SAND2013-5497P

Exceptional service in the national interest

Sandia National Laboratories



A Glimpse into the the Next Decade of Supercomputing: An Overview of Sandia's Advanced Test Bed Project

S.D. Hammond (and many other people)

sdhammo@sandia.gov

U.S. DEPARTMENT OF ENERGY NNSA

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

This work is from a big team...

Sandia National Laboratories

- **Advanced Test Bed Project Management** - Jim Ang, Jim Laros and Sue Kelly
- **System Administrators** - Jason Repik, Victor Kuhns, Jim Brandt and Ann Gentile
- **PMAT and Mantevo** - Richard Barrett, Courtenay Vaughan, Mike Heroux, Jagan Jayaraj, Li Tang, Christian Trott
- **Trinity Procurement Team** - Doug Doerfler and team
- **SST Team** - Arun Rodrigues, Scott Hemmert, Brian Barrett, Jon Wilson, John Vandyke
- **Local System Administrators** - Bill Goldman and team
- **Vendors** - Intel, AMD, NVIDIA etc
- **Very Patient Nearby Workers** - sorry for the noise!

Who's the Speaker?

Sandia National Laboratories

- Si Hammond

- Member of the Technical Staff in 01422 (Scalable Architectures)
- Originally a Post-Doc
- Started life in the UK

- Testbed projects, HPC simulators (Sandia's SST), compilers, programming models, code optimization etc
- CSRI/146

What's in this talk?

Sandia National Laboratories

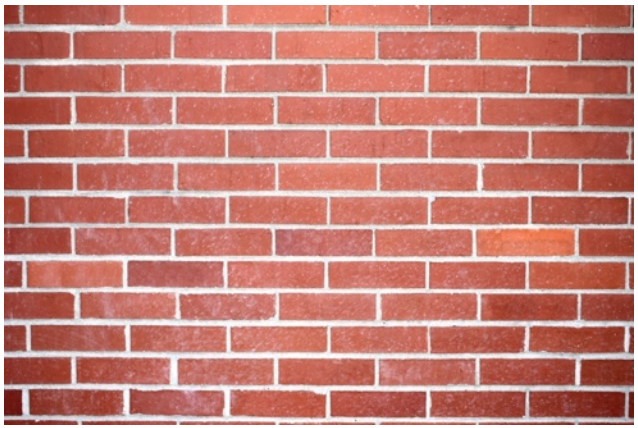
- First half: quick overview of parallel computing and some background for Exascale
  - How do we do parallel computing?
  - Why do we bother running things in parallel - just get a faster processor?
  - What are we trying to achieve?
- Second half: overview of test-bed and potential future Exascale technologies?
  - Deep Dive on Intel Xeon Phi
  - Deep Dive on AMD Fusion APU
- Some conclusions and thoughts about our challenges

PARALLEL COMPUTING

SOLVING A BIGGER PROBLEM WITH MORE WORKERS

Faster Results with Parallelism

- Imagine we have a large workflow, lots of activities we need to get done.
  - How can we make the larger ones go faster?



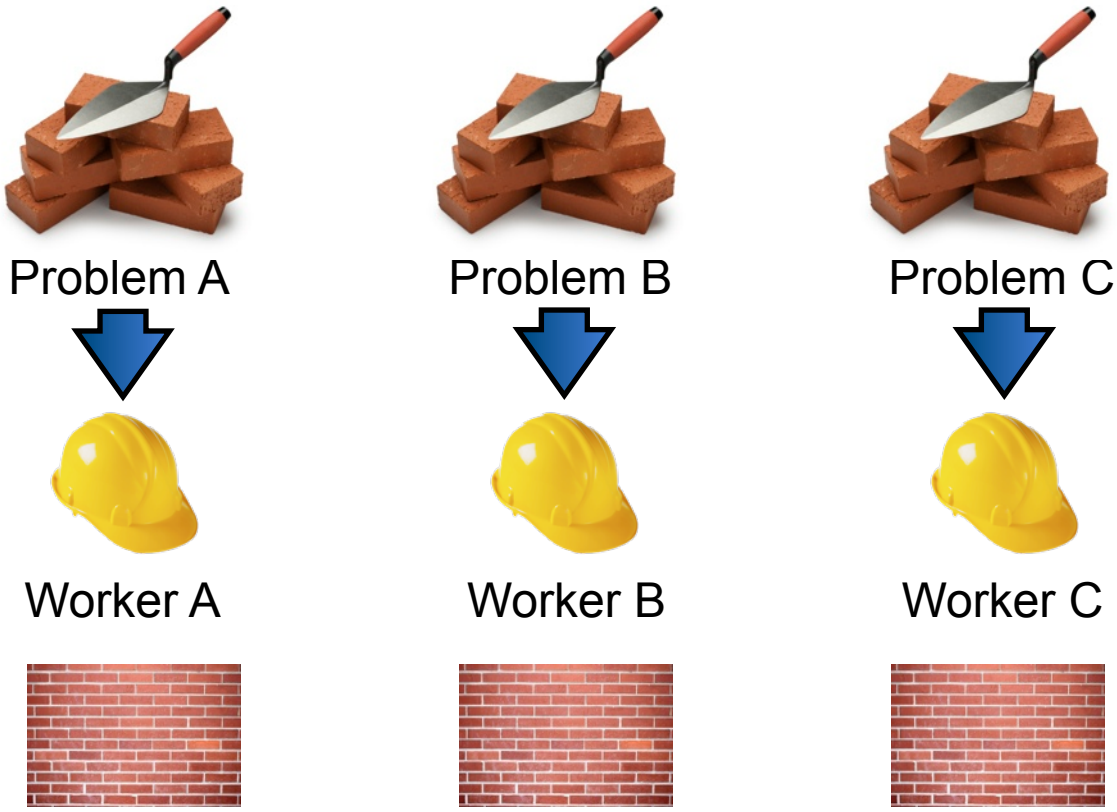
Activity: Build a wall

Inputs

Set of Workers

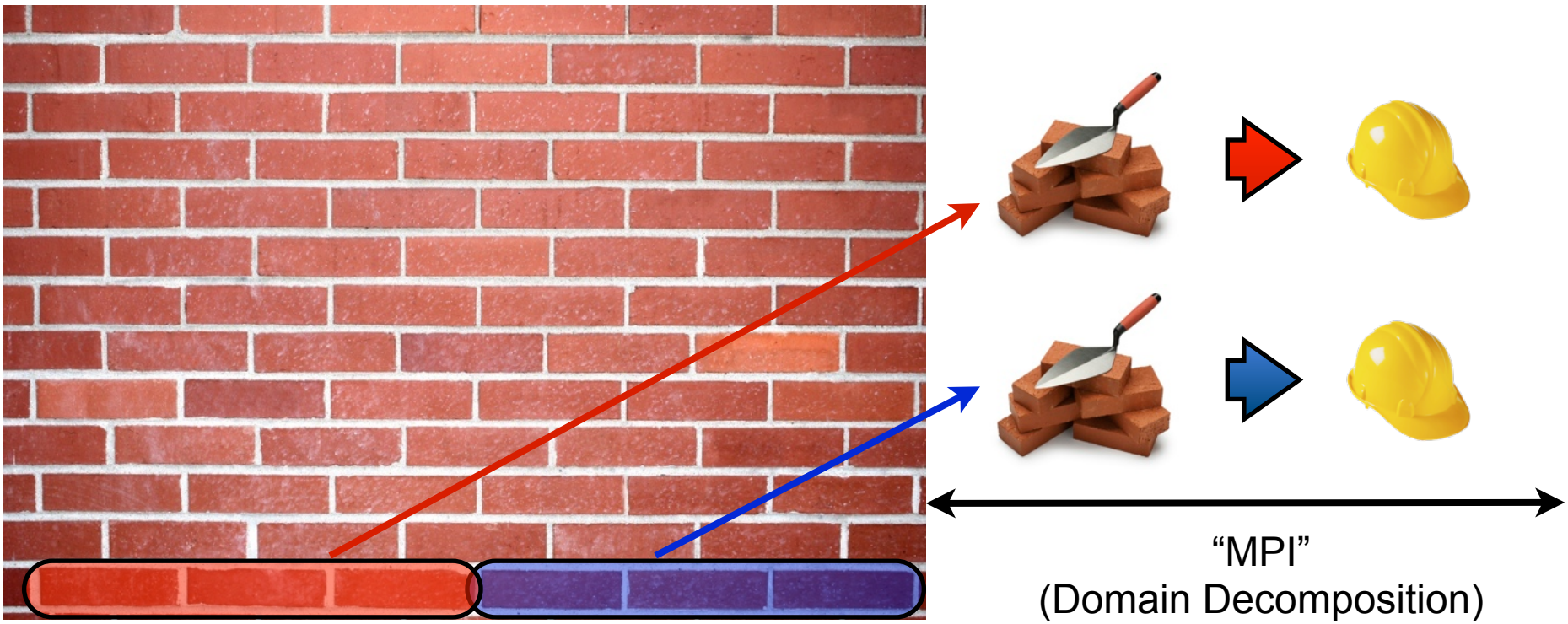
Getting Work done in Parallel

- Allocate one problem per worker
  - We think of this as “embarrassingly parallel” computing - lots of small pieces of work getting done in parallel
  - Can get this done on loosely coupled workstations



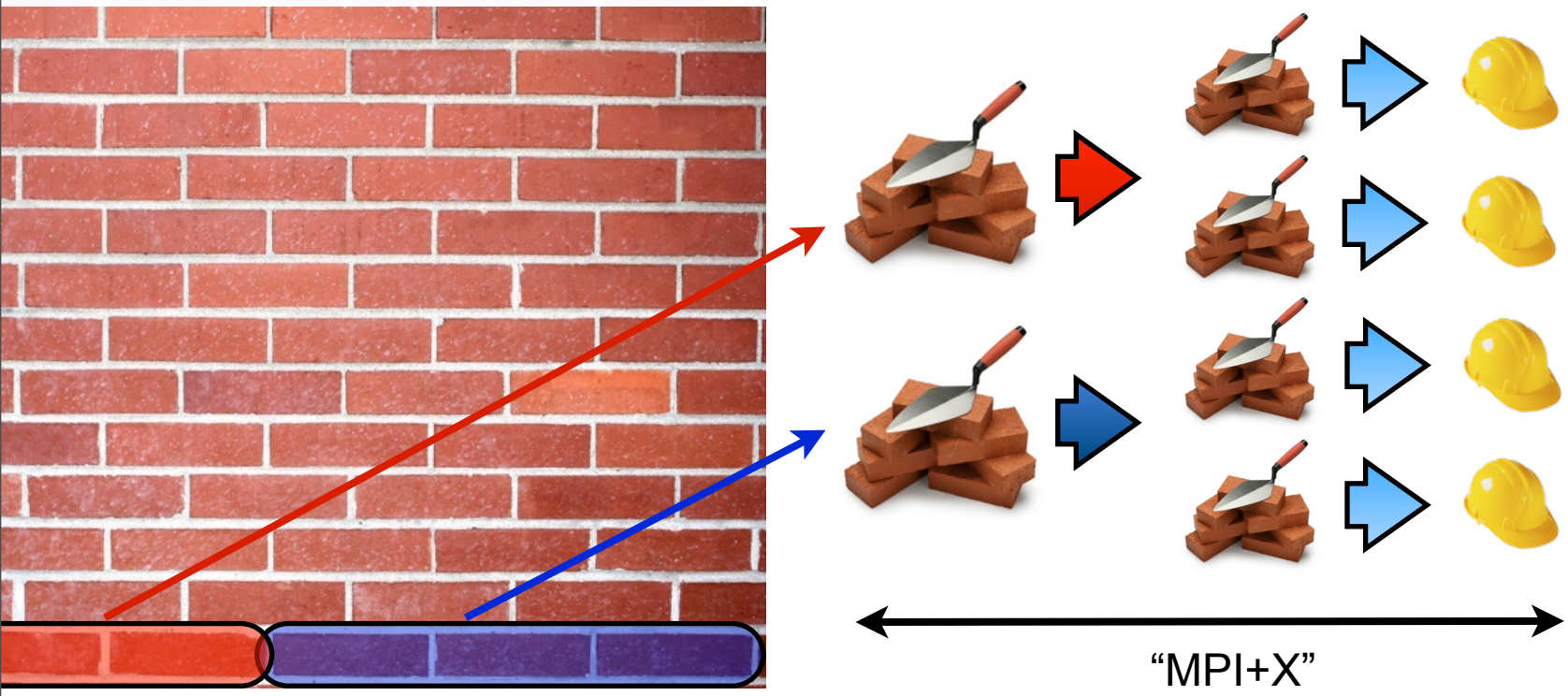
Getting Work done in Parallel

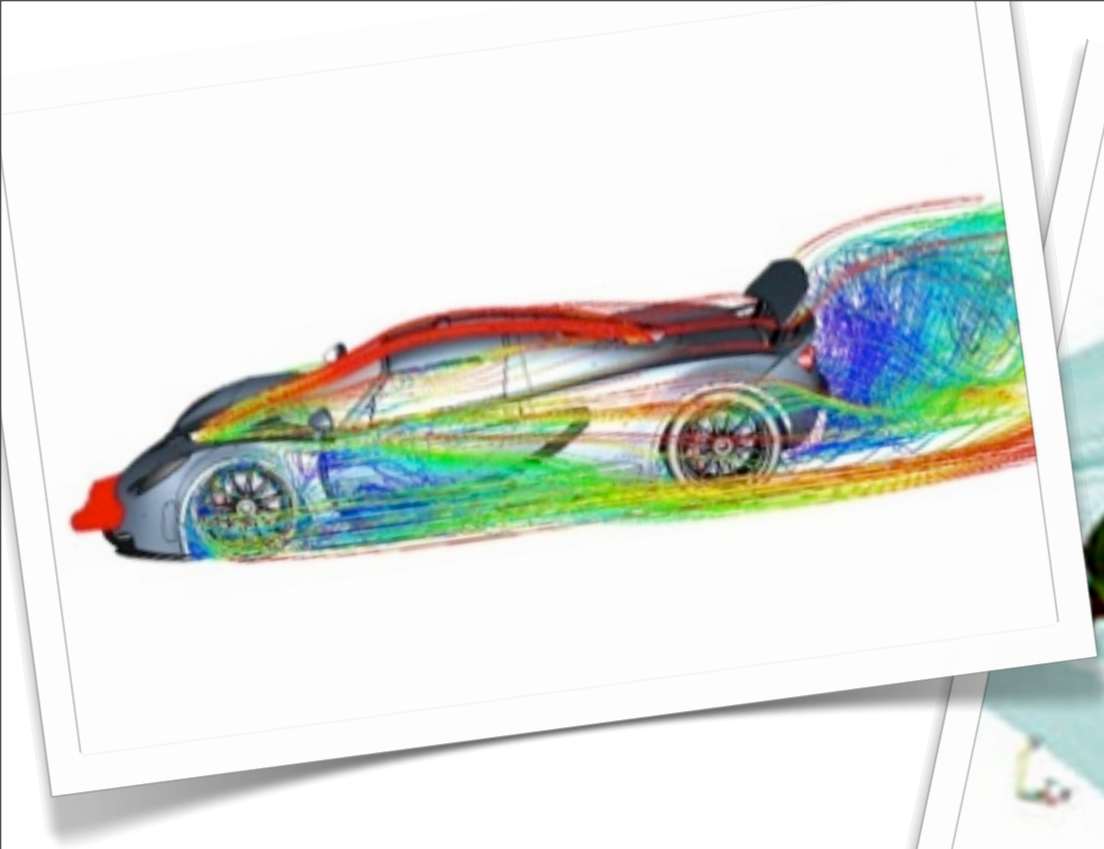
- What if I have a big problem to solve and not lots of smaller ones?
  - Break it up into smaller pieces and then distribute to the workers

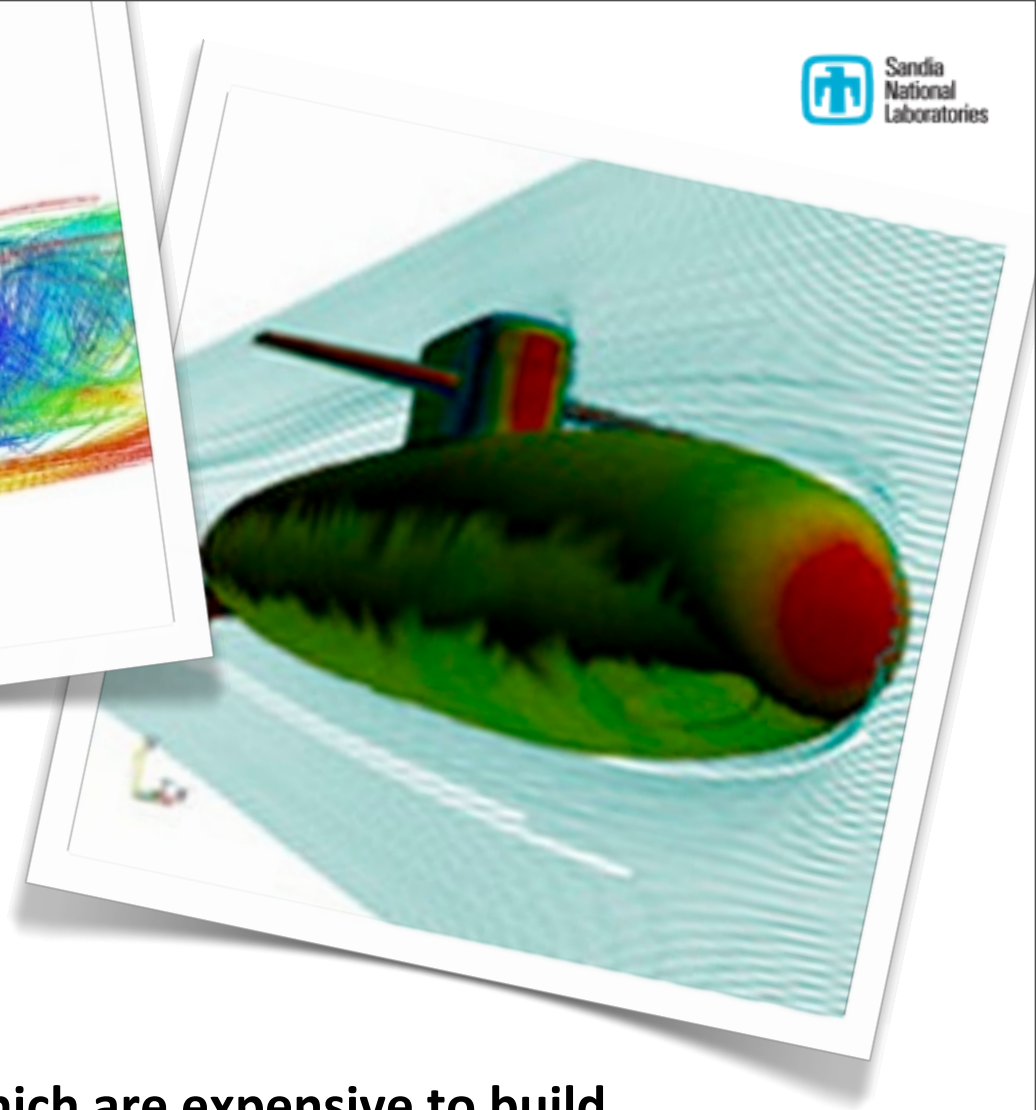


Getting Work done in Parallel

- What if I have a big problem to solve and not lots of smaller ones?
  - Break it up into smaller pieces and then distribute to the workers







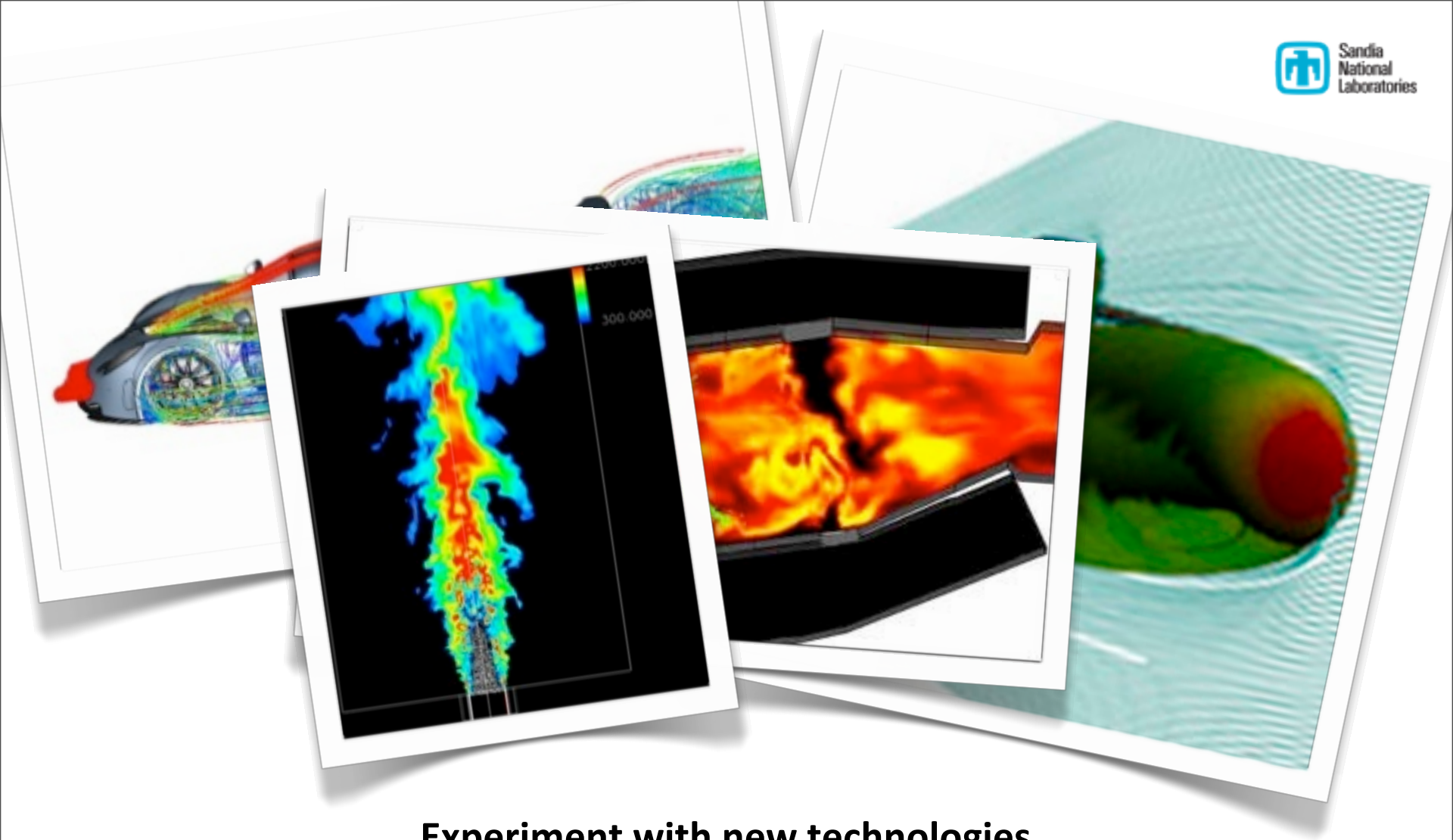
Sandia National Laboratories

Simulate objects which are expensive to build

Multiple iterations on a computer are cheaper and faster than a full product development, are things behaving correctly?

Optimize the behavior of existing designs

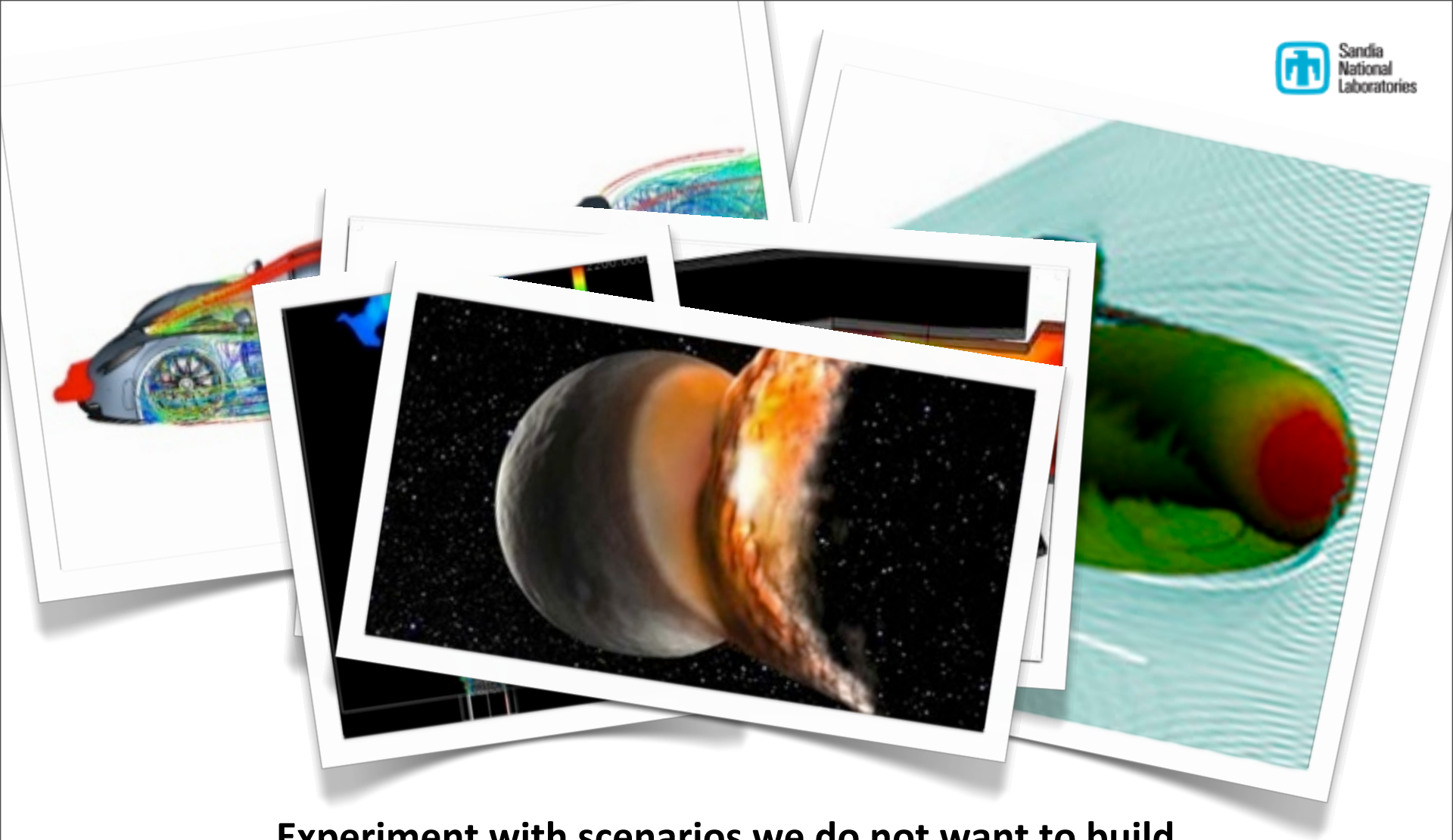
Work out the where the limits of existing designs are - how far can we push technology to meet demands of efficiency



Sandia National Laboratories

Experiment with new technologies

Explore novel approaches by parameter sweeping across inputs using methods may not yet know how to build



Sandia National Laboratories

Experiment with scenarios we do not want to build

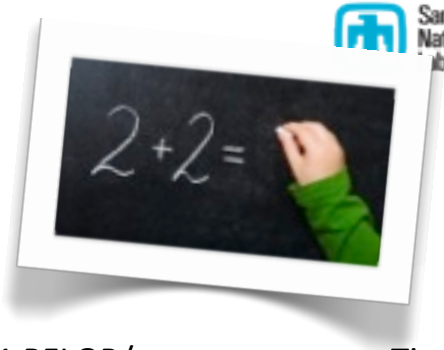
Simulate and research how physics, objects etc may react to event we do not want to actually create

BACKGROUND TO EXASCALE

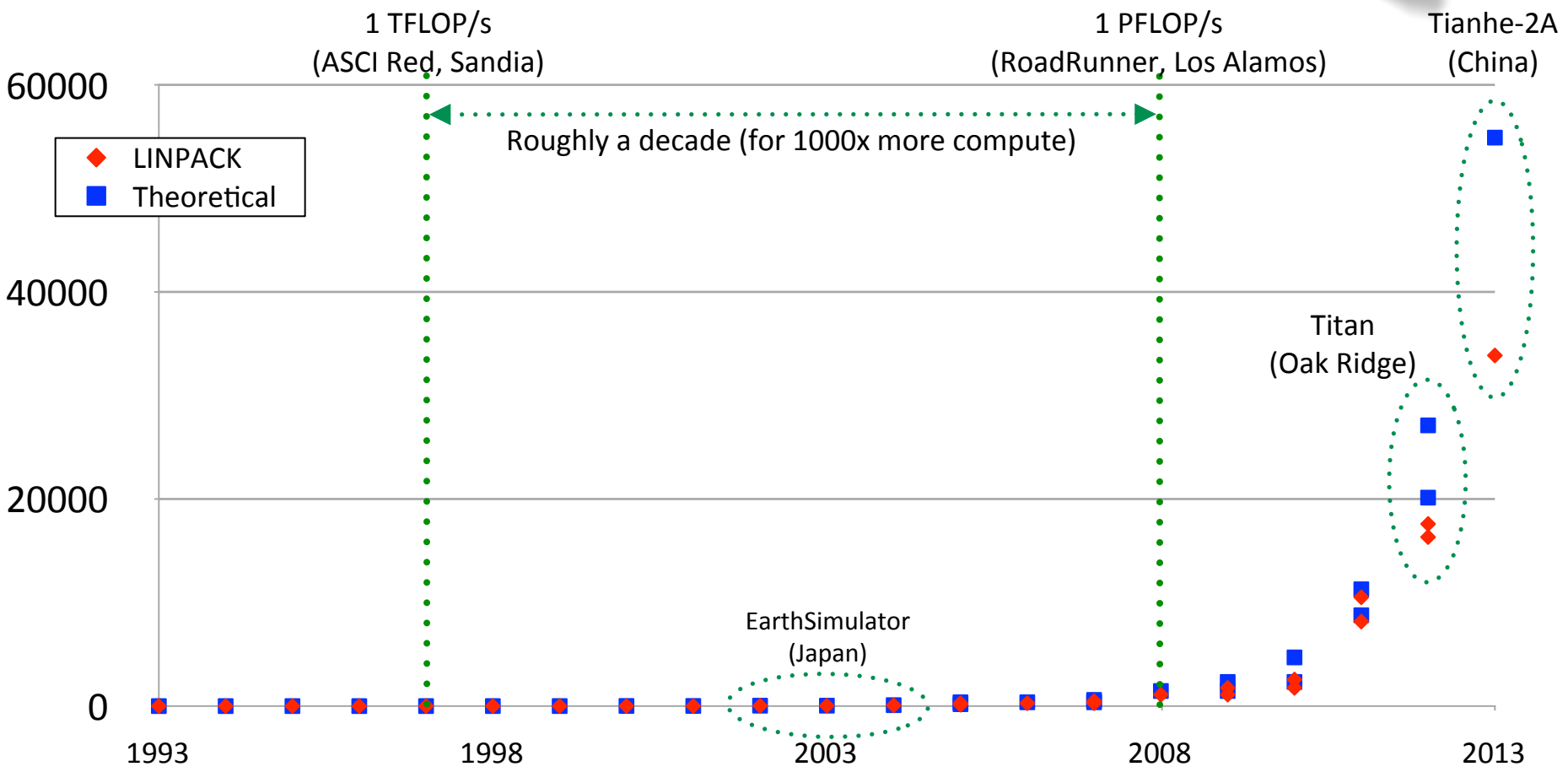
How do we benchmark?

- Need to have a reproducible measure of performance which everyone agrees is acceptable
- Favorite is to use the High Performance LINPACK (HPL) benchmark which measures the compute performance of dense linear algebra matrix-matrix multiplication (http://www.top500.org)
- Established in 1970s and, at the time, mapped well to algorithms of the day

What is a “FLOP”



- HPL is known for producing a “FLOP/s” count
  - **FL**oating-point **OP**erations per **S**econd



The Problems with HPL and FLOP/s

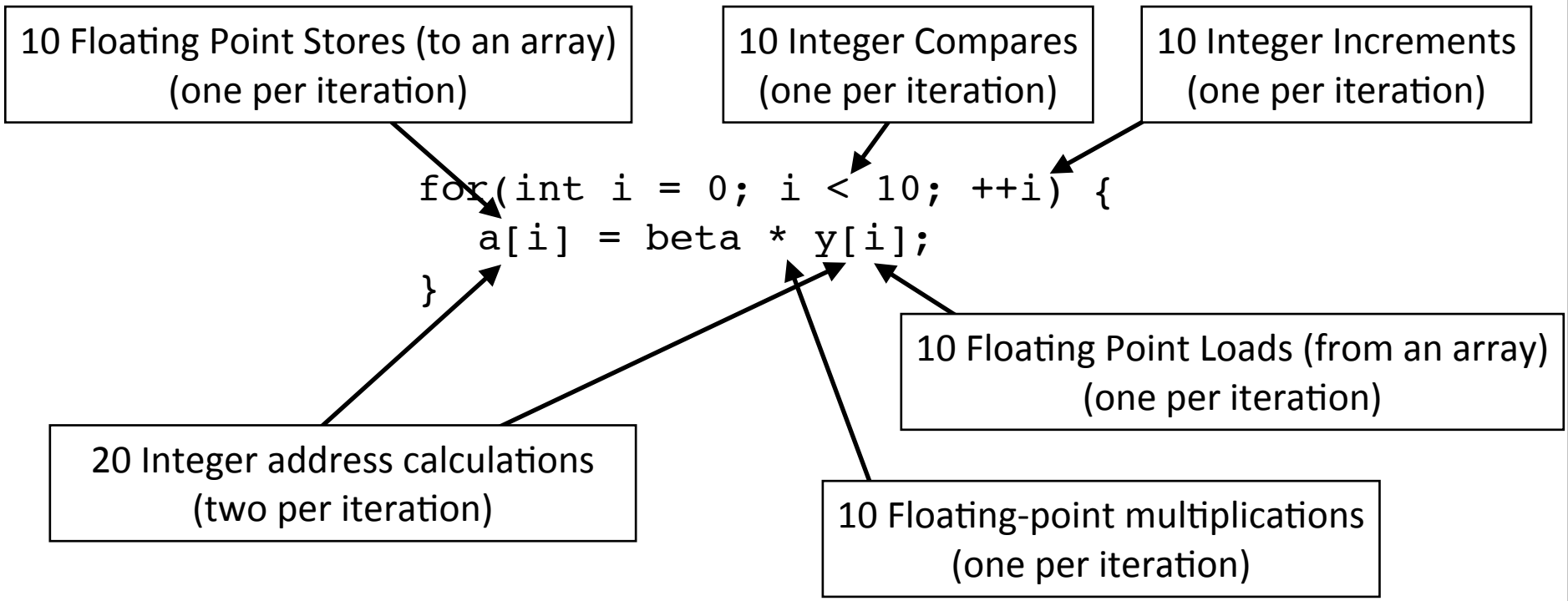
Sandia National Laboratories

- Floating-point arithmetic is only part of the story
  - Means optimizing solely for “FLOP/s” misses important parts of what is takes to really run complex applications

```
for(int i = 0; i < 10; ++i) {
   a[i] = beta * y[i];
}
```

The Problems with HPL and FLOP/s

- Floating-point arithmetic is only part of the story
  - Means optimizing solely for "FLOP/s" misses important parts of what is takes to really run complex applications



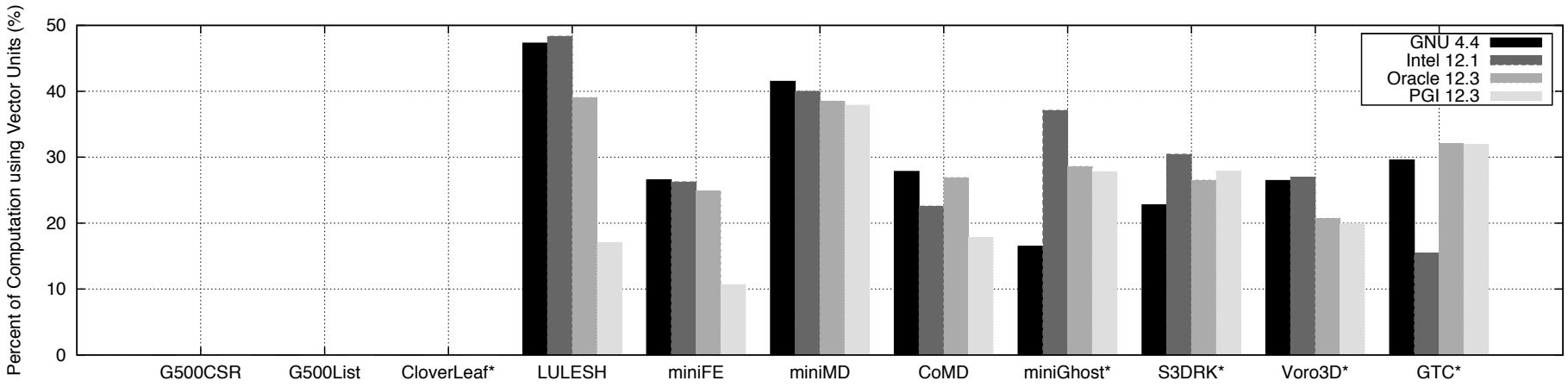
The Problems with HPL and FLOP/s

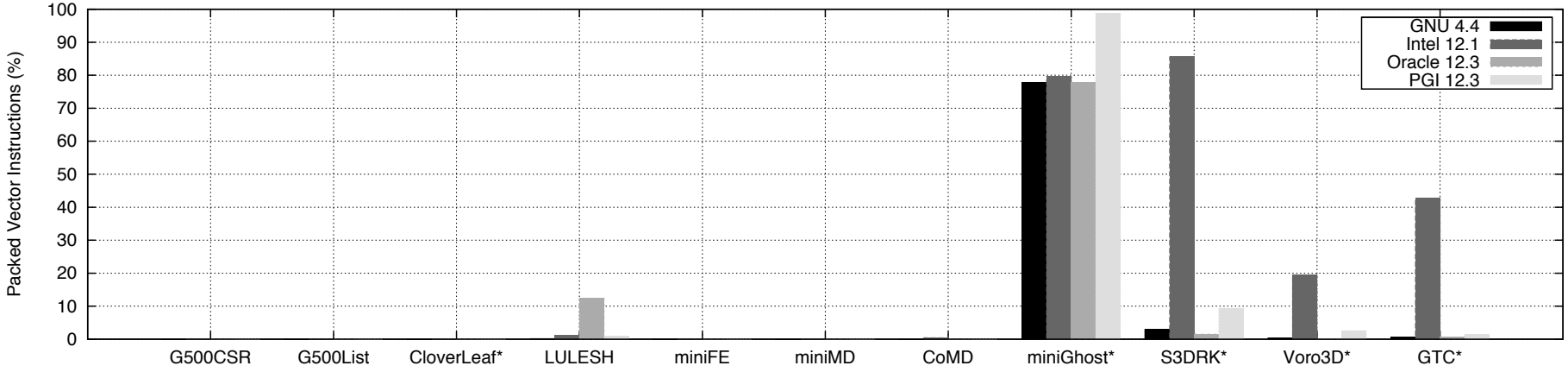
Sandia National Laboratories

- So floating-point operations end up being small proportion of even simple algorithms
  - Even so, the floating point operations take a long time to compute (due to complexity of algorithm)

- Research at Sandia is finding that most of our applications are actually bottlenecks on complicated memory access patterns
  - The real pressure is in accessing sparse operations from across the memory space
  - Breaks conventional memory technologies, prefetching strategies and often upsets caches and TLBs (address maps)
  - Significantly more expensive in time than floating-point operations

Early Study (SSE Vectorization)





What is Exascale?

- In 2008 Los Alamos established the world's first Petascale computer (= 1000 TFLOP/s = 10^{15} FLOP/s)
- Exascale is 1000 PFLOP/s (so a 1000x increase in compute power)
- Want to keep track with the improvements every decade or so, means we would want to deliver in 2018
  - Few people believe this is going to be possible given the constraints of energy use, cost, research, state of algorithms etc
- Must be <u>useful</u> to delivering <u>real</u> science

What can we do with Exascale?

Sandia National Laboratories

- Having so much compute power brings unique challenges but also incredible opportunities
  - Climate Modeling
  - Cleaner combustion and new energy sources
  - Safer sources of power (e.g. nuclear reactors)
  - Advanced material models (lighter planes, stronger bridges)
- Opportunity to take existing science and refine computational simulations to much higher level of fidelity
  - Can we find artifacts in the problem domains which we couldn't previously see
- Economic incentive to improve competitiveness

Challenges

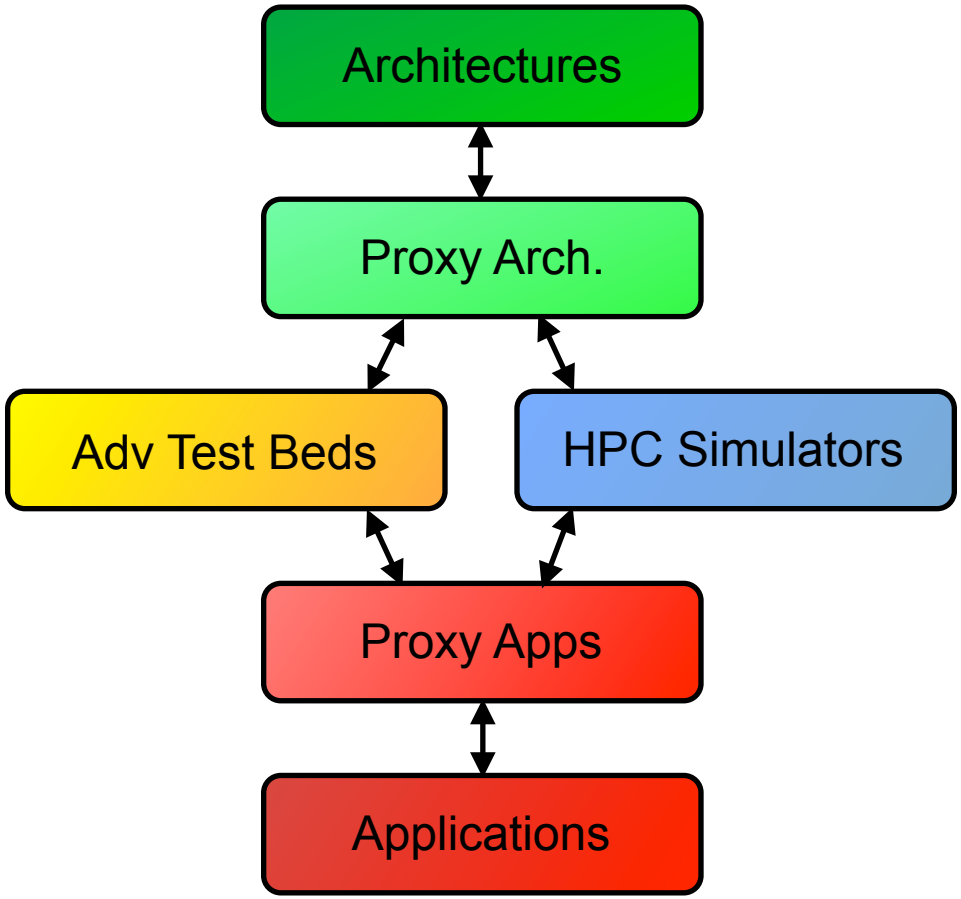
Sandia National Laboratories

- Exascale presents many challenges for the community:
  - Very increased scale (go from a million parallel elements to a billion+)
  - Heat dissipation and energy consumption
  - Algorithms are not *very* parallel (traditionally MPI only, no threads)
  - Reliability of some many machine parts
- Needs sustained research by leading computer sites to deliver genuinely novel approaches
- Moving target
  - No one knows what the hardware will be, optimizing and developing algorithms without knowing the end machine

Research At Sandia

Integrated Approach

- Leverages expertise in:
  - System software
  - Applications
  - Computer Architectures
  - Electronic Engineering
  - Mathematics
  - Vendor relations/industry
- Means we can learn at many levels of fidelity



Advanced Test Bed Projects at Sandia

Sandia National Laboratories

- Sandia is leading the NNSA/ASC test bed initiative
- **Plan:** Field small installations of potential future hardware systems for research, benchmarking, system software *etc*
  - Available for research community across the USA including universities, national laboratories and even industry
  - Renew hardware as vendors release updates or new technologies
  - Act as a proving ground for new applications and algorithms
- Generally keep systems open to encourage collaboration and wider participation
  - Exascale needs a broader audience than just the NNSA in order to be cost effective and attractive to vendors

Advanced Test Bed Projects at Sandia

Sandia National Laboratories

- Is not *just* about hardware - we must have a good software story as well
  - Investigation of system software, profilers, compilers, debuggers *etc*
  - Optimization and porting of existing algorithms and applications
    - Can we make what we already have go faster?
  - New programming models, languages and approaches

- Spread the research to be as broad as possible so we don't overlook any potential solutions
  - Use results to evaluate practically what could be achieved at the lab
  - Remembering we have a large legacy code base which cannot easily be rewritten in short periods of time (or small amounts of funding)

What Test Beds do we have?

- **Compton**
  - Dual-socket Intel Xeon with dual Intel Xeon Phi/MIC cards
  - InfiniBand interconnect
- **Shannon**
  - Dual-socket Intel Xeon with dual NVIDIA Kepler K20X GPU cards
  - InfiniBand interconnect
- **Teller**
  - Single-socket AMD Fusion APU (CPU + GPU fused)
  - InfiniBand interconnect
- **Volta (installed over summer)**
  - Single cabinet Cray XC30 with Intel Xeon processors
  - Cray Aries interconnect (“Dragonfly” topology)

What applications are we using?

Sandia National Laboratories

- Broad range of "mini-apps" developed across the DOE laboratories (including NNSA/ASC and Office of Science)
  - **Originally developed under the Mantevo project at Sandia**
    - miniFE - Finite Element generation and solve
    - miniMD - Molecular dynamics (Lennard-Jones and EAM (Neighbor Lists))
    - miniGhost - Ghost communication exchanges
    - miniSMAC - Computational fluid dynamics being developed
  - **Co-design Centers from DOE**
    - LULESH - hydrodynamics kernels
    - CoMD - Molecular dynamics (Lennard-Jones, EAM (Cell-Lists))
    - SMC/S3D - chemistry and combustion
    - OpenMC - Monte Carlo particular transport algorithms
  - **Other mini-apps from NNSA/ASC laboratories**

http://www.mantevo.org

Programming Languages and Libraries

Sandia National Laboratories

- In order to get applications to run on these new architectures we often have to write key algorithms in new technologies
- **Threading** - a lightweight process which has its own local storage and its own separate control flow
  - OpenMP, pthreads, qthreads, Intel Cilk Plus, Intel TBB, CUDA, OpenCL
- **Vectorization** - vector "lanes" are unique pieces of data but we apply the same instruction to each lane ("data-parallel")
  - Intel/GCC intrinsics, CUDA (to some degree), OpenCL (to some degree)
- Most new hardware needs both threading and vectorization

Deep-Dives

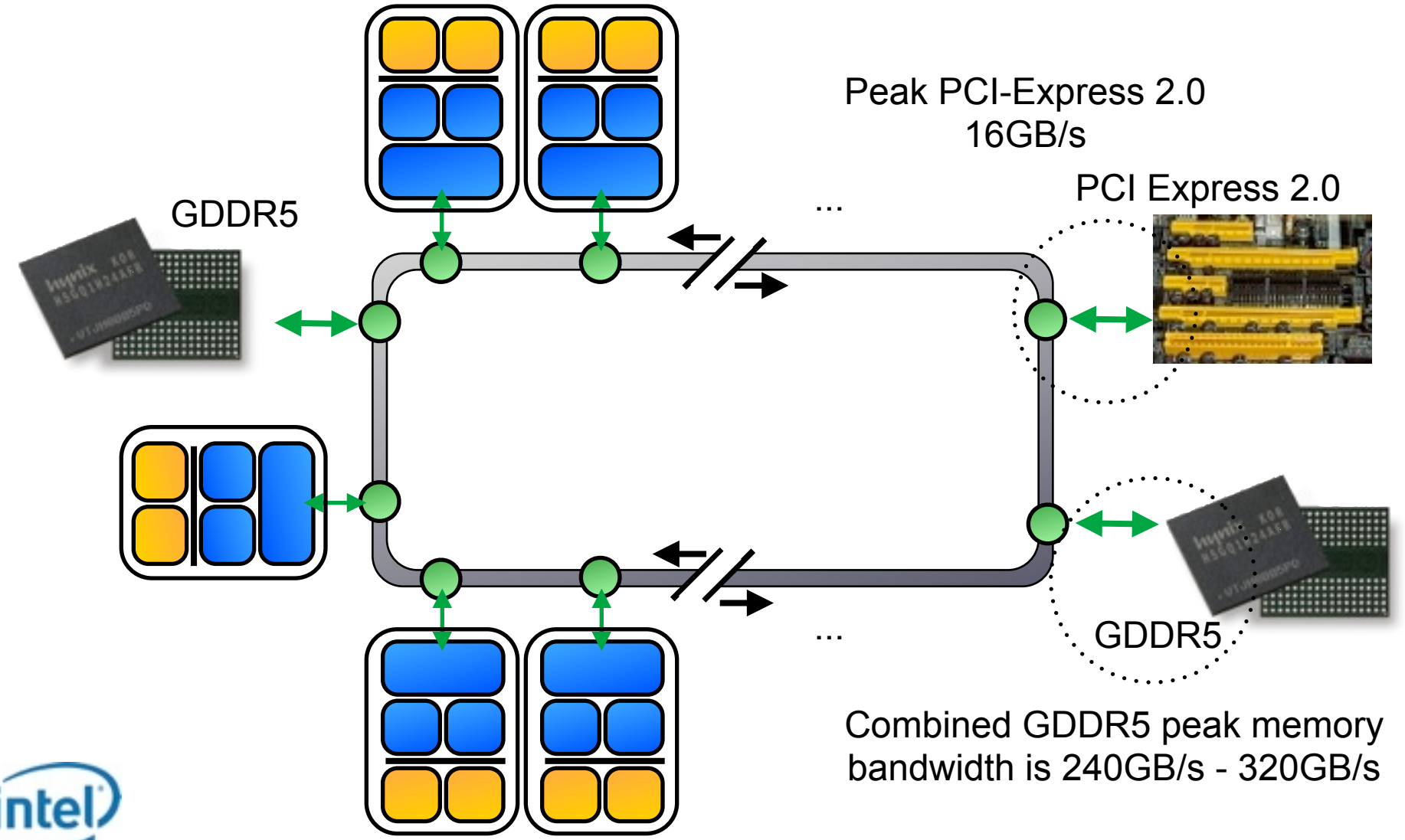
Sandia National Laboratories

- This talk is going to discuss:
  - Intel Xeon Phi ("MIC") Knights Corner
  - AMD APU Fusion (hybrid CPU and GPU)

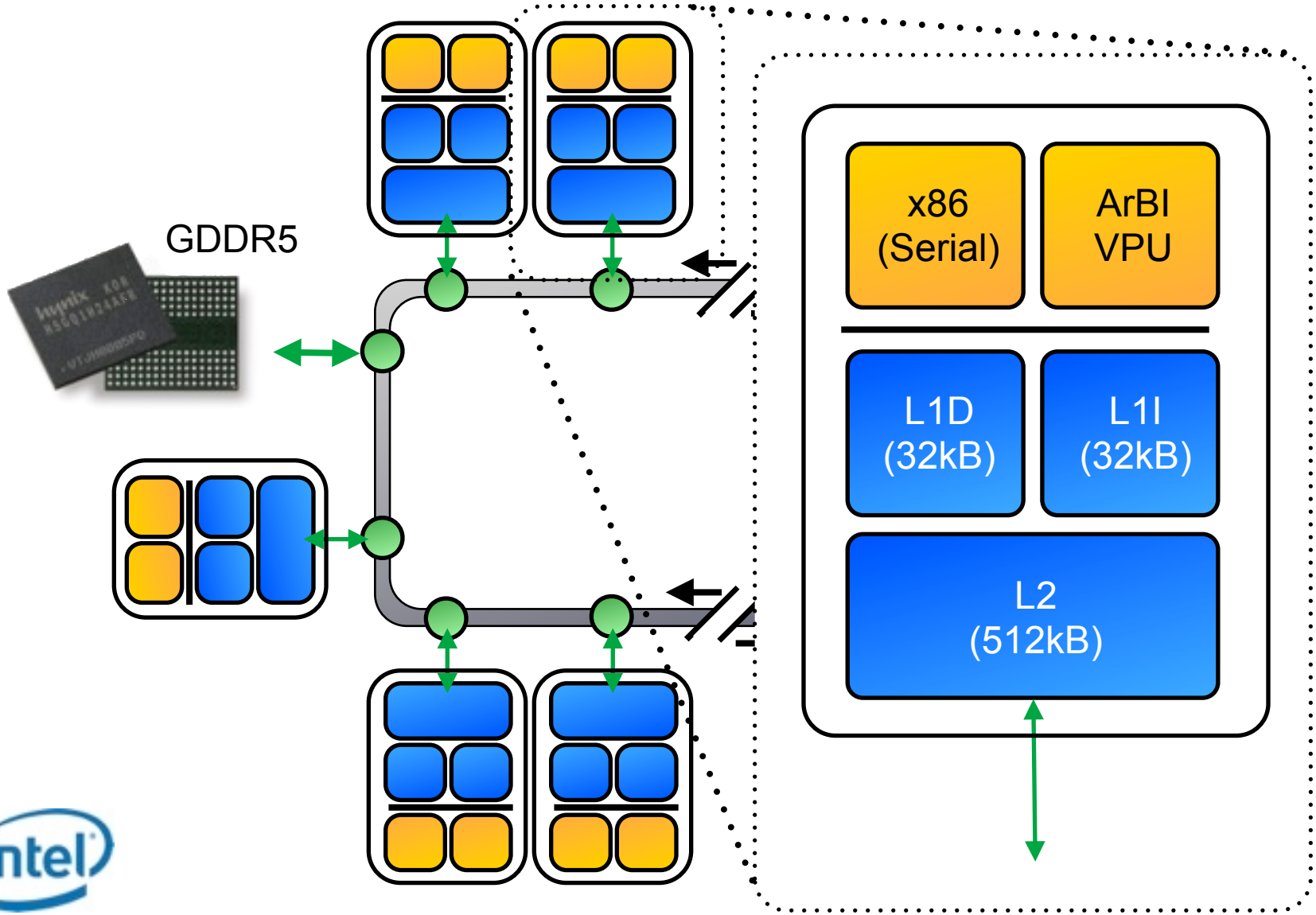
- These are sentinels for the future of computing
  - Large increases in parallelism - threads and vectors
  - Different memory technologies
  - Different vendor choices in level of processor power
    - Some vendors think more, smaller cores are better
    - Some vendors thing medium, more power cores are better
    - We have to begin to think about both - and in the end make our choice

Deep Dive on Intel Xeon Phi

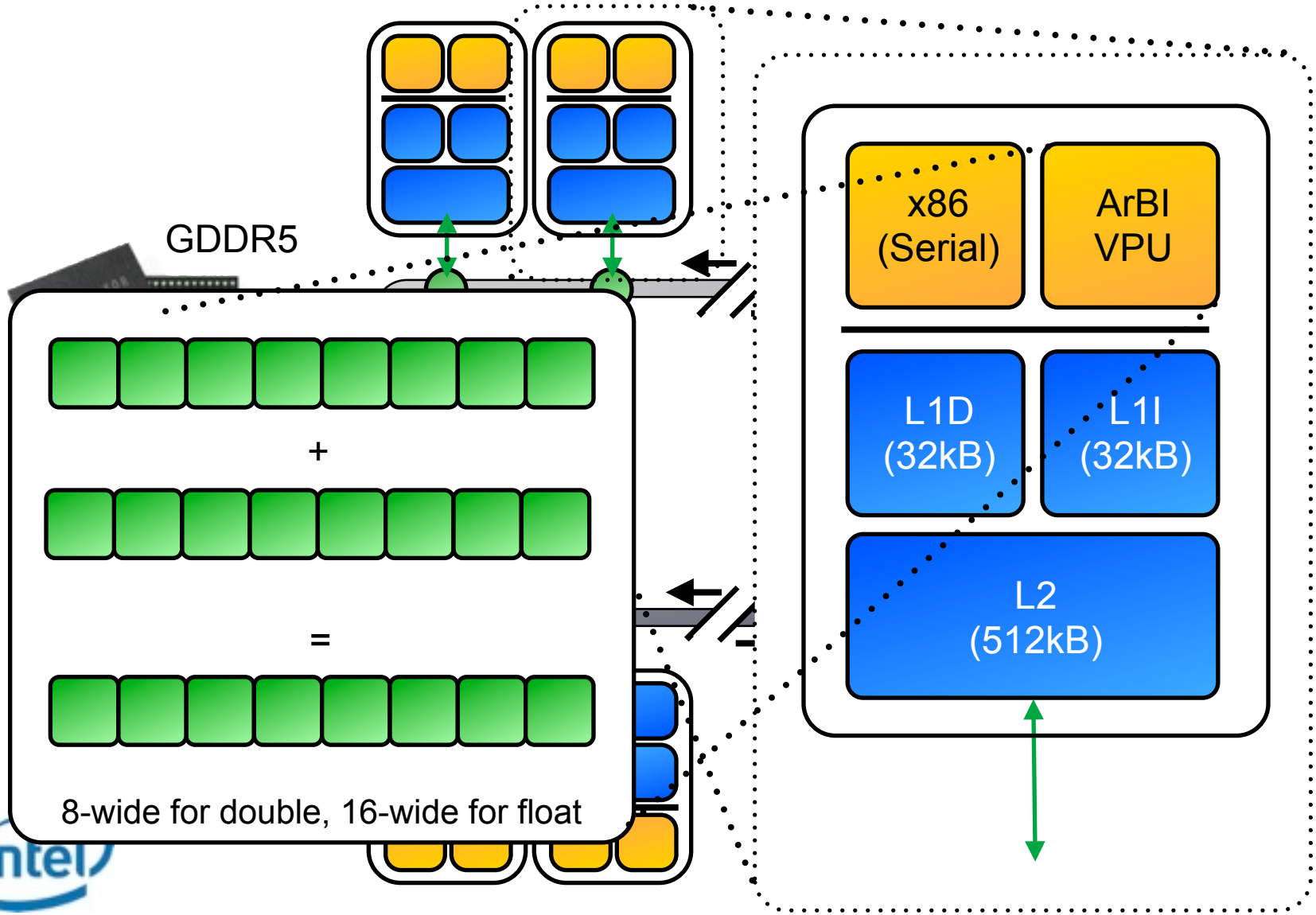
Intel MIC Architecture



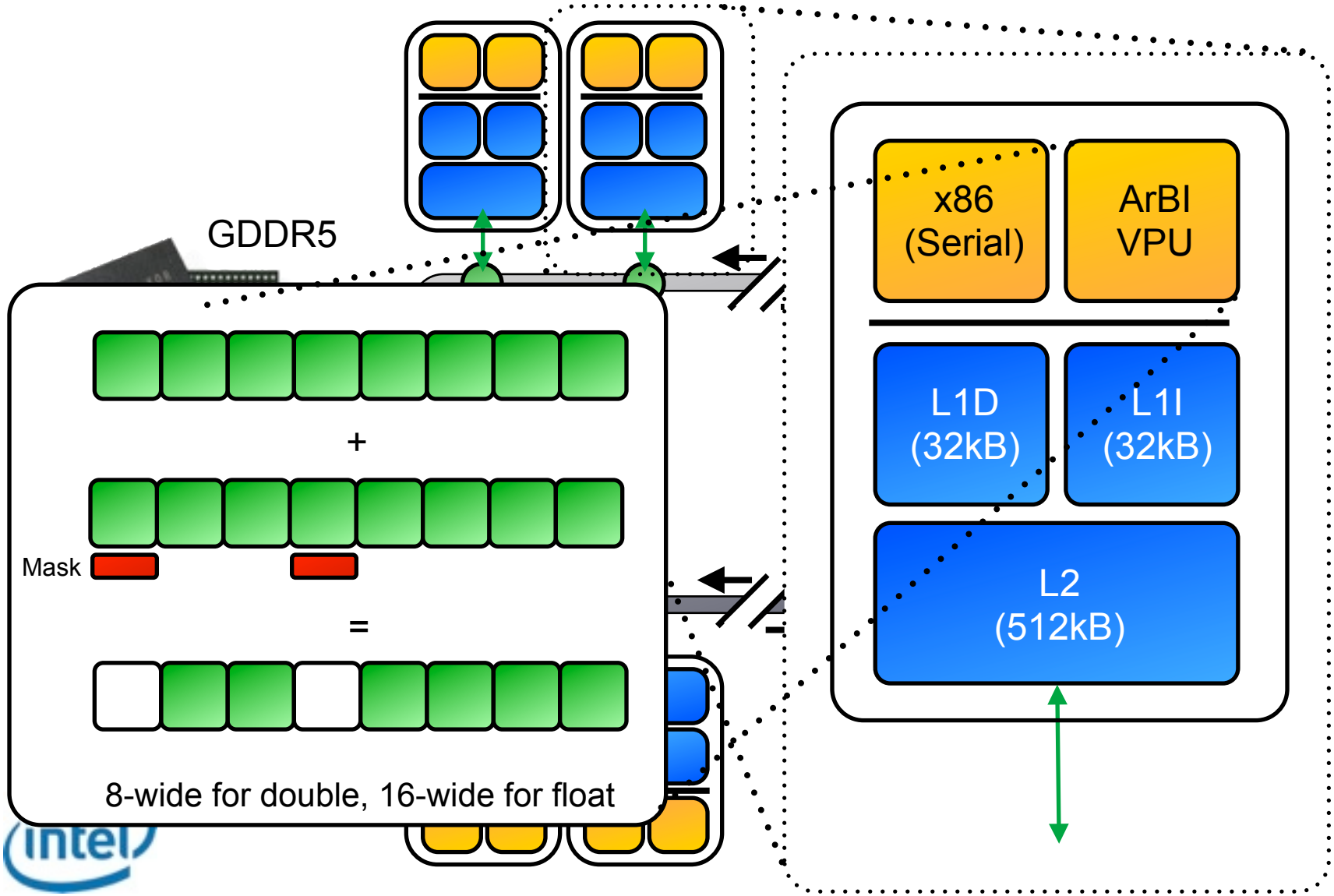
Intel MIC Architecture



Intel MIC Architecture



Intel MIC Architecture



Intel MIC Architecture

Sandia National Laboratories

- Vector Processing Unit ("VPU")
  - 512-bit = 8 doubles, 16 floats
  - Vector Gather and Scatter support (including gather prefetch to L2)
  - Cross-lane operations - reduction operations within VPU register
- x86 cores execute only basic x86
  - Will not permit binary compatibility with Xeon hosts
  - Four-way Hyper-Threading ("Third generation")
    - Single thread can issue at best every second clock cycle
- L2 caches are distributed and directory managed
  - Reduces traffic on the ring bus but *can* increase latency

Compton Specification

Sandia National Laboratories

- Cards used in Compton are pre-production B0/B1 cards
  - Intel will offer the 5110P (Q4'12) and 3100 (Q2'13)
  - We will switch to use C0-stepping cards soon (faster clock, more mem)

- Specifications:
  - 57-cores, 4-way Hyper-threaded = 228 threads @ 1.1GHz
  - 6GB GDDR-5 memory running peak of approx. 240GB/s
  - 1 TFLOP/s of double precision peak
    - Intel have demonstrated a KNC running LINPACK at 1 TFLOP/s
  - 225 - 300W TDP (includes memory, PCIe I/O, co-processor *etc*)

- Dual cards per node, one host Sandy Bridge E5 per Knights Corner (highest PCIe bandwidth)

Programming Knights Corner

Sandia National Laboratories

- Knights Corner has been designed to be compatible with traditional Xeon host processors (in so far as programming)
  - This does not mean the same code is always performant
  - This does not imply binary compatibility
  - General rule is code which performs well on Xeon Phi works well on Xeon, but not necessarily the inverse
- Programming models:
  - Threads, threads and more threads!
    - Intel Cilk Plus, Intel Thread Building Blocks, pthreads, OpenMP
  - MPI
    - On-card and between cards

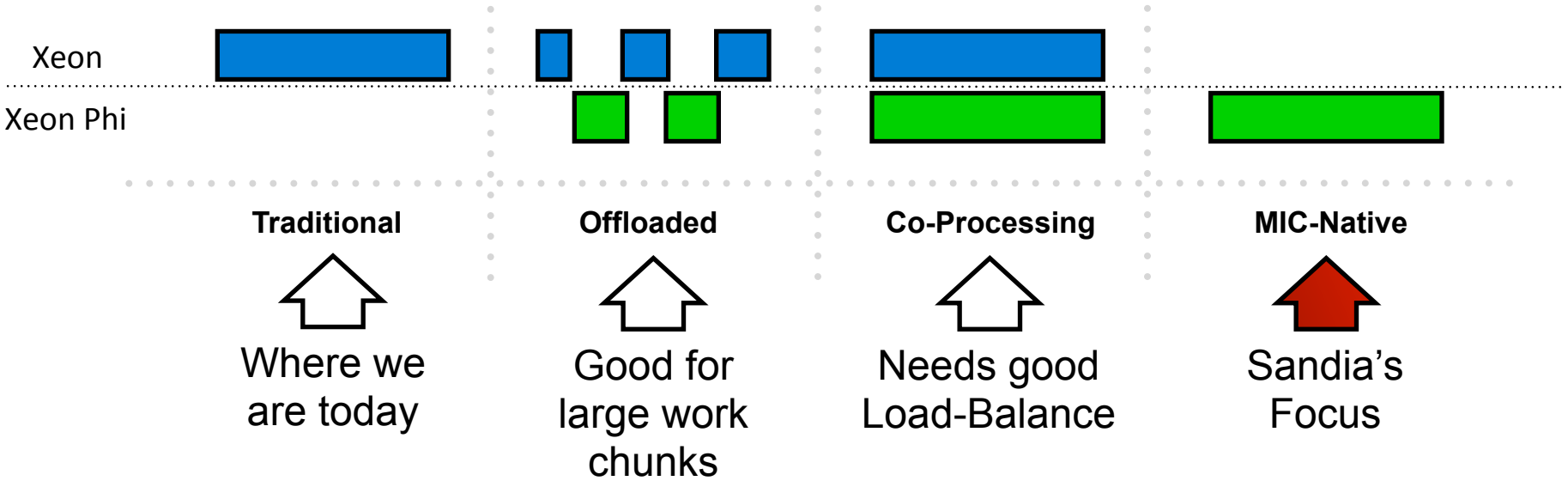
Programming on Xeon Phi

Sandia National Laboratories

- Programming on Xeon Phi is based mainly around threads and vectors
  - Let the compiler work the vectors out, see when it does not (it warns you)

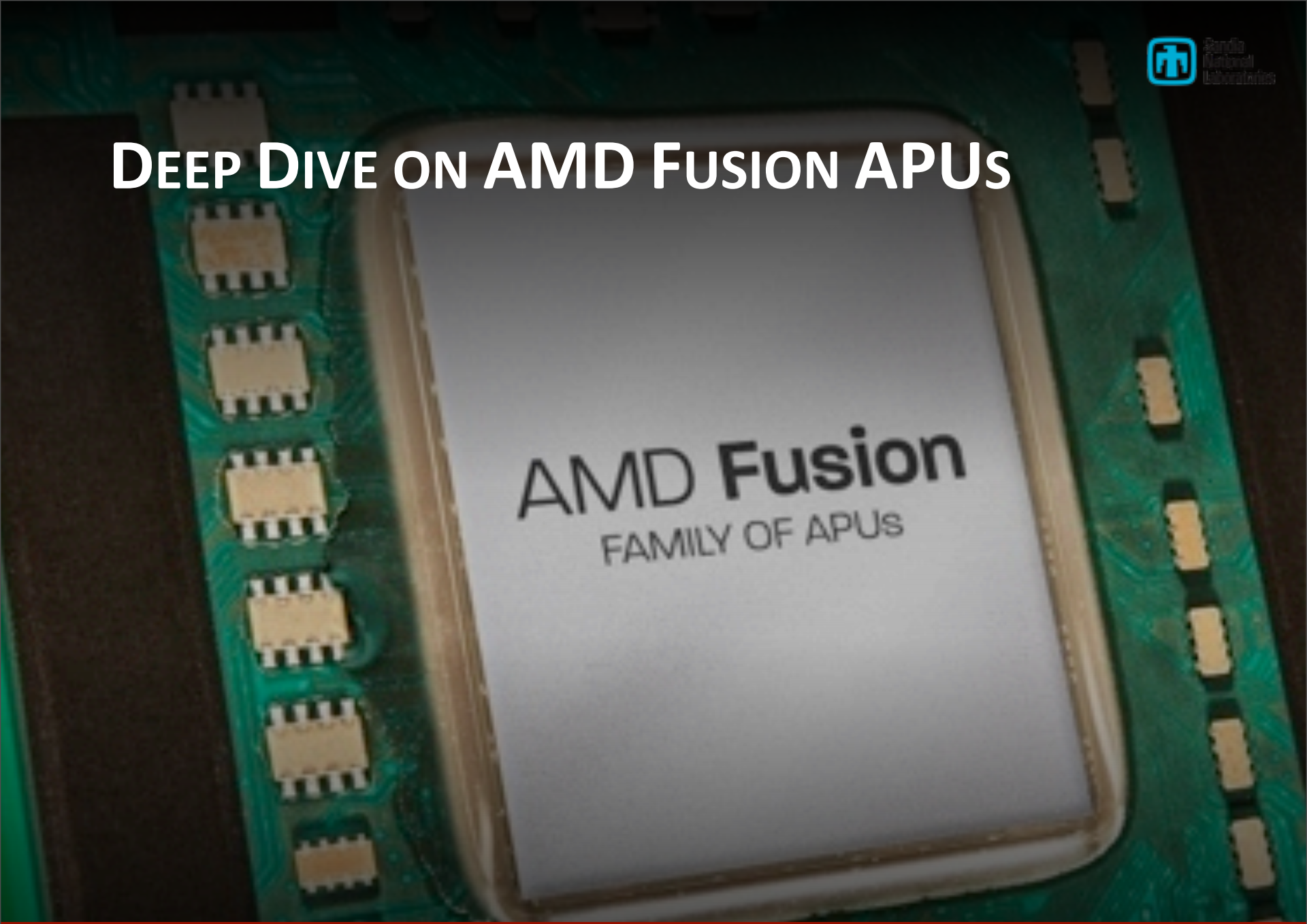
- Rich choice of threading
  - Intel Thread Building Blocks (“TBB”)
  - OpenMP Directives (Parallel-For and Task-based)
  - Intel Cilk Plus
  - pthreads
  - Sandia’s qthreads library

Where to run what?



- Knights Corner functions as a node (think node within a node)
  - SSH directly to card, scp files etc (full TCP stack)
  - Run `main(int argc, char* argv[])` directly on card
  - No need for host interactions (get the Xeon out of the way)

Deep Dive on AMD Fusion APUs

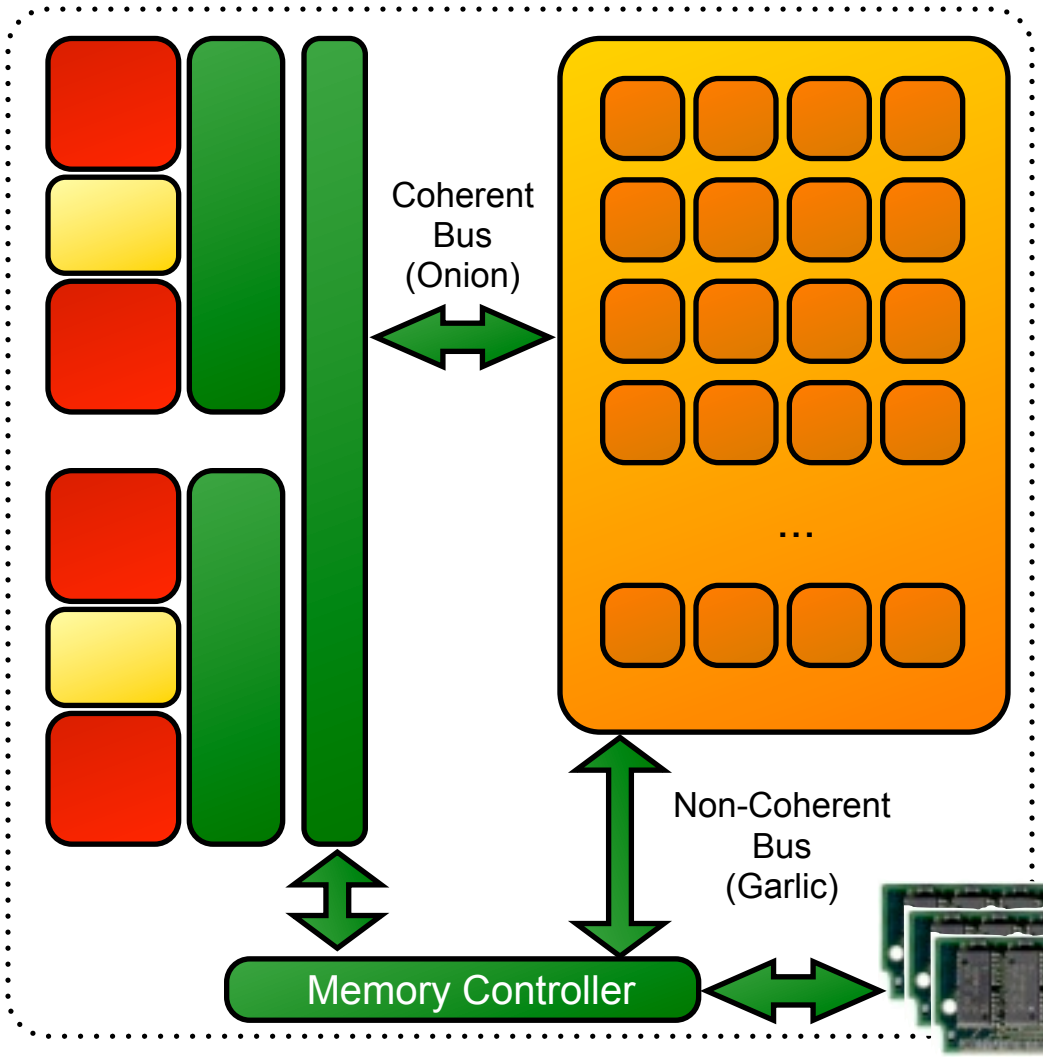


AMD Fusion (APU)

Sandia National Laboratories

CPU

2 x Core Pairs
3.8GHz



GPU

HD 7660D
800 MHz
384-cores

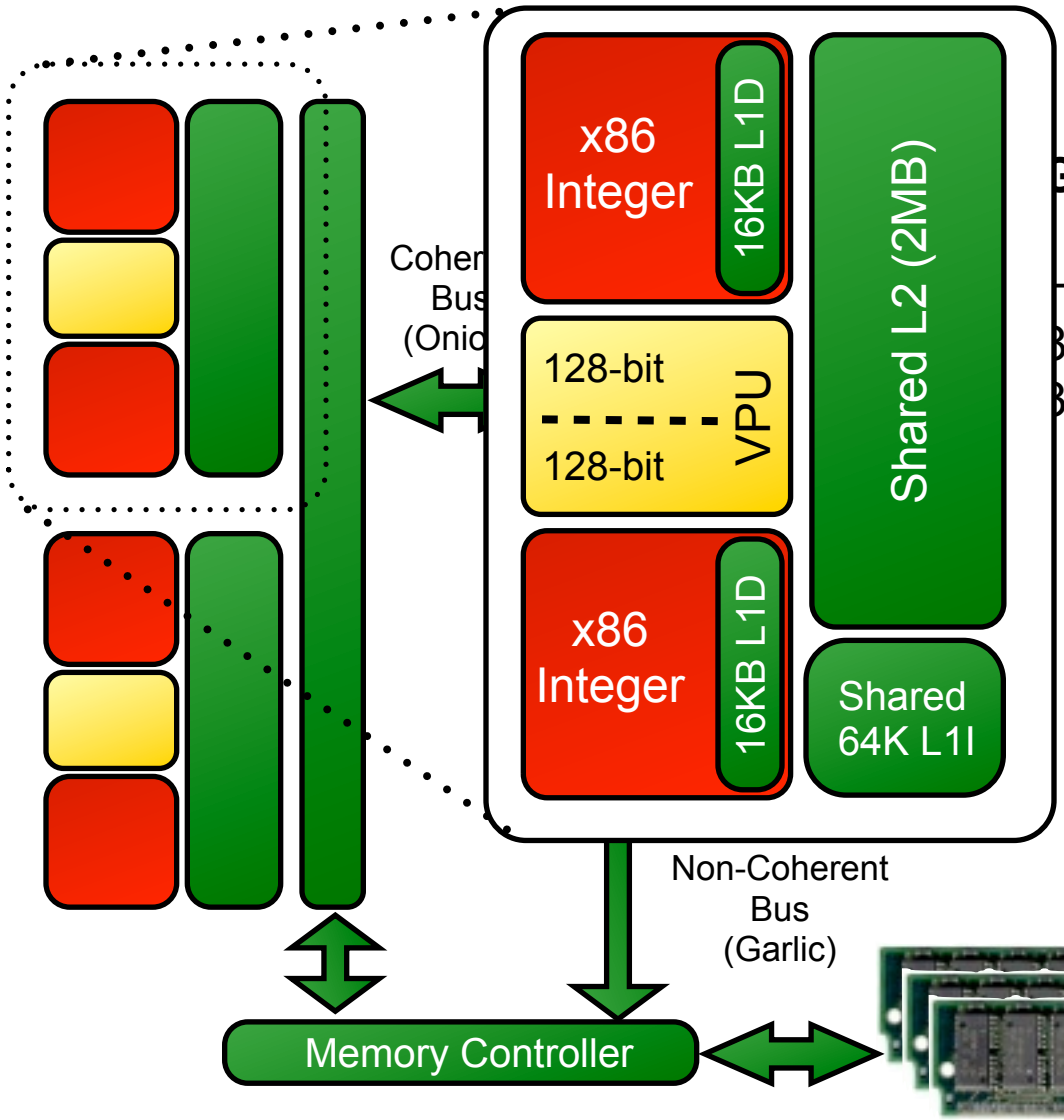
AMD Smarter Choice

AMD Fusion (APU)

Sandia National Laboratories

CPU

2 x Core Pairs
3.8GHz



GPU

HD 7660D
800 MHz
384-cores

AMD Smarter Choice

Teller Specification

Sandia National Laboratories

- 104 nodes of 1 x A10 APU
  - 16GB RAM per node
- InfiniBand QDR Interconnect
- Integrated by Penguin Computers
- Designed for power measurement experiments

Programming APUs

Sandia National Laboratories

- CPU components of APUs run traditional x86 binary
  - AVX compatible (close to Sandy Bridge implementation of AVX 1)
  - FMA-4 implementation (currently only AMD)
- GPU components run OpenCL
  - OpenCL so far has to be written by hand - time consuming process
  - Not currently C++ compatible (non-standard extensions available)
  - CAPS and PGI have support for OpenACC in the latest compilers
- "Zero-copy" is enabled by driver switches and environment variables (but frequently causes issues on the nodes)

Memory Constraints in APUs

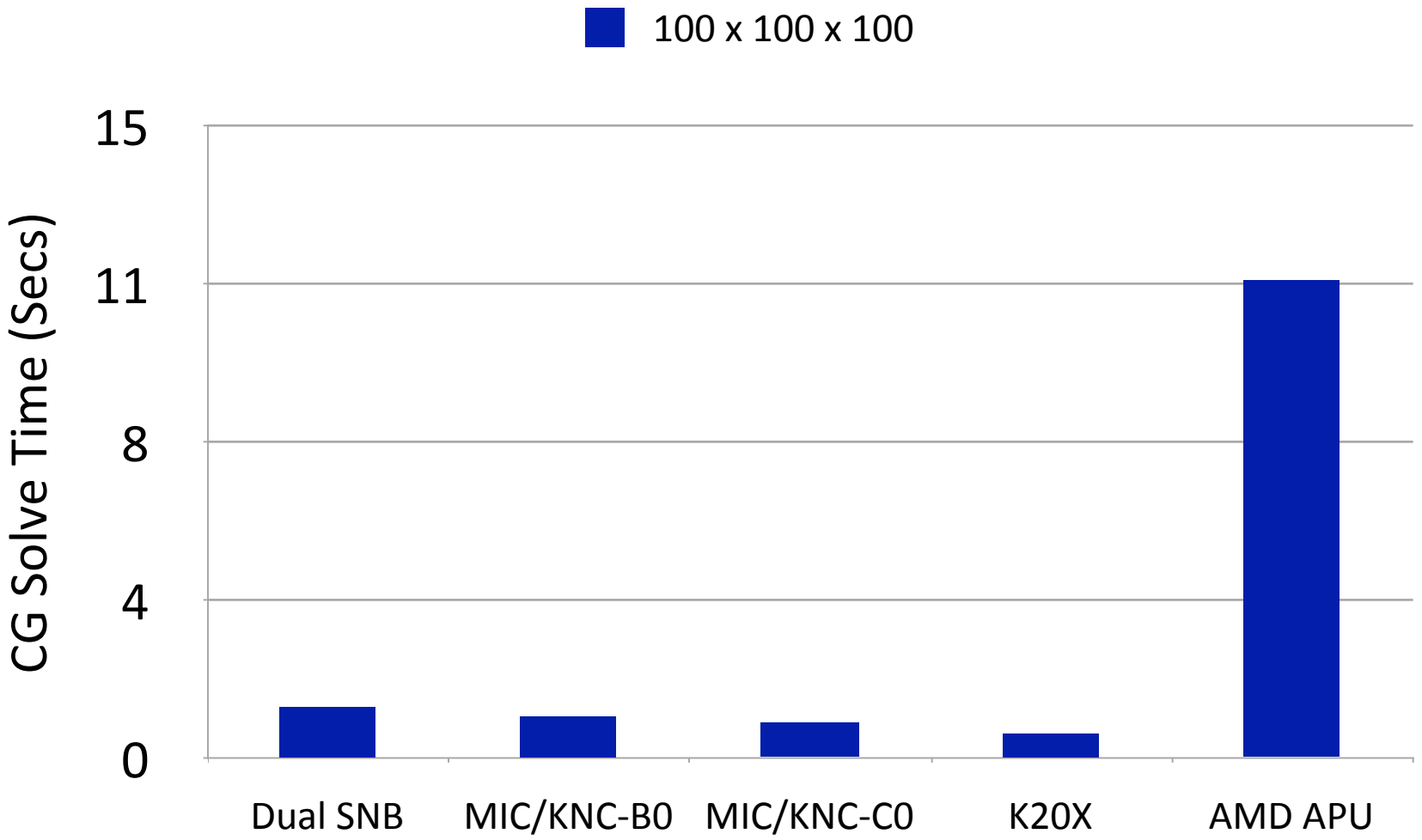
Sandia National Laboratories

- AMD intends to provide ‘zero-copy’
  - *Pointer-as-a-pointer* - you can allocate memory on CPU and use on GPU
  - Eventually coherent (by end of kernel)
- Current work is based around driver
  - Limitations are 32-bit, zero-copy bandwidth is not being achieved on our systems - implies problems
  - Current limit is approximately 80% of 256MB of BIOS allocated memory
    - Way too little for our algorithms, made worse by OpenCL 50% allocate
    - High cost associated with streaming blocks to GPU
- Eventually APU memory subsystem is still DDR3
  - STREAM Triad peaks at 2 threads, 13.1 GB/s
  - lmbench estimates 17.58GB/s read, 7.53GB/s write

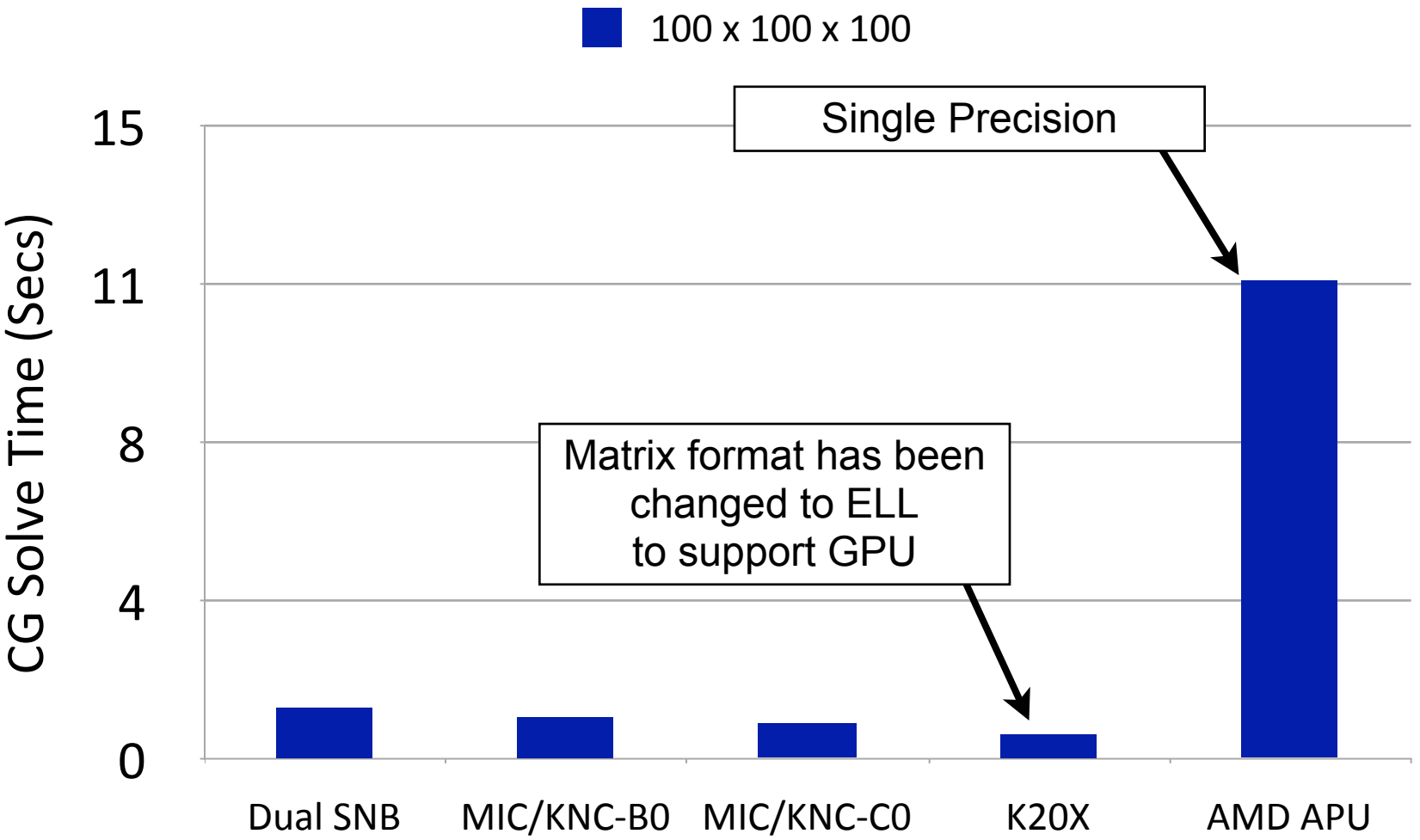
Sandia National Laboratories

PERFORMANCE RESULTS

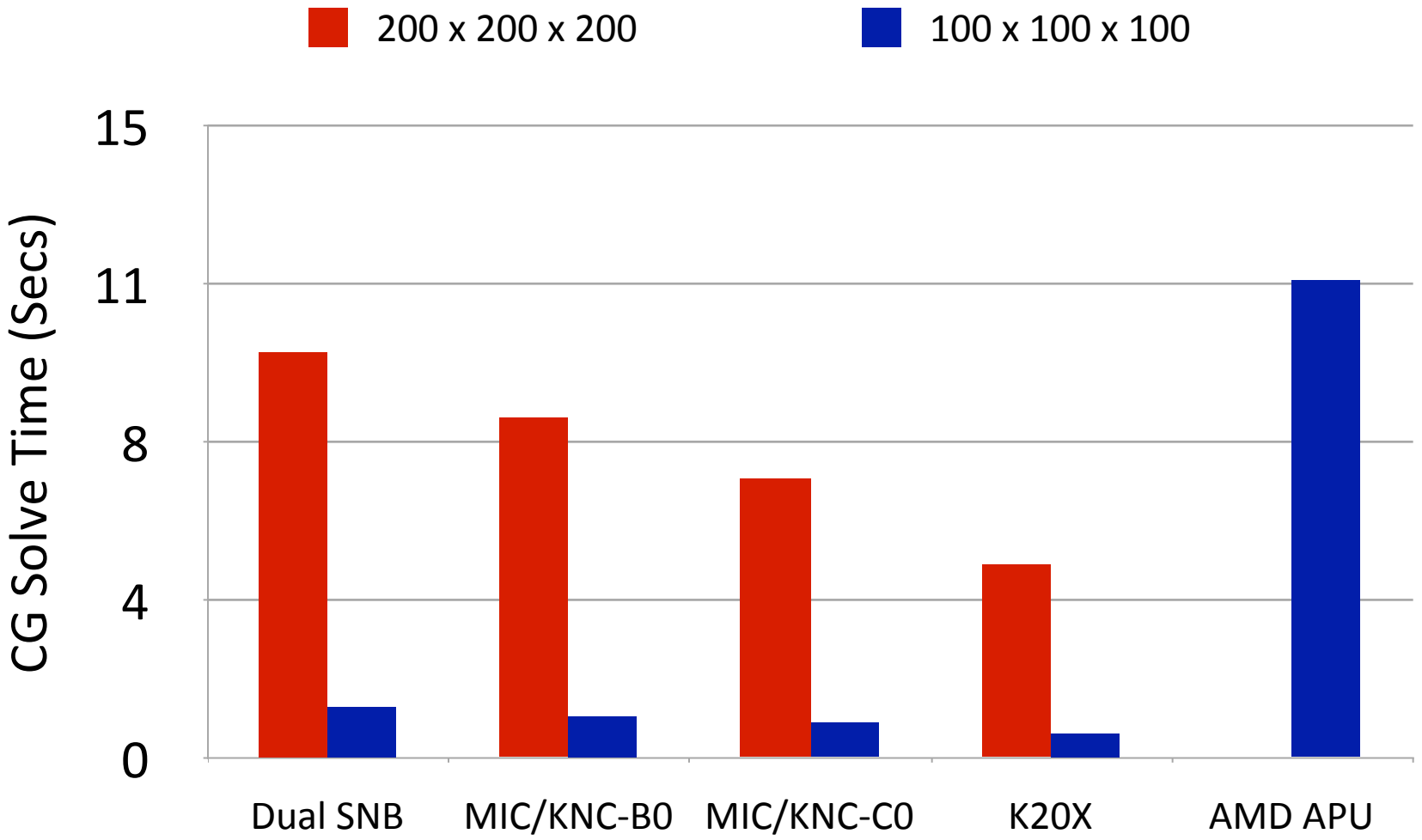
Performance



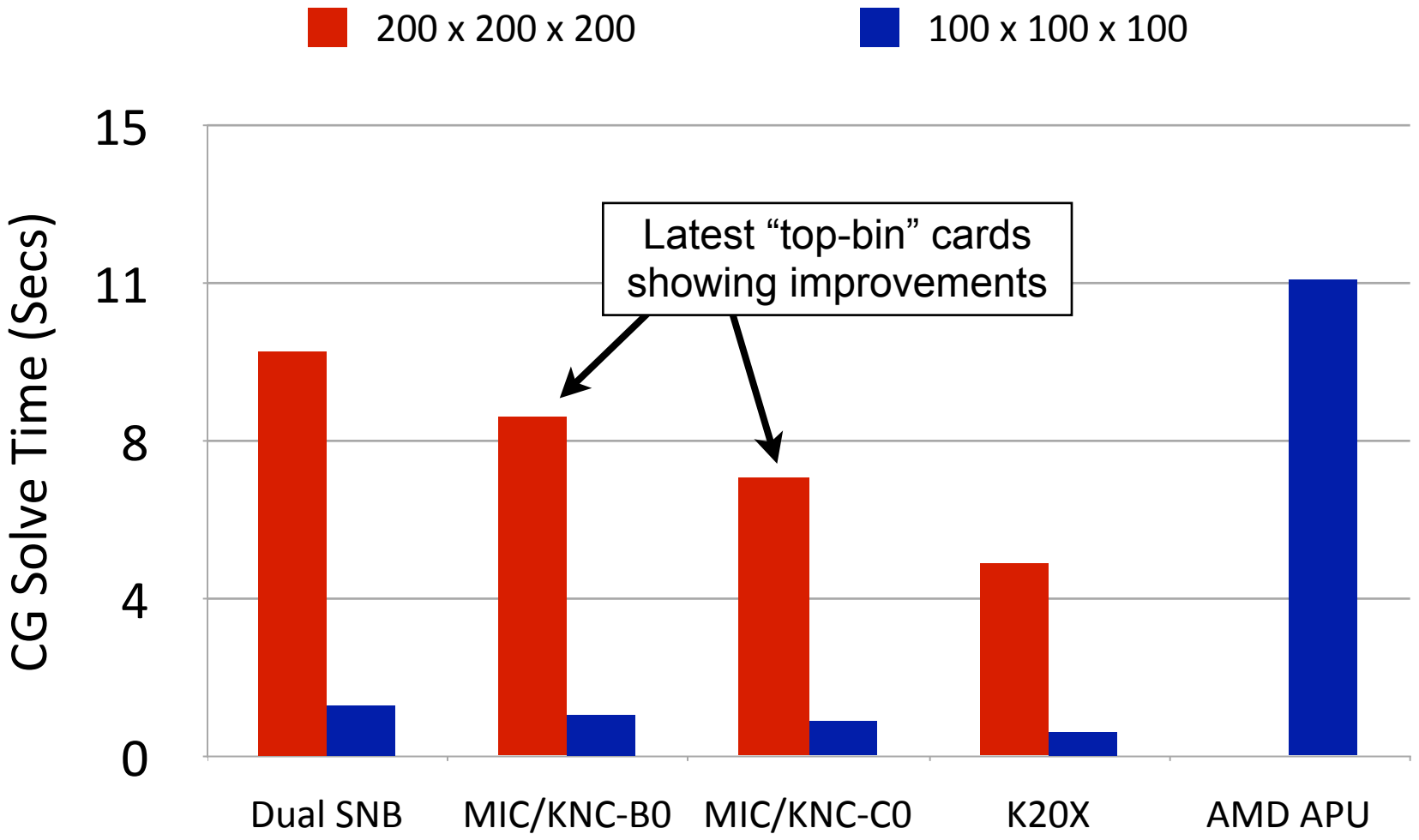
Performance



Performance

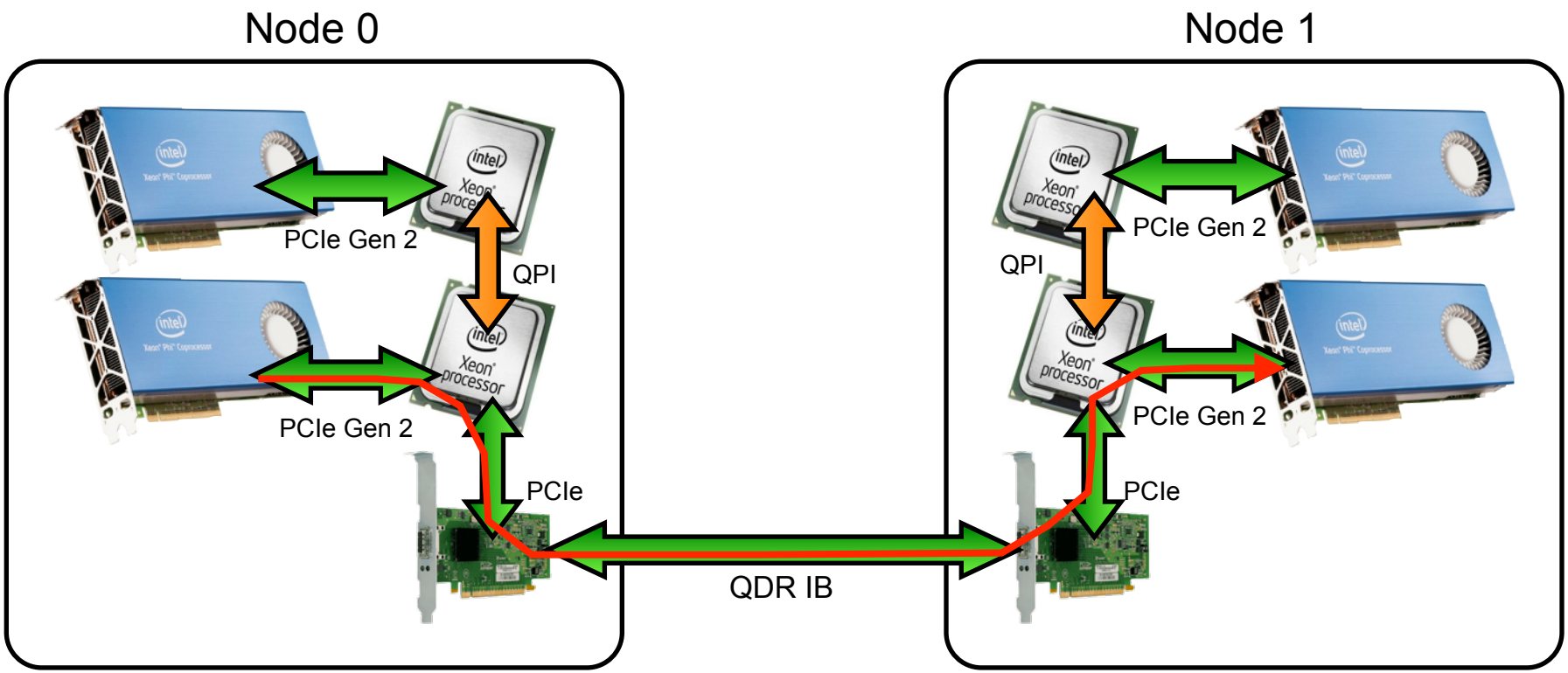


Performance

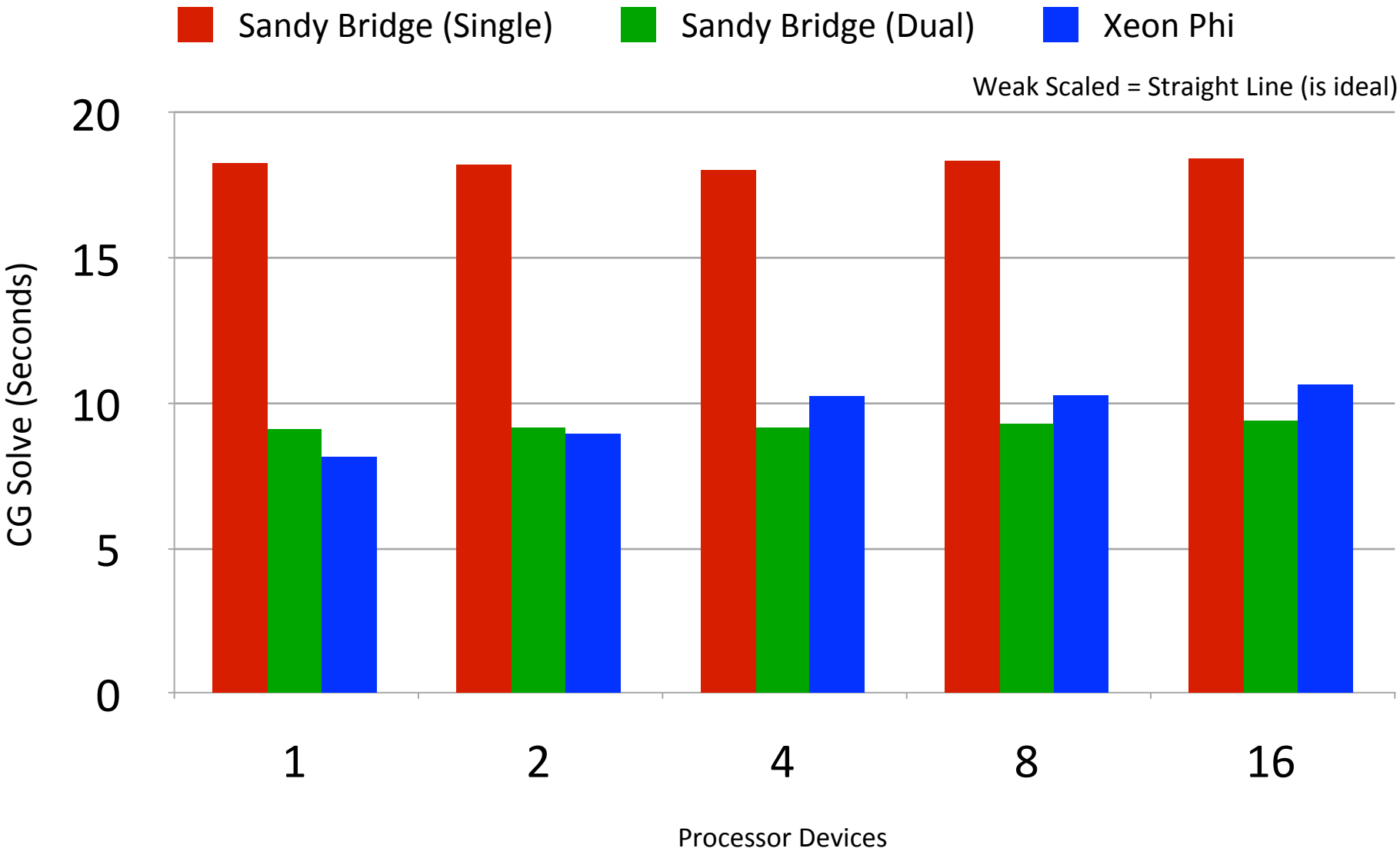


Many MICs in Parallel

- Because we have used an “MPI+X” approach we can run on multiple MIC cards



Many MICS in Parallel



What is coming to a test bed soon?

Sandia National Laboratories

- Intel's latest Ivy Bridge Xeon processors
  - 10 - 12 cores per socket
- Investigating options for ARM processors
  - The same processors found in your iPhone and iPad
- Potential for DSP processors
  - Perform some mathematics operations incredibly fast
- FPGA-based acceleration and IBM POWER7

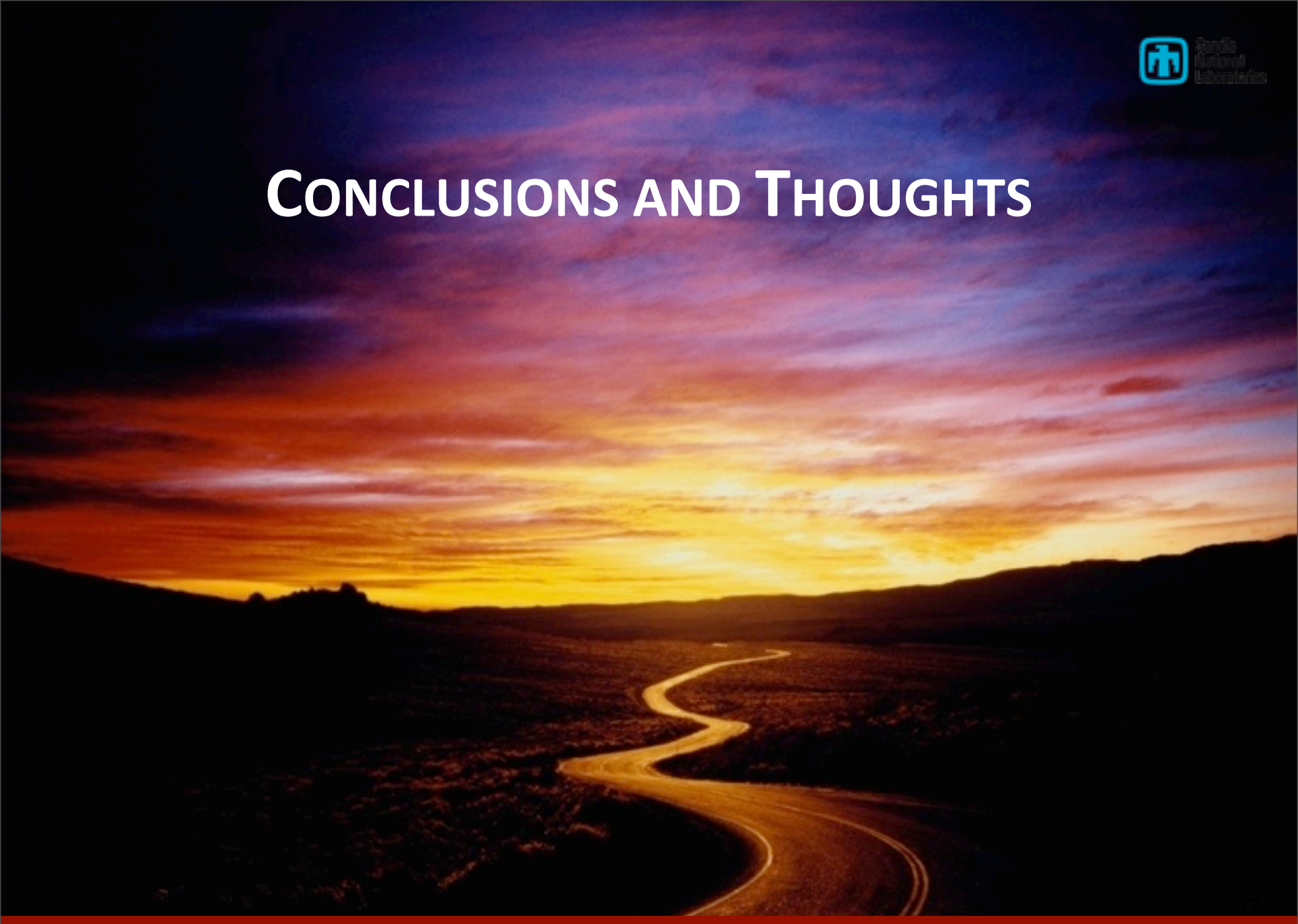
Snapshot of Technologies (2013)

Sandia National Laboratories

	MPI	Intrins.	CUDA	OpenAcc	OpenCL	Kokkos Array	Cilk+	OpenMP		TBB	ArBB/ CEAN	qthreads	pthreads	MKL/ Math Lib.	Adv. Lang.	DSLs
								S	T							
NVIDIA (GPU)																?
AMD (CPU)																?
AMD (APU)		?														?
Intel (CPU)																?
Intel (MIC)																?
IBM (BG)										?						?
ARM																?

miniFE																
miniMD																
miniGhost																
LULESH																
S3D																
CoMD																

Conclusions and Thoughts



Trends in Computing

Sandia National Laboratories

- Slower processor clock rates
  - Means we are getting fewer scalar calculations done per second for a single unit of computation
- Vast increases in parallelism
  - We are getting more done across many compute units per second if we can work out how do it
- Memory operations are relatively more expensive
  - Storing results in main memory is becoming more expensive (in energy and time) meaning we need to work out how to re-use results and avoid unnecessary memory transfers

Trends in Algorithms

Sandia National Laboratories

- Threading and Vectorization
  - Towards much higher levels of parallelism, algorithms are more likely to stand the test of time if they can work at varying levels of threads and vector widths
- Reduced communications
  - Where possible algorithms will switch to use fewer communications, expensive for power and time
- Higher levels of asynchronicity
  - Reducing the need for algorithms to synchronize will create higher levels of utilization and may lead to reduced power consumption

Exascale by 2018?

Sandia National Laboratories

- Exascale is going to be very challenging
  - Problem of properly integrated machines is not similar enough (yet) to Google and Facebook to be directly relevant to industry
  - We must use commodity where we can to drive down cost
    - Problems of reliability
- Real challenge won't be in the hardware
  - Applications are going to have to change
  - Large legacy code bases are going to be very problematic
    - Particularly for labs like Sandia, Los Alamos and Lawrence Livermore than need to get science done to meet funding
  - In the end it will come to down to tradeoffs in parallelism, performance and possibly in accuracy

What's next?

Sandia National Laboratories

- Climate community is already talking about needing ZetaFLOP/s
  - Modeling of climate change and risks will continue to push our demands
- What happens at the high-end eventually bleeds down to the workstation and computer (and maybe even your iPad)
  - You can get TFLOP/s in your workstation already
  - Today's iPad is as powerful as old Cray machines
- Even "lower" end computing will benefit and may need new algorithms
  - One in a generation opportunity for us to enhance and influence the future of technology

Want to try out the test beds?

- Speak to your manager
  - WebCARS access for any Sandia employee
  - May need extra details on your FN-plan (Foreign Nationals)
- HAAPs Confluence Page to help you get started
- Access for every one
- Feedback always welcome

Sandia National Laboratories

Exceptional service in the national interest

sdhammo@sandia.gov