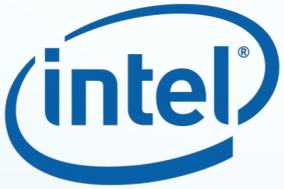
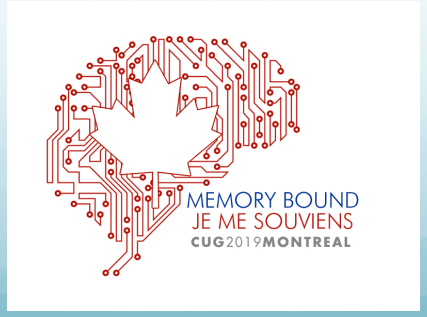
Sustained Petascale Direct Numerical Simulation on Cray XC40 Systems

Bilel Hadri, Matteo Parsani, Maxime Hutchison
Alexander Heinecke, Lisandro Dalcin, David Keyes





Nek5000

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Nek5000, an incompressible Navier-Stokes solver,
  - fast and scalable high-order solver for computational fluid dynamics
  - Gordon Bell award in 1999
  - https://nek5000.mcs.anl.gov/

- Nek5000 selected by the Center for Efficient Exascale Discretizations (CEED),
  - a co-design center within the U.S. Department of Energy (DOE) Exascale Computing Project (ECP),
  - help applications leverage future exascale systems by exploiting efficiently and effectively the hardware
  - delivering a significant performance gain over conventional low-order methods.

NexBoX

- **Goal:**
  - **Present an overview NekBox, an optimized incompressible Navier-Stokes solver branched from Nek5000 for today and future architecture**
- **Present performance study on three different Cray XC40**
  - **Shaheen(KAUST)**
  - **Cori(NERSC)**
  - **Trinity(LANL)**
- **MPI+X and asynchronous communication are usually required to scale on modern architectures.**
- **In contrast, NekBox is pure MPI with synchronous communication interfaces.**
- **We will show that this execution model leads to excellent performance results even with high order of scheme**

Why High Order schemes ?

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Using High Order schemes to get accuracy and try to achieve time-to-solution faster
- Produce a software infrastructure capable of supporting exascale applications then we should be aligned with the IESP roadmap

- Increase arithmetic intensity (ratio of flop/s to Byte/s)
- Increase SIMD-style concurrency within a node (strong scaling)
- Decrease synchronization, in frequency or global scope, or both
- Absorb resilience features into software, relieving hardware where possible

NekBox

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- NekBox solves the incompressible Navier-Stokes equations and advection-diffusion equations for scalars

$$\frac{\partial u}{\partial t} + u \cdot \nabla u = -\frac{1}{\rho}\nabla p + \nu \nabla^2 u + f \qquad \nabla \cdot u = 0 \qquad \frac{\partial \phi_i}{\partial t} + u \cdot \nabla \phi_i = \alpha_i \nabla^2 \phi_i + q_i$$

- Implements the spectral element method (SEM) & spends the run-time in
  - Sparse computations & communications
    - direct stiffness summations
    - coarse part of the pressure Poisson solve
  - Vector-vector or matrix-vector operations
    - inner products
    - rescaling with geometrical factors or diagonal matrices
  - Matrix-matrix operations (operations that strongly depend on SEM order of accuracy)

Our Strategies

1. Auto-generated assembly implementations for small matrix multiplies.
   - With very high order, spectral elements result in arithmetically intensive kernels near compute-bandwidth parity.
   - MKL does not provide reasonable performance for the small general matrix
   - To ensure optimal performance on a variety of instruction sets, we employ LIBXSMM
     - https://github.com/hfp/libxsmm

2. Aggressive inter-procedural loop merging.
   - To maximize cache utilization, loops over elements are aggressively coalesced.
   - When data is written but not re-used, non- temporal stores avoid read-for-ownership memory operations.
   - Nearest-neighbor communication interfaces are implemented element-wise, improving locality of buffer loading and providing opportunities for communication and computation overlap.

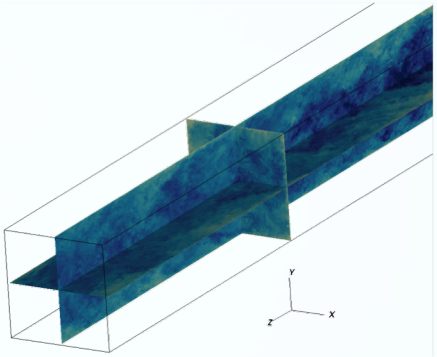
Our Strategies (2)

3. **Reduction in global coarse problem size.**
   - **reduction in problem size improves scalability and time to solution despite very high order problems being more arithmetically intensive and more accurate.**

4. **Selective evaluation in single precision.**
   - **Single-precision preconditioning is implemented to reduce costs when double-precision is unnecessary.**

5. **I/O with lossless compression. Already strong I/O performance is supplemented by lossless LZ4 compression.**
   - **The compression is light-weight enough to hide behind the disk I/O bottleneck.**

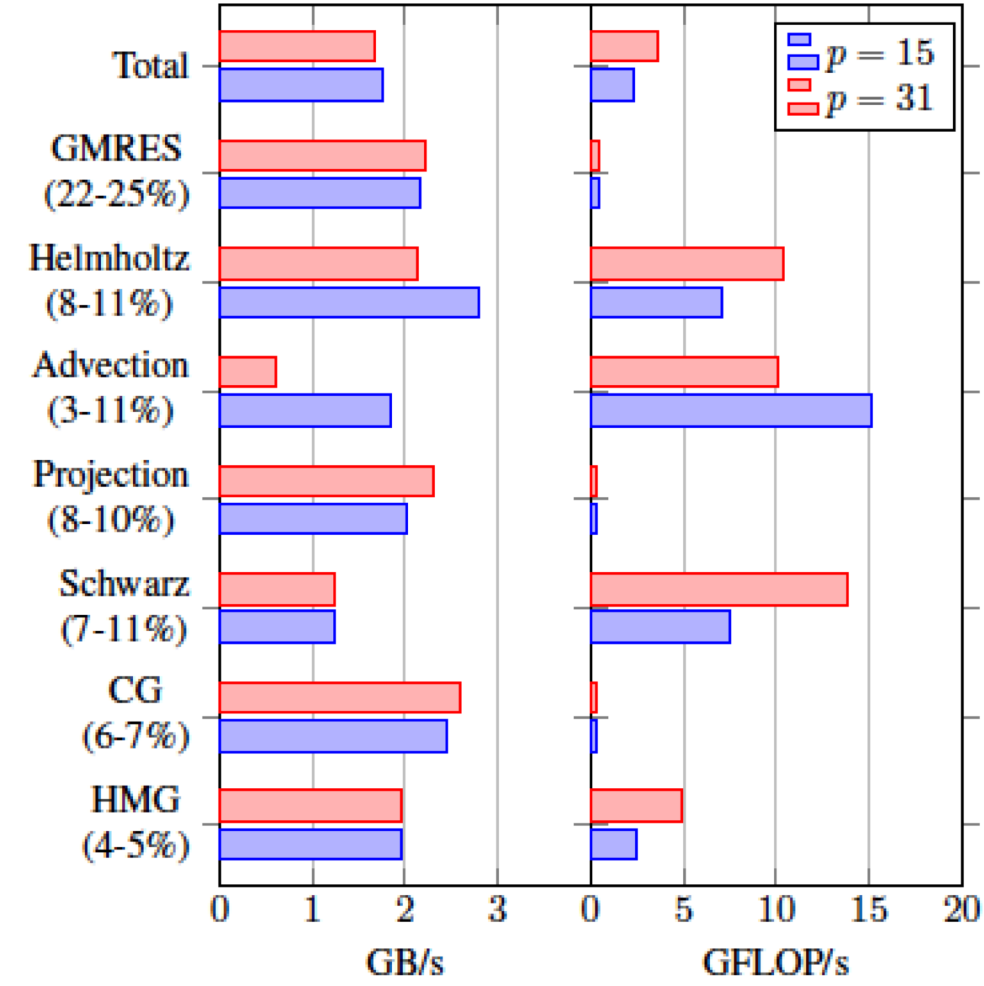
Evaluation and Performance



- Test case
  - DNS of secondary flows in a duct at bulk Re=100,000
  - Performance study performed using 16th- (p=15) and 32nd-order (p=31) SEM algorithm

- Platforms
  - Shaheen2, Cray XC40 , 36-cabinet Cray composed of 6,174 dual socket compute nodes based on 16 core Intel Haswell
    - 57% of the maximum global bandwidth between the 18 groups of two cabinet
    - CLE 5UP02, then CLE6 UP04/5
  - Trinity, Cray XC40, 9,436 of similar node resulting into a total of 301,056 cores and over 1.2 PB of aggregate memory
    - CLE6 UP01
  - Cori , Cray XC40 with 9304 Intel KNL nodes
    - CLE6 UP01 CDT16.10

NekBox profiling

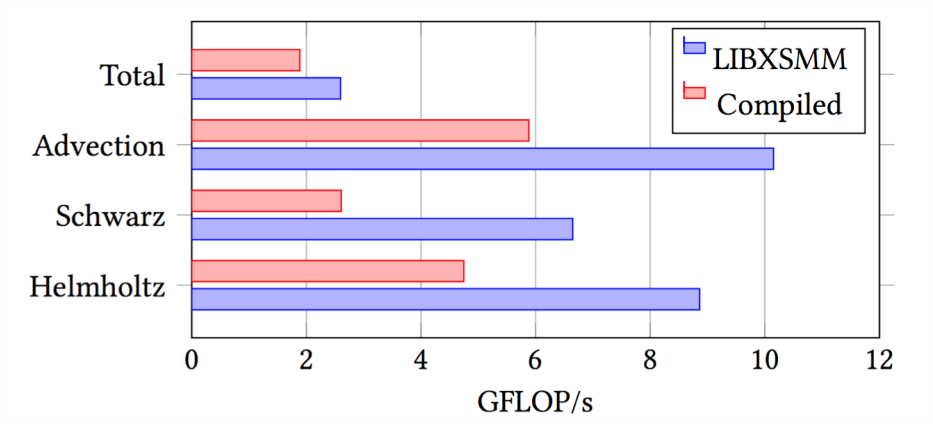


Key code sections for fully developed turbulent flow at p = 15 and p = 31 on 8192 nodes of Trinity XC40.

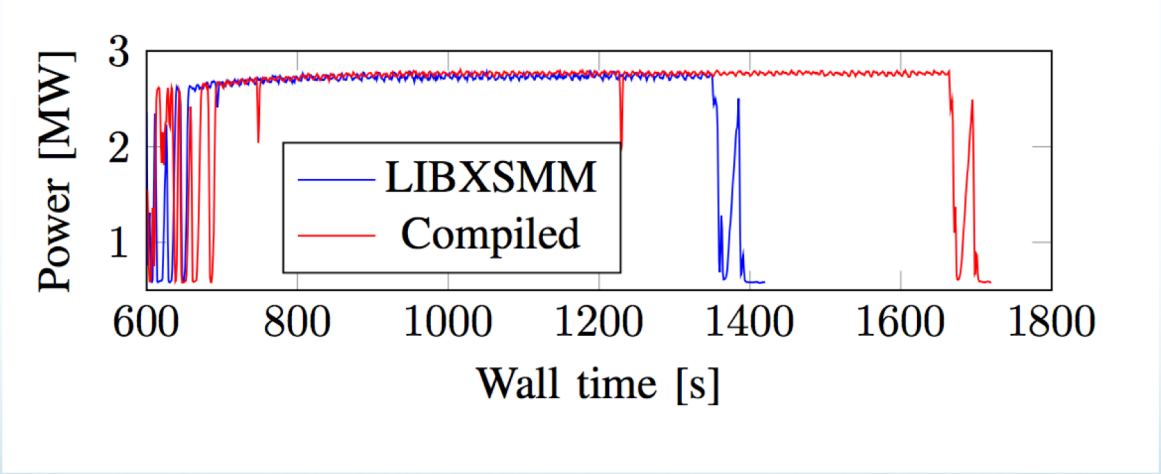
LIBXMM Performance on Shaheen

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

~69 billion points using 6144 nodes on Shaheen



2x on the total performance is reduced bv communications and vector workloads

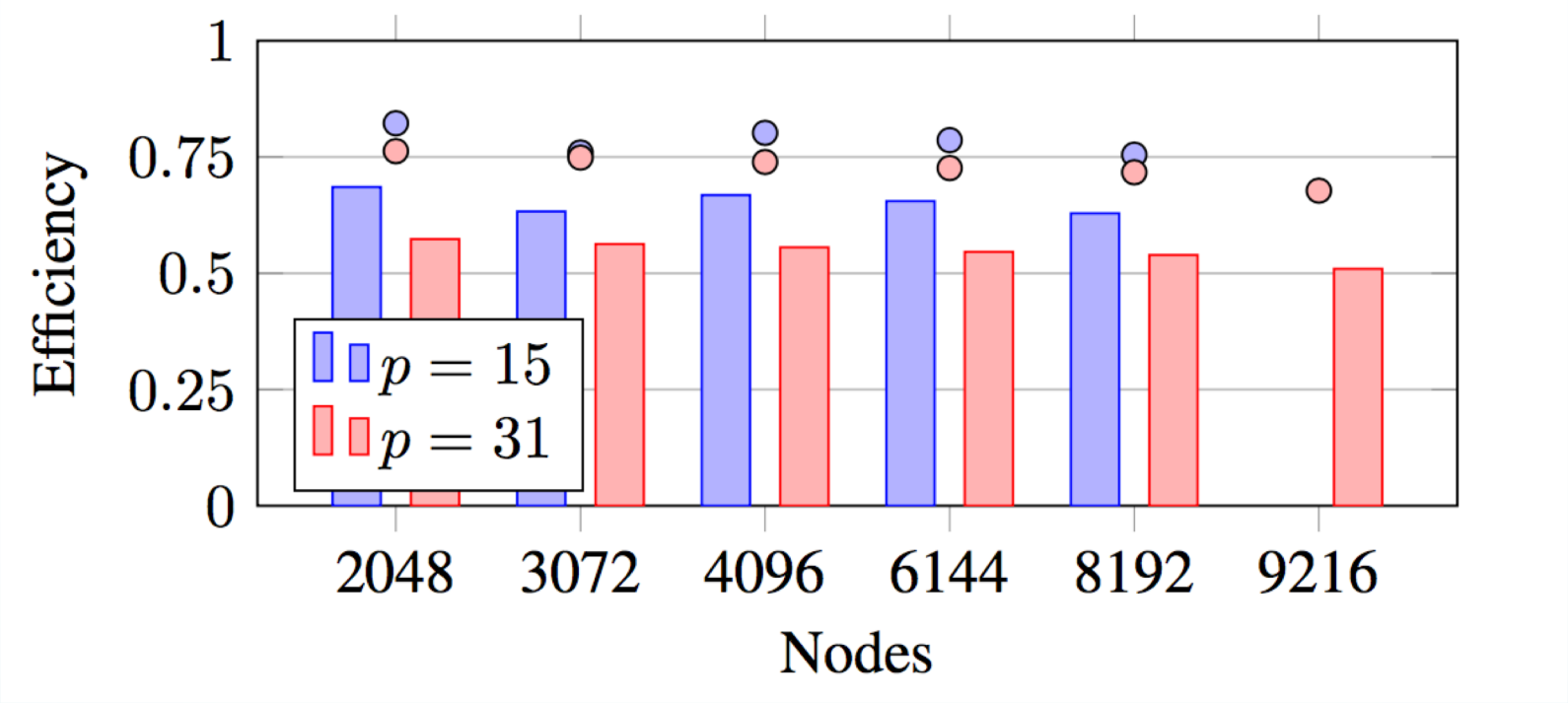


761.9 kWh vs. 532.59 kWh ~ 30% saving!

Weak Scaling on Trinity

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

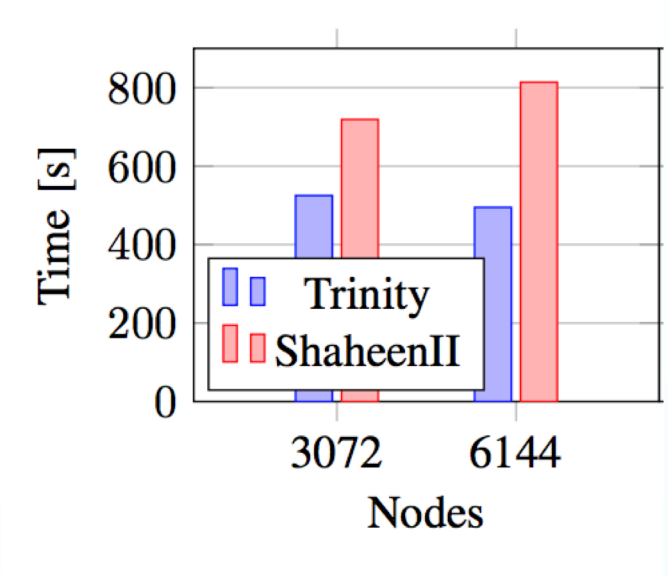
Overall efficiency on Trinity XC40 262,144 grid points per core with respect to memory bandwidth for weal scaling. Points show parallel efficiency based on 1 node



On 294,912 cores with *p=31* we achieve **516 TB/s** (~50% of peak memory bandwidth)

Sustained 1.05 PFLOP/s over the entire **production run** simulation

Trinity vs Shaheen (v1)



- **Counter to our expectation, ShaheenII is 40% to 65% slower than Trinity.**
  - **22% of Aries theoretical bandwidth can be achieved on Trinity while it's 57% on ShaheenII**

- **Degraded performance on ShaheenII is similar to the synthetic results in the regression testing**

Regression Testing

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

Shaheen Regression testing

Data per core	Unit	Min	Max	Average
HPL	[GFLOP/s]	29.2	31.9	30.6
STREAM TRIAD	[GB/s]	3.3	3.8	3.7
nektester, $s = 1$	[GFLOP/s]	31.7	36.2	34.9
nektester, $s = 768$	[GB/s]	2.6	3	2.8
MPI allreduce (1 double)	[μs]	130.6	130.7	130.6
Gather Scatter	[μs]	571.4	1686.5	1431.5

Trinity Regression testing

Data per core	Unit	Min	Max	Average
nektester, $s = 1$	[GFLOP/s]	31.8	35.5	33.7
nektester, $s = 768$	[GB/s]	2.6	2.8	2.7
MPI allreduce (1 double)	[μs]	83.5	83.5	83.5
Gather/Scatter	[μs]	354.7	914.9	651.1

- **The relative percentage difference between the best and weak nodes can reaches up to 15% on Shaheenand around 11% on Trinity .**
- **MPI allreduce Shaheen is around 60% slower than Trinity.**
- **Shaheen using CLE 5.2 Up 04 OS, while all the performance runs on Trinity were achieved with an under-populated network and using the CLE 6.0 Up 01 with MPICH 7.3.1.**
- **Updated with mpich7.3.1 didn’t improve results**

SRT: Shaheen Regression Testing

- Loading similar programming environment and linking with same cray-mpich version did not improve significantly the performance.
- Upgrading ShaheenII to CLE 6.0 was scheduled only in early 2018.
- The HSN on Shaheen is configured with 8 optical network connections between every pair of cabinets while on Trinity, the HSN is configured with 2 optical network connection.
- we suspected some issue in the network and in order to isolate the link not performing as expected,
  - Employ the test_links tool developed at Cray that was specifically redesigned for Cray XC systems.
  - This tool is now part of the regular regression testing protocol after every maintenance or unscheduled downtime
  - *Regression Testing on Shaheen Cray XC40: Implementation and Lessons Learned: Bilel Hadri, Samuel Kortas, Robert Fiedler, George S. Markomanolis, CUG 2017*

Test links

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Test_links was originally developed for the Gemini network on Cray XE/XK systems for Blue Water.
- Cray recently redesigned for Cray XC systems for Aries interconnect for KAUST needs to validate the HSN health.
- Test_links evaluates the bandwidth of all interconnect links in any allocation of nodes and identifies links with lower (by a specified amount) than the best or the average bandwidth for links of the same type.
- Bandwidths for each link are also recorded in tables for comparison between sets of results obtained at different times, so that one may determine whether and how individual link performance has changed over time.

Test links (2)

جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

- Four different types of links are evaluated, including:
  - the PCIe links from the four compute nodes on a blade to the single Aries router on that blade,
  - the copper links between blades in the same chassis,
  - the copper links between blades in different chassis of the same group,
  - the optical links between groups.
- Test completes in about 7min using 6174 nodes.

	Description	Expected Performance
Dimension 1	Optical links between groups	60 GB/s
Dimension 2	Copper links between different chassis	8.5 GB/s
Dimension 3	Backplane within a chassis	3.5 GB/s
Dimension 4	PCIe connections from nodes to aries router	5 GB/s

Test links (3)

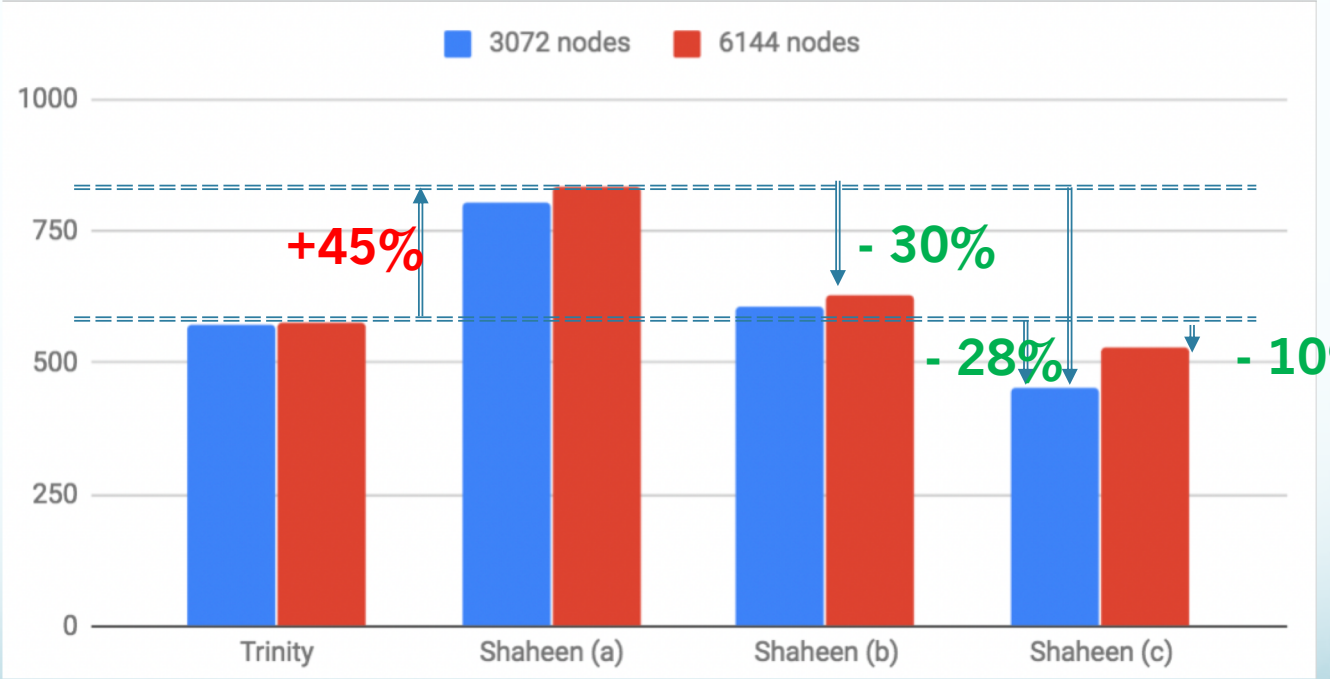
جامعة الملك عبدالله للعلوم والتقنية
King Abdullah University of Science and Technology

```
ANALYSIS OF RESULTS FOR ARIES DIMENSION  2

 For aries with              3  available compute nodes:
 Highest  bandwidth for   3 nodes       8.19175
 Lowest   bandwidth for   3 nodes       8.10847
 Average  bandwidth for   3 nodes       8.15648
 Std. dev.     for   3 nodes     0.241659E-01

 For aries with              4  available compute nodes:
 Highest  bandwidth for   4 nodes       8.55541
 Lowest   bandwidth for   4 nodes       4.34258
 Average  bandwidth for   4 nodes       8.51178
 Std. dev.     for   4 nodes     0.772598E-01


OUTLIER(MORE THAN 5.0% BELOW AVERAGE)FOR ARIES DIMENSION 2
NID  tx ty tz  NID  tx ty tz BdwidthGB/s nodes % deviation
1644, 4,1,11,  1775,  4,3,11,   4.34258      4       48.98
1708, 4,2,11,  1775,  4,3,11,   6.59278      4       22.55
1775, 4,3,11,  1644,  4,1,11,   4.34258      4       48.98
1775, 4,3,11,  1708,  4,2,11    6.59278      4       22.55
```

→ Need to disable nid01755 and its corresponding blade and repeat again the test

Weak scaling (2)

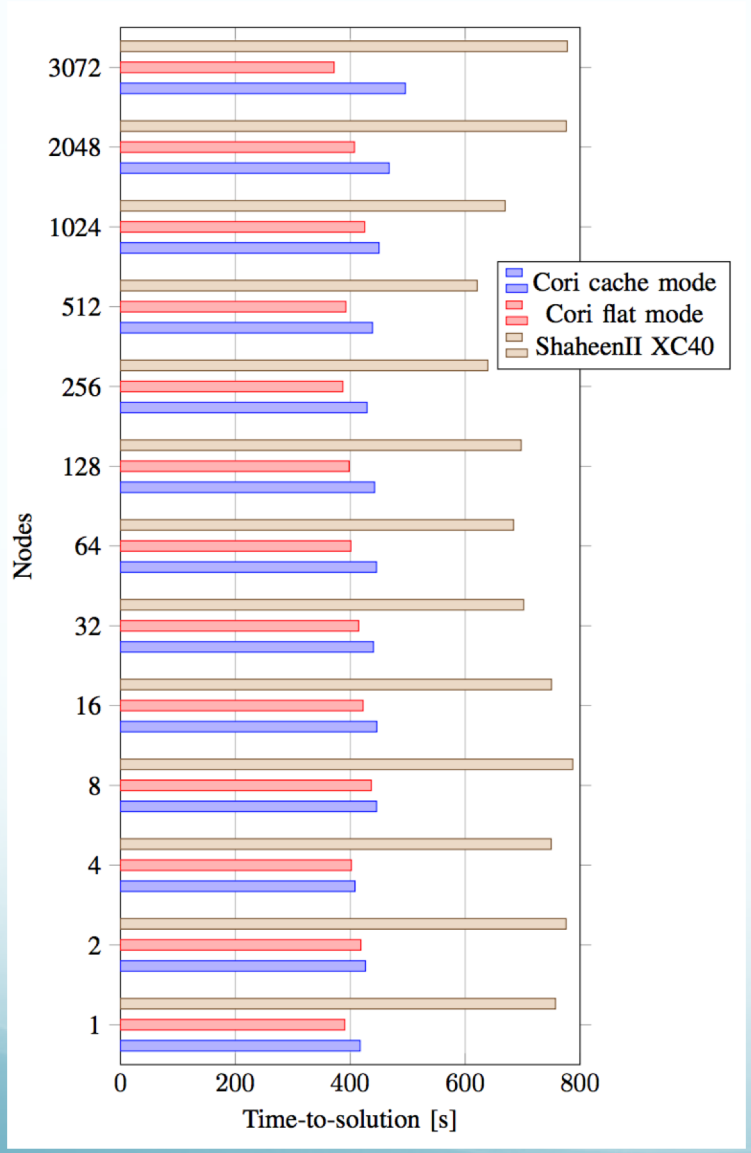
Detecting and isolation weak links, MPI_allreduce test is much faster, down to 86.9µs (very close to Trinity values) while it used to be 130µs as reported.



Performance comparison between Trinity with CLE 6.0,

a/ Shaheen CLE5 b/ Shaheen CLE5 with regression testing , c/ Shaheen CLE 6.4 with regression testing and updated CDT

Cori and Shaheen

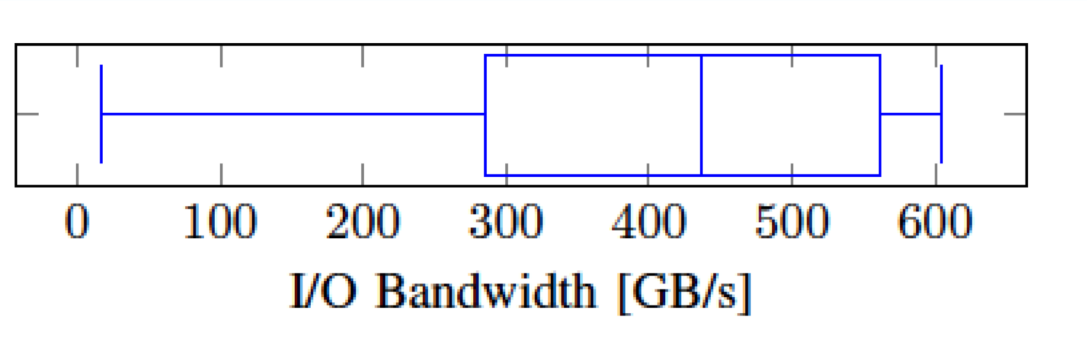


Settings:

- 131,072 grid points per core
- 32 cores per node on Shaheen XC40
- 64 cores per node on Cori XC40
- On KNL: LIBXSMM knows how to target
- AVX-512 (only difference)
- Full production run (e.g. 27 statistics)

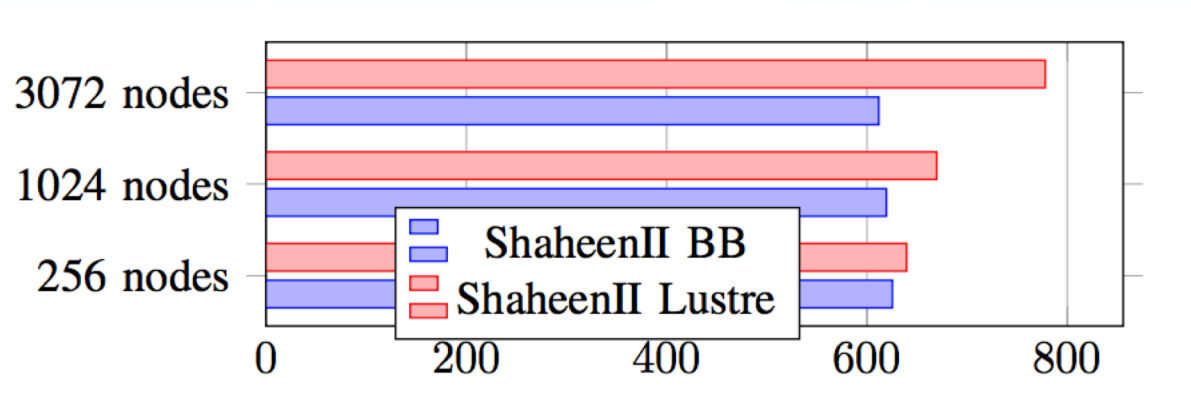
On 3072 KNL nodes:
378 TFLOP/s, 310 TB/s in aggregate

I/O Performance



Distribution of I/O bandwidths on 8192 nodes of Trinity XC40 during heavy third party disk usage

Burst Buffer



- Tuning I/O with Burst Buffer(BB)
  - Test case of 256 nodes and a total of 375 GB of data, using 2 DataWarp nodes, Burst Buffer is only 3% faster than Lustre
  - Peak write performance achieved was 560 GB/s while a maximum of 180 GB/s in writing on Lustre on 3072 nodes
    - provided over 3.11x better throughput using 128 DataWarp nodes
    - overall 23% of performance with 12.8 Billions grid point test case (4.5 TB of data written)

Conclusions

- We have used NEKBOX to perform extreme scale direct numerical simulations
- The simulations sustained over 1 PFLOP/s and 500 TB/s memory bandwidth on 9216 nodes of Trinity XC40, for an overall efficiency greater than 50%.
- On Cori, using KNL, we reached:
  - 378 TFLOP/s
  - an aggregated bandwidth of 310 TB/s ,
    - 2.11x faster time to-solution compared to the same number of Haswell nodes

Conclusions(2)

- Comparison with Trinity, Cori and Shaheen identified hardware issue at scale, thanks to a regression testing.
- Effect of programming environment versions and OS can lead to better performance.
- Regular comparison on standard benchmark shall be shared between HPC centers managing similar architecture systems
  - to detect issues in the hardware or software environment
  - to allow the release of a more reliable and performing HPC programming environment on the Cray XC40 and beyond systems to the users.

Thanks !

- For computer time, this research used the resources of the
  - KAUST Supercomputing Laboratory in Thuwal, Saudi Arabia;
  - NERSC, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231;
  - Trinity project managed and operated by Los Alamos National Laboratory and Sandia National Laboratories