

LA-UR-16-24061

Approved for public release; distribution is unlimited.

Title: Parallel Computing Summer Research Internship Overview

Author(s): Robey, Robert W.
Nam, Hai Ah
Schoonover, Joseph Arthur
Garrett, Charles Kristopher
Aguilar Garcia, Nickole A.

Intended for: Report

Issued: 2017-06-12 (rev.1)

Disclaimer:

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By approving this article, the publisher recognizes that the U.S. Government retains nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Parallel Computing Summer Research Internship Overview



**Parallel Computing Summer
Research Institute
Workshop Team***

June 8th, 2017

*Robey, B., Nam, H., Garrett, K.,
Schoonover, J., Garcia, N.


Operated by Los Alamos National Security, LLC for the U.S. Department of Energy's NSA

Who Are We?

- **Workshop Co-Leads**
 - Bob Robey, XCP-2, Eulerian Codes
 - Hai Ah Nam, CCS-2, Computational Physics and Methods
 - Kris Garrett, CCS-2, Computational Physics and Methods
 - Joe Schoonover, CCS-2, Computational Physics and Methods
- **Workshop Coordinator – Nickole Aguilar Garcia**
- **ISTI Director – Stephan Eidenbenz**

Who are you?

- PCSRI Students @ NSEC
 - Jordan Fox
 - Jennifer Soter
 - Jacob Carroll
 - Siddhartha Bishnu
 - Prerna Patil
 - Trokon Johnson
 - Rachel LeCover
 - Kirtus Leyba
 - Nils Carlson
 - Donald Kruse
- PCSRI Affiliate @ NSEC
 - Shane Fogerty
- PCSRI Affiliates
 - Justin Sunu - bldg 200, rm 212 # 10
 - William Rosenberger - bldg 200, rm 212 # 10
 - Jose Navarro - bldg 200, rm 212 #5
 - Robert Martin-Short – EES-16, Otowi across from badge office
 - Daniel Dunning - ???
 - Tim Dunn - Library basement
 - Bryan Kaiser - ???

3 sentences

- Where you are from
- Future Plans
- What you expect to get out of the workshop

State of the “Art”

I might as well flame a bit about my personal unhappiness with the current trend toward multi-core architecture. To me, it looks more or less like the hardware designers have run out of ideas, and that they're trying to pass the blame for the future demise of Moore's Law to the software writers by giving us machines that work faster only on a few key benchmarks! I won't be surprised at all if the whole multi-threading idea turns out to be a flop ...

Torch is passed

Other people understand parallel machines much better than I do; programmers should listen to them, not me, for guidance on how to deal with simultaneity.

Interview with Donald Knuth, 2008, author of **The Art of Computer Programming**

The challenge of parallel algorithms has been passed to the next generation of computer programmers

That's us!

“If you build it, they will come”



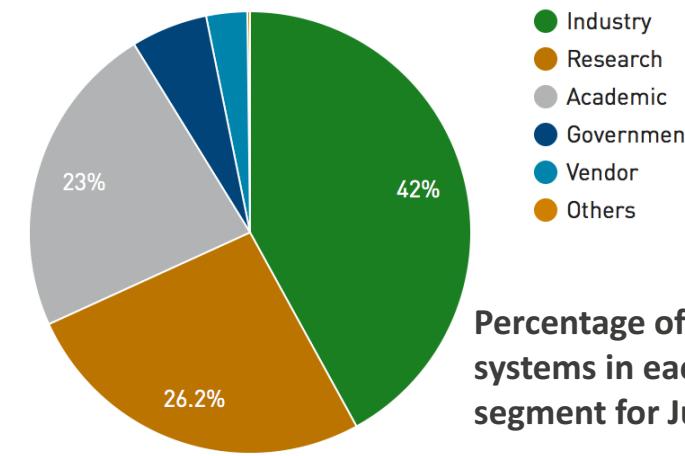
And so we built them. Multiprocessor workstations, massively parallel supercomputers, a cluster in every department ... and they haven't come. Programmers haven't come to program these wonderful machines. Oh, a few programmers in love with the challenge have shown that most types of problems can be force-fit onto parallel computers, but general programmers, especially professional programmers who "have lives", ignore parallel computers. ...

The overwhelming majority of programmers will not invest the effort to write parallel software.

- **Mattson, Sanders, Massingill, Preface to Patterns for Parallel Programming, 2005**

Workshop Purpose: The needs are growing...everywhere

- Parallel computing skills are a critical need due to gaps in university domain science degree programs and a growing complexity and scale of computing platforms
- Skilled scientists and engineers are more important than the hardware – we should invest as much in their development as we do in hardware R&D.



Percentage of the Top500 systems in each HPC segment for June 2015

International Data Corporation (IDC) study of talent and skill impacting HPC data centers found the HPC workforce aging and retiring, and **93% of HPC centers having difficulty hiring staff with the requisite skills** as a result.

~E. Joseph, S. Conway, and J. Wu. A study of the talent and skill set issues impacting hpc data centers conducted on behalf of the us department of energy. Technical report, IDC, July 2010.

Workshop Purpose: Preparing the next generation

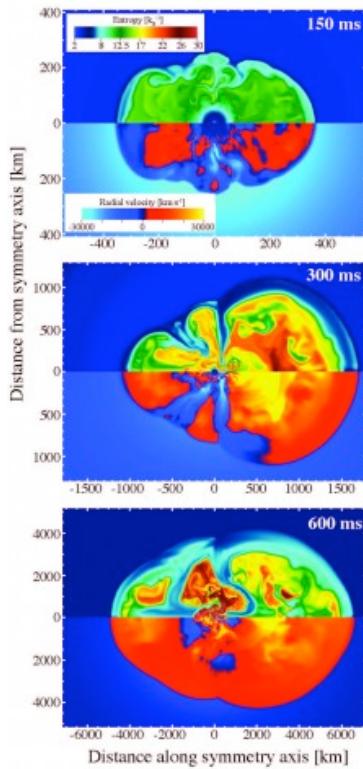
- NSF has recently recognized this gap and has a NSF/IEEE-TCPP Curriculum Initiative on Parallel and Distributed Computing – Core Topics for Undergraduates. (Bob Robey is on the committee this year)
- But this effort targets only Computer science and Computer Engineering – not domain scientists.
 - 80% of Titan users are not CS majors – domain scientists
 - More effort is needed to prepare the next generation
 - It takes a community to solve HPC challenges to tackle real-world problems

A 2013 Networking and Information Technology Research and Development Program (NITRD) publication concluded, “**current approaches to HEC (high-end computing) workforce development and education are inadequate to address today's needs in HEC centers and scientific disciplines that depend upon HEC; as demands upon HEC increase, this gap will widen.**”

~High End Computing Interagency Working Group. Education and workforce development in the high end computing community. Technical report, NITRD.

HPC: An Essential Tool to Advance Scientific Discovery

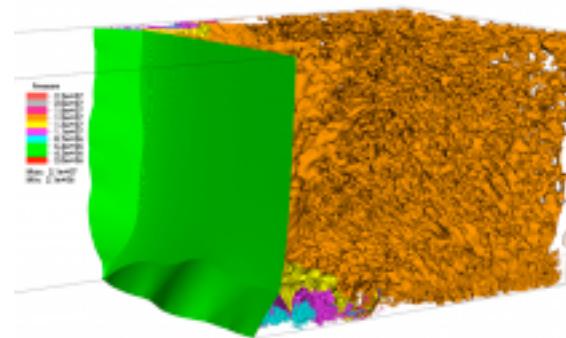
- Basic science research at national labs and academia



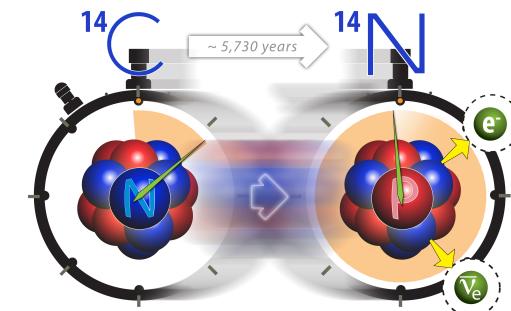
Astrophysics:

2D simulations of the evolution of the entropy (upper half) and radial velocity (lower half) of a supernovae explosion

www.nersc.gov



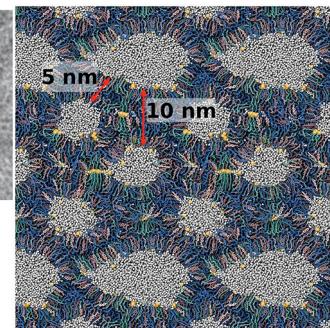
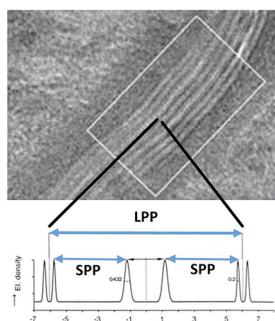
High-Speed Combustion and
Detonation: Alcf.anl.gov



Nuclear Physics:
Understanding the anomalous
long lifetime of carbon-14

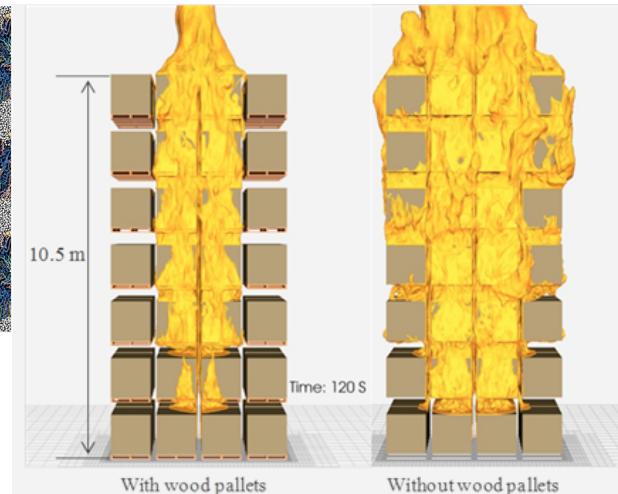
HPC: An Essential Tool to Solve Real-World Problems

- Industry
 - Testing is expensive & time-consuming
- For the National Interest



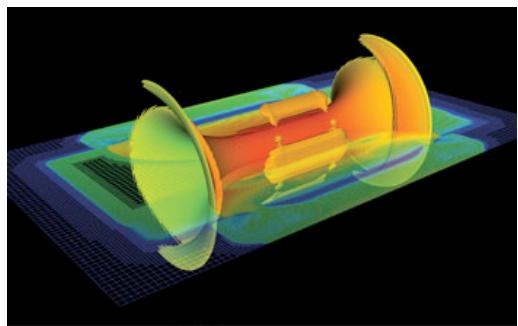
Simulation to predict epidermal response to compound (P&G)

<https://www.olcf.ornl.gov/>



Insurance company FM Global simulates warehouse fires

<https://www.olcf.ornl.gov/2016/01/05/fighting-fire-with-firefoam/>

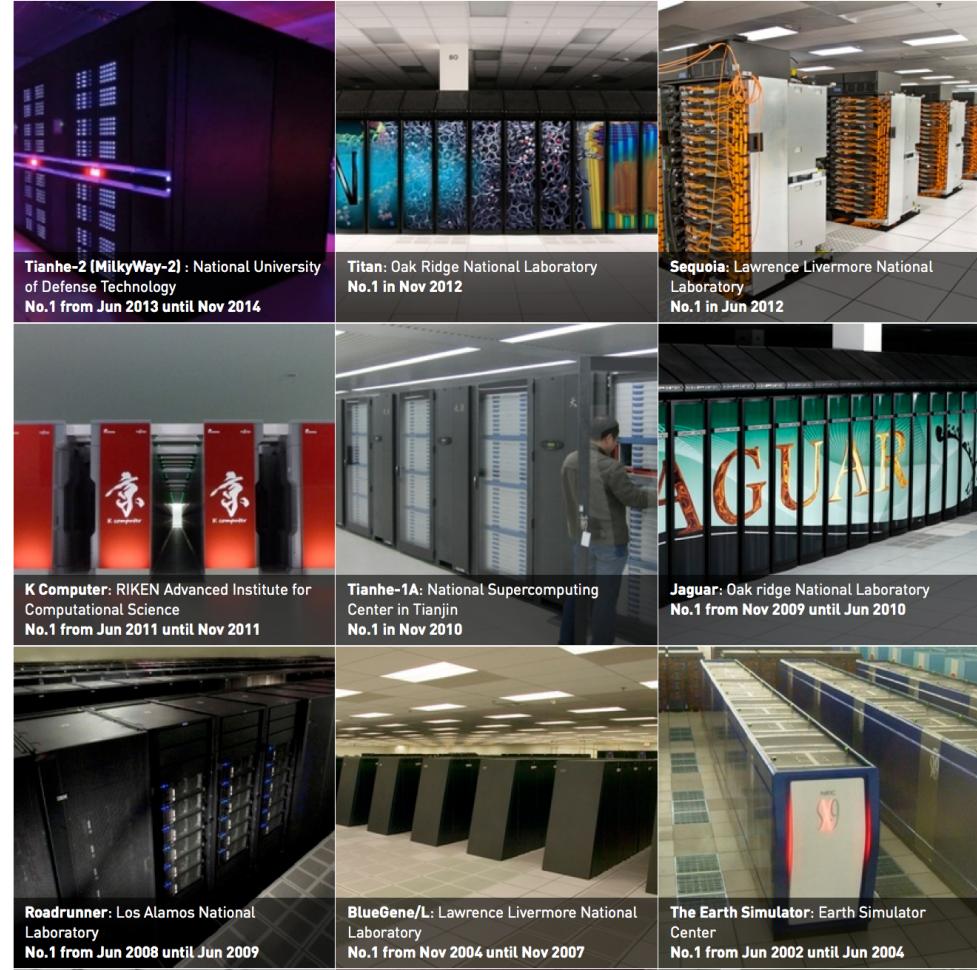


Stockpile Stewardship: First full-system, three-dimensional simulations of a nuclear weapon explosion are performed. (2002)

<http://www.lanl.gov/about/history-innovation/innovation-timeline.php>

Large-Scale Resources

- **Hero-runs**
 - 1 simulation using the entire resource (e.g. supernova explosion)
- **Time to solution**
 - Investigate a parameter space, e.g. materials by design (too many options for experiment alone)



Resources for this Program

- **LANL Resources**
 - Grizzly
 - 1490 Nodes
 - 2 Sockets, 18 cores Broadwell per node
 - Wolf
 - 616 Nodes
 - 2 Sockets, 8 cores Sandy Bridge per node
 - Darwin
 - All the things



Resources for this Program

- **Blue Waters at Illinois**
 - Partition 1
 - 22,640 XE Nodes
 - 2 Sockets, 8 core* Bulldozer per node
 - Partition 2
 - 4,228 XK Nodes
 - 1 Socket, 8 core* Bulldozer per node
 - 1 Kepler GPU



Resources for this Program

- **Cori at Nersc**
 - Partition 1
 - 2388 Haswell Nodes
 - 2 Sockets, 16 Cores per Node, 2.3 GHz
 - Partition 2
 - 9,688 Xeon Phi (Knights Landing) Nodes
 - 1 Socket, 64 Cores per Node, 1.4 GHz

HPC Itself is an Area of Active Research

- With a vibrant & interconnected community
 - Programming Models
 - Programming Tools
 - File system
 - Network
 - Processors
 - Resilience
 - Algorithms
 - Numerical Methods
 - Workflows
- Close integration with vendors (Cray, Intel, NVIDIA, IBM, SGI, HP, etc.)



HPC and the Wall

- **At Exascale**

- Target is a 20 MW system (current is ~10 MW)
- Current technology would require a 100 MW system and is largely responsible for pushing the target dates for the first exascale system out to 2020 or 2022.
- At 20 MW, the power cost of running the system would be 2-3 times the purchase cost of the system

- **Other walls**

- Storage
- End of Moore's Law
- Fault rate



Important to the Nation

- **On July 29, 2015, the White House issued an executive order establishing the National Strategic Computing Initiative (NSCI)**
 - To ensure the US continues its leadership role in HPC and its use in solving complex scientific and national problems.
 - The push to Exascale & Data-Driven HPC
- **Department of Energy (energy.gov)**
 - Office of Science (science.energy.gov/)
 - Advanced Scientific Computing Research Program (ASCR)
 - User Facilities (ORNL, ANL, NERSC/LBL)
 - INCITE: Free large-scale system allocation based on competitive allocation proposal process – show a need & good science
 - NNSA (nnsa.energy.gov)



What is Parallel Computing?

- a. MPI
- b. Exposing simultaneous computations in an astrophysics application
- c. Processor hyper-threading
- d. Asynchronous I/O
- e. All of the above

Parallel computing is not one thing. It is many different technologies and concepts ranging from the physics to the hardware.

What's Hard About Parallel Computing?

- **Finding and exposing parallelism**
 - Within the constraints imposed on choices of methods and algorithms by a given application domain
- **Constructing solutions that function on—and ideally make efficient use of—available computing platforms**
 - Managing hierarchies
 - Processing elements of varying capabilities
 - ... organized into processors of varying sizes
 - ... connected within nodes with varying numbers of sockets
 - ... with access to multiple levels of caches, DRAM, non-volatile memory, and storage
 - ... and associated non-uniform memory access and affinity effects
 - ... connected into systems with varying interconnect topologies

What's Hard About Parallel Computing? (continued)

- **Maintaining the solution you've developed**
 - As platforms change
 - As language or other standards evolve
 - As dependencies are updated or become obsolete
- **Detecting and correcting defects**
 - An N -way parallel application has N times as many changes to trip over bugs
 - In its own code
 - In software it depends on
- **Effort spent on optimization can quickly move from “easy wins” to “unsolved problems”**

Amdahl's Law (1967)

$$SpeedUp(N) = \frac{1}{S + \frac{P}{N}}$$

- Where **P** is the parallel fraction of the code, **S** is the serial fraction, which means **P+S = 1**, and **N** is the number of processors
- Try **S= 20%, 10%** and plot speedup – the parallel speedup is limited by the serial section of the code (20x for 5%, 10x for 10%, 5x for 20% serial)

Gustafson-Barsis's Law (1988)

- Pointed out that parallel code runs wanted to increase the size of the problem as more processors are added. If the problem size grows proportionally to the number of processors:

$$SpeedUp(N) = N - S \cdot (N - 1)$$

- Where **N** is the number of processors, and **S** is the serial fraction
- Net result is that a larger problem can be solved in the same time by using more processors and now parallelization is useful

Parallel Computing Is Not a New Field

- **Single instruction, multiple data (SIMD) architectures:**
 - Thinking Machines CM-1 (1985); Sun UltraSPARC I (1995); Intel Pentium MMX (1996)
- **Massively parallel systems and clusters:**
 - Intel iPSC/1 (1985); Intel Touchstone Delta (1992)
- **Message passing:**
 - Caltech Cosmic Cube (1985); NX (1988); PVM (1989); MPI 1.0 (1994)
- **Shared-memory programming:**
 - Cray SHMEM (1993); OpenMP 1.0 (1997)
- **Graphics processing units (GPUs) for scientific computing:**
 - E.S. Larsen, D. McAllister, “Fast matrix multiplies using graphics hardware”, Proceedings of Supercomputing 2001
- **Partitioned global address space languages:**
 - Global Arrays (1994)

... But Neither the Challenges Nor the Possible Solutions are Static

- Intel Xeon E3 v5 (Skylake): Q4 2015
- IBM POWER8: June 2014
- Nvidia Pascal: May 2016, P100: 2017
- Intel Xeon Phi (Knights Landing): now
- C++14: August 18, 2014; C++17 release process
- Fortran 2008: September 2010; Fortran 2015 in development
- MPI 3.0: September 21, 2012; MPI 3.1: June 4, 2015
- OpenMP 4.5: November 15, 2015
- CUDA 7.5: September 8, 2015, Since then, CUDA 8 and CUDA 9 released
- Threading Building Blocks 4.4 Update 3: February 11, 2016
- Chapel 1.12: October 1, 2015

Parallel Code and Me?

- **Computationally demanding scientific code projects should plan for parallelism**
 - Think parallel
 - Consider impact of algorithm choice
 - Consider parallel patterns

Best to introduce parallelism early in code development efforts.

Workshop Goals

- **Provide solid HPC education for next-generation workforce**
- **Create a common language across disciplines and break down barriers from science domain to hardware**
- **Explore algorithms, methods, and technologies based on architectural features**
- **Instill good software development practices**
- **Develop collaboration skills and processes**

Workshop Goals (continued)

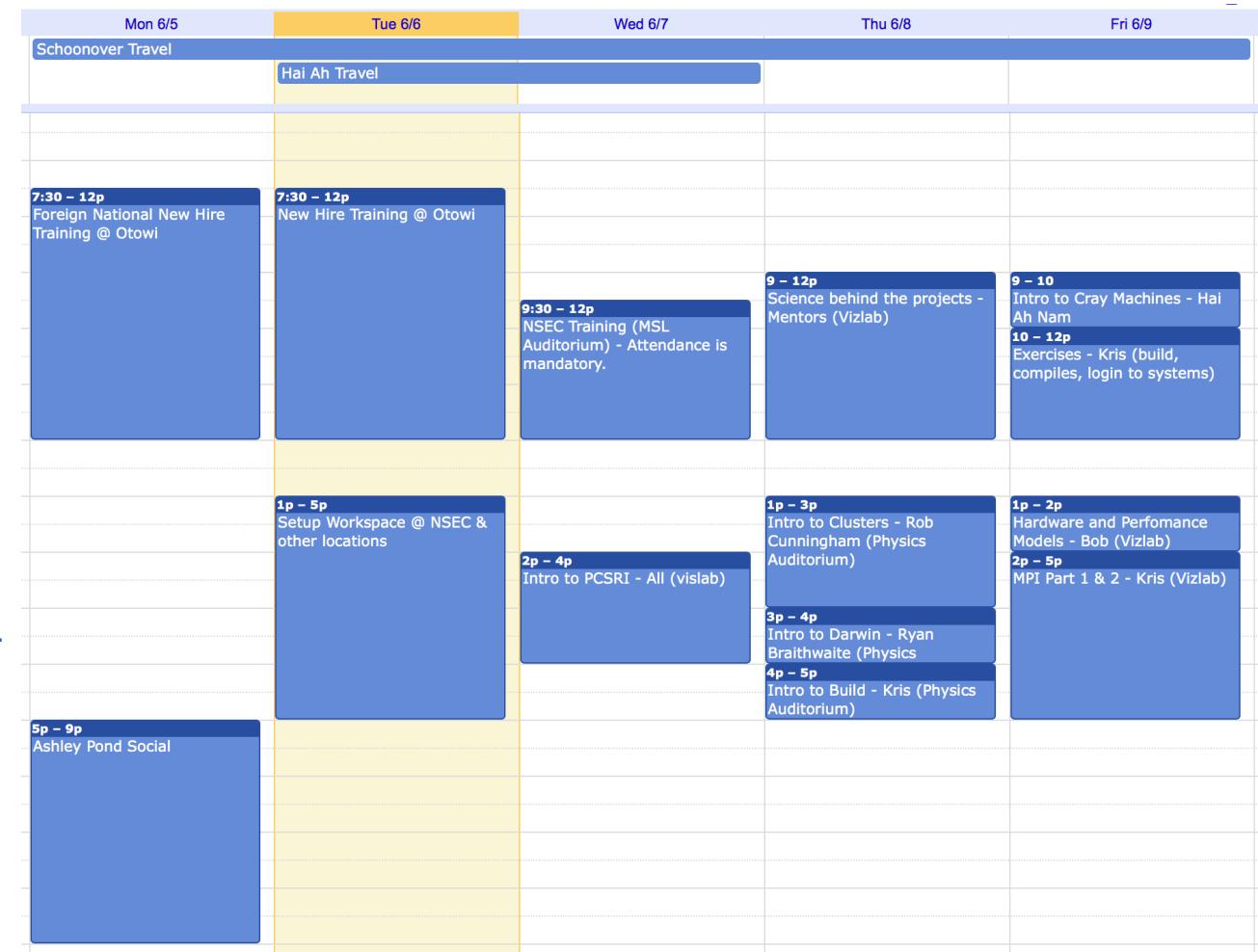
- **Establish a new staff pipeline for LANL**
 - Expecting a 50% turnover in staff over the next decade
 - Over half of staff historically have started in student programs
- **Of course, students who go on to other organizations help develop our community**
 - Informing other organizations about needs of the leading scientific laboratories

Three phases to Summer Internship

- **Early introduction phase with 2-3 lectures a day for three weeks**
- **Middle phase begins with project formulation**
 - Projects have been developed and assigned earlier this year
 - need to start thinking of publication (poster, report, etc) early due to lead time for review
- **End phase**
 - Complete writing of report or poster preparation
 - Student Symposium – August 8th
- **Internship ends Friday August 11th**

The Schedule (Week 1)

- Roughly:**
 - 60% parallel computing
 - 25% domain science
 - 15% software development and engineering
- See link from Hai Ah's email for calendar**
<https://calendar.google.com/calendar/embed?src=s91udajg1gilrk4n71cephanvo%40group.calendar.google.com&ctz=America/Denver>
- Try just**
<http://calendar.google.com>



Expectations for the Internship

- **Nominally 8 hour workday, hour for lunch, evening mostly free – practice work-life balance**
 - Some flexibility, but need to be available for mentors, other students and staff
 - Hours for building are 6:30 am to 5:30 pm. Badge will not let you in after 5:30.
 - Leaning towards consensus time of 8:30 – 5:30 with 1 hour lunch
 - Discussion – also discuss with your mentor(s)
 - Post deviations to this time at workspace and leave note if gone more than half-hour
- **Business casual dress, but towards casual end – common-sense dress (no flip-flops, no offensive t-shirts) – we will have visitors on many days and will be recording video**
- **Teamwork is critical and expected**
 - It is important to participate in lectures and projects
- **If sick, stay home**
 - Call or email PIs or staff
- **Time off – school or personal, please talk to us ahead of time**

Office Rules – When in Doubt, Ask

- Personal laptops are permitted in office space, but regular use is strongly discouraged.
- Do not plug any storage device into a LANL computer, especially thumbdrives. LANL issued thumbdrives only – see staff.
- No photos, no cameras (lab-wide on all lab property)
- Cell phone OK in this building (also Otowi and Library), but turn-off blue-tooth
- Tag your bag
- No tailgating – everyone swipes
- Wear your badge while on-site, and not while off-site
- Rules may be different in other buildings!!

Information handling

- **Export control – we may encounter some export controlled codes**
 - Maintain proper access controls
 - Share only with need to know for work purposes
 - Proper license agreements for those without citizenship (ask us)
- **Public release (LA-UR) requires a review from supervisor and lab (5 day lead time)**
 - Get draft reviewed early – modifications are permitted and can be submitted later.
 - Good practice – have at least one document of work done reviewed by end of summer so you can share outside lab
- **Non-disclosure agreement (NDA) with commercial organizations limits release of information**
 - Generally performance results before public release of products is restricted (check with us). There have been nodes covered by NDA on Darwin and other early systems we may get access to.

Open and Closed Facility

- **We have been one since the beginning**
 - Currently about 10-20% Foreign Nationals
- **LANL fights hard to maintain this dual status**
- **Please observe the procedures in-place**
- **Foreign nationals need to know export control rules too!**
- **Open science – LA-URs, textbook materials are examples**
- **”Proprietary” or program specific knowledge not transferrable and share only with “need-to-know”.**
- **Think copyright – need permissions from owners to use in other settings.**

Travel Reimbursement

- You should schedule time with Nickole to process your travel reimbursement for your travel to Los Alamos
- Please be considerate of her time and do as much as you can to help fill out the forms
 - Have all your receipts organized
 - Meals are done per-diem – you get a set amount and do not need receipts for meals
 - Personal vehicle is done per mile traveled
 - Airline tickets, taxi, shuttles, etc need receipts
- Goal is to get all travel reimbursement submitted by next Friday – it creates an accounting problem if it is not done within a short time of the travel

Project Shared Information

- <http://gitlab.lanl.gov>
 - ParallelComputing_2017 / General
 - Group – ParallelComputing_2017
- **Let's collect notes on the wiki for setup and administrative details**
- **I'll add all the students to the group as you get accounts**
 - Rachel LeCover
- **Check utrain.lanl.gov for training requirements for yourself**

Welcome, and thanks

Backup slides

Parallel Computing

Working Definition and Techniques

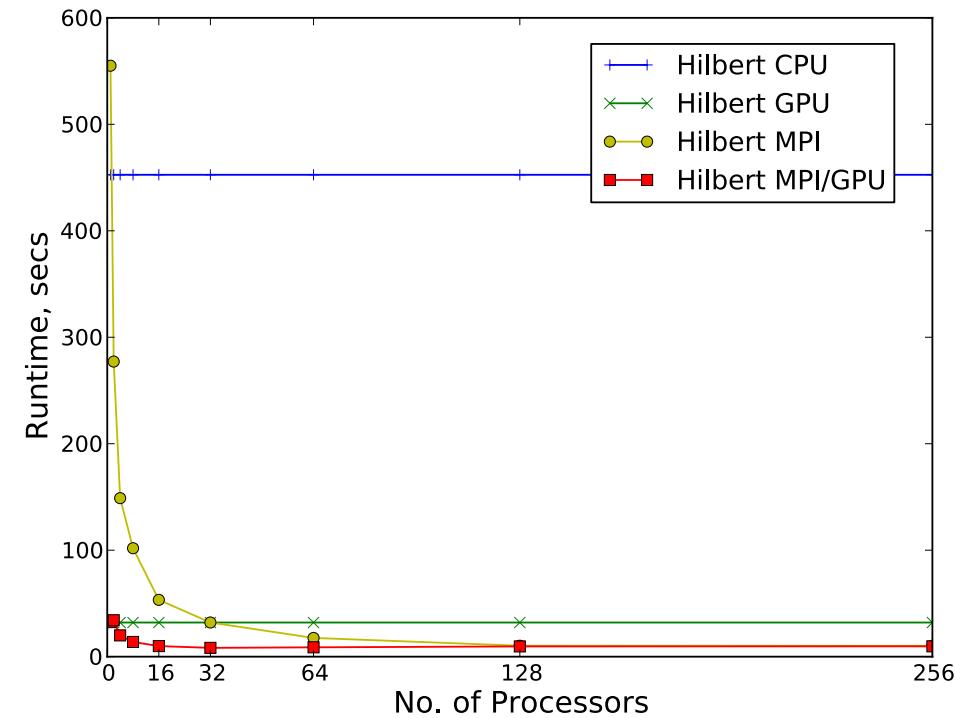
- **Making use of multiple hardware components at the same time**
- **Basic Parallel Techniques**
 - Data Parallel
 - Single Instruction, Multiple Data, SIMD (Vector)
 - Multiple Instruction, Multiple Data, MIMD (Threading and Distributed processes)
 - Task Parallel
 - Pipelining (assembly line)

Parallel Scaling Plots

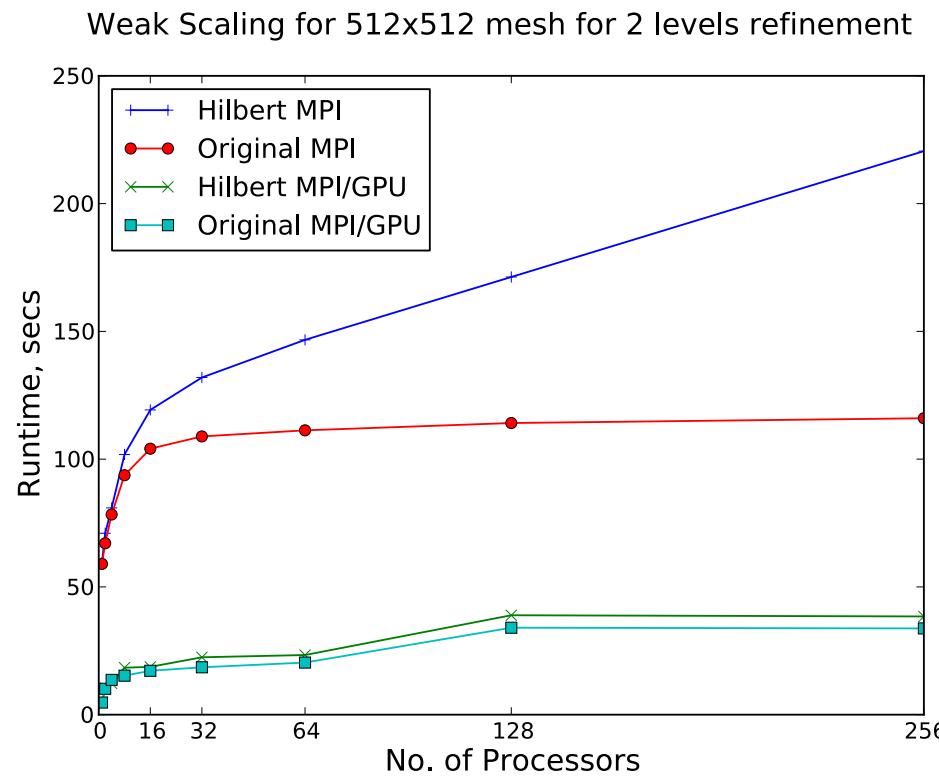
- **Parallel performance is usually shown as plots of runtime relative to the number of processors**
 - Strong Scaling keeps problem size constant and increases processors. Results reflect Amdahl's law
 - Weak Scaling increases problem size proportionally to the number of processors and reflects Gustafson-Barsis's law
- **Plot examples follow for MPI and MPI/GPU**

Parallel Scaling Plots – Strong Scaling

- Note the sharp drop-off in run-time and a leveling off (or even slow-down) of code at larger number of processors.
- Serial run-time for CPU and GPU shown for reference as a horizontal line (not usually included in reports). It is shown here so you can see the overhead of a 1 processor parallel run (560 secs vs 460 secs).



Parallel Scaling Plots – Weak Scaling



Current Parallel Programming for HPC is coarse-grained

Dominant technique is data parallel using MPI

- **Distribute large arrays across processors and split up work on these sections of code**
- **Rest of data is replicated and is done on every processor**
 - Counter-intuitive at first – usually beginning parallel programmers want to do this work on one processor, but then you must communicate to rest – more expensive and error-prone
- **Programs remain largely the same with “ghost-cell exchanges” at the processor boundaries**

Next Generation Parallelism is fine-grained!

- Each loop must be split up among many threads
- Loop must give same results if done in any order – quick test, try doing loop in reverse order

1. *for(i = 1; i<n-1; i++){
 a[i] = a[i-1]/4+a[i]/2+a[i+1]/4;
 }
 }*

Loop dependency, incorrect?

2. *a_{minus} = a[0];
 for(i = 1; i<n-1; i++){
 a[i] = a_{minus}/4+a[i]/2+a[i+1]/4;
 a_{minus} = a[i];
 }
 }*

Loop dependency

3. *for(i = 1; i<n-1; i++){
 a_{new}[i] = a_{old}[i-1]/4+a_{old}[i]/2+a_{old}[i+1]/4;
 }
 }*

No dependency

Swap_ptr(a_{new}, a_{old}, a_{tmp})