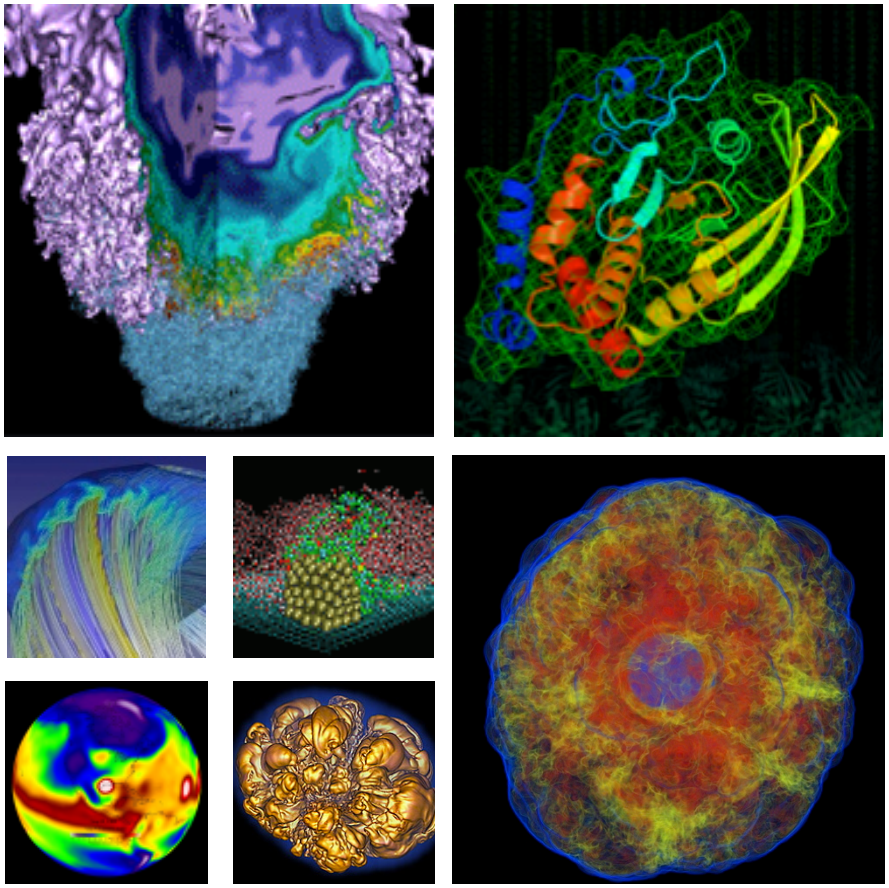
NUG Monthly Meeting



Richard Gerber Helen He, Zhengji Zhao, Chris Daley

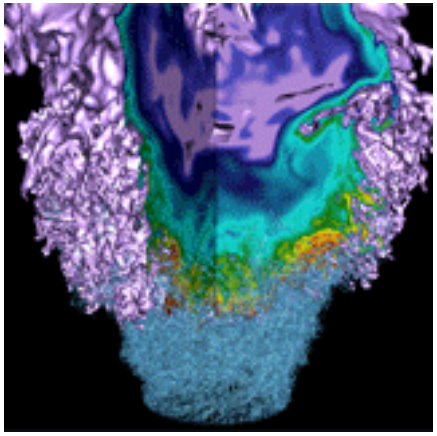
NUG Monthly Meeting
October 2, 2014

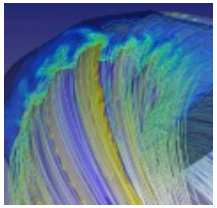
Agenda

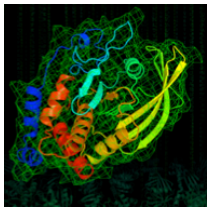
- Hopper and Edison Status Updates
- ERCAP (Allocations) Update
- Queue Committee Topics
- NUGEX Elections
- NUG 2015 Meeting Planning
- Mini-Seminar: Steps to improve the performance of a hydrodynamics application on the Intel MIC architecture

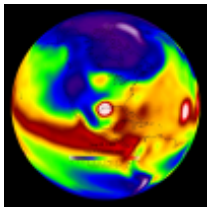
Edison Update

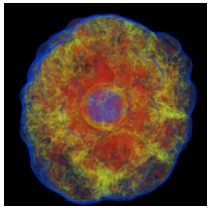
Zhengji Zhao, User Services

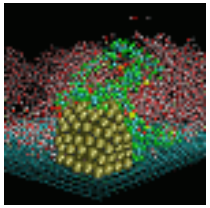












Edison Updates

- **9/25 7:30 PDT - 9/28 1:30 PDT DIMM replacement**
  - A total of 14,461 DIMMs were replaced, and tested; Memory is still clocked at 1600 MHz until all nodes are upgraded.
  - /scratch1 and /scratch2 were upgraded to cs_1.3.1
  - Bash security patch installed
  - GPFS upgraded to 3.5.0-20e6
- **Dedicated science runs from PI CS Chang's group**
  - 9/24 7:30 -9/25 7:30 PDT
  - 9/28 1:30 PDT-3:30 PDT
- **Maintenance on 10/7/14, Tuesday 8:00 -12:00 PDT**
  - System software maintenance

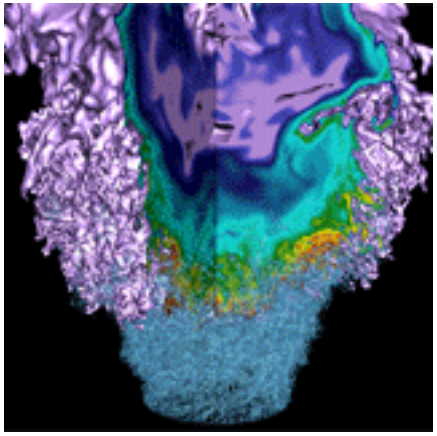
Thanks all for your patience when Edison went through a long maintenance.

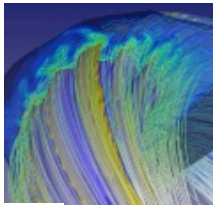
Edison Updates

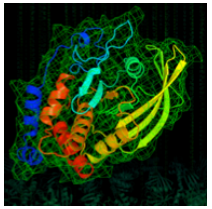
- **CDT (Cray Developer Toolkit) 1.19 was set to default**
  - cray-petsc/3.5.1.0 bug discovered – use cray-petsc/3.4.4.0 instead
  - Issues with cray-hdf5-parallel/1.8.13 module - use cray-hdf5-parallel/1.8.12

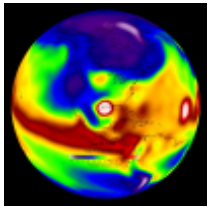
Hopper Update

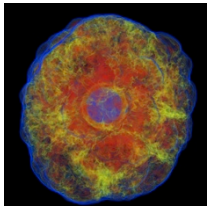
Helen He, User Services

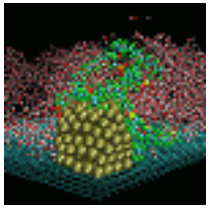












Inactive OST (Object Storage Target)

- An OST failed on /scratch2.
- An OST is one "unit" of the Lustre file system; /scratch2 has 156 OSTs.
- A file may be stored completely or partially (striped) on a single OST.
- If any part of a file is on this inactive OST, it will not be accessible.
- Individual users have been contacted with the list of their affected files.
- Cray and the storage system vendor are trying to recover files on the failed OST, but the outcome is uncertain.

Some Jobs Running Slowly

- Jobs using GPFS file systems (such as /project, /global/scratch2, /global/homes) or shared libraries may be affected.
- The root cause is not fully understood.
- A full system reboot last Friday did not resolve the problem.
- A *critical* bug has been opened with Cray.
- NERSC and Cray staff are still investigating.

Miscellaneous Issues

- **Mail notification is broken for certain users.**
  - Email appears to come from adm@*something*.cluster
  - Some of your mail servers reject this address
  - Possible workaround: use #PBS –M <gmail or other address> in your batch script.
  - Fix to come soon.
- **GPFS upgrade broke cmake on /project and /home file systems**
  - cmake worked on Lustre file systems, which caused a lot of confusion!
  - A patch has been applied yesterday, so cmake works now.

Upcoming System Maintenance, Wed., Oct 8

- **The scheduled maintenance is for software patches and some file systems work, including work to bring the bad OST back online.**

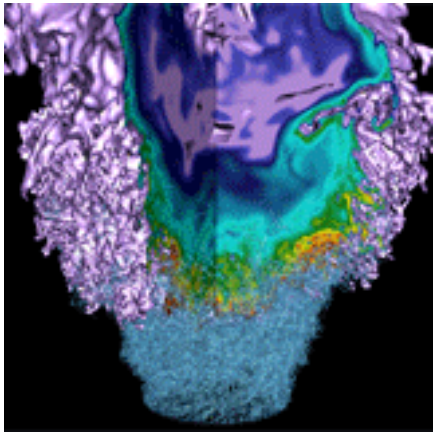
- **We hope to have more knowledge about DVS/GPFS slowness issue and possible patches soon so we can apply a fix during Wednesday's maintenance.**

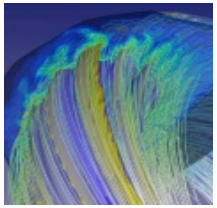
Serial Queue on Hopper and Edison

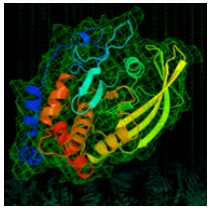
- Official announcement is planned for next week.
- Serial queue allows multiple users, multiple executables to share a single compute node.
- You must use special commands such as "qsub.serial", "qstat.serial", etc. to manage jobs in the serial queue.
- More details on web site about using the serial queue.
- Submit any questions to consult@nersc.gov.

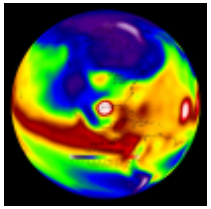
2015 NERSC "DOE Production" Allocation Requests

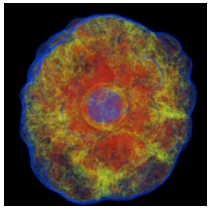
Francesca Verdier, Allocations Manager
fverdier@lbl.gov

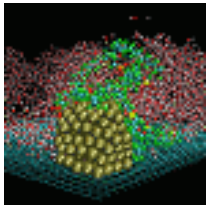










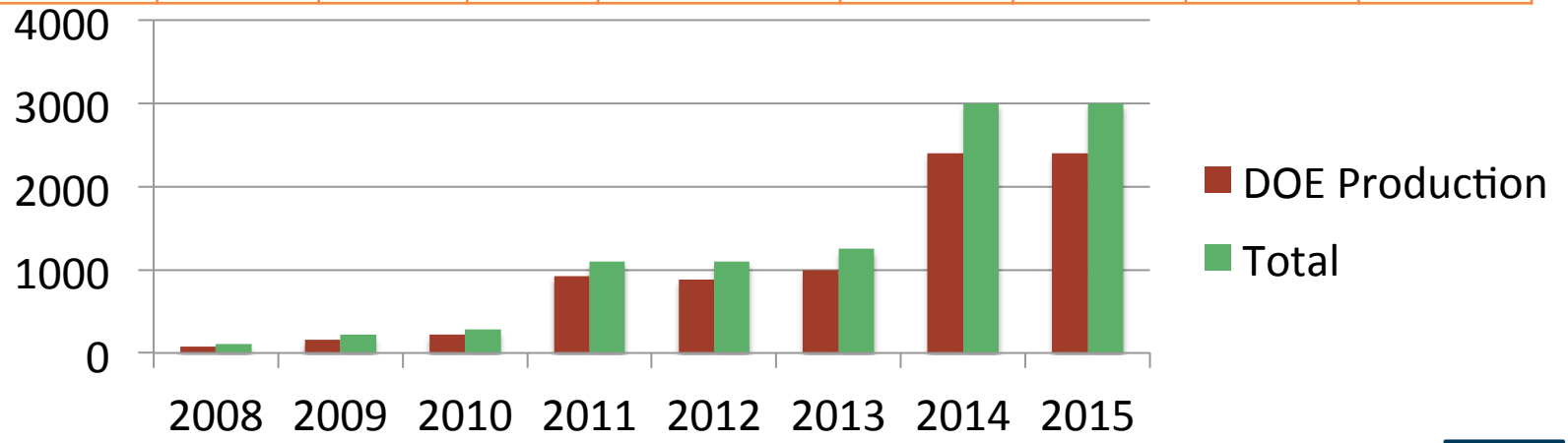


NERSC Allocation Programs

Allocation Type	% of DOE Allocation	Description
"DOE Production"	80% 2.4 B	Allocated by DOE Program managers in the six offices of science & SBIR. Applicants need to be DOE funded or show their work fits in the DOE mission.
ALCC	10% 300 M	ASCR Leadership Computing Challenge – a DOE program run by ASCR to promote areas of interest to DOE.
NERSC Director's Reserve	10% 300 M	In 2015 NERSC will not run the NERSC Initiative for Scientific Exploration (NISE) program. The Director's Reserve will be used to support NESAP - the NERSC Exascale Science Application Program, as well as other projects selected by the NERSC director.
Education, Startup	N/A	Small awards made by NERSC from "overhead time".
Sponsored	N/A	For hardware purchased with other funds.

MPP Allocations 2007 – 2015 (millions of Hopper-based hours)

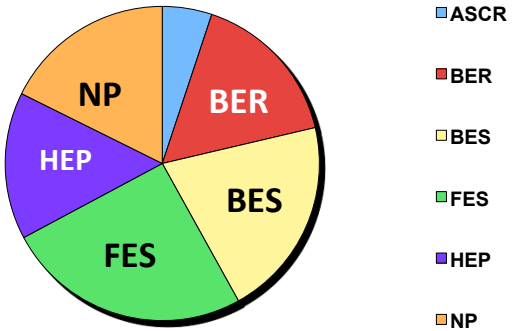
	2008	2009	2010	2011	2012	2013	2014	2015
DOE Production	73.1	160.5	224	922	880	1,000	2,400	2,400
INCITE	10.4	20	-	-	-	-	-	-
DOE reserve / ALCC	10.4	20	28	88.5	110	125	300	300
NERSC reserve / NISE	10.4	20	28	88.5	110	125	300	300
Totals	**104.4**	**220.5**	**280**	**1,099**	**1,100**	**1,250**	**3,000**	**3,000**



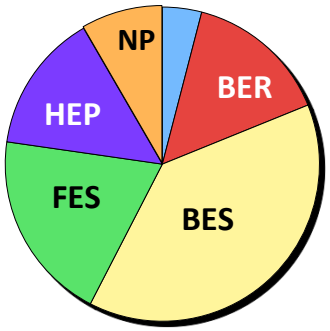
2015 MPP Requests (M hours)

Office	Requested (M)	Num	Office Charges 10/1/14 (M)
ASCR	196.1	64	73.9
BER	623.2	104	271.2
BES	791.7	294	708.1
FES	969.7	60	360.3
HEP	581.9	60	262.7
NP	678.6	44	153.0
SBIR	26.9	8	13.4
Totals	**4,492.3**	**634**	**1,842.7**

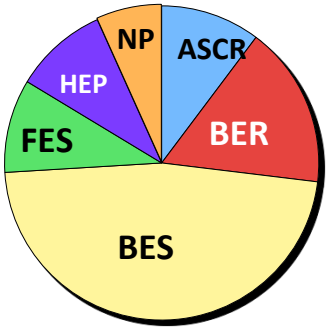
Percent of hours requested



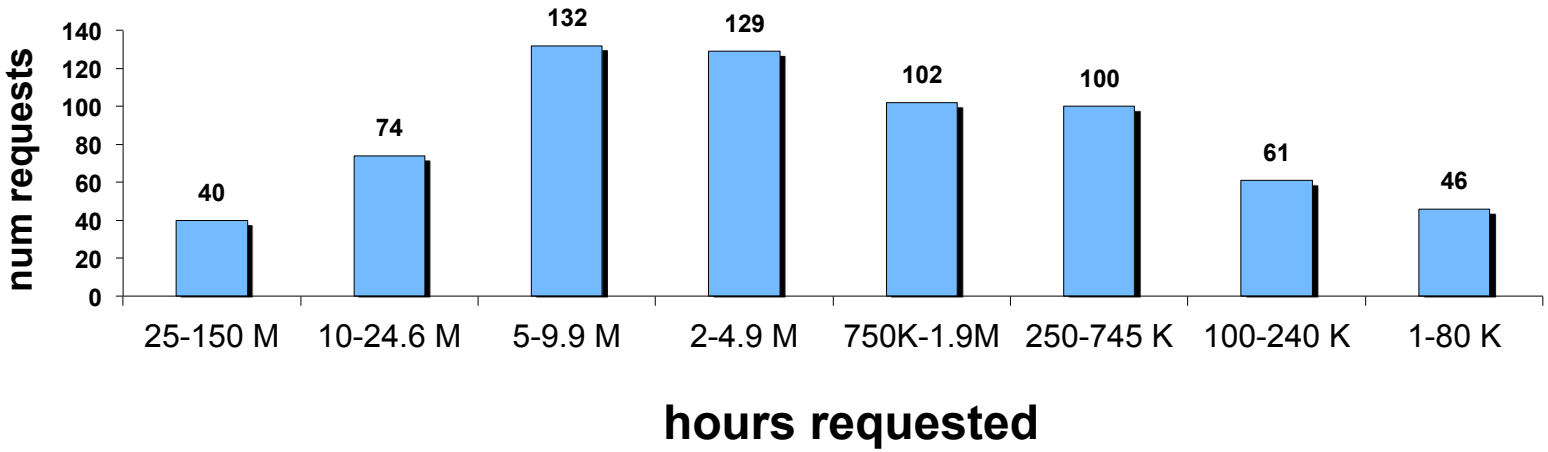
Percent of charged hours



Percent of projects



2015 MPP Request Distribution
(634 requests total)



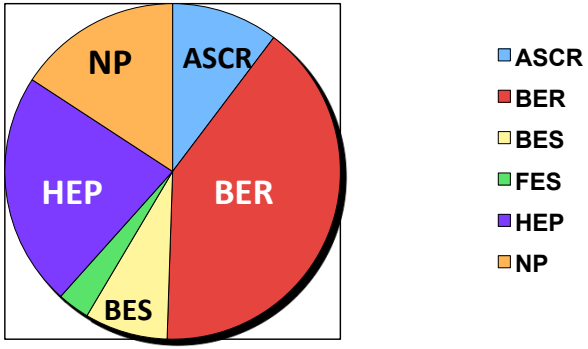
- **Largest requests:**
  - 150 M hours (Fusion)
  - 150 M hours (Fusion)
  - 125.3 M hours (NP QCD)
  - 120 M hours (NP QCD)
  - 110 M hours (NP Theory)
  - 3 requests for 100 M hours (Fusion, HEP Astro, HEP QCD)

- **10 projects requested 25% of total amount requested**
- **36 projects requested 50% of total**
- **109 projects requested 75% of total**

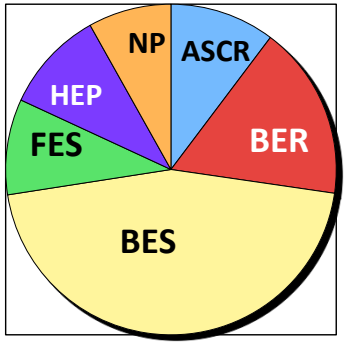
2015 HPSS Requests (M SRUs)

Office	SRUs requested	# projects
ASCR	37.3	67
BER	145.7	111
BES	29.0	295
FES	11.2	61
HEP	81.8	65
NP	57.2	53
SBIR	0.4	8
Totals	**362.8**	**660**

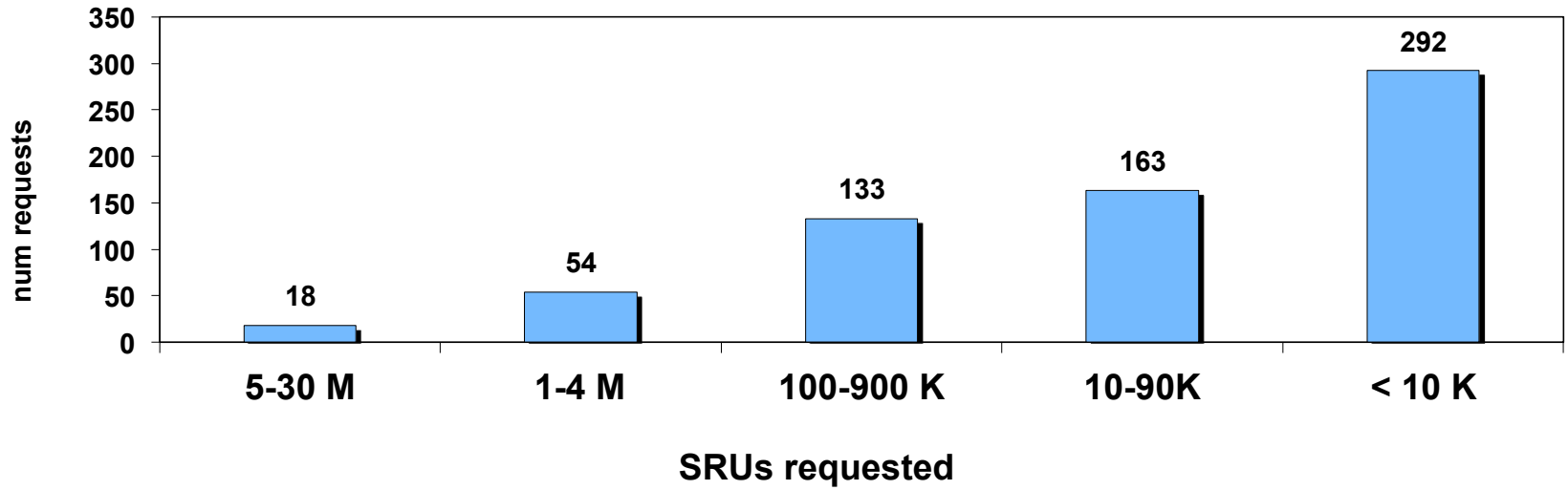
Percent of SRUs Requested



Percent of Projects



2015 HPSS Request Distribution
(660 requests total)



- **Largest HPSS requests:**
  - 30 M SRUs (BER Biology)
  - 25 M SRUs (BER Climate)
  - 20 M SRUs (ASCR Math)
  - 20 M SRUs (NO Theory)
  - 15.55M M SRUs (BER Climate)
  - 15 M (BER Climate)
- **4 projects requested 26% of the total amount requested**
- **12 projects requested 51% of total**
- **38 projects requested 75% of total**

Queue Committee Items

- **Suggestion to increase premium queue limits on number of nodes and wall time**
  - Better turn around for debugging codes and workflows
  - Currently 1024 nodes and 12 hours
  - Straw man proposal: 4096 nodes and 24 hours
  - Need to monitor usage in premium? Currently 4.5% of usage in premium, 5% in debug on Edison.
- **Call for additional topics**

NUGEX Elections

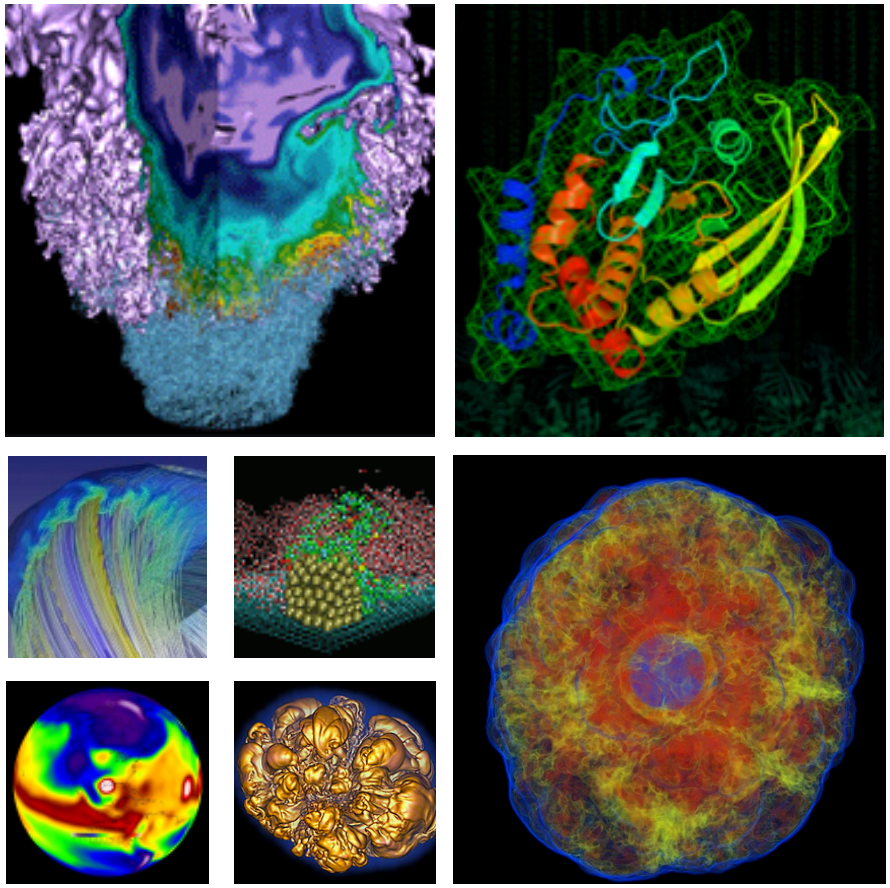
- **Eight seats on NUGEX are up for election in December 2014**
  - Fusion – 2: Ethier, Vay
  - High Energy Physics – 3: Borrill, Gottlieb, Tsung
  - Nuclear Physics – 2: Kasen, Savage
  - At large – 1: Newman

- **Contact Frank Tsung (tsung@physics.ucla.edu) if you are interested in running for one of these spots.**

NUG 2015

- **Time to start planning for NUG 2015**
- **Site: Berkeley/Oakland**
- **Call for volunteers for planning committee**
  - Pick a theme, line up speakers, topics, set agenda
  - Set dates
  - Training topics
- **Possible dates**
  - Feb 3-5, 2015
  - Feb 10-12, 2015
  - ?

Steps to improve the performance of a hydrodynamics application on the Intel MIC architecture



Chris Daley

NERSC Advanced Technologies Group

2 Oct 2014

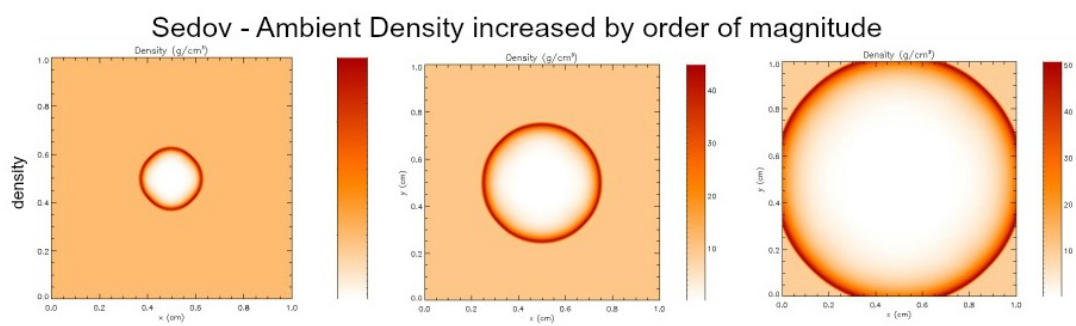
NERSC 40 YEARS at the FOREFRONT 1974-2014

U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

FLASH case study

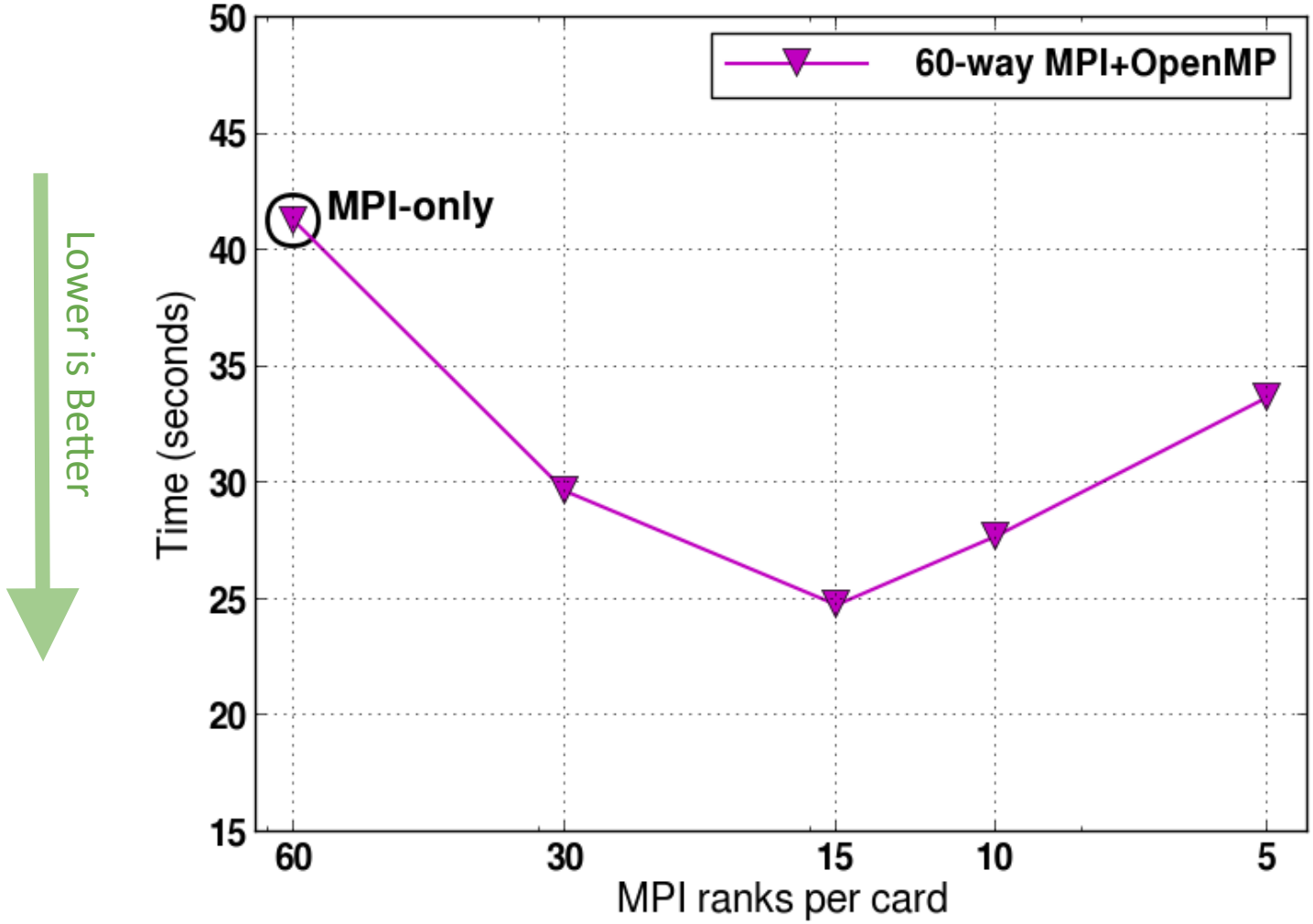
- **FLASH is astrophysics code with explicit solvers for hydrodynamics and magneto-hydrodynamics**
- **Parallelized using**
  - MPI domain decomposition AND
  - OpenMP multithreading over local domains or over cells in each local domain
- **Target application is a 3D Sedov explosion problem**
  - A spherical blast wave is evolved over multiple time steps
  - Use configuration with a uniform resolution grid and use 100^3 global cells
- **The hydrodynamics solvers perform large stencil computations**



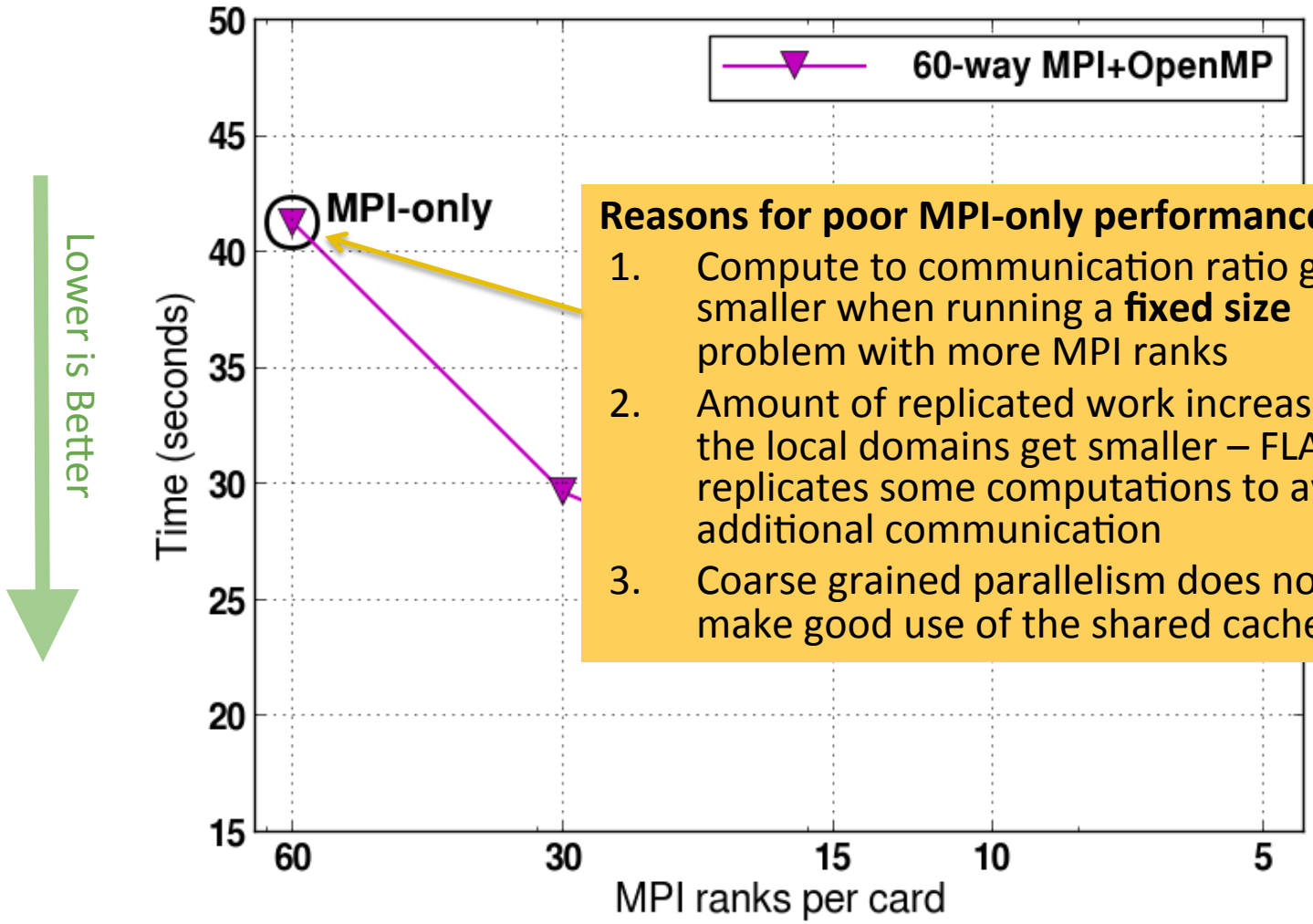
The Xeon-Phi (MIC) architecture: NERSC's Babbage Testbed

- **45 Sandy-bridge nodes with Xeon-Phi co-processor**
- **Each Xeon-Phi co-processor has**
  - 60 cores
  - 4 HW threads per core
  - 8 GB of memory
- **Multiple ways to program with co-processor**
  - As an accelerator
  - As a self-hosted processor (ignore Sandy-bridge)
  - Reverse accelerator
  - We chose to test as if the Xeon-Phi was a stand alone processor to mimic Knight's Landing architecture

Performance exploration on 1 MIC card



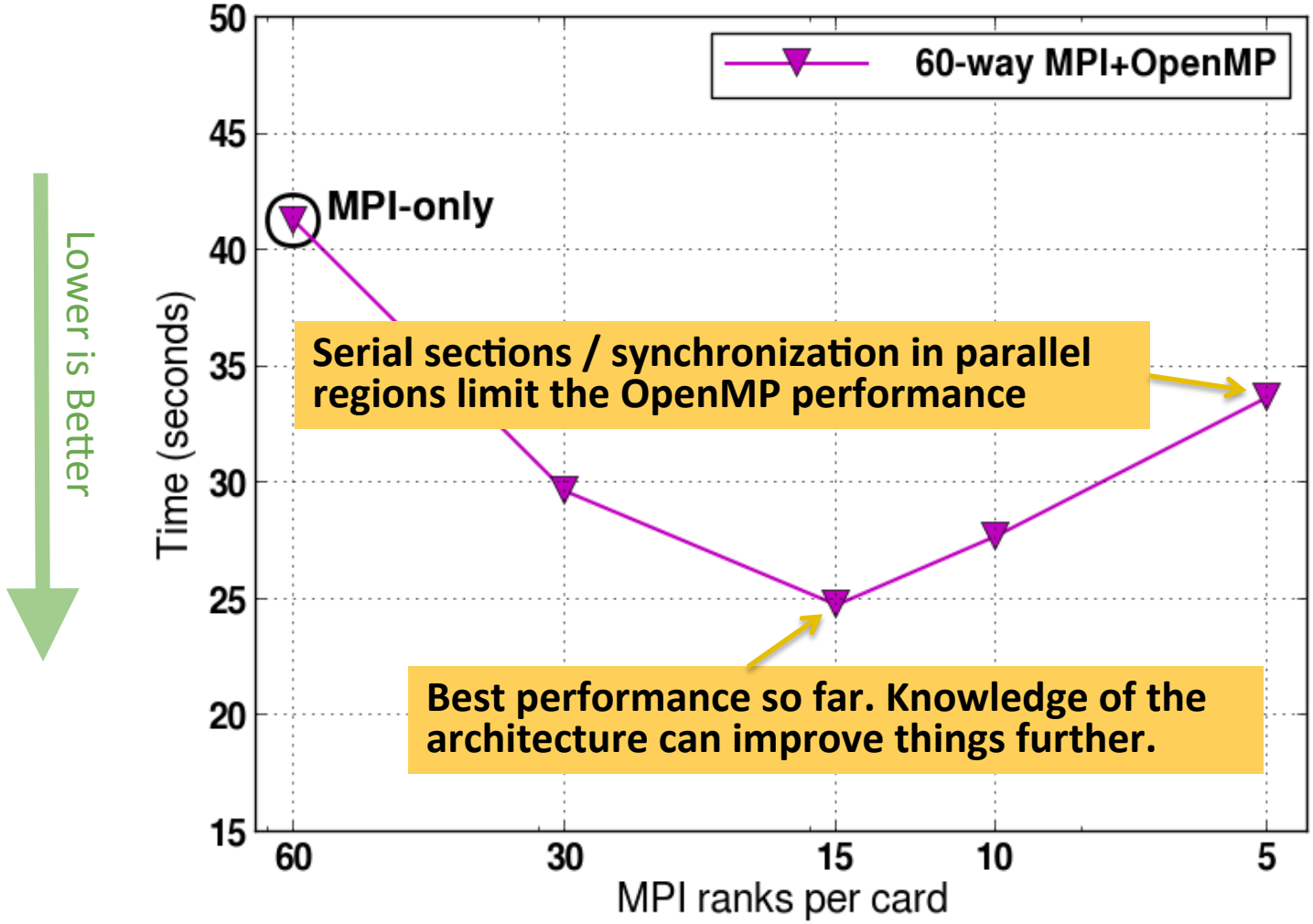
Performance exploration on 1 MIC card



Reasons for poor MPI-only performance

1. Compute to communication ratio gets smaller when running a **fixed size** problem with more MPI ranks
2. Amount of replicated work increases as the local domains get smaller – FLASH replicates some computations to avoid additional communication
3. Coarse grained parallelism does not make good use of the shared caches

Performance exploration on 1 MIC card



Getting more performance out of a KNC core

- **Harder to keep the cores busy compared to more traditional multicore architectures, such as NERSC's Edison machine**
  1. More cores
     - Program must expose more parallelism
  2. In-order instruction execution
     - The KNC processor cannot reorder instructions to avoid stalls
  3. Less cache per core
     - Babbage: **L1 = 32 KiB, L2 = 512 KiB**
     - Edison: **L1 = 64 KiB, L2 = 256 KiB, L3 = 2.5 MiB**
  4. Higher main memory latency
     - Babbage: **288 ns ≈ 302 clock cycles** [Fang, Jianbin, et al. "An Empirical Study of Intel Xeon Phi." arXiv preprint arXiv:1310.5842 (2013)]
     - Edison: **82 ns ≈ 200 clock cycles** [http://www.nersc.gov/users/computational-systems/edison/configuration/compute-nodes/]

Getting more performance out of a KNC core

Very high memory bandwidth: 164 GB/s read; 76 GB/s write; 120 GB/s write with streaming stores

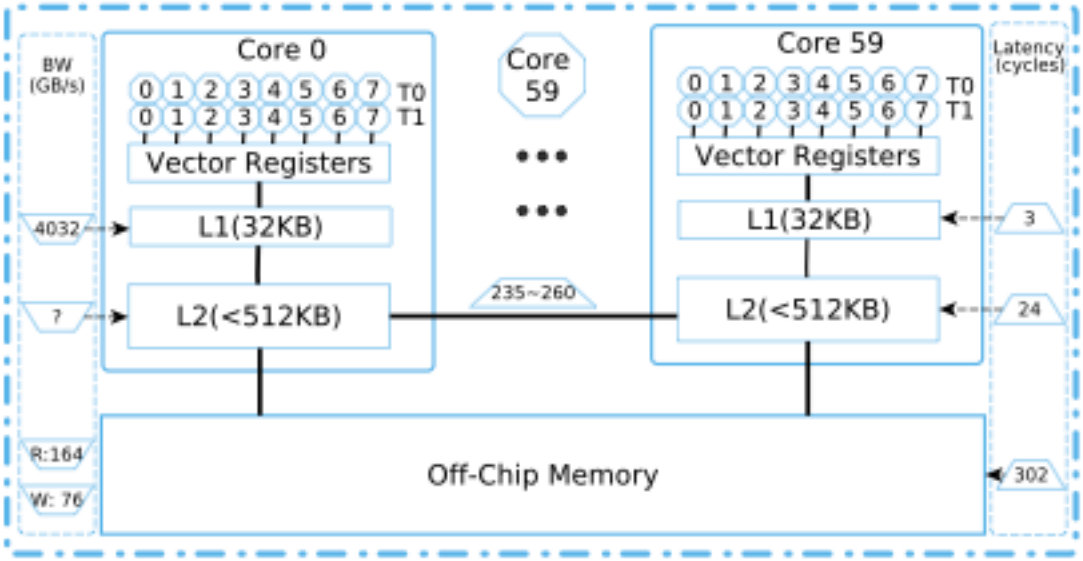
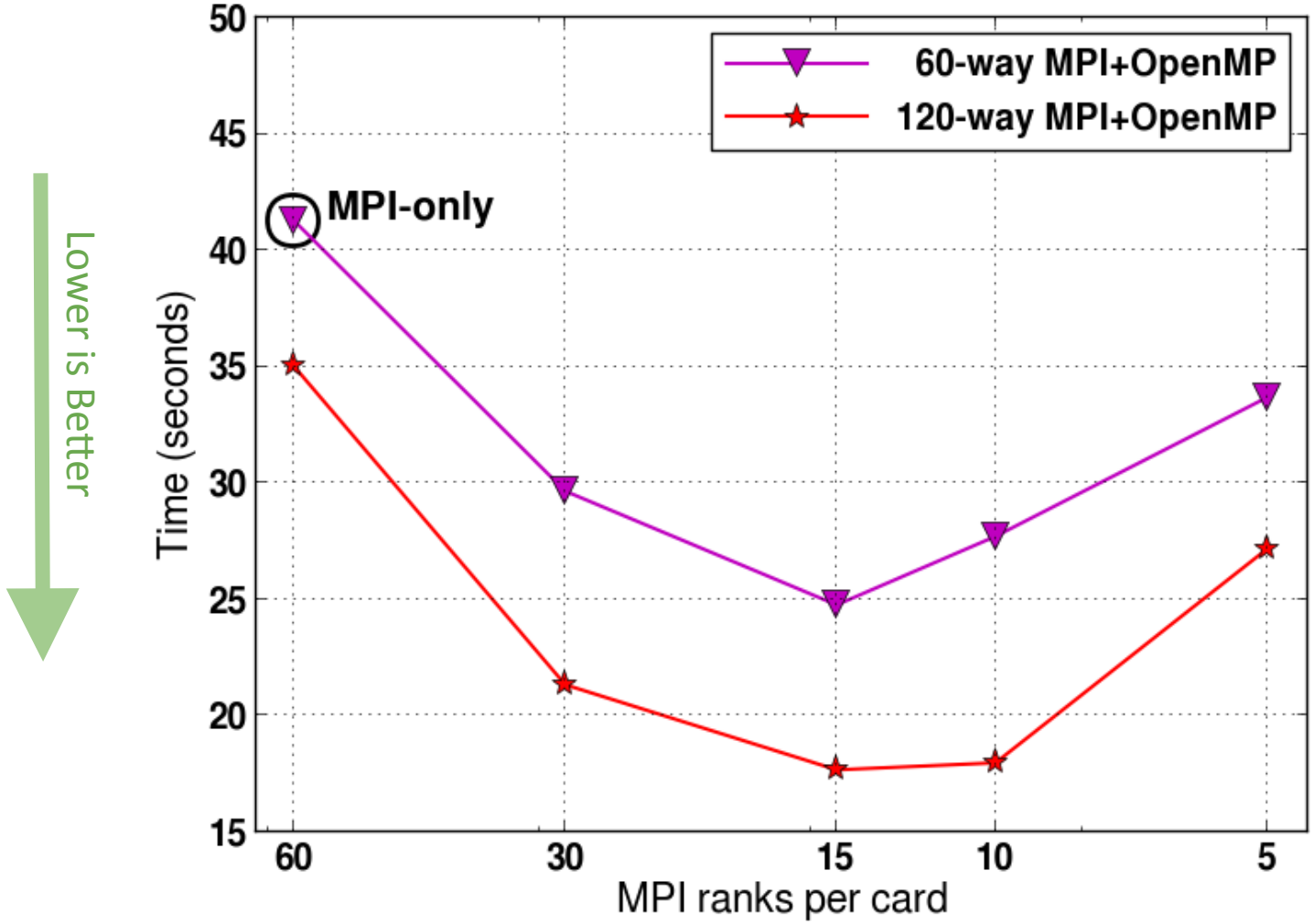


Figure taken from Fang, Jianbin, et al. "An Empirical Study of Intel Xeon Phi." arXiv preprint arXiv:1310.5842 (2013)

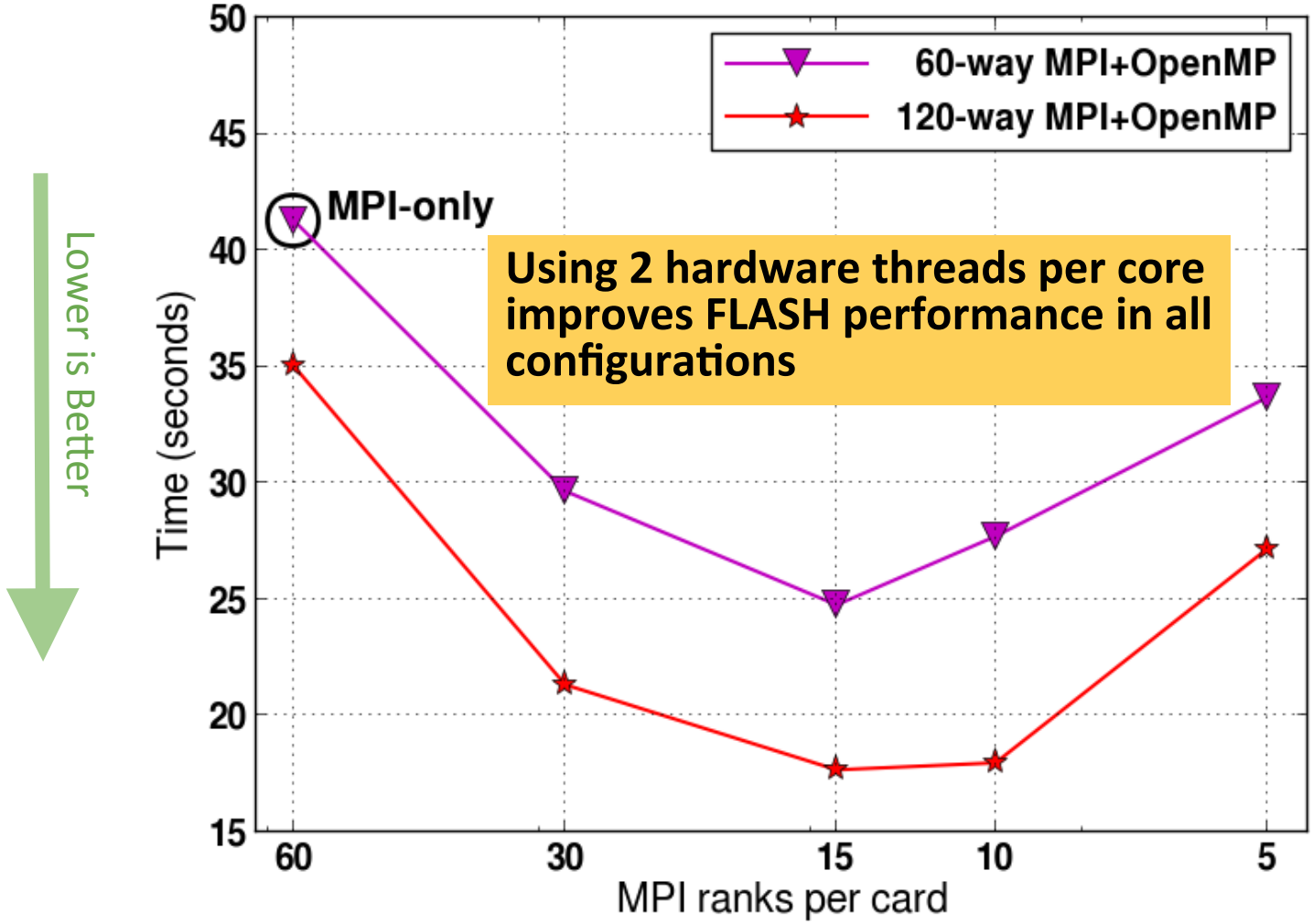
Poor GDDR5 memory latency: 302 clock cycles ≈ 288 ns. Very significant when a program has non-regular memory accesses

- **Using multiple hardware threads per core allows us to hide processor stalls / memory latency**
  - It also allows us to achieve the peak performance of a core: a single thread can only issue instructions every other clock cycle.

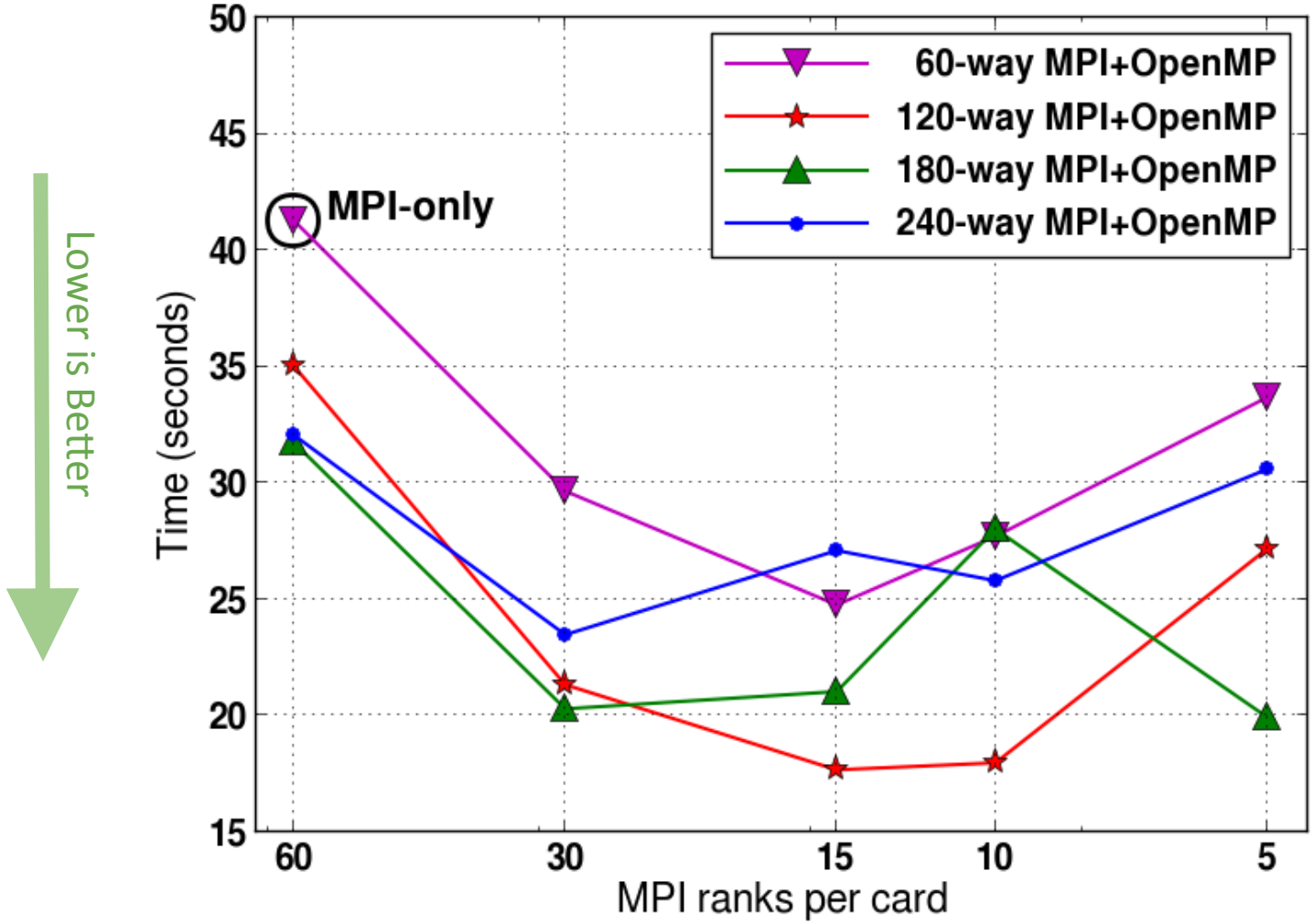
Performance exploration on 1 MIC card



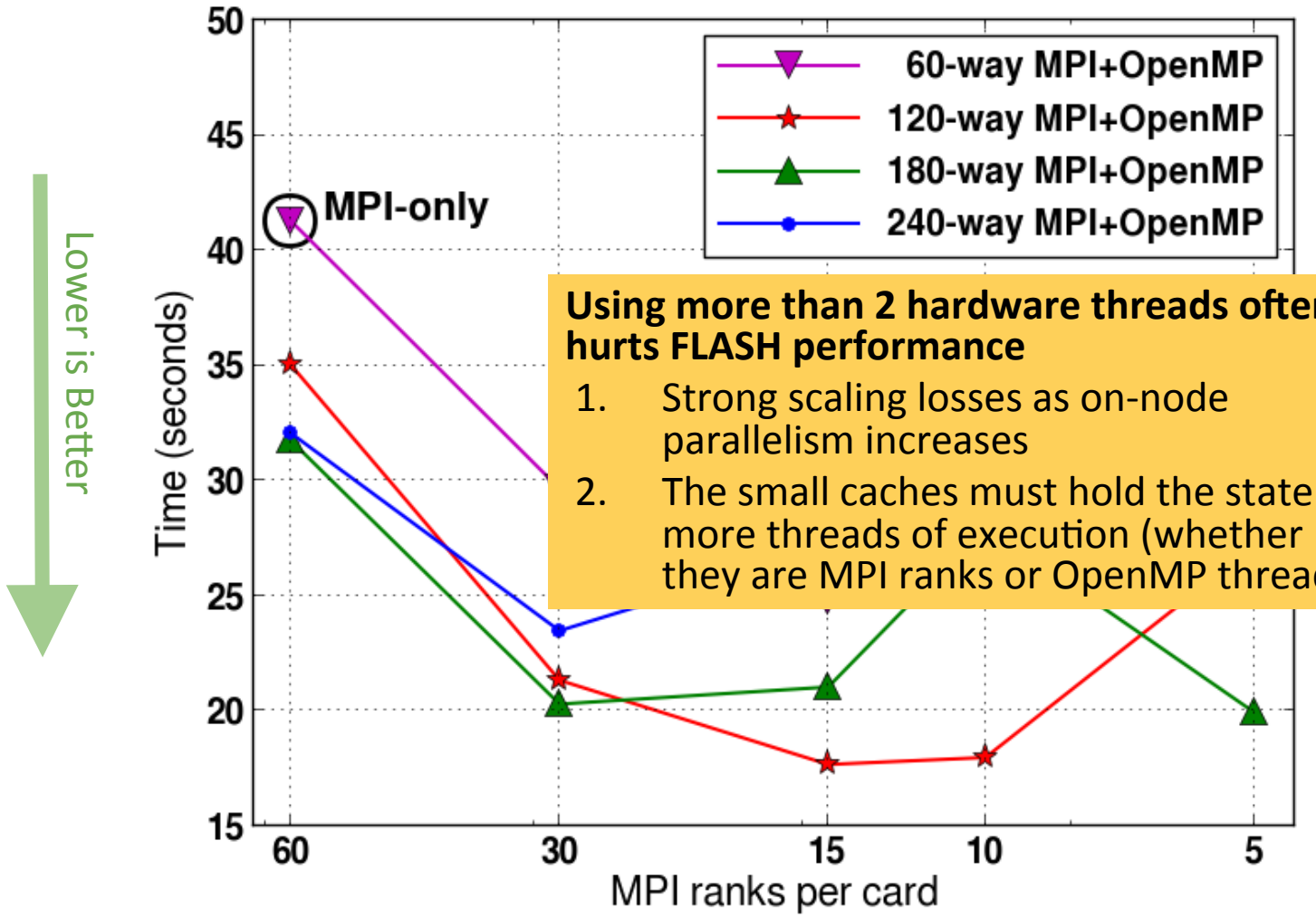
Performance exploration on 1 MIC card



Performance exploration on 1 MIC card



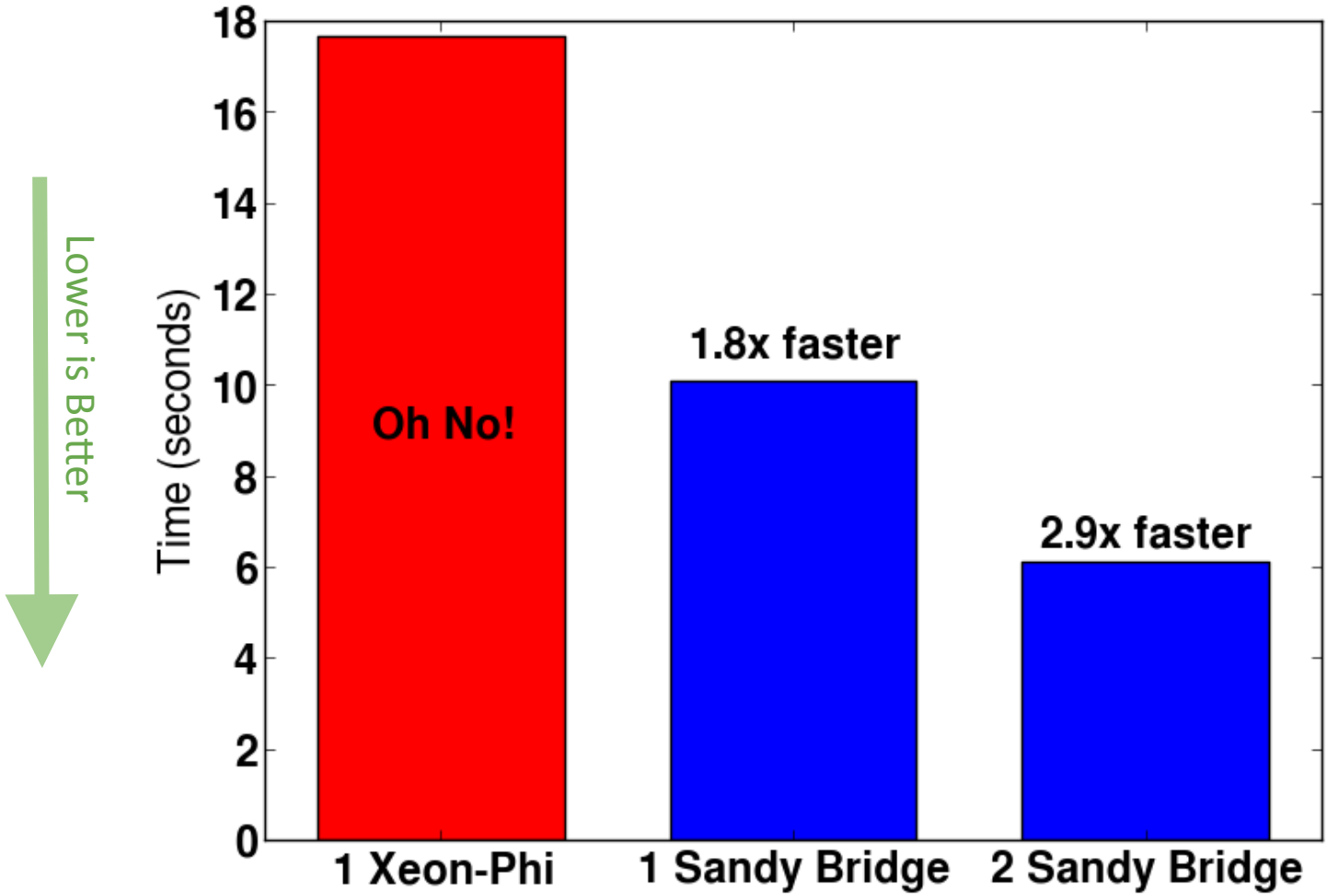
Performance exploration on 1 MIC card



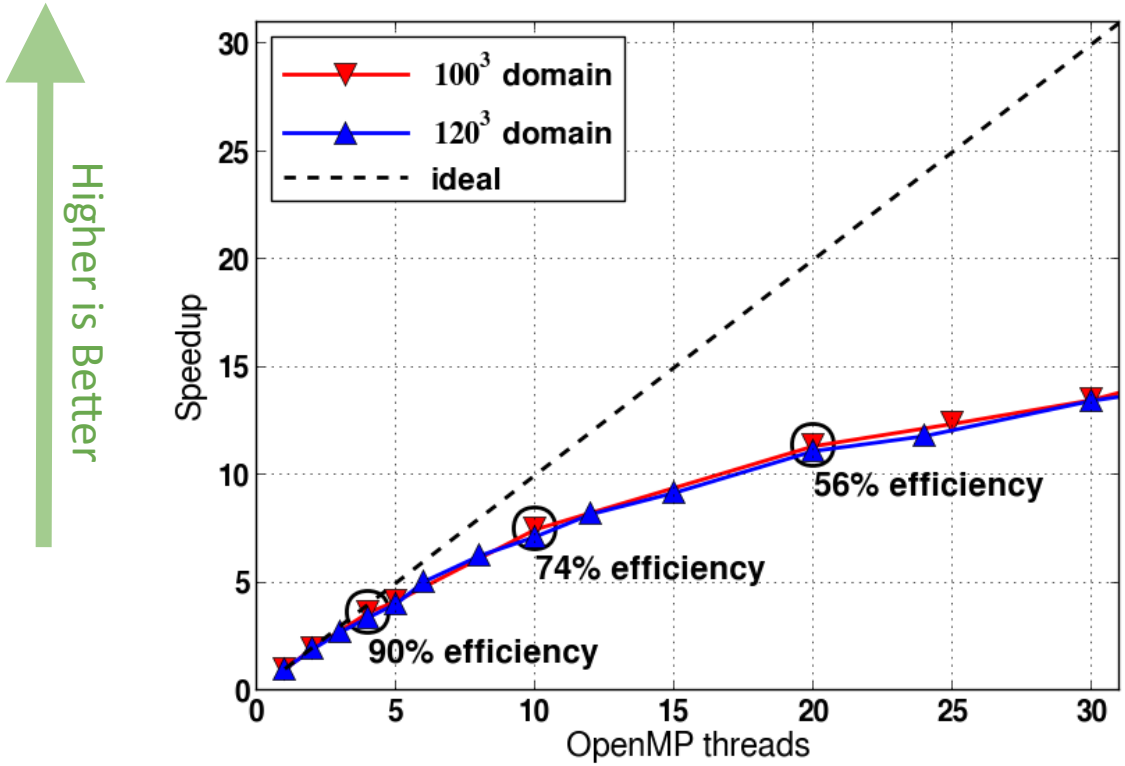
Using more than 2 hardware threads often hurts FLASH performance

1. Strong scaling losses as on-node parallelism increases
2. The small caches must hold the state for more threads of execution (whether they are MPI ranks or OpenMP threads)

Initial best KNC performance vs host



MIC performance study 1: thread speedup



- 1 MPI rank per MIC card and various numbers of OpenMP threads
- Each OpenMP thread is placed on a separate core
- 10x thread count ideally gives a 10x speedup

- Speedup is not ideal
  - But it is not the main cause of the poor MIC performance
  - ~70% efficiency @ 12 threads (as would be used with 10 MPI ranks per card)

Vectorization is another form of on-node parallelism

```
do i = 1, n
    a(i) = b(i) + c(i)
enddo
```

$$\begin{pmatrix} a_1 \\ \dots \\ a_n \end{pmatrix} = \begin{pmatrix} b_1 \\ \dots \\ b_n \end{pmatrix} + \begin{pmatrix} c_1 \\ \dots \\ c_n \end{pmatrix}$$

Intel Xeon Sandy-Bridge/Ivy-Bridge:	4 Double Precision Ops Concurrently
Intel Xeon Phi:	8 Double Precision Ops Concurrently
NVIDIA Kepler GPUs:	32 SIMT threads

Data parallelism - vectorization

- **Vectorization is another form of on-node parallelism**

```
DO i= 1, 100
  a(i) = b(i) + c(i)
ENDDO
```

```
DO i= 1, 100, 4
  a(i) = b(i) + c(i)
  a(i+1) = b(i+1) + c(i+1)
  a(i+2) = b(i+2) + c(i+2)
  a(i+3) = b(i+3) + c(i+3)
ENDDO
```

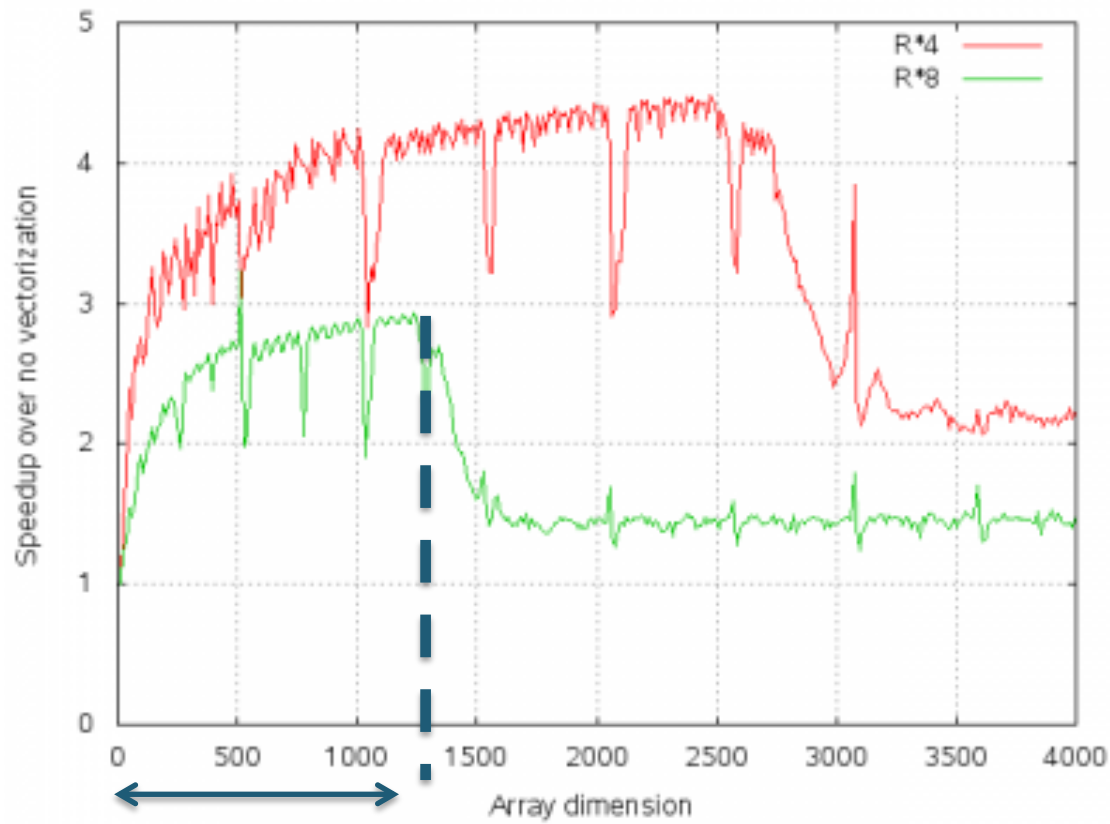
- A single instruction will launch many operations on different data;
- Most commonly, multiple iterations of a loop can be done "concurrently." (Simpler control logic, too.)
- Compilers with optimizations setting turned on will attempt to vectorize your code.

A vectorization experiment on Edison

- **The vector unit is only useful if we can get data to it at a fast enough rate**

The figure shows the vectorization gain when running the following kernel on Edison as a function of array size n

```
do it = 1, itmax
  do i = 1, n
    c(i) = a(i) + b(i)
  end do
end do
```



Up to **3x gain** when all accesses hit in L1

… falling to **1.4x gain** when all accesses hit in L2

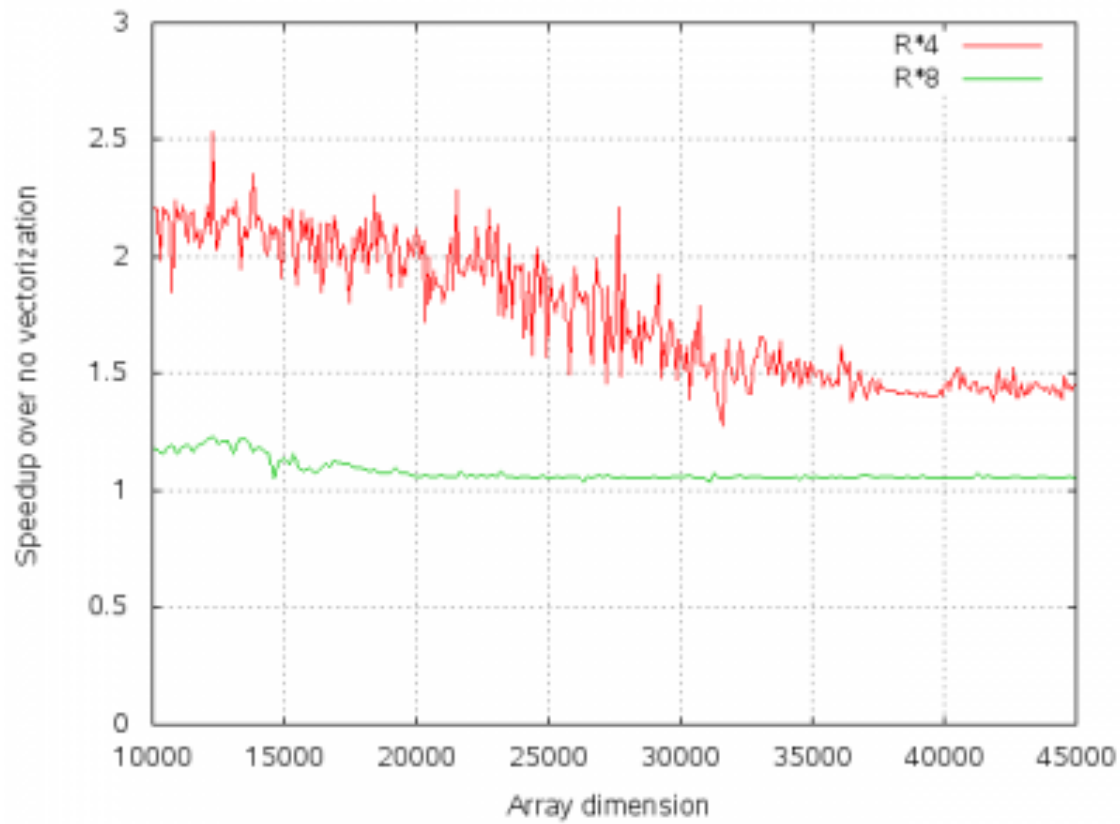
https://www.nersc.gov/users/computational-systems/edison/programming/vectorization/

A vectorization experiment on Edison

- **The vector unit is only useful if we can get data to it at a fast enough rate**

The figure shows the vectorization gain when running the following kernel on Edison as a function of array size n

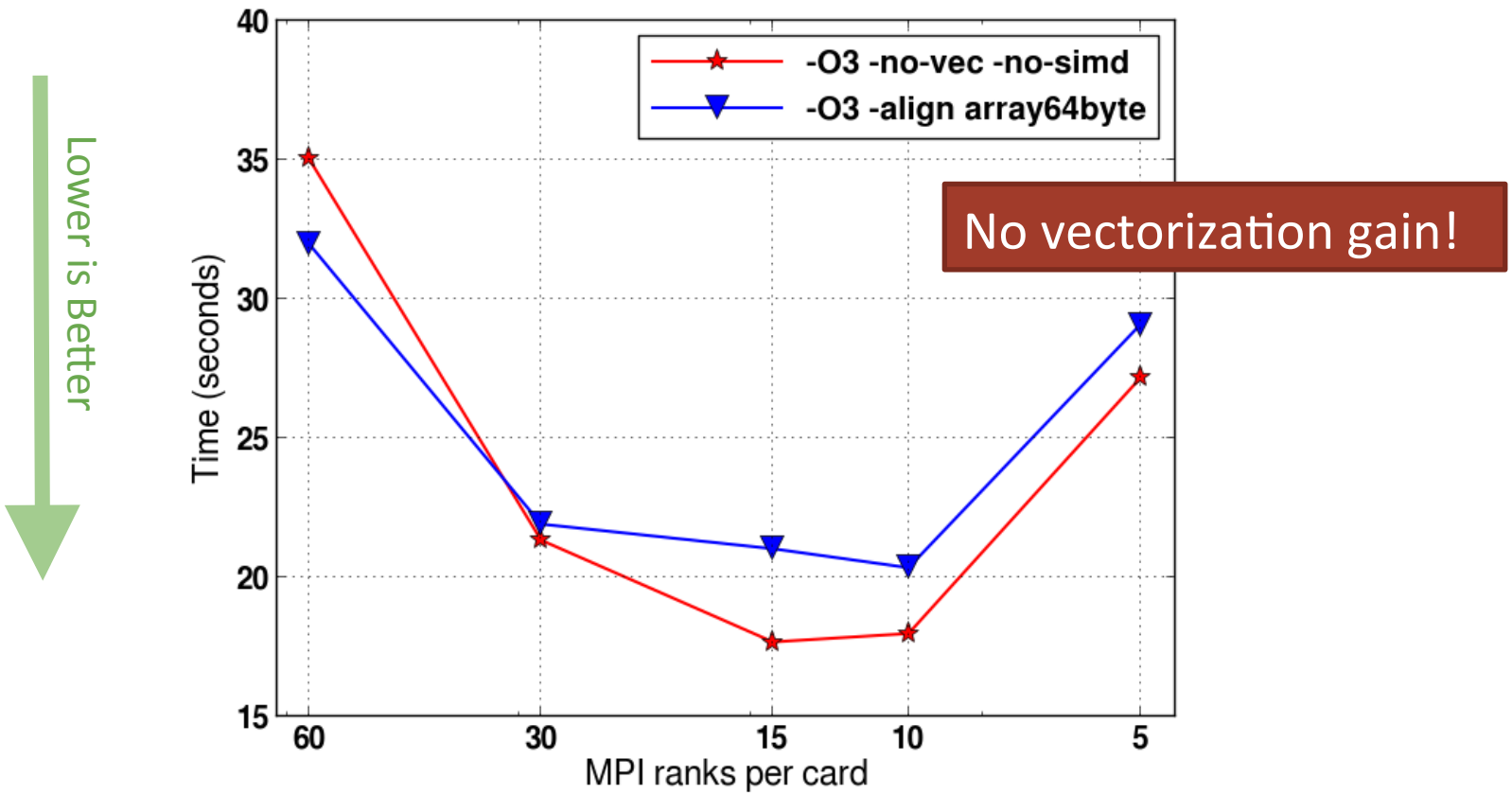
```
do it = 1, itmax
  do i = 1, n
    c(i) = a(i) + b(i)
  end do
end do
```



… falling to **1.06x gain** when all accesses hit in L3

https://www.nersc.gov/users/computational-systems/edison/programming/vectorization/

MIC performance study 2: vectorization



- We find that most time is spent in kernel subroutines which update fluid state 1 grid point at a time

Changes to enable vectorization

- **The kernel subroutines are passed a single grid point along with its "halo" of neighboring grid points**
  - The data for 1 grid point is laid out as a structure of fluid fields, e.g. density, pressure, …, temperature next to each other: A(HY_DENS:HY_TEMP)
  - Minimal opportunity for vectorization because most expressions operate on a single fluid field at a time
- **We need to restructure to take advantage of vectorization**
  - Both code layout and data structure changes

0 → 1 → 2

```
do n = 1, 5
  A(HY_DENS) =
end do
```

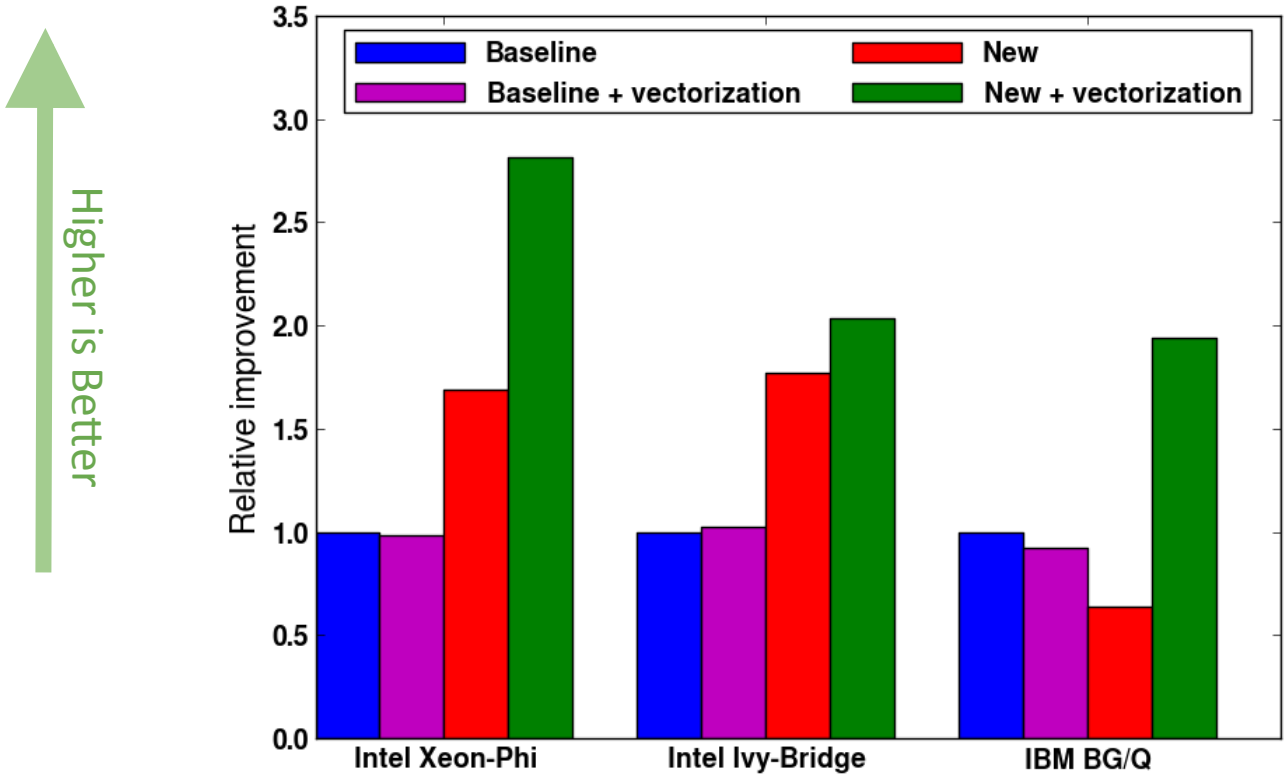
```
do i = 1, GRID_PTS
  do n = 1, 5
    A(HY_DENS,i) =
  end do
end do
```

```
do n = 1, 5
  do i = 1, GRID_PTS
    ! A(i,HY_DENS) =
    dens(i) =
  end do
end do
```

- **Move loop over grid points into the kernel; change loop order; create a separate array for each fluid field to make alignment simpler**

Impact of vectorization

- **We tested these changes on part of a data reconstruction kernel**

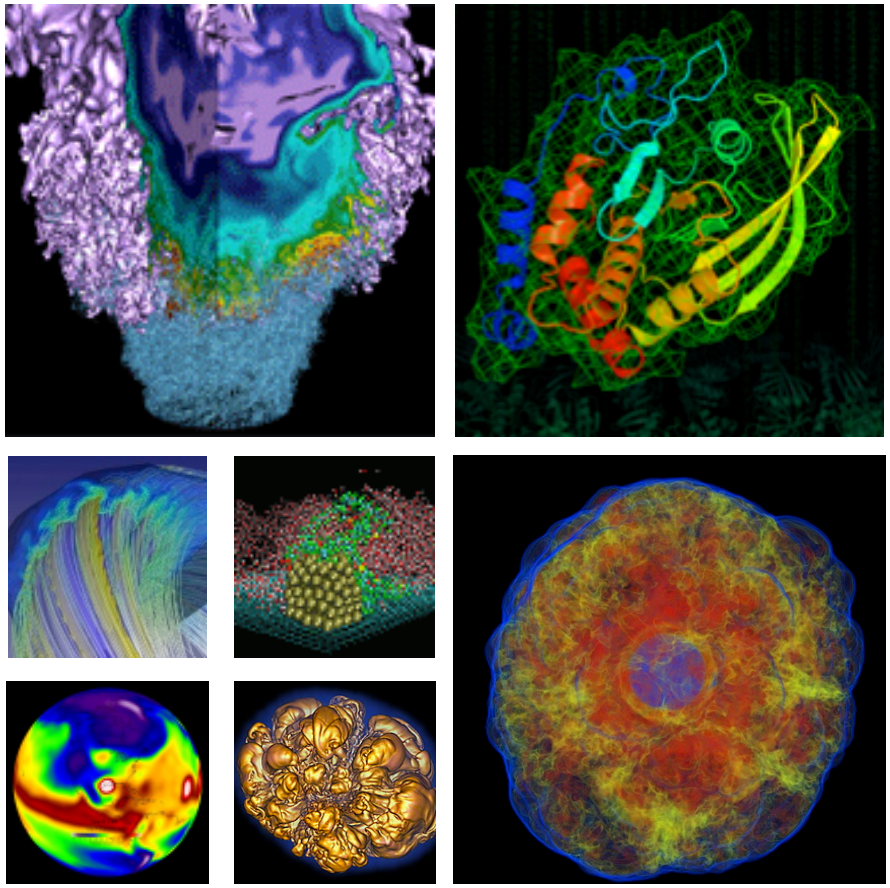


- **The new code compiled with vectorization options gives the best performance on 3 different platforms**

Case study summary

- **FLASH on MIC**
  - MPI+OpenMP parallel efficiency – OK
  - Vectorization – **zero / negative gain** …must restructure!
    - Compiler auto-vectorization / vectorization directives do not help the current code
- **Changes needed to enable vectorization**
  - Make the kernel subroutines operate on multiple grid points at a time
  - Change the data layout by using a separate array for each fluid field
    - Effectively a change from array of structures (AofS) to structure of arrays (SofA)
- **Tested these proof-of-concept changes on a reduced hydro kernel**
  - Demonstrated improved performance on Ivy-Bridge, BG/Q and Xeon-Phi platforms

NERSC's next supercomputer: Cori



U.S. DEPARTMENT OF ENERGY | Office of Science

BERKELEY LAB

Cori Configuration

- **64 Cabinets of Cray XC System**
  - Over 9,300 'Knights Landing' compute nodes
  - Over 1,900 'Haswell' compute nodes
    - Data partition
  - Cray Aries Interconnect
- **Lustre File system**
  - 28 PB capacity, 432 GB/sec peak performance
- **NVRAM "Burst Buffer" for I/O acceleration**
- **Significant Intel and Cray application transition support**
- **Delivery in mid-2016; installation in new LBNL CRT**

Intel "Knights Landing" Processor

- Next generation Xeon-Phi, >3TF peak
- Single socket processor - Self-hosted, not a co-processor, not an accelerator
- Greater than 60 cores per processor with support for four hardware threads each; more cores than current generation Intel Xeon Phi™
- Intel® "Silvermont" architecture enhanced for high performance computing
- 512b vector units (32 flops/clock – AVX 512)
- 3X single-thread performance over current generation Xeon-Phi co-processor
- High bandwidth on-package memory, up to 16GB capacity with bandwidth projected to be 5X that of DDR4 DRAM memory
- Higher performance per watt

Thank you.

Application readiness lessons learned

- Improving a code for advanced architectures can improve performance on traditional architectures.
- Inclusion of OpenMP may be needed just to get the code to run (or to fit within memory)
- Some codes may need significant rewrite or refactoring; others gain significantly just adding OpenMP and vectorization
- Profiling/debugging tools and optimized libraries will be essential
- Vectorization important for performance