REDUCING JET AIRCRAFT NOISE BY HARNESSING THE HETEROGENEOUS XK NODES

Allocation: Blue Waters Professor/120 Knh
PI: Daniel J. Bodony[1]
Collaborators: Simon Garcia De Gonzalo[1], Wen-mei Hwu[1]

[1]University of Illinois at Urbana-Champaign

EXECUTIVE SUMMARY

Reducing the noise emitted by commercial and military aircraft to alleviate their adverse environmental and health impacts requires computational resources that exceed current petascale computers. Future exascale computers are expected to have architectures where most of the floating-point computing capacity is handled by accelerators; however, no single architecture is expected to be dominant. We focus on the programming models and runtime systems required to support computational science on future exascale computers.

INTRODUCTION

Gas turbine-powered commercial and military aircraft create intense noise at take-off that negatively impacts near-airport communities and creates a health hazard for carrier-borne personnel. Predicting and, ultimately, reducing jet noise requires immense computing resources because of the large range of temporal and spatial scales present in the hot, turbulent exhaust gases. Future exascale computers are expected to open new opportunities in jet noise reduction, but their architectures will differ from current leadership-class computers to improve power-per-FLOP performance. In particular, the bulk of the floating-point operations will be performed on accelerators that may be highly simplified computing elements similar to modern-day GPUs connected to task-managing CPUs, host-less accelerators similar to Intel Xeon Phis, or single-chip heterogeneous system architectures (HSA) in which the GPU and CPU sit on the same die. Without architecture clarity, computational science codes must adapt and be sufficiently flexible to run on all three kinds of systems. We use the XK nodes on Blue Waters as a stand-in example of one type of a future exascale computer to test a new programming model and runtime system capable of using all three types of future computers. We focus on the specific needs of typical computational fluid dynamics algorithms with low FLOP-to-load ratios.

METHODS & RESULTS

For low FLOP-to-load ratio algorithms the biggest challenge to achieving performance is the bandwidth between the memory system and the computing elements. The three classes of proposed exascale architectures have very different memory-compute layouts, so our work has focused on isolating the expression of the algorithm from the underlying hardware description and permitting on-the-fly runtime optimization (see Fig. 1). The *H-MxPA* [1] cross-compilation and runtime tool uses C++AMP [2] to express the algorithm in a platform-independent manner, a compilation back-end supported by LLVM [3], and a dynamic runtime that partitions work between the available CPU (if present) and computing element(s) (if present) by closely monitoring performance. In addition, on modern CPU-GPU nodes like Blue Waters' XK7s, memory coherency must also be ensured between the host and device by the runtime.

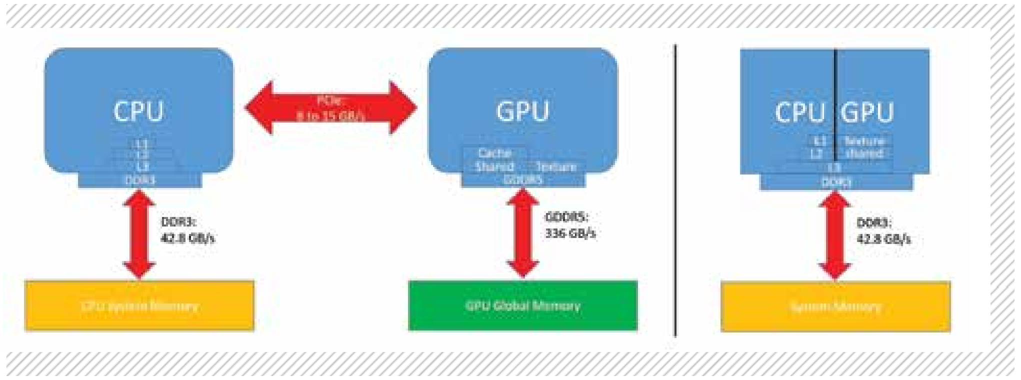

FIGURE 1: Different memory architectures and bandwidths for legacy CPU-accelerator systems (left) and cache-coherent HSA systems (right).

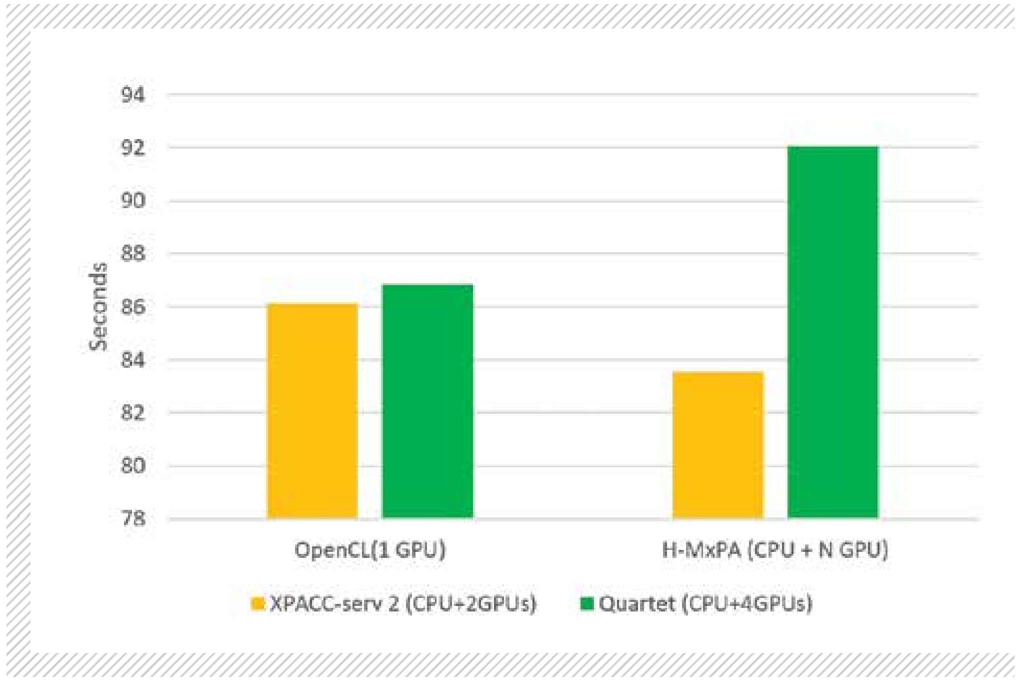

FIGURE 2: Performance of RK4 update step on two legacy CPU-GPU systems (smaller is better). XPACC-serv-2 has one Intel E5-2680v2 connected to two NVIDIA K40m GPUs, while Quartet has one Intel i7-3820 connected to four NVIDIA K20c GPUs.

Applying *H-MxPA* to the update step in the standard fourth order Runge-Kutta (RK4) algorithm, which contains six memory operations (five loads and one store) and six FLOPs (four additions and two multiplies), to legacy CPU-GPU nodes showed the loss of performance due to the memory bandwidth and cache coherency requirements (Fig. 2). For these runs the work distribution was hand-tuned. These performance data show explicitly the penalties incurred on CPU-GPU systems where host and device memories are connected via a PCI bus and speak to the performance improvement possible when the two elements are co-located on a single die, as on an HSA-type machine. It is noteworthy that the pre-exascale machines to be built by IBM-NVIDIA for the Department of Energy use NV-LINK and hardware-supported cache coherency [4].

WHY BLUE WATERS

The XK nodes of Blue Waters represent an early vision of the future exascale computers that will be realized in the 2020-2025 timeframe. The architecture and software stack of Blue Waters' XK nodes provide relevant representative platforms on which to develop software technologies capable of using heterogeneous computers. The size of the Blue Waters XK partition and its connection to the XE partition provide a unique opportunity to build and test computational science algorithms that can handle intra- and inter-node inhomogeneity at leadership-class scale.

NEXT GENERATION WORK

The cross-compilation and runtime development being performed on Blue Waters will become the foundation on which our next-generation computational fluid dynamics (CFD) code will be built. With the flexibility provided by *H-MxPA* we will use the future Track-1 system to demonstrate the efficiency with which computational science can be performed on the pre-exascale computer as well as advance the science of CFD codes used by the engineering and science communities.