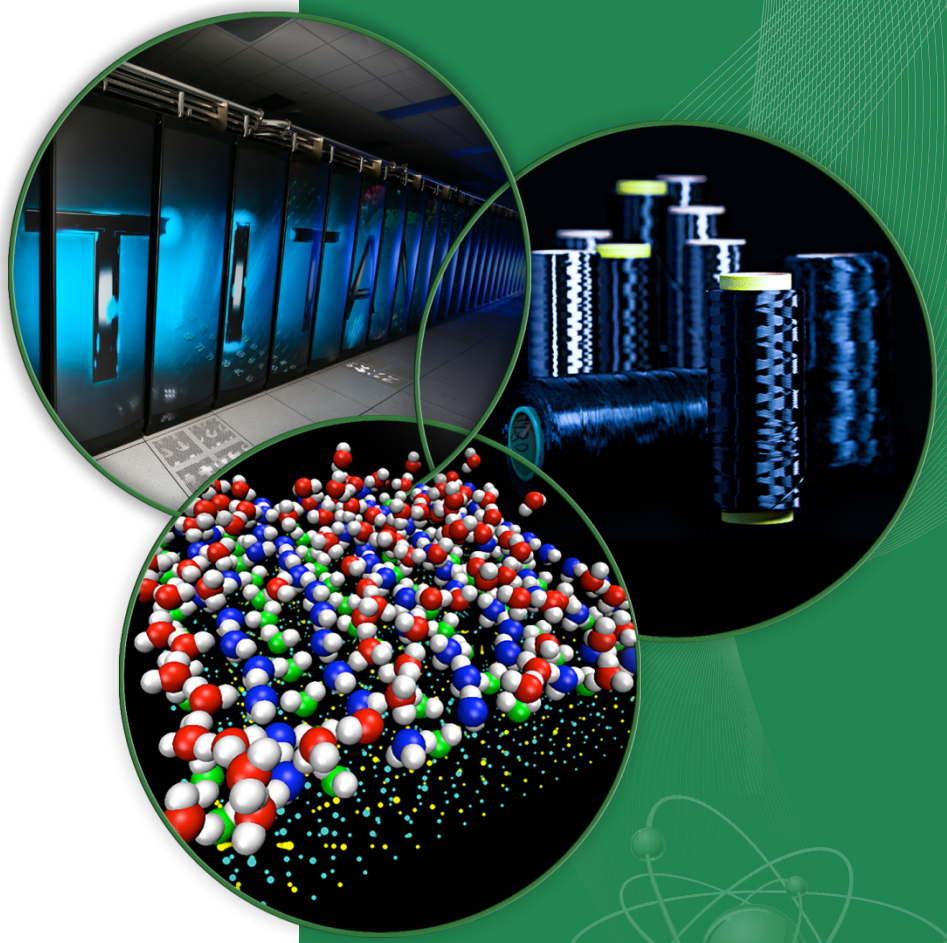
# Best Practices @ OLCF

## (or, *How to OLCF*)

Bill Renaud
OLCF User Support



ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory

# General Information

- The goal of this presentation is to give a brief introduction to OLCF's systems and policies
- This is by no means an all-inclusive presentation
- Feel free to ask questions

# OLCF Best Practices

1. Stay informed
2. Access the software you need
3. Discover how Titan & Eos differ from typical clusters
4. Understand the batch queue system
5. Master the `aprun` command
6. Learn OLCF's project & allocation policies
7. Familiarize yourself with available data storage options
8. Develop a data strategy
9. Become Lustre® savvy
10. Optimize HPSS usage
11. Know how to get help

# Best Practice 1:

# Stay informed



OAK RIDGE
National Laboratory

# Staying Informed

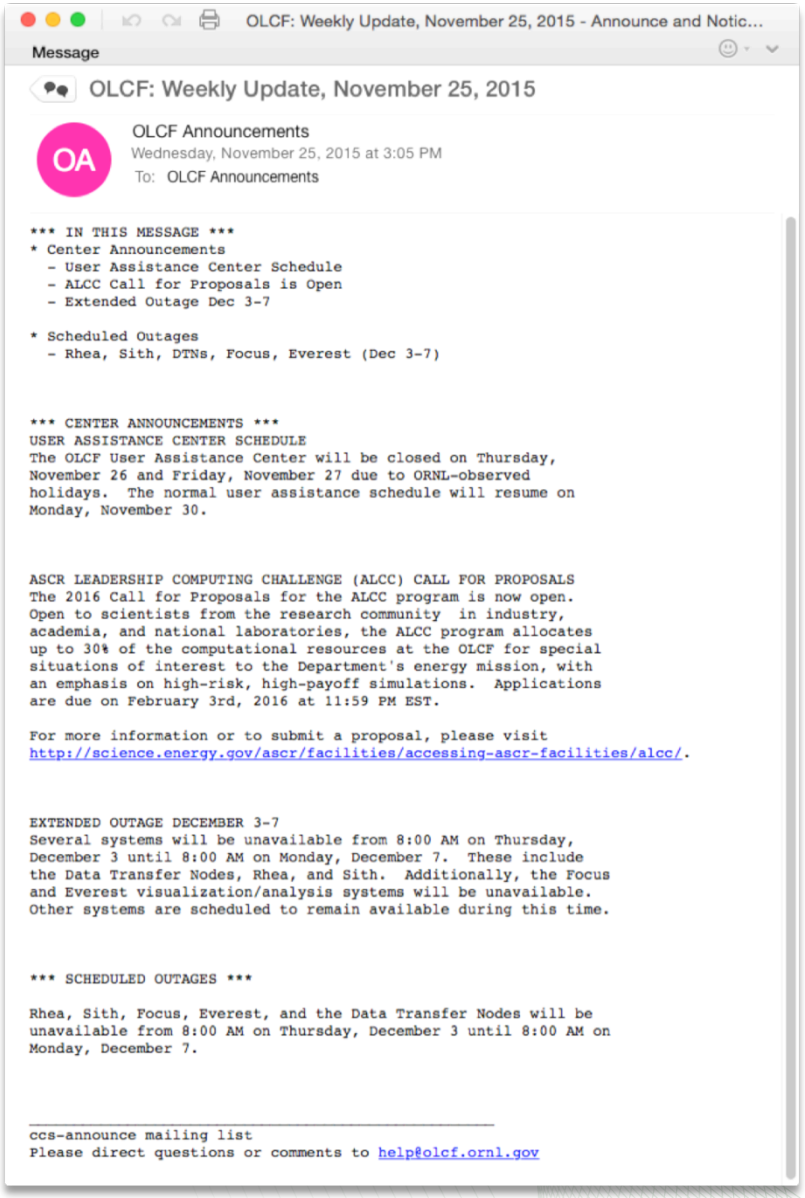
- OLCF provides multiple layers of user notifications about system status and downtimes
  - Email lists
  - Status indicators on olcf.ornl.gov
  - Twitter (@OLCFStatus)
- For more information, see the OLCF website:
  http://www.olcf.ornl.gov/kb_articles/communications-to-users/

# Staying Informed
## *Email Lists*

- *Announce* lists
  - All users are required to be members
  - System-specific (*-announce@email.ornl.gov) & center-wide (ccs-announce@email.ornl.gov) lists
  - Used for major announcements, weekly notice, etc.
- *Notice* lists
  - "Recent" users (active within last 2 weeks) are members; permanent opt-in or opt-out available (contact help@olcf.ornl.gov)
  - System-specific lists only (*-notice@email.ornl.gov)
  - Used for minor updates, system status, etc

# Staying Informed
## *Weekly Update*

- Sent on Wednesday
- Contains announcements about outages, training, etc.
- **All OLCF users should receive this email**



OLCF: Weekly Update, November 25, 2015 - Announce and Notic...

Message

OLCF: Weekly Update, November 25, 2015

OA

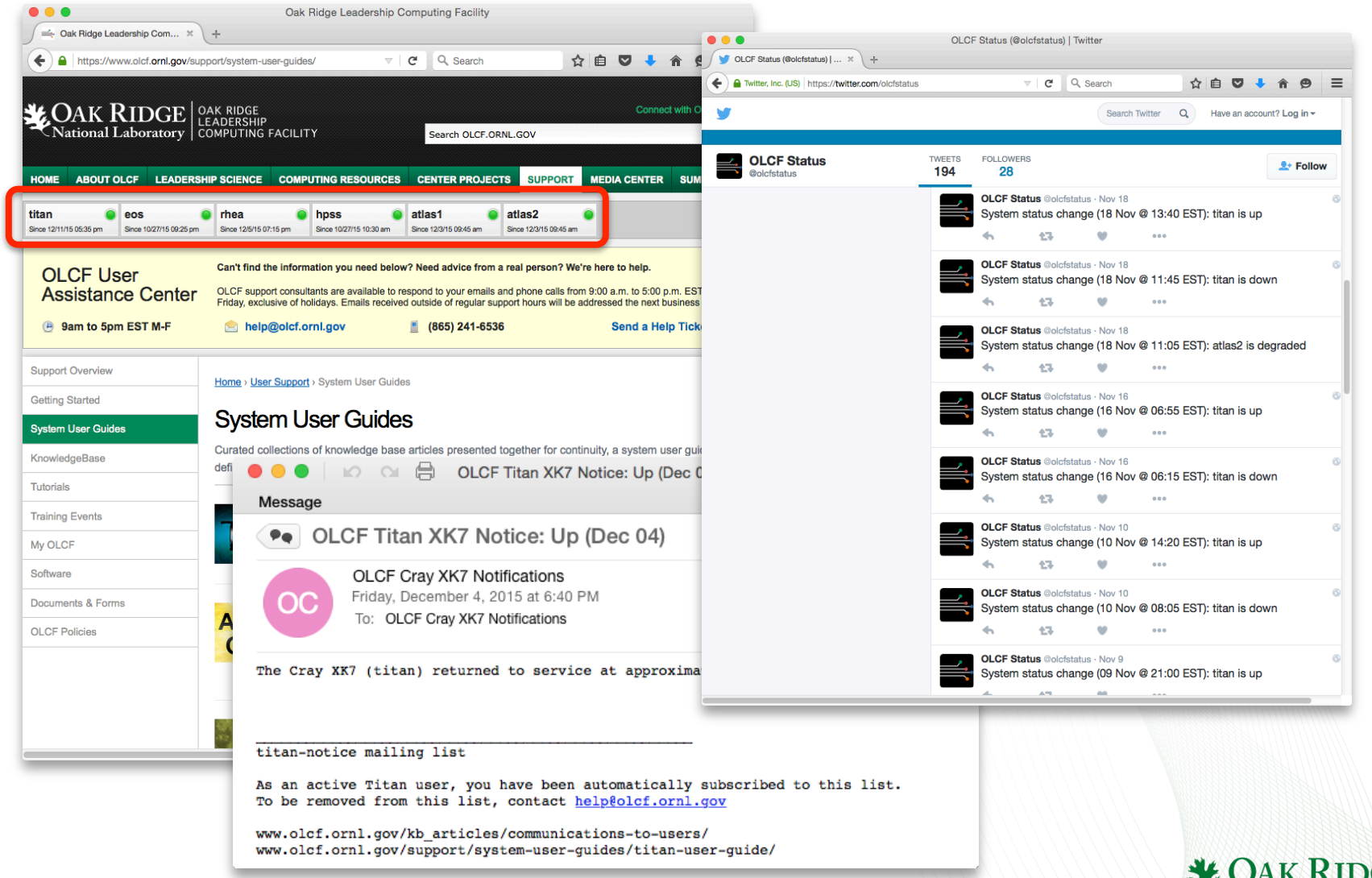
OLCF Announcements
Wednesday, November 25, 2015 at 3:05 PM
To: OLCF Announcements

```
*** IN THIS MESSAGE ***
* Center Announcements
  - User Assistance Center Schedule
  - ALCC Call for Proposals is Open
  - Extended Outage Dec 3-7

* Scheduled Outages
  - Rhea, Sith, DTNs, Focus, Everest (Dec 3-7)


*** CENTER ANNOUNCEMENTS ***
USER ASSISTANCE CENTER SCHEDULE
The OLCF User Assistance Center will be closed on Thursday,
November 26 and Friday, November 27 due to ORNL-observed
holidays.  The normal user assistance schedule will resume on
Monday, November 30.


ASCR LEADERSHIP COMPUTING CHALLENGE (ALCC) CALL FOR PROPOSALS
The 2016 Call for Proposals for the ALCC program is now open.
Open to scientists from the research community  in industry,
academia, and national laboratories, the ALCC program allocates
up to 30% of the computational resources at the OLCF for special
situations of interest to the Department's energy mission, with
an emphasis on high-risk, high-payoff simulations.  Applications
are due on February 3rd, 2016 at 11:59 PM EST.

For more information or to submit a proposal, please visit
http://science.energy.gov/ascr/facilities/accessing-ascr-facilities/alcc/.


EXTENDED OUTAGE DECEMBER 3-7
Several systems will be unavailable from 8:00 AM on Thursday,
December 3 until 8:00 AM on Monday, December 7.  These include
the Data Transfer Nodes, Rhea, and Sith.  Additionally, the Focus
and Everest visualization/analysis systems will be unavailable.
Other systems are scheduled to remain available during this time.


*** SCHEDULED OUTAGES ***

Rhea, Sith, Focus, Everest, and the Data Transfer Nodes will be
unavailable from 8:00 AM on Thursday, December 3 until 8:00 AM on
Monday, December 7.


_______________________________________________
ccs-announce mailing list
Please direct questions or comments to help@olcf.ornl.gov
```

# Staying Informed
## *System Status*

- Logs from monitoring software parsed to make educated guess on system status
- Status is sent to multiple destinations
  - OLCF Website
  - Twitter (@OLCFStatus)
  - System “notice” email lists
- Fairly accurate, but still a fully automated process
  - Possibility of both false positives and false negative
  - We do take some measures to mitigate this

# Staying Informed
## *System Status*






OAK RIDGE National Laboratory

# Best Practice 2:

# Access the software you need



OAK RIDGE
National Laboratory

# Finding Software

- Basic commands are part of the default environment
- Other software is provided via the `module` command
- Compilers, development tools (debugging, profiling, optimization), various libraries & scientific applications are all available
- Software information is available on the our website
  - Available software listing
    https://www.olcf.ornl.gov/support/software/
  - Changes to default versions (planned as well as historical)
    https://www.olcf.ornl.gov/kb_articles/software-news/
  - Some software documentation is available

# Software Availability & Installation

- Cray-provided software (compilers, MPI, CUDA, etc.) goes through testing before we receive it (so there's a delay before we receive it)
- You are free to install software in your directories (subject to export control, licensing, etc.)
- If you think software would be of general interest, you might ask us to install it for general use
  - Via the Software Request Form
    http://www.olcf.ornl.gov/support/software/software-request/
  - Alternatively, via email to help@olcf.ornl.gov

  *N.B.: Requests are sent our software council for approval*

# Using Modules

- Much of our software is managed with the Environment Modules package
  - Modules are managed with commands like "`module avail`", "`module load`", etc.
  - For detailed information, see online documentation
    http://modules.sourceforge.net
    https://www.olcf.ornl.gov/kb_articles/using-modules/
- Important notes
  - Pay attention to dependency error messages when loading/changing modules
  - On Cray systems, change the `PrgEnv` module to change compiler (& not the `pgi`/`intel`/etc. modules)

# Best Practice 3:

# Discover how Titan & Eos differ from typical clusters



OAK RIDGE National Laboratory

# Understanding Titan and Eos

- Our Cray XK7 (Titan) and XC30 (Eos) may seem similar to a standard Linux cluster but there are important differences
  - Heterogeneous nodes (login/service vs. compute nodes)
  - Compiling
  - Batch job environment

# Titan/Eos Node Types

- Titan and Eos are heterogeneous systems
  - Somewhat specialized nodes for different tasks
    - Login/service nodes handle general interaction, compiling, and managing batch jobs
      - If you are typing a command, you're on a login/service node
    - Compute nodes simply run parallel jobs
      - Can only be accessed via `aprun` (no login access)
  - Filesystems may differ
    - Compute nodes don't mount user home
    - Compute nodes mount project home, but it is read-only and can be as much as 30 minutes out of date
  - Different processor, memory, and network configurations

# Titan Compute Nodes

- 16-core AMD Opteron™ processor
  - Contains 8 “Bulldozer” modules, each with 2 integer cores and a shared floating point unit
- 1 NVIDIA® Tesla® K20X GPU
- 32 GB CPU memory plus 6 GB GPU memory
- Connects to 3d torus via Gemini ASIC (shared w/1 other node)
- Cray terminology (as used in `aprun` documentation)
  - ***Compute Unit*** **–** A single Bulldozer module
  - ***NUMA Node*** **–** A collection of 4 Bulldozer modules

# Eos Compute Nodes

- 2 8-core Intel® Xeon® processors
  - Hyper-Threading is supported
    - If enabled, the OS views each physical core as 2 virtual cores
- 64 GB memory
- Connects to Dragonfly interconnect via Aries ASIC (shared w/3 other nodes)
- Cray terminology (as used in `aprun` documentation)
  - ***Compute Unit*** **–** A single physical core (or 2 virtual cores)
  - ***NUMA Node*** **–** A single 8 (physical) core Xeon® processor

# Compiling for the XK7 and XC30

- Because of different node types, you are actually cross-compiling
  - This can make utilities such as `autoconf` and `cmake` challenging to use
- Compiling for batch/login nodes
  - Not common, but occasionally necessary
  - See §7.2 of the Titan User Guide for examples
    https://www.olcf.ornl.gov/support/system-user-guides/titan-user-guide/

# Compiling for the XK7 and XC30

- Controlled by a combination of module files

| | |
|---|---|
| `PrgEnv-?` | Loads compiler, math, MPI, etc. modules |
| `craype` | Loads compiler wrappers (`cc`/`CC`/`ftn`) |
| `cudatoolkit` | Loads GPU libraries |
| `hdf5`, `netcdf`, etc. | Miscellaneous other libraries |

- Compiler commands (C: `cc`, C++: `CC`, Fortran: `ftn`) are common & independent of compiler vendor
- Many libraries automatically linked
  - MPI, Math, Scientific, GPU, etc.
  - No need to add `-lmpi`, `-lblas`, etc.
  - Another challenge for automated build tools

# Best Practice 4:

# Understand the batch queue system



OAK RIDGE National Laboratory

# Running Batch Jobs

- Batch system is TORQUE combined with Moab® (both from Adaptive Computing)
  - Titan & Eos integrate these with Cray's ALPS
- More detailed batch job/batch queue information is available in our System User Guides on the website
  - Sample scripts
  - Queue limits
  - Scheduling policies

# Running Batch Jobs

- Jobs are charged based on what you make unavailable to others, not what you use
  - If you request 1,000 nodes but only use 1, your project will be charged for 1,000
  - Charge (core-hours) is computed as:

    *core-hours = elapsed walltime * nodes allocated * cores per node*

    - On Titan & Eos, *cores per node* = 30
    - Elsewhere, *cores per node* = 16
- Parallel job launcher differs from system to system
  - Titan/Eos: `aprun`
  - Clusters: `mpirun`

# Scheduling Basics

- The basic priority factor is job submit order
- Scheduling policies modify the “apparent” submit time to adjust priority
- Scheduling policies differ from system to system based on that system’s mission
  - e.g. Titan heavily favors leadership-class jobs
- Smaller jobs can backfill even if larger jobs have higher priority

# Determining Why a Job Isn't Running

- Can be a number of reasons
  - Insufficient nodes available
  - Pending downtime
  - Queue policy
  - Dependency on another job not met
- Use `checkjob <jobid>` to diagnose
  - Use `-v` for verbose mode
  - Reason for job not running is usually near the end (although the verbiage may be confusing)
- Use `showres` to see upcoming reservations/outages
  - Also shows running jobs…look carefully

# Determining Why a Job Isn't Running

- Insufficient resources available

```
NOTE:  job cannot run  (insufficient available procs:  3264 available)
```

- Unresolved depencency

```
NOTE:  job cannot run  (dependency 1851375 jobcomplete not met)
```

- Queue policy issue

```
NOTE:  job violates constraints for partition titan (job 1854474
violates active HARD MAXJOB limit of 2 for qos smallmaxjobs user
partition ALL (Req: 1  InUse: 2))

BLOCK MSG: job 1854474 violates active HARD MAXJOB limit of 2 for qos
smallmaxjobs user partition ALL (Req: 1  InUse: 2) (recorded at last
scheduling iteration)
```

# Determining Why a Job Isn't Running

- Pending maintenance period/reservation
  - This is example output from `showres`
  - Look for alphanumeric `ReservationIDs` (those without letters are likely currently running jobs)
  - Note the full system reservation starting in 15 hours (`PM.128`)…is my job asking for more than 15 hours?

```
ReservationID  Type S     Start         End  Duration    N/P    StartTime

DTNOutage.127  User -  15:05:13  1:03:05:13  12:00:00   11/176  Tue Jan 28 08:00:00
PM.128         User -  15:05:13  1:03:05:13  12:00:00 18688/299008 Tue Jan 28 08:00:00
PM_login.129   User -  15:05:13  1:03:05:13  12:00:00   16/2048 Tue Jan 28 08:00:00
```

# Determining When a Job Will Start

- Can be difficult to tell since jobs submitted in the future can affect it
- Several utilities can give a general idea of start time
  - `showstart <jobid>` is not always reliable
  - `showq` output is sorted by priority (can be helpful)
  - `mdiag -p` gives detailed priority info
    - Shows boosts based on job size
    - Shows overallocation penalties

# Dealing With Failed Nodes

- Sometimes nodes fail when your job starts
  - Since node allocation has already happened, the batch system can't replace them
  - You can work around this by requesting "extra" nodes

```
#!/bin/bash
#PBS –lnodes=104
...
APRUN_RETURN_VALUE=1
while [[ $APRUN_RETURN_VALUE –ne 0 ]]; do
   aprun –n1600 ./a.out
   APRUN_RETURN_VALUE=$?
done
...
```

# Best Practice #5:

# Master the `aprun` command



OAK RIDGE National Laboratory

# Important `aprun` Options

| Option | Description |
|---|---|
| `-n` | Total number of MPI tasks |
| `-N` | Number of MPI tasks per physical node (<=16 on Titan, varies on Eos) |
| `-S` | Number of MPI tasks per NUMA node (<=8 on Titan, varies on Eos) |
| `-d` | Reserve this number of cores for each MPI task & its threads |
| `-j` | Number of tasks per Compute Unit<br>*On Titan, `-j1` idles 1 core per Bulldozer, `-j2` uses both cores*<br>*On Eos, `-j1` disables Hyper-Threading while `-j2` enables it* |
| `-cc` | Binds tasks to cores (prevents the OS from migrating tasks between cores within a node) |
| `-r` | Assigns system services to a compute core. Specified as `-r1`. Reduces the cores available for your code, but may improve performance by preventing unexpected context switches (i.e. limits “jitter”) |

# `aprun` Environment Variables

| Variable | Description |
|---|---|
| `OMP_NUM_THREADS` | Number of OpenMP threads to spawn per MPI task (can also be controlled by calls in the code) |
| `MPICH_RANK_REORDER_METHOD` | Controls how tasks are placed on nodes<br>0: Round robin [0+4], [1+5], [2+6], [3+7]<br>1: SMP style [0+1], [2+3], [4+5], [6+7]<br>2: Folded [0+7], [1+6], [2+5], [4+3]<br>3: Custom |
| `MPICH_RANK_ORDER` | The name of a file which contains the rank order for `MPICH_RANK_REORDER_METHOD=3` |
| `APRUN_XFER_LIMITS` | If set, shell limits (i.e. things set with `limit` or `ulimit`) will be passed to the compute nodes |

# `aprun` Notes

- For OpenMP codes, you ***must*** use both `OMP_NUM_THREADS` and the `-d` option to `aprun`
  - The system spawns the number of threads specified by `OMP_NUM_THREADS` (or appropriate calls in the code)
  - The system schedules those across the number of cores reserved by the `-d` option

  *N.B.: If you don't specify a `-d` option, the system will default to 1…meaning all threads go to a single core!*

# `aprun` Notes

- Floating-point-intensive codes on Titan may get better performance with `-j1` (i.e. "idle" one integer core per Bulldozer)
  - *No, this is not "wasting" half of your cores!*
- Consider the total `aprun` task layout you've requested when determining how many nodes your job needs
- Don't run >100 simultaneous `aprun` commands (If you need to do this, contact us about *wraprun)*
- For more information/documentation, try "`man aprun`" and/or "`man intro_mpi`"

# Common `aprun` Error Messages

- `aprun` provides a great deal of control in task layout
  - Because of this flexibility, it can be easy to request a task layout that cannot be fulfilled
  - Any of several error messages will be provided if there is a job layout problem
    - These are described on the upcoming slides
    - There are very subtle differences between these errors and it's easy to get confused, so feel free to ask questions
    - There's a slide showing some examples that lead to these errors… feel free to try your own experiments

# Common `aprun` Error Messages

```
aprun: [NID 94]Exec a.out failed: chdir /autofs/
na1_home/user1 No such file or directory
```

- You tried to access a non-existent directory on a compute node (the key is "*No such file or directory*")
  - When `aprun` starts a task on a compute node, it will cd into the `$CWD` of the shell from which `aprun` was called
  - Remember, compute nodes don't mount all directories
  - Your batch script starts in `$HOME`, so you'll always need to do a `cd`
- To fix, run `aprun` from a directory visible to compute nodes

*Also remember to place input files, .so files, etc. in directories visible to compute nodes (failure to do so won't cause this error, but this seems like a logical place to mention it)*

# Common `aprun` Error Messages

```
apsched: claim exceeds reservation's node-count
```

- You are trying to use more physical nodes than are available to your job because…
  - The `aprun` request requires more nodes than the job requested, or
  - Your request was correct, but at launch time some nodes were discovered to be down (see potential fix later in this presentation)
- To fix
  - Double-check your aprun/node request and re-submit
  - If the request appears to be correct, you may wish to contact us to see if there were node failures

# Common `aprun` Error Messages

`apsched: claim exceeds reservation's CPUs`

- There's a problem with the intra-node layout of your job
  - You're requesting more NUMA nodes than are present on the physical node (i.e. `-N4 -S1`)
  - You've "deactivated" half of your cores with `–j1`, but your remaining `aprun` options require some/all of the deactivated cores (i.e. `-j1 -S8` on Titan or `-j1 -S16` on Eos)
- To fix, modify your `aprun` command to ensure
  - You're not requesting more NUMA nodes than are present on each physical node (currently 2 for both Titan and Eos)
  - You're not idling cores/disabling Hyper-Threading while simultaneously requesting those resources via `-S`, `-d`, or other options

# Common `aprun` Error Messages

```
apsched: -S value cannot exceed max CPUs/NUMA node
apsched: -S times -d cannot exceed max CPUs/NUMA node
```

- You're trying to use more cores per NUMA node than are physically present
  - The "`exceeds reservation's CPUs`" error is similar but indicates the NUMA node has enough cores but you've deactivated some; this error means it doesn't have enough cores no matter what
- To fix, make sure your `-S` request (or `-S` times `-d` if you specify both) is less than the number of cores per NUMA node (8 for Titan; 16 for Eos)

# Examples of `apsched` errors

*Examples assume 1 Titan node (16 cores—2 NUMA nodes w/8 cores each)*

```
$ aprun -n4 -N2 ./a.out
apsched: claim exceeds reservation's node-count
```
*…because 4 tasks (`-n4`) @ 2 per node (`-N2`) requires 2 nodes but we only have 1*

```
$ aprun -n2 -S1 -j1 -d8 ./a.out
apsched: claim exceeds reservation's CPUs
```
*…because `-S1` with `-d8` requires all 8 cores per NUMA node, but `-j1` idles half of them*

```
$ aprun -n4 -N4 -S1 ./a.out
apsched: claim exceeds reservation's CPUs
```
*…because `-N4` with `-S1` requires 4 NUMA nodes/node, but there are only 2*

```
$ aprun -n4 -S2 -d16 ./a.out
apsched: -S times -d cannot exceed max CPUs/NUMA node
```
*…because `-S2` with `-d16` requires 32 cores/NUMA node, but there are only 8*

```
$ aprun -n16 -S16 ./a.out
apsched: -S value cannot exceed max CPUs/NUMA node
```
*…because `-S16` requires 16 cores/NUMA node, but there are only 8*

# Best Practice 6:

# Learn OLCF's project & allocation policies



OAK RIDGE National Laboratory

# System and Group Access

- Projects are
  - Granted access to systems
  - Assigned specific Unix groups
- Users are assigned to projects & “inherit” the project’s system accesses & Unix groups
  - Users aren’t directly added to systems/groups
  - To access another group’s data, you need to join their project –*or*– have them place the data in `$WORLDWORK`

# Finding Your Project's ID and Allocation

- Use `showproj` and `showusage` to list projects and usage, respectively
- Both commands have a help option (run with `-h`)

```
$ showproj

user1 is a member of the following project(s) on titan:
  abc123

$ showusage

titan usage for the project's current allocation period:
                                  Project Totals           user1
 Project      Allocation        Usage    Remaining          Usage
_____________________________|_____________________________|_____________
 abc123         7350000     |      11138      7338862   |         12
```

# Managing Your Allocation

- Projects are ***NOT*** disabled for going over allocation
  - But jobs will have a lower priority to facilitate some degree of "fairshare" with projects that have allocation remaining
    - 30 day penalty for slightly over (usage 100-125% of allocation)
    - 365 day penalty for usage >125% of allocation
- Since we don't disable projects, we don't issue refunds (*per se*)
  - If many jobs were affected by a system issue, let us know (we can delay the priority reduction)
  - This has the same effect as a refund but is easier to manage

# Project Closeout

- When your project ends, you'll no longer be able to access OLCF resources
  - Even if you're continuing on other projects, you won't be able to access the closed project's storage areas
- Users will be given a month for data retrieval
  - You won't be able to access the main resources…you will need to use the Data Transfer Nodes

# Best Practice 7:

# Familiarize yourself with available data storage options



OAK RIDGE National Laboratory

# Data Storage Locations

- User-centric and project-centric areas
- Home, work (scratch), and archival areas
- Multiple storage technologies

| Storage Area | Technology | Location* |
|---|---|---|
| User *home* | NFS | `/ccs/home/$USER` |
| Project *home* | NFS | `/ccs/proj/project_id` |
| User *work* | Lustre® | `$MEMBERWORK/project_id` |
| Project *work* | Lustre® | `$PROJWORK/project_id` |
| Globally-shared *work* | Lustre® | `$WORLDWORK/project_id` |
| User *archive* | HPSS | `/home/$USER` |
| Project *archive* | HPSS | `/home/project_id` |

**These are the recommended ways to reference the different locations; they're not necessarily the absolute path names.*

# Directory Permissions

- Home directory permissions
  - User home default is `0750` & can be changed
  - Project home is `0770` & can't be changed
- Archive directory permissions
  - User home default is `0700` & can be changed
  - Project home is `0770` & can't be changed
- Scratch directory permissions can't be changed
  - `$MEMBERWORK`, `$PROJWORK`, & `$WORLDWORK` have different defaults (see the Data Management User Guide)
  - Move files between scratch areas instead of changing permissions on the current directory

# Data Backups

- NFS directories are backed up to a limited extent
  - Hourly/daily/weekly snapshots are in `/autofs/nccs-svm1_home?/.snapshot`
  - `ls -lnd /ccs/home/$USER` will tell you which subdirectory of `/autofs` to use

```
$ ls -lnd /ccs/home/$USER
lrwxrwxrwx 1 0 0 31 May 26  2015 /ccs/home/user1 -> /autofs/nccs-svm1_home2/user1
```

*Note: -n not really needed…simply used for brevity (numeric uid/gid)*

- Lustre® **is not** backed up
- HPSS **is not** backed up

# Monitoring Storage Usage

- For home directories, use the `quota` command

```
$ quota
quota: error while getting quota from master:/cm/shared for user1 (id 98765): Connection refused
quota: error while getting quota from master:/home for user1 (id 98765): Connection refused
Disk quotas for user user1(uid 98765):
     Filesystem  blocks   quota   limit   grace   files   quota   limit   grace
nccsfiler3.ccs.ornl.gov:/vol/home1
                       8  10485760 10485760               2  4294967295 4294967295
nccsfiler4.ccs.ornl.gov:/vol/home2
                 8669108  15728640 15728640           51766  4294967295 4294967295
```

- For Lustre® directories, use `lustredu`

```
$ lustredu $MEMBERWORK/stf007
Last Collected Date        Size    File Count  Directory
2014-01-10 12:30:08      31.21 MB           9  /lustre/atlas1/stf007/scratch/user1
```

- For archive directories, use `showusage`

```
$ showusage -s hpss
HPSS Storage in GB:
               Project Totals      user1
Project        Storage             Storage
_______________|___________________|_____________
user1          |    550.65         |    550.65
abc123         |   2107.18         |    106.52
```

# Data Storage Documentation

- Data Management User Guide
  https://www.olcf.ornl.gov/computing-resources/data-management/data-management-user-guide/
- Data Management Policy
  https://www.olcf.ornl.gov/kb_articles/data-management-policy

# Best Practice 8:

# Develop a Data Strategy



OAK RIDGE
National Laboratory

# Data Considerations

- OLCF systems can generate large volumes of data very quickly
- It's important to develop a project data strategy as soon as possible
- Several things to consider
  - Where project members will store data
  - File ownership/permissions
  - Transferring data to other locations

# Data Accessibility

- Files in the *project work* area will often inherit the project's group id
  - May not be the case for multiple levels of subdirectories
- Still must consider file ownership and permissions
  - Users have no control over ownership (can't run `chown`)
  - Users can control permissions with `chmod` and `umask`
- Try to catch issues early (much easier to "fix" hundreds of files than tens of thousands)

# Transferring Data

- Data transfer nodes (`dtn.ccs.ornl.gov`) are the preferred place for external data transfer
- Titan external login nodes & batch nodes work well for internal HPSS transfers
- Several ways to transfer your data
  - External (to OLCF): `bbcp`, `scp`, `gridftp`, globus.org
  - Internal: `hsi`/`htar` along with those listed above
- Start early/transfer data as it's generated

# Data Transfer Documentation

- System User Guides
  https://www.olcf.ornl.gov/support/system-user-guides/
- Data Management User Guide
  https://www.olcf.ornl.gov/computing-resources/data-management/data-management-user-guide/
- Presentations from prior training events
  https://www.olcf.ornl.gov/wp-content/uploads/2015/02/Data-Transfer-Options1.pptx

# Best Practice 9:

# Become Lustre® savvy



OAK RIDGE
National Laboratory

# Lustre®

- We'll use some Lustre® terms, but a full discussion of it is beyond the scope of this talk (but can be found on the website)
  https://www.olcf.ornl.gov/kb_articles/lustre-basics/
- It's critical that you understand your application's I/O
  *…otherwise it is difficult to apply these best practices*
- The upcoming slides summarize data from
  - Our Spider Best Practices page
    https://www.olcf.ornl.gov/kb_articles/spider-best-practices/
  - Sarp Oral's IO Best Practices presentation from the 2015 User Meeting
    https://www.olcf.ornl.gov/wp-content/uploads/2015/02/OLCF-IO-Best-Practices.pdf

# Lustre® Striping

- Lustre “stripes” files across multiple storage targets
  - `lfs getstripe` shows the striping of an existing file/directory
  - `lfs setstripe` sets striping for a new file/directory
- You have control over several stripe characteristics
  - *count*: The number of OSTs over which the file is striped
  - *size*: The size of each file chunk
  - *index*: The OST on which the first stripe will be placed
- There are some settings that should be avoided (see the next few slides)
  - General theme: minimize impact on metadata

# Lustre® Best Practices

- When setting up striping…
  - Use a stripe count of 1 for small files or for directories that will contain many small files
  - Use a stripe count greater than the default (which is currently 4) for large, shared files
- …and ***do not***…
  - Use a stripe count >512
  - Use a stripe count of -1 (stripes over all OSTs)
  - Use a stripe index (starting OST) other than -1
  - Stripe your entire scratch area wholesale

# Lustre® Best Practices

- Avoid editing/building/compiling in Lustre®
- Use `ls -l` only when absolutely necessary
- Perform `stat` operations from a single task
- Open files read-only whenever possible
- Read small, shared files from a single task
- Don’t fill up individual OSTs
- Use large and stripe-aligned I/O whenever possible
- Limit the number of files in a single directory
- Use high-level I/O libraries/middleware if possible
  *ADIOS, HDF5, pNetCDF, community-developed libraries, etc.*

# Lustre® Purge Guidelines

- Lustre® is regularly purged (i.e. old files are deleted) to ensure ample free space & optimal performance
  - See the Data User Guide for the purge interval (different areas are purged at different intervals)
  - No notice—***archive important files as soon as possible!***
  - Delete files as soon as you know they're not needed
- The purge is an overall good, but can be troublesome (i.e. files purged before your job starts)
  - Easiest way to handle is to have your job check for needed files & do an `hsi get` if they've been purged
  - Please don't do a wholesale `touch` operation

# Best Practice 10:

# Optimize HPSS usage



OAK RIDGE
National Laboratory

# The High Performance Storage System

- HPSS is the proper location for long-term storage
- Project home and project work areas offer a common area for files shared among project members, but neither are long-term storage
  - Home areas are somewhat space constrained
  - Lustre® areas are subject to the purge
  - There's still a need for a backup location
- HPSS is accessed via the `hsi` and `htar` commands

# HPSS Best Practices

- Don't run multiple simultaneous transfers
  - You're likely not getting the parallelism you expect
- File size best practices
  - For optimal transfer performance, use files ≥ 768GB
  - Minimum recommended file size is 512MB
    - Smaller files will be handled but read/write performance may be negatively affected
    - If you have numerous small files, we recommend bundling with `htar` to achieve the 512MB threshold
  - When using `htar`, no individual member file can be ≥ 64GiB (but the archive itself can be)
- Avoid numerous consecutive `hsi get` calls

# HPSS Best Practices

- Bad practice-successive `hsi get` calls

```
$ hsi get file1

$ hsi get file2

$ hsi get file3

$ hsi get file4
```

- Good practice-create a list file & call `hsi` once

```
$ cat getfiles.lst
get <<EOF
file1
file2
file3
file4
EOF

$ hsi "in getfiles.lst"
```

# Best Practice 11:

# Know how to get help



OAK RIDGE
National Laboratory

# Where do I find documentation?

- OLCF Website/System User Guides
  https://www.olcf.ornl.gov
  https://www.olcf.ornl.gov/support/system-user-guides/
- CrayDocs
  http://docs.cray.com
- NVIDIA hosted documentation
  http://docs.nvidia.com

# Training Opportunities

- We host numerous training events through the year
  - Monthly User Conference Call
  - Software-specific courses
  - GPU Hackathons
- Watch for announcements in the Weekly Update

# Working With User Support

- Email is often the best option to contact us
  - Especially for sending long/complicated error messages
  - Send as many error messages as possible (or place them in a file & direct us to the file)
  - "Send" us codes by creating a `.tar` file & directing us to it
    - `$WORLDWORK`, anyone?
    - More efficient that sending code/.tar file via email
    - Include all files necessary to run
- Start a new ticket for new issues instead of replying to an old ticket
  - Helps us in classifying/searching through old tickets
  - Gives it greater visibility

# Requesting a priority boost/higher walltime limit/purge exemption/etc

- Request can be made from our Documents & Forms page (in the third section of the page)
  https://www.olcf.ornl.gov/support/documents-forms/
  - Reviewed by Resource Utilization Council, so make requests well in advance to allow for review
  - If requesting job priority, make sure you submit the job… they often run more quickly than you expect

# OLCF Best Practices

1. Stay informed
2. Access the software you need
3. Discover how Titan & Eos differ from typical clusters
4. Understand the batch queue system
5. Master the `aprun` command
6. Learn OLCF's project & allocation policies
7. Familiarize yourself with available data storage options
8. Develop a data strategy
9. Become Lustre® savvy
10. Optimize HPSS usage
11. Know how to get help

# Finally…

- We’re here to help you
- Questions/comments/etc. can be sent to the OLCF User Assistance Center
  - Staffed 9AM – 5PM US Eastern Time (exclusive of ORNL holidays)
  - help@olcf.ornl.gov
  - (865) 241-6536

*THANK YOU!*

# *Bonus Material*

# Best Practice 12:

# Authentication Basics



OAK RIDGE
National Laboratory

# Authenticating to OLCF Systems

- Interactive login access to OLCF systems is via Secure Shell (SSH)
- Our systems use “two-factor” authentication via user-selected PINs and RSA SecurID tokens
  - “Two-factor” means you’re using two of the three methods of user identification/authentication
    - Something you *have*: Your SecurID token which generates pseudorandom, time-sensitive codes
    - Something you *know*: Your PIN, which only you know
    - We don’t use the third, something you *are* (e.g. biometrics)
  - Other authentication methods, such as password and public key authentication, are not permitted

# Common Login Issues

- SSH doesn't prompt for a username
  - By default, it uses your username on the client system
  - You must tell SSH if your username differs
    - Command line: `ssh olcfusername@home.ccs.ornl.gov`
    - Can set this in `~/.ssh/config` file
    - Various ways to do this in graphical SSH clients
  - SSH won't tell you what username it's using

# Common Login Issues

- SSH prompts for a password
  - Typically happens after three PASSCODE failures
  - This is a fallback behavior of SSH
  - If you see this, kill the process (`Ctrl-C`) & try again
    - We don't use passwords, so there's nothing you can enter that will work

```
Enter PASSCODE:
Enter PASSOCDE:
Enter PASSCODE:
user1@titan.ccs.ornl.gov's password:
```

# Common Login Issues

- RSA token gets out of sync with the server
  - Sometimes you may be prompted for the ‘next tokencode’
    - *Tokencode*: The 6 digit number on your RSA token
    - *PIN*: The 4-8 digit number known only to you
    - *PASSCODE*: Your PIN followed by the current tokencode
  - When this happens, enter (only) the next tokencode your RSA token generates

```
Enter PASSCODE:
Wait for the tokencode to change, then enter the new tokencode :
```

# Miscellaneous RSA/SSH Tips

- Once you've used a tokencode, you can't re-use it
- If your PASSCODE has failed twice, let the tokencode change before you try again
  - Numerous failed attempts will result in your token being locked out