Power and Performance on Frequency Scaling Multicore Processors

Xi Yang

Supervisors: Peter Strazdins, Steve Blackburn and Eric McCreath

1 Introduction

The speed of CPU and IO devices has been improved dramatically in the last 10 years. However, the gap between CPU and IO devices is still huge. Nowadays, mainstream CPUs support dynamic frequency scaling. Therefore, we have a chance to evaluate how frequency scaling affects the performance and the power consumption of benchmarks which have different characteristics. Based on the experiment results, we propose a new OS architecture to reduce the power consumption of IO intensive systems without sacrificing the performance.

We picked up four kinds of benchmark: CPU intensive, memory intensive, network intensive and disk intensive. With the help of the CPU power meter, we record the execution time and the power consumption of benchmarks under different CPU frequencies.

2 Experiment environment

Tested machine has one Intel i5-670 CPU which has two cores, four hardware threads. 19 frequency stages are provided by i5-670: 3459MHz 3458MHz 3325MHz 3192MHz 3059MHz 2926MHz 2793MHz 2660MHz 2527MHz 2394MHz 2261MHz 2128MHz 1995MHz 1862MHz 1729MHz 1596MHz 1463MHz 133MHz 1197MHz. It has a RTL8111/8168B PCI Express Gigabit Ethernet card.

To measure the power consumption of the CPU, a current sensor is placed on the 12V power line which provides power for the CPU package. One Atmel AVR stick samples the current and posts data to the USB port. The sample rate of AVR stick is 50Hz. A cat program logs power values when the benchmark is running.

The CPU, the memory and the disk intensive benchmarks are picked up from the sysbench benchmark suite. The network benchmark is a home-brew benchmark.

• CPU intensive benchmark: 8 working threads handle 10000 request. Each request consists in calculation of prime numbers up to 10000.
• Memory intensive benchmark: 8 working threads read 3GB memory randomly.
• Disk intensive benchmark: 8 working threads read 2GB files sequentially with 'direct' flag. The flag makes sure that the data is read from the disk but not the buffer in memory.
• Network intensive benchmark: 4 working threads are running in tested machine. Each thread copies a 100MB file from a remote machine.

3 Result

As shown in Figure 1b, when the frequency is decreased from 3459MHz to 1197MHz, both of the performance and power consumption are reduced three times. Figure 2b shows that the memory intensive benchmark has similar trend with the CPU intensive benchmark. The difference between the CPU intensive benchmark and the memory intensive benchmark is that the CPU intensive benchmark draws higher power than the memory intensive benchmark in highest frequency, as shown in Figure 1a and Figure 2a.

Unlike CPU and memory intensive benchmarks, the performance of the disk intensive benchmark is not affected by scaling the CPU frequency. As shown in Figure 3b, unlike the execution time, power consumption is reduced when the CPU frequency is scaled down. If we compare Figure 3b with Figure 1b and Figure 2b, unlike CPU intensive and memory intensive benchmarks, compared with the power consumption in highest frequency, the disk intensive benchmark consumes 70% power in the lowest frequency. The reason is indicated in Figure 3a. Because the gap between the fast CPU and the slow disk, the CPU has to wait for the disk. When the core waits for the disk, it enters power saving idle state. If the core resides in idle state, the power consumption of the core drops to the lowest level. The green line of Figure 3a shows that although the CPU consumes three times power in the highest frequency when it is in active state, it has spent a lot of time in idle state. As a result, the average power consumption is decreased but not as much as CPU and memory intensive benchmarks. As shown in Figure 4b, the network intensive benchmark has similar trend with the disk intensive benchmark.

4 Proposed OS architecture

As shown in the last section, when we reduce the CPU frequency, execution time of disk and network operations is not affected but the power consumption is decreased. As the number of cores is increasing, we have to utilize these cores. At the same time, we have to reduce contentions between them. We propose a new OS architecture as shown in Figure 5 in which a core is dedicated to handle disk and network operations in low frequency. Results of benchmarks indicate that the low frequency core is able to handle these IO operations as fast as high frequency cores but draws less power. Because the disk and the Ethernet card are handled by one core. Contentions of shared data structures between cores could be eliminated. It is even possible to achieve high throughput with the low frequency core.