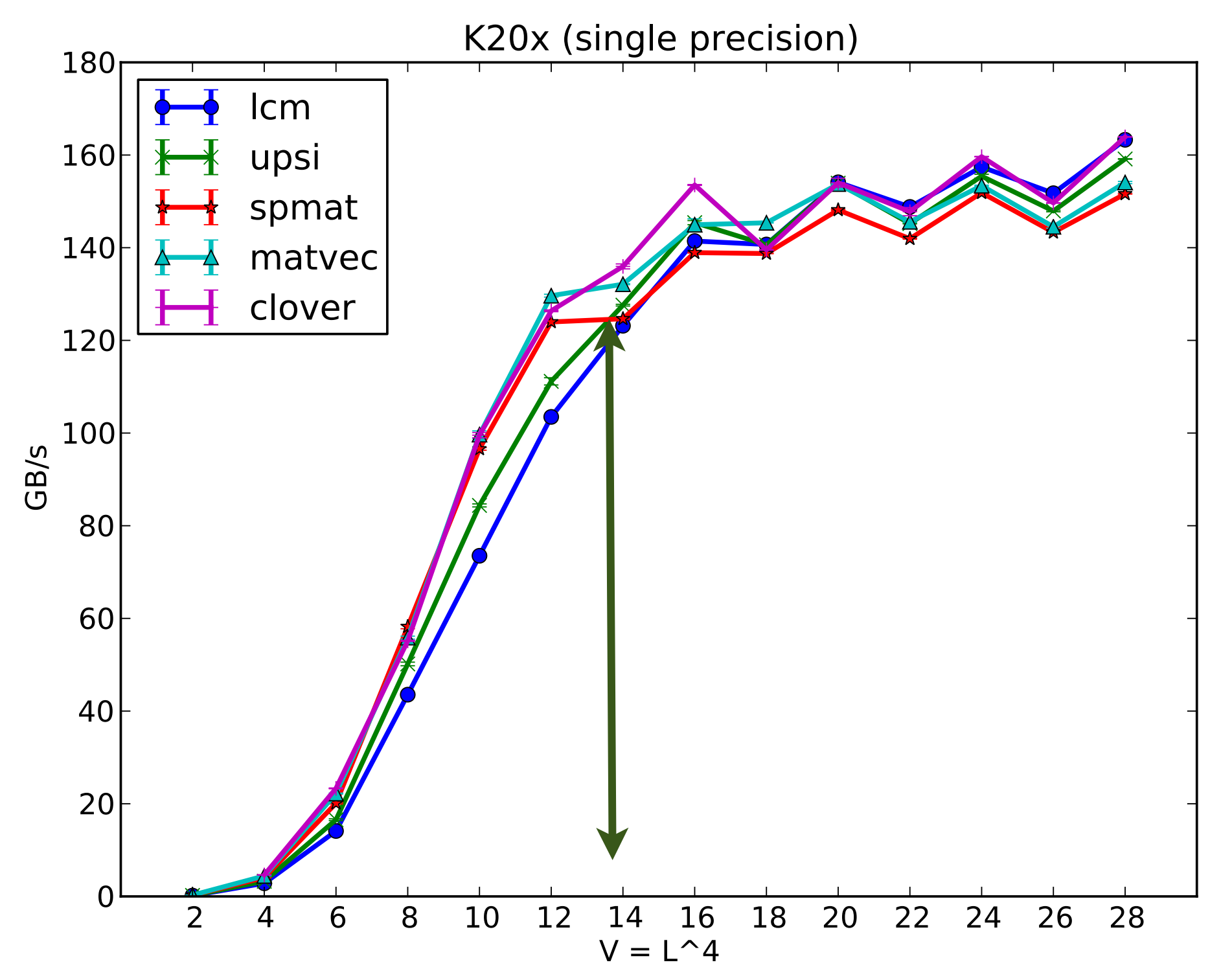
Porting Lattice QCD Calculations to Novel Architectures

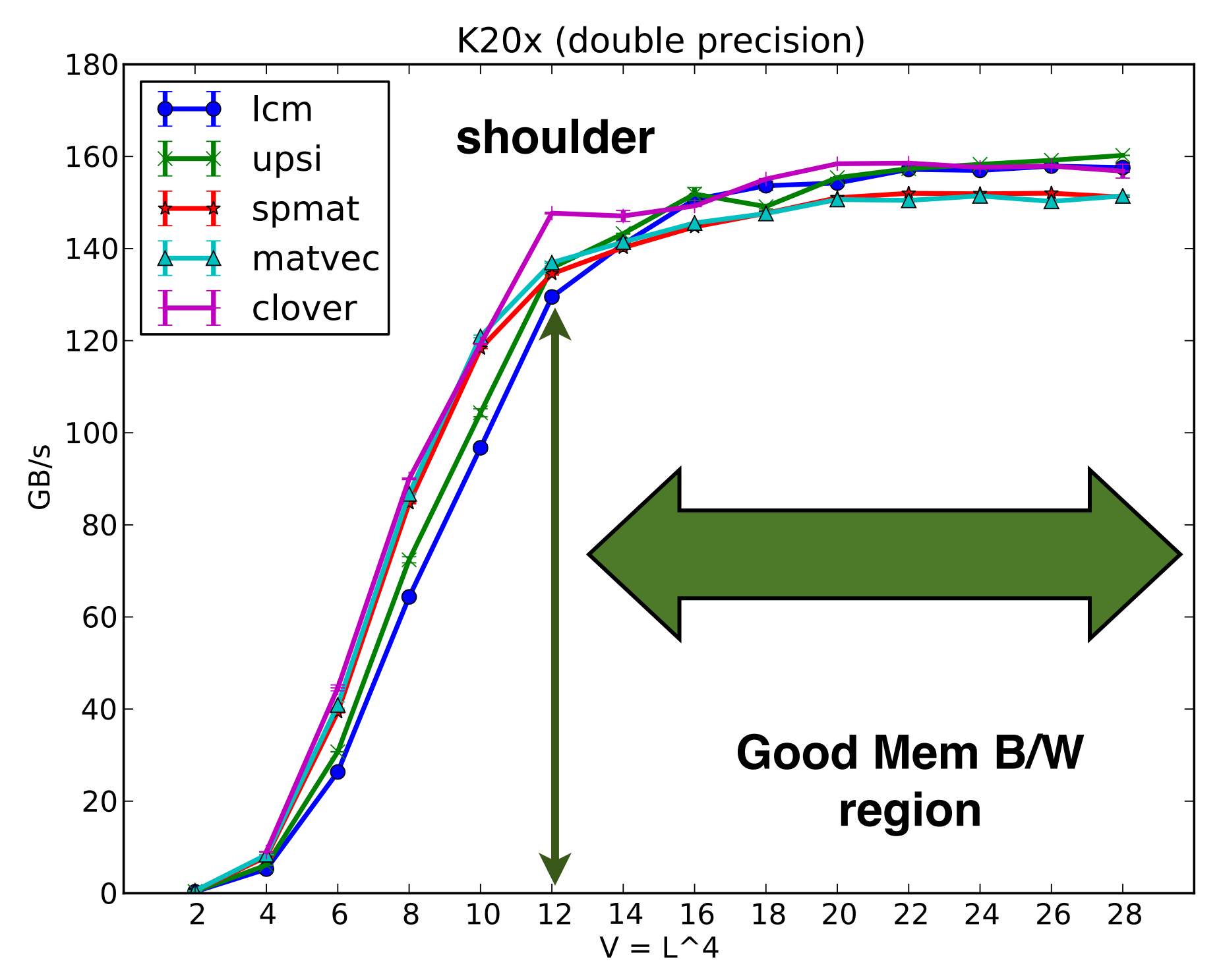
Balint Joo, Frank Winter, Jefferson Lab, for the USQCD Collaboration
Computing Properties of Hadrons, Nuclei and Nuclear Matter from Quantum Chromodynamics
bjoo@jlab.org, fwinter@jlab.org

QDP-JIT and QUDA: Enabling Chroma on GPU Based Leadership Architectures

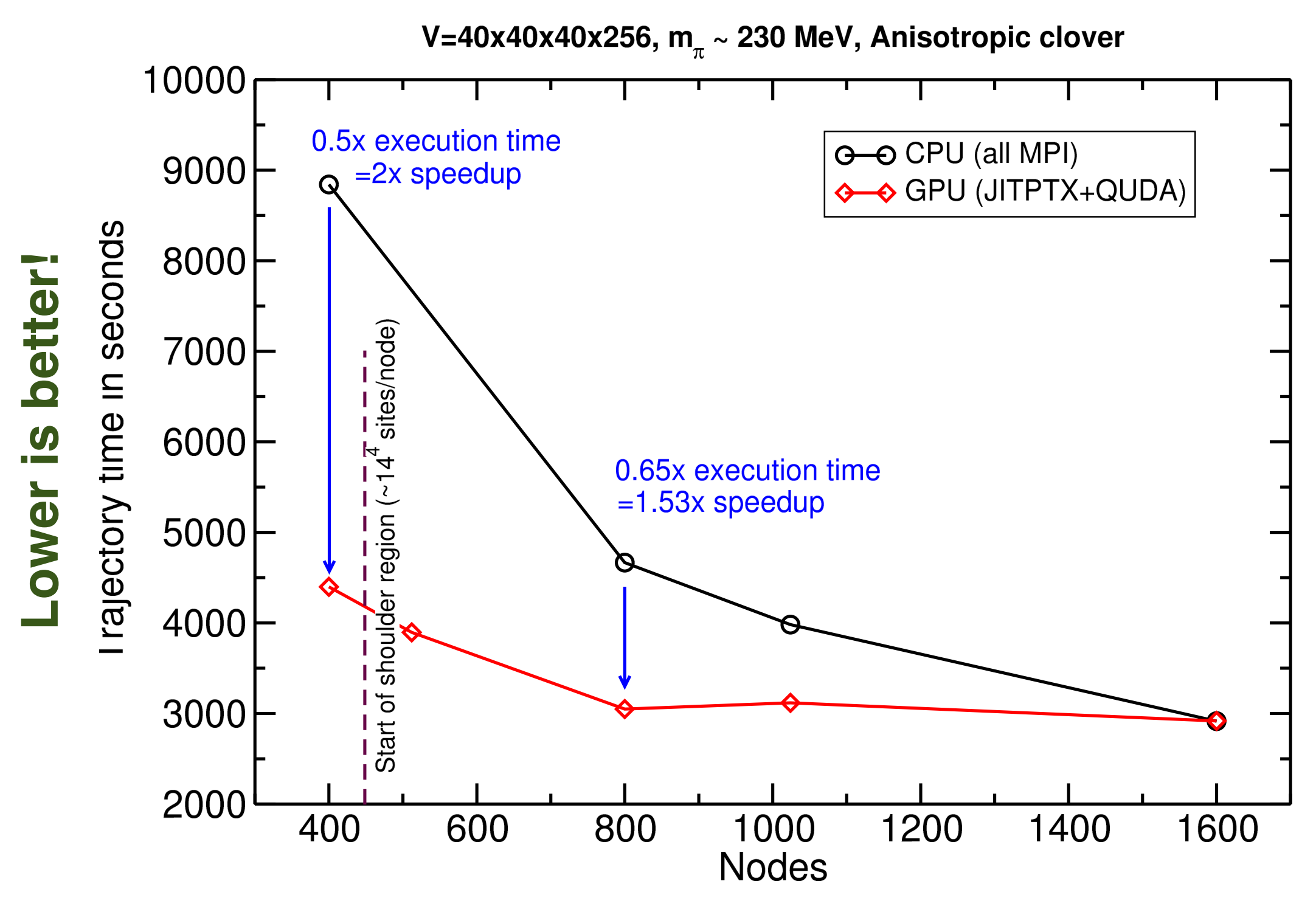
The Chroma software system is the standard workhorse of the gauge generation phase of LQCD calculations in Cold Nuclear Physics on leadership class systems such as OLCF Titan, which feature GPU accelerated nodes. Chroma has long enjoyed accelerated solvers utilizing libraries such as QUDA. Gauge generation, however, requires the whole application to be accelerated, to avoid *Amdahl's Law effects* in parts of the code *outside of QUDA*.

QDP-JIT is an implementation of the QDP++ layer on which Chroma is built. QDP++ expression templates are *compiled into code generators* which generate *CUDA-PTX kernels* for the expressions when first run; at which point *grid and block dimensions are autotuned.* QDP-JIT features a memory manager which can *page data between host and device memories automatically* and *rearrange data to the most optimal layout* (e.g. for coalesced access) *without the need to instrument the large Chroma code with #pragma annotations*

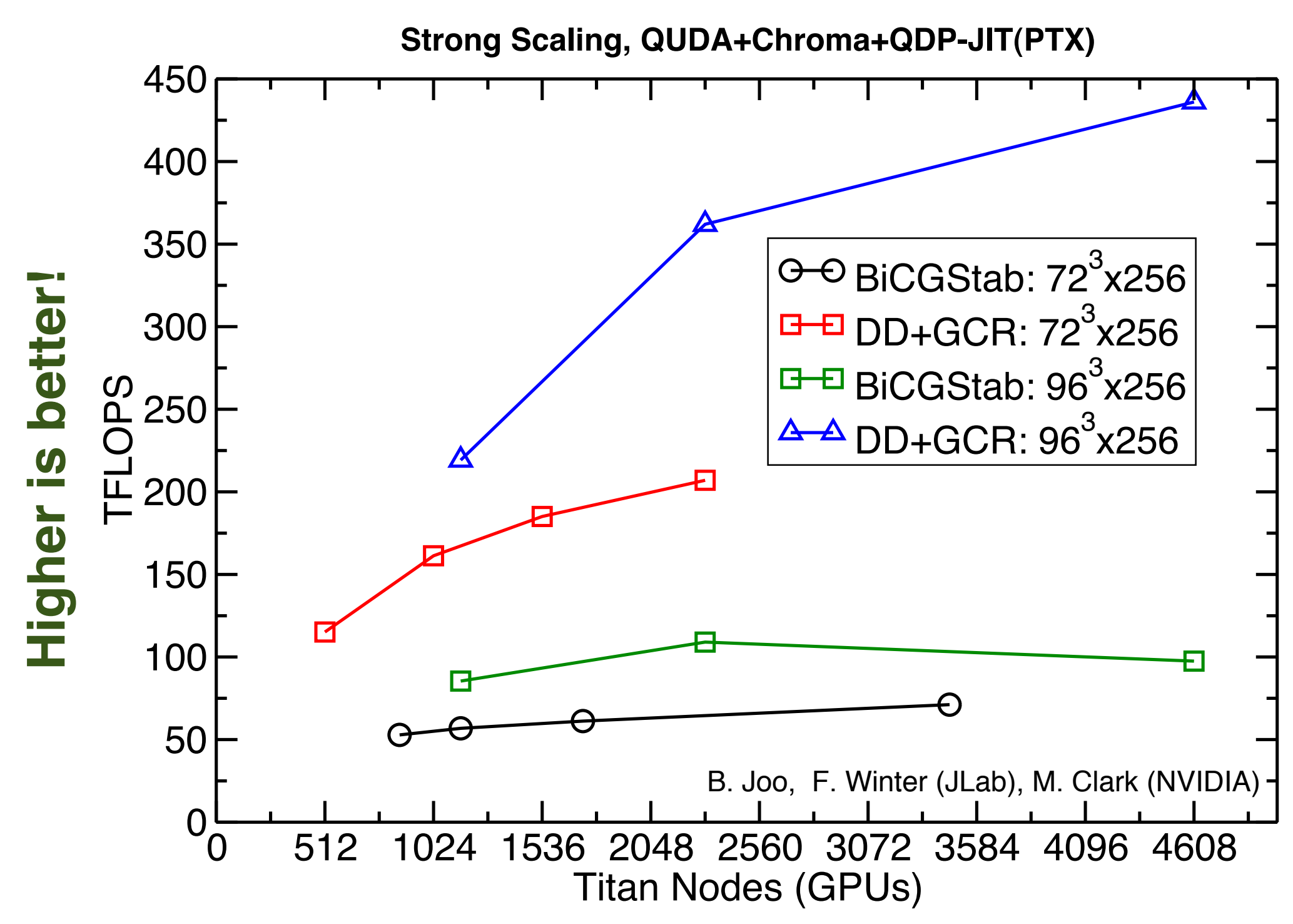




Single node performance of QDP-JIT on NVIDIA K20x GPUs for some test expressions. Memory B/W is saturated at around 150-160 GF or about 80% of peak. Problems smaller than 12^4-14^4 sites are too small to saturate the B/W.



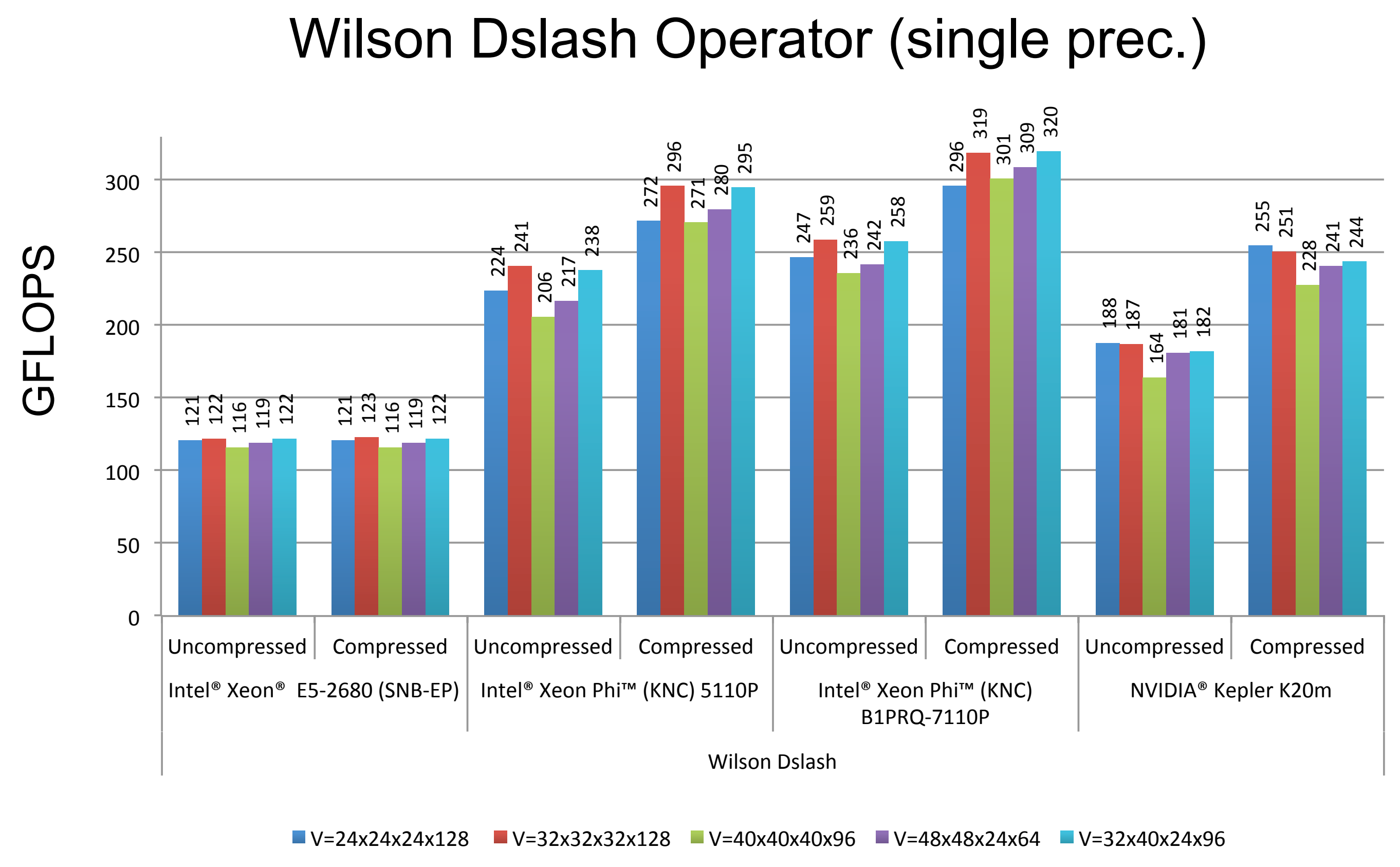
Strong Scaling of HMC Gauge Generation on Titan using the CPU or the GPUs with QDP-JIT and QUDA. Substantial acceleration is reached, however the strong scaling behaviour is worse than for the CPU. We attribute this in part to PCIe bottlenecks and in part to entering into the "shoulder region" where the local volume per GPU becomes small

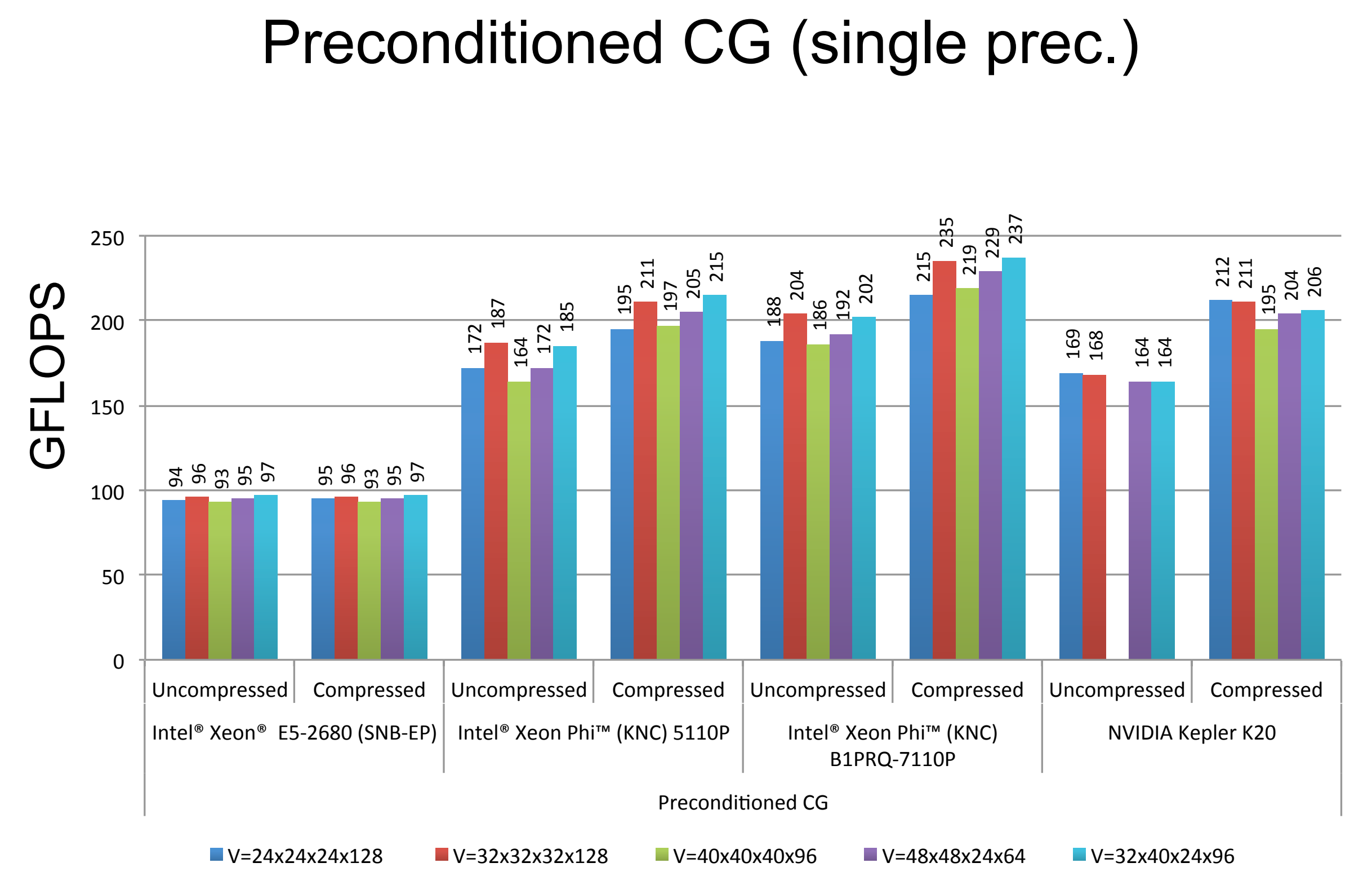


Strong Scaling of GCR Solver on Titan, using larger volumes. Scaling Clearly improves with growing volume. The performance of the DD+GCR algorithm at 2304 GPUs grows by nearly 2x going from a volume of 72^3x256 to 96^3x256 sites (a growth in the volume of 2.37x)

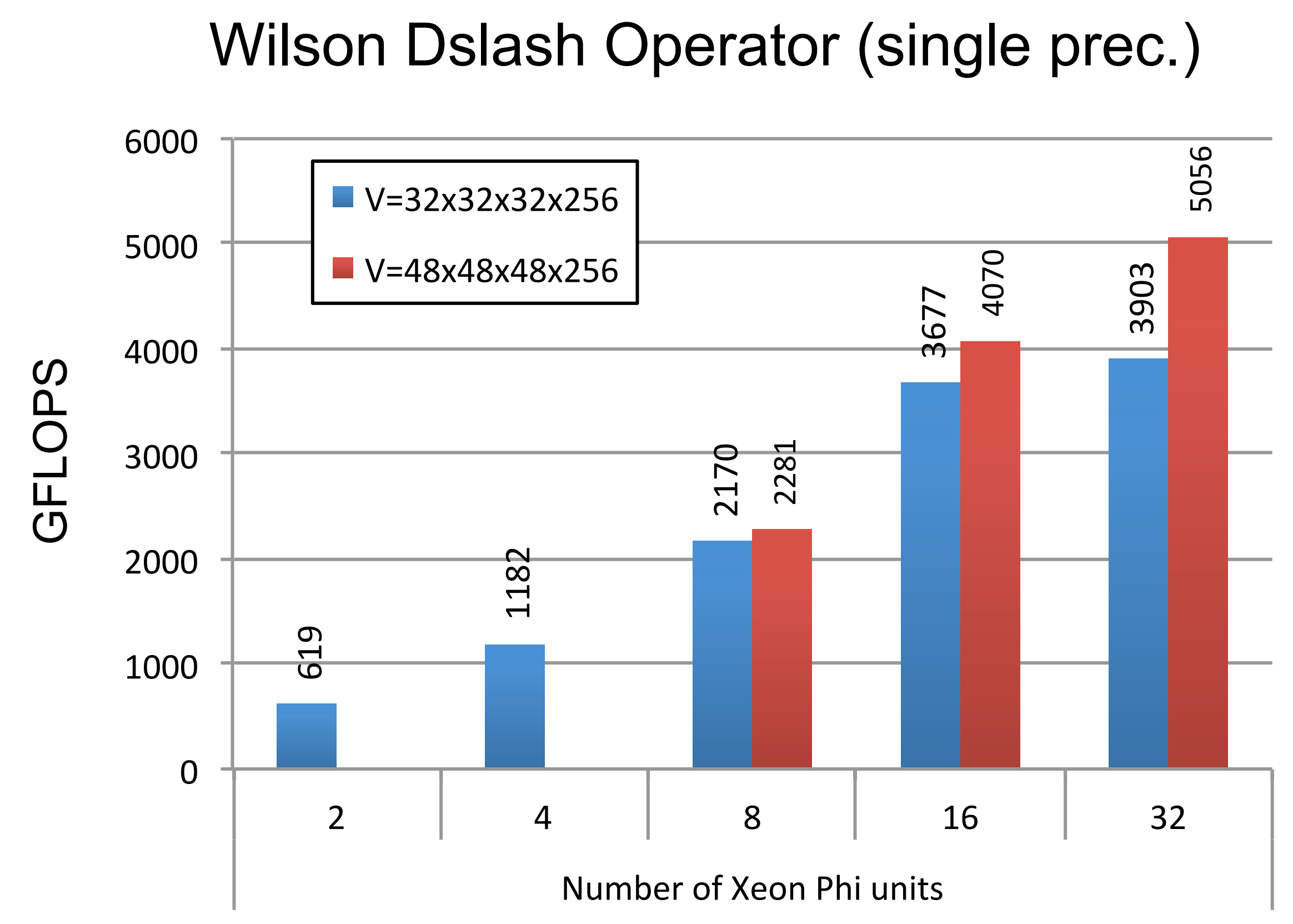
Optimizing for Xeon Phi

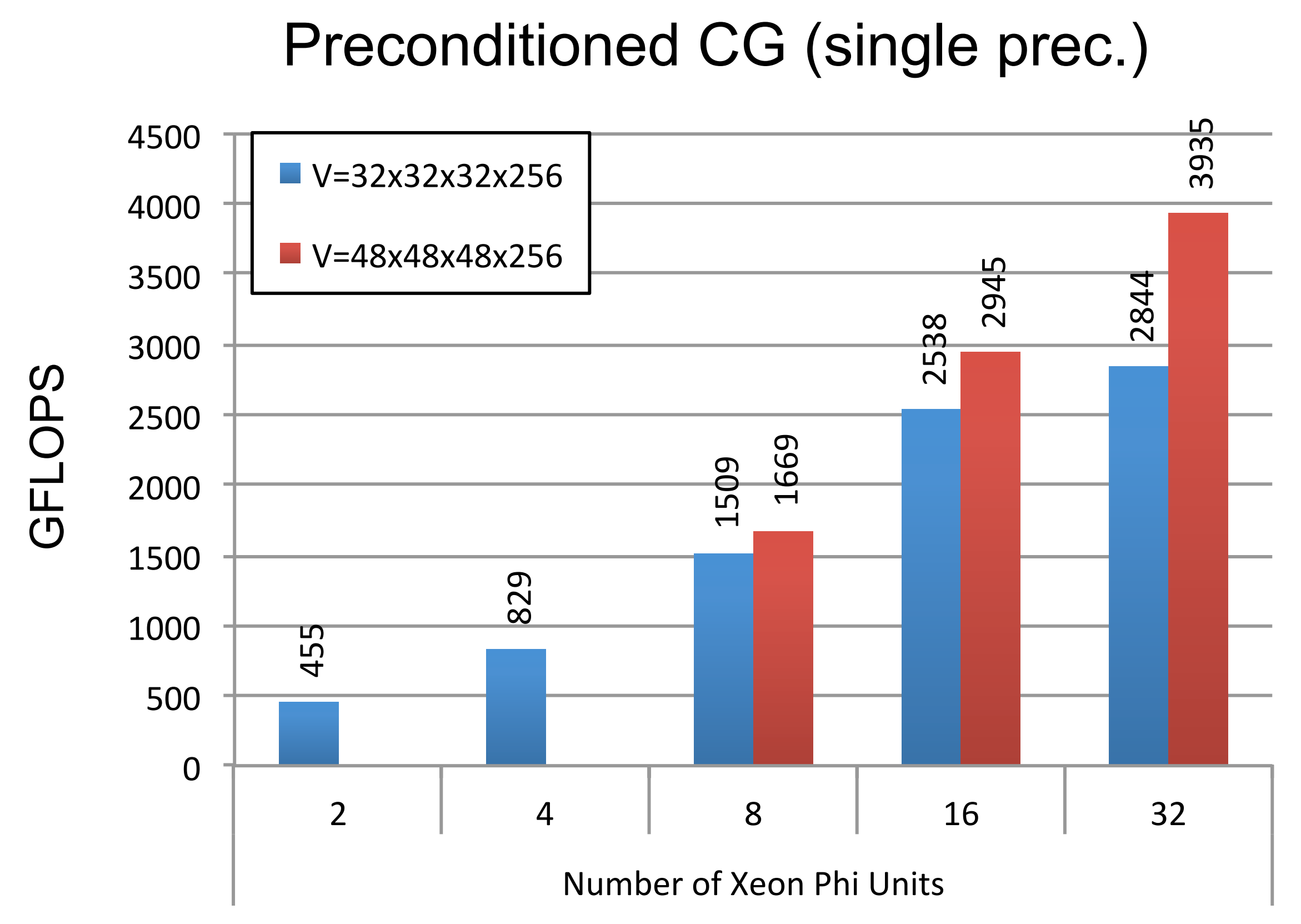
We have been working in close collaboration with *Intel Parallel Labs* to develop high performance implementations of Lattice QCD kernels for Intel Xeon Phi. To achieve good performance requires careful attention to *vectorization*, cache-blocking, and mapping threads to blocks in a load balanced way. We wrote a *code-generator* to abstract *vector intrinsics* and to allow us to vary the spacing of prefetch instructions. The *multi-node code uses an MPI Proxy to pick the optimal path* betwen device nodes.The f*ramework has been retargeted* to AVX on Intel Sandy Bridge and should be straightforward to apply to other multi-core, cache and vector based architectures such as BG/Q.





Performance of LQCD Kernels on Single Devices compared with performance of QUDA on Tesla K20 GPUs. The CPU performances are also excellent. (B. Joo, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. Lee, P. Dubey, W. Watson, ISC 2013, LNCS 7904, pp40-54, 2013)





Performance of LQCD Kernels on Multiple Xeon Phi nodes of the Intel Endeavor Cluster, using an MPI communications Proxy (B. Joo, D. Kalamkar, K. Vaidyanathan, M. Smelyanskiy, K. Pamnani, V. Lee, P. Dubey, W. Watson, ISC 2013, LNCS 7904, pp40-54, 2013)

Conclusions

QDP-JIT will allow effective exploitation of accelerated leadership resources, and forms the basis of our work in *partnership with the SUPER SciDAC Institute* to create a Domain Specific Compilation Framework for lattice QCD. The lessons learned can be applied to other domain specific frameworks using expression templates. Our work with Xeon Phi seeks to discover approaches for highly performant code on this architecture, targeting large scale Xeon Phi resources (e.g. Stampede) and to quantify the "Ninja Gap" between optimized and "regular" code to help design future compilers. Finally, the micro-benchmarks developed through our collaboration with Intel can be used to stress and evaluate proposed architectural changes on future version of Xeon Phi