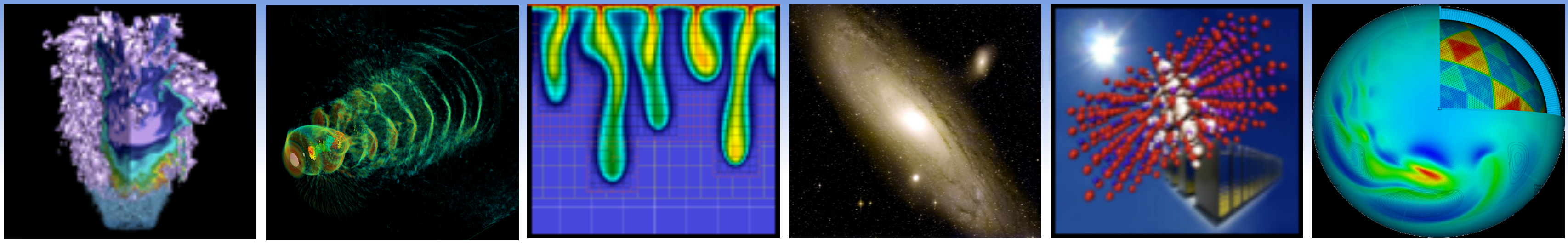


Exploiting NERSC systems for Bioinformatics

Shane Canon
Lawrence Berkeley Lab

JGI Training
May 2, 2011

The Challenges and some Work Arounds

- Scheduling policies don't facilitate through-put workloads
- No local disks
- Longer wait times
- Restricted environment on compute nodes
- Limited external network connectivity

- Use tools like MySGE and taskfarmer to batch up work
- Use Cacher if possible
- Target large workloads that block other work
- Use CRAY_ROOTFS=DSL
- Use a tcp relay on a front-end node

Why does NERSC not provide better support for serial jobs?

- DOE requires NERSC to meet certain metrics related to job sizes.
- Scheduling policy favor large jobs to insure they can get run
- Cray systems are optimized for running parallel jobs.

Running high-throughput workloads can still be done and systems like Hopper are big enough to make this worth while.

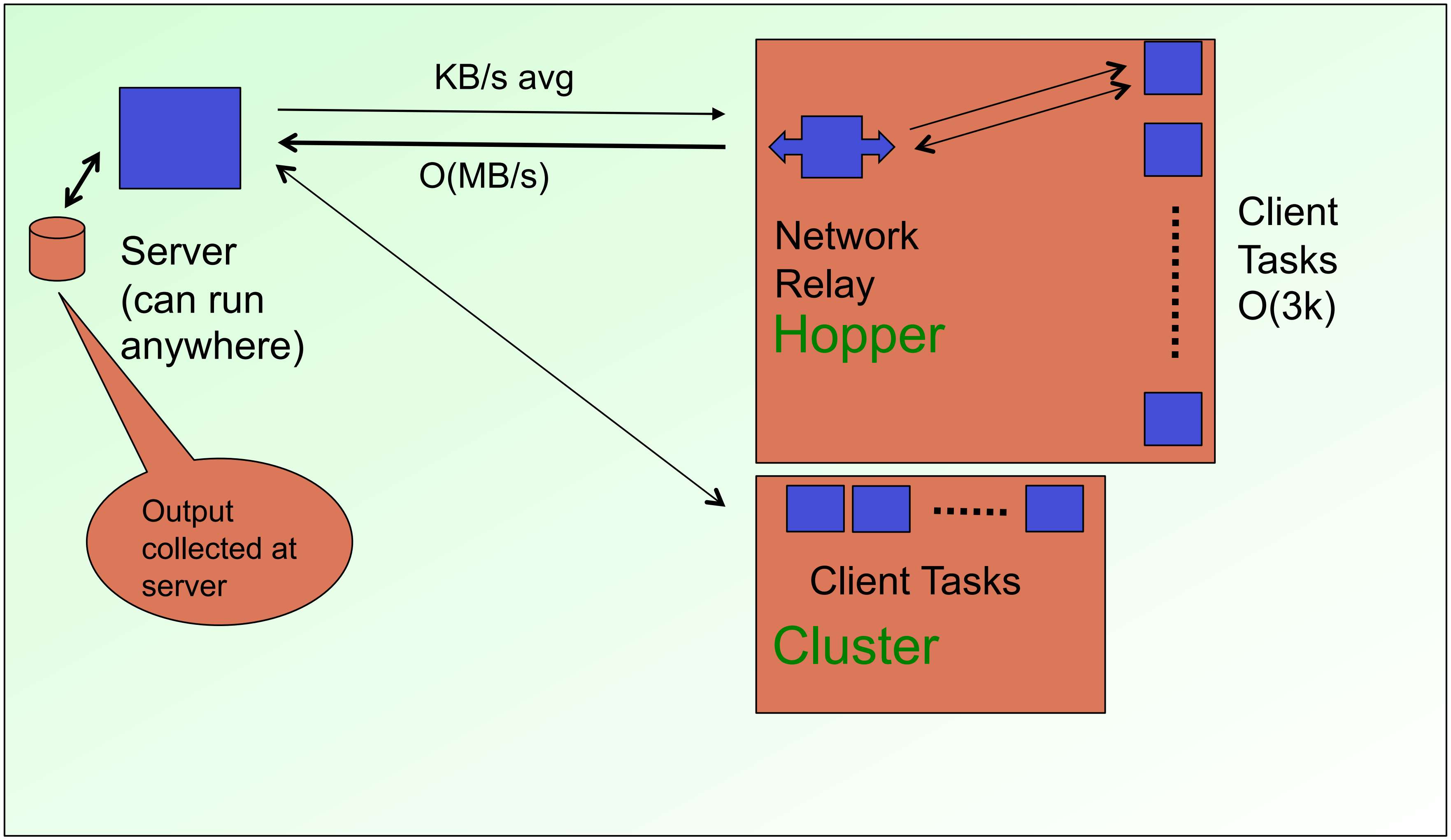
Resources and Tools

- Allocation: m342 4M hours
- Magellan Serial queue (-q mag_serial)
- Advanced reservations
- Task Farmer (blastalltf)
- MySGE
- Cacher
- Ported applications

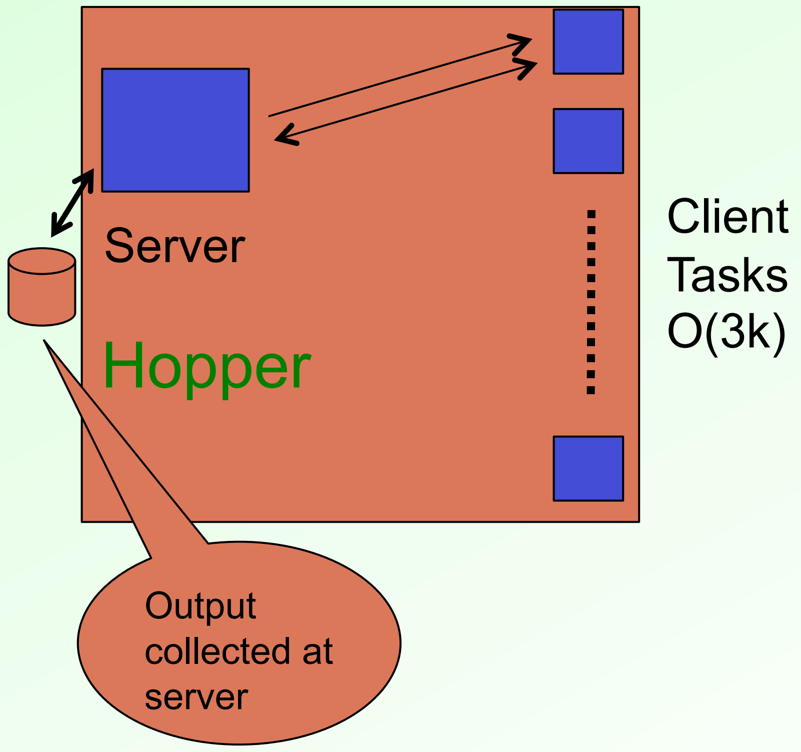
Task Farmer

- A framework for running serial codes against a large fasta file (assumes that order is not important)
- Can run with an arbitrary number of compute nodes.
- Will automatically re-run steps after a timeout window
- Output is automatically aggregated to the server

Architecture (Distributed Mode)



Architecture (Local Mode)



Task Farmer Details

- Server takes Fasta input and splits it on the fly into small batches (default 32 sequences)
- Clients request a batch of sequences from server
- Each iteration is tracked for completion
- Output is sent back to server and appended to the output filename

Running BLAST

- Add this to your dot file….

```
module use --append /usr/common/tig/Modules/modulefiles
module use --append /usr/common/jgi/Modules/modulefiles


module load taskfarmer/1.3
```

Example Blast Batch Job

```
#!/bin/sh

#PBS -N blastp
#PBS -q regular
#PBS -l mppwidth=7200,mppnppn=24,walltime=1:00:00
#PBS -A m342
#PBS -V

cd $PBS_O_WORKDIR

#
# Setup task farmer
export TF_HOME=/global/common/carver/tig/taskfarmer/stats/
export STAGE=$TF_HOME/share/taskfarmer/stage.cacher
export PATH=$PATH:/project/projectdirs/genomes/apps/bin:$TF_HOME/bin
```

Example Blast Batch Job

```
# Specify the entire path to the database.  This is used by the stage script to cache
# the DB on the compute nodes.  So don't just change the tfrun line.
# When you copy files to Lustre scratch, be sure to increase the stripe count to improve
# performance.  Here is how...
# mkdir $SCRATCH/db
# lfs setstripe $SCRATCH/db -c -1
# cp </path/to/db/nr*> $SCRATCH/db/
#
export DB=/scratch/scratchdirs/canon/db/nr
```

Example Blast Batch Job

```
# Tweaks.  You can tune the batch size so that each step takes a reasonable amount of time.
# The batch size is the number of sequences processed in each step.
export BATCHSIZE=16
# Max expected time to process the previous number of sequences.  Adjust appropriately.
export SERVER_TIMEOUT=2700

# This can be used to monitor progress.  View...
# https://portal-auth.nersc.gov/project/genomes/stats/tfs.html?source=/project/genomes/sf/
    status-contigs.js
export STATUSFILE=/project/projectdirs/genomes/www/sf/status-contigs.js

# Game time.  tfrun will handle launching the compute processes.
# Merge all your inputs into one mongo file.
tfrun blastall -i mer330.faa -o mer330.faa.blastout -d $DB -p blastp -e 10 -F "m S" -m 9
```

Task Farmer Considerations

- Try to determine a good estimate of your run time
- Adjust the timeout or batch size to insure that enough time is allowed for each iteration
- Use the stage.cacher to cache DB files.
- Reduce output as much as possible
- Start small and dial up.

MySGE

- Starts a parallel job using the system scheduler
- Spawns an SGE scheduler as the users
- Uses aprun to launch SGE execution daemons on the compute nodes (as the user)
- User submits to the personal SGE scheduler to run jobs

Demonstration

MySGE Considerations

- While the virtual private cluster is running, your NERSC repo is getting charged (even if the cores are idle).
- Still need to consider I/O issues. Launching a 1024 jobs can create a large IO load

Ported Applications

Thanks to Rob Egan, Alex Copeland, and others for starting to port applications

- ABySS
- Bowtie
- BWA
- bio-perl

Final Notes

- MySGE and Taskfarmer are home-grown and still under development (expect bugs)
- Hopper is currently overloaded (long waits)
- NERSC is looking at methods to natively support serial jobs (stay tuned)