CloudSocket: Smart Grid Platform for Datacenters

Seil Lee[†‡], Hanjoo Kim[†], Seongsik Park[†], Seijoon Kim[†], Hyeokjun Choe[†], Chang-Sung Jeong[‡] and Sungroh Yoon[†*]
[†]Electrical and Computer Engineering, Seoul National University, Seoul 08826, Korea
[‡]Electrical Engineering, Korea University, Seoul 02841, Korea
[*]Correspondence: sryoon@snu.ac.kr

***Abstract*—Today's datacenters are equipped with diverse computing and storage devices for handling a myriad of data and normally consume a significant amount of electrical energy. This paper proposes a smart grid inspired methodology to monitor and profile the energy consumption of a datacenter, with the aim of providing information useful for reducing the peak power consumption of the datacenter. Our energy measurement platform is named *CloudSocket*, and each CloudSocket unit can measure the power consumption of an individual computing node and periodically transmit the measurement information wirelessly to the coordinator unit that can manage many CloudSockets simultaneously. We tested our methodology with a 32-node grid system that runs Apache Spark for large-scale data analytics. Analyzing our experimental results reveals how and where the peak power of each node in the grid overlaps, providing opportunities for informative coordination of the computing components for overall power reduction.**

I. Introduction

With advancements in information technology (IT) and mobile devices, data is expanding exponentially in various areas [1]. Datacenters are generating interest as the main infrastructures that can store, manage, and analyze such data [2].

Datacenters generally draw huge amounts of electrical energy. In light of recent trends, it is expected that the power consumption of datacenters will double in the next five years. In addition, approximately 80% of power production depends on fossil fuels that emit greenhouse gases [3]. Therefore, power efficiency in datacenters remains critical. Additionally, deep learning and other parallel/distributed analysis methods [4] usually involve a tremendous number of computations [5].

To maintain stable power, power provisioning is inevitable due to unwanted high-peak power [6]. Although high-peak power can occur in a short period, exceeding capacity can cause a catastrophe such as a blackout, which can be fatal in a datacenter [7]. However, power-provisioning techniques generate extra costs related to additional facilities and power consumption. There exist system-level power-reduction techniques based on dynamic voltage scaling [8], IO optimization by storage access pattern profiling [9], and architecture exploration using low-power storage devices [10]. The effectiveness of such techniques for a datacenter is yet to be validated.

In this paper, we propose a novel method for monitoring the energy consumption of a datacenter. Our approach is inspired by the next-generation power grid also known as a smart grid, which produces and distributes energy efficiently. A smart grid typically requires the use of advanced metering infrastructure (AMI) systems [11], which can monitor precise relationships between power supply and demand in real time to build efficient power networks. We similarly construct a small-scale smart grid system for a datacenter. The power consumption of each worker node in the datacenter is measured by a measurement device named *CloudSocket*. By using our approach, we can save the unnecessary cost of power provision because the power consumptions of each worker node is measured and predicted accurately so that the distribution of power of nodes is adjusted properly.

II. Related Work

To characterize power consumption patterns of home appliances, Park et al. [12] developed a simulation platform for modeling power loads. To extend this work to a distributed environment, Mukherjee et al. [13] measured the power consumption of individual nodes in a datacenter by using a commercial power profiling device. Feng et al. [14] investigated power consumption patterns over time for a distributed computing system. These approaches are limited in that they cannot provide a way to simulate or measure the power consumption of individual nodes separately.

Using extra devices to measure the power consumption of individual nodes will increase overall costs. Nonintrusive load monitoring (NILM) was introduced to avoid the cost of using an additional measurement device [15]. NILM predicts the power consumption of individual nodes from the total power consumption measured by using a disaggregation technique based on the power-consumption signature of each load.

However, disaggregation of devices with similar signatures is difficult. Furthermore, identifying characteristics of appliance power signatures is challenging for datacenters because servers in a datacenter are always on unlike home appliances.

Ferreira et al. [16] presented a technique for mapping power consumption in a datacenter at power panel level to individual nodes by installing a USB device on a specific node for synchronous detection. Nevertheless, this method cannot analyze datacenter workloads from a power consumption perspective.

III. Proposed Methodology

A. CloudSocket Hardware

Figure 1 depicts the basic architecture of CloudSocket. It is composed of two parts, namely the power strip with sensors inserted and the CloudSocket main board. CloudSocket is capable of measuring the power of eight worker nodes simultaneously, as shown in Figure 1-(A). In addition, CloudSocket

978-1-5090-5142-7/16/$31.00

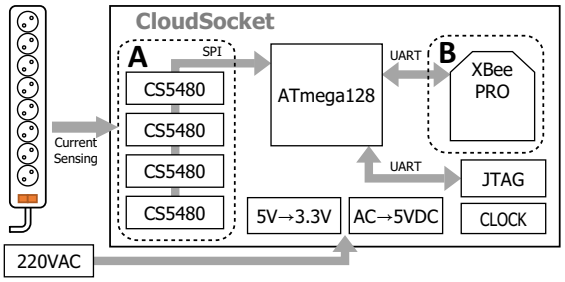

Fig. 1. Block diagram of CloudSocket, whose main features are (A) simultaneous monitoring of eight nodes and (B) wireless communication

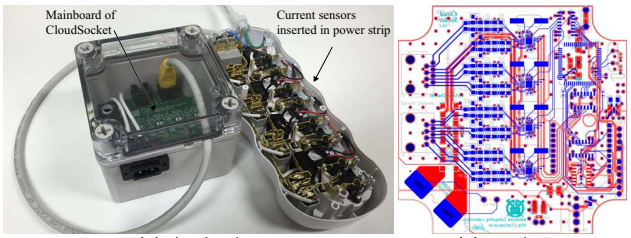

Fig. 2. (A) CloudSocket (mainboard, enclosure, and current sensors are shown) and (B) layout of our PCB design

can transmit the measured data wirelessly using the Zigbee module shown in Figure 1-(B).

The current sensors were inserted into a power strip that links to the main board. In the power strip, we opened connections on individual switches and then reconnect them after placing the current sensors, as shown in Figure 2-(A). The current channel requires two connections for a measurement of one worker node, which is coupled to the main board by two strands of Cat 5 Ethernet cables. The main board part is depicted in the left of Figure 2-(A), and the open power strip is shown in the right of Figure 2-(A). Each CloudSocket includes four CS5480s [17] and can measure the power consumption of eight worker nodes. The microcontroller unit (MCU) accesses four units of CS5480 microchip joined to the MCU by serial peripheral interface (SPI), and it then collects the power observation data in round-robin fashion. Resolution of the measurement is adjustable by setting sample counts from the sensors. We set the master clock to 4.096MHz (thus, one measurement per second by setting the sample count to 4,000).

An XBee RF module was used for wireless communication. The Zigbee protocol meets and supports IEEE 802.15.4 standards and enables highly scalable networks at low power and cost [18]. The bandwidth of the protocol has a maximum RF data rate of 250 kbit/s. In our experiments to be presented, we created a star network by making a number of CloudSockets and collected the power consumption data therefrom.

Our device is powered by transforming 220VAC into 5.0VDC and 3.3VDC without using an additional DC supply. The electrical expenditure of CloudSocket itself is 1W according to measurements conducted using an off-the-shelf product [19], which is comparable with the power consumption of a clock radio [20]. Given that each of the worker nodes used in our experiments consumed approximately 30–100W, the energy consumption of CloudSocket itself is negligible.

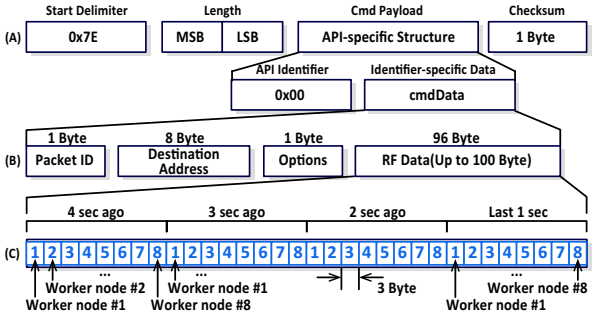

Fig. 3. (A) API command, (B) packet, and (C) data frame

Our PCB design is shown in Figure 2-(B). To prevent electric shock and short circuits, we added a high-quality enclosure, such as that depicted in Figure 2-(A).

B. CloudSocket Software

The wireless network used forms a star topology composed of a coordinator socket and multiple CloudSockets. The coordinator socket differs from each CloudSocket in that it has no power measurement functionality, focusing on gathering data from CloudSockets through wireless communication.

After CloudSocket initializes, it measures the electrical power consumption of the worker nodes attached (each CloudSocket can measure multiple nodes) once every second and stores the measurement in its buffer memory. As stated early, the MCU accesses four CS5480 microchips in a round-robin fashion, and each microchip quantifies the power consumption of two worker nodes as a type of 3-byte data (24-bit, two's complement) for each worker node. As a result, 24 bytes of data is generated and stored in the buffer memory for each second. In every four seconds, the collected 96 bytes of data in the buffer is transmitted to the coordinator socket.

Figure 3 shows the structure of the wireless communication packet. The XBee module can handle various commands. The structure of the API command is depicted in Figure 3-(A). The API identifier represents that the command means transmission, and a packet includes the address of the receiver and the measurement data, as shown in Figure 3-(B). The packet ID is the serial number of a packet, and the destination address is the 64-bit hardware address of the coordinator socket. The measurement data in Figure 3-(C) is enumerated in consecutive order, which is 96 bytes buffered for four seconds, and then sent to the destination (the coordinator socket).

The coordinator socket determines network association only if personal area network (PAN) ID and the channel number match. To create a scalable network, although a new CloudSocket is unknown to the coordinator socket initially, if it attempts to participate in the network association with the authorized PAN ID and channel number, the network is then expanded automatically. The packets received at the

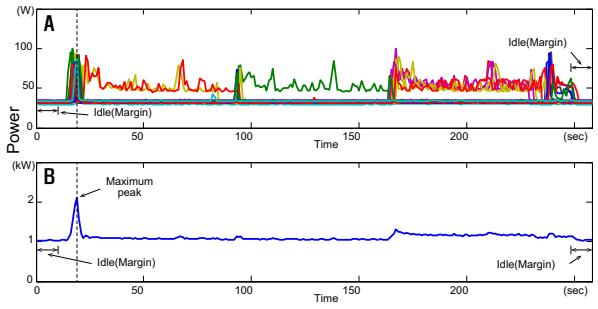

Fig. 4. Power consumption pattern of LDA, with (A) individual power and (B) total power

coordinator socket are gathered by using the hardware address of the sending CloudSocket.

IV. Experiments

A. Setup

We created a distributed environment for experiments. Each worker node had an Intel i7-4790 CPU with 16GB main memory and a 2TB HDD. We connected a set of 32 worker nodes through a Gigabit Ethernet switching hub. In addition, we equipped an NVIDIA GeForce GTX 970 GPU on each worker node. Each worker node ran Ubuntu 14.04 OS and Apache Spark. We measured the power of worker nodes while them executing cluster analysis with latent Dirichlet allocation (LDA) in MLlib [21], a machine learning library provided by Apache Spark. Each job was to process the 20 newsgroups data set [22]. To collect power profiling data from using GPUs, we performed image classification on the DeepSpark [23] framework with the ImageNet Large Scale Visual Recognition Competition (ILSVRC) 2012 data set [24].

B. Experimental Results

The LDA results are shown in Figure 4. Only a few of the worker nodes participated in the operation for most of the execution time of approximately four minutes on average. As shown in Figure 4-(A), the power consumption of worker nodes overlapped at the beginning of the workload. Furthermore, the dominant peak of the power consumption considering the total power consumption shown in Figure 4-(B) was measured once in the early stage of the operation; it occurs at around 20 seconds.

The experiments with the ILSVRC 2012 data using DeepSpark took approximately 14 hours. It involved two main steps, namely data preparation and model learning. We were able to measure the difference between the patterns in the two steps at around 2,500 seconds; the change of steps is illustrated in Figures 5-(A) and (B). As shown in Figures 5-(C) and (D), enlarged at around 15,000 seconds in the model-learning step, the power consumption pattern repeated to a certain degree. We conjecture that this may have been caused by various reasons, such as the network access patterns of the algorithm, relatively low power consumption, and frequent updates of the parameters by CPU/GPU for learning.

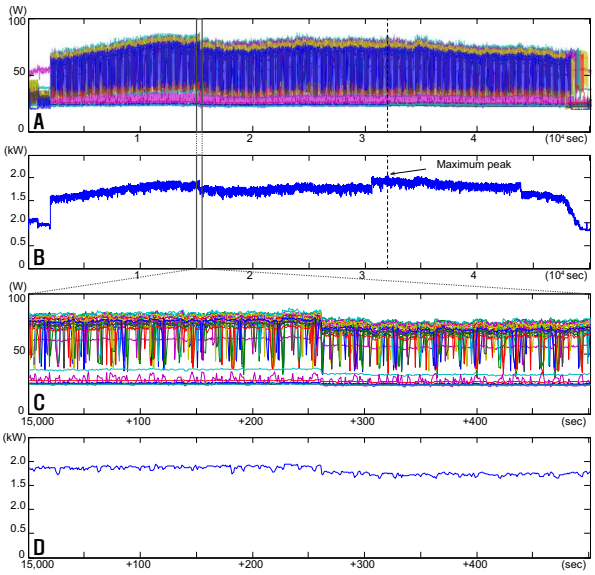

Fig. 5. Power consumption pattern of DeepSpark. (A) Individual power, (B) total power, (C) partial section of (A), and (D) partial section of (B)

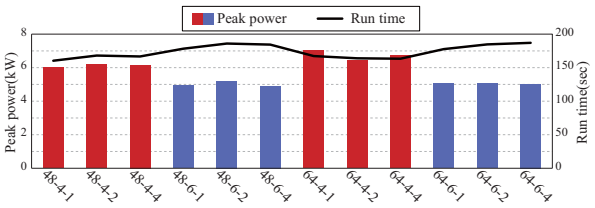

Fig. 6. Peak power for all nodes. Labels of x-axis represent [Number of Executors]-[Executor Memory in GB]-[Executors core]

Figure 6 shows the results of repetitions for the experiments shown in Figure 4 with varying the parameters of the Apache Spark platform used. We alternated the number of executors, the executor memory capacity in GB, and the number of executor cores over the sets of {48, 64}, {4, 6} and {1, 2, 4}, respectively. The label of x-axis represents the Spark parameters as [Number of Executors]-[Executor Memory in GB]-[Executors core]. The results indicate that the peak power of the entire system is low when [Executor Memory in GB] is 6. In other words, for the parameters of the Spark platform tested, [Executor Memory in GB] has the highest correlation with the peak power. This suggests that we can effectively reduce the peak power by adjusting the parameters of the distributed environment.

V. Discussion

A. Regarding Maximum Peak Power Reduction

As seen in Figure 4, the peak power of all the worker nodes appears at the beginning of algorithm execution. The maximum value was especially dominant in LDA, as shown in Figure 4. Figure 7 shows the factors other than power consumption during the execution of the algorithm. The solid line indicates the average value of power consumption of the

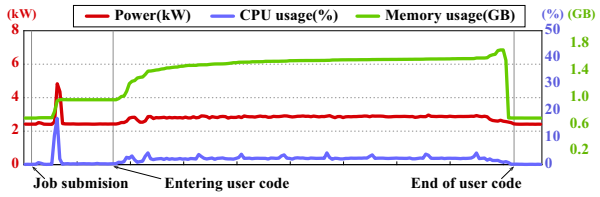

Fig. 7. Hardware resource usage

CPU and memory used, and the tags on the x-axis illustrate how certain execution processes affect the power consumption from a software perspective.

Figure 7 shows how the power consumption associated with CPU usage and memory-CPU usage may affect the peak power. When we search for an optimized balance of resources between the CPU and memory, it would thus be advantageous to invest more on CPU rather than memory in terms of power reduction. Furthermore, we observed that the maximum peaks, as shown in Figure 4, were generated between the point after the user job was submitted to the Spark platform and prior to the start of the algorithm. In other words, the peak is related to the resource scheduling in the Spark or Hadoop platform, regardless of the algorithm used. In addition, we can observe that the pre-peaks appear when many worker nodes use CPU resources. The pre-peaks tended to be dominant especially in the period when few nodes participated in the algorithm, as shown in Figure 4.

B. Future Work

Our goal with the proposed platform was to predict the peak power of a datacenter using CloudSocket and to eventually rebalance the workload for more efficient power provision. In many cases, intensive power consumption and the maximum peak were observed before the user program started. It would thus be necessary to conduct further research on how to reduce the peak incurred by the Spark platform itself. For instance, we discovered that some parameters of the Spark platform played critical roles in shaping the overall peak, as shown in Figure 6. Additionally, the power reduction problem could be formulated as an instance of mathematical optimization problem for more theoretical evaluation of different power reduction approaches.

ACKNOWLEDGMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R7117-16-0235, Development of HPC System for Accelerating Large-scale Deep Learning), by a research grant from SK Hynix Inc., and by the Brain Korea 21 Plus Project in 2016.

REFERENCES

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big data: The next frontier for innovation, competition, and productivity," 2011.
[2] L. A. Barroso, J. Clidaras, and U. Hölzle, "The datacenter as a computer: An introduction to the design of warehouse-scale machines," *Synthesis lectures on computer architecture*, vol. 8, no. 3, pp. 1–154, 2013.
[3] International Energy Agency, *CO_2 Emissions from Fuel Combustion (2015 Edition)*. OECD/IEA, 2015.
[4] Y. Jeon and S. Yoon, "Multi-threaded hierarchical clustering by parallel nearest-neighbor chaining," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 9, pp. 2534–2548, 2015.
[5] Q. V. Le, "Building high-level features using large scale unsupervised learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8595–8598.
[6] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2. ACM, 2007, pp. 13–23.
[7] H. Blodget. (2011) Amazon's cloud crash disaster permanently destroyed many customers' data. [Online]. Available: http://www.businessinsider.com/amazon-lost-data-2011-4
[8] J.-B. Lee, M.-J. Kim, S. Yoon, and E.-Y. Chung, "Application-support particle filter for dynamic voltage scaling of multimedia applications," *IEEE Transactions on Computers*, vol. 61, no. 9, pp. 1256–1269, 2012.
[9] B. Seo, S. Kang, J. Choi, J. Cha, Y. Won, and S. Yoon, "Io workload characterization revisited: A data-mining approach," *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 3026–3038, 2014.
[10] D. Kim, K. Bang, S.-H. Ha, S. Yoon, and E.-Y. Chung, "Architecture exploration of high-performance pcs with a solid-state disk," *IEEE Transactions on Computers*, vol. 59, no. 7, pp. 878–890, 2010.
[11] H. Farhangi, "The path of the smart grid," *Power and energy magazine, IEEE*, vol. 8, no. 1, pp. 18–28, 2010.
[12] S. Park, H. Kim, H. Moon, J. Heo, and S. Yoon, "Concurrent simulation platform for energy-aware smart metering systems," *Consumer Electronics, IEEE Transactions on*, vol. 56, no. 3, pp. 1918–1926, 2010.
[13] T. Mukherjee, G. Varsamopoulos, S. K. Gupta, and S. Rungta, "Measurement-based power profiling of data center equipment," in *Cluster Computing, 2007 IEEE International Conference on*. IEEE, 2007, pp. 476–477.
[14] X. Feng, R. Ge, and K. W. Cameron, "Power and energy profiling of scientific applications on distributed systems," in *Parallel and Distributed Processing Symposium, 2005. Proceedings. 19th IEEE International*. IEEE, 2005, pp. 34–34.
[15] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.
[16] A. Ferreira, W. El-Essawy, J. C. Rubio, K. Rajamani, M. Allen-Ware, and T. Keller, "Bcid: An effective data center power mapping technology," in *Green Computing Conference (IGCC), 2012 International*. IEEE, 2012, pp. 1–10.
[17] Cirrus Logic, Inc. Three channel energy measurement ic. [Online]. Available: https://www.cirrus.com/cn/pubs/proDatasheet/CS5480_F3.pdf
[18] Alliance, ZigBee, "IEEE 802.15. 4, ZigBee standard," *On http://www.zigbee.org*, 2009.
[19] X4-LIFE. Inspector 2. [Online]. Available: http://www.x4-life.de/produkt/inspector-ii/
[20] A. Alaskan. Power consumption table. [Online]. Available: http://www.absak.com/library/power-consumption-table
[21] Spark mllib. [Online]. Available: http://spark.apache.org/mllib/
[22] 20 newsgroups. [Online]. Available: http://qwone.com/~jason/20Newsgroups/
[23] H. Kim, J. Park, J. Jang, and S. Yoon, "Deepspark: Spark-based deep learning supporting asynchronous updates and caffe compatibility," *arXiv preprint arXiv:1602.08191*, 2016.
[24] Imagenet large scale visual recognition challenge 2012. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2012/