Application Requirements for Next-Generation OLCF Systems

Wayne Joubert
Scientific Computing Group
Oak Ridge National Laboratory

2014 OLCF User Meeting

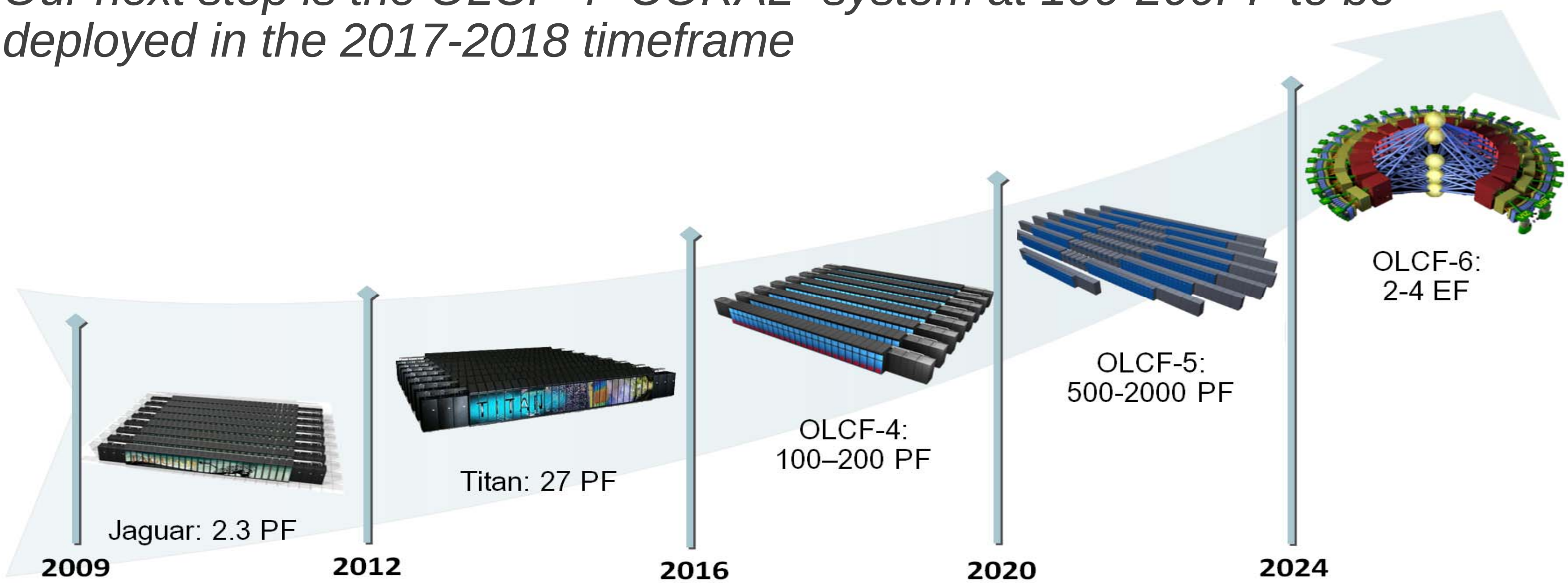
OLCF 2024 Roadmap

We have increased our systems capability at our center since 2004 by a factor of 10,000X

We are currently at 27PF, however the Exascale reports have pointed out that systems will be required at the level of 1 EF and beyond in order to reach critical DOE science goals

Our facilities plan therefore is to deploy a system of up to 4 EF capability by 2024

Our next step is the OLCF-4 “CORAL” system at 100-200PF to be deployed in the 2017-2018 timeframe



Application Requirements

DOE Agency Priority Goal: "Identify programmatic drivers and technical requirements in coordination with other Departmental mission areas to inform future development of high performance computing capabilities and in anticipation of capable exascale systems ." (***DOE Strategic Plan 2014-2018***)

To help us understand the compute capabilities needed for future systems from the standpoint of applications, we periodically go through an applications requirements process

https://www.olcf.ornl.gov/media-center/center-reports

The Requirements Process

- We take input from multiple sources:
  - DOE science mandates
  - OLCF Leadership Computing usage statistics for the recent past to understand how users are using our systems
  - A requirements survey to elicit project requirements from OLCF-supported projects
  - Discussions with computer hardware and software vendors on capabilities of next-generation offerings
- We seek to understand items/specifications needed in order for science teams to meet their science objectives, for example:

system hardware requirements	science model requirements
system software requirements	algorithm requirements
compilers, libraries, tools	application codes
data storage, access, analysis	development processes
workflow management	

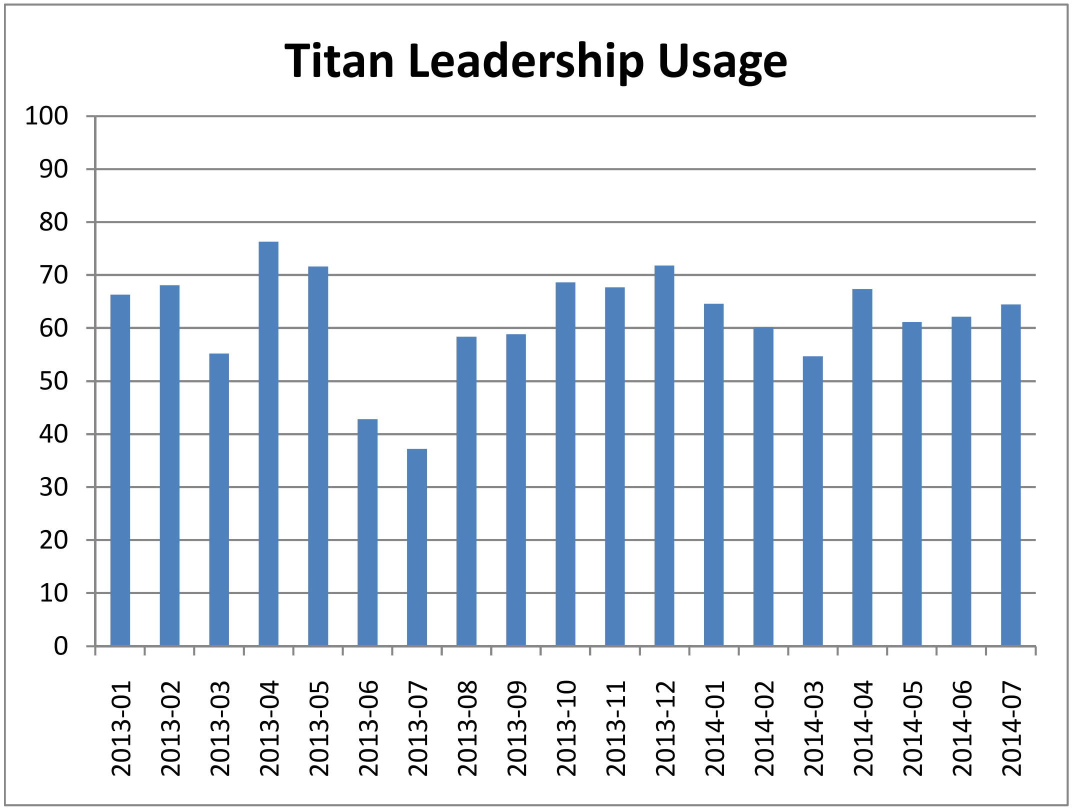
1. OLCF System Usage

System logs tell us the demand to run large jobs continues to be high and growing

The "leadership metric": what fraction of our core-hours are used in jobs requiring >20% of the full system

The average for Titan has been 62% YTD

Up from 43% for JaguarPF over its lifetime



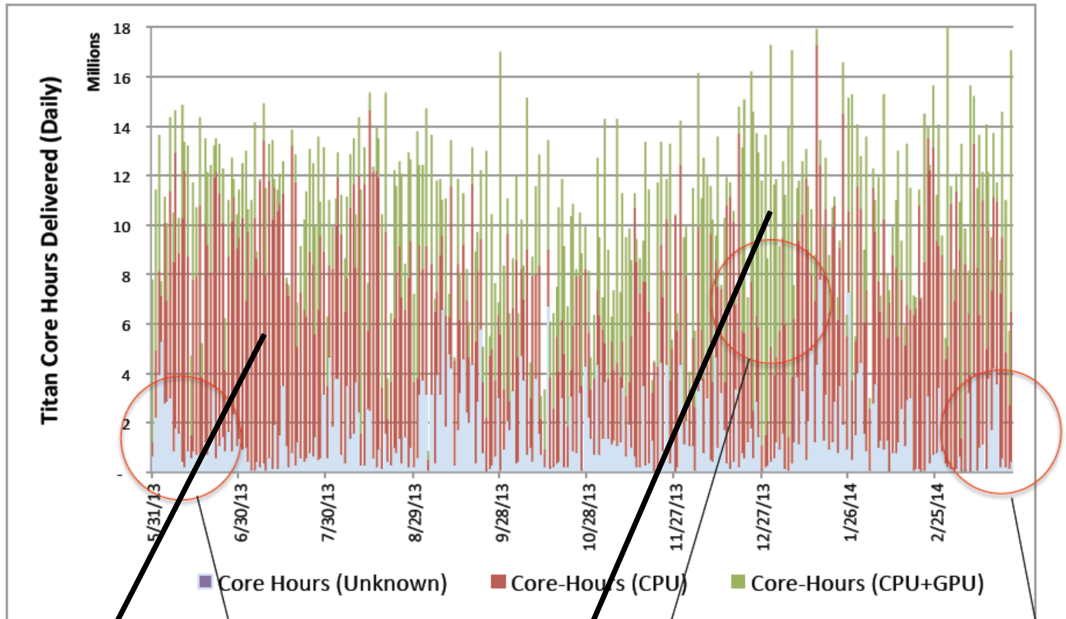
Titan GPU Usage

Many projects are making effective use of accelerated heterogeneous nodes on Titan

We are improving our tools for tracking GPU usage

Usage is substantial and growing

Assessing GPU Usage with ALTD



Rocky start using ALTD… lots of edge cases escaped.

Great *apparent* use of the GPU by the workflow, but no way to quantify it.

Unknowns are 14% of total delivered hours since May 31, 2013.

Slide courtesy Jim Rogers / GTC 2014

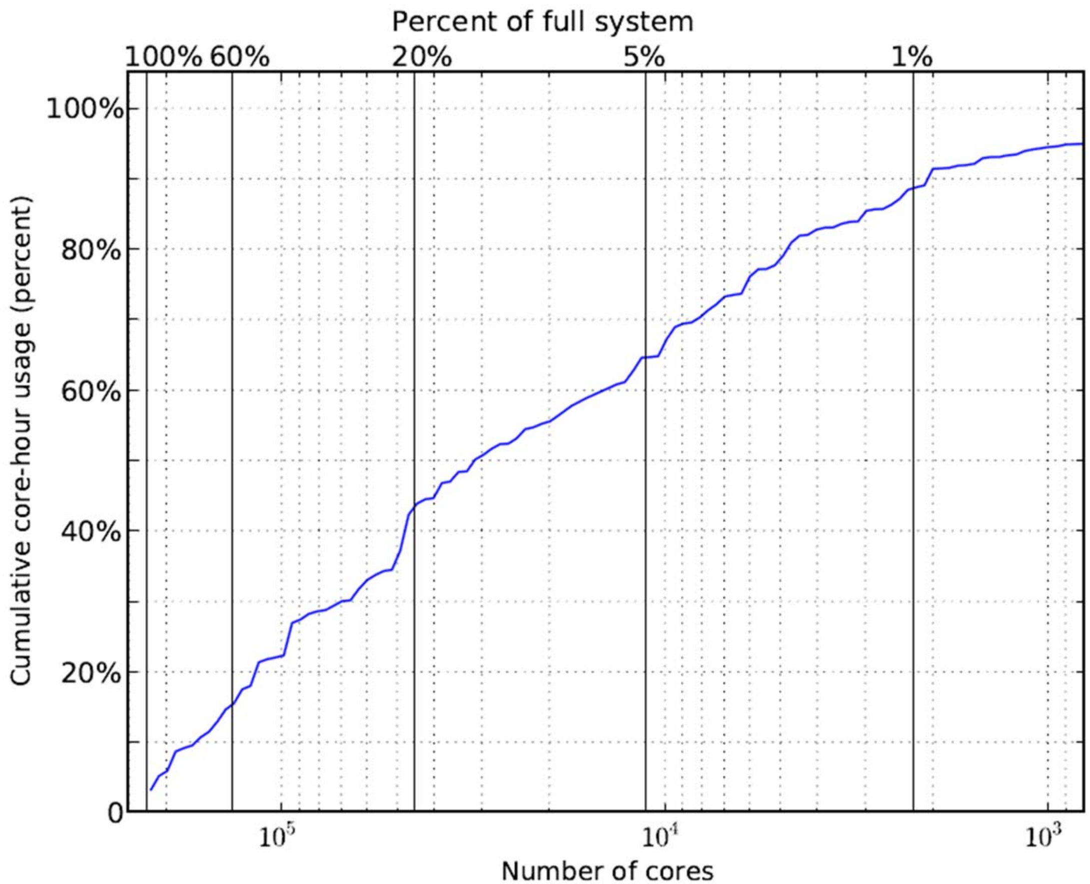
Red: CPU usage

Green: CPU+GPU usage

Average Job Size - JaguarPF

Users need to be able to run jobs of all sizes

Cumulative graph shows the distribution of how core-hours are used with respect to job size



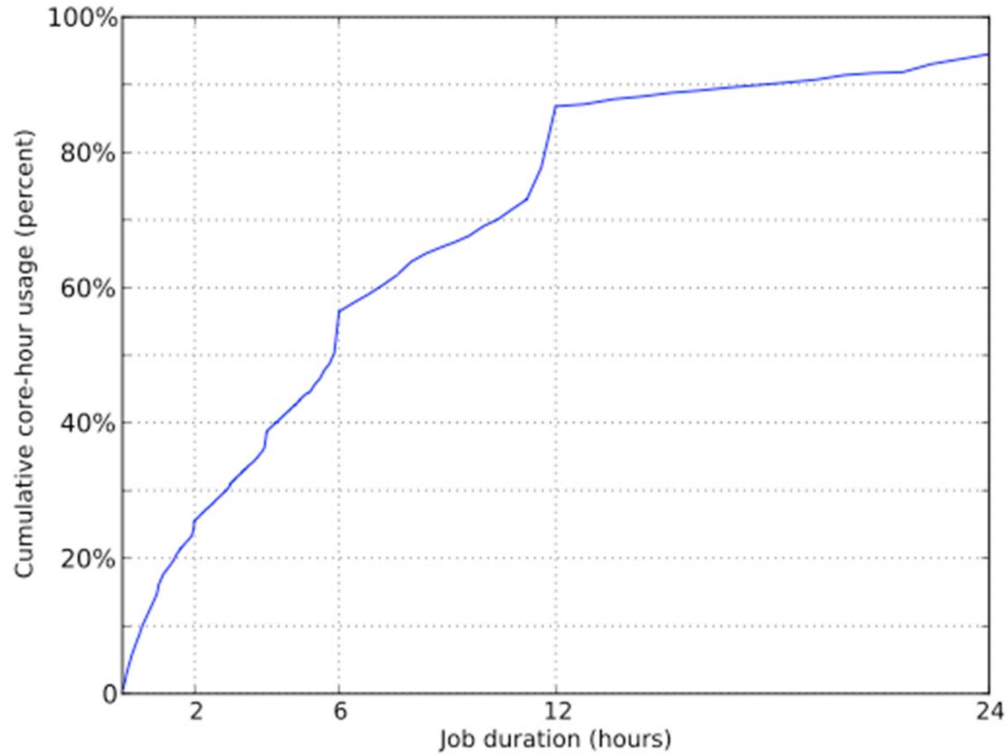
Average Job Duration - JaguarPF

Users have learned how to limit their job sizes, using e.g. checkpoint/restart when necessary

Cumulative graph showing the distribution of how core-hours are used with respect to job duration

88% of core-hours spent on jobs < 12 hours

Half of core-hours spent on jobs of < 6 hours



Algorithmic Motifs

Application	Structured Grids	Unstructured Grids	FFT	Dense Linear Algebra	Sparse Linear Algebra	Particles	Monte Carlo
NWCHEM			X	X			
S3D	X			X	X	X	
XGC		X				X	
CCSM	X		X		X		
CASINO							X
VPIC	X					X	X
VASP			X	X			
MFDn					X		
LSMS				X			X
GenASiS		X			X		
MADNESS		X	X	X			
GTC	X				X	X	X
OMEN	X				X		
Denovo	X			X	X	X	X
CP2K	X				X	X	
CHIMERA	X			X	X	X	
DCA++				X			X
LAMMPS	X		X			X	
DNS	X			X	X	X	
PFLOTRAN	X	X		X	X		X
CAM	X		X	X	X	X	
QMCPACK						X	X
TOTALS:	12	4	6	11	12	11	8

Top-used codes span the whole range of algorithmic motifs

2. 2013 Requirements Survey

- Sent out survey in spring 2013 to OLCF project teams
- Received responses from 21 code teams representing 18 projects across 15 science domains
- This was a large sample of the 31 INCITE projects representing 60% of resources allocated on OLCF systems.
- Science teams reaffirmed that they will not be able to meet their science goals at all or as fast without deployment of a system with significantly more capability than 30PF.

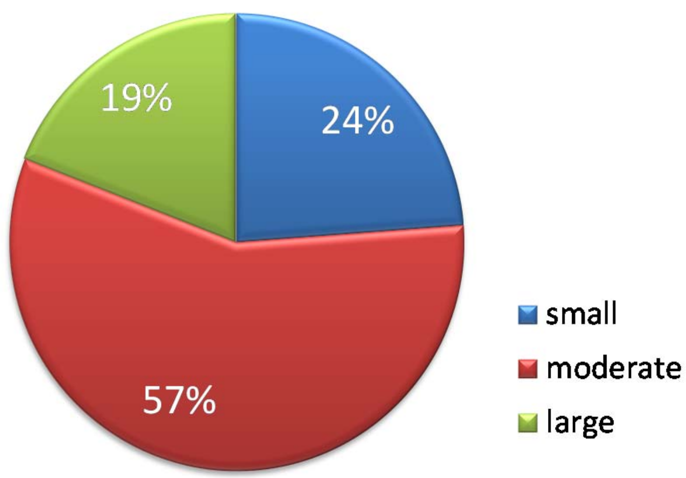
Hardware Feature Requirements

- Users were asked to rank the relative importance of hardware features for the next system
- In the past "more flops" has been the most important
- Now users are starting to feel the need for more memory bandwidth as well

Hardware feature	Ranking	Hardware feature	Ranking
Memory bandwidth	4.4	Wan network bandwidth	3.7
Flops	4.0	Memory latency	3.5
Interconnect bandwidth	3.9	Local storage capacity	3.5
Archival storage capacity	3.8	Memory capacity	3.2
Interconnect latency	3.7	Mean time to interrupt	3.0
Disk bandwidth	3.7	Disk latency	2.9

Additional Parallelism in Applications

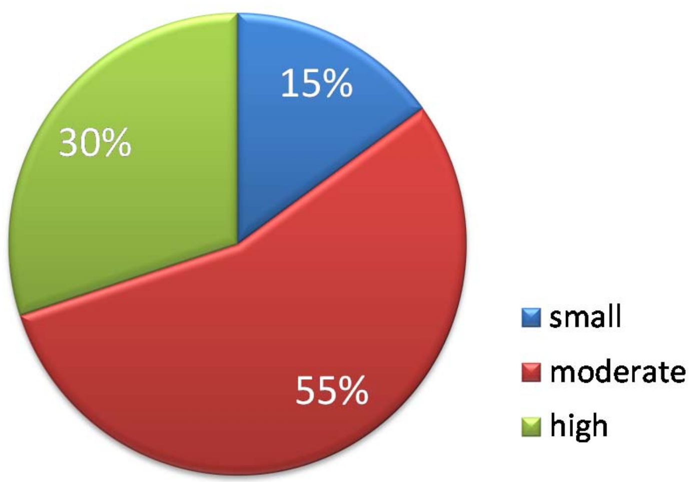
- Most users feel that they still have significant additional parallelism that can be extracted from their applications for future systems
- They are using a wide range of tools to access this parallelism, e.g., MPI, OpenMP, CUDA, OpenACC, libraries, OpenCL and Pthreads



Additional available parallelism in application codes, by number of respondents

Difficulty Exploiting Advanced Hardware

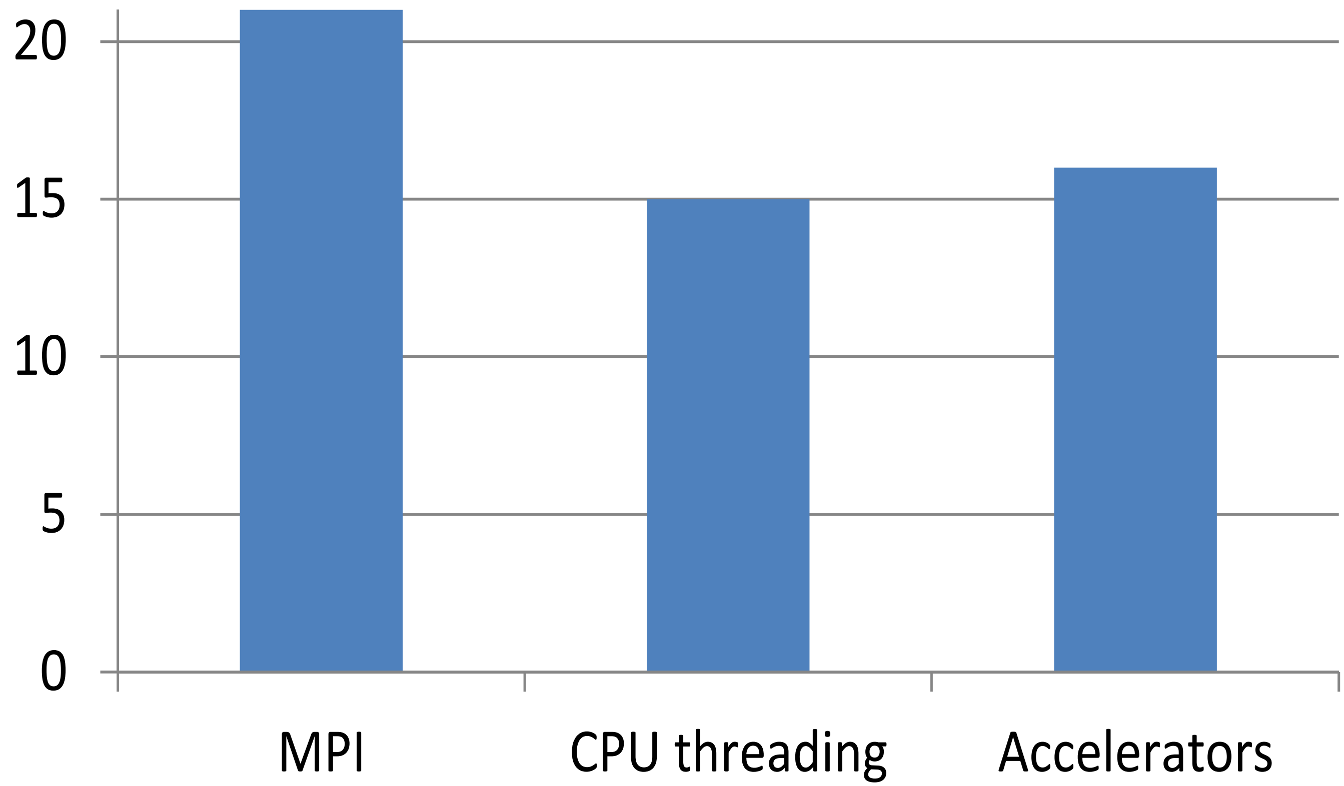
- Users are for the most part extraordinarily willing to exploit advanced hardware—to “do whatever it takes”
- However, they expressed concerns about lack of performance portability because of differing hardware and programming APIs and immaturity of some programming models



Assessment of difficulty in exploiting advanced hardware, by number of respondents

Adoption of Levels of Parallelism

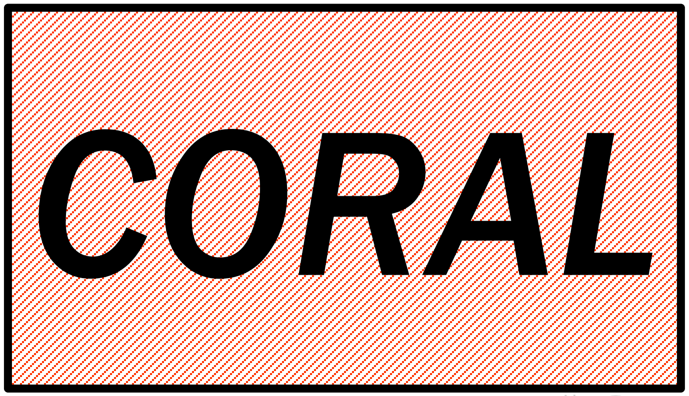
- Most users surveyed indicated they had already adopted some form of node-level threading, either for CPU or accelerator



Current code levels of parallelism, by number of respondents.

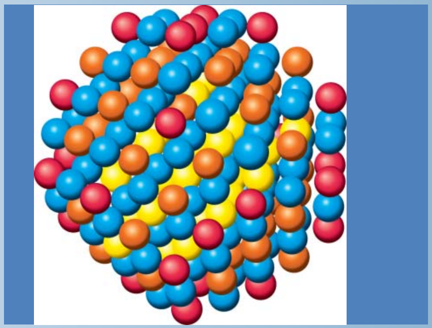
3. Application Readiness

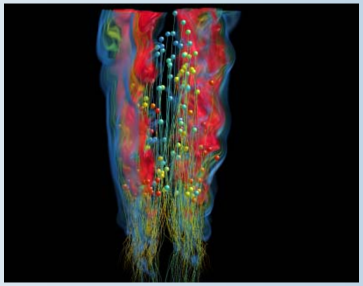
- We went though an application readiness process for Titan (CAAR, Center for Accelerated Applications Readiness)
- We expect to have another readiness process for the OLCF-4 CORAL system
- We believe the lessons learned from the CAAR effort will help us understand requirements for CAAR-2



Center for Accelerated Application Readiness (CAAR)

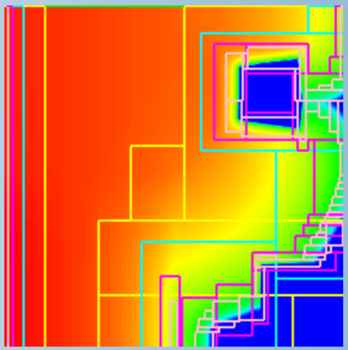
WL-LSMS
Illuminating the role of material disorder, statistics, and fluctuations in nanoscale materials and systems.

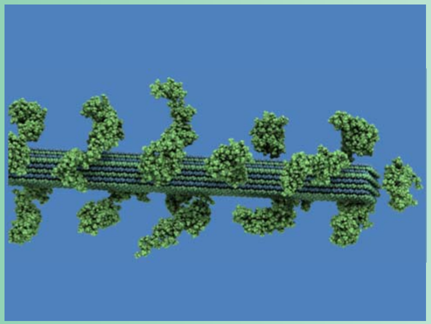




S3D
Understanding turbulent combustion through direct numerical simulation with complex chemistry.
.

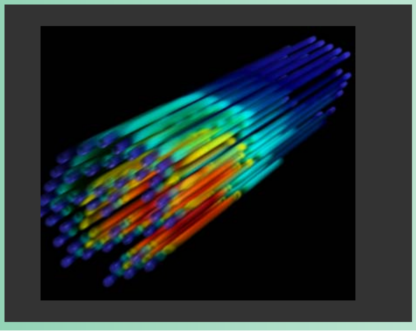
NRDF
Radiation transport – important in astrophysics, laser fusion, combustion, atmospheric dynamics, and medical imaging – computed on AMR grids.





LAMMPS
A molecular description of membrane fusion, one of the most common ways for molecules to enter or exit living cells.

CAM-SE
Answering questions about specific climate change adaptation and mitigation scenarios; realistically represent features like precipitation patterns / statistics and tropical storms.



Denovo
Discrete ordinates radiation transport calculations that can be used in a variety of nuclear energy and technology applications.

Slides courtesy Bronson Messer

Application characteristics inventory

App	Science Area	Algorithm(s)	Grid type	Programming Language(s)	Compiler(s) supported	Approx. LOC	Communication Libraries	Math Libraries
CAM-SE	climate	spectral finite elements, dense & sparse linear algebra, particles	structured	F90	PGI, Lahey, IBM	500K	MPI	Trilinos
LAMMPS	Biology / materials	molecular dynamics, FFT, particles	N/A	C++	GNU, PGI, IBM, Intel	140K	MPI	FFTW
S3D	combustion	Navier-Stokes, finite diff, dense & sparse linear algebra, particles	structured	F77, F90	PGI	10K	MPI	None
Denovo	nuclear energy	wavefront sweep, GMRES	structured	C++, Fortran, Python	GNU, PGI, Cray, Intel	46K	MPI	Trilinos, LAPACK, SuperLU, Metis
WL-LSMS	nanoscience	density functional theory, Monte Carlo	N/A	F77, F90, C, C++	PGI, GNU	70K	MPI	LAPACK (`ZGEMM`, `ZGTRF`, `ZGTRS`)
NRDF	radiation transport	Non-equilibrium radiation diffusion equation	structured AMR	C++, C, F77	PGI, GNU, Intel	500K	MPI, SAMRAI	BLAS, PETSc, Hypre, SAMRSolvers

Code Changes for Titan

Application	API	Code Modifications
WL-LSMS	CUDA, library	• Rewrote code (LSMS_3) to allow more flexible node parallelism • Used BLAS3 functions from library and custom code
CAM-SE	CUDA Fortran	• Used new chemistry package with more parallelism • Fused element loops • Flattened data structures
S3D	OpenACC	• Permuted loops across code to expose coarse-grain parallelism
LAMMPS	CUDA, OpenCL	• Ported short-range force and other calculations to GPU • Replaced FFTs with MSM for long-range forces
Denovo	CUDA	• Implemented new algorithm to expose a new axis of parallelism • Restructured sweep kernel for data locality and more threading

Some Lessons Learned

- Projects were successful obtaining substantial code performance improvements using the GPUs
- 70-80% of developer time was spent in code restructuring, regardless of the parallel API used. Because of this, we feel confident that the porting effort will make it much easier to port these codes to OLCF-4.
- Codes need as much lead time as possible to prepare for a substantially different computing system

Some Lessons Learned

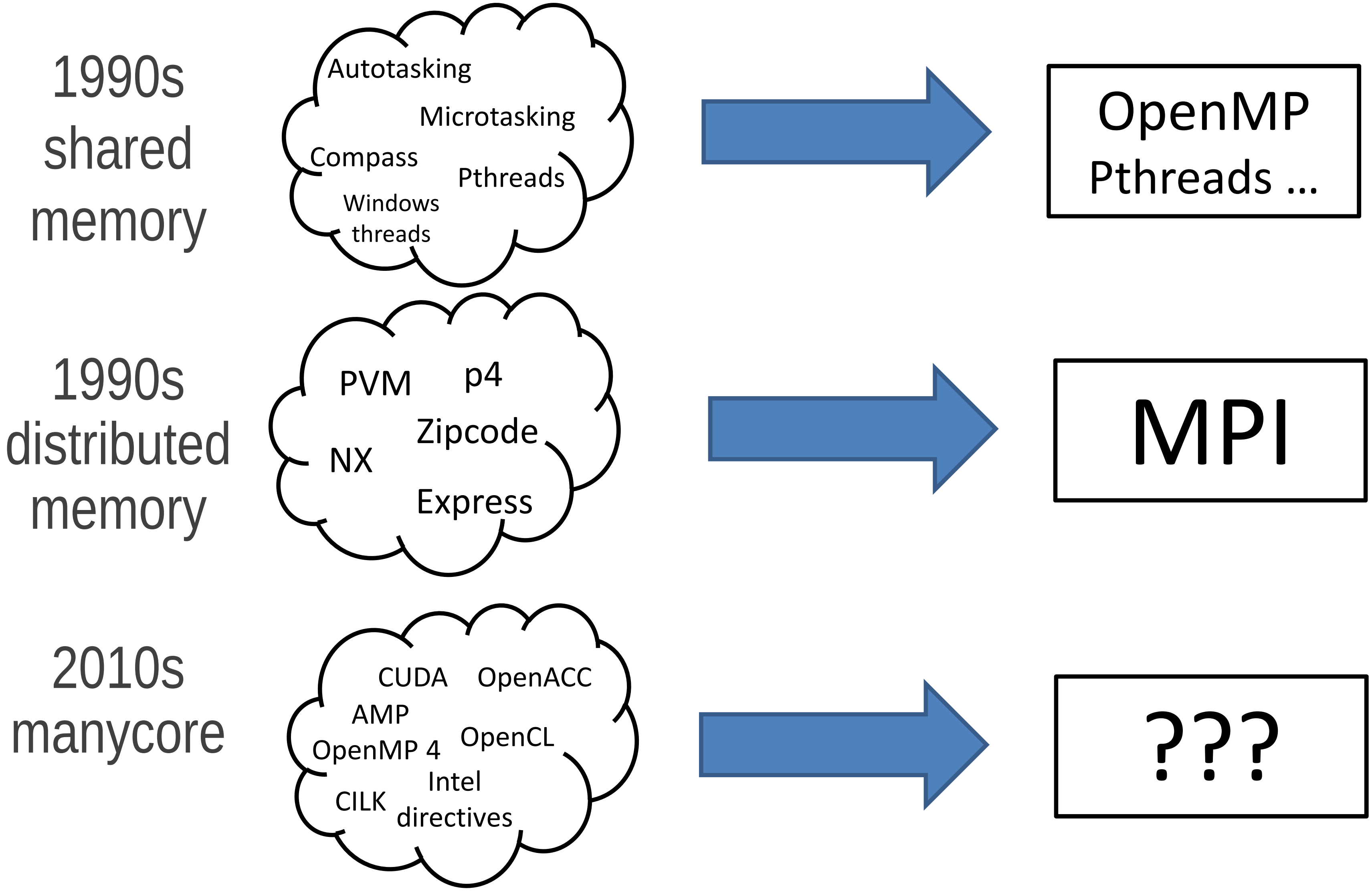
- Some codes are more easy to port than others, depending on various factors:
  - Code execution profile—flat or hot spots
  - The code size (LOC)
  - Structure of the algorithms—e.g., available parallelism, high computational intensity
- Since this was a new effort and tools were not mature, the port required 1-3 person-years per code. We expect now for this to be much shorter.

Performance Portability

- The diversity of new kinds of system and programming API is making performance portability a growing concern (NVIDIA, Intel, AMD, …)
- The industry is consolidating into a fairly uniform configuration of hierarchical parallelism in the form of vector/multithreading/core/socket/node/system
- Because of this uniformity, the 70-80% of porting effort spent on code restructuring can be leveraged across these architectures
- However, the lack of consolidation of programming APIs for this hardware makes code porting and maintenance needlessly difficult

API Standardization: Past Experience

Repeated pattern of hardware/API disruption, consolidation, standardization



Programming for Portability

- How codes are dealing with portability at present:
  – Interface to accelerated library with multiplatform support (LSMS)
  – Thin code layer with platform-specific interface (LAMMPS, Denovo)
  – Directives that can be changed for different platforms (S3D)
  – C++ generic programming for composing platform-specific operations (Trilinos)
- What we can do
  – Urge vendors to agree upon performance portable standards and make these well-supported in software
  – In the meantime attempt to design codes defensively to prepare for future changes, e.g., isolate data layout issues from business logic of code

Questions?

Wayne Joubert
joubert@ornl.gov

We acknowledge the assistance of Bronson Messer, Mike Brown, Matt Norman, Markus Eisenbach, Ramanan Sankaran, Fernanda Foertter, Valentine Anantharaj, Jack Wells, Buddy Bland, Ashley Barker and Jim Rogers.

The research and activities described in this presentation were performed using the resources of the Oak Ridge Leadership Computing Facility at Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC0500OR22725.

Supplementary slides

Performance results:

How Effective are GPUs on Scalable Applications?

OLCF-3 Early Science Codes

Very early performance measurements on Titan

	XK7 (w/ K20x) vs. XE6	Cray XK7: K20x GPU plus AMD 6274 CPU Cray XE6: Dual AMD 6274 and no GPU Cray XK6 w/o GPU: Single AMD 6274, no GPU
Application	**Performance Ratio**	**Comments**
S3D	1.8	• Turbulent combustion • **6% of Jaguar workload**
Denovo sweep	3.8	• Sweep kernel of 3D neutron transport for nuclear reactors • **2% of Jaguar workload**
LAMMPS	7.4* (mixed precision)	• High-performance molecular dynamics • **1% of Jaguar workload**
WL-LSMS	3.8	• Statistical mechanics of magnetic materials • **2% of Jaguar workload** • 2009 Gordon Bell Winner
CAM-SE	1.8* (estimate)	• Community atmosphere model • **1% of Jaguar workload**

Slide courtesy Buddy Bland

Coming changes

- New developments announced by vendors
  - Burst buffers – different checkpoint strategies, out-of-core algorithms
  - 3-D stacked memory – changes balance of flop rate / memory speed; different caching structure
  - Self-hosting accelerators – changes how latency-optimized vs. bandwidth-optimized code is combined
- More speculative
  - Application-level resiliency
  - Application-level power usage adaptivity
  - Extreme core counts per chip, e.g., Adapteva 64,000 cores by 2018
  - Intel CPU + FPGA on-die (???)
  - Altera FPGA programmable with OpenCL (???)