Performance and Energy-Efficiency of High Throughput Applications on Heterogeneous Multicore Systems

Patrick B. Millar, Marquette University
Advisor. Dr. Rong

Introduction

High-performance and energy-efficiency are goals that every computer system strives to achieve. As computing power is increasing, the amount of power required to maintain the systems is also increasing to unsustainable levels. Thus, computer systems must be examined in order to maximize the performance while keeping energy usage at reasonable levels.

In this experiment a heterogeneous multicore system specifically running high throughput applications was examined. The objective of the research was to find the optimal configuration of component usage to achieve high-performance and energy-efficiency.

Experimental Methodology

The TSP and FSM experiments conducted used three different computational unit distributions. These distributions were pure CPU, pure GPU, and combined CPU+GPU. The computational units executed simultaneously and performed as much work as possible during the limited runtime of each program.

The two evaluation metrics that were recorded to analyze results were performance and energy-efficiency. Performance is defined as the amount of useful computations completed per second; energy-efficiency is the amount of useful computations that can be completed per Joule.

The performance was calculated for each execution by recording the amount of useful computations completed and runtime of each program. Additionally, the energy-efficiency was found by recording the total power consumption and comparing the value to the useful computations completed.

Experimental Platform

The experiments were all performed on the a GPU-accelerated multicore computer Sandy. Sandy has 16 Xeon SandyBridge E5-2670 CPU cores as well as 2 Nvidia K20 GPUs. Each K20 GPU contains 4992 cores.

Application 1: TSP

The Travelling Salesman Problem is an extremely computationally intensive program. TSP functions by continuously attempting to find the shortest path between a multitude of different points.

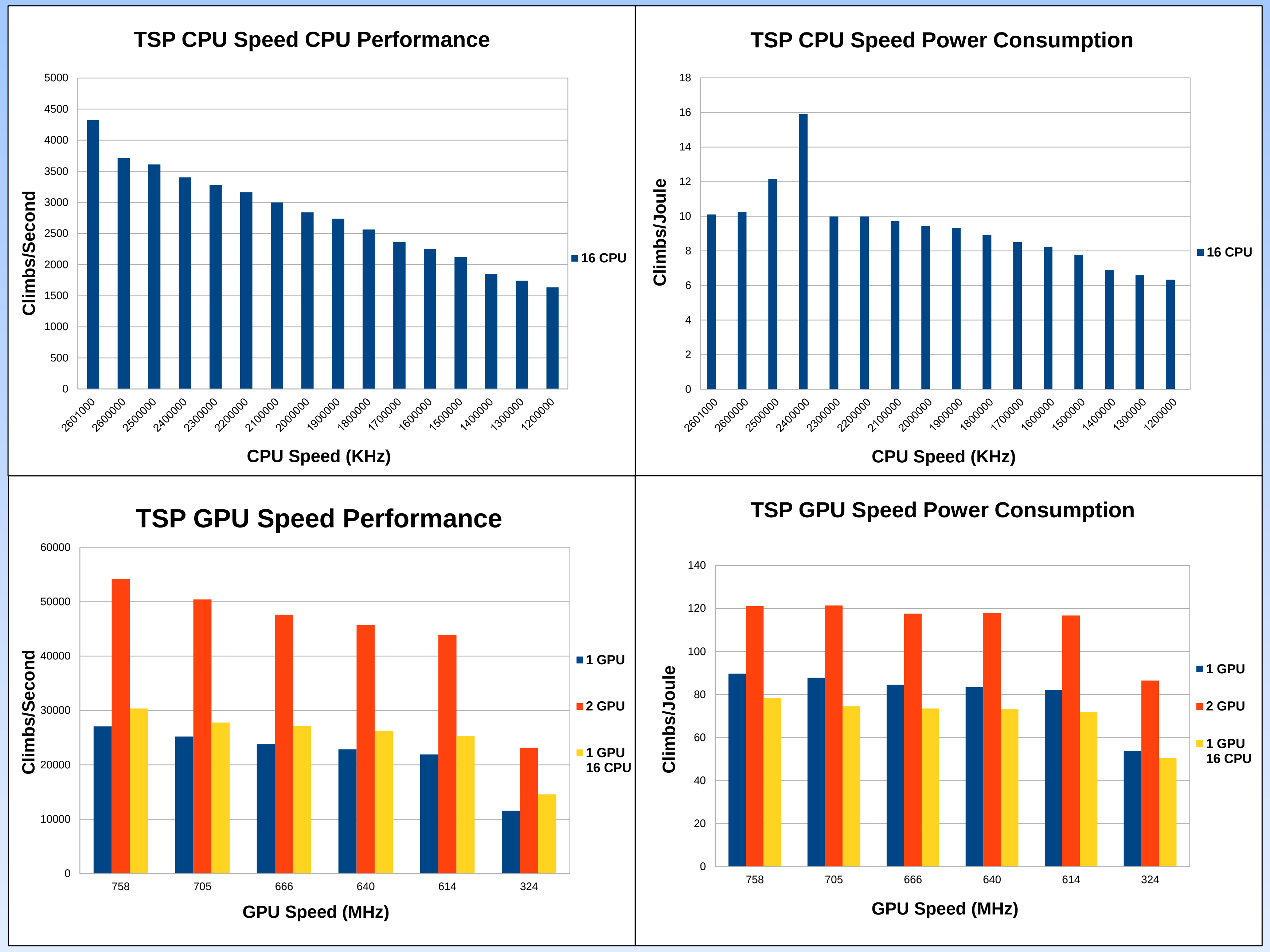

Figure 1 (Top Left): Peak CPU performance is found at the highest CPU frequency 2.601 GHz.

Figure 2 (Top Right): The optimal CPU energy efficiency occurs when the CPU is set at the frequency of 2.4 GHz.

Figure 3 (Bottom Left): Peak GPU performance is found at the highest GPU frequency 758 MHz.

Figure 4 (Bottom Right): The optimal GPU energy efficiency occurs when the GPU is set at the frequency of 705 MHz.

Application 2: FSM

The Finite State Machine program operates by constantly performing calculations to predict the next state. Once the prediction is complete, the state change occurs and a new change begins.

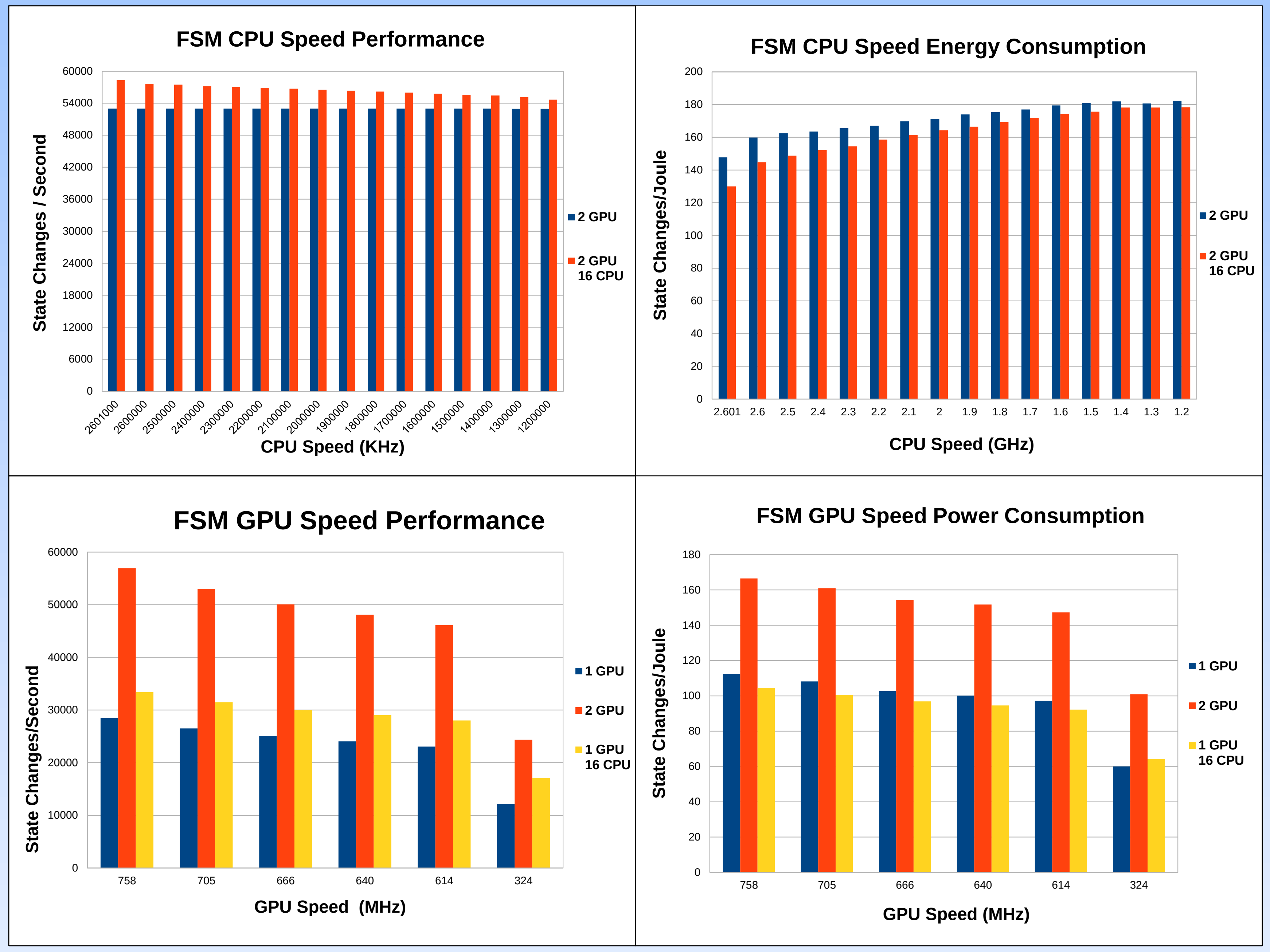

Figure 5 (Top Left): Peak system performance is found at the highest CPU frequency 2.601 GHz.

Figure 6 (Top Right): The optimal CPU energy efficiency occurs when the CPU is set at the frequency of 1.2 GHz.

Figure 7 (Bottom Left): Peak GPU performance is found at the highest GPU frequency 758 MHz.

Figure 8 (Bottom Right): The optimal GPU energy efficiency occurs when the GPU is set at the frequency of 758 MHz.

Conclusion

The findings showed that the performance on Sandy was the best while using all components available at the highest clock speed for both the CPU and GPU. This will only be true on other systems if the application allows for overlapped execution of heterogeneous components. Additionally, the results revealed that for maximum energy-efficiency on this system, both GPUs should be used at their highest clock speed 758 MHz, and the CPU should be disabled entirely. The default clock speed 705 MHz is extremely energy-efficient as well, but has slightly worse performance than the highest speed. Overall, the findings show that high speed GPUs will vastly outperform CPUs and conserve more energy while running high throughput applications.

References

[1] Martin Burtscher. A Scalable Heterogeneous Parallelization Framework for Iterative Local Searches. 2013

[2] Qiang Liu and Wayne Luk. Heterogeneous Systems for Energy Efficient Scientific Computing. 2012.

[3] Luk, Chi-Keung, Sunpyo Hong, and Hyesoon Kim. Qilin: Exploiting Parallelism on Heterogeneous Multiprocessors with Adaptive Mapping. 2009.