Energy-Efficient Execution of Multicore Parallel Programs under Limited Power Budget

Sunil Kumar
IIIT-Delhi, India

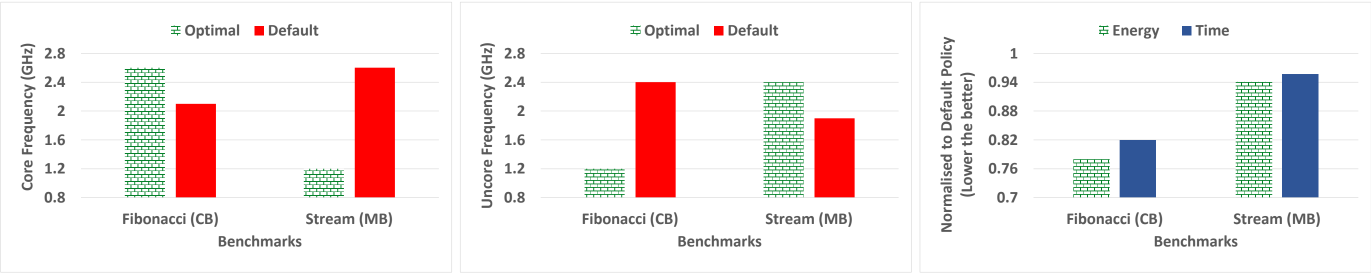
Vivek Kumar
IIIT-Delhi, India

Sridutt Bhalachandra
Lawrence Berkeley National Laboratory, USA

Future exascale systems are likely going to be power limited. Efficient use of the available power budget will be necessary to maximize performance. Cache coherent multicore processors with hundreds of cores will be the building blocks of one class of these exascale systems. Most multicore processor architectures are a combination of core and uncore elements. The uncore includes all chip components outside the CPU core, such as shared caches, memory controllers, and interconnects. Dynamic Voltage and Frequency Scaling (DVFS) and Uncore Frequency Scaling (UFS) are the two widely used techniques for limiting power usage for core and uncore. Another popular power-saving knob, Power Cap (PCAP), has been shown to automatically throttle the core and uncore frequencies under the specified power budget based on an application's Memory Access Pattern (MAP). However, the core and uncore frequencies set by the hardware's PCAP policy can easily lead to performance degradation as it is naive. Existing approaches generally fail to provide a one-stop solution for dynamically configuring core and uncore frequencies under a specified PCAP with different parallel programming models and applications.

An application's MAP can be primarily classified as Memory-Bound (MB) or Compute-Bound (CB). MB applications have a high number of memory accesses compared to CB applications. Figure 1 presents a motivational analysis demonstrating the effects of incorrect core and uncore frequency settings by the default PCAP policy for CB and MB applications when compared to the optimal. We used recursive Fibonacci number calculation and Stream Triade as CB and MB applications, respectively. Figures 1(a) and 1(b) compare the core and uncore frequencies set by the processor's default PCAP policy v/s the hardcoded optimal frequencies under the same PCAP. Figure 1(c) shows that using an optimal frequency setting can help improve both performance and energy savings.

This poster presents a work-in-progress runtime-based one-stop solution for achieving an energy-efficient execution on multicore processors under a limited power budget. The proposed runtime dynamically adapts the core and uncore frequencies on Intel processors without requiring training runs. Our runtime will be oblivious to the parallel programming model and the concurrency decomposition techniques used in an application. We plan to use a daemon thread that periodically profiles an application's MAP through hardware performance counters. The daemon would then explore optimal core and uncore frequencies under a given PCAP based on the application's MAP for achieving an energy-efficient execution without impacting performance.



(a) Core frequency settings

(b) Uncore frequency settings

(c) Optimal execution behaviour

Fig. 1. Analysis of Default v/s Optimal settings under 90W PCAP on a dual socket 12-core Intel Xeon Gold 6126 processor (125W TDP)