Predictive Performance Tuning of OpenACC Accelerated Applications

Shahzeb Siddiqui[1], Saber Feki[2]
Division of Computer, Electrical and Mathematical Sciences and Engineering[1], KAUST Supercomputing Laboratory[2]
King Abdullah University of Science and Technology, Kingdom of Saudi Arabia

Abstract

Graphics Processing Units (GPUs) are gradually becoming mainstream in supercomputing as their capabilities to significantly accelerate a large spectrum of scientific applications have been clearly identified and proven. Moreover, with the introduction of high level programming models such as OpenACC [1] and OpenMP 4.0 [2], these devices are becoming more accessible and practical to use by a larger scientific community. However, performance optimization of OpenACC accelerated applications usually requires an in-depth knowledge of the hardware and software specifications. We suggest a prediction-based performance tuning mechanism [3] to quickly tune OpenACC parameters for a given application to dynamically adapt to the execution environment on a given system. This approach is applied to a finite difference kernel to tune the OpenACC gang and vector clauses for mapping the compute kernels into the underlying accelerator architecture. Our experiments show a significant performance improvement against the default compiler parameters and a faster tuning by an order of magnitude compared to the brute force search tuning.

Application

We used in our experiments the isotropic finite difference kernel which constitutes the building block for the Reverse Time Migration (RTM) application and Full Waveform Inversion (FWI), extensively used by the oil and gas exploration industry for velocity model building and seismic imaging of the sub-surface. The Reverse Time Migration application uses forward modeling and backward migration using a finite difference kernel that solves the acoustic wave equation.

$$\frac{1}{c^2}\frac{\partial^2 P}{\partial t^2} = \frac{\partial^2 P}{\partial x^2} + \frac{\partial^2 P}{\partial y^2} + \frac{\partial^2 P}{\partial z^2}$$

where c is the velocity of the propagated wave and P is the pressure wavefield.

The 3D finite difference stencil scheme is 8th order in space and 2nd order in time. We plan in the future to extend this study to different orders in space to explore the impact of computation intensity on the parameter choices of the auto-tuner.

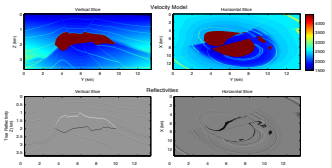

Fig. 2: 3D SEG/EAGE salt velocity model and corresponding reflectivity models obtained by using RTM [4]

Predictive Performance Tuning Methodology

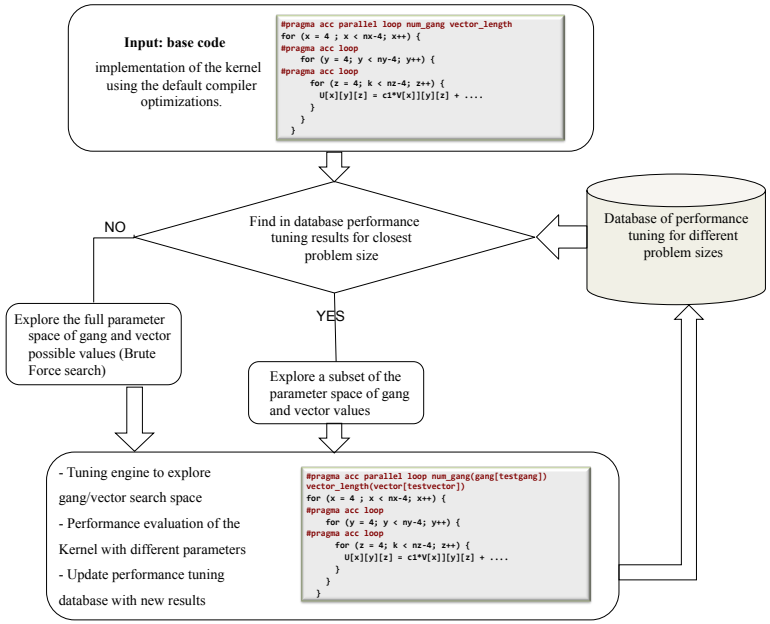

Fig. 1: Schematic of the automatic performance tuning procedure using brute force search and predictive tuning

Experimental Results

We present here the performance results of applying the suggested tuning methodology on the RTM isotropic modeling kernel executing on a K20c NVIDIA GPU. As shown in Figure 3, the tuning procedure using either the brute force search or the predictive method resulted in a better performance than the compiler default tuning. A performance increase of up to 80% is recorded against the performance of the compiler tuned code. Figure 4 shows the required time for tuning a given problem size with the brute force and the predictive tuning methods. Our experiment shows that the tuning time is reduced by a factor of 18X to 52X by using the predictive tuning approach; at the same time, the enhanced application performance is comparable to the tuned version using the brute force search approach.

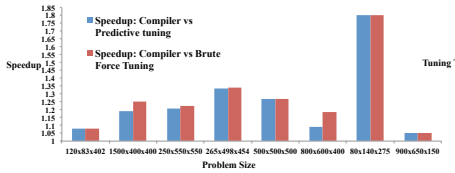

Fig. 3: Performance speedup analysis using the different tuning methodologies against the compiler tuning

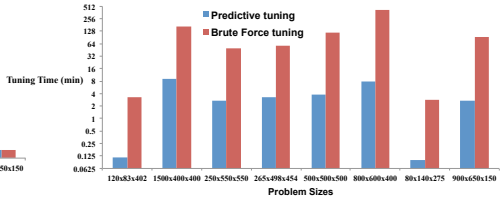

Fig 4: Tuning time for brute force versus predictive tuning

Conclusion

A prediction-based approach to perform runtime performance tuning for OpenACC accelerated applications is suggested. The performance results obtained by our tuning methodology on a finite difference kernel showed a respectable performance gain against the compiler tuned code. Furthermore, the time needed for tuning is reduced by an order of magnitude compared to the naive brute force search technique. This motivates future development of this methodology and its application to a larger spectrum of scientific applications.

Future Work

- Automation of the tuning process and its application during the execution of the application.
- Explore different computational intensities by increasing, decreasing the order in space of the finite difference kernel.
- Experiment on different GPU architectures and generation to track the performance of a kernel across architectures and the resulting tuned gang/vector values.

References

[1] OpenACC standard *www.**openacc-standard**.org/*

[2] OpenMP 4.0 specification
www.openmp.org/mp-documents/OpenMP4.0.0.pdf

[3] Saber Feki, Edgar Gabriel. ***A Historic Knowledge Based Approach for Dynamic Optimization*** , in proceedings of the International Conference on Parallel Computing, 2009, P. 389 - 396

[4] Yunsong Huang, Gerard T. Schuster. ***Multisource least-squares migration of marine streamer and land data with frequency-division encoding***, Geophysical Prospecting, 2012, Vol. 60, P. 663–680

Acknowledgments

- Michael Young (KAUST IT) for system support
- Tareq Males (KAUST CEMSE) for providing sequential C code
- NVIDIA for donating the hardware