2013 IEEE Eighth International Conference on Networking, Architecture and Storage

PEAM: Predictive Energy-Aware Management for Storage Systems

Xunfei Jiang*,Ji Zhang*, Mohammed I. Alghamdi†, Xiao Qin*, Minghua Jiang‡, and Jifu Zhang§

*Department of Computer Science and Software Engineering, Auburn University, Auburn, AL 36849-5347
{xzj0009, jzz0014, xqin}@auburn.edu
†Department of Computer Science, Al-Baha University, Al-Baha City, Kingdom of Saudi Arabia, mialmushilah@bu.edu.sa
‡College of Mathematics and Computer Science, Wuhan Textile University, Wuhan 430073, China
§School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan 030024, China

***Abstract*—This paper presents a novel Predictive Energy-Aware Management (PEAM) system that is able to reduce the energy costs of storage systems by appropriately selecting data transmission methods. In particular, we evaluate the energy costs of three methods (1. transfer data without archiving and compression; 2. archive and transfer data; 3. compress and transfer data) in preliminary experiments. According to the results, we observe that the energy consumption of data transmission greatly varies case by case. We cannot simply apply one method in all cases. Therefore, we design an energy prediction model that can estimate the total energy cost of data transmission by using particular transmission methods. Based on the model, our predictive energy-aware management system can automatically select the most energy efficient method for data transmission. Our experimental results show that our system performs better than simply selecting any one among the three methods for data transmission in terms of energy efficiency.**

***Keywords*-predictive; energy-aware; storage system;**

I. Introduction

Due to the rapid growth of data volume in data centers, efficiently managing a massive amount of data becomes a challenging problem. For example, Facebook maintains over 260 billion images (20 petabytes of data) in their distributed storage systems. There are one billion photos (around 60 terabytes) uploaded by users each week [6]. Data-management mechanisms issue big data operations to optimize file placement and achieve improved I/O performance in data centers. These big data operations inevitably introduce extra performance and energy overheads due to frequent data movement among servers and storage systems. For example, in the Google file system, data would be moved back and forth among storage nodes to keep workload and disk space balanced across the storage system [10]. To offer good reliability, storage systems maintain one or more replicas for each file. Upon the arrival of new data, the storage systems create replicas on multiple storage nodes (see, for example, GFS [10] and HDFS [24]). To enhance energy efficiency of data centers, one may have hot data migrated to a portion of fast storage nodes that continuously provide services, while turning other nodes archiving cold data into the standby mode [20][17].

An increasing number of energy conservation techniques have been proposed to reduce the surprisingly high energy costs of data centers. According to a report [5], data centers contribute to nearly 1.5% (i.e., 4.5 billion dollars in 2006) of total electricity consumption in the U.S.. Moreover, the electricity costs contribute to a large portion of energy consumption in data centers [12]. Even worse, many data centers are rapidly growing in storage capacity to meet the needs of big data newly collected on daily basis.

Designing energy-efficient data-movement policies are crucial for the next-generation storage systems. Our study in this paper is motivated by the following three factors.

1) the lack of study on the impacts of data movement among servers on energy consumption of cluster storage systems,
2) the capability of estimating the total energy cost of data movement over network in data centers, and
3) the possibility of reducing the energy costs of data-movement schemes by selecting an appropriate data-movement policy.

In this paper, we first conduct a preliminary experiment, where three data-movement policies are applied to transfer real-world data sets between two servers. We observe that the total energy consumption of data movement significantly varies case by case in the experiment. Unfortunately, none of the investigated policies can achieve the best energy efficiency among all the tested cases. For example, directly transferring a large number of small files over the network is an inefficient method. Network latencies lead to noticeable delays during the data-movement process, thereby increasing the energy costs of transferring the massive amount of data. On the other hand, archiving or compressing a large data set before a transmission induces extra CPU energy consumption.

The preliminary findings motivate us to develop a predictive energy-aware management system called PEAM. Under dynamically changing workload conditions, PEAM aims to intelligently select the most appropriate data-movement policy based on predicted energy costs. At the heart of PEAM

978-0-7695-5034-3/13 $26.00 © 2013 IEEE
DOI 10.1109/NAS.2013.20

is our proposed energy prediction model, which estimates the energy costs of specific data transmission methods. The energy prediction model integrates our new performance model with the recently developed energy/thermal models in the context of storage systems. We evaluate the total energy costs of all methods on two larger real datasets. The experimental results show evidence that PEAM makes accurate decisions on selecting the best data-transmission method to improve energy efficiency. We also demonstrate that PEAM outperforms the existing solutions when it comes to energy efficiency in data storage systems.

Organization. The rest of this paper is organized as follows. The next section presents prior studies and related research issues. Section III describes our preliminary experiments and observations. In Section IV, we propose a predictive energy-aware management for storage systems. The experimental results are shown in Section V. Finally, Section VI concludes the paper.

II. Relative work

A. Thermal-aware Resource Management Strategies

Improving data-center energy efficiency becomes increasingly important. Techniques reducing energy consumption of computing facilities and cooling systems make a major contribution to advance energy-efficient data centers. For example, thermal-aware resource management strategies reorganize data or redistribute workloads to achieve balanced temperature distribution among data nodes.

Sharma *et al.* designed a $thermal\text{-}load\text{-}balancing$ framework, in which local and regional policies are applied to dynamically distribute the workload across servers in a data center to reduce energy consumption [23]. Their simulation results show that equipment reliability can be improved by placing an asymmetric workload and uniformly distributing temperature in data centers. Tang *et al.* proposed an optimal recirculation process in homogeneous data centers [27]. A thermal-aware task scheduling algorithm, XInt, is able to minimize recirculation costs by balancing the workload within a data center. Tang *et al.* discovered that cooling costs highly depend on peak inlet temperatures [28]. In order to lower cooling power, Tang *et al.* designed a task assignment policy, MPIT-TA, which reduces peak inlet temperatures. Their simulation results show that MPIT-TA saves at least 20% of cooling energy.

A handful of studies were focused on temperature-aware load balancing strategies [21][22]. In these studies, a customized threshold is set to limit CPU temperatures, thereby conserving CPU energy consumption. If the CPU temperatures exceed the threshold, the CPU's voltage and frequency are dynamically adjusted at the cost of increased execution times.

B. Thermal Models

After investigating IBM's 5-1/4-in fixed disk drives, Eibeck *et al.* proposed a thermal model that predicts the transient temperatures of disk drives [9]. Tan *et al.* created a three-dimensional transient temperature model used to evaluate disk temperatures when frequent seeking operations are performed [26]. By considering five components (internal drive air, spindle motor, the base and cover of the disk, the voice-coil motor, and disk arms), Gurumurthi *et al.* built a comprehensive model that calculates the thermal behaviour of a hard disk [11]. Their findings show that heat generated by the components make contributions on disk temperatures. Kim *et al.* investigated impacts of seek times on disk temperatures [15]; they studied the thermal behaviors of disks by varying platter types and the number of platters.

Lin *et al.* proposed approaches to coordinating processors and memory to improve system performance and/or power efficiency during memory thermal emergency [16]. They designed the adaptive core gating (DTM-ACG) and coordinated DVFS (DTM-CDVFS) schemes as well as a thermal model to predict DRAM temperatures. Their experiments conducted on real platforms show that the two schemes exhibit 6.7% and 15.3% of improvements in terms of performance.

C. Compression Methods

Data compression techniques have been widely applied to achieve good space efficiency in storage systems and to shorten data retrieval time. The compression techniques are able to reduce data sizes; however, the existing techniques introduce extra CPU overhead. Compression ratios of a certain method can vary greatly for different file types.

Cannane and Williams proposed a semi-static phrase-based scheme called XRAY [7]. An offline model was first built by training samples selected from data collection. Then, the entire collection can be compressed online in a single pass. The experimental results show that their method performs very well for large general-purpose collection compression, especially in the case when an individual record or document is required to be decompressed.

Reetuparna *et al.* explored the performance and energy behaviours of data compression on Network-on-Chip (NoC) [8]. Two configurations examined in their study include Cache Compression (CC) and Compression in the Network Interface Controller (NIC). Decompression latency can be hidden by overlapping with NoC communication latency. The simulation results show that the compression-on-NoC method achieves energy savings by 20%.

D. Predictive Thermal Management

Srinivasan and Adve demonstrated a performance-effective Dynamic Thermal Management (DTM) for multimedia applications [25]. In their study, a predictive DTM

algorithm was developed to efficiently use response mechanisms. The experimental results confirm that the DTM algorithm performs significantly better than the existing reactive DTM algorithms.

Ramos and Bianchini built a software structure for Internet services (C-Oracle), in which the best reaction is selected by predicting and evaluating temperature and performance impacts of various thermal management reactions [19]. C-Oracle effectively manages thermal emergencies without unnecessary performance degradation.

The impact of data movement on energy efficiency of storage systems has not been fully explored in the aforementioned technologies. Keeping workload balanced and uniform temperature distribution across servers in a data center lead to frequent data migrations, which in turn give rise to increased energy costs caused by data reads, writes, and transmission over network interconnections. To significantly reduce energy consumption incurred by data migrations, we are motivated to design and investigate thermal-aware data migrations – an open issue that is currently unresolved. Appropriately and dynamically selecting an energy-efficient data-migration policy can potentially reduce the overall energy and cooling costs in data centers. In this paper, we propose a predictive thermal management strategy that judiciously makes the best data-migration decisions by predicting thermal and energy impacts of each data migration.

III. Preliminary Results

To characterize the overall energy cost of data transmissions over network interconnections, we start this study by investigating the performance and thermal behaviours of various data transmission strategies. In this section, we first describe a testbed and three data transmission methods used in our preliminary experiments. Next, we conduct the experiments on two real datasets and illustrate thermal impacts made by these three strategies. Finally, we demonstrate the motivation of our predictive energy-aware management for storage systems.

A. Testbed

The testbed consists of two Linux servers connected through the fast Ethernet. Table I summarizes the configuration details of the servers performing as nodes of a storage cluster. In the experiment, CPU and disk temperatures are collected from embedded device sensors. The inlet and outlet temperatures of the storage nodes are monitored by four sensors attached to the nodes.

We transfer two real-world datasets between the two storage nodes, the results of which are presented in the following two subsections. Three data-transmission strategies examined in this preliminary experiment are listed below.

- Method 1: **Direct Transmission (DT)**. Transfer data over the network without any data archiving and compression.

Table I Testbed Configurations

	Node 1	**Node 2**
CPU	Intel(R) Celeron(R) 450@2.2GHz	
Network	1 GigaBit Ethernet network card	
Disk	WD-500GB Sata disk([3])	WD-160GB Sata disk([2])
Operating System	Ubuntu 10.04(lucid) Linux kernel 2.6.32-43	Ubuntu 10.04(lucid) Linux kernel 2.6.32-38

- Method 2: **Archived Transmission (AT)**. Data is archived before transmission over network. This strategy reduces overheads (e.g., network latencies) incurred by transferring a large number of small files.
- Method 3: **Compressed Transmission (CT)**. Data is compressed before transmissions. This method performs very well if a high compression ratio can be achieved and the compression process is completed in a short time period.

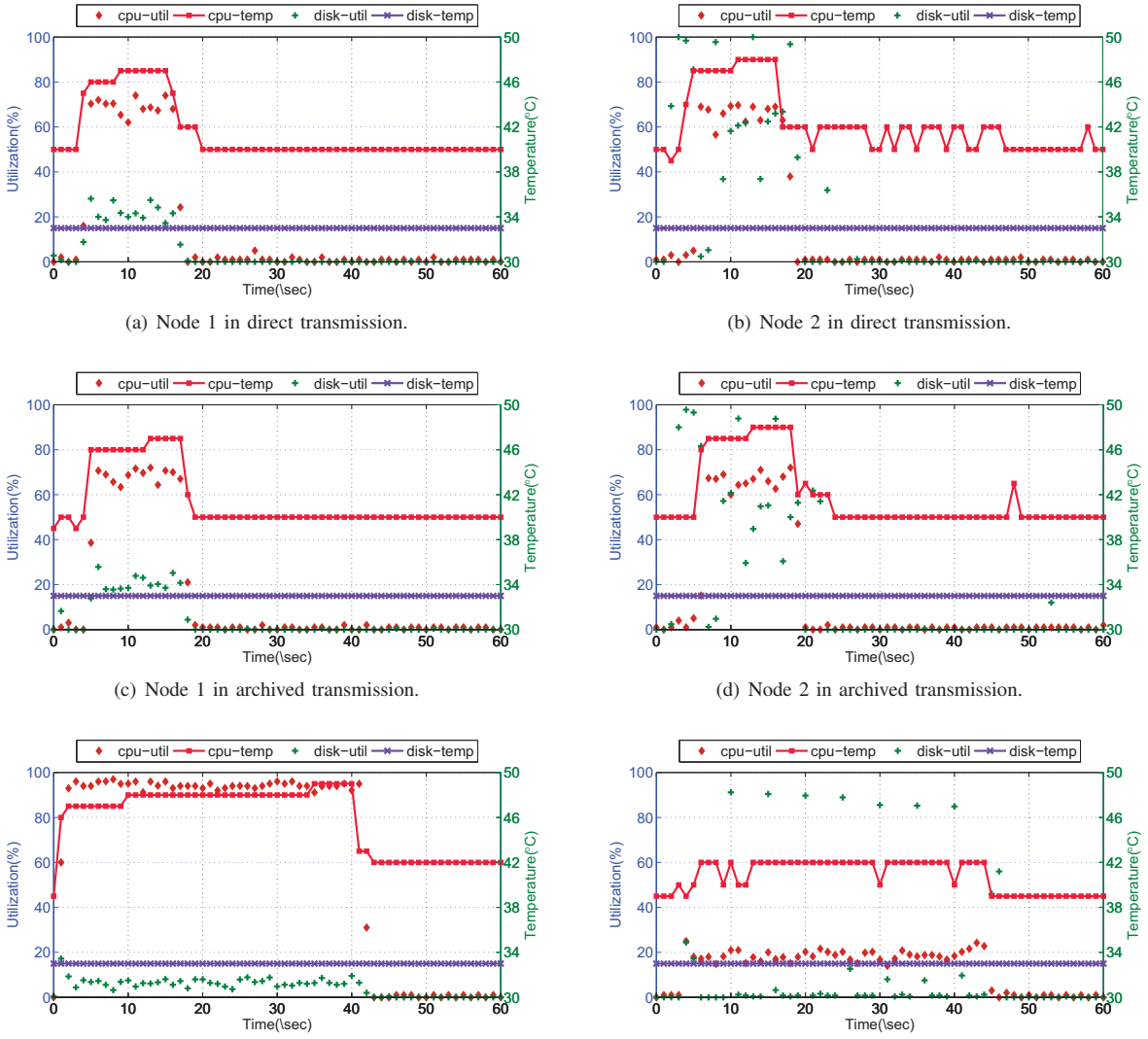
B. Transferring A Single Text File

In the first experiment, we apply the above three methods to transfer a single text file of 507.7 MB from node 1 to node 2.

Fig. 1 displays the temperature and utilization of CPUs and disks during the data transmission of a large text file. We observe that the execution times of DT and AT are very close; however, CT is an outlier doubling the execution time of both DT and AT. Regardless of the methods, CPU temperatures significantly increase, whereas disk temperatures stay unchanged. Constant disk temperatures are reasonable because disks have relatively longer heat-up periods (i.e., 30 minutes) [13]. Staying in the active state for a short period (e.g., less than one minute) has no significant impact on the disk temperature.

Figs. 1(a), 1(c), and 1(e) show that node 1's CPU utilization and temperature goes up rapidly, whereas disk utilization remains at a low level. The CT scheme gives rise to extremely high CPU utilization because the compression process is very computation intensive. On the other hand, CT's disk utilization is simply half of those of the other two methods. DT and AT have similar CPU and disk utilizations. Figs. 1(b), 1(d), and 1(f)) reveal that node 2's CPU utilization is close to that of node 1 under the DT and AT cases, except that node 2's CPU utilization is only one fifth of that of node 1 in the CT case. Thus, the CPU temperature of node 2 under DT is also lower than those of the same node under the other two methods. For all the three strategies, node 2 has lower disk utilization than node 1.

Table II summarizes the execution times and file size, as well as compression ratios. In this table, N1 and N2 represent node 1 and node 2, respectively. CT enjoys a compression ratio of 21.9%; data is not compressed in the other two methods. DT exhibits the shortest execution time



(a) Node 1 in direct transmission.

(b) Node 2 in direct transmission.

(c) Node 1 in archived transmission.

(d) Node 2 in archived transmission.

(e) Node 1 in compressed transmission.

(f) Node 2 in compressed transmission.

Figure 1: Performance of 1 text file transmission.

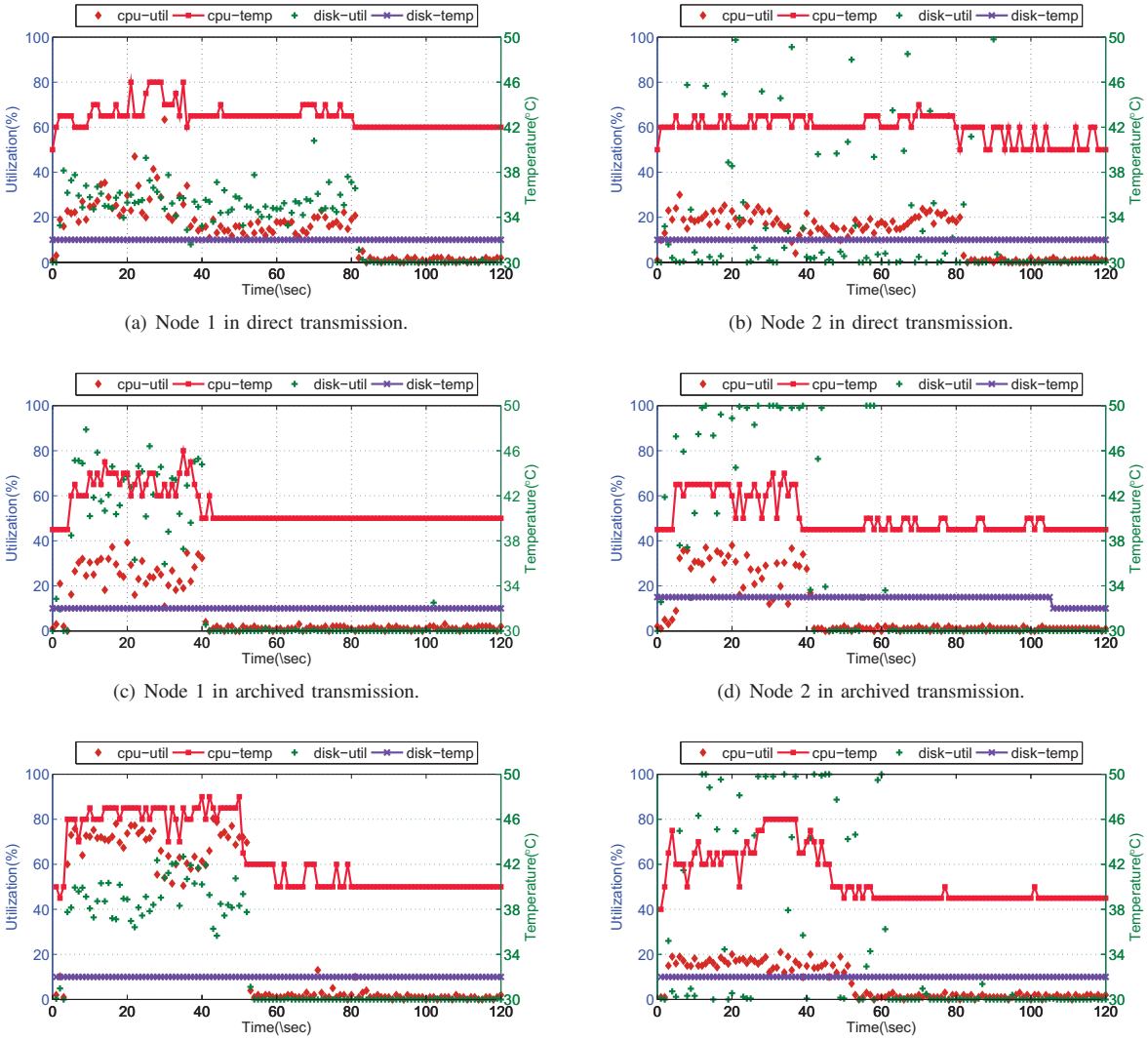
Table II Summary of single text file transmission.

Methods	**DT**		**AT**		**CT**	
	N1	**N2**	**N1**	**N2**	**N1**	**N2**
Execution Time(s)	17	17	18	20	42	47
AVG U_{CPU}(%)	65.7	63.9	63.0	61.5	93.4	17.9
AVG U_{Disk}(%)	20.3	65.0	19.3	55.9	6.8	19.0
MAX T_{CPU}(°C)	47	48	47	48	49	43
MAX T_{Disk}(°C)	33	33	33	33	33	33
Data Transferred(MB)	507.7		507.7		111.2	
Compression Ratio(%)	100		100		21.9	
Total Energy Cost(J)	4036.9		4459.2		9952.8	

among the three test strategies.

The temperatures and utilizations of CPU and disks are summarized at the bottom of Table II. We observe that CT suffers from the highest CPU utilization on node 1 due to compression overhead, whereas in node 2, CPU utilization is lower than those in the other two methods. The peak CPU temperature of node 2 under the CT method is the lowest among all the methods. The first two methods share similar thermal impact on the two nodes. By comparing the overall energy cost of these three methods, we observe that DT is the most energy-efficient approach.

In short, we conclude that the archiving and compression process leads to high CPU temperature and utilization, which in turn have noticeable impact on the total energy cost in storage systems.



(a) Node 1 in direct transmission.

(b) Node 2 in direct transmission.

(c) Node 1 in archived transmission.

(d) Node 2 in archived transmission.

(e) Node 1 in compressed transmission.

(f) Node 2 in compressed transmission.

Figure 2: Performance of transferring Linux kernel files.

C. Transferring Source Code Files

We evaluate a second case where Linux source code files are transferred between two storage nodes. Fig. 2 reveals temperatures and utilizations of CPUs and disks where the three data transfer strategies are adopted. The detailed results are summarized in Table III.

We observe from the table that AT achieves the best performance in terms of execution time. The CT scheme only transfers 103.8 MB of data, which is 23% of the original data size, over the network. However, CT does not exhibit the shortest transmission time due to extra overhead caused by data compression and decompression. When it comes to the AT method, even the size of data transferred over the network is larger than that of DT; the transmission time of

Table III Empirical results of transferring Linux source code files.

Methods	**DT**		**AT**		**CT**	
	N1	**N2**	**N1**	**N2**	**N1**	**N2**
Execution time(s)	81	90	40	60	49	57
AVG U_{CPU}(%)	20.8	16.1	24.2	17.4	68.6	15.7
AVG U_{Disk}(%)	27.4	23.0	56.7	69.1	45.9	61.5
MAX T_{CPU}(°C)	46	44	46	44	48	46
MAX T_{Disk}(°C)	33	32	32	33	33	32
Data Transferred(MB)	454.8		475.8		103.8	
Compression Ratio(%)	100		100		23	
Total Energy Cost(J)	16164		15938		16718	

DT is much shorter than that of AT. This performance trend is reasonable because the Linux kernel package contains a large number (i.e., 40,927) of small files. Transferring these small files one by one takes a long time due to network latencies. Merging small files into a single large file helps to reduce the network overhead.

Like findings obtained from the first experiment, the compression process results in the highest CPU temperature and utilization in the case of CT. Although the peak disk temperature is different from that observed in the first experiment, the peak temperature remains unchanged in all the methods during the execution period (see Fig. 2). From the thermal behaviour's perspective, DT and AT are more thermal friendly than CT. From the energy's perspective, AT consumes less energy than the other two strategies.

D. Motivation of the Predictive Thermal Management

The above preliminary findings suggest that it is challenging to accurately estimate energy costs of data transmissions due to the following three reasons. First, the total energy cost (including computing and cooling costs) caused by data transmissions depends on CPU and disk temperatures, transmission times, and compression ratios. Second, there is a lack of energy-efficient data-transfer strategies that can fit the needs of a wide range of cases. The DT scheme can energy efficiently transfer a single large text file (see Section III-B); whereas AT is the most energy-efficient strategy to transfer a large number of small files (see Section III-C). The impact of data compression on energy consumption largely relies on the features of files being transferred. Third, data transmissions occur frequently in cluster storage systems. It is impractical to manually choose the best data-transfer strategy in a dynamic computing environment, where the features of transferred files are continually changing. To address this problem, we design a predictive energy-aware management system or PEAM. There are two phases incorporated in PEAM. The first phase is to predict energy consumption incurred by executing each candidate data-transfer strategy. Predictions are obtained by comprehensively considering compression ratios, transmission times, file types, and data sizes. The second phase is a straightforward selection made by comparing the predicted energy costs induced by the candidate strategies. The details on PEAM are illustrated in the next section.

IV. Design

Motivated by the preliminary results, we propose a predictive energy-aware management system called PEAM. Three modules integrated in PEAM are an energy predictor, a method selector, and a data-transmission monitor. The energy predictor module predicts the energy consumption of data transmissions carried out by a particular method. The method selector chooses the best data-transfer strategy based on energy predictor's estimates. The monitor keeps track of data transmissions. Before a data transfer is initiated, the module sends a request to the method selector; the data start being transferred after a feedback is received from the selector.

A. The Framework of PEAM

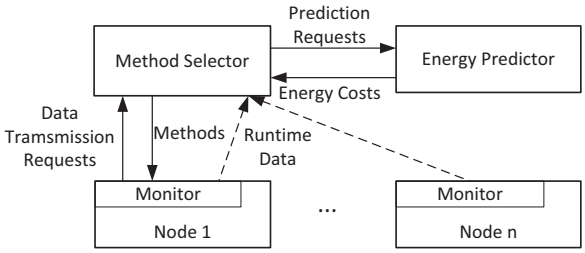


Figure 3: The framework of the predictive energy-aware management (PEAM).

Fig. 3 displays a cluster equipped with n storage nodes. Our predictive energy-aware management system – PEAM – runs on each node. The monitor module gathers runtime information related to data transmissions, file metadata, and storage nodes (e.g., temperatures and utilizations). When a data transmission is detected, the module sends a request to the method selector, which makes a decision on the most energy-efficient data-transfer strategy.

The Method Selector not only maintains candidate data-transfer strategies, but it also judiciously chooses the best strategy to reduce energy consumption. Fig. 3 shows that upon the arrival of a data-transmission request, the Method Selector forwards the request along with all the candidate strategies to the energy predictor. According to an energy estimate offered by the predictor, the Method Selector notifies the monitor module of a candidate strategy that will cause the lowest energy cost to transfer the data.

The Energy Predictor, shown in Fig. 4, provides the energy estimates of data transmissions handled by a particular strategy. In the PEAM system, the predictor is focused on the overall energy consumption of a cluster storage system. Thus, the overall energy cost includes both computing energy cost and cooling cost. We build a performance model to quantify CPUs and disks utilization as well as data transmission time. With these information in place, the computing energy cost can be computed by a computing model (see (7)). Moreover, the cooling cost can be calculated by integrating a thermal model and the coefficient of performance model (a.k.a., COP).

B. The Energy Predictor Model

The Energy Predictor model consists of the following sub-models.

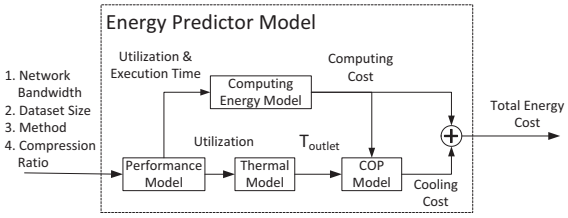


Figure 4: Framework of the Energy Predictor module. (COP: Coefficient of Performance)

1) Performance Model: The performance model derives CPU/disk utilization and data-transmission time from the information provided by prediction requests; such information includes network bandwidth, dataset size, data transmission methods, and compression ratios. Compression schemes and their compression ratios for given file types are maintained in the model as a static data structure. The execution time of a data-transmission process is made up of data transmission time and compression/decompression time if it is applicable. The compression/decompression time is determined by data size and compression methods. If a data-transmission strategy does not apply data compression techniques, the compression/decompression time should be ignored. Obviously, data compression overhead might be offset by time saved in transferring data over the network.

The utilization of CPUs and disks can be derived as a function of $Method$ (i.e., a data-transfer method) and $R_{compression}$ (i.e., compression ratio). Thus, we have

$$U_{CPU} = g(Method, R_{compression}), \tag{1}$$

$$U_{disk} = h(Method, R_{compression}), \tag{2}$$

where U_{CPU} and U_{disk} are average CPU and disk utilizations.

We express the execution time of a data-transmission process as:

$$\begin{aligned} T_{execution} =& k(size, Method, R_{compression}, Bandwidth) \\ =& T_{read} + T^{Method}_{pre-proc} + T_{send} \\ &+ T_{receive} + T^{Method}_{after-proc} + T_{write} \end{aligned} \tag{3}$$

where $size$, $R_{compression}$, and $Bandwidth$ denote the data size, compression ratio, and network bandwidth. $T_{execution}$ is the execution time if $Method$ is applied to transfer the data. T_{read} is the time spent in reading the original file to cache on the source node, and T_{read} depends on the $size$ value. $T^{Method}_{pre-proc}$ is the time of pre-processing the data with a specific method; for example, with the DT method, the data should be compressed in the source node's cache. $T^{Method}_{after-proc}$ is the time of processing the transferred data (e.g., decompression). T_{send} and $T_{receive}$ are sending and receiving times of the data delivered over the network; T_{send} and $T_{receive}$ are affected by $Bandwidth$ and $R_{compression}$. T_{write} is the time spent in writing the received data to a destination disk.

2) Thermal Model: The thermal model estimates outlet temperatures of a storage node based on its CPU and disk utilizations. CPU temperatures, which are sensitive to CPU utilization, can be expressed as:

$$T_{CPU}(t) = f_{CPU}(T^{CPU}_i, T_A, U_{CPU}, t), \tag{4}$$

where T^{CPU}_i and T_A denote initial CPU temperature and ambient temperature. U_{CPU} represents CPU utilization, and t is the CPU running time under a specific utilization.

Differing from CPU temperatures, disk temperatures are not noticeably sensitive to disk utilizations during a short period of time. However, if a disk is active for a longer period, the disk's temperature is affected by its utilization [14]. The disk temperature can be modelled as:

$$T_{disk}(t) = f_{disk}(T^{disk}_i, T_A, U_{disk}, t), \tag{5}$$

where T^{disk}_i and T_A are initial disk temperature and ambient temperature. U_{disk} represents disk utilization. t is the time that disk works in active state.

Since CPU and disk are two major contributors to outlet temperatures of storage nodes, we use the following outlet temperature model to quantify the thermal impact of CPU and disk activities on outlet temperatures (see [14] for the details of the model).

$$T_{outlet} = T_{inlet} + \alpha * T_{CPU} + \beta * T_{disk} + \gamma, \tag{6}$$

where T_{inlet} and T_{outlet} are the inlet and outlet temperatures of a storage node. α is the thermal impact from CPU temperatures, β is the impact from disk temperatures, and γ represents the impact of other components on the outlet temperature.

3) Computing Energy Power Model: We use (7) to calculate the computing energy power, where P_i is the power of a component that is sitting idly, $U_{component}$ refers to the utilization of the component in storage nodes. $P^{max}_{component}$ and $P^{idle}_{component}$ are the power when the component works in full capacity and is in the idle state, respectively.

$$P_C = P_i + \Sigma(U_{component} * (P^{max}_{component} - P^{idle}_{component})) \tag{7}$$

4) COP: A Cooling Power Model: Fig. 5 shows that COP values increase as the supply temperature of computer room air conditioning (CRAC) goes up [18]. A large COP value indicates a high energy efficiency of a storage system.

$$COP(T) \;=\; 0.0068 * T^2 + 0.0008 * T + 0.458 \tag{8}$$

Equation (8) defines COP as a ratio of removed heat to the energy cost of a cooling system for heat removal [18]. The supply temperature of CRAC (i.e., Computer Room Air Conditioning) is denoted by T. The cooling cost is inversely proportional to the COP value.

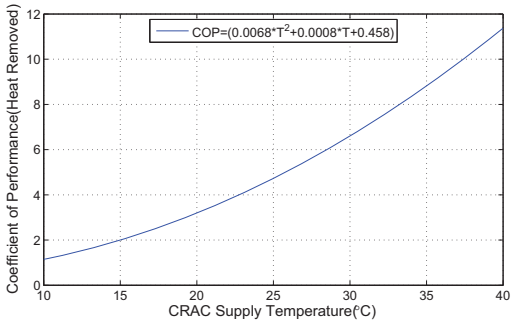


Figure 5: Coefficient of the performance curve for the chilled-water CRAC units at the HP Labs Utility Data Center [18]

The cooling power P_{AC} can be derived from COP using (9).

$$P_{AC} = \frac{P_C}{COP(T)}, \tag{9}$$

where P_C is the computing energy power.

With the computing and cooling power in place, we can express the overall power as:

$$P_{Total} = P_C + P_{AC}, \tag{10}$$

V. Experiments

In this section, we compare our PEAM with the three baseline solutions using two real-world datasets. For each dataset, we evaluate execution time and energy consumption caused by transferring the data from one node to another within a storage cluster. The testbed description can be found in Section III.

A. Datasets

The first tested 60 GB dataset is the Human Genome sequences, which is available from NIH's (National Institutes of Health) NCBI website[1]. Each large sequence file contains the DNA sequence of an entire chromosome. These DNA sequences are widely used in the bioinformatics research (e.g., sequence alignments and gene predictions). The second 50 GB dataset is comprised of millions of songs[2] archived in a multimedia storage system. Maintaining a large number of multimedia files becomes increasingly challenging in terms of improving performance and energy efficiency of storage systems. For example, five million photos are uploaded to instagram every day [1]; Youtube receives 48 hours of new video every minute [4].

B. Human Genome Dataset

Fig. 6 displays the transmission time and total energy cost of transferring the Human Genome data using the four strategies. CT takes much longer than the other three methods because compressing and decompressing a large dataset is computationally expensive. DT and AT perform very similarly. Transferring data over the network takes most of the time, which depends on data size and network bandwidth. More importantly, AT is 15 seconds slower than DT due to the data-archiving overhead. DT does not need to repeatedly build the network connections. The performance of our PEAM is close to that of DT. PEAM's only extra overhead is induced selecting transfer strategy. We ignore this overhead in our experiments because it is too small (i.e., less than 10 ms) compared with total execution times. Our evidence shows that the computational overhead of the energy predictor and method selector module in PEAM is as low as a few milliseconds.

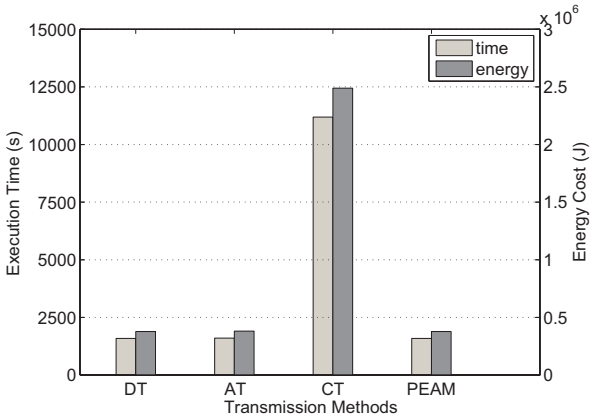


Figure 6: Transmission time and total energy cost of transferring the Human Genome dataset.

Not surprisingly, CT spends much more energy transferring the Genome data than the other strategies because both CPU and disks take a longer time period to complete the data transmission task. Compared with AT, DT saves 2964 Joule energy during the transmission process. AT is not energy-efficient because AT takes an extra 15 seconds on data archiving and de-archiving, thereby leading CPUs and disks active for a longer time span. Thus, extra energy is consumed by the CPUs and disks. Due to the archiving and de-archiving time, the CPU temperature increases, which in turn causes additional energy costs for the cooling system. The energy consumption of PEAM is very close to that of DT because PEAM only suffers from negligible overhead.

C. Multimedia Dataset

Fig. 7 plots the transmission time and total energy cost of moving the real-world multimedia dataset by applying the four strategies. Similarly, the transmission time of CT is the highest due to the compression process. Interestingly, DT is 458 seconds slower than AT because the multimedia dataset contains millions of small files. When it comes to the

[1] ftp://ftp.ncbi.nih.gov/genomes/H_sapiens
[2] http://www.infochimps.com/collections/million-songs

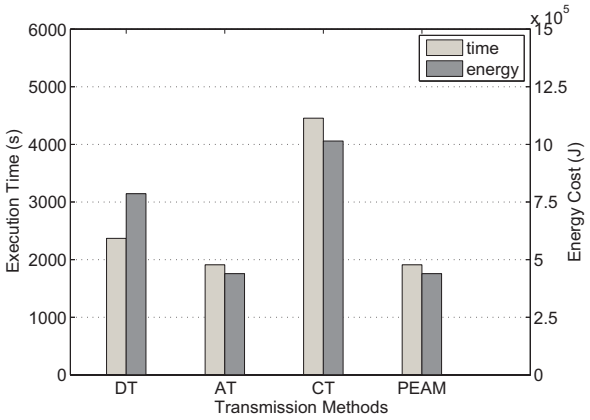


Figure 7: Execution time and total energy cost of moving multimedia dataset.

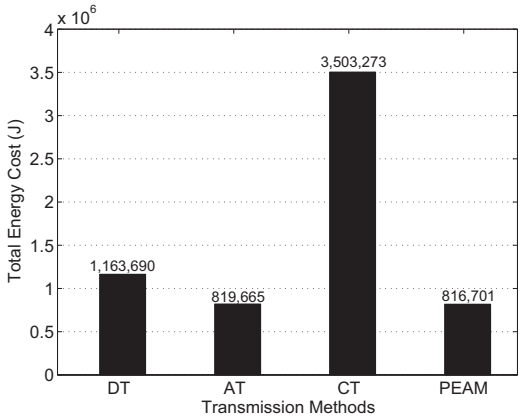


Figure 8: Overall energy cost of moving two datasets.

DT scheme, network latency becomes a major performance bottleneck in data transmission. The network connections must be created for each small file; such overhead becomes considerably expensive when the number of small files is excessively large. Thus, even with the overhead of data archiving and de-archiving, AT performs better than DT in terms of transmission time.

Similar to the CT tested in the Human Genome case, CT in the multimedia case exhibits the lowest energy efficiency among all the strategies due to its long transmission times. DT consumes much more energy than AT because of DT's long transmission times. PEAM accurately predicts that AT is the most energy-efficient and selects AT to transfer the dataset. Hence, both the transmission time and energy cost of PEAM is close to the AT scheme. The overhead of PEAM is proved to be negligible.

D. Overall Evaluation

Using two real-world datasets, we demonstrate that PEAM can accurately and quickly predict the performance of the candidate data-transfer strategies. We assume that one method is selected for data transmissions in the tested storage system. Fig. 8 shows that CT's performance is the worst among the four methods. In addition, DT is energy efficient when it is applied to transfer the Human Genome dataset; DT consumes more energy while moving the multimedia dataset. PEAM is able to accurately and automatically predict and select the best method, and, therefore, compared with DT, AT, and CT, PEAM reduces the energy consumption by 346,989 J, 2,964 J, and 2,686,572 J, respectively.

VI. Conclusion

Surprisingly high energy consumption of data centers makes it demanding to improve energy efficiency of large-scale storage systems. In modern data centers, data management introduces big data operations to achieve high I/O performance by judiciously placing files. Big data operations can incur both performance and energy overheads due to frequent data movement. We aim to reduce the energy costs of data centers by offering an energy-aware data management strategy to improve energy efficiency of data storage systems.

In this paper, we first characterized the thermal and performance behaviours of three data transmission methods. The preliminary findings motivate us to develop a novel predictive energy-aware management system called PEAM, which is capable of dynamically choosing the most appropriate data transmission method to reduce energy consumption caused by large data transfers among storage nodes. Energy estimates are calculated by a validated model, which integrates our new performance model with the recently designed energy/thermal models. Our experimental results showed PEAM makes accurate decisions on selecting the most energy-efficient data transmission method. The results also demonstrated that PEAM can significantly improve energy efficiency of large-scale storage systems in data centers.

Acknowledgment

This research was supported by the U.S. National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0917137 (CSR), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), CNS-0855251 (CRI), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS). Mohammed Alghamdi's research was supported by AL-Baha University.

References

[1] Instagram statistics. http://www.mediabistro.com/alltwitter/social-media-stats-2012_b30651.

[2] Wd1600aajs specification. http://www.wdc.com/wdproducts/library/SpecSheet/ENG/2879-701277.pdf.

[3] Wd5000aaks specification. http://www.wdc.com/wdproducts/library/SpecSheet/ENG/2879-701277.pdf.

[4] Youtube statistics. http://www.youtube.com/t/faq.

[5] U. E. P. Agency. Report to congress on server and data center energy efficiency. Technical report, August 2007.

[6] D. Beaver, S. Kumar, H. C. Li, J. Sobel, and P. Vajgel. Finding a needle in Haystack: facebook's photo storage. In *Proceedings of the 9th USENIX conference on Operating systems design and implementation*, OSDI'10, pages 1–8, Berkeley, CA, USA, 2010. USENIX Association.

[7] A. Cannane and H. E. Williams. A general-purpose compression scheme for large collections. *ACM Trans. Inf. Syst.*, 20(3):329–355, July 2002.

[8] R. Das, A. Mishra, C. Nicopoulos, D. Park, V. Narayanan, R. Iyer, M. Yousif, and C. Das. Performance and power optimization through data compression in network-on-chip architectures. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pages 215 –225, feb. 2008.

[9] P. Eibeck and D. Cohen. Modeling thermal characteristics of a fixed disk drive. *Components, Hybrids, and Manufacturing Technology, IEEE Transactions on*, 11(4):566 –570, dec 1988.

[10] S. Ghemawat, H. Gobioff, and S.-T. Leung. The Google file system. In *Proceedings of the nineteenth ACM symposium on Operating systems principles*, SOSP '03, pages 29–43, New York, NY, USA, 2003.

[11] S. Gurumurthi, A. Sivasubramaniam, and V. K. Natarajan. Disk drive roadmap from the thermal perspective: A case for dynamic thermal management. *SIGARCH Comput. Archit. News*, 33(2):38–49, May 2005.

[12] http://www.datacenterdynamics.com/research/energy-demand 2011-12. Global data center energy demand forecasting. Technical report, DatacenterDynamics, 2011.

[13] X.-F. Jiang, M. I. Alghamdi, J. Zhang, M. A. Assaf, X.-J. Ruan, T. Muzaffar, and X. Qin. Thermal modeling and analysis of storage systems. In *Proc. the 31st IEEE Int'l Performance Computing and Communications Conf*, 2012.

[14] X.-F. Jiang, J. Zhang, M. I. Alghamdi, M. A. Assaf, X.-J. Ruan, T. Muzaffar, and X. Qin. Thermal modeling of hybrid storage clusters. *Journal of Signal Processing Systems*, 2013, in press.

[15] Y. Kim, S. Gurumurthi, and A. Sivasubramaniam. Understanding the performance-temperature interactions in disk i/o of server workloads. In *High-Performance Computer Architecture, 2006. The Twelfth International Symposium on*, pages 176 –186, feb. 2006.

[16] J. Lin, H. Zheng, Z. Zhu, and Z. Zhang. Thermal modeling and management of dram systems. *IEEE Transactions on Computers*, 99(PrePrints), 2012.

[17] A. Manzanares, X. Qin, X. Ruan, and S. Yin. Pre-bud: Prefetching for energy-efficient parallel i/o systems with buffer disks. *ACM Transactions on Storage (TOS)*, 7(1):3, 2011.

[18] J. Moore, J. Chase, P. Ranganathan, and R. Sharma. Making scheduling "cool": temperature-aware workload placement in data centers. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC '05, pages 5–5, Berkeley, CA, USA, 2005. USENIX Association.

[19] L. Ramos and R. Bianchini. C-oracle: Predictive thermal management for data centers. In *High Performance Computer Architecture, 2008. HPCA 2008. IEEE 14th International Symposium on*, pages 111 –122, feb. 2008.

[20] X. Ruan, S. Yin, A. Manzanares, J. Xie, Z. Ding, J. Majors, and X. Qin. ECOS: An energy-efficient cluster storage system. In *Performance Computing and Communications Conference (IPCCC), 2009 IEEE 28th International*, pages 79 –86, dec. 2009.

[21] O. Sarood, A. Gupta, and L. Kale. Temperature aware load balancing for parallel applications: Preliminary work. In *Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on*, pages 796 –803, may 2011.

[22] O. Sarood and L. V. Kale. A 'cool' load balancer for parallel applications. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 21:1–21:11, New York, NY, USA, 2011. ACM.

[23] R. Sharma, C. Bash, C. Patel, R. Friedrich, and J. Chase. Balance of power: dynamic thermal management for internet data centers. *Internet Computing, IEEE*, 9(1):42 – 49, jan.-feb. 2005.

[24] K. Shvachko, H. Kuang, S. Radia, and R. Chansler. The Hadoop Distributed File System. In *Mass Storage Systems and Technologies (MSST), 2010 IEEE 26th Symposium on*, pages 1–10, may 2010.

[25] J. Srinivasan and S. V. Adve. Predictive dynamic thermal management for multimedia applications. In *Proceedings of the 17th annual international conference on Supercomputing*, ICS '03, pages 109–120, New York, NY, USA, 2003. ACM.

[26] C. Tan, J. Yang, J. Mou, and E. Ong. Three dimensional finite element model for transient temperature prediction in hard disk drive. In *Magnetic Recording Conference, 2009. APMRC '09. Asia-Pacific*, pages 1 –2, jan. 2009.

[27] Q. Tang, S. Gupta, and G. Varsamopoulos. Thermal-aware task scheduling for data centers through minimizing heat recirculation. In *Cluster Computing, 2007 IEEE International Conference on*, pages 129 –138, sept. 2007.

[28] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos. Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. *IEEE Trans. Parallel Distrib. Syst.*, 19(11):1458–1472, Nov. 2008.