Test Plan for Demonstrating Importance of Multi-Accel Apps for ARA Evaluation

Riley Wood

March 17, 2016

Summary

Take the same application with three different implementations: unaccelerated, meaning all computation is done in the primary core; single-accelerated, meaning the application uses one accelerator to assist with computation; and multi-accelerated, meaning the application uses many accelerators to assist with computation. The first serves as a control, and I will make a comparison between the single- and multi-accelerated applications. I will run these tests on a computer that uses the OS and drivers to manage accelerators. This is the standard, slow management scheme that experimental accelerator-rich architectures (ARAs) are seeking to address. My comparison of the single- and multi-accelerated apps will look at the overhead involved in the management of accelerators. I expect the multi-accelerated app to perform faster than the single-accelerated app; however, I also expect a greater percentage of time will be spent doing management overhead. I anticipate numbers similar to those presented in Jason Cong's "Architecture Support for Accelerator-Rich CMPs. Essentially, the multi-accelerated app will not be as fast as it could be were overhead costs reduced. I would use such a result as proof that the performance of multi-accelerated apps would be more dramatically improved than that of single-accelerated apps with the introduction of efficient ARA support, and therefore ARAs should be tested with apps that use many accelerators.

1 Introduction

With the end of Dennard scaling, power density now increases as transistors get smaller. Consequently, architects cannot utilize all of the transistors on chip. Large areas of the CPU must be throttled or turned off to stay within power budgets, and this utilization wall is expected to grow even more limiting in the future. Specializing the CPU is one approach to solving this problem. On a specialized CPU, computation will require fewer transistors to be on at once. The tradeoff is that there now need to be many different specialized blocks which each compute a unique workload, and managing these is an architectural challenge. Recently, new architectures have come out of the research community which attempt to solve this problem. These primarily deal with managing memory and communication between specialized blocks (Accel store, ARC).