

# Discovering Energy Usage Patterns on Scientific Clusters

## Extended Abstract

Matthew Bae  
Harvey Mudd College  
mbae@hmc.edu

Wucherl Yoo (Advisor)  
Lawrence Berkeley National  
Laboratory  
wyoo@lbl.gov

Alex Sim (Advisor)  
Lawrence Berkeley National  
Laboratory  
asim@lbl.gov

Kesheng Wu (Advisor)  
Lawrence Berkeley National  
Laboratory  
kwu@lbl.gov

## 1. INTRODUCTION

As scientific clusters are steadily growing, energy consumption is also growing as a problem. We can see power consumption well above 10 MW in today's top supercomputers.[4] Literature on energy awareness on smaller scales have helped address energy consumption and efficiency, and tools such as dynamic voltage and frequency scaling (DVFS) have helped increase energy efficiency.[2] However, energy consumption on petascale systems has only been addressed within the most recent decade. Within the past few years, energy measurement tools have been tested on supercomputers.[1, 5]

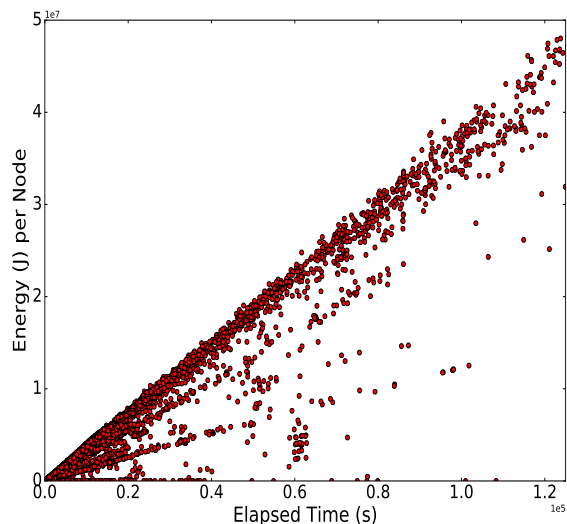
Discovering system resource usage patterns in clusters have been increasingly harder to find. As clusters grow, there is an increase in the volumes of data, number of machines, and exploited parallelism. This has produced increasing interactions of hardware components within clusters. Our work is motivated by observations that discovering system resource usage patterns can be conducted from monitored performance measurements from scientific clusters.

## 2. METHODS

Data was gathered on Cori, a supercomputer at Lawrence Berkeley National Laboratory. It has 1630 compute nodes with 32 physical cores per node, and each node has 128 GB of memory. Users submit jobs to Cori, and jobs are broken down into jobsteps. For our plots, points represent individual jobsteps for the month of May 2016.

We took two paths in analyzing the data. One path we took was to aggregate the job steps so that we had tuples of job characteristic values that corresponded to unique jobs. In doing this preprocessing, we are able to detect patterns on a per-job resolution. Another approach we took was not aggregating the job steps and instead creating tuples of job characteristic. In analyzing the data on a finer resolution, we can create analyses and detect patterns on specific parts of jobs, which is the path we took.

We also implemented an interactive plot viewer to help find pattern detections. The features include annotations, coloring based on threshold values of characteristics, and lassoing. One can use the annotations to look at specific points on the plots and read their job characteristics. Colorings based on threshold values can be used to see trends in job characteristics by partitioning the graph into two groups



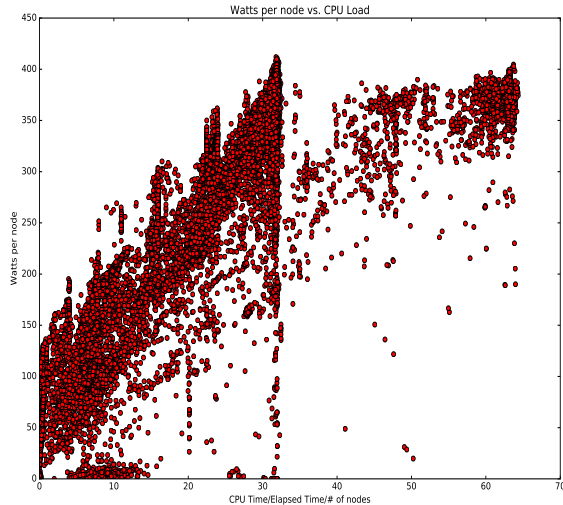
**Figure 1: Energy/# of nodes vs. Elapsed time. For all plots, CPU and Elapsed time are measured in seconds.**

based on the threshold. Lassoing lets one draw around a region and select jobs within the region for further analysis.

## 3. RESULTS

In figure 1, we can see that there are several trends formed by several lines. Each line corresponds to different values of cores/node, where the most prominent line has the largest cores/node value of 32. What this tells us is that for similar CPU times per node, we can see different amounts of energy output per node. In Figure 2, for similar cpu loads, there are large spreads of watts per node. In other words, we have different power consumption for similar CPU business.

In Figure 3, we identified clusters of energy consumption from the same application which use the same number of nodes using the interactive plot. They show different patterns depending on CPU utilization, memory usage, and number of CPUs. When memory usage is low as in the red and blue clusters, the watts per node increases with CPU



**Figure 2: Watts per node vs. CPU Time/Elapsed Time/# of nodes**

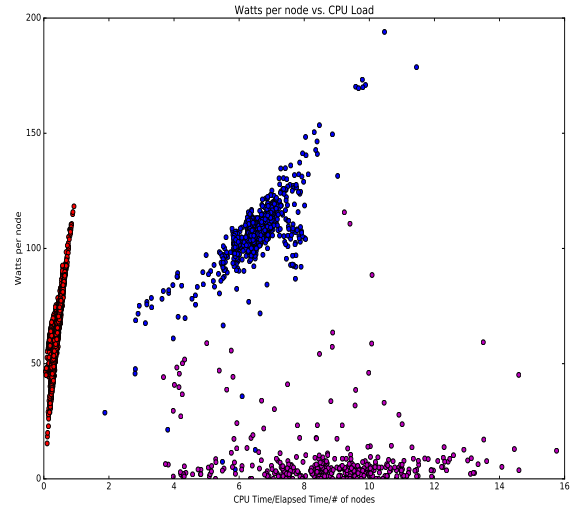
time/ Elapsed time (CPU utilization). Memory usage is higher in the purple cluster, and when memory usage is high, watts per node does not increase even with an increase in CPU utilization. This results from contention on the memory subsystem (shared cache and memory bandwidth). It follows that the purple region's executions are inefficiently executed and show higher elapsed time and CPU time. As a result, we believe that the number of CPUs per node needs to be decreased for high memory usage jobs executions in order to avoid contention. Our results show that assigning the proper number of CPUs per node is crucial for energy consumption.

## 4. CONCLUSION

In this summary, we have shown energy usage patterns on NERSC's supercomputer Cori. By analyzing SLURM logs and applying tools such as Apache Spark and Python, we were able to develop different methods for finding energy usage patterns. We developed an interactive plotting tool that is able to find energy usage patterns. Using the tool, we identified the assigned number of CPUs per node can change the energy usage pattern from the same application. As future work, we will investigate how to increase energy efficiency by optimizing the number of CPUs per node depending on the resource usage patterns of job executions.

## 5. ACKNOWLEDGEMENTS

I would like to thank my mentors for their guidance and my fellow interns at SDM for the support. This work was supported by the Office of Advanced Scientific Computing Research, Office of Science, of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. This work was also supported in part by the U.S. Department of Energy, Office of Science, Office of Workforce Development for Teachers and Scientists under the Science Undergraduate Laboratory Internship program.



**Figure 3: Clusters found from Figure 2 from the same application using 80 nodes. Colored based on CPU/# of nodes for values of 1, 12, and 32 from left to right respectively. Memory usage is highest in the purple region.**

## 6. REFERENCES

- [1] B. Austin and N. Wright. "Measurement and interpretation of microbenchmark and application energy use on the Cray XC30." In Energy Efficient Supercomputing Workshop of SC'14 (E2SC), 2014.
- [2] S. Byna and N. Wright. "Evaluation of Parallel I/O Performance and Energy with Frequency Scaling on Cray XC30." In Cray User Group (CUG), 2015.
- [3] "Cori Configuration." National Energy Research Scientific Computing Center. NERSC, n.d. Web. 15 July 2016.
- [4] J. Dongarra, M. Meuer, H. Simon, and E. Strohmaier. "Top 500 List: June 2016." Top500.org, 2016.
- [5] G. Fourestey, B. Cumming, L. Gilly, and T. Schulthess. "First Experiences with Validating and Using the Cray Power Management Database Tool." CoRR, abs/1408.2657, 2014.
- [6] "Intel Xeon Processor E5-2698 V3 (40M Cache, 2.30 GHz) Specifications." ARK Product Launch. Intel, n.d. Web. 15 July 2016.