Core Resource Allocation and Power Management Optimization Using PARSEC Benchmark and MARS Framework

Nicolas Casilli, Northeastern University Class of 2022; Biswadip Maity, PhD Candidate at U.C. Irvine; Tiago Muck, PhD; and Nikil Dutt, PhD
NSF REU IoT-SITY Summer Research Program
Donald Bren School of Information & Computer Sciences, University of California, Irvine
Northeastern University

ABSTRACT

With the rise in popularity of multicore systems, there emerges a unique problem regarding constraints on power management and resource allocation. Thus, it is important to create a framework that enables all programs to benefit from performance gains by improving how processing resources are allocated in a multicore system. Through the use of the Heartbeat API, application performance can be measured to determine whether the current framework is optimizing resource usage as well as how the framework can be modified to intelligently optimize power and resource allocation management in real time. By adding the Heartbeat API to PARSEC Benchmarks, we obtained performance data on different applications with different thread counts to see how resource intensive those programs are as well as how much power those programs consume.

PARSEC AND MARS

What is PARSEC and why do we need it?

PARSEC is a benchmark suite that contains 13 different applications that represent expectations for future code with regards to features like: multiple threads, emerging workloads, diversity, and not HPC-focused. The diversity of these programs range from video and imaging processing to data mining and financial analysis.

What is MARS and why do we need it?

MARS is a framework that is planted between the OS and kernel levels of computer architecture. Currently, MARS records performance evaluations of applications annotated with the Heartbeat API and better optimizes processor resource allocation. When activated, the framework can change the performance of programs to match desired power saving levels.

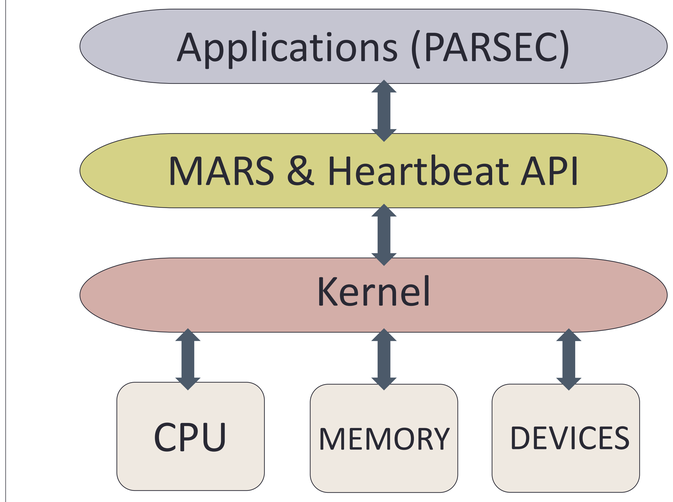
HEARTBEATS AND DOCKER

What is Heartbeats API and why do we need it?

Heartbeats API is a modifications system added to PARSEC applications used to transfer bits to/from the program and the kernel. These bits are referred to as “beats” and are implemented in computationally intensive functions in programs. Beat counts give information on the processing intensity required by certain programs as well as how much power those programs use within the context of the MARS framework.

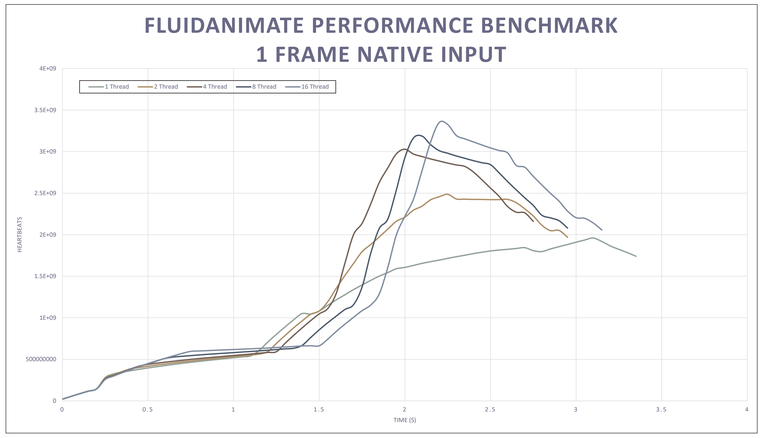
Why do we need Docker?

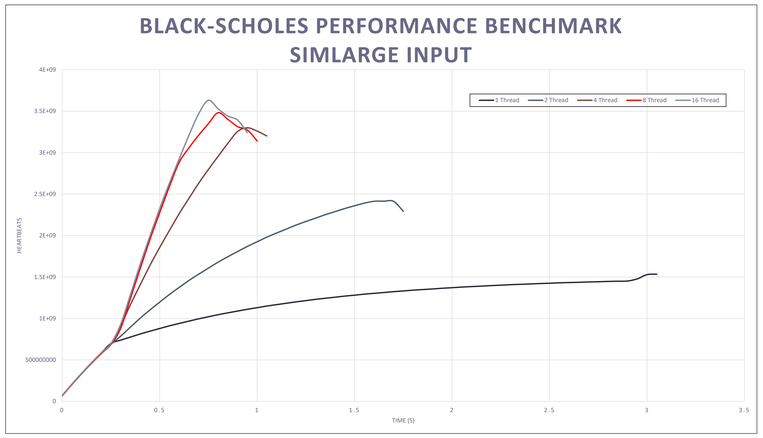
Docker is needed to create an isolated environment to enable researchers on other systems to compile, run tests, and produce data using the MARS framework.



Location of MARS and Heartbeat in relation to other features of modern computer architectures

RESULTS





Beats v. Time for PARSEC applications: Fluidanimate and Blacksholes with thread counts of: 1, 2, 4, 8, 16, and 32.

OBJECTIVES

To ensure the progress of the research, it was necessary to create an isolated Docker environment to run MARS and PARSEC applications on any machine and understand and apply the Heartbeats API to the necessary PARSEC programs. These programs needed to be modified and cross compiled. After cross compilation, the programs were run on an ODROID embedded with the MARS framework. These runs were conducted with differing thread counts so that information can be gained regarding ideal program runtime parameters. The final objective of the lab was to see how resource allocation would theoretically differ in multicore systems using the MARS framework when also accounting for real-time power management.

RESULTS

As opposed to not having any operational modified PARSEC applications, there are 10, of which 5 were run for data acquisition. The data acquired in the lab involved application performance during different thread count configurations (1, 2, 4, 8, 16, 32). In relation to this, it was found that performance gains improved at a much smaller rate at a configuration larger than 8 threads. A final result is the successful integration of Docker into the research lab, as the Docker enables other researchers to produce experimental results remotely.

CONCLUSIONS

MARS can be used as a QoS(Quality of Service) metric from the application side so that runtime resource managers/policy designers can more intelligently make decisions. This could be applied to using beat count parameters to establish a threshold of processing power so that performance can be assured during critical executions. The nature of the work conducted was very intermediary, but many conclusions were drawn describing the behaviors of the various PARSEC benchmarks, like each application’s unique execution functions.