

# POSTER: Scheduling HPC Workloads on Heterogeneous-ISA Architectures

Mohamed L. Karaoui, Anthony Carno, Rob Lyerly, Sang-Hoon Kim, Pierre Olivier, Changwoo Min, Binoy Ravindran  
{karaoui,acarno,rlyerly,sanghoon,polivier,changwoo,binoy}@vt.edu  
Virginia Tech

## Abstract

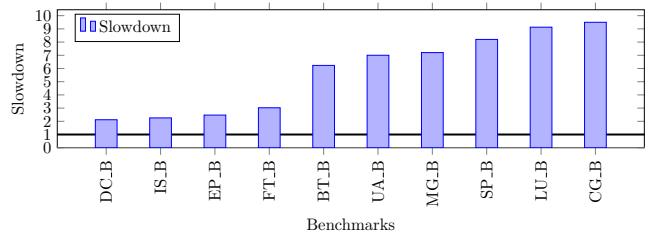
In this paper, we investigate the effectiveness of multiprocessor architectures with ISA-different cores for executing HPC workloads. Our envisioned design point in the heterogeneous architecture space is one with multiple cache-coherency domains, with each domain hosting cores of a different ISA and no coherency between domains. We prototype such an architecture using an Intel Xeon x86-64 server and a Cavium ThunderX ARMv8 server, interconnected using a high-speed network fabric. We design, implement, and evaluate policies for scheduling HPC applications with the goal of maximizing workload makespan. Our results reveal that such an architecture is most effective for workloads that exhibit diverse execution times on ISA-different CPUs, with gains exceeding 60% over ISA-homogeneous architectures. Furthermore, cross-ISA execution migration can yield gains up to 38%.

**CCS Concepts** • Computer systems organization → Heterogeneous (hybrid) systems;

## 1 Introduction

The “end of Moore’s Law” has forced chip vendors to design alternate architectures to advance performance and energy efficiency boundaries. Such architectures have included multicore and manycore chips that exploit hardware parallelism; CPUs with heterogeneous micro-architectural properties, partially overlapping instruction-set architectures (ISAs), and various forms of accelerators and programmable hardware that exploit heterogeneity. While commercial heterogeneous architectures largely use a single ISA (e.g., x86 or ARM), the academic research community has explored alternate points in the design space, including ISA heterogeneity. Exploration in this particular design space includes many

forms – shared-memory chip multiprocessors [2, 7], multiprocessors with multiple cache-coherent domains (and no coherence between domains) [3], and composite-ISA cores [6].



**Figure 1.** Execution-time slowdown of NPB benchmarks on ThunderX compared to Xeon, using only one core.

Heterogeneous architectures generally benefit applications that exhibit diversity (e.g., CPU/memory intensity, SIMD behaviors) [4]. To understand this in a multi-ISA setting, we measured the execution times of the NPB benchmark suite [1] on two ISA-different machines: an Intel Xeon machine (x86-64, 8 cores, 2.3 GHz) and a Cavium ThunderX machine (ARMv8, 48 cores, 2.0 GHz). Figure 1 shows the slowdown of each NPB application on the ThunderX with respect to its Xeon execution. Even though the Xeon has better micro-architectural features (i.e., few “beefy” cores) yielding better single-threaded performance, the ThunderX has a higher core count (6 times more “wimpy” cores). This will likely benefit when executing workloads with inherent parallelism such as one that is composed of a single multi-threaded application, multiple single-threaded applications that are multiplexed together, or some combination thereof.

Figure 1 therefore raises interesting questions: for what workloads can a heterogeneous-ISA architecture yield better makespan than a homogeneous-ISA one? What scheduling policies are effective for optimizing makespan on heterogeneous ISAs? When would it be effective to migrate applications across ISA-different cores to exploit idle cores?

We assume single-threaded CPU/memory-bound HPC applications with single phases.

## 2 Scheduling Heuristics

Let  $R$  be the ratio of the number of cores between the two processors. For the Xeon/ThunderX setup,  $R$  is  $6 = \frac{48}{8}$ , which means that the high core-count processor (i.e., the ThunderX)

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

PPoPP '19, February 16–20, 2019, Washington, DC, United States

© 2019 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-6225-2/19/02.

<https://doi.org/10.1145/3293883.3295717>

can execute  $R$  more applications concurrently. However, the high number of cores does not always translate into more number of applications executed – that depends on the type of application. Indeed, different applications have different *slowdowns* on the low core-count processor compared to the high core-count processor (example in Figure 1).

When the slowdown of the workload is less than  $R$ , the low core-count processor performs better (Rule 1). When the workload slowdown is higher than  $R$ , the high core-count processor performs best (Rule 2). These two rules apply for workloads with one type of job: “high” or “low” slowdown.

However, when the workload is composed of mixed jobs, the heterogeneous system may outperform the homogeneous one (Rule 3). The performance depends on the ratio between the number of jobs in the workload.

Since the problem of mapping applications to cores is NP-complete in general, we consider heuristics. For the homogeneous system, we use the well-known Longest Processing Task first (LPT) algorithm. This algorithm minimizes tardy jobs by scheduling the longest tasks first.

For the heterogeneous system, we consider a simple heuristic that maps applications to cores according to their slowdown: high slowdown applications are placed on big cores; low slowdown ones on small cores. We augment this with a simple cross-ISA execution migration policy: when there is an idle big core with an empty job queue, applications are migrated from the small to big core. Cross-ISA migration is accomplished using Popcorn Linux [2]: for each application, the Popcorn compiler generates multiple binaries, one per ISA, wherein all symbols have the same addresses and sizes. During execution, when a migration decision is made, the Popcorn run-time transforms the application’s state (registers and stack) between ISAs, and the Popcorn OS migrates memory pages between ISAs, lazily and on-demand.

### 3 Experimental Results

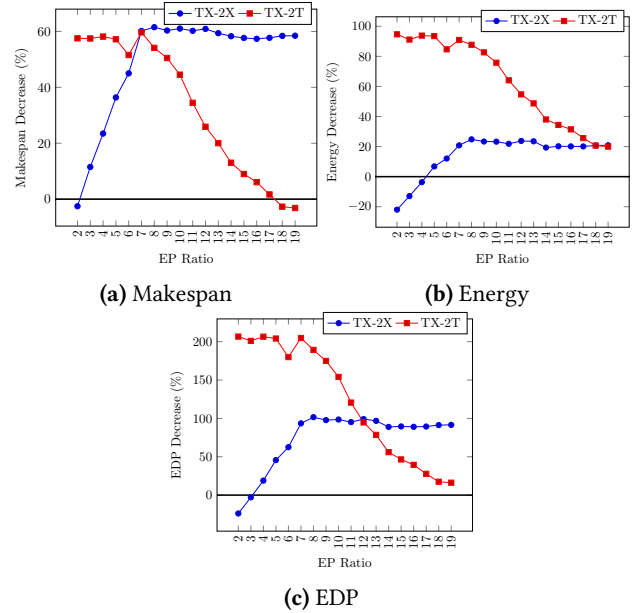
Our evaluation used the NPB suite. We used one heterogeneous-ISA system and two homogeneous-ISA systems, built using the Xeon and ThunderX machines described in Section 1. We computed application slowdown by profiling them offline.

To validate Rules 1 and 2, we execute a job queue of single jobs from the NPB suite: CG, MG, and EP. EP is the only job that performs better on the two ThunderX system. This is expected since only EP has a low slowdown. To validate Rule 3, we execute a job queue composed of two applications: EP-MG and EP-CG.

Figure 2a shows the makespan decrease of the EP-CG workload compared to the homogeneous systems. The heterogeneous system outperforms them both when the ratio varies between 7 EP\_B : 1 CG\_B and 13 EP\_B : 1 CG\_B.

Figures 2b and 2c compare the systems in terms of energy consumption and Energy Delay Product (EDP).

To analyze the effect of cross-ISA migration on the het-



**Figure 2.** Percentage decrease of makespan, energy, and EDP of the heterogeneous-ISA system compared to the two homogeneous-ISA systems. The workload is composed of NPB’s EP and CG applications, with a queue size of 1024 jobs, with varying ratios of EP to CG on the x-axis.

erogeneous system, we executed all experiments with and without migration and compared the performance. At best, the performance is improved by 38%. At worst, the performance is degraded by only 6%.

Detailed results are available in [5].

### 4 Acknowledgments

This work is supported by ONR under grant N00014-16-1-2711 and NAVSEA/NEEC under grant N00174-16-C-0018.

### References

- [1] D. H. Bailey, E. Barszcz, et al. The NAS parallel benchmarks. *The International Journal of Supercomputing Applications*, 5(3):63–73, 1991.
- [2] A. Barbalace, R. Lyerly, C. Jelesnianski, A. Carno, H.-R. Chuang, V. Legout, and B. Ravindran. Breaking the boundaries in heterogeneous-ISA datacenters. *ASPLOS*, 2017.
- [3] F. X. Lin, Z. Wang, and L. Zhong. K2: A mobile operating system for heterogeneous coherence domains. *SIGPLAN Not.*, 49(4), Feb. 2014.
- [4] S. Mittal. A survey of techniques for architecting and managing asymmetric multicore processors. *ACM Computing Surveys (CSUR)*, 2016.
- [5] Mohamed L. Karaoui, Anthony Carno, Rob Lyerly, Sang-Hoon Kim, Pierre Olivier, Changwoo Min, Binoy Ravindran. Scheduling HPC Workloads on Heterogeneous-ISA Architectures. Technical report, Virginia Tech, 2019. URL <http://www.popcornlinux.org/images/publications/popcorn-scheduling-poster-ppopp2019.pdf>.
- [6] A. Venkat. *Breaking the ISA Barrier in Modern Computing*. PhD thesis, UC San Diego, 2018.
- [7] A. Venkat and D. M. Tullsen. Harnessing ISA diversity: Design of a heterogeneous-ISA chip multiprocessor. In *ISCA*, 2014.