPARTMENT OF ENERGY | Office of Science

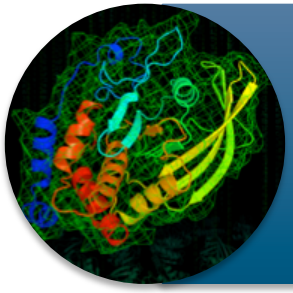
BERKELEY LAB

Science Strategies @ NERSC



Science at Scale

Petascale to Exascale



Science through Volume

Thousands to Millions of Simulations

Science in Data

Petabytes to Exabytes

MATERIALS PROJECT

A Materials Genome Approach

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

Enter formulas

e.g., Fe2O3 Fe3O4

Search

Database Statistics

19120 materials

3050 bandstructures

214 intercalation batteries

4158 conversion batteries

Materials Explorer

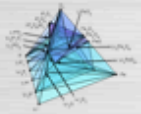
Search for materials information by chemistry, composition, or property.

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.

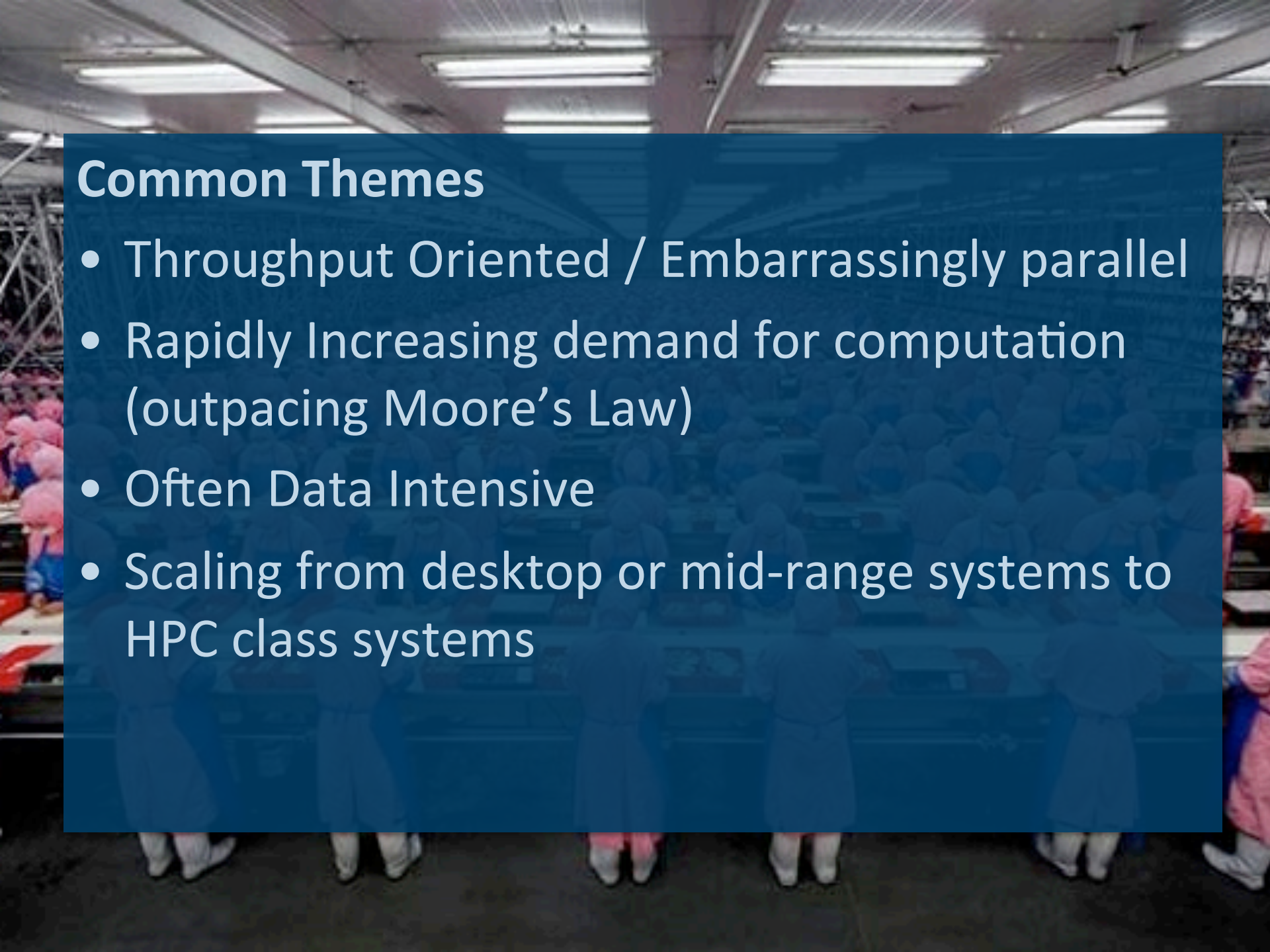
Structure Predictor

Predict new compounds using data-mined substitution algorithms.

Press Highlights

The New York Times

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 systems to HPC class systems

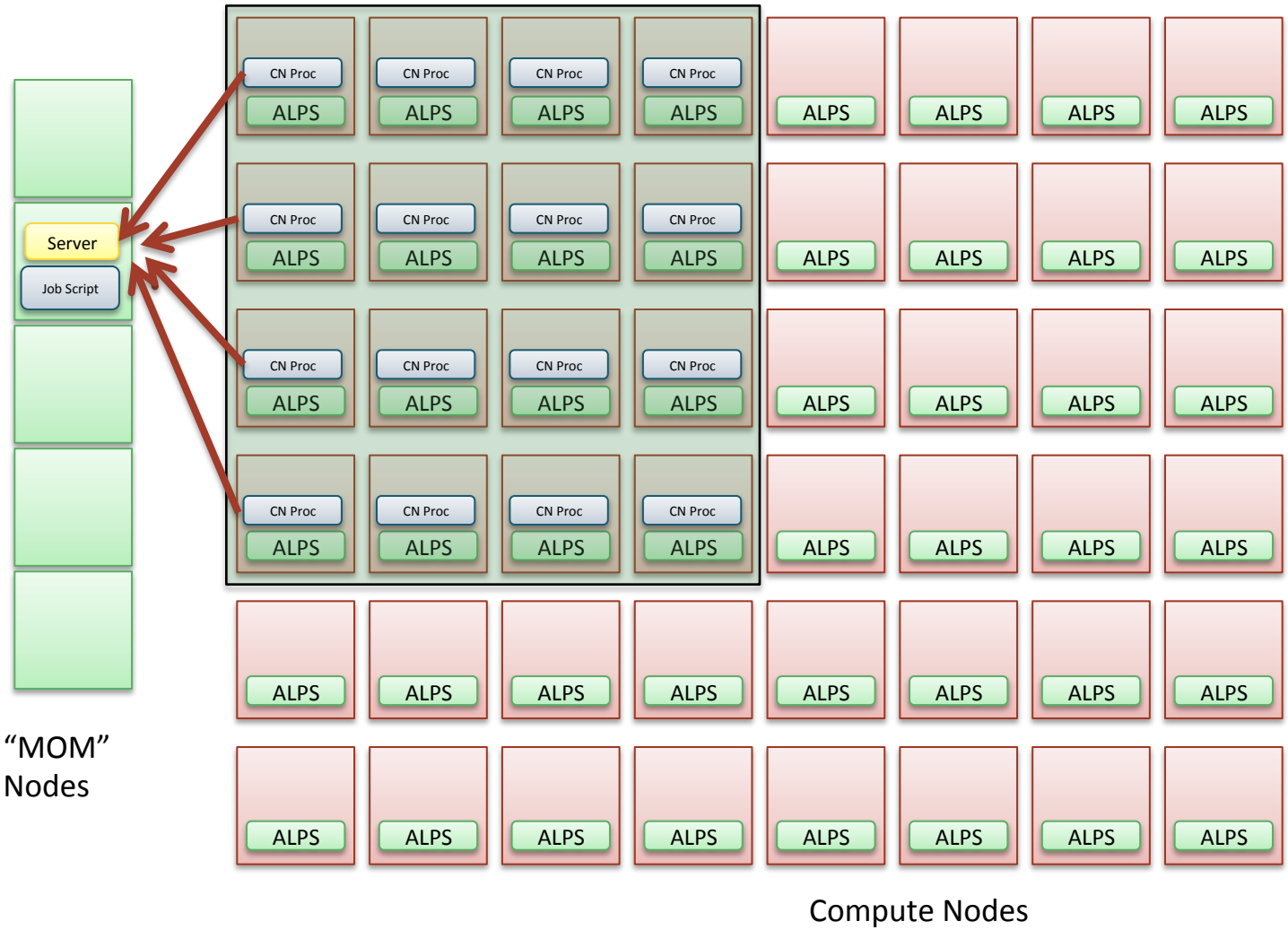
Approaches

- **Throughput Queues**
- **Private/User Allocation**
  - Task Farmer (NERSC Developed or Cray Provided)
  - MyHadoop
  - MySGE
- **Shared**
  - CCM/Torque
- **Hybrid?**
  - High-Throughput Queue Systems

Throughput Queues

- **Serial Queue on Carver**
  - 150 running
  - 20 eligible
  - Best for serial jobs needing full Linux stack
- **Throughput Queue on Hopper**
  - 250 running
  - 500 eligible
  - Best for high-throughput, small concurrency jobs

Private Allocation



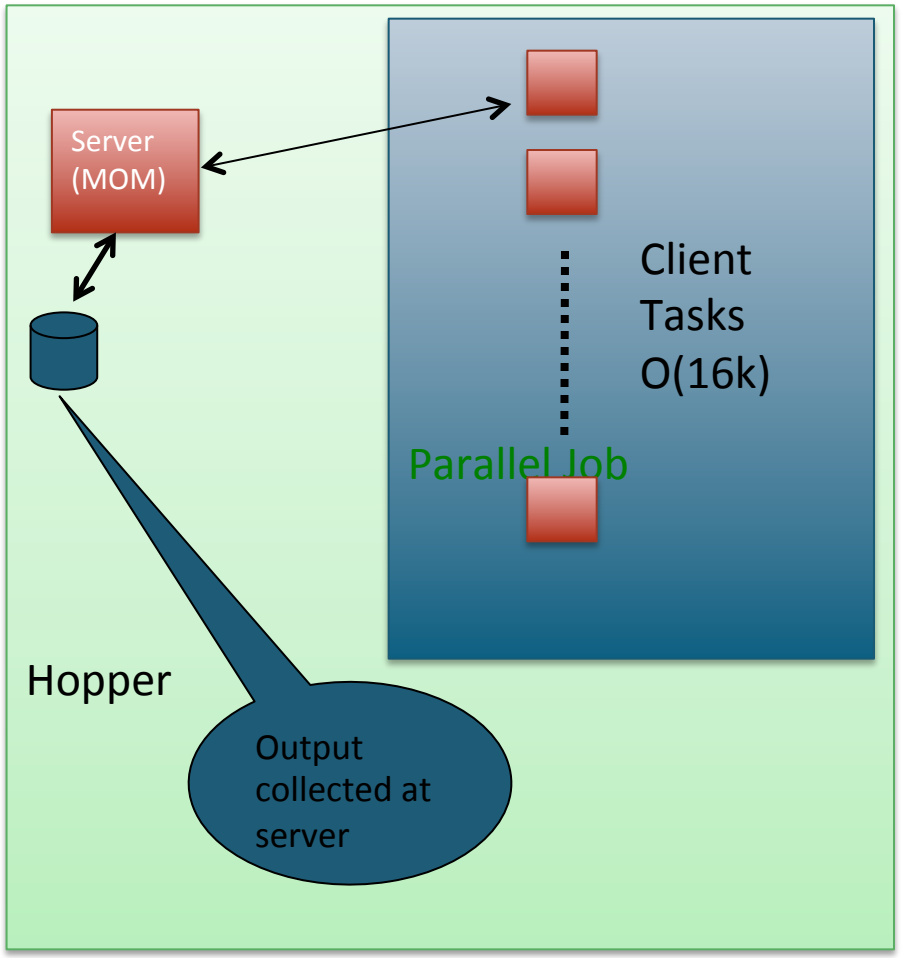
TaskFarmer

Server

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

Client

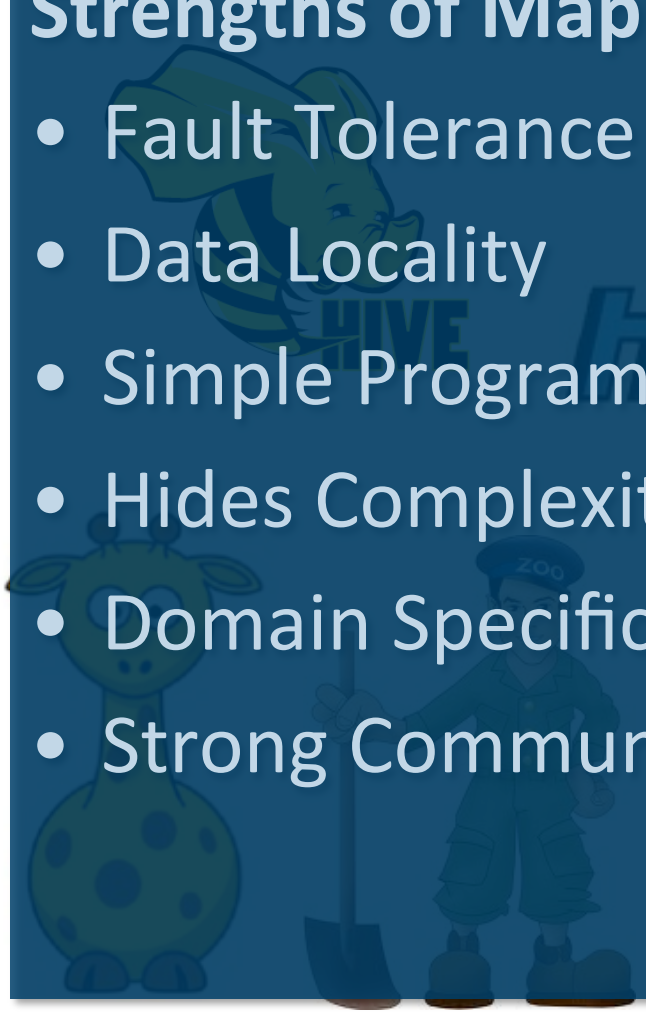
- Can run any executable or script
- Gathers command line arguments from server
- Fetches input from server and pushes back results

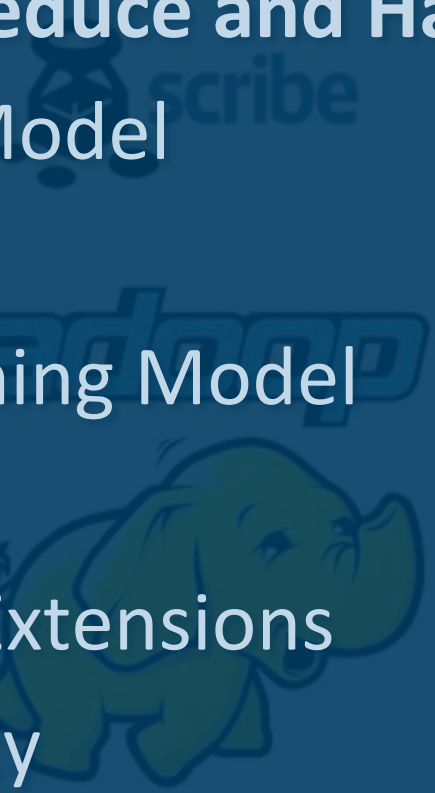


Hadoop/MapReduce

Strengths of MapReduce and Hadoop

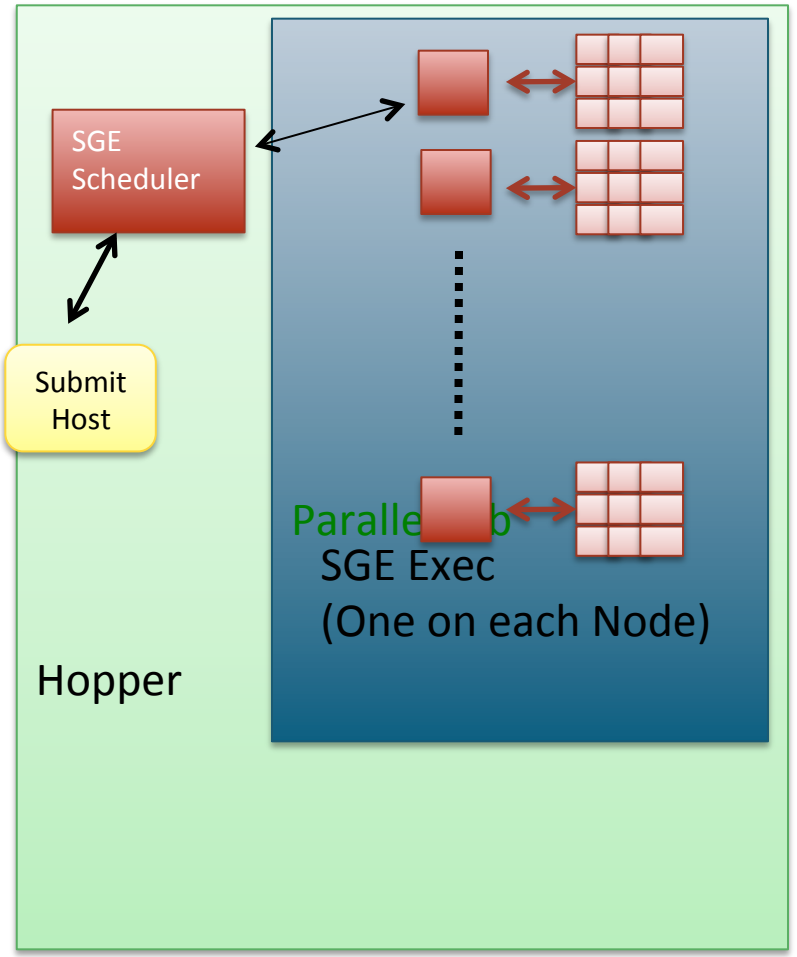
- Fault Tolerance Model
- Data Locality
- Simple Programming Model
- Hides Complexity
- Domain Specific Extensions
- Strong Community



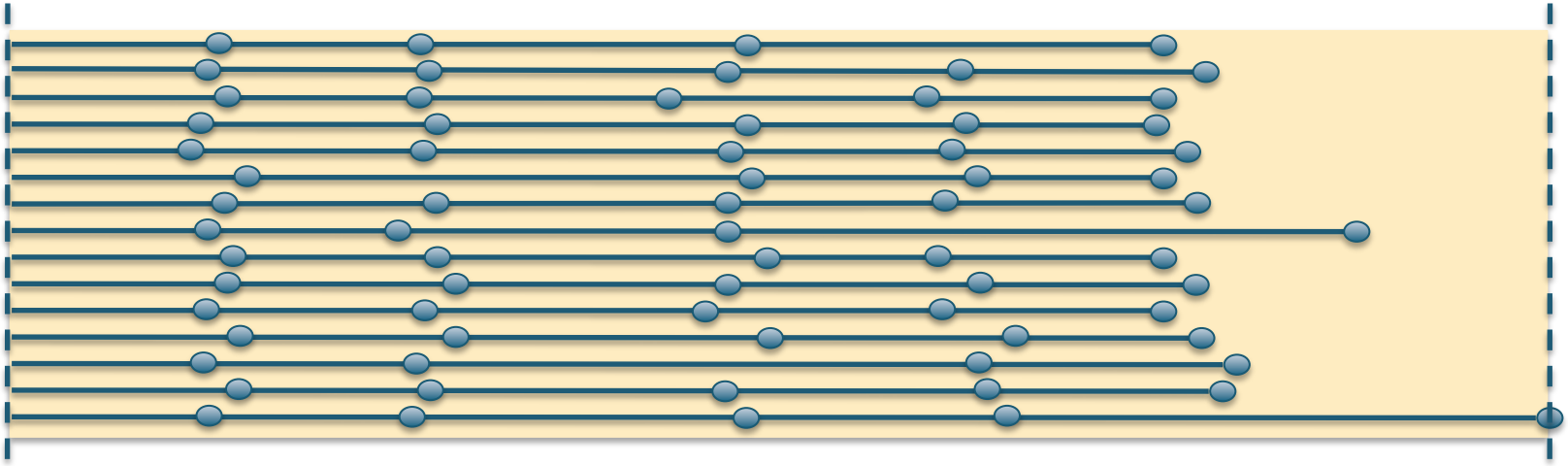


MySGE

- 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



Downside to Private Approach

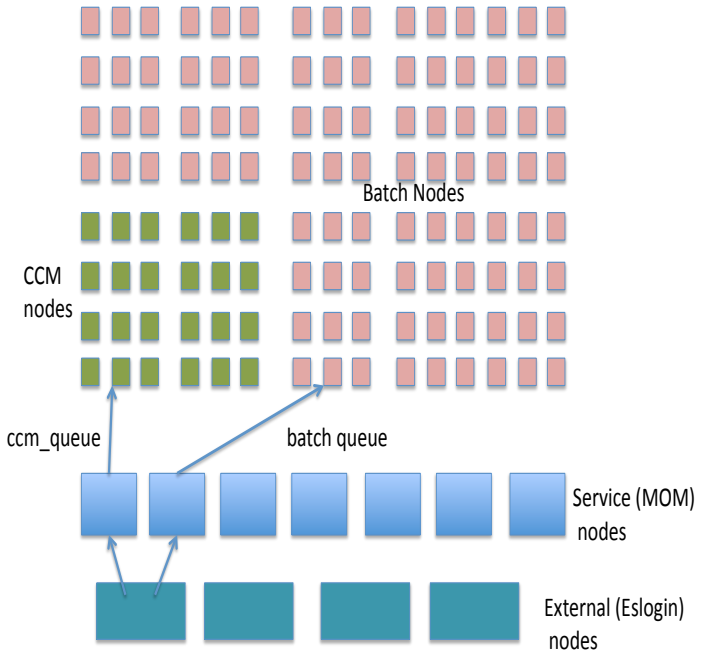


- Load imbalance can lead to wasted resources and additional charging
- Other users can't take advantage of idle cores

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

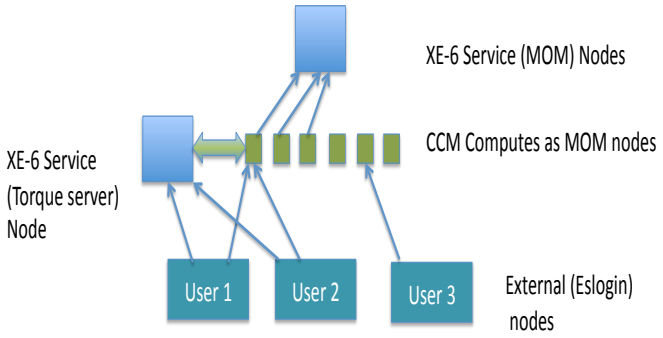
Using CCM to run a shared-node serial workload

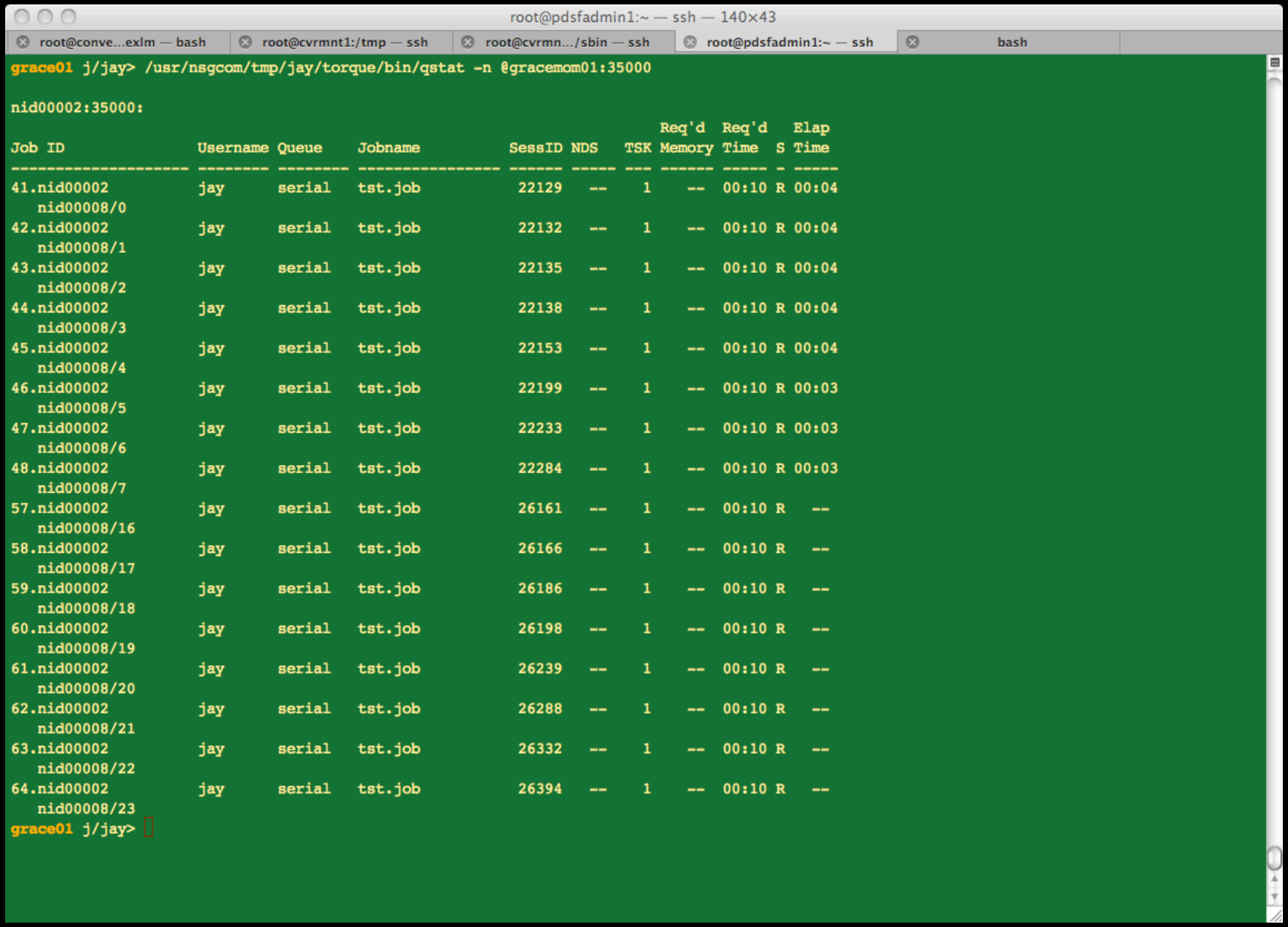
- **CCM can be used to "convert" XE-6 (MPP) compute nodes into standard "cluster-like" nodes with a regular Linux environment.**
- **To run a serial workload on these "CCM nodes" requires they be accessible as regular cluster nodes to the batch system**
  - This cannot be done using the regular batch 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

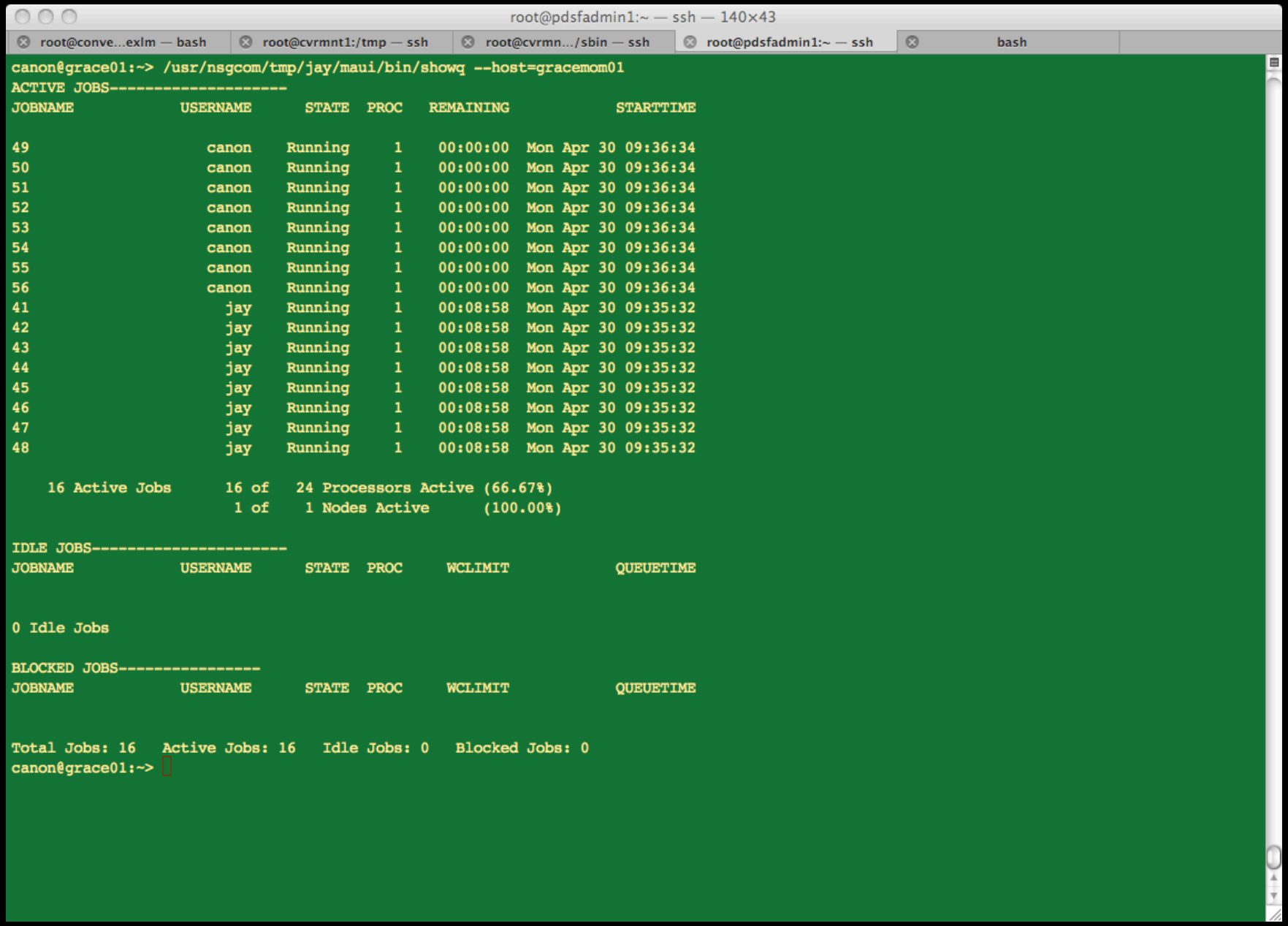


Mechanics of running a shared-node serial workload

- "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:
Job ID Username Queue Jobname SessID NDS TSK Req'd Memory Req'd Time S Elap 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)**

Other Challenges

- Policy and Fairness – Many of the challenges result from policies, not just technical
- I/O and Staging Common Files – Python or Perl libraries, large common references, etc
- Porting – Moving applications to Cray can be difficult
- Still not like a Cluster – No local disk, limited networking

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
- **Evaluating more fine-grained tasked based Schduler**
  - Use of external message queue (i.e. AMQP)

Closing Thoughts

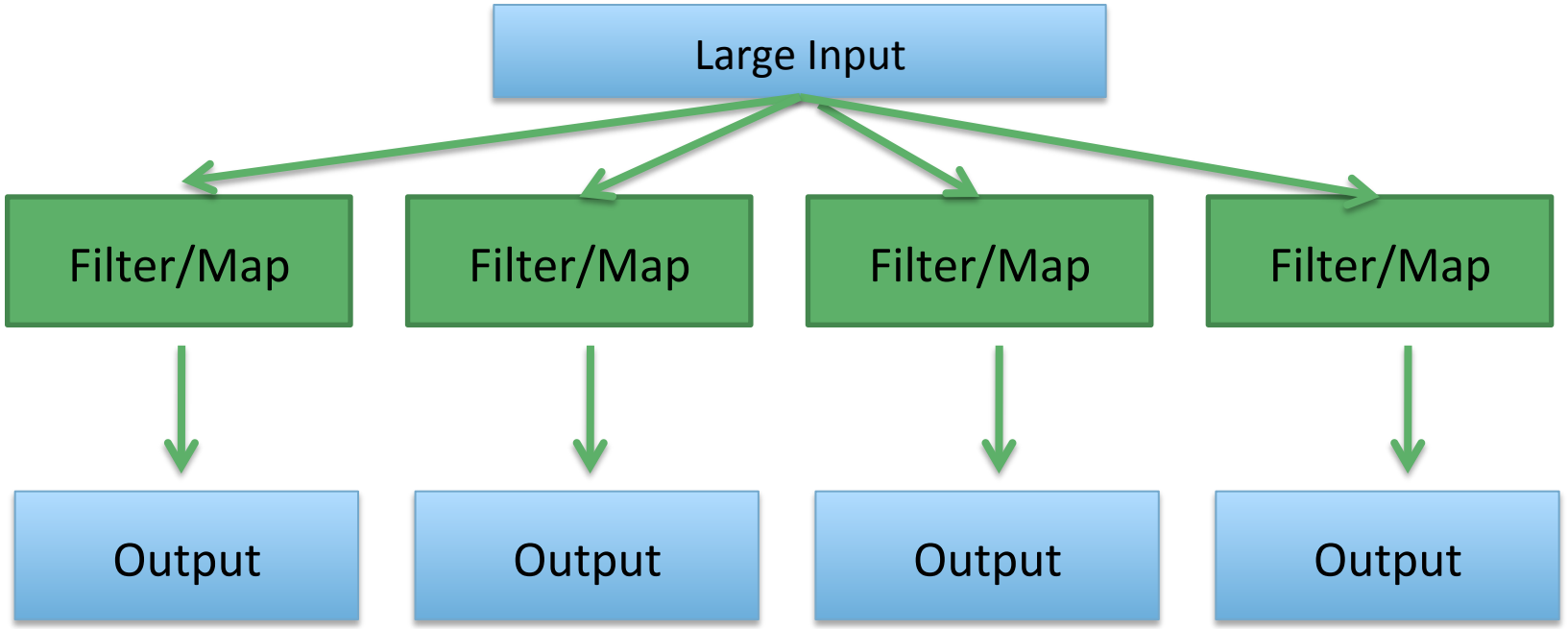
- **Increasing demand to support new workloads**
  - Driven by improving instruments
  - New classes of modeling and simulation
- **NERSC has developed four approaches to supporting new workloads and is exploring others**

National Energy Research Scientific Computing Center

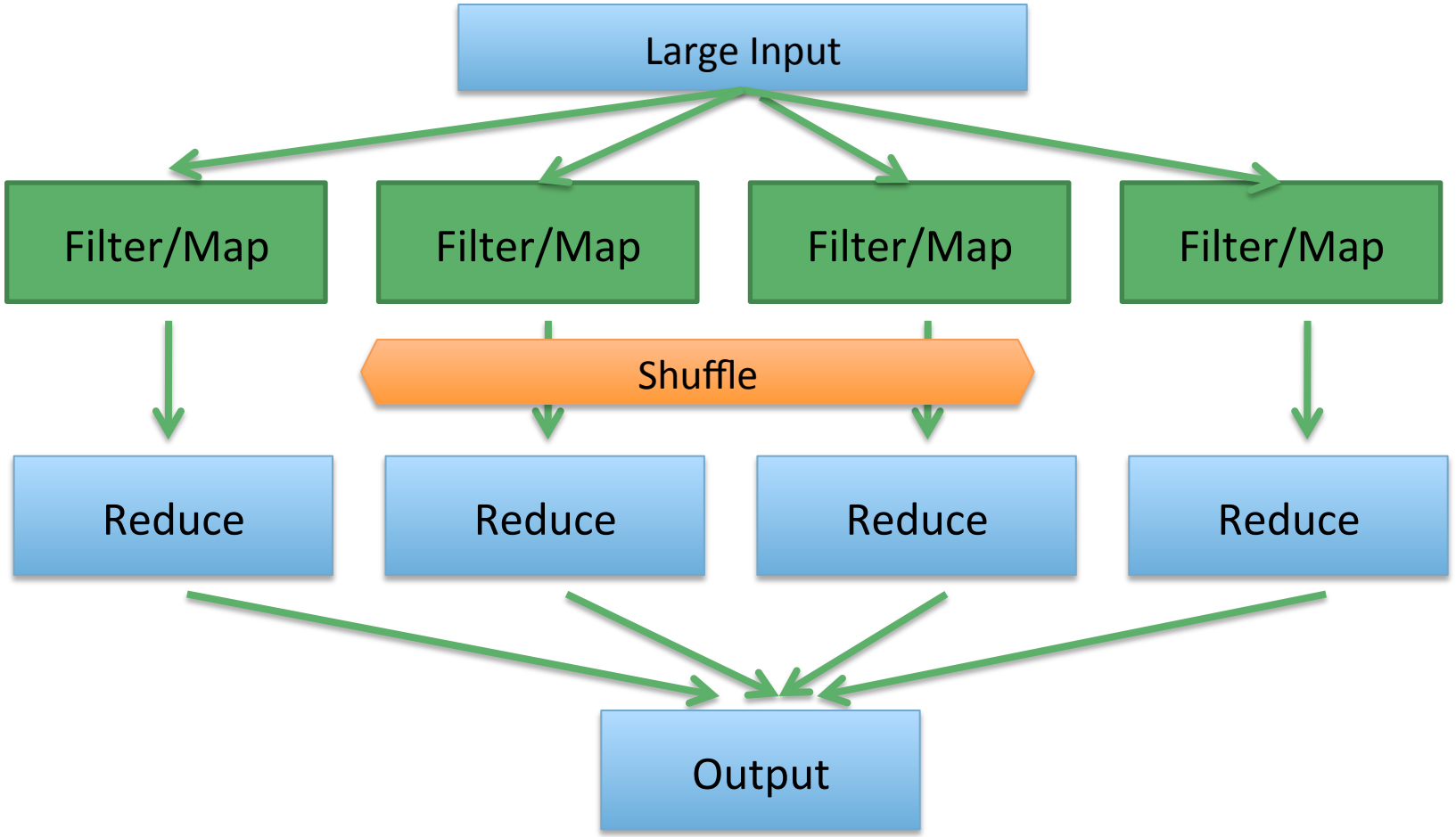
Why does NERSC Support This?

- Users need it
- Important science can be achieved
- Accelerate specific analysis
- Small fraction of a large system is significantly larger than available systems
- Even “Capability” jobs often have through-put oriented components (pre-computing, analysis)

Map/Array Job



Map/Reduce


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

Complex Workflows