Petascale X-Ray Scattering Simulations with GPUs

Abhinav Sarje **Xiaoye S Li** **Slim Chourou**
Computational Research Division
Lawrence Berkeley National Laboratory

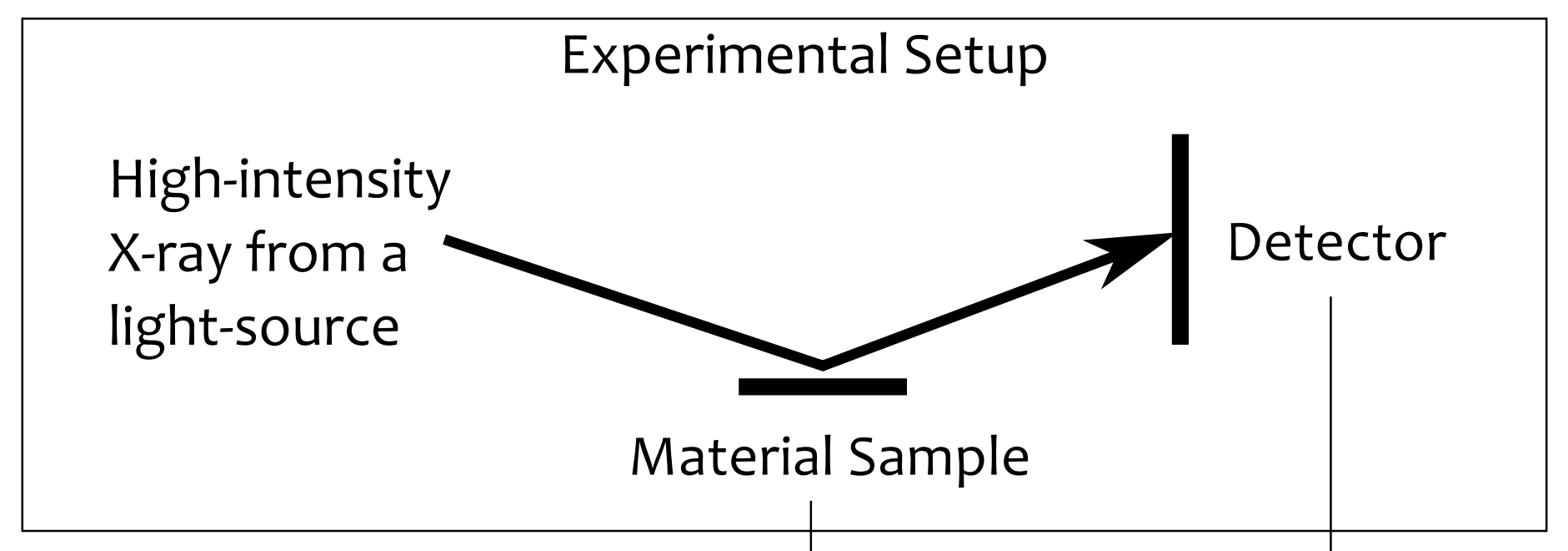
Alexander Hexemer
Advanced Light Source

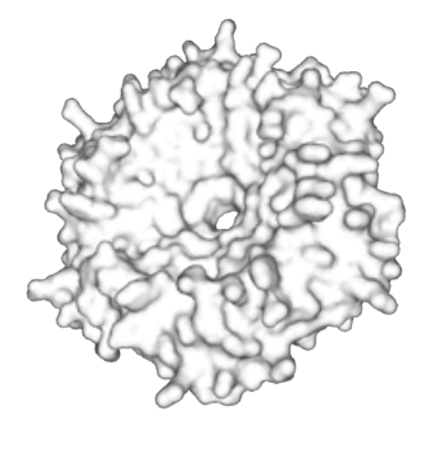
Abstract

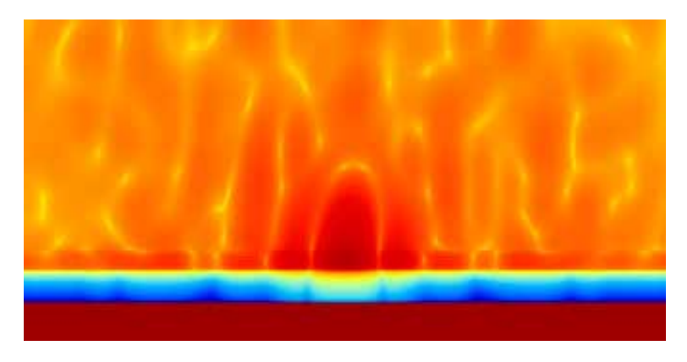
We present **HipGISAXS**, a massively-parallel high-performance X-ray scattering simulation code, its development, and architecture-aware optimization and tuning. HipGISAXS is short for **Hi**gh-**P**erformance **GISAXS**. GISAXS, or **G**razing **I**ncidence **S**mall **A**ngle **X**-ray **S**cattering, is a kind of X-ray scattering, used widely for probing nanostructures in material samples at synchrotron light-sources. Analysis of the data generated at such light-sources has been a bottleneck due to its high computational needs as well as high data generation rate. We address this challenge through the use of massively-parallel high-performance computing. We target clusters and supercomputers built with multi-cores and many-core processors. We exploit the high parallelism available in graphics processors and demonstrate the effectiveness of such platforms for X-ray scattering simulations and data analysis. We present detailed performance results of HipGISAXS on various Nvidia GPUs, achieving almost 2 Petaflops on Titan, a Cray XK7, and compare with traditional systems.

Background

GISAXS is a widely used tool by scientists for the characterization of macromolecules and nanoparticle systems based on their structural properties at nanoscales. A major application of this is in the characterization of materials for the design and fabrication of energy-relevant nanodevices, such as photovoltaic cells.



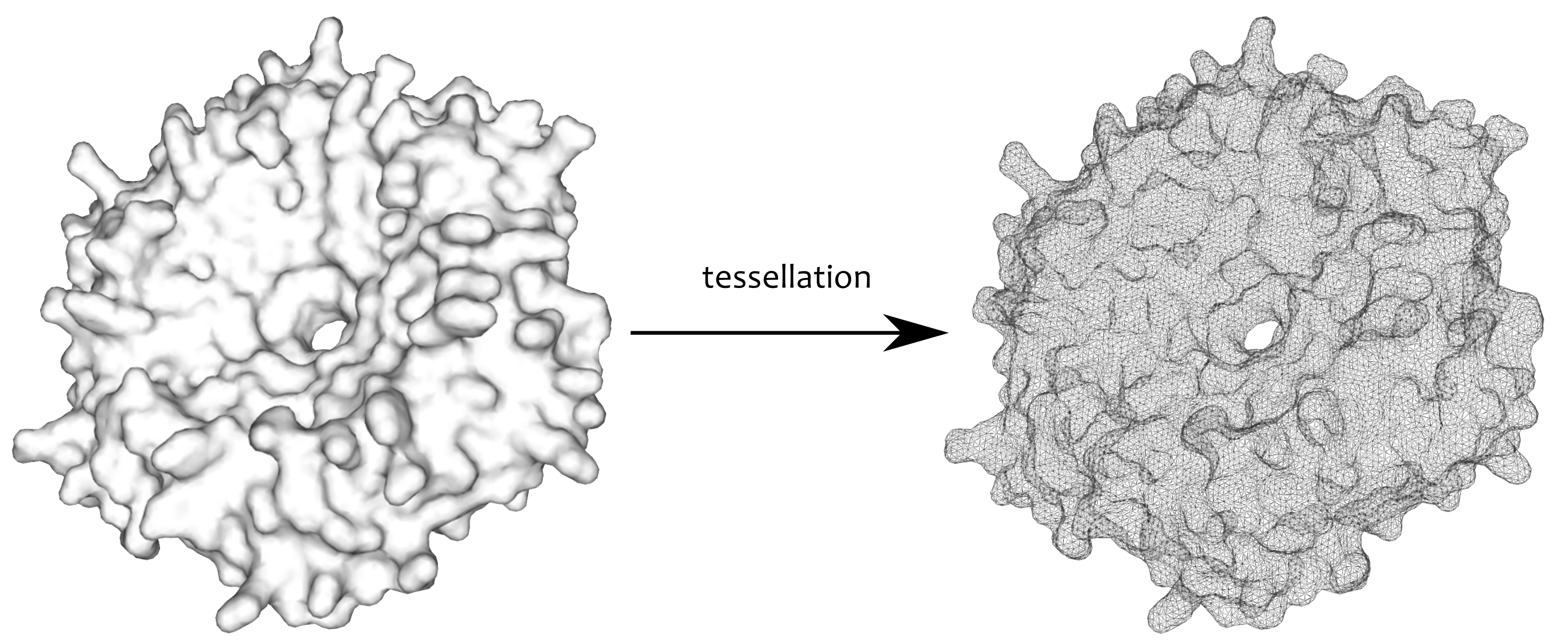

A complex protein molecule


A scattering pattern

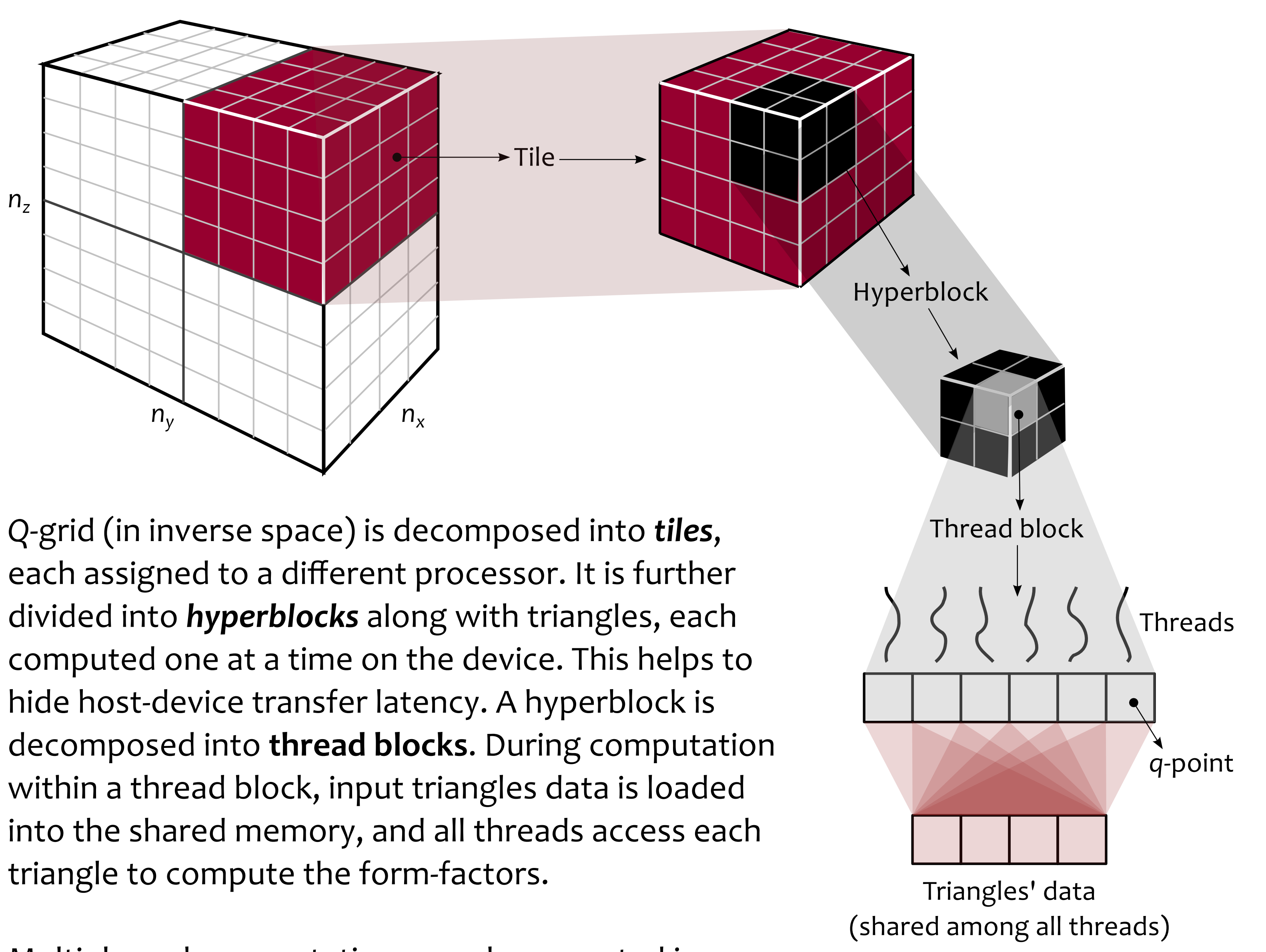
The Computational Kernel

This GISAXS scattering pattern simulation code is based on the **Distorted Wave Born Approximation** (DWBA) theory, and involves a large number of compute-intensive form- factor calculations. A **form-factor** at a given point $\vec{q}$ in the inverse space is computed as an integral over the shape volume functions of the nanoparticles in a given sample. By applying Green's theorem, this volume integral can be transformed into a surface integral over the shape boundary. The scattered light intensity at a point is proportional to form-factors at that point. A simulated sample structure is taken as an input in the form of discretized structure-surfaces, such as a triangulated surface. For our computational purposes, form-factor is described as a summation over the discretized surface of the shapes under consideration. Such form-factors are computed for all points in a inverse space grid, the **Q-grid**.

$$F(\vec{q}) = -\frac{i}{q^2}\int_{S(\vec{r})} e^{i\vec{q}_r\cdot\vec{r}} q_n(\vec{r})\, d^2\vec{r} \xrightarrow{\text{triangulate structure}} -\frac{i}{q^2}\sum_{t=1}^{n_t} e^{i\vec{q}\cdot\vec{r}_t} q_{n,t} s_t$$



Parallelism Hierarchy



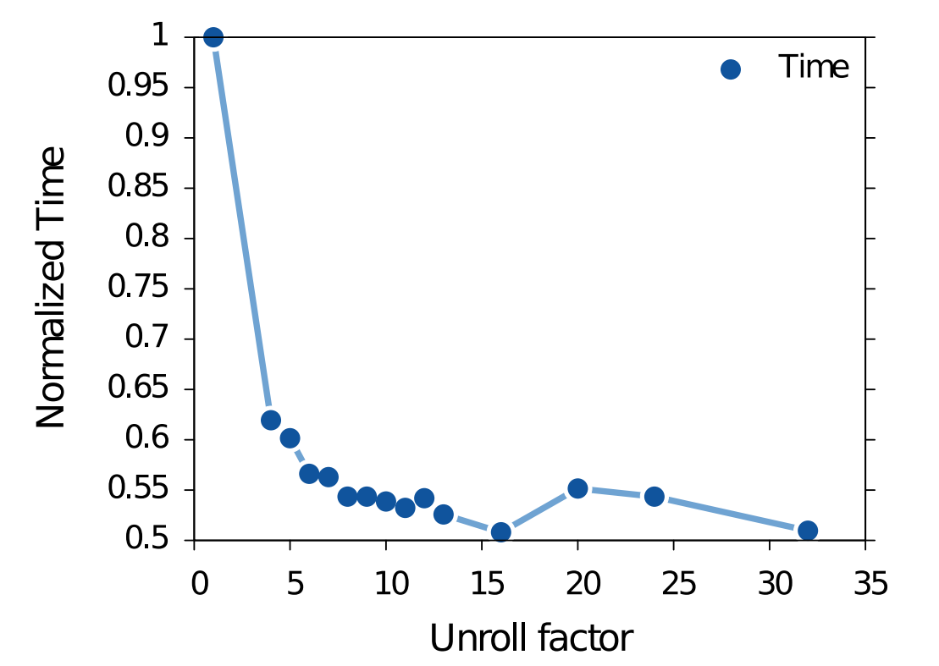
Q-grid (in inverse space) is decomposed into ***tiles***, each assigned to a different processor. It is further divided into ***hyperblocks*** along with triangles, each computed one at a time on the device. This helps to hide host-device transfer latency. A hyperblock is decomposed into **thread blocks**. During computation within a thread block, input triangles data is loaded into the shared memory, and all threads access each triangle to compute the form-factors.

Multiple such computations may be executed in a single run, each with different experimental configuration. Each configuration is executed in parallel, and this becomes the highest parallelism level in the hierarchy.

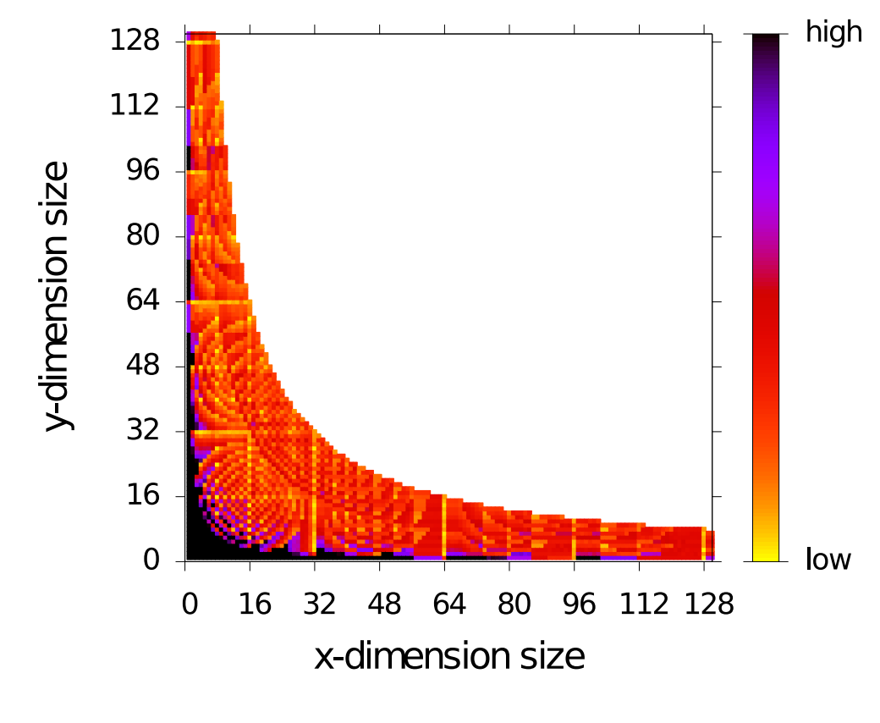
GPU Optimizations

We use the CUDA programming model to off-load the computational kernel on the GPUs. To exploit the processing power of these GPUs, our Kepler specific optimizations include the following:

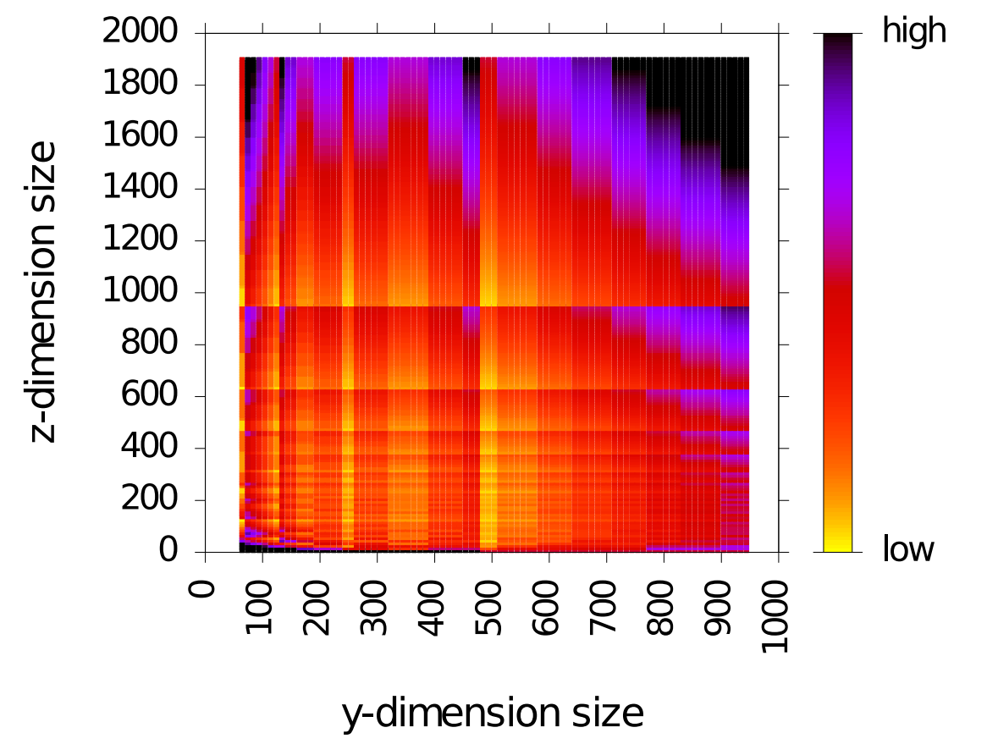
- ***k*-buffering** to hide CPU-GPU memory latencies.
- **Shared memory usage** for input data reuse, memory coalescing and memory traffic reduction within the GPU.
- **Padding of triangle definitions** to achieve optimal memory alignment, totaling 32 bytes per triangle.
- Innermost triangles' **loop unrolling** (see graph.)
- Use of **instrinsics** and **hand-coded PTX code** to optimize at instruction level.
- Autotuning of decomposition parameters, hyperblock size, and CUDA thread-block size (see figures.)



Performance trend with respect to the innermost loop unroll factor. We select a factor of 16.



Trend of kernel execution times over the search space of *x* and *y* dimensions of the CUDA thread-block size. Note the optimal regions in yellow.



Trend of kernel execution times over the search space of the y and z dimensions of decomposed hyperblocks.

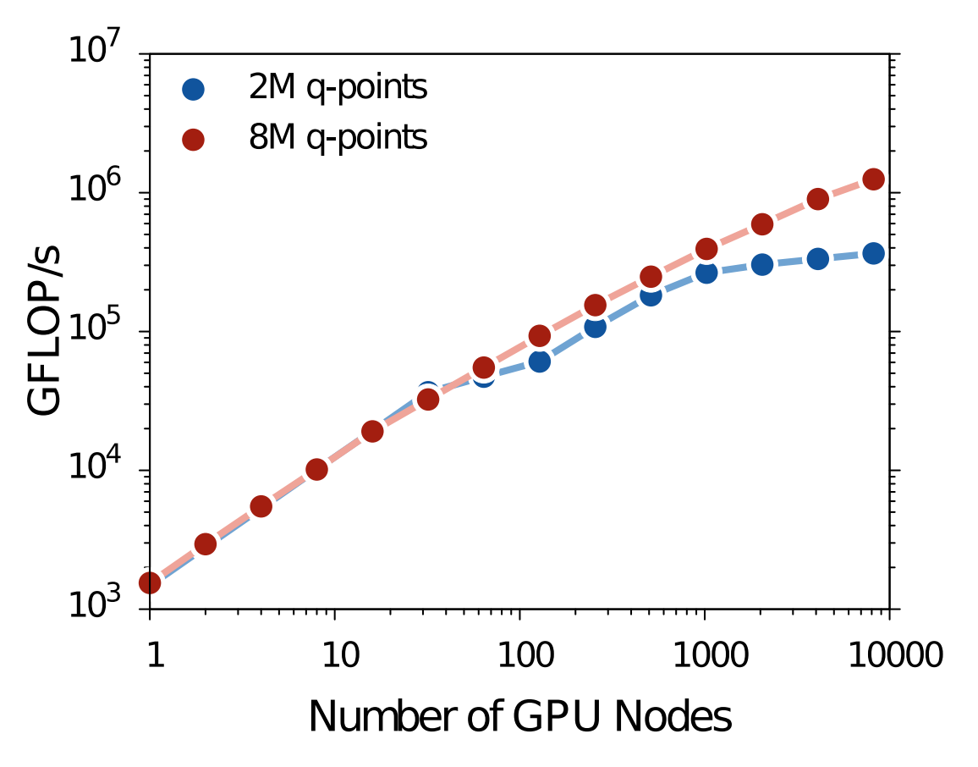
Performance on Single and Multiple GPUs

Performance on single K20X and K40m GPUs, compared with dual AMD Magny Cours (24 cores) and dual Intel Sandy Bridge (16 core) processors:

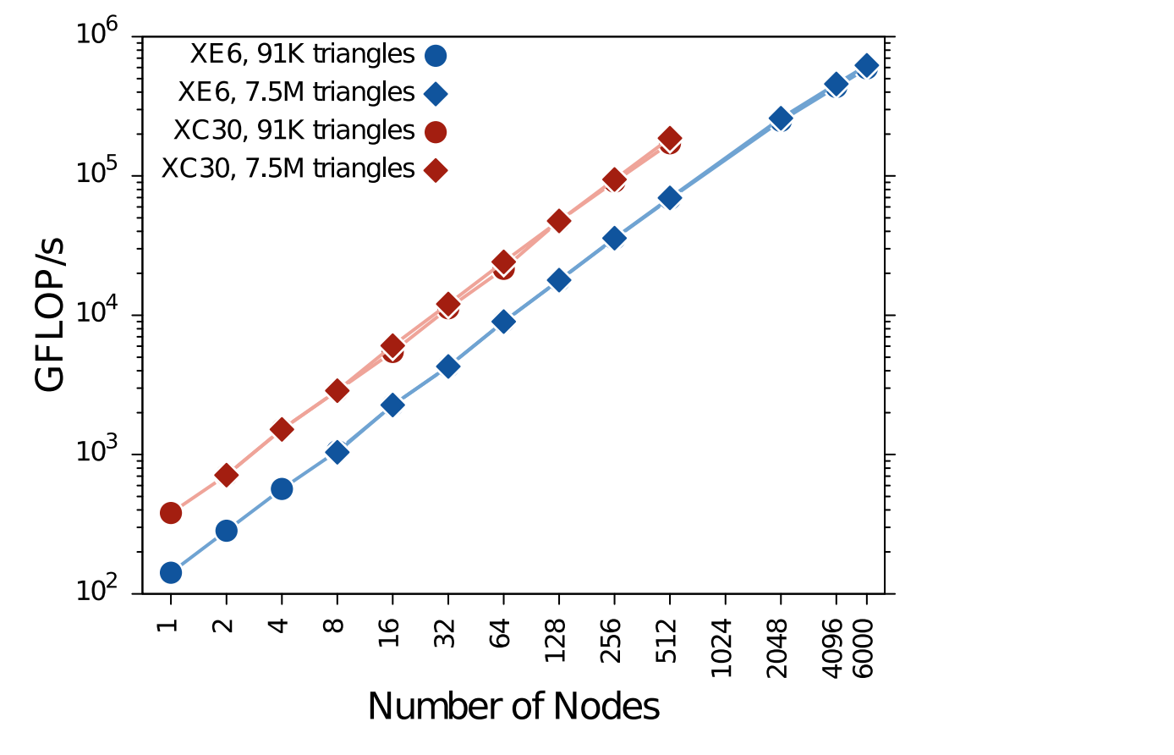
		Performance in GFlop/s			
No. Triangles	**No. *q*-points**	**K20X**	**K40m**	**AMD Magny Cours**	**Intel Sandy Bridge**
6600	2M	2961	3206	142	378
6600	8M	3002	3247	142	378
91753	2M	2955	3192	142	379
91753	8M	2998	3241	142	380

Our optimized kernel achieves **76% of the theoretical peak** performance on both Nvidia K40m and K20X GPUs, **35% of the theoretical peak** on AMD Magny Cours, and **57% of the peak** on Intel Sandy Bridge CPUs.

Performance of HipGISAXS on the **Cray Xk7 Titan supercomputer** (with a K20X GPU per node) compared with **Cray XE6 Hopper supercomputer** (with dual AMD Magny Cours, 24 cores, per node) and **Cray XC30 Edison supercomputer** (with dual Intel Sandy Bridge, 16 cores, per node) are shown below.



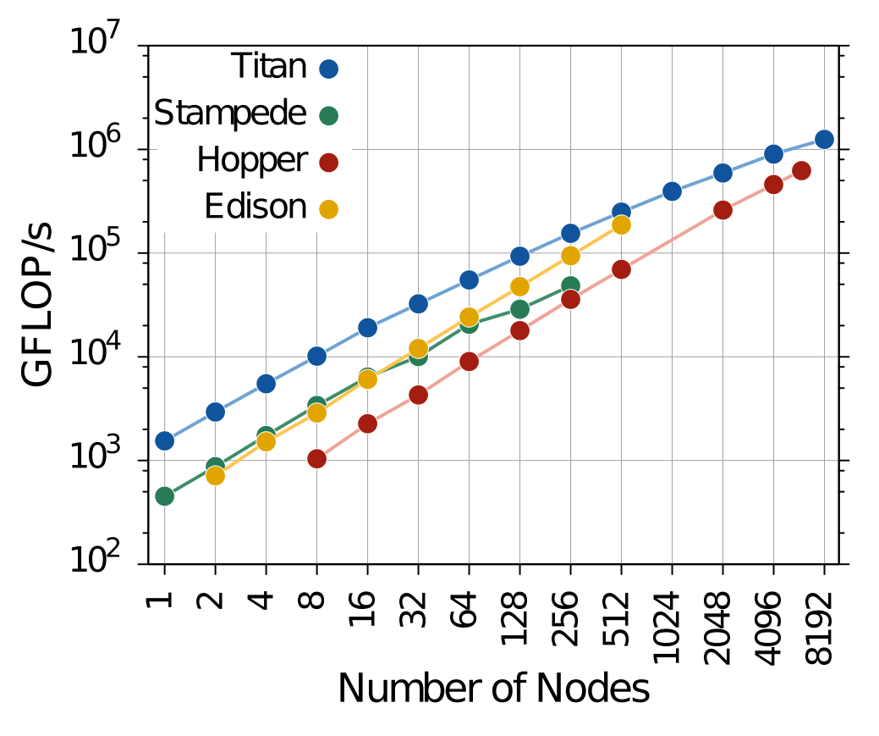
Strong scaling with respect to the number of nodes on Titan for input with 7.5M triangles.



Strong scaling of HipGISAXS on Cray XE6 (Hopper) and Cray XC30 (Edison) with respect to number of nodes.

Conclusions

HipGISAXS is enabling fast X-ray scattering simulations and data analysis using parallel computing. GPU powered systems prove to be ideal platform for computations in HipGISAXS. It is able to achieve **2 Petaflops using 8192 nodes on Titan** (Cray Xk7). For comparison, it achieves **0.63 Petaflops using 144000 cores on Hopper** (Cray XE6), **0.2 Petaflops using 8192 cores on Edison** (Cray XC30), and **0.1 Petaflops using 1024 nodes on Stampede** (Intel MIC cluster).
We also note that GPUs proved to be the most energy efficient as well with a performance per watt of **9 GFlops/W** on Nvidia K20X GPU, compared to just **1.3 GFlops/W** on AMD MagnyCours, **3.3 GFlops/W** on Intel Sandy Bridge processors, and **2 GFlops/W** on an Intel MIC co-processor.



A comparison of strong scaling of HipGISAXS on Titan with other systems, with multi-core processors (Hopper, Edison), and MIC coprocessor (Stampede).

Acknowledgements

This work was supported by the Director, Office of Science, U.S. Department of Energy (DoE) under contract no. DE-AC02-05CH11231. This research used resources of the National Energy Research Scientic Computing Center (NERSC), which is supported by the Office of Science, U.S. DoE, under contract no. DE-AC02-05CH11231, and of the Oak Ridge Leadership Computing Facility (OLCF), which is supported by the Office of Science, U.S. DoE, under contract no. DE-AC05-00OR22725. The authors also acknowledge the Texas Advanced Computing Center (TACC) for providing HPC resources that have contributed towards the reported results.

References

1. S Chourou, A Sarje, X Li, E Chan, and A Hexemer. *HipGISAXS:* A High Performance Computing Code for Simulating Grazing Incidence X-Ray Scattering Data. *Journal of Applied Crystallography*, vol 46, no 6, pp 1781-1795, 2013.
2. A Sarje, X Li, S Chourou, E Chan, and A Hexemer. Massively Parallel X-ray Scattering Simulations. In *International Conference for High Performance Computing, Networking, Storage and Analysis (Supercomputing)*, pp 46:1-46:11, 2012.
3. A Sarje, X Li, and A Hexemer. Tuning HipGISAXS on Multi and Many Core Supercomputers. In *Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems at Supercomputing*, 2013.