A Machine-by-Machine Analysis of a Bi-Objective Resource Allocation Problem

Ryan Friese[1], **Tyler Brinks**[1,2], **Curt Oliver**[1]
Anthony A. Maciejewski[1], **Howard Jay Siegel**[1,2], **and Sudeep Pasricha**[1,2]
[1]Department of Electrical and Computer Engineering
[2]Department of Computer Science
Colorado State University
Fort Collins, CO, 80523

Abstract—*As high performance computing systems continually become faster, the operating cost to run these systems has increased. A significant portion of the operating costs can be attributed to the amount of energy required for these systems to operate. To reduce these costs it is important for system administrators to operate these systems in an energy-efficient manner. To help facilitate a transition to energy-efficient computing, the trade-offs between system performance and system energy consumption must be analyzed and understood. We analyze these trade-offs through bi-objective resource allocation techniques, and in this paper we explore an analysis approach to help system administrators investigate these trade-offs. Additionally, we show how system administrators can perform "what-if" analyses to evaluate the effects of adding or removing machines from their high performance computing systems. We perform our study using three environments based on data collected from real machines and real applications. We show that by utilizing different resource allocations we are able to significantly change the performance and energy consumption of a given system, providing a system administrator with the means to examine these trade-offs to help make intelligent decisions regarding the scheduling and composition of their systems.*

Keywords: bi-objective optimization; energy-aware computing; heterogeneous computing; resource allocation

1. Introduction

As large computing systems (e.g., supercomputers, clusters, datacenters) have increased in size and performance, the costs of operating these systems have increased as well. A significant portion of these costs can be attributed to the amount of energy that is required to run these systems. Between the years 2000 and 2006 the energy consumption more than doubled for high performance computing (HPC) systems, resulting in servers and datacenters accounting for approximately 1.5% of the total United States energy consumption for that year [1]. This amounts to approximately 61 billion kWh, or $4.5 billion in electricity costs. Energy consumption by HPC systems has continued to increase; from 2005 - 2011 the electricity consumption of these systems has increased by 56% worldwide [2].

Due to the increased electricity use and costs, some system administrators are now faced with the challenge of operating under limitations on electricity usage. To operate efficiently under these limitations, it is important to understand the trade-offs between system performance and system energy consumption. In [3] and [4] it was shown that increasing the energy consumption of a system often leads to an increase in the performance of the system, and vice-versa. Based on these studies, it is imperative for system administrators to analyze the trade-offs between energy consumption and performance of their systems to operate at a desirable level.

In this research, we examine how utilizing different resource allocations (i.e., mapping of tasks to machines) on a given system can allow us to analyze the trade-offs between energy consumption and performance for that system. The current state of the art resource managers, such as MOAB, are unable to reasonably determine the trade-offs between performance and energy based on our experience with it in cluster computing environments.

We model a heterogeneous distributed computing environment used to execute a workload consisting of a bag of tasks. In such an environment, a task may have different execution and power consumption characteristics when executed on different machines. This behavior requires one to explore different resource allocations to optimally manage the energy consumption and performance of the system. We define a resource allocation to be a complete mapping of all the tasks in the bag to the machines. The competing nature of minimizing energy consumption and increasing system performance allows this problem to be modeled as a bi-objective optimization problem.

Many bi-objective optimization algorithms exist, such as those found in [5], that can be adapted and used to produce resource allocations for our problem. We take the solutions produced by such algorithms and create graphical representations that allow us to analyze the performance and energy trade-offs. We produce plots that show general trends between performance and energy consumption, as well as more detailed graphs that allow us to analyze how different allocations use the system on a machine-by-machine basis. Analysis of these graphs can help system administrators find allocations that will allow the system to run at a specified energy/performance level as well as identify inefficiencies in their systems. Additionally, system administrators may desire to simulate the effect and observe the performance and energy consumption implications of adding or removing machines to the system. This could lead to the design of more efficient and cost effective computing systems.

We examine the trade-offs between energy consumption and performance for three different environments. Each environment is based on a set of real machines and real tasks. By analyzing numerous resource allocations, we show that for each environment the behavior of the systems can differ greatly, allowing system administrators to select a resource allocation that best fits the needs of their system.

In this paper we make the following contributions:

1) Perform a machine-by-machine analysis of how different resource allocations can affect the perfomance and energy consumption of a given system.
2) Provide an analysis approach that can identify both energy efficient and energy-inefficient machines, allowing system administrators to use this knowledge to help build and manage their systems.
3) Demonstrate the versatility of our analysis technique using three different heterogeneous environments.

The remainder of the paper is organized as follows. Related work is discussed in Section 2. We explain the system model in Section 3. In Section 4, we describe our bi-objective optimization problem. Our experimental setup is detailed in Section 5. Section 6 analyzes our simulation results. Finally, our conclusion and future work is given in Section 7.

2. Related Work

Several prior efforts have examined bi-objective resource allocation problems in large computing environments.

The bi-objective genetic algorithm NSGAII [6] is adapted for use within the resource allocation domain in [3] and [4]. System-level analyses are performed, specifically looking at the trade-offs of energy and makespan [3] or energy and utility [4]. Our current work performs an in-depth analysis on a machine-by-machine level with a realistic system model.

A bi-objective heterogeneous task scheduling problem between makespan and reliability is presented in [7] and [8]. Instead of reliability, our work investigates the trade-offs between makespan and energy consumption with a machine-by-machine allocation analysis.

In [9], the authors solve a bi-objective optimization between makespan and robustness for a heterogeneous scheduling problem. Solutions are created using a weighted sum simulated annealing heuristic, where one run of the heuristic produces a single solution. In our work we are concerned with multiple solutions, and analyzing how their allocations change based on their location in the search space.

A bi-objective flowshop scheduling problem between makespan and total tardiness is modeled in [10]. Solutions are created using a Pareto-ant colony optimization approach. While we could use methods such as this Pareto-ant approach, the focus of our paper is on the resulting allocations, not how they are created.

The authors of [11] model a homogeneous job-shop scheduling problem between makespan and energy consumption. We are interested in analyzing the behavior of systems that consist of heterogeneous machines, which significantly changes the problem and solution space.

An energy-constrained heterogeneous task scheduling problem is examined in [12]. In this environment, the energy constraint is realized by modeling devices with limited battery capacity in an ad-hoc wireless network. In our work, we are not directly concerned with an energy constraint, though by analyzing different solutions from a Pareto front, a solution could be picked that meets an energy constraint if it was needed.

In [13], the authors try to minimize energy consumption while trying to meet a makespan robustness constraint. Because there is a constraint on the makespan robustness, this is not a bi-objective optimization problem, and does not involve the type of machine-by-machine analysis that we preform.

There are many environments that can be modeled as dynamic resource allocation problems. One such environment is [14], where the system must complete as many tasks as possible by their individual deadlines while staying within the energy budget of the system. This environment does not perform a machine-by-machine analysis to to investigate the trade-offs between energy and makespan.

3. System Model

3.1 Machines

We model a heterogeneous suite of $\underline{M}$ machines, where each machine is one of $\underline{MT}$ machine types. Because we are modeling a heterogeneous system, machine type A may be faster for some tasks than machine type B, but may be slower for other tasks [15]. Machines are also heterogeneous in power consumption. We assume that each machine can only execute a single task at a time, similar to the Colorado State University ISTeC Cray [16]. Once a machine finishes executing all of its assigned tasks it shuts down, and no longer consumes any energy.

3.2 Workload

We model a workload environment where we have a bag of $\underline{T}$ tasks, and each task belongs to a given task type. Every task is known before the schedule is created. Due to the heterogenous nature of the system, each task type

i executing on machine type j will have known execution (Estimated Time to Compute (ETC)) and power consumption (Estimated Power Consumption (EPC)) characteristics denoted as ETC(i,j) and EPC(i,j). Tasks of the same task type have the same ETC and EPC characteristics. In resource allocation, it is common to assume the availability of such characteristics (e.g. [17], [18], [19], [20], [21]). These values may be taken from historical sources ([20], [19]) or may be constructed synthetically for simulation purposes ([22], [15]).

4. Bi-Objective Optimization

4.1 Overview

Many interesting engineering problems deal with multiple objectives. It is often the case that these objectives are competing with one another, and optimizing for one objective may cause the performance of another objective to decrease. It therefore becomes important for one to analyze the behavior (trade-offs) between these objectives. In our research we are trying minimize system makespan (Section 4.2.1) while trying to minimize system energy consumption (Section 4.2.2).

4.2 Objective Functions

4.2.1 Minimizing Makespan

One objective is to minimize makespan, which is defined as the time when all tasks have finished executing. Makespan is used to measure the performance of the system.

The makespan for a specific resource allocation, denoted μ, is the maximum machine finishing time in the system. The finishing time of a machine is the time at which all tasks T_m assigned to machine m have finished executing. Let $t_m \in T_m$, $\Upsilon(t_m)$ be the task type of t_m, $\Omega(m)$ be the machine type of m, and

$$F_m = \sum_{\forall t_m \in T_m} ETC(\Upsilon(t_m), \Omega(m)). \tag{1}$$

Makespan is given as

$$\mu = \max_{\forall m \in M} F_m. \tag{2}$$

4.2.2 Minimizing Energy Consumed

The other objective is to minimize total energy consumed. This is defined as the total amount of energy consumed by the machines to execute all tasks. The Expected Energy Consumption (EEC) of a given task t on a given machine m is

$$EEC[\Upsilon(t), \Omega(m)] = ETC[\Upsilon(t), \Omega(m)] \times EPC[\Upsilon(t), \Omega(m)]. \tag{3}$$

The total energy consumption for the system is

$$E = \sum_{\forall m \in M} \sum_{\forall t_m \in T_m} EEC[\Upsilon(t_m), \Omega(m)]. \tag{4}$$

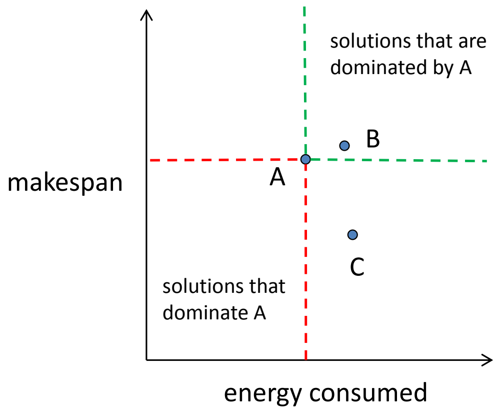

Fig. 1: Illustration of solution dominance for three solutions: A, B, and C. Solution A dominates solution B because A has lower energy consumption as well as a lower makespan. Neither solution A nor C dominate each other because A uses less energy, while C has a lower makespan.

4.3 Generating Solutions

In general, bi-objective optimization problems have a *set* of optimal solutions (not a single solution). This set of solutions is known as the Pareto optimal set, represented as the Pareto front in objective space, defined as the set of known solutions for which no better solutions in any objective have been found [23]. Pareto fronts are useful for analyzing the trade-offs between two objectives. A Pareto front is calculated from existing solutions, but it is not known where the true optimal set lies.

For a solution to exist within the Pareto optimal set, it must not be dominated by any other solution. Domination is defined as one solution being better than another solution in at least one objective, and better than or equal to in the other objective. A simple illustration of dominance is shown in Fig. 1. We have three solutions: A, B, and C. B is dominated by A because A has a lower makespan as well as a lower energy consumption. Thus any solution located in the upper right quadrant would be dominated by A, while any solution located in the lower left quadrant would dominate A. Neither solution A nor C dominate each other because A has a lower energy consumption, and C has a lower makespan. Both of these solutions would then be a part of the Pareto optimal set.

Pareto fronts can be generated using any number of algorithms, such as those found in [5]. The Pareto fronts found in this paper were created using the Non-Dominating Sorted Genetic Algorithm II (NSGAII) [6] adapted for use within the scheduling domain as described in [3]. Note that The method for generating the Pareto fronts are not the focus of this paper, rather it is the analysis of the resource allocations on the Pareto front.

Table 1: Machines Types (designated by CPU)

1	AMD A8-3870k
2	AMD FX-8159
3	Intel Core i3 2120
4	Intel Core i5 2400S
5	Intel Core i5 2500K
6	Intel Core i7 3960X
7	Intel Core i7 3960X @ 4.2 GHz
8	Intel Core i7 3770K
9	Intel Core i7 3770K @ 4.3 GHz

Table 2: Task Types

1	C-Ray
2	7-Zip Compression
3	Warsow
4	Unigine Heaven
5	Timed Linux Kernel Compilation

5. Experimental Setup

5.1 Datasets

To accurately model the relationships between machine performance and energy consumption in the ETC and EPC matrices, we used the method outlined in [4] where a dataset consisting of five applications executed on nine machines [24] is used to create a larger synthetic dataset. The synthetic data set resembles the original data set in terms of heterogeneity characteristics, such as the coefficient of variation, skewness, and kurtosis [25]. The original machine types (designated by CPU) are listed in Table 1 and the original task types are listed in Table 2. The original data contained both execution times and total system power consumption (measured at the outlet) for each task on each machine. Each machine used 16GB of memory, a 240GB SSD, and ran Ubuntu 12.04 (but had different processors).

5.2 Experiments

We considered three test environments each with 36 machines. The first test environment only used machine types 1 and 2 and there were 18 machines of each type in the system. The second test environment utilized machine types 1-6, and there existed six machines per type. Finally the third test environment consisted of the full set of nine machine types with four machines belonging to each type. The bag of tasks for each test environment was identical and consisted of 1000 tasks distributed among 30 task types (the five original task types and 25 synthetic task types).

6. Results

In Figs. 2- 4 we present the results of our experiments for the two machine type, six machine type, and nine machine type environments, respectively. In each fig., the subfig. "F" shows the Pareto front for each environment, where the x-axis is the total system energy consumption measured in megajoules(smaller is better) and the y-axis is the makespan of the system measured in minutes (smaller is better). Each individual marker in these plots represents a complete resource allocation. In each of these Pareto fronts, we see that makespan decreases (e.g. system performance increases) as the energy consumption of the system increases. This is consistent with the results from [3] and [4]. To better understand why this trend occurs, we analyzed five separate resource allocations from the Pareto front (the square markers in each of the Figs. 2F, 3F, and 4F).

For each of our selected resource allocations for each environment, we plotted the completion time and energy consumption of each machine, and grouped the machines by machine type as seen in subfigs. A-E for each environment. Subfigs. A-E range from illustrating the minimum energy consumption allocation in subfig. A to the minimum makespan allocation in subfig. E. For the minimum energy allocations (subfig. A), each task ends up being assigned to a machine that is part of the machine type that executes that task with the least amount of energy.

In the A subfigs. (in all three environments), we find that for each environment there is one machine type that has a longer finishing time than the other machine types (the left graph in the subfig.), and thus determines the makespan for this solution. This occurs because that machine type has more tasks for which it is the minimum energy machine type (implying it is a more energy-efficient machine type), thus those tasks will prefer to run on machines of that type, forcing the completion time for those machines to increase. There can exist many allocations that minimize energy consumption, therefore to be in the Pareto front, a minimum energy allocation must dominate by lowering the makespan of the system. The resource allocation heuristic accomplishes this by decreasing the finishing times of machines in the longest finishing time machine type. This results in balanced finishing times for machines of that type as can be seen in the subfigs. The other machine types have unbalanced finishing times among their machines because they do not have any effect on the makespan of the system. Another interesting observation is that in the environments with six and nine machine types (Subfigs. 3A and 4A) there exist machines that do not execute even a single task. This is because these machine types are not the minimum energy machines for any task.

Examining the subfigs. from A to E, we find the completion times of all the machines start to become balanced. This is because to lower the system makespan, a resource allocation must distribute the tasks to all the machines so that one machine (or machines in a machine type) is not forced to execute significantly more tasks than the other machines. Hence, working through the Pareto front (from left to right) the machine completion times become more balanced (lower makespan) by forcing tasks to run on machine types that consume higher amounts of energy. This becomes most apparent in the E subfigs., where the machines have balanced

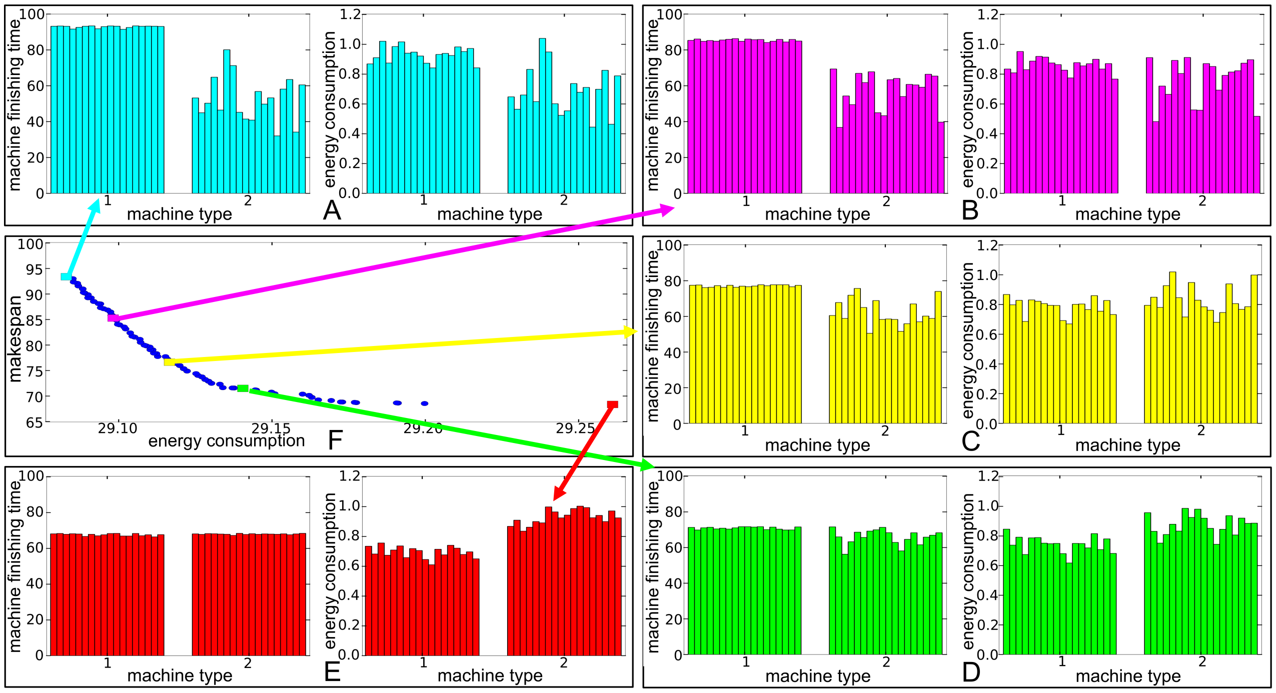

Fig. 2: Pareto front (F) and five resource allocations (A-E) for an environment with two machine types. The Pareto front illustrates the trade-offs between energy consumption in megajoules (x-axis) and makespan in minutes (y-axis). In A-E, the left graph shows the completion of each machine in minutes (y-axis) in the system. The right graph shows the energy consumption of each machine in megajoules (y-axis). In both graphs the machines are grouped by machine type (x-axis).

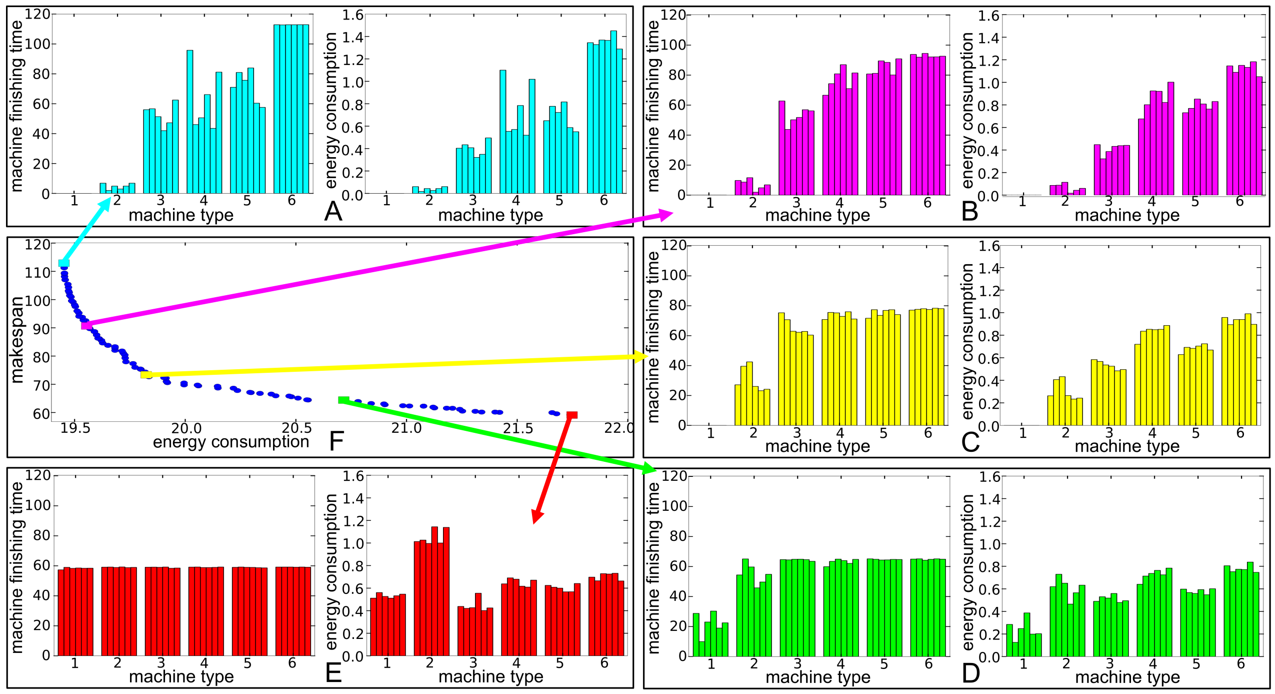

Fig. 3: Pareto front (F) and five resource allocations (A-E) for an environment with six machine types. The Pareto front illustrates the trade-offs between energy consumption in megajoules (x-axis) and makespan in minutes (y-axis). In A-E, the left graph shows the completion of each machine in minutes (y-axis) in the system. The right graph shows the energy consumption of each machine in megajoules (y-axis). In both graphs the machines are grouped by machine type (x-axis).

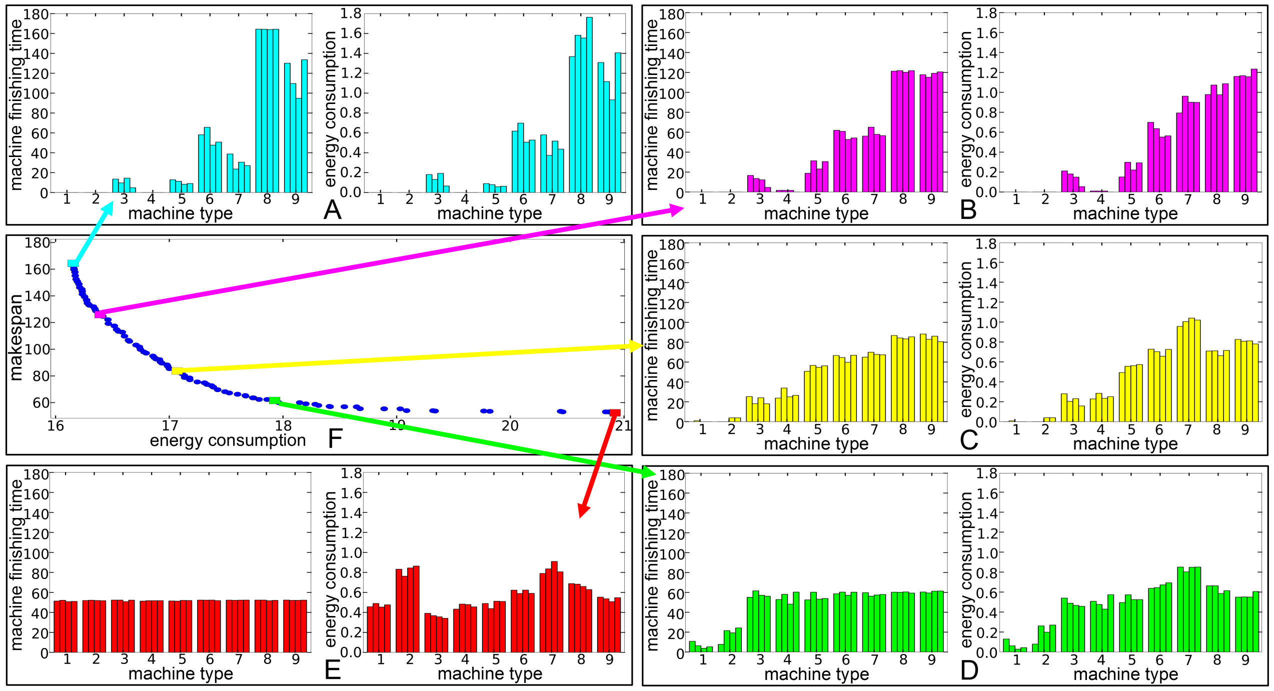

Fig. 4: Pareto front (F) and five resource allocations (A-E) for an environment with nine machine types. The Pareto front illustrates the trade-offs between energy consumption in megajoules (x-axis) and makespan in minutes (y-axis). In A-E, the left graph shows the completion of each machine in minutes (y-axis) in the system. The right graph shows the energy consumption of each machine in megajoules (y-axis). In both graphs the machines are grouped by machine type (x-axis).

finishing times (left graph) but their energy consumptions (right graph) are not balanced. This shows that as makespan decreases, the amount of energy consumed increases.

It is important to note in Figs. 3 and 4 that even as the allocations are decreasing makespan, there are still certain machines that execute very few tasks, and it is not until the lowest makespan allocation that they execute a comparable number of tasks as the other machines. This occurs because the trade-off in decreasing makespan versus the amount of energy the system would consume is not large enough to warrant using these energy-inefficient machines.

By examining Pareto fronts and various resource allocations from within those Pareto fronts, system administrators can gather important information about the operation of their systems. This includes finding machines and machine types that are energy-inefficient. With this knowledge, a system administrator may decide to leave these machines off to save energy unless it is absolutely necessary to finish a workload as fast as possible. Additionally, they can see which machine types are being utilized the most and make future purchasing decisions based on this information.

Finally, this study provides and example of how system administrators can perform “what-if” analyses. For example, what if we add more machine types to the system, what if we add more machines of a specific machine type, or what if we turn off certain machines? All of these scenarios could be simulated and then analyzed by the system administrator to help them decide how to best manage their system. We illustrate the power of these type of questions by comparing our three test environments against one another. We see that as we increase the number of machine types in the environment, we are able to both lower the makespan and have a smaller total energy consumption for the system. There are many reasons this may occur, the most straightforward is that more powerful and energy efficient machine types are added. Another reason is that additional machine types may increase the heterogeneity of the system, resulting in task-machine affinity being exploited. We are also able to see that it may be better to invest in machines that are of types 8 and 9 as they are the machines that execute the most tasks in the most energy efficient manner, while it may be best to not use machine types 1 or 2 at all as they both consume more energy than the other machine types.

The analyses performed in this work cannot be done by evaluation of only the ETC and EPC characteristics, rather, they require a more comprehensive analysis of the complex interaction between the workload, machines, two objectives.

7. Conclusions and Future Work

Energy-efficient computing is becoming very important due to the need for greater performance and the rising costs of energy consumption. System administrators must have tools that will allow them to evaluate the energy and performance characteristics of their systems. In this work, we provide a tool that allows system administrators to study the trade-offs between system performance and system energy consumption. We show that by analyzing individual resource allocations we can examine how a given system is distributing and executing tasks depending on the performance and energy consumption desired. This investigation can help identify the degree of energy-inefficiency of the machines, as well as allow system administrators to perform "what-if" analyses to determine the effect of adding or removing machines from their systems.

There are many directions for future work. These include examining different performance and cost objectives such as: maximizing robustness, minimizing temperature, maximizing utility, and minimizing monetary costs. We would like to enhance our power model by utilizing dynamic voltage and frequency scaling techniques to save additional energy.

8. Acknowledgements

This work was supported by the National Science Foundation (NSF) under grant number CNS-0905399, the NSF Graduate Research Fellowship, and by the Colorado State University George T. Abell Endowment. Any opinion, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. This research also used the CSU ISTeC Cray System supported by NSF Grant CNS-0923386. The authors thank M. Oxley and T. Hansen for their valuable comments.

References

[1] Environmental Protection Agency, "Report to congress on server and data center energy efficency," http://www.energystar.gov/ia/partners/prod_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf, Aug. 2007.

[2] J. Koomey, "Growth in data center electricity use 2005 to 2010," *Analytics Press*, Aug. 2011.

[3] R. Friese, T. Brinks, C. Oliver, H. J. Siegel, and A. A. Maciejewski, "Analyzing the trade-offs between minimizing makespan and minimizing energy consumption in a heterogeneous resource allocation problem," in *2nd Int'l Conf. on Advanced Communications and Computation (INFOCOMP 2012)*, Oct. 2012, p. 9.

[4] R. Friese, B. Khemka, A. A. Maciejewski, H. J. Siegel, G. A. Koenig, S. Powers, M. Hilton, J. Rambharos, G. Okonski, and S. W. Poole, "An analysis framework for investigating the trade-offs between system performance and energy consumption in a heterogeneous computing environment," in *IEEE 22nd Heterogeneity in Computing Workshop (HCW 2013)*, May 2013.

[5] A. Konak, D. W. Coit, and A. E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial," *Journal of Reliability Engineering ans System Safety*, vol. 91, pp. 992–1007, Sept. 2006.

[6] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr. 2002.

[7] J. J. Dongarra, E. Jeannot, E. Saule, and Z. Shi, "Bi-objective scheduling algorithms for optimizing makespan and reliability on heterogeneous systems," in *19th Annual ACM Symp. on Parallel Algorithms and Architectures (SPAA '07)*, 2007, pp. 280–288.

[8] E. Jeannot, E. Saule, and D. Trystram, "Bi-objective approximation scheme for makespan and reliability optimization on uniform parallel machines," in *14th Int'l Euro-Par Conf on Parallel Processing (Euro-Par '08)*, 2008, vol. 5168, pp. 877–886.

[9] B. Abbasi, S. Shadrokh, and J. Arkat, "Bi-objective resource-constrained project scheduling with robustness and makespan criteria," *Applied Mathematics and Computation*, vol. 180, no. 1, pp. 146–152, 2006.

[10] J. Pasia, R. Hartl, and K. Doerner, "Solving a bi-objective flowshop scheduling problem by Pareto-ant colony optimization," in *Ant Colony Optimization and Swarm Intelligence*, 2006, vol. 4150, pp. 294–305.

[11] Y. He, F. Liu, H.-j. Cao, and C.-b. Li, "A bi-objective model for job-shop scheduling problem to minimize both energy consumption and makespan," *Journal of Central South University of Technology*, vol. 12, pp. 167–171, Oct. 2005.

[12] J.-K. Kim, H. J. Siegel, A. A. Maciejewski, and R. Eigenmann, "Dynamic resource management in energy constraind heterogeneous computing systems using voltage scaling," *IEEE Parallel and Distrubted Systems*, vol. 19, no. 11, pp. 1445–1457, Nov. 2008.

[13] J. Apodaca, B. D. Young, L. Briceno, J. Smith, S. Pasricha, A. A. Maciejewski, H. J. Siegel, S. Bahirat, B. Khemka, A. Ramirez, and Y. Zou, "Stochastically robust static resource allocation for energy minimization with a makespan constraint in a heterogeneous computing environment," in *9th IEEE/ACS Int'l Conf. on Computer Systems and Applications (AICCSA '11)*, Dec. 2011, pp. 22–31.

[14] B. D. Young, J. Apodaca, L. D. Briceno, J. Smith, S. Pasricha, A. A. Maciejewski, H. J. Siegel, B. Khemka, S. Bahirat, A. Ramirez, and Y. Zou, "Deadline and energy constrained dynamic resource allocation in a heterogeneous computing environment," *Journal of Supercomputing*, vol. 63, Feb. 2013.

[15] S. Ali, H. J. Siegel, M. Maheswaran, D. Hensgen, and S. Ali, "Representing task and machine heterogeneities for heterogeneous computing systems," *Tamkang Journal of Science and Engineering*, vol. 3, no. 3, pp. 195–207, 2000.

[16] (2013, Mar.) ISTeC Cray high performance computing (HPC) system. http://istec.colostate.edu/istec_cray/.

[17] P. Chitra, R. Rajaram, and P. Venkatesh, "Application and comparison of hybrid evolutionary multiobjective optimization algorithms for solving task scheduling problem on heterogeneous systems," *Applied Soft Computing*, vol. 11, no. 2, pp. 2725–2734, Mar. 2011.

[18] M. K. Dhodhi, I. Ahmad, A. Yatama, and I. Ahmad, "An integrated technique for task matching and scheduling onto distributed heterogeneous computing systems," *Journal of Parallel and Distributed Computing*, vol. 62, no. 9, pp. 1338–1361, Sept. 2002.

[19] A. Khokhar, V. Prasanna, M. Shaaban, and C.-L. Wang, "Heterogeneous computing: Challenges and opportunities," *IEEE Computer*, vol. 26, no. 6, pp. 18–27, June 1993.

[20] A. Ghafoor and J. Yang, "A distributed heterogeneous supercomputing management system," *IEEE Computer*, vol. 26, no. 6, pp. 78–86, June 1993.

[21] M. Kafil and I. Ahmad, "Optimal task assignment in heterogeneous distributed computing systems," *IEEE Concurrency*, vol. 6, no. 3, pp. 42–50, Jul.-Sep. 1998.

[22] P. Lindberg, J. Leingang, D. Lysaker, S. U. Khan, and J. Li, "Comparison and analysis of eight scheduling heuristics for the optimization of energy consumption and makespan in large-scale distributed systems," *Supercomputing*, vol. 59, no. 1, pp. 323–360, Jan. 2012.

[23] V. Pareto, *Cours D'economie Politique*. Lausanne: F. Rouge, 1896.

[24] "Intel core i7 3770k power consumption, thermal," http://openbenchmarking.org/result/1204229-SU-CPUMONITO81#system_table, Jul. 2012, accessed: 07/24/2012.

[25] A. M. Al-Qawasmeh, A. A. Maciejewski, H. Wang, J. Smith, H. J. Siegel, and J. Potter, "Statistical measures for quantifying task and machine heterogeneities," *Journal of Supercomputing*, vol. 57, no. 1, pp. 34–50, Jul. 2011.