Green Computing with Geo-Distributed Heterogeneous Data Centers

Sudeep Pasricha[*†], Ninad Hogade[*], Howard Jay Siegel[*†], Anthony A. Maciejewski[*]
[*]Dept. of Electrical and Computer Engineering, [†]Dept. of Computer Science
Colorado State University, Fort Collins, Colorado 80523-1373
{sudeep, ninad.hogade, hj, aam)@colostate.edu

Abstract—**Cloud computing in data centers, as an alternative to computing on local machines, has become increasingly popular over the past decade. The need to reduce latency and improve bandwidth for customers has led cloud service providers to scale their data centers across the globe. Such geo-distributed data centers can be physically closer to various groups of target customers, enabling improved performance for their applications. But geo-distributed data centers can be an expensive proposition and require significant investment and justification in terms of return on investment (ROI). In this paper, we present a framework to leverage heterogeneous geo-distributed data centers to reduce electricity costs for cloud computing service providers. Our framework performs intelligent workload management across geo-distributed data centers to minimize the overall energy costs, while considering heterogeneity in data center compute capability, cooling power, workload co-location interference, time-of-use (TOU) electricity pricing, green renewable energy, net metering, and peak demand pricing distribution. Our experimental results indicate that our best technique can achieve an average of 61% cost reduction compared to the state-of-the-art.**

Keywords – geo-distributed data centers; workload management; memory interference; peak shaving; net metering

I. Introduction

The rapid growth of cloud computing has led to an evolution of data center deployment to support cloud computations. While centralized data centers were quite common in the past, more recently there has been a trend towards deploying data centers across geographically diverse locations [1], [2]. Distributing data centers geographically across the globe has traditionally offered many benefits, including better performance (lower latency) and lower cost for customers, due to customers being in physical proximity to the data center, and better resilience to environmental hazards and unpredictable failures, due to the redundancy that distributed data centers enable.

Another strong motivation to geographically distribute data centers is to reduce operating expenditures by exploiting time-of-use (TOU) electricity pricing [3]. Electricity prices are not constant, but rather follow a TOU pricing model where the cost of electricity varies based on the time of day [4]. Electricity prices are higher when total electrical grid demand is high and fall during periods when electrical grid demand is low [5]. Beyond the TOU electricity costs, most utility providers also charge a flat-rate (peak demand) fee based on the highest (peak) power consumed at any instant during a given billing period, e.g., month [6]. Reducing electricity costs has been a major focus of data center management, as it can lead to higher profits for cloud service providers as well as lower costs for customers. Minimizing costs is also important as the annual electricity expenditure for powering data centers has, in some cases, surpassed the costs of purchasing the equipment itself [7]. Recent studies confirm these trends: China's data centers alone are on track to use more energy than all of Australia by 2023 [8].

Relocating workloads among geo-distributed data centers is one effective approach to reduce electricity expenditures. Geo-distributed data centers have significant heterogeneity in energy costs and performance, due to various factors including: (a) different TOU pricing across sites, as they are often located in different times zones; (b) the use of on-site green/renewable energy sources, e.g., solar and wind, to different extents across sites; (c) the availability (or absence of) of net metering, which is a billing mechanism that gives renewable energy customers credit on their utility bills for the excess clean energy they sell back to the grid [9]; (d) different peak demand pricing from utility providers across sites; and (e) the availability of diverse resources within a data center, such as dynamic voltage and frequency scaling (DVFS), cooling infrastructure, and heterogeneity across compute nodes (e.g., different power and performance characteristics).

In this paper, we advocate for designing heterogeneity-aware geographical load distribution (HGLD) techniques to reduce overall electricity costs for cloud data centers. We propose a geo-distributed workload management framework that uses detailed models for data center computation and cooling power, workload co-location interference, TOU electricity pricing, renewable energy, net metering, and peak demand pricing distribution to enable significant reduction in overall energy costs for cloud computing. We present several strategies for heterogeneous geo-distributed data center management that attempt to concentrate the workload in regions with the lowest TOU electricity and peak demand pricing available at any given time, as well as to optimally utilize the heterogeneity available within the data center. The strategies described in this paper are a synthesis of our extensive prior work in the area of enabling heterogeneity in high performance computing, and were originally published in [10]-[13]. Our approach is most impactful for environments where historical execution information about the workload is readily available or can be predicted (e.g., using machine

978-1-7281-5416-9/19/$31.00 ©2019 IEEE

learning techniques). Examples of such environments exist in industry, e.g., DigitalGlobe, Google; military computing installations, e.g., Department of Defense; and government labs, e.g., National Center for Atmospheric Research.

II. System Model

A. Overview

We propose a framework for a heterogeneous geo-distributed resource manager (HGDRM) that consists of a high-level manager to distribute incoming workload requests and migrate already-allocated requests to geographically distributed data centers. The goal of the HGDRM is to minimize the total energy cost of the system while servicing all requests. Each data center has its own local workload management system that takes the workload assigned to it by the HGDRM and maps requests to compute cores within the data center. We first describe the system model at the geo-distributed level and then provide details into the models of components at the data center level.

B. Geo-Distributed Level Model

We consider a rate-based workload management scheme, where workload arrival rate can be predicted over the decision interval called epoch [16]. In our work, an epoch length is one hour, and a 24-epoch period represents a full day. Over the course of an epoch, the workload arrival rates can be reasonably approximated as constant, e.g., the Argonne National Lab Intrepid log shows mostly-constant arrival rates over large intervals of time [17].

We assume that the beginning of each epoch represents a steady-state scheduling problem where we assign execution rates, i.e., reciprocal of the execution time, of a set of I workload task-types to D data centers. A task-type $i \in I$ is characterized by its arrival rate, and the estimated time required to complete a task of that task-type on each of the heterogeneous compute nodes in each P-state. The assignment problem at the geo-distributed level is to assign execution rates for each task-type i to each data center $d \in D$ such that total energy cost across all data centers is minimized, with the constraint that the execution rates of all task-types meet their arrival rates, i.e., all tasks complete without being dropped or unexecuted.

C. Data Center Level Model

Each data center d houses a number of compute nodes that are arranged in a hot aisle/cold aisle manner (Figure 1), and a cooling system comprised of a number of computer room air conditioning (CRAC) units. Heterogeneity exists among compute nodes, where nodes vary in their execution speeds, power consumption characteristics, and number of cores. Cores within a compute node are homogeneous, and each core is DVFS-enabled to allow independent configuration of its P-states.

Recall that our HGDRM determines the distribution of tasks of each task-type among all data centers. In an epoch τ, at each data center d, we assign a desired fraction of time $DF_{i,k}(\tau\longleftarrow$ that each core k will spend executing tasks of type i and the P-state $PS_{i,k}(\tau\longleftarrow$ each core is configured to when executing tasks of type i. We assume tasks will run serially until completion. That is, a core sharing its time among multiple tasks implies that a scheduler will assign different tasks to execute on the core in such a manner that, over a long period of time (i.e., steady-state), the amount of time a core k spends executing a task of type i would equal its assigned $DF_{i,k}(\tau\longleftarrow$ value. At the data center level, we assign $DF_{i,k}(\tau\longleftarrow$ and $PS_{i,k}(\tau\longleftarrow$ such that power dissipation is minimized, and the sum of the execution rates of all task-types on cores in all data centers equal the arrival rate, ensuring that the arriving workload is fully executed.

The power consumption of a compute node consists of the static overhead power consumption (equal to the amount of power consumed when the system is idle) and the additional dynamic power consumed when cores are executing tasks. The heat generated by compute nodes is removed by the CRAC units. The airflow within the data center causes heat generated from nodes to propagate to other nearby nodes, thereby increasing the inflow temperature of those nodes. Using the notion of thermal influence indices [19] that were derived using computational fluid dynamics (CFD) simulations, we can calculate the steady-state temperatures at compute nodes and CRAC units in each data center. Because we assume the same physical layout for each of the data centers (Figure 1), we use thermal influence indices derived for one data center layout based on an average workload that would be executed by the data center. The outlet temperature of each compute node is a function of the inlet temperature, the power consumed, and the air flow rate of the node. The inlet temperature of each compute node is a function of the outlet temperatures of each CRAC unit and the outlet temperatures of all compute nodes of the same data center [18]. For all nodes, the inlet temperature of each node is constrained to be less than or equal to the red-line temperature (maximum allowable node temperature).

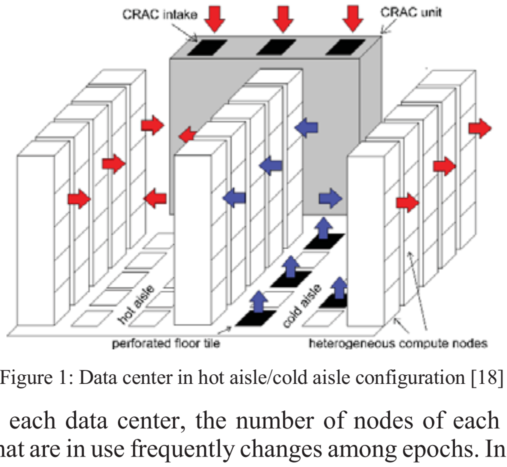


Figure 1: Data center in hot aisle/cold aisle configuration [18]

At each data center, the number of nodes of each node-type that are in use frequently changes among epochs. Inactive nodes are placed in a sleep state but entering and exiting this sleep state takes some time due to the actions required in both hardware and software to transition the system between states. Each node that is active is considered to be active for the entire epoch, which requires that any node transitioning to/from a sleep state does so during the epoch following/before the current epoch, respectively.

Each data center is equipped with and partially powered by a renewable energy source. Every location can have either solar power, wind power, or some combination of both. We use our power models with historical data [20] to predict the renewable power available at each data center. *Net metering* allows data center operators to sell back the excess renewable power generated on-site to the utility company. When the excess power is added into the grid, utility companies pay a fraction of the retail price. This fraction is called the net metering factor, α. Most utility providers also charge a flat-rate (peak demand) fee based on the highest (peak) power consumed at any instant during a given billing period, e.g., month. We consider this peak demand price per kW at each data center *d* and attempt to minimize peak power consumption (henceforth referred to as *peak shaving*) when making allocation decisions.

Tasks competing for shared memory in multicore processors inside data center servers can cause severe performance degradation, especially when competing tasks are memory-intensive [21]. The memory intensity of a task refers to the ratio of last-level cache misses to the total number of instructions executed. We adapt a linear regression model from [22] and train it with various benchmarks on a set of server class multicore processors that define the nodes used in our study (see Table 1 in Section V for a description of these processors). This model for execution time prediction under co-location interference is derived from real workloads and machines with a small mean percent error of approximately 7%. The linear regression model uses a set of model features, i.e., inputs, based on the current tasks assigned to a multicore processor to predict the execution time of a target task *i* on core *k* in the presence of performance degradation due to interference from task co-location. These features are the number of applications co-located on that multicore processor, the base execution time, the clock frequency, the average memory intensity of all applications on that multicore processor, and memory intensity of application *i* on core *k*.

III. Problem Formulation

We consider a scenario with multiple data centers sharing a single workload. The system is assumed to be undersubscribed in the sense that the system is expected to have enough computation resources to complete the workload without requiring that any tasks be dropped. Though the system is undersubscribed, individual data centers may be executing at full capacity. The tasks originate off-site from the data centers, and we make the simplifying assumptions that the transmission time and (network transfer) cost from a task origin to a data center is equivalent for all data centers. The objective of an HGDRM is to minimize monetary electricity cost of the geo-distributed system while ensuring that the workload is completed according to the arrival constraints discussed earlier. The problem is especially challenging when considering the variable amount of renewable power available at each data center, the heterogeneity of compute nodes within a data center, and the additional constraint that the entire workload must complete without dropping any tasks. Having information about TOU electricity pricing, peak demand pricing, a prediction of the amount of renewable power, net metering policy at each data center, the incoming workload, and the execution speeds of task-types on the heterogeneous compute nodes allows our HGDRM to make intelligent decisions for allocating the workload, as briefly discussed next in Section IV.

IV. Heuristic Descriptions

The HGDRM allocates the incoming workload not only to individual data centers, but also to specific nodes within each data center. The HGLD problem is NP-hard [24], and therefore we propose three resource management heuristics for HGDRM, with each having different levels of detail of the system model available to it.

Force-directed load distribution (FDLD) is a variation of force-directed scheduling [23], a technique often used for optimizing semiconductor logic synthesis. FDLD is an iterative heuristic that selectively performs operations to minimize system forces until all constraints are met. We adapt the FDLD approach proposed in [24] to the rate-based allocation environment we have outlined in Section II, and enhance it to propose two new FDLD based heuristics to solve our problem. Our baseline FDLD heuristic is the one proposed in [24], which we enhance with simple over-provisioning (FDLD-SO) to compensate for performance degradation due to co-location. This allows the FDLD heuristic to meet the execution rate constraint at a given data center. This heuristic over-provisions all task-types equally by scaling estimated task execution rates by a factor φ. Our first new heuristic improves upon FDLD-SO by using task aware over-provisioning (FDLD-TAO) to estimate co-location effects for each task-type by a factor specific to each task-type *i*, φ_i. For both FDLD-SO and FDLD-TAO, the degree of over-provisioning (φ and φ_i, respectively) is determined empirically through simulation studies to provide values that give the system the best possible performance. Lastly, our second new heuristic (FDLD-CL) uses co-location models to account for co-location effects when calculating task execution rates.

We also designed a third heuristic: a genetic algorithm for load distribution with co-location awareness (GALD-CL). The Genitor style [25] GALD-CL heuristic has two parts: a genetic algorithm based HGDRM, and a local data center level greedy heuristic that is used to calculate the fitness value of the genetic algorithm. The initial population is generated by randomly partitioning the global task arrival rate for each task-type *i* in an epoch across all data centers. More details of all three heuristics can be found in [10].

V. Experiments

A. Experimental Setup

Experiments were conducted for three geo-distributed data center configurations containing four, eight, and sixteen data centers. Locations of the data centers in the three configurations were selected from major cities around the continental United States to provide a variety of wind and solar conditions among sites at different times of the day (see Figure 2). Experiments for the configuration with four data centers used locations one through four from Figure 2, while experiments using configuration with eight and sixteen data

978-1-7281-5416-9/19/$31.00 ©2019 IEEE

centers used locations one through eight and one through sixteen, respectively. The sites of each configuration were selected so that each configuration would have an even east coast to west coast distribution to better exploit TOU pricing, peak demand pricing, net metering, and renewable power.

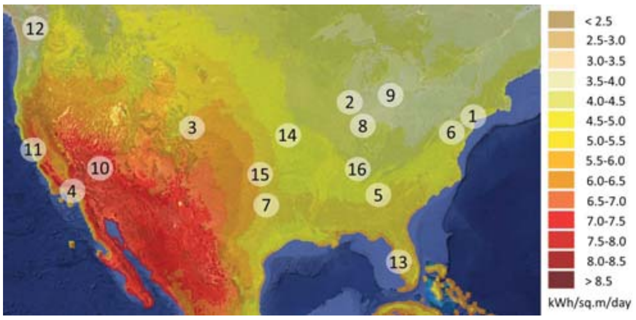


Figure. 2: Location of data centers overlaid on solar irradiance intensity map (average annual normal irradiance); wind data collected but not shown [20]

TABLE 1: Node Processor Types Used in Experiments

Intel processor	# cores	L3 cache	frequency range
Xeon E3-1225v3	4	8MB	0.8 - 3.20 GHz
Xeon E5649	6	12MB	1.60 - 2.53 GHz
Xeon E5-2697v2	12	30MB	1.20 - 2.70 GHz

Each data center consists of 4,320 nodes arranged in four aisles, and is heterogeneous within itself, having nodes from either two or three of the node-types given in Table 1, with most locations having three node-types and per-node core counts that range from 4-12 cores. For CRAC units, the red-line temperature was set to 30°C, which is on the high end of ASHRAE's temperature guidelines [26]. Nodes placed in a sleep state by a heuristic are considered to be in the Advanced Configuration and Power Interface (ACPI) node sleep state *S3*, where RAM remains powered on. Sleep state *S3*, also commonly referred to as suspend or standby, allows greatly reduced power consumption while still possessing a small latency to return to an active operating state. Sleep power for all nodes is calculated as a fixed percentage of static power for each node-type, assumed to be 16% based on a study of node power states [27]. The Coefficient of Performance (CoP) of the CRAC unit was determined empirically by simulating workloads with different memory intensity classes at each data center location, and its value ranges between 1.43 and 2.08 for different configurations. The time of each epoch τ was set to be one hour. The time required to transition a node to or from a sleep state was conservatively assumed to be five minutes. The electricity prices used during experiments, were taken directly from Pacific Gas and Electric (PG&E) Schedule E-19, which is for commercial locations consuming between 500kW and 1MW [28]. The peak demand prices per kW are given in Table 2.

We assume that each data center has peak renewable power generating capacity equivalent to its maximum power consumption [29]. Renewable power at each location was either wind power, solar power, or a combination of the two. Solar and wind data was obtained from the National Solar Radiation Database [20]. An example of the renewable power available at different locations is given in Figure 3. In net metering, data centers send excess power back to the grid and utility companies pay back the customer a fraction of retail price. In most cases, the net metering factor is 1; in very few cases, it is less than 1; and in some cases, it is 0, i.e., net metering is not available at that location [30]. Each task-type used in our experiment is representative of a different benchmark from the PARSEC [14] and NAS parallel [15] benchmark suites. Task execution times and co-located performance data for the task of the different memory intensity classes were obtained from running the benchmark applications on the nodes listed in Table 1 [22].

TABLE 2: Peak Demand Prices Used in Experiments

data center location	peak demand price ($\$/kW$)	data center location	peak demand price ($\$/kW$)
New York	11.04	Detroit	14.54
Chicago	3.82	Las Vegas	8.25
Denver	6.75	San Francisco	13.01
LA	8.91	Seattle	3.29
Atlanta	8.11	Tampa	10.25
Baltimore	3.84	Kansas City	6.39
Dallas	11.88	Oklahoma City	6.20
Indianapolis	10.57	Nashville	5.09

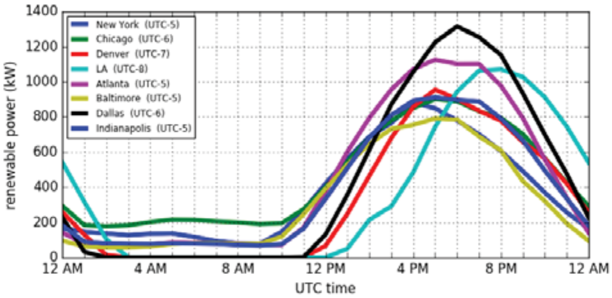


Figure 3: Renewable power available at eight locations

B. Results

We analyzed the total system energy cost for the four heuristics discussed in Section IV (FDLD-SO [24], FDLD-TAO, FDLD-CL, and GALD-CL). For each heuristic, we evaluate four variants: (1) without peak shaving and net metering, (2) with net metering only, (3) with peak shaving only, and (4) with both peak shaving and net metering. Simulations were performed with hybrid (mix of memory and compute intensive) workloads for four, eight, and sixteen data center configurations. For each configuration, the average performance improvement of each heuristic over the FDLD-SO heuristic with no peak shaving and no net metering is shown in Table 3. The GALD-CL heuristic was limited to a run time of approximately one hour. FDLD heuristics for four, eight, and sixteen locations completed on average in two, six, and eighteen minutes per epoch simulated, respectively. These experiments confirm that all FDLD heuristics can perform well for smaller and larger problem sizes but the GALD-CL heuristic consistently performs the best for all problem sizes.

TABLE 3: Energy Cost Reduction Comparison

	heuristic	no PS and no NM	NM only	PS only	PS and NM
4 data centers	FDLD-SO FDLD-TAO FDLD-CL GALD-CL	0.0% 12.3% 15.0% 49.5%	0.2% 14.5% 15.7% 58.2%	23.4% 33.4% 36.7% 67.3%	23.6% 34.0% 37.9% 75.8%
8 data centers	FDLD-SO FDLD-TAO FDLD-CL GALD-CL	0.0% 16.9% 19.6% 40.9%	0.3% 16.8% 20.8% 45.7%	20.8% 31.5% 36.9% 54.5%	21.1% 32.5% 38.3% 60.8%
16 data centers	FDLD-SO FDLD-TAO FDLD-CL GALD-CL	0.0% 11.1% 14.2% 33.1%	0.4% 12.2% 16.1% 36.4%	21.7% 30.9% 33.4% 44.0%	24.5% 32.9% 36.6% 46.6%

PS = peak shaving, NM = net metering

Table 3 shows similar energy cost reduction results for all FDLD variants in the cases of the data center configurations containing four, eight, and sixteen data centers running hybrid workloads. But for GALD-CL, we notice that the energy cost reduction decreases with the increasing number of data centers. Here, as the number of data centers in the group grows larger, the problem size increases and the number of GALD-CL generations that can take place within the time limit (one hour by default) decreases, which decreases the performance of GALD-CL.

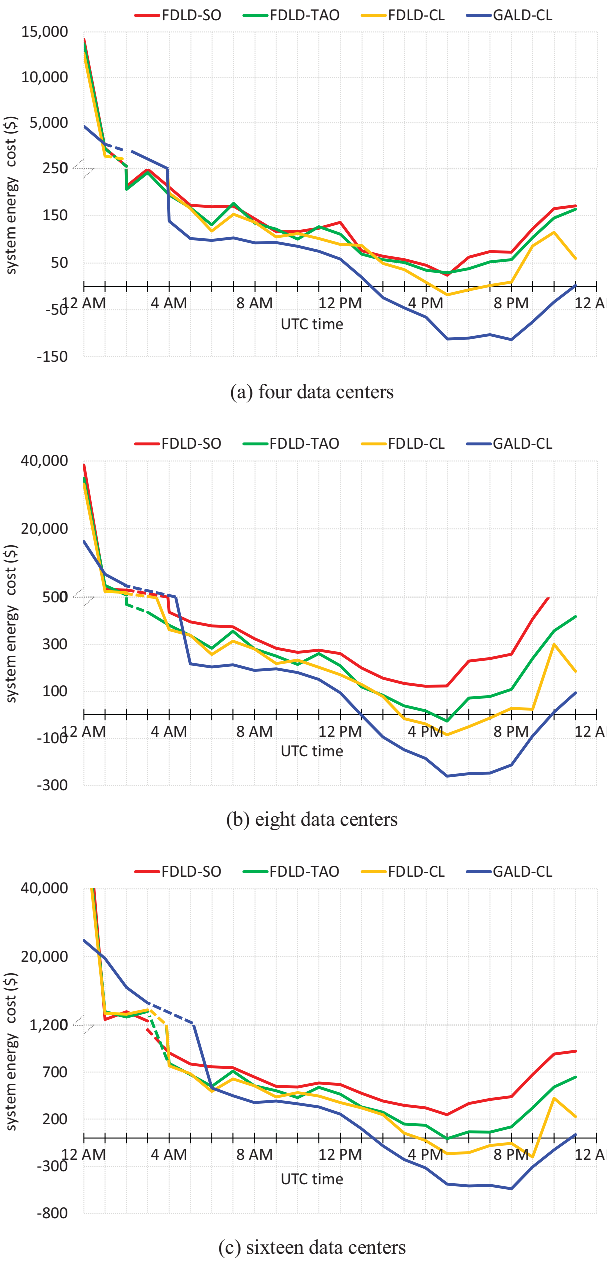
TABLE 4: Impact of GALD-CL Run Time

	GALD-CL epoch	no PS and no NM	NM only	PS only	PS and NM
4 data centers	1 hour	49.5%	58.2%	67.3%	75.8%
8 data centers	1 hour 2 hours	40.9% 46.7%	45.7% 55.2%	54.5% 62.5%	60.8% 72.2%
16 data centers	1 hour 4 hours	33.1% 40.1%	36.4% 49.4%	44.0% 61.6%	46.6% 69.9%

PS = peak shaving, NM = net metering

To better understand how the GALD-CL solution quality is impacted by the heuristic's run time, we increase the GALD-CL run time in proportion to the increase in number of data centers. We execute GALD-CL for about one hour for four data centers, about two hours for eight data centers, and about four hours for sixteen data centers. The results from Table 4 show that GALD-CL is capable of performing well for larger problem size, when given more time. For comparison, Table 4 also includes results for GALD-CL executed for about one hour for a group of four, eight, and sixteen data centers. It should be noted that even when allowing the GALD-CL to execute for one hour, it still provides the system a significant energy cost reduction in comparison to all the FDLD heuristics as shown in Table 3.

For most of our experiments, we analyzed the total system cost for each heuristic over one day. Figure 4 shows a more detailed view of the system operating cost at one-hour intervals over the course of a day for four, eight, and sixteen data centers executing a hybrid workload. The four resource management heuristics in this study consider both peak shaving and net metering. Net metering causes the plots to go into the negative region in certain epochs, which represents the case when the system earns money by selling excess renewable power back to the utility companies. The operating cost for each heuristic is very high during the first few epochs because the period for which the results are shown represents the first day of the month where the initial peak demand cost is added. After a few epochs, the performance of FDLD-CL came close to that of GALD-CL but was not able to surpass its performance.



(a) four data centers

(b) eight data centers

(c) sixteen data centers

Figure 4: System energy costs for each heuristic over a day for net metering factor sensitivity analysis, for a configuration with (a) four, (b) eight, and (c) sixteen data center locations running the hybrid workloads

VI. CONCLUSIONS

In this paper, we studied the problem of minimizing the energy costs for cloud computing by leveraging geographically distributed heterogeneous data centers. Renewable (green) energy, peak demand, and workload co-location interference at data centers have a significant impact on energy consumption. We capture these effects by including net metering, peak shaving, data center cooling, and task co-location models in our workload distribution techniques. We analyzed several techniques that possess varying degrees of co-location interference prediction information. We demonstrated that including these models and effects in the decision-making process of the resource management heuristics resulted in a lower overall energy cost. This is achieved by reducing or eliminating node over-provisioning while still meeting all required workload execution rate constraints. Our proposed FDLD-CL and GALD-CL heuristics resulted in 37% and 61% lower operational costs on average, respectively, than an approach from prior work (FDLD-SO [24]). However, to implement our approach in a real system, it must be used with some form of a workload prediction technique, e.g., [31].

ACKNOWLEDGMENTS

The authors thank Eric Jonardi, Mark Oxley, and Daniel Dauwe for their valuable contributions on this work. This work is supported by the National Science Foundation (NSF) under grant CCF-1813370.

REFERENCES

[1] "Data center locations," http://www.google.com/about/datacenters/inside/locations/index.html. Accessed Aug. 1, 2019.

[2] "Global infrastructure," http://aws.amazon.com/about-aws/ global-infrastructure/. Accessed Aug. 1, 2017.

[3] Y. Li, et al., "Operating cost reduction for distributed internet data centers," in 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, May 2013, pp. 589–596.

[4] "What is time-of-use pricing and why is it important?" http://www.energy-exchange.net/time-of-use-pricing/. Sep. 1, 2019.

[5] "Dynamic pricing," http://whatis.techtarget.com/definition/ dynamic-pricing. Accessed Aug. 1, 2017.

[6] "Demand charges," http://www.stem.com/resources/learning/. Accessed Aug. 1, 2017.

[7] C. Hsu, "Rack PDU for green data centers," in Data Center Handbook, H. Geng, Ed. John Wiley and Sons, Inc., Nov. 2014.

[8] "Greenpeace: China's data centers on track to use more energy than all of Australia," [Online]: https://www.datacenterknowledge .com/asia-pacific/greenpeace-china-s-data-centers-track-use-more-energy-all-australia

[9] "Net metering," http://freeingthegrid.org/. Accessed Aug. 1, 2017.

[10] N. Hogade, S. Pasricha, A. A. Maciejewski, H.J. Siegel, M. Oxley, and E. Jonardi, "Minimizing energy costs for geographically distributed heterogeneous data centers", IEEE Transactions on Sustainable Computing (TSUSC), vol. 3, no. 4, Oct-Dec 2018, pp. 318–331.

[11] E. Jonardi, M. A. Oxley, S. Pasricha, A. A. Maciejewski, and H. J. Siegel, "Energy cost optimization for geographically distributed heterogeneous data centers," in 6th International Green and Sustainable Computing Conference (IGSC '15), Dec. 2015, 6 pp.

[12] A. M. Al-Qawasmeh, S. Pasricha, A. M. Maciejewski, and H. J. Siegel, "Power and thermal-aware workload allocation in heterogeneous data centers", IEEE Transactions on Computers, vol. 64, Iiss. 02, Feb. 2015, pp. 477–491,

[13] M. Oxley, S. Pasricha, T. Maciejewski, H.J. Siegel, and P. Burns, "Online resource management in thermal and energy constrained heterogeneous high performance computing," IEEE International Conference on Big Data Intelligence and Computing (DataCom), Aug. 2016, pp. 604–611.

[14] "PARSEC benchmark suite," http://parsec.cs.princeton.edu/ index.htm/. Accessed Aug. 1, 2017.

[15] "NAS parallel benchmarks," https://www.nas.nasa.gov/ publications/npb.html. Accessed Aug. 1, 2017.

[16] P. Bodik, et al., "Automatic exploration of datacenter performance regimes," in Workshop on Automated Control for Datacenters and Clouds (ACDC), June 2009, 6 pp.

[17] D. G. Feitelson, D. Tsafrir, and D. Krakov, "Experience with using the parallel workloads archive," Journal of Parallel and Distributed Computing, vol. 74, no. 10, Oct. 2014, pp. 2967–2982.

[18] M. A. Oxley, E. Jonardi, S. Pasricha, A. A. Maciejewski, H. J. Siegel, P. J. Burns, and G. A. Koenig, "Rate-based thermal, power, and co-location aware resource management for heterogeneous data centers," Journal of Parallel and Distributed Computing, vol. 112, no. 2, Feb. 2018, pp. 126–139.

[19] H. Bhagwat, et al., "Fast and accurate evaluation of cooling in data centers," Journal of Electronic Packaging, vol. 137, no. 1, Mar. 2015.

[20] NREL, "National solar radiation database," https://mapsbeta. nrel.gov/nsrdb-viewer/. Accessed Apr. 15, 2015.

[21] S. Govindan, J. Liu, A. Kansal, and A. Sivasubramaniam, "Cuanta: Quantifying effects of shared on-chip resource interference for consolidated virtual machines," in 2nd ACM Symposium on Cloud Computing (SOCC '11), Oct. 2011, 14 pp.

[22] D. Dauwe, E. Jonardi, R. D. Friese, S. Pasricha, A. A. Maciejewski, D. A. Bader, and H. J. Siegel, "HPC node performance and energy modeling with the co-location of applications," The Journal of Supercomputing, vol. 72, no. 12, Dec. 2016, pp. 4771–4809.

[23] P. G. Paulin and J. P. Knight, "Force-directed scheduling for the behavioral synthesis of ASICs," IEEE Transactions on Computer-Aided Design (TCAD), June 1989.

[24] H. Goudarzi and M. Pedram, "Geographical load balancing for online service applications in distributed datacenters," in IEEE CLOUD, June 2013, pp. 351–358.

[25] D. Whitley, "The GENITOR algorithm and selective pressure: Why rank-based allocation of reproductive trials is best," in 3rd lnternational Conference on Genetic Algorithms, 1989, pp. 116– 121.

[26] "Thermal guidelines for data processing environments-expanded data center classes and usage guidance," Technical report, American Society of Heating, Refrigerating, and Air- Conditioning Engineers, Inc., 2011, 45 pp.

[27] C. Isci, et al., "Agile, efficient virtualization power management with low-latency server power states," in International Symposium on Computer Architecture, June 2013, pp. 96– 107.

[28] Pacific Gas and Electric Company, "Electric schedule e-19," http: //www.pge.com/tariffs/tm2/pdf/ELEC SCHEDS E-19.pdf. Accessed Apr. 15, 2015.

[29] G. Cook, et al., "Clicking clean: How companies are creating the green internet," Greenpeace Inc., Washington, DC, Apr. 2014.

[30] "Net metering policies," http://www.dsireusa.org/. Accessed Aug. 1, 2017.

[31] R. N. Calheiros, E. Masoumi, R. Ranjan, and R. Buyya, "Workload prediction using ARIMA model and its impact on cloud applications' QoS," IEEE Transactions on Cloud Computing, vol. 3, no. 4, Oct. 2015, pp. 449–458.