LA-UR-17-29914

Title: Code Modernization of VPIC

Author(s): Bird, Robert Francis
Nystrom, William David
Albright, Brian James

Intended for: APS DPP, 2017-10-23/2017-10-27 (Milaukee, Wisconsin, United States)

Issued: 2017-10-29 (Draft)

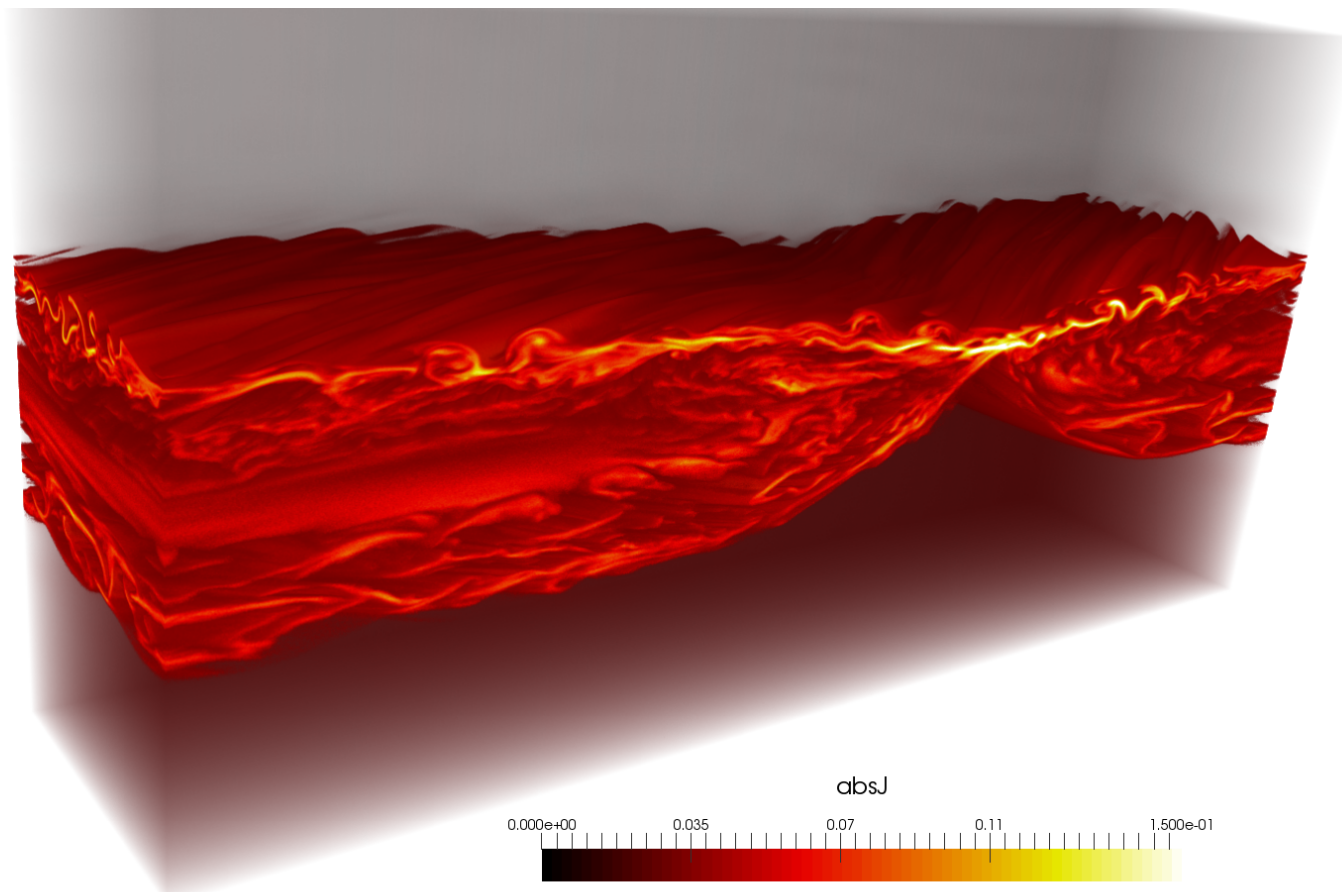
Code Modernization of VPIC

***R. F. Bird*, W. D. Nystrom, and B. J. Albright**
Los Alamos National Laboratory, Los Alamos, NM

Contact: bird@lanl.gov

VPIC

- VPIC is a single-precision 3D relativistic, electromagnetic particle-in-cell (PIC) plasma simulation code which has demonstrated world class performance
- VPIC expresses parallelism at three levels:
  - MPI;
  - Phreads;
  - Vector intrinsics;
- Excellent performance, reduced portability (and readability!)


absJ
0.000e+00
0.035
0.07
0.11
1.500e-01

Goals of this work

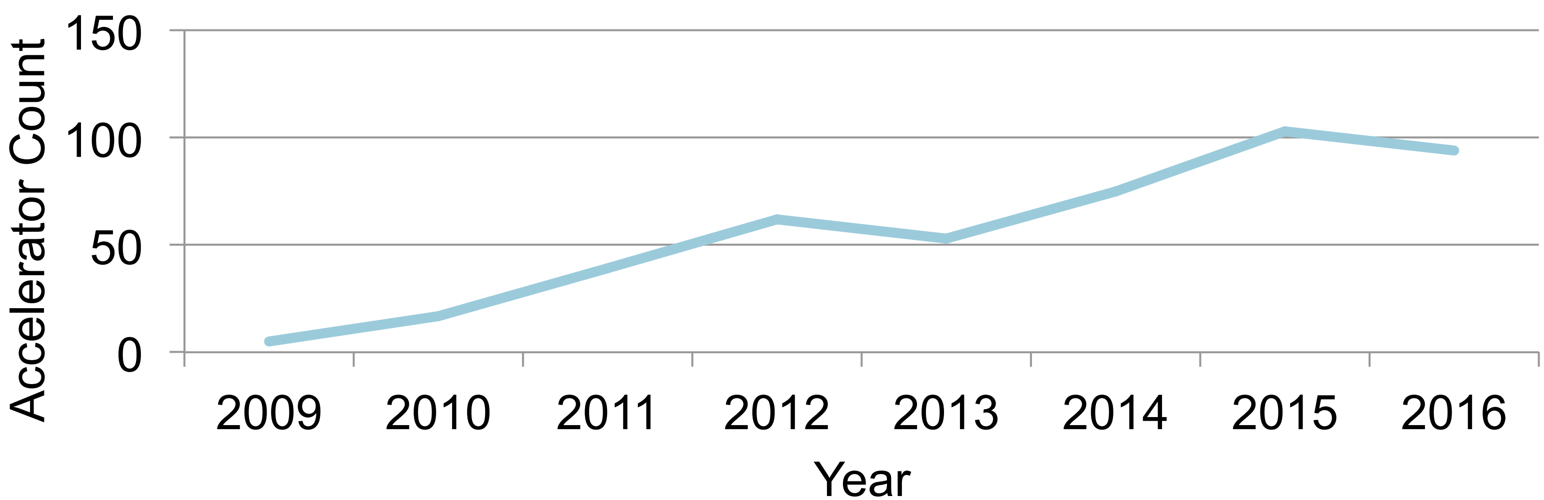
- *Investigate* techniques we can apply to VPIC in order to improve it's portability, performance, and usability
- *Evaluate* the effectiveness of such techniques, and assess their suitability for the main VPIC code branch
- *Understand* the wider implications of such codes changes, and how they can affect the overall code landscape as we move towards exa-scale

Heterogeneity

- Heterogeneous platforms are becoming increasingly common in the modern HPC landscape
  - Figure 1 shows the significant increase in accelerator based supercomputers in the TOP500. With the majority of the Top 10 relying on non-traditional architectures
- This diversity offers increased computational performance, at the cost of programmability
  - Codes often need to be re-written to use the new hardware
  - Unclear if a good, single source, solution exists. Attempts such as OMP4.5 and OpenACC try and bridge this gap
- Cost of maintaining code scales with both it’s complexity and also it’s variability

Heterogeneity

Figure 1: Accelerator Count by Year for machines in the TOP500



Code Portability

- Many codes outlive the machines they will be run on, and have been around for decades (legacy codes)
- It is desirable for such codes to be portable -- able to effectively utilize a variety of hardware
  - Often this means be able to make use of: CPUs, GPUs, and Co-processors (Intel Xeon Phi)
- This can be achieved through a single-source solution (more desirable), or through the development of a code version for each hardware type (less desirable)

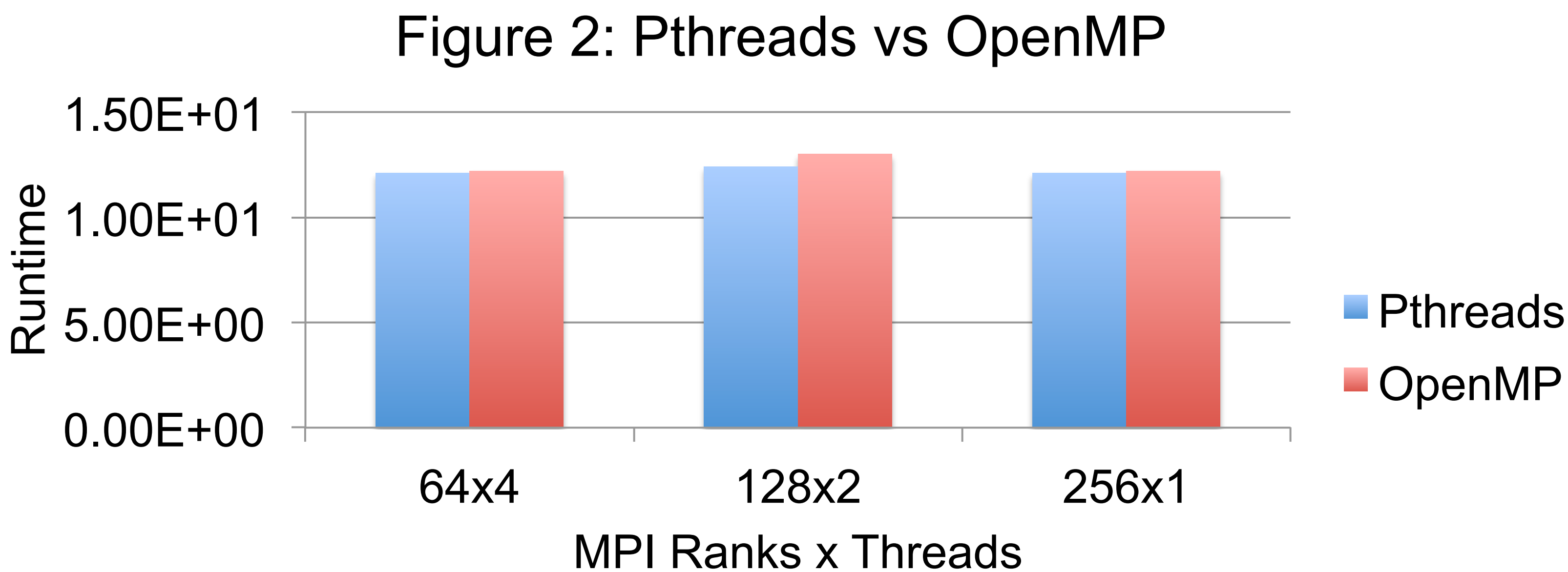
Code Performance

- It is not sufficient for codes to be portable, they also need to offer good performance
- Often these ideas are combined, referred to as *portable performance*
  - This is very hard to define concisely!
- Typically people say a code offers portable performance when it can:
  - Run on the required hardware
  - Achieve a reasonable level ("good enough") percent of peak performance available for that hardware

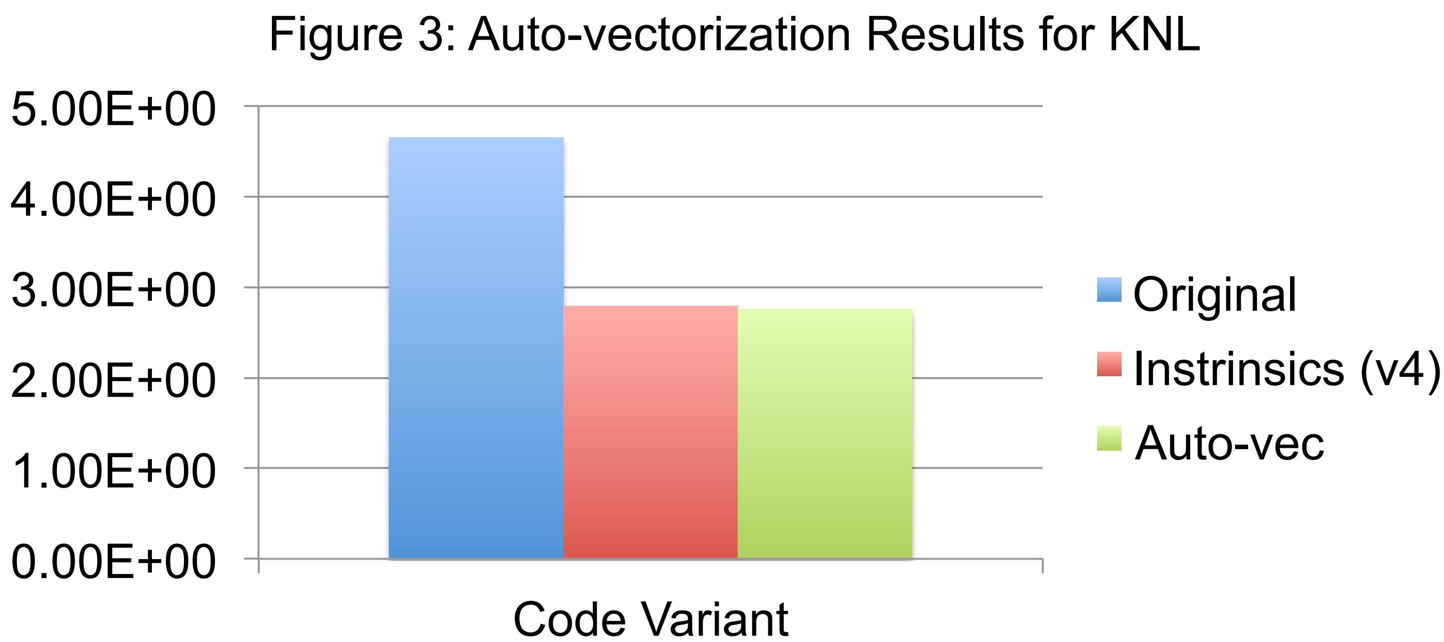
Previous Work

- In our previous work we presented the OpenMP port of VPIC, which was able to achieve identical performance to the Pthreads implementation.
- We also presented the new auto-vectorizing version of the code which is able to match the performance of the hand written intrinsics.
- These represent a shift towards simple more usable code.

Threading Model Results



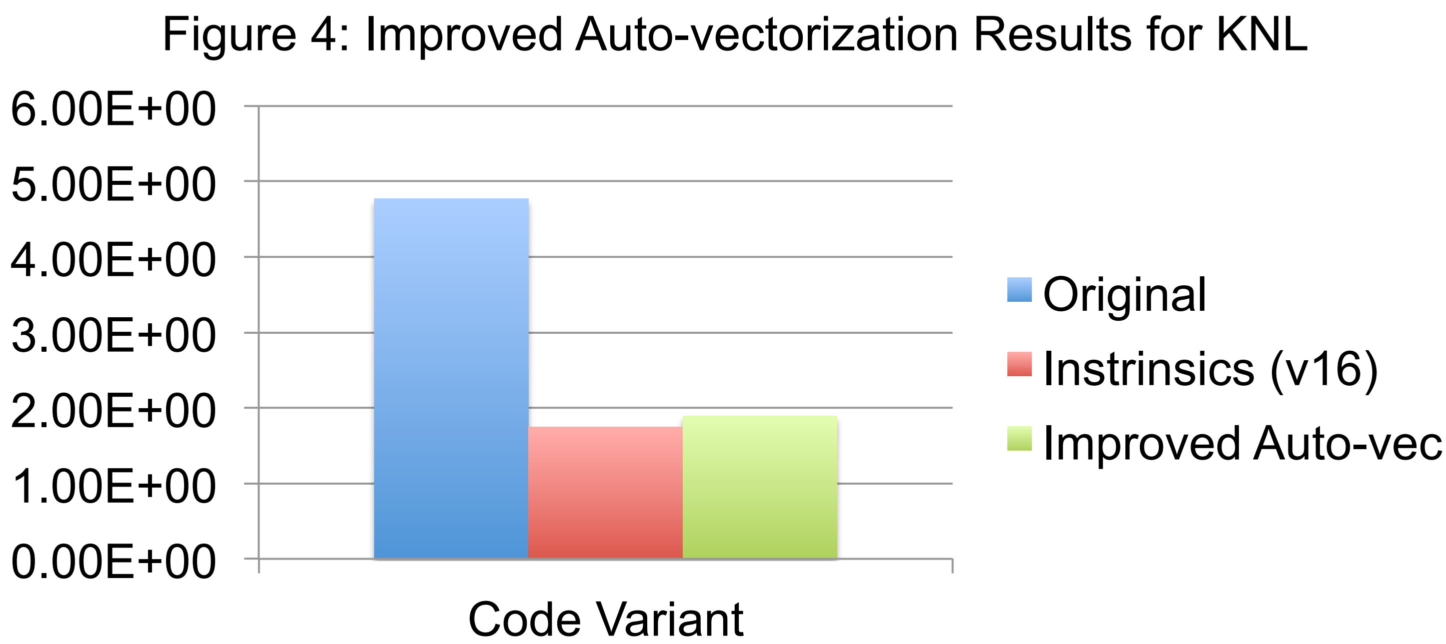
Auto-vectorization Results



Previous Work

- Since the previous work, the hand written intrinsics have been extended to 16 wide vectors, and performance has been improved to match (Figure 4).
- Work has continued, and new techniques have been developed to deal with complex data patters in a SIMD fashion, including the introduction of a SIMD Queue (more on this later)

Auto-vectorization Results



VPIC Enhancement

- Various efforts have been implemented to enhance the performance of VPIC. These include:
  - Techniques to increase vectorization (SIMD Queue)
  - Techniques to reduce memory gathers (per cell iteration)
  - Techniques to reduce memory scatters (scatter array privatization)
  - Hand tuning of the advance_p vectorization

SIMD Queue

- VPIC algorithm needs to do extra work on particles which leave a given cell
- Traditional approach has a small number of particles per SIMD lane taking an expensive code branch to deal with this
- VPIC enhanced to instead collect up particles into a small buffer of size (2)VLEN, and perform the extra computation once it's full
- Allows branch to be executed in full 16 wide SIMD => improved vectorization
- Expressed as a generic SIMD Queue, applicable to many other algorithmic motifs.

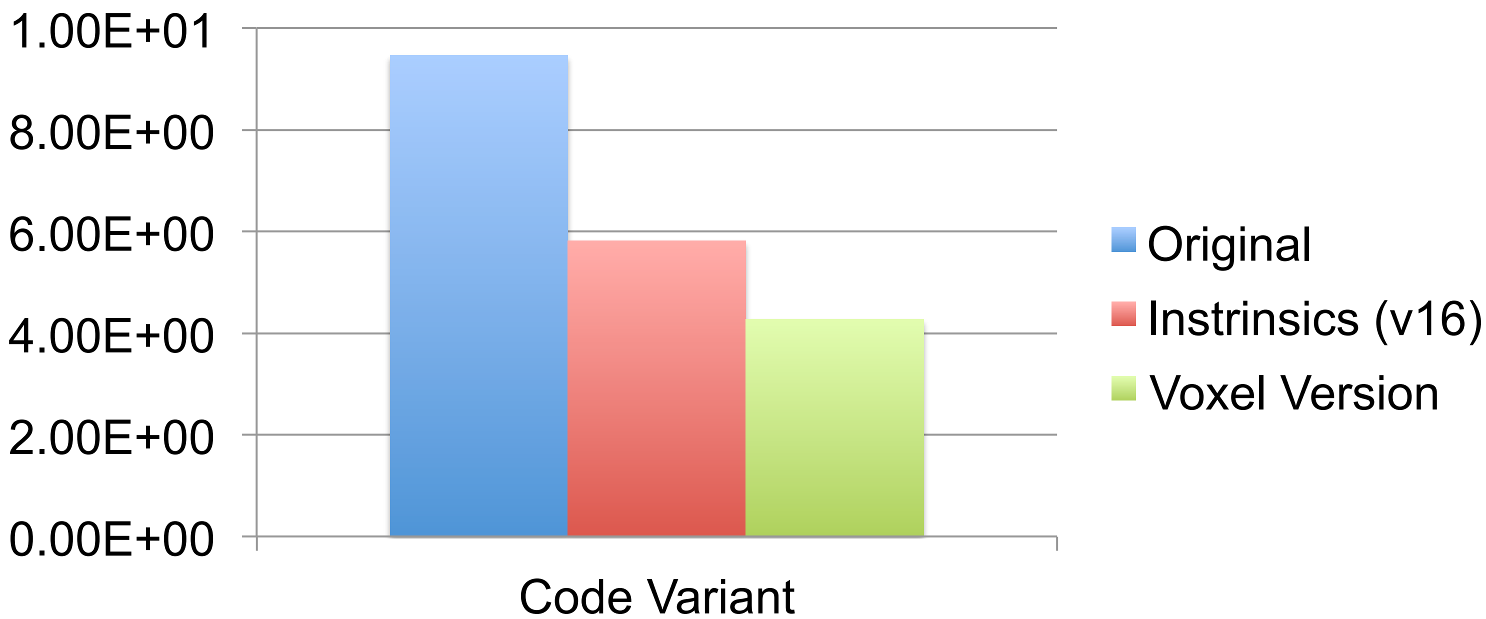
Per-cell iteration

- Switch to a scheme where parallelism is expressed over cells, not over particles.
- For low order particles, only a single field cell is needed (removing gathers).
- Significantly reduces the need for field data duplication
- Requires particles to [always] be well sorted, which incurs an overhead
- Potential to reduce need for data duplication (or the amount of atomics needed for a GPU computation)

Per-cell Iteration

```
for (int icell = 0; icell < num_cells; icell++)
{
   int num_particles = particles_per_cell[icell];
   for (int i = 0; i < num_particles; i++)
   {
      ... // physics
   }
}
```

SIMD Queue + Per-cell



HBM Scaling

- KNL's offer on-package high bandwidth memory, which has significantly higher memory bandwidth than DDR.
  - To fully exploit the capabilities of KNL, we need to use this.
- PIC problems are frequently memory intensive, and are often scaled to fit within node memory.
- For a given problem (with fixed memory requirements), we may see super-linear increases in performance from exclusively using HBM.
- To test this we can scale a problem into HBM
  - Figure 5 shows a ~46% Speedup for 8 KNL nodes

HBM Scaling

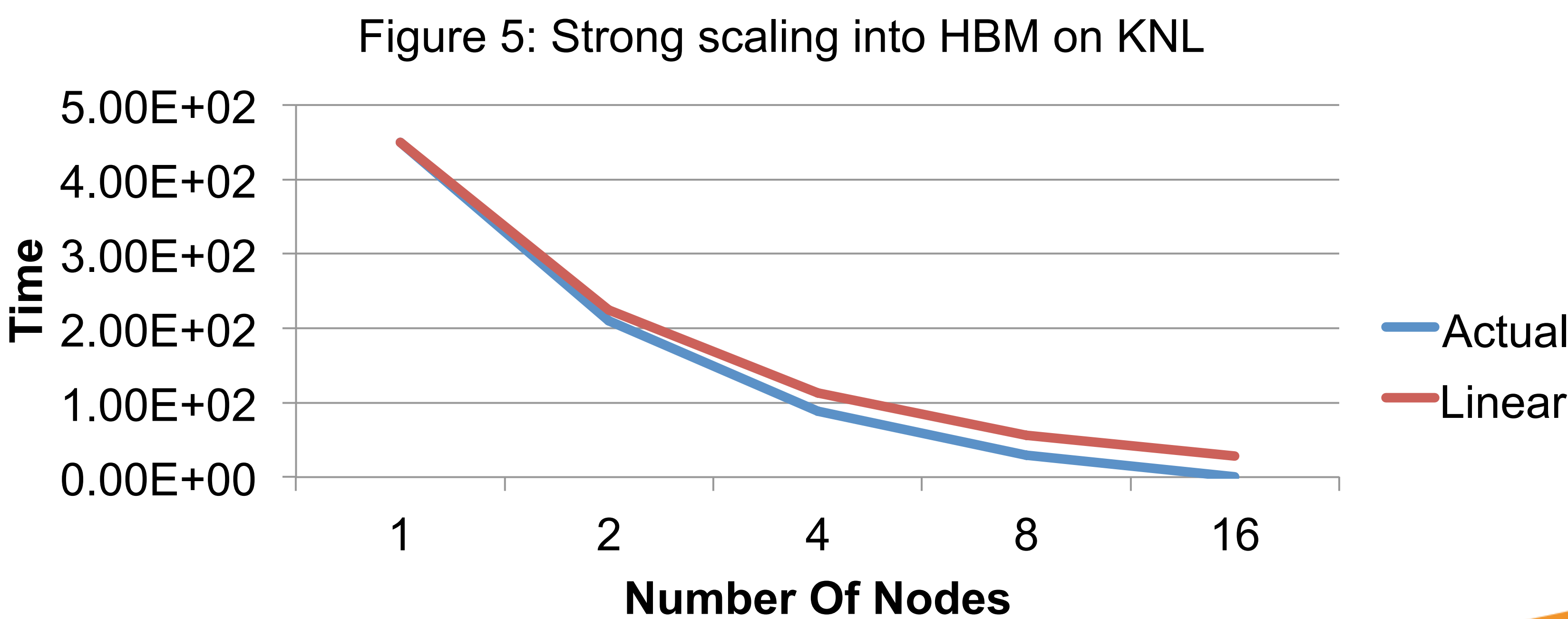


Figure 5: Strong scaling into HBM on KNL
Time
5.00E+02
4.00E+02
3.00E+02
2.00E+02
1.00E+02
0.00E+00
1
2
4
8
16
Number Of Nodes
Actual
Linear

Open MP 4.5

- OpenMP 4.5 allows for GPU offload of code using pragmas
  - Good attempt at a “single source” solution
  - Can be run on the CPU or the GPU
- Typically uses pragmas such as:
  - #pragma omp distribute parallel for
  - Limitation: lack of evidence to show that the single source solution will give near-optimal performance on all hardware

Open MP 4.5

General Code Skeleton for OpenMP 4.5

```
#pragma omp target data …

#pragma omp target teams num_teams(256) thread_limit(128)
{
  #pragma omp distribute parallel for
  for (int i=0; i<n; i++)
  {
     … // physics
   }
}
```

VPIC + Open MP 4.5

- OpenMP 4.5 Implemented for main VPIC particle pusher kernel
- Successfully offloaded to GPU
- Data copy represent a significant performance overhead
- Future work can look to address this be ensuring more kernels are resident on the device.

Automated Testing

- To support on-going code development (including the addition of new physics capabilities), regression testing is being added.
- Aims to detect regressions in accuracy and to detect if bugs/ errors are introduced.
- Possible extension to include performance regression.
- Crucial step towards becoming a wider used community code.

Conclusion

- Lots of exciting development is on-going
- Code is moving towards a cleaner and more usable format, whilst still retraining exceptional performance.
- Starting to prepare for GPU machines, but the best path isn't yet clear
- Moving to a place where it can function as a productive open source code, with external contributors.

References

- VPIC Source Code: https://github.com/losalamos/vpic
- Bird, R., Peters, E., Nystrom, D., & Albright, B. (2016, October). Improving the Performance and Portability of VPIC.
- K. J. Bowers, B. J. Albright, L. Yin, B. Bergen, and T. J. T. Kwan, “Ultrahigh performance three-dimensional electromagnetic relativistic plasma simulation”, Physics of Plasma, vol 15, no. 5, 2008.
- K. J. Bowers, B. J. Albright, B. Bergen, L. Yin, J. Barker, and D. J. Kerbyson, “0.374 Pflop/s Trillion-Particle Kinetic Modeling of Laser Plasma Interaction on Roadrunner”, SC 2008: Proceedings of the 2008 ACM/IEEE Conference on Supercomputing, (Piscataway, NJ, USA: IEEE Press) pp 1-11.
- K. J. Bowers, B. J. Albright, L. Yin, W. Daughton, V. Roytershteyn, B. Bergen, and T. J. T. Kwan, “Advances in petascale kinetic plasma simulation with VPIC and Roadrunner”, Journal of Physics: Conference Series 180 (2009) 012055.
- TOP500: https://www.top500.org

Acknowledgements

- Work performed under the auspices of the U. S. Dept. of Energy by the Los Alamos National Security, LLC Los Alamos National Laboratory under contract DE-AC52-06NA25396 and supported by the LANL LDRD program.