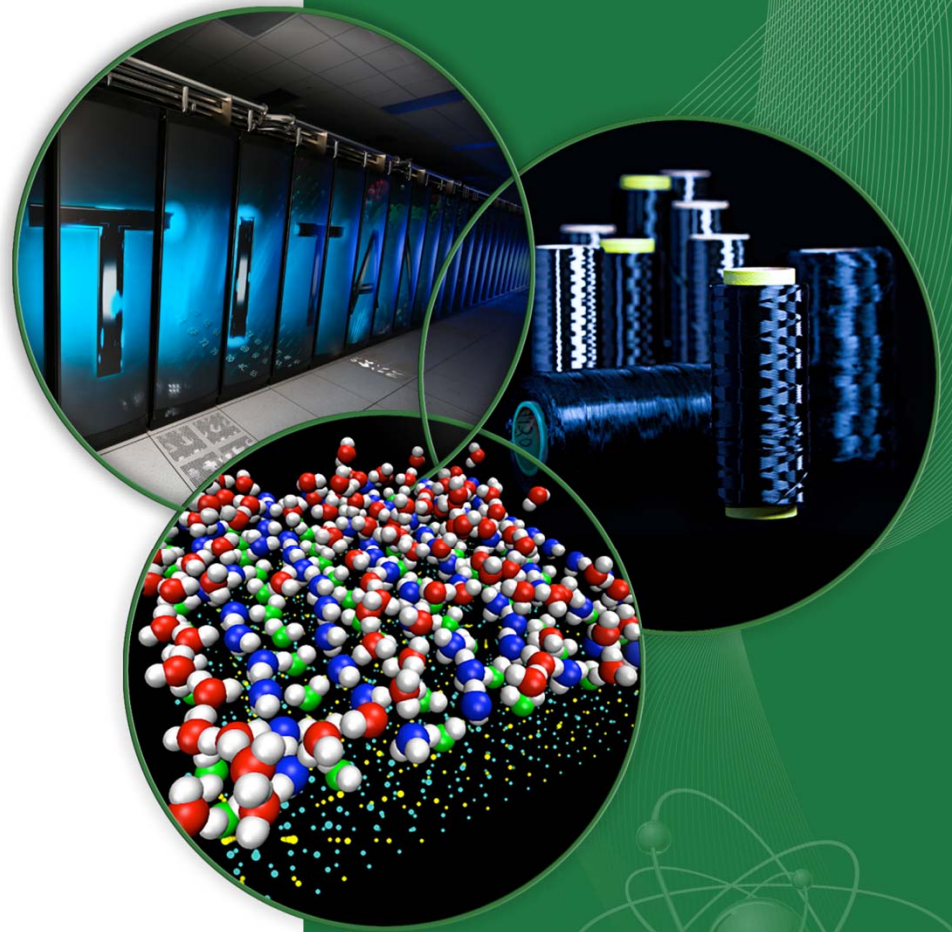


Year in Review: User Survey Results

Bill Renaud

Oak Ridge Leadership Computing Facility



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

 **OAK RIDGE**
National Laboratory

First...Thank You!

- Thanks to everyone who completed a survey
- The user survey is vitally important to us
 - Required by DOE
 - More importantly, it allows us to know what we're doing right and what needs improvement

Selected Highlights

- Once again, handled by ORISE
- 367 respondents (just under 30%)
- Good mix of INCITE, ALCC and DD projects
- Good mix of new and previous users
 - Over 2/3 had been users less than 2 years
- Overall satisfaction remains high (4.41/5.0)
 - Only 5% gave an overall rating of “neutral” (3.0/5.0)
 - No “dissatisfied” or “very dissatisfied” ratings

What You Like

- Both the people and the hardware
- Large job turnaround
- OLCF communications

What Improvements You'd Like To See

- Queue policy
- Faster account turnaround
- Fewer outages
- Training improvements
- Better communication of announcements/system status
- Disk/storage issues
 - Small quotas
 - Data access/sharing
 - Purge timeline

How We're Responding

- Staff members review results and then meet to identify how to address concerns
- Surveys have guided us to several system enhancements and policy changes
 - Scheduled DTNs
 - NetApp quota increase
 - Filesystem updates (\$WORLDWORK, etc.)
 - Cross-system workflows
 - HPSS enhancements
 - New dashboard (soon)
- The next few presenters will expand on these

Finally, ...

- Once again, thank you for completing the survey
- The 2014 User Survey is due out in early October
 - Please join in...help us help you!
 - Your opinion *DOES* matter to us
 - We'd love to have a 100% response rate

Year in Review: Systems - Eos and Rhea



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

Eos

- Cray XC-30
- 744 Nodes
- Prototyping for non-accelerated codes
- Available to INCITE, ALCC users
- Available to DD users upon request

Eos

- Aries Interconnect with Dragonfly network topology
 - Dragonfly is a “low-diameter” topology
 - Maximum “hops” between any two nodes is minimized
- 64GB RAM in each node
- Intel Xeon E5-2670 features 16 physical cores
 - 32 virtual CPUs / node via HyperThreading
- Shares the center-wide Atlas filesystem

Rhea

- 512 node commodity cluster
- Available for any Pre- and Post-processing work
 - Grid preparation
 - Visualization
 - Data reduction
 - And more
- Available to all OLCF users

Rhea

- Dual Intel Xeon E5-2650 packages in each node
 - 16 physical cores
 - 32 virtual CPUs via HyperThreading
- 64GB of RAM
- 4X FDR Infiniband interconnect
- Shares the center-wide Atlas filesystem
- Ideal for supporting automatic pre-/post-processing workflows

Cross-System Workflows

- OLCF now provides cross-system batch submission
- For example, Titan jobs can
 - submit viz tasks to Rhea
 - submit transfer tasks to the DTNs
- And vice-versa
 - DTNs can start simulations on Titan after data is unloaded from HPSS
 - Rhea can start simulations on Titan after input decks have been generated

Year in Review: Data Management At OLCF

Suzanne Parete-Koon

Oak Ridge Leadership Computing Facility



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

 **OAK RIDGE**
National Laboratory

Filesystem Upgrade

- The Spider center-wide filesystem was upgraded in January.
 - 32 petabytes of disk space. Aggregate speed of 1 TB/s, however, this bandwidth is shared so the effective speed is much lower per user.
- After the initial upgrade users may have noticed sluggish performance
- These problems have largely been mitigated, by moving Titan's Lustre client back to version 1.8.6, and by patches to the Lustre server however, if you continue to see issues please report them to help@olcf.ornl.gov
- The 1.8 Lustre clients running on Titan do not support stripe counts greater than 160.

Directory Structure for 2014

	Member Work	Project Work	World Work
Description	Scratch area	Scratch Area for Sharing data within a project	Scratch Area for sharing data between projects.
Location	\$MEMBERWORK	\$PROJWORK	\$WORLDWORK
Quota	10 TB	100 TB	10TB
Purge	14 days	90 days	14 days
Access	May alter permissions to share with project	All project members have access	All OLCF users can access

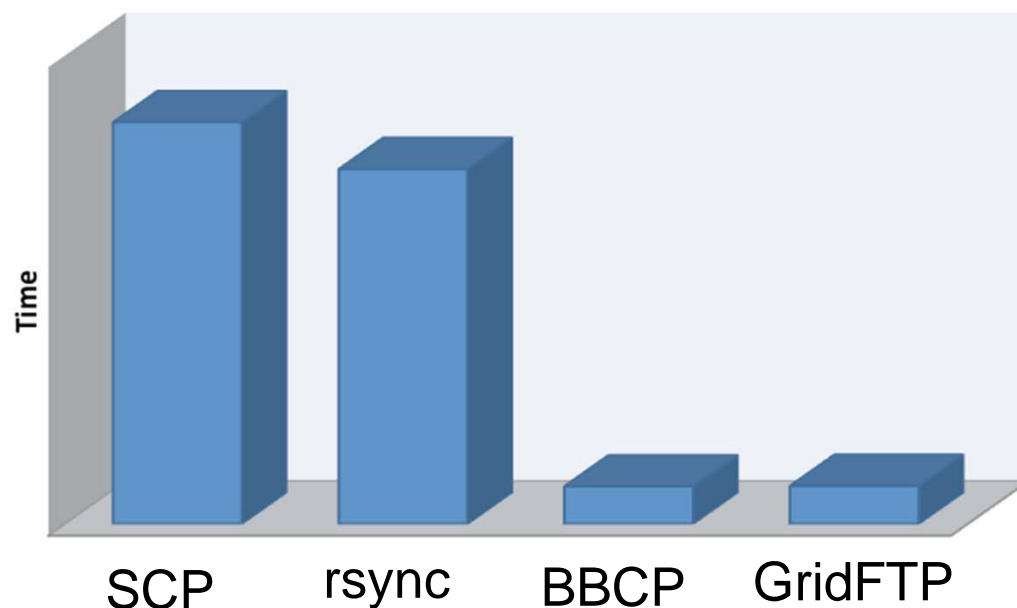
Data Transfer Nodes

- Data transfer nodes should be used for all remote and local data transfers.
- 2 interactive data transfer nodes
 - dtn.ccs.ornl.gov
- 14 Batch scheduled data transfer nodes
 - 2 batch nodes dedicated to hpss transfers.
 - nodes do not suffer from contention
 - dtn jobs may be launched from Titan, Rhea or Eos.

Data Transfer Software

New in 2014: A Supported Globus Endpoint `olcf#dtn`

- Globus and `globus_url_copy` require an Open Science Grid Certificate
- Speeds for tools:



4 parallel streams:

- `bbcp -s4`
- `GridFTP -p4`

Data Management Users Guide

- We have organized a New Data Management User Guide
 - Data management policy
 - Directory Structures of the filesystems
 - Data transfer

Look for this icon on the systems guide page:

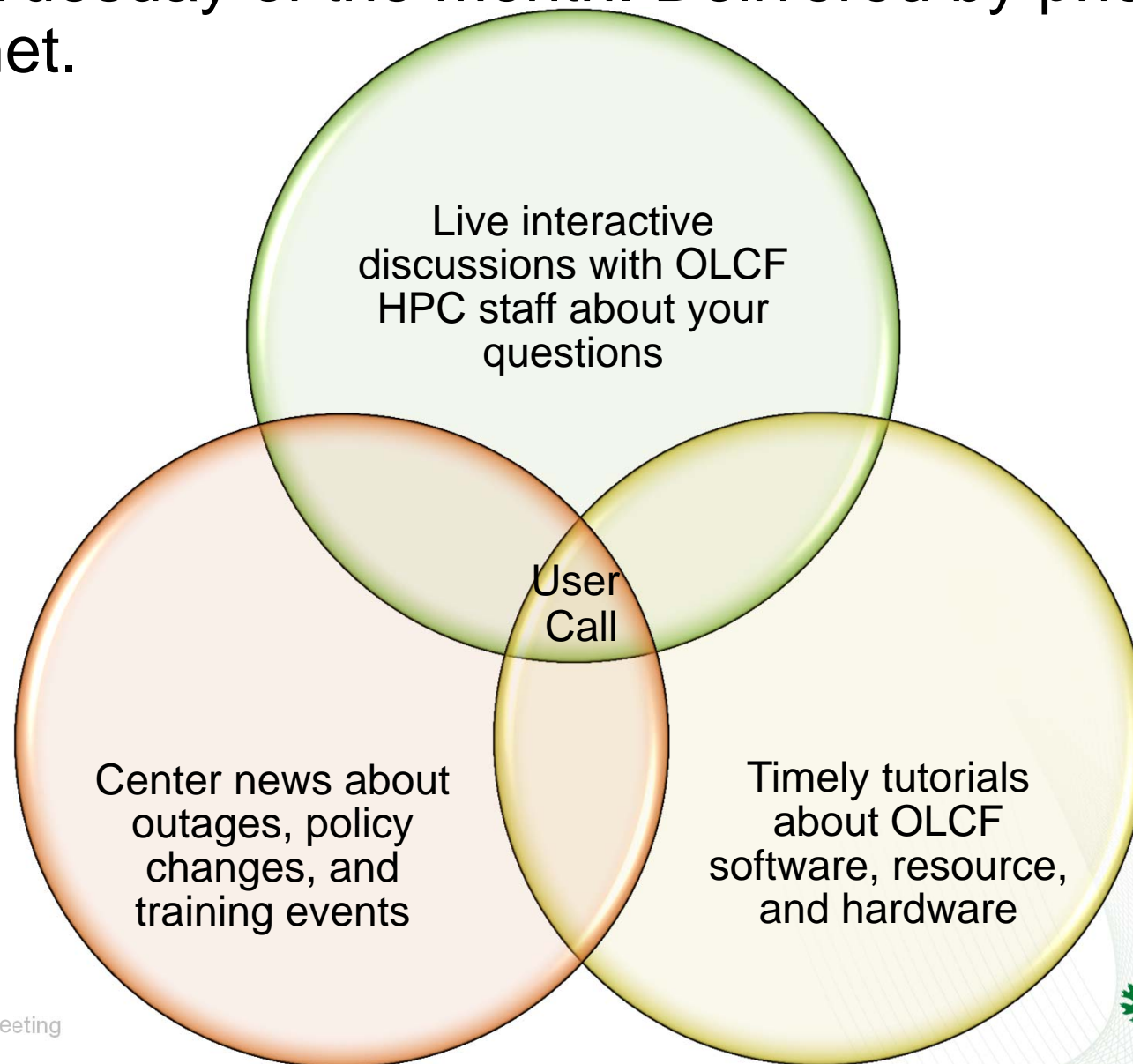


HPSS Improvements

- In the next year the OLCF will bring many new resources to the HPSS archive
 - The addition of a larger disk cache will mean fewer retrievals from tape
 - 40 Gigabit/s Ethernet will speed up staging from tape to disk cache
 - New tape drive technology will speed reads and writes to tape
- Upcoming upgrades to HPSS software will increase performance to the database which should help storing and retrieving files

OLCF User Call

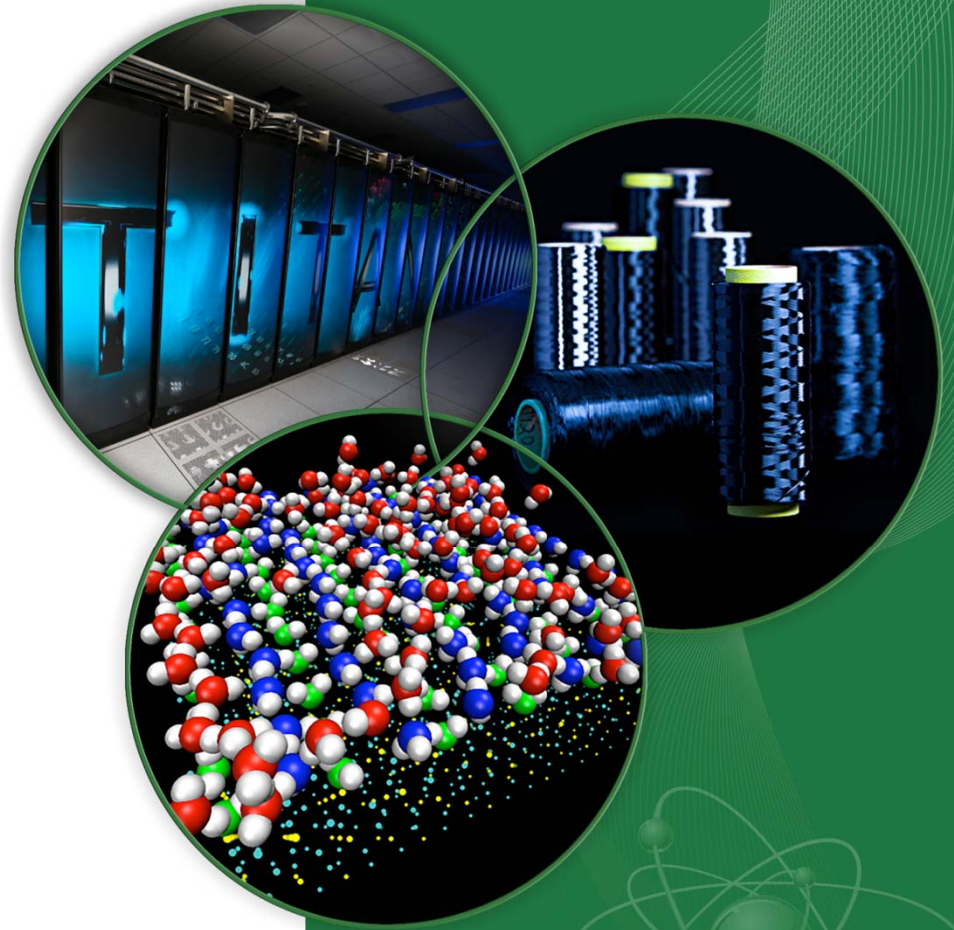
- First Tuesday of the Month. Delivered by phone and Internet.



Year in Review: Accelerator Support

Adam Simpson

Oak Ridge Leadership Computing Facility

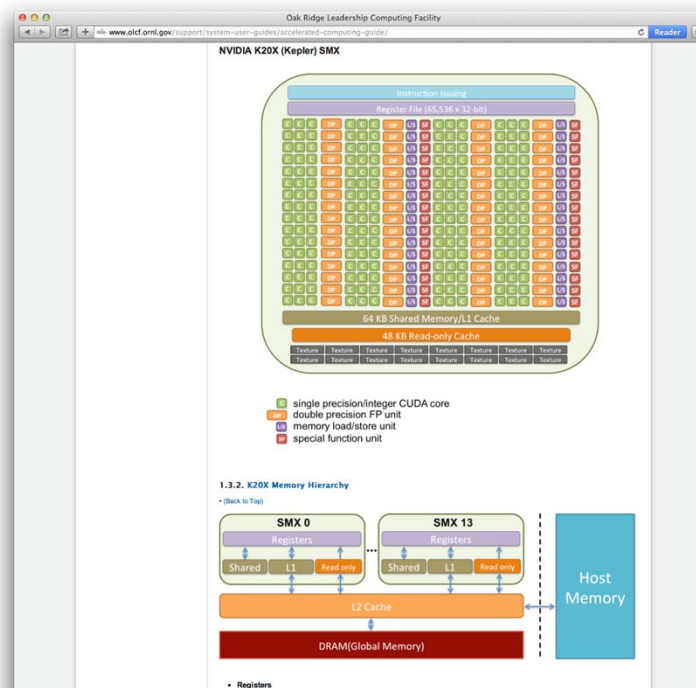


ORNL is managed by UT-Battelle
for the US Department of Energy

 **OAK RIDGE**
National Laboratory

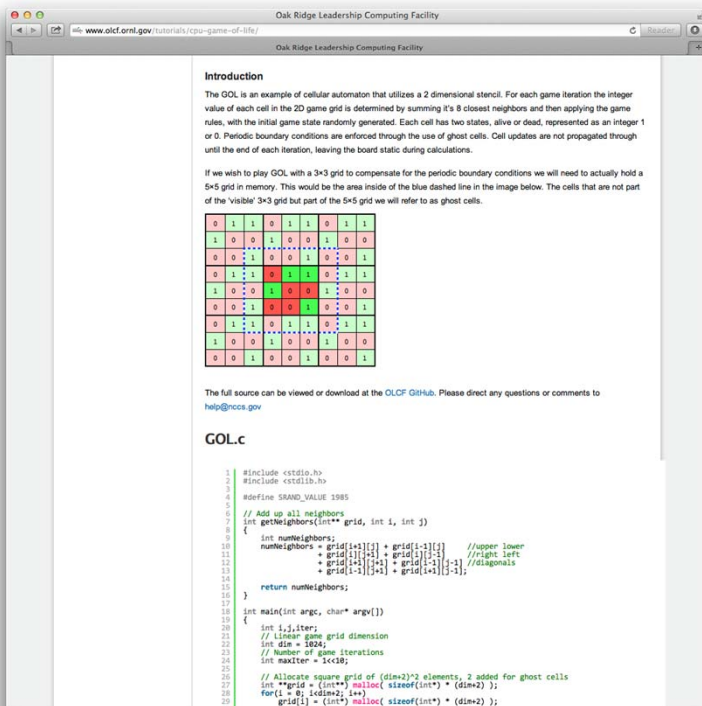
Accelerated Computing User Guide

- User survey indicated preference for online documentation
- Created Accelerated Computing User Guide
 - Comprehensive guide for GPU computing on Titan



Tutorials

- Programing and use examples
 - C and Fortran
 - Highlight HPC specific features and functions
- Several more to be released soon



Introduction

The GOL is an example of cellular automaton that utilizes a 2 dimensional stencil. For each game iteration the integer value of each cell in the 2D game grid is determined by summing it's 8 closest neighbors and then applying the game rules, with the initial game state randomly generated. Each cell has two states, alive or dead, represented as an integer 1 or 0. Periodic boundary conditions are enforced through the use of ghost cells. Cell updates are not propagated through until the end of each iteration, leaving the board static during calculations.

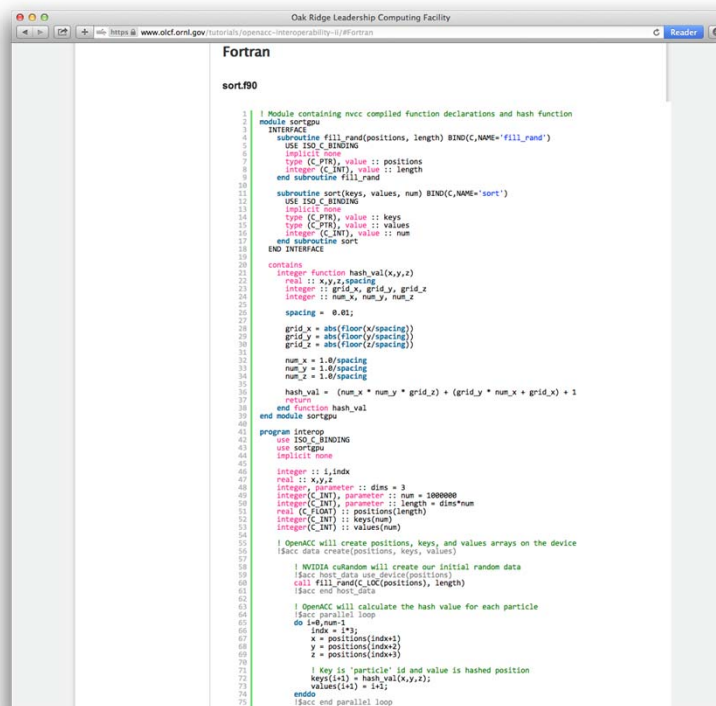
If we wish to play GOL with a 3x3 grid to compensate for the periodic boundary conditions we will need to actually hold a 5x5 grid in memory. This would be the area inside of the blue dashed line in the image below. The cells that are not part of the visible 3x3 grid but part of the 5x5 grid we will refer to as ghost cells.

0	1	0	1	0	1	1
1	0	0	1	0	0	1
0	0	1	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	0	1
1	0	0	1	0	0	0
0	0	1	0	0	1	0
0	1	1	0	1	1	0
1	0	0	1	0	0	0
0	0	1	0	0	1	0

The full source can be viewed or download at the [OLCF GitHub](#). Please direct any questions or comments to help@olcf.gov

GOL.c

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 #define RAND_VALUE 1985
5
6 // Add up all neighbors
7 int getNeighbors(int** grid, int i, int j)
8 {
9     int numNeighbors = 0;
10    numNeighbors = grid[i+1][j] + grid[i-1][j] //upper lower
11    + grid[i][j+1] + grid[i][j-1] //right left
12    + grid[i-1][j+1] + grid[i-1][j-1] //diagonal
13    + grid[i+1][j+1] + grid[i+1][j-1];
14
15    return numNeighbors;
16 }
17
18 int main(int argc, char* argv[])
19 {
20     int i,j,iter;
21     // Linear game grid dimension
22     int dim = 1024;
23     // Number of game iterations
24     int maxiter = 10000;
25
26     // Allocate square grid of (dim+2)*2 elements, 2 added for ghost cells
27     int **grid = (int**) malloc( sizeof(int*) * (dim+2) );
28     for(i = 0; i<dim+2; i++)
29         grid[i] = (int*) malloc( sizeof(int*) * (dim+2) );
```



Fortran

sort.f90

```
1 ! Module containing nvcc compiled function declarations and hash function
2 module sortgpu
3 interface
4     subroutine fill_rand(positions, length) BIND(C,NAME='fill_rand')
5     use ISO_C_BINDING
6     type(C_PTR), value :: positions
7     integer(C_INT), value :: length
8 end subroutine fill_rand
9
10 subroutine sort(keys, values, num) BIND(C,NAME='sort')
11 use ISO_C_BINDING
12 implicit none
13 type(C_PTR), value :: keys
14 type(C_PTR), value :: values
15 integer(C_INT), value :: num
16 end subroutine sort
17 END INTERFACE
18
19 contains
20 integer function hash_val(x,y,z)
21     real :: x,y,z,spacing
22     integer :: grid_x, grid_y, grid_z
23     integer :: num_x, num_y, num_z
24
25     spacing = 0.01
26
27     grid_x = abs(floor(x/spacing))
28     grid_y = abs(floor(y/spacing))
29     grid_z = abs(floor(z/spacing))
30
31     num_x = 1.0/spacing
32     num_y = 1.0/spacing
33     num_z = 1.0/spacing
34
35     hash_val = (num_x * num_y * grid_x) + (grid_y * num_x + grid_x) * 1
36
37     return
38 end function hash_val
39
40 module sortcpu
41
42 program sortcpu
43 use ISO_C_BINDING
44 use sortgpu
45 implicit none
46
47 integer :: i,indx
48 real :: x,y,z
49 integer, parameter :: dims = 3
50 integer(C_INT), parameter :: num = 1000000
51 integer(C_INT), parameter :: length = dims*num
52 real(C_FLOAT) :: positions(length)
53 integer(C_INT) :: keys(num)
54 integer(C_INT) :: values(num)
55
56 ! OpenACC will create positions, keys, and values arrays on the device
57 !$acc data create(positions, keys, values)
58
59 ! NVIDIA cuRAND will create our initial random data
60 !$acc host_data use_device(positions)
61 call fill_rand(C_LOC(positions), length)
62 !$acc end_host_data
63
64 ! OpenACC will calculate the hash value for each particle
65 !$acc parallel loop
66 do i=0,num-1
67     indx = i*3
68     x = positions(indx+1)
69     y = positions(indx+2)
70     z = positions(indx+3)
71
72     ! key is 'particle' id and value is hashed position
73     keys(i+1) = hash_val(x,y,z)
74     values(i+1) = i+1;
75 enddo
76 !$acc end_parallel loop
```

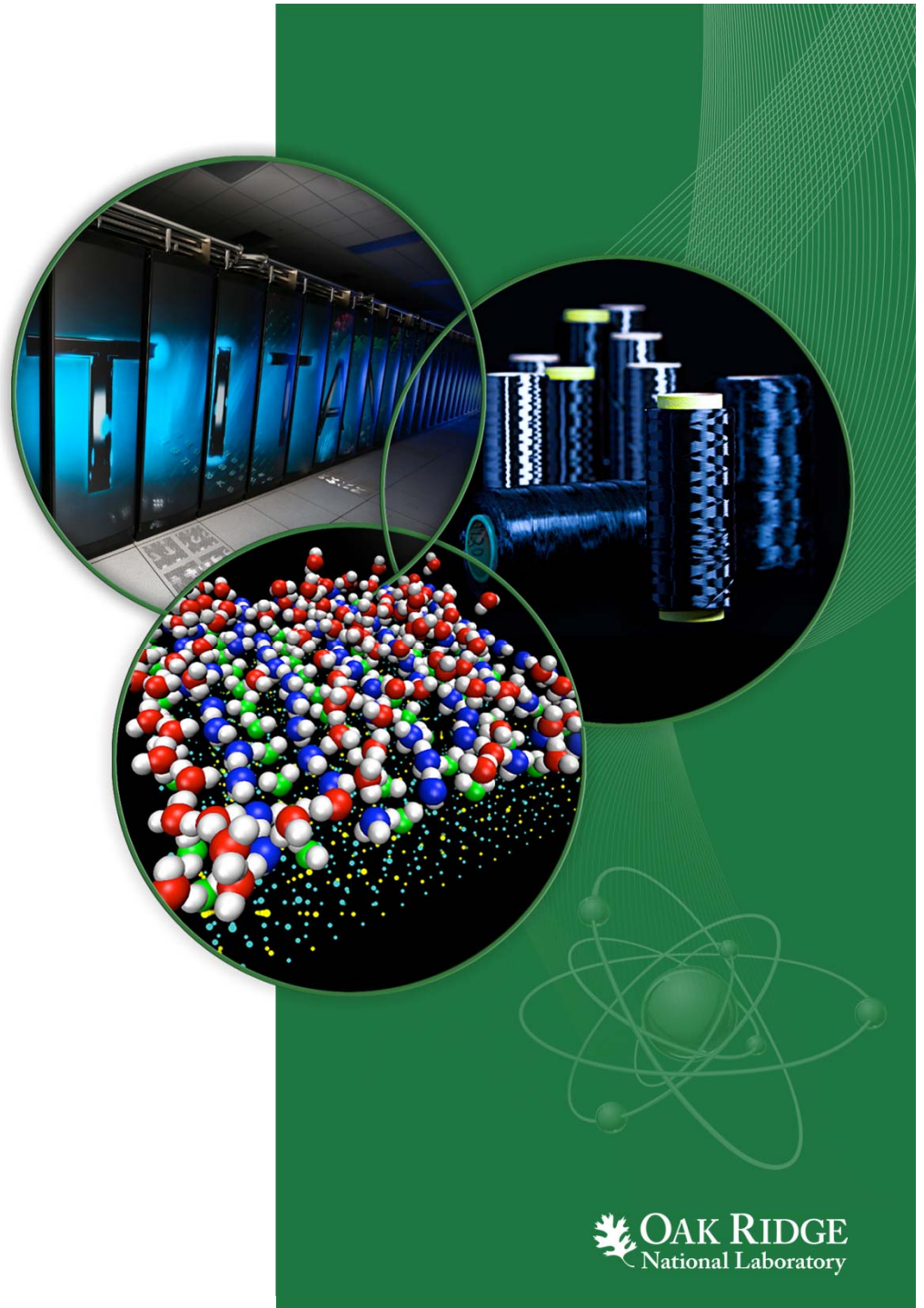
User Feedback

- User feedback determines documentation topics and delivery format
- Must hear from users to provide relevant HPC documentation
- Email any suggestions to help@olcf.ornl.gov

Year in Review: OpenMP/OpenACC and Training Updates

Fernanda Foertter

Oak Ridge Leadership Computing Facility



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

 **OAK RIDGE**
National Laboratory



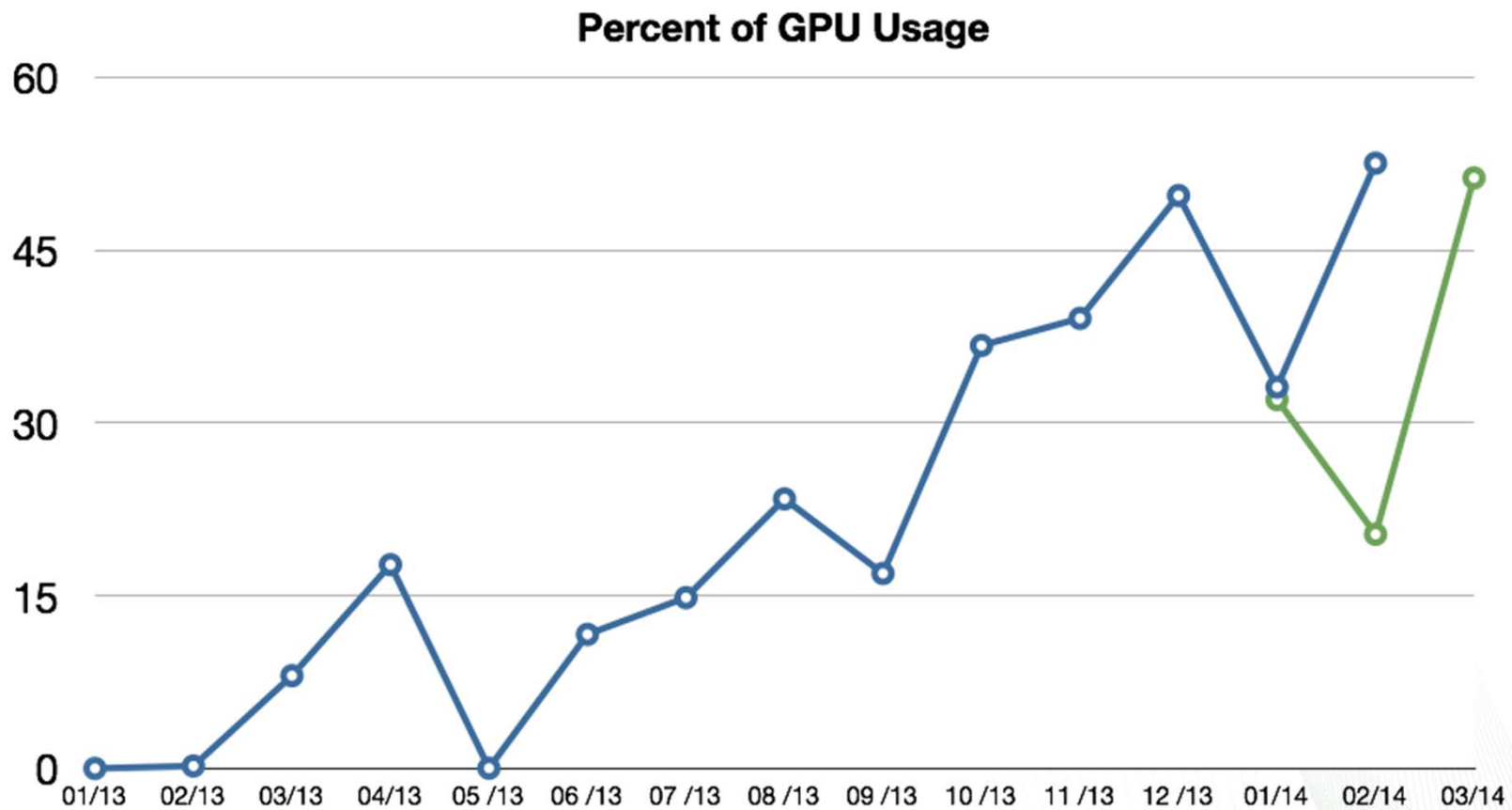
OpenMPTM

OpenACC
Directives for Accelerators

OpenMP and OpenACC activities

- Weekly calls (2h)
 - Recommendation on standards, syntax, features, behavior
- User use cases
- Bugs, behavior
- Communication with compiler developers

Increasing Usage of GPUs



As measured by ALTD against linked libraries

Many of these run on Titan

Growing OpenACC Activity					
Weather / Climate / Ocean	Chemistry	Physics	Oil & Gas	Astronomy	Fluid Dynamics
CAM-SE	CASTEP	CloverLeaf	AWP-ODC	CHIMERA	FVCF-NIP
COSMO (Physics)	CoMD	GENE	EMGS ELAN	IUmd	HemeLB
DYNAMICO	GAMESS CCSD(T)	GTC	MOT-TDVIE	RAMSES	NEK5000
FIM	GAUSSIAN	LULESH	Seismic CPML	X-ECHO	PALM GPU
Harmonie	MiniMD	MiniGHOST	SEISMO		UPACS
HBM	ONETEP	S3D	TeraP		ZFS
ICON	Quantum Espresso				
NICAM	SMMP				
NEMO GYRE					
NIM					
NTSU Snow Simulator					
OLAM					
PALM-GPU					
ROMS					
SCALE-LES					
WRF					

OpenACC
Directives for Accelerators

HPC User Assistance Team at OLCF

The screenshot shows a web browser window with the URL <https://www.olcf.ornl.gov/support/training-events/>. The page header includes the Oak Ridge National Laboratory logo and navigation links. A search bar is present with the text "Search OLCF.ORNL.GOV". Below the header is a navigation menu with links: HOME, ABOUT OLCF, LEADERSHIP SCIENCE, COMPUTING RESOURCES, CENTER PROJECTS, USER SUPPORT (highlighted), MEDIA CENTER, and SC13. A status bar shows the status of various systems: titan, eos, rhea, and hpss, all with green status indicators and timestamps. The main content area features a yellow banner for the "OLCF User Assistance Center" with contact information: "9am to 5pm EST M-F", "help@olcf.ornl.gov", and "(865) 241-6536". Below this is a sidebar with a list of support resources: Support Overview, Getting Started, System User Guides, KnowledgeBase, Tutorials, Training Events (highlighted), My OLCF, and Software. The main content area displays the "Training Events" page, which includes a breadcrumb trail: Home > User Support > Training Events. The page title is "Training Events" and the description is "Archives of training material presented at on-site or tele-conference OLCF trainings." Below this is a section titled "Upcoming Events" with a table listing events.

Event and Venue	Date and Time
2014 OLCF Users Meeting and Getting Started at OLCF Tutorial ORNL Conference Center, Tennessee A (Room 202A), Bldg. 5200	Jul 22, 2014 - Jul 24, 2014 8:00 pm to 3:00 pm

Training Focus This Past Year

- More webcasts
- More specific topics
- More in-depth topics
- Shorter topics
- Repository (github)
- Open to the public

To come: short screencasts

Upcoming Training Activities

OpenACC Hack-a-thon

Oct 27, 2014 - Oct 30, 2014

Compiler Directives Weekly Lunch Webinars

Bldg. 5200, Room 219

Jun 9, 2014 - Jun 9, 2014

12:00 pm to 1:30 pm

OpenACC Tutorials

JICS Auditorium, Building 5100

Jul 15, 2013 - Jul 18, 2013

9:00 am to 2:00 pm



- Fernanda Foertter
- ccs-training@email.ornl.gov
@hpcprogrammer
(unofficial!)

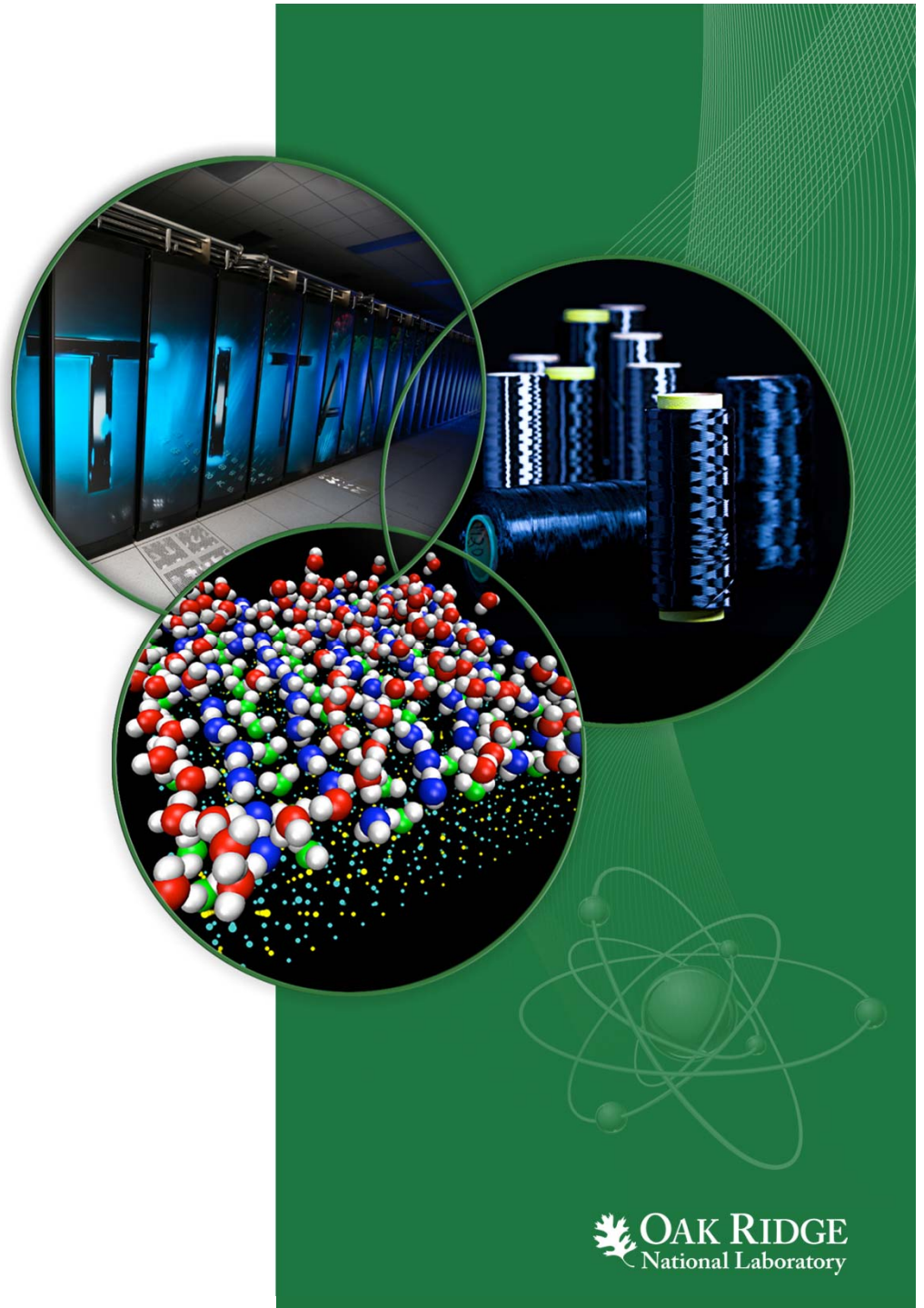
 OAK RIDGE
National Laboratory

Year in Review: Software - Tools

Chris Fuson

Oak Ridge Leadership
Computing Facility

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



 **OAK RIDGE**
National Laboratory

Vendors and Software Developers

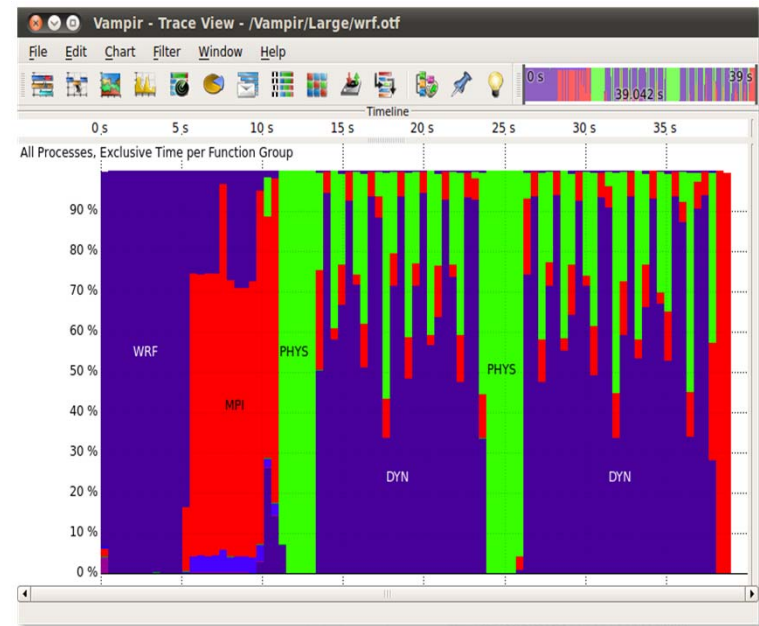
- Accounts on Titan and internal development systems
- Allows and speeds the process
 - verify
 - build
 - system specific features
 - work bugs
 - support
- Examples
 - **Nick Forrington**
 - *DDT*
 - **Frank Winkler**
 - *Vampir*
 - **Scott Atchley**
 - *aprun layout*

New Features in DDT

- Logbook view
 - Logs the user's interaction with DDT (e.g. stacks, locals, tracepoints)
 - Enabling comparative debugging and repeatability.
- Version control integration (Git / Mercurial / Subversion)
 - Shows annotation information against source code
 - Add breakpoints and tracepoints to track changes in a particular revision
- VisIt
 - Visual preview of how multi-dimensional & distributed arrays will be displayed.
 - Visualize multiple arrays at a vispoint
- More information in user guide
 - Press F1 while running DDT
 - <http://content.allinea.com/downloads/userguide.pdf>

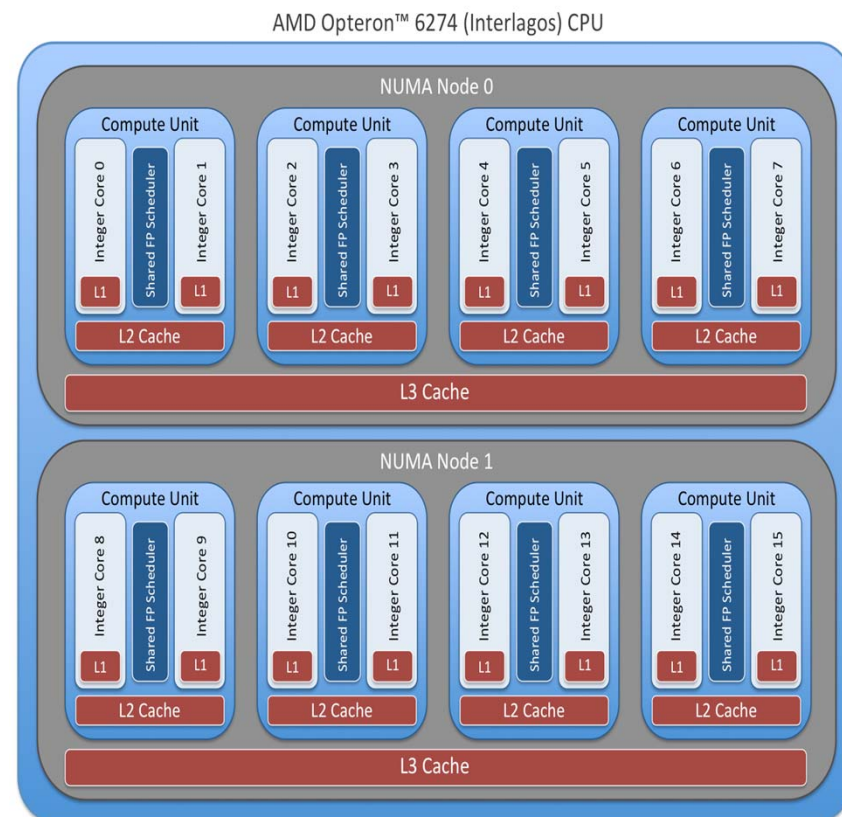
New Features in Vampir

- Score-P (Successor of VampirTrace)
 - Common instrumentation and measurement infrastructure for parallel codes
 - Supports a number of performance analysis tools
 - Periscope, Tau, Cube, Scalasca, Vampir
 - Available on Titan, Eos and Rhea
 - module load scorep
- Vampir 8.3.0
 - New Summary Timeline chart
 - Available on Titan, Eos and Rhea
 - module load vampir
- www.olcf.ornl.gov/kb_articles/software-vampir/
- www.olcf.ornl.gov/kb_articles/software-scorep/



Aprun - Floating-Point Contention

- Titan Node
 - Each Bulldozer compute unit:
 - 2 Integer Cores
 - 1 Floating Point Unit (FPU)
- By default, aprun will:
 - place 16 processes per node
 - place the processes sequentially starting at core 0
- Floating point intensive codes
 - Can see 2x speed-up by not sharing FPU
 - `$ aprun -n8 -S4 -j1 ./a.out`



Aprun - Floating-Point Contention

- Noticed many codes requesting partial nodes but not spreading tasks ideally

\$ aprun -n8 ./a.out

NUMA 0								NUMA 1							
Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7	Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7
0	1	2	3	4	5	6	7								

NUMA 0								NUMA 1							
Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7	Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7
0	1	2	3					4	5	6	7				

\$ aprun -n8 -S8 ./a.out

- One task per compute unit: **-j1**

\$ aprun -n8 -S4 -j1 ./a.out

NUMA 0								NUMA 1							
Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7	Core 0	Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7
0		1		2		3		4		5		6		7	

Aprun - Floating-Point Contention

- Wrap aprun to check partial node requests
- Warning message:

```
$ aprun -n32 -S4 ./a.out
```

```
-----  
APRUN usage: requested less processes than cores (-S 4) without  
using -j 1 to avoid floating-point unit contention
```

- Aprun still executed, only message
 - Set the environment variable APRUN_USAGE_QUIET to suppress these messages.
- www.olcf.ornl.gov/support/system-user-guides/titan-user-guide/#2941