Decision: Reject

Submission: Dwarfs on Accelerators: Enhancing OpenCL Benchmarking for Heterogeneous Computing Architectures
Contributors: Johnston, Milthorpe

Key for the below column headings: show

Summary of Reviews of pap309s1: Dwarfs on Accelerators: Enhancing OpenCL Benchmarking for Heterogeneous Computing Architectures

Reviewer	Relevance	Soundness	Importance	Originality	Presentation	Action	Conf
Reviewer 1	MODERATE (Matches one topic in list) (4)	MODERATE (Multiple minor errors) (4)	MODERATE (Worth reading) (4)	LOW (Little new beyond already published work) (3)	HIGH (Almost no grammatical errors) (5)	WEAK REJECT (Could be convinced to accept) (3)	MODERATE (Maybe someone saw something that I missed) (4)
Reviewer 2	HIGH (Matches multiple topics in list) (5)	HIGH (Few minor errors) (5)	LOW (Someone may find it useful) (3)	MODERATE (New results building on previous publications) (4)	MODERATE (Well organized but some grammatical errors or need to improve organization in minor ways) (4)	WEAK REJECT (Could be convinced to accept) (3)	MODERATE (Maybe someone saw something that I missed) (4)
Reviewer 3	MODERATE (Matches one topic in list) (4)	LOW (At least one significant error) (3)	MODERATE (Worth reading) (4)	MODERATE (New results building on previous publications) (4)	LOW (Rife with grammatical errors or difficult to decipher graphs) (3)	WEAK ACCEPT (Could be convinced to reject) (4)	MODERATE (Maybe someone saw something that I missed) (4)
Reviewer 4	HIGH (Matches multiple topics in list) (5)	MODERATE (Multiple minor errors) (4)	MODERATE (Worth reading) (4)	MODERATE (New results building on previous publications) (4)	LOW (Rife with grammatical errors or difficult to decipher graphs) (3)	WEAK ACCEPT (Could be convinced to reject) (4)	HIGH (I have little doubt that I am right) (5)
Averages:	**4.5**	**4.0**	**3.8**	**3.8**	**3.8**	**3.5**	**4.3**

Commitee Comments and Notes jump

Review of pap309s1 by Reviewer 1

top

Review

The paper describes the Extension of the OpenDwarfs benchmarking suite with variable problem sizes suitable to show the performance characteristics for different layers of the memory hierarchy in modern CPU and Accelerator architectures. Sampling of the performance has been improved through the introduction of statistical tooling, including sufficient numbers of samples taken to ensure a beta of 0.8. The resulting benchmark suite is then used to compare execution times and, with a reduced range of devices, energy use. Evaluated hardware includes desktop GPUs manufactured by NVidia and AMD, NVidia Kepler HPC GPUs, Intel CPUs (Server and Desktop) and an Intel Xeon Phi Coprocessor (although with severe restrictions). The final goal of the contribution is to serve as a basis to evaluate algorithmic workload in regards to their suitability for different types of hardware.

While the presented establishment of benchmarks as a base for further research is an important goal, the presented contribution is mainly incremental and I'm not fully convinced that it deserves a separate paper. What is not addressed in detail is the question of the quality (and purpose) of the underlying Open Dwarfs benchmarks. While OpenCL aims to provide portability, it is not addressed to what extent performance portability is preserved. As the authors explicitely consider the effect of problem size and memory hierarchy, it would be interesting to know whether the benchmarks include respectie optimisations. The authors state that they removed device-specific tuning implemented in prior versions of the suite for their version. However, to fully utilize a GPU, one may need to use different programming style compared to MIC devices, FPGAs or CPUs. Here, the same code is executed across all devices. Whether the respective kernels in that way achieve a relevant fraction of the performance that is achievable on the several devices, is not addressed.

The paper presents the results of executing the benchmark suite on the aforementioned range of devices. Reported values are kernel execution time, and, for a narrow subset of the hardware, kernel execution energy use, for different problem sizes. It does not, however report other quantities such as, e.g. (floating) point operations per second or memory bandwidth (where applicable).

The claimed inclusion of MIC architectures is curtailed by lack of support for OpenCL, leading to its exclusion from all but one of the results. Energy is only measured on a single CPU and a single GPU.

Technical aspects:
The graphs in the paper are difficult to read, especially in black and white printout. Unifying the styles may be a good idea, as well as possibly the inclusion of vertical lines in the execution time graphs, in order to enable the reader to easily connect the results with the architecture they were executed on. Consider using different color hues to make it easier to keep them apart.

Small issues:
• Last sentence intro: including GPUs, ...: use consisting of, include implies existence of something not mentioned.
• Software setup: included in 16.1.1: What is referred to? driver/SDK?
• Why are there differences in the software base (e.g. different GCCs)?
• Related Work: Maybe include recent work on OpenDwarfs?
• 4.4.5 Some part too detailed. One does not need info that -p 1 -d 0 -t 0 is a CPU on that particular host. Information such as this should be left to the code documentation published with the library itself.

Review of pap309s1 by Reviewer 2

top

Review

This paper describes several refinements to the OpenDwarfs benchmark suite and then proceeds to benchmark each kernel on a range of hardware.

Overall this paper is a good read, the text has a good pace and the descriptions are good. The presentation of the graphs/box plots however is challenging (as is usual for such a broad collection of results). The reviewer can't help wondering if some kind of table might be better given the considerable space allocated to the diagrams. At times, it is difficult to see which hardware item is performing the best because of such a similar clumping or poor scaling of the axes. It would also aid in future reproducibility.

The description that Intel has removed support for "AVX2 vectorization (using the -xMIC-AVX512 flag)" is confusing -- AVX2 is 256-bit wide, are both AVX512 and AVX2 removed, or just AVX512, or just AVX2 (would be hard to remove AVX2 without AVX512 as well)? This is important for the KNL results because a significant aspect to the hardware performance is no longer available. Is this even a genuine comparison of *hardware* at this stage?

This raises an interesting question -- whether OpenCL really is gaining traction in the HPC community, as described in the front matter. It seems that there is sluggish vendor support for the standard and a growing preference for other directive solutions (OpenMP, OpenACC). This review appreciates the subjectivity of this statement, but it raises an interesting question as to whether the performance seen is really a testament to the hardware performance or the level of optimization applied by the vendor to their SDK. Comparison with equivalent benchmarks in other forms (e.g. OpenMP) may be interesting to help remove this doubt.

Overall, the paper could do with something extra in the form of the results to remove the feeling that this is really a catalogue of performance results -- considerable parts of the paper are devoted to descriptions of the benchmarks, but this might be better used to describe/compare more of the results obtained from LibSciBench instead. It would make for a more complex level of analysis that showed good

insight.

Summary - reasonable paper but deeper insight would really add to the value of the article within the conference.

Review of pap309s1 by Reviewer 3

top

Review

This paper presents an extension of the OpenDwarfs OpenCL benchmark suite. Specifically, the authors have integrated a performance measurement tool called LibSciBench to record timing events in addition to the hardware events collected via PAPI, and they have modified the applications to insert calls to this tool. In conjunction with existing tools, LibSciBench also enables the reporting of energy measurements. The authors also have modified the input problem sizes to track increasing cache sizes in the memory hierarchy, and they have added new applications and fixed existing applications with respect to correctness. The paper performs a cursory analysis of the applications on 15 different systems to show portability.

I found this paper somewhat difficult to evaluate. Having new and/or better benchmarks that are portable over a wide range of systems and that can provide high quality measurements sounds great. The problem I had with the paper is that it doesn't do a good job pointing out which functionality or results are now new given their enhancements as compared to the non-enhanced OpenDwarf OpenCL benchmark suite. I'd really like to know what new results/data collection are possible given their enhancements, but that is not clearly discussed in the results section although the results section says the purpose of the evaluation is to "demonstrate the benefits of the extensions". I want to know concretely whether the results in Figure 2 and 3 were possible before these extensions; if not, what was possible previously? My understanding is that the results in Figure 5 were not obtainable before these modifications.

I also had trouble sifting through the discussion of the results in Section 5.1. The presentation of the results seemed very adhoc, with random bits of information from the graphs being pointed out instead of a systematic discussion of the graphs. I frequently had trouble determining how I was expected to see an observation in the graphs because the text didn't point out what specific data points to be examining when discussing a result. For example, the text says k-means CPU execution times were comparable to GPU. How is this to be observed in the graphs? I definitely see this to be true in tiny, but there is a good deal of variation in the other 3 graphs; is that the point? Similarly, the discussion of Spectal Methods dwarfs baffled me. How is the reader to see that the execution times match the higher memory latency of the L3 cache in the graph? Or, how does the reader see in the graph that the CPU devices frequently access main memory while the GPUs' wind up with a lower memory access latency? These may be explanations for the

differences in the execution times, but these observations about memory latency are not *seen* in the graphs. You would need supporting graphs that presented data about average memory access times to support this explanation.

In a related concern, what do the numbers in reference to systems mean in the discussion of older vs. new GPUs? The systems are not numbered in the table introducing them or on the x-axes.

Review of pap309s1 by Reviewer 4

top

Review

This paper takes an existing Open Dwarfs benchmark suite and enhances the suite by adding new benchmarks, improving (replacing in the case of the FFT) some of the benchmarks and adding additional problem sizes. The work to improve this suite is solid.

To gather up performance data, they instrumented the benchmark suite with LibSciBench and presented the performance study. Unfortunately, the performance results and analysis felt constrained by the 10 pages. The graphs were hard to read (font tiny and graphs small), and in some of the analysis, there was not adequate explanation.

I liked the Table 1 for the hardware, and wished that the software configuration be given in a table as well. It would have made it easier to understand.

I did not understand why the problem sizes were based on the memory hierarchy of the Skylake CPU --- the explanation given was inadequate. It felt like there was a bias there.

I disagree with the premise in the introduction that justifies looking at OpenCL. The emphasis is on the portability of the codes, and then using that, there is justification for the performance study. The problem is that even with OpenCL, one can write applications that perform well on one architecture, can port to another, but to perform as well on another architecture, a different algorithm or optimization strategy may need to be used. So while functionally portable, there can be architectural bias to prevent performance portability. I did not feel that this paper acknowledge this.

Committee Comments for Authors

top

None