Thermal Modeling and Management of Storage Systems

by

Xunfei Jiang

A dissertation submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Auburn, Alabama
August 2, 2014

Keywords: Thermal, Modeling, Management, Storage Systems

Approved by

Xiao Qin, Associate Professor of Computer Science and Software Engineering
Cheryl Seals, Associate Professor of Computer Science and Software Engineering
David Umphress, Associate Professor of Computer Science and Software Engineering
Saad Biaz, Associate Professor of Computer Science and Software Engineering

Abstract

Energy consumption of data storage systems has increased significantly for the past decades. There is an urgent need to build energy-efficient data storage systems. Computing cost of IT facilities and cooling cost of air conditioners contribute to a large portion of the total energy consumption of data centers. A large amount of researchers focus on reducing the computing cost by balancing workload or powering off idle data nodes to save energy. In recent years, growing attention has been paid to decreasing the cooling cost. Temperature is a major contributor to cooling cost, and thermal management has become a popular topic in building energy-efficient data centers. Extensive research of thermal impacts of processors and memories has been presented in literature, however, the thermal impacts of disks have not been fully investigated.

In this dissertation, experiments are conducted to characterize the thermal behavior of processors and disks by using real-world benchmarks (e.g., postmark and whetstone). The profiling results show that disks have comparable thermal impacts as processors to overall temperature of a data node. Then, we develop an approach to generate thermal models for estimating temperatures of processors, disks, and data nodes. We validate the thermal models by comparing the predictions with real measurements by temperature sensors deployed on data nodes. We further propose an energy model to estimate the total energy cost of data nodes. Finally, by applying our thermal and energy models, we propose thermal management strategies for building energy-efficient data centers. These strategies include a thermal-aware task scheduling strategy, thermal-aware data placement strategies for homogeneous and hybrid storage clusters, and a predictive thermal-aware data transmission strategy.

Acknowledgments

This dissertation would not have been completed without invaluable guidance, experience sharing, constant support and encouragement from my advisor, people in our research group and family members during my study at Auburn University.

First and foremost, I would like to give my most sincere and deepest gratitude to my advisor, Dr. Xiao Qin, for his great efforts, trust and patience in my work. I will never forget his extensive knowledge in the field of storage systems and inexhaustible enthusiasm for research, which keeps inspiring and driving me to accomplish my research. When working on the book chapter "Thermal Modeling and Management of Storage Systems in Data Centers", his insightful advice and suggestion helped and enlightened me in setting up accurate motivations behind the research, building a thermal model and two concrete thermal-aware strategies that can reduce cooling and energy costs in data centers.

I am also tremendously grateful to be advised by my committee members, Dr. Cheryl Seals, Dr. David Umphress and Dr. Saad Biaz, who reviewed my proposal and dissertation documents. They gave me a number of valuable suggestions, by which my dissertation had been substantially improved. I would like to show my appreciation for Dr. Brian Thurow as my university reader.

Working with our research group is fantastic. I owe my gratitude to Xiaojun Ruan, Zhiyang Ding, Shu Yin, James Majors, Yun Tian, Jiong Xie, Yixian Yang, Maen Al Assaf, Ji Zhang, Ajit Chavan, Tausif Muzaffar, Sanjay Kulkarni and Yuanqi Chen, who helped me with paper writing, experimental result collection and group discussions. In addition, all the professors and students in the Department of Computer Science and Software Engineering are greatly appreciated, because an excellent atmosphere for study and research is created and maintained by everyone.

Finally and most importantly, the endless love from my family is the most powerful strength that keeps me fighting for my research. My mother Taochun Xie, my father Jizhong Jiang and my husband Ji Zhang always stay with me, cheering for achievement and overcoming all difficulties.

To my parents

and Ji Zhang

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Thermal modeling and management techniques have been widely studied in recent years. Research shows that thermal management could increase energy efficiency of data centers. Previous research studied the thermal impacts of processors on data storage nodes; however, the thermal impacts of disks are not fully investigated. In our study, we consider the thermal impacts of both processors and disks, and propose a thermal model to estimate the outlet temperature of a data storage node based on the activities of processors and disks. By applying our thermal model into an energy consumption model, we estimate the total energy cost of data nodes. Furthermore, we also evaluate the impact of different thermal management strategies on energy consumption.

Energy consumption of data centers has increased very quickly recently [111] [57], among which the portion that computing cost and cooling cost take is growing significantly. Statistics show that computing cost and cooling cost of data centers take up to 25% of the total energy cost of data centers [20]. There have been studies to analyse the performance and energy consumption of data centers. Chen *et al.* studied the task-based energy consumption of cloud storage systems [34] and proposed StressCloud to analyse the performance and energy consumption of the systems [33]. Zhang and Fu presented power profiling results on a cloud test bed by combining hardware and software that achieves power and energy profiling at server granularity [120]. In order to reduce the energy cost of data centers, much effort has been made to reduce the computing cost and cooling cost of storage systems.

1.1 Motivations

Our proposed thermal model is indispensable for next-generation storage clusters because of the following five factors:

1. the ever-increasing cooling and energy costs of large-scale storage clusters,
2. the impact of hybrid storage on thermal management of data centers,
3. the importance of reducing thermal monitoring cost,
4. the capability of estimating the cooling cost of a data center, and
5. the lack of study on the impacts of hard drives and solid state disks on outlet temperatures of storage nodes in a cluster.

With the increase of energy consumption and cooling costs of large-scale storage clusters, there is an urgent need for data center designers to address the energy efficiency issues [10]. Conventional energy-saving approaches for data centers include improving the energy efficiency of computing facilities as well as cooling systems.

Cooling costs contribute a large portion of the total energy cost of data centers [49][10]. For instance, the power and cooling cost to support IT equipments take more than half of the total energy cost of a data center [49]. Previous studies demonstrate that energy efficiency could be enhanced by reducing the energy dissipation in cooling systems [83][113]. Reducing outlet temperatures or optimizing air recirculation can improve energy efficiency [108]. Moreover, load balancing strategies were proposed to gain good temperature distribution. Recent studies show that reducing outlet temperatures of servers in a data center could save up to 40% energy consumption [83]. Lowering outlet temperatures of storage nodes not only conserves cooling cost, but also improves the reliability and lifetime of disks [90][116].

A handful of studies have focused on modeling the energy consumption of storage clusters in the past years. For example, an energy model is introduced to estimate the power consumption of storage nodes running under specific workloads [16]. Unfortunately, thermal models of storage clusters are still in their infancy. Little attention has been paid to the thermal impact of disks, including HDDs and SSDs, on the energy efficiency of cooling systems in data centers.

Deploying temperature sensors on storage nodes of a cluster is an usual method to monitor the storage cluster's temperature. For each data node, one needs to apply at least two sensors to obtain the inlet and outlet temperatures. If temperatures of other interior devices of a node need to be monitored, additional sensors must be set up. Although this traditional approach is practical for measuring temperatures of small-scale storage clusters, it becomes a sophisticated solution when a storage cluster has thousands of nodes. It is extremely expensive to set up a huge number of sensors in a large-scale storage cluster; deploying sensors also leads to extra energy cost. Thermal models are a promising alternative to obtain temperatures of storage clusters.

Building a data center is a huge investment for enterprises. Estimating the energy costs, which include cooling cost and power cost, offers an important guideline in the designing phase. Simulations and thermal models help data center designers make decisions on thermal management during the planning phase.

A variety of factors impact the outlet temperatures of storage nodes. A study shows that inlet temperatures and CPU utilization affect the outlet temperatures of data nodes [108]. In a second study, a temperature model was proposed using historical temperature data and airflow of a data center [71]. When it comes to the thermal behavior of disks, Kim *et al.* investigated the relationship among disk seek time, inter-seek time, and disk temperatures [65]. They also observed that the number and size of flatters in a disk affect its temperatures. In enterprise-level Tera-data centers, a single node is capable of supporting more than 100 disks [88]. The temperature of these large number of disks within a data node plays a crucial role in impacting the outlet temperatures of the node. However, there is lack of studies on the impact of hard drives and solid state disks on outlet temperatures of storage nodes in a cluster.

In addition, with the growth of data transmission through networks, the energy cost of these transmission activities becomes another important issue in maintaining energy-efficient data centers. For example, for the famous social network website, Facebook, the worldwide monthly active users increase from 100 million in the third quarter of 2008 to 1.28 billion in the first quarter of 2014 [106]. There are 72 million links shared, 300 million photos uploaded, 2.5 billion status updated, and 2.7 billion likes and comments are made every day [11]. For such a huge amount of data transmissions through the internet, the energy consumption of these transferring activities are considerably high. If we could reduce the energy consumption of these data transmissions, we would gain a large deduction in both the computing and also the cooling cost of data centers.

1.2 Contributions

We introduce a modeling approach to build thermal models for estimating the outlet temperature of a storage node and propose thermal-aware management strategies for data storage systems. We make the following three contributions.

- First, we generate the thermal profile of a storage server. The profiling results are obtained by running CPU-intensive and I/O-intensive workloads imposed by Whetstone [9] and Postmark [63], respectively. When the CPU/disk is running under various workload scenarios, we monitor CPU/disk temperature as well as the inlet and outlet temperatures of the data node. We study not only the thermal behavior of a hard disk drive but also a solid state disk.

- Second, we build a thermal model to estimate outlet temperatures of data nodes using inlet temperatures, CPU and disk workloads. This model can predict outlet temperatures from CPU and disk utilizations.

- Third, we propose thermal management strategies to build energy-efficient data storage systems.

1.3 Organization

The rest of this dissertation is organized as follows. The next chapter presents prior studies and related research works. In Chapter 3, preliminary temperature models are proposed for estimating the out temperature of a data node when processors or disks are in idle or fully utilized. In Chapter 4, experiments are conducted to study the thermal behaviors of processors and disks under various utilizations, and a thermal modeling approach is presented for predicting the temperature of processors and disks by taking into account their utilizations. Furthermore, a outlet temperature model is developed for data nodes. These thermal models are validated against real-world measurement acquired by temperature sensors.

In Chapter 5, a thermal management strategy is proposed for task scheduling in data centers. Then, two data placement strategies are stated for homogeneous and hybrid storage systems in Chapter 6. Chapter 7 presents a predictive thermal-aware management strategy for data transmission. Finally, Chapter 8 concludes the dissertation, and points out the future research.

Chapter 2

Related Work

Energy consumption of data centers has increased significantly in the past years. Numerous methods are proposed to save energy cost for data centers. This chapter briefly presents previous research in building energy-efficient data centers. Research of computing and cooling cost reduction is introduced, and thermal models that play critical roles in predicting cooling cost are also studied. Features and previous studies of solid state disks are investigated. A lot of thermal management strategies have been raised to save energy by considering the thermal impacts on cooling cost of data centers. In addition, data compression is another method to save energy by reducing the data set size and improving the performance of data transmission.

2.1 Energy-efficient Data Centers

Tens of thousands of data centers around the world are consuming huge amount of energy. Increasing business companies, IT companies, and institutes are planning to build their own data centers. A study by DatacenterDynamics demonstrates that worldwide investment in data centers in 2012 had increased by 22.1% up to 105 billion dollars compared with 2011, and this investment is going to grow by another 14.5% to 120 billion dollars for 2013 [62].

Research shows the rapid increment of energy consumption of data centers [51] [57] [111]. A report announces that 1,500 TWh of electricity, which is nearly 10% of world electricity generation, is used by the world's Information-Communication-Technologies (ICT) ecosystem annually [81]. Furthermore, global data centers are estimated to consume (as of 2010) from 250 to 350 TWh every year. A reason behind the striking energy consumption in data centers is the rapid growth of computing and storage capacity in recent years. For instance, Facebook has invested more than 1 billion in IT facilities that power its social network, which now serve more than 845 million users in a month around the world [80].

Cloud computing has become a popular topic in recent years. A study shows that coal and nuclear, which generate severe air pollution, are used to satisfy these large amount of electrical energy demand [41]. Apple, HP, IBM, Facebook, and Mircosoft are using dirty energy to power their growing cloud data centers. Confronting with the rapid increment of energy consumption and severe air pollution, growing attention has been paid to build energy-efficient data centers [12] [43] [53] [69]. At the same time, small or medium sized organizations began to move their computing applications to an Internet-based "cloud" platform in order to improve energy efficiency [114].

Computing cost and cooling cost are major components of total energy consumption for data centers. Computing cost refers to the electronic energy cost that makes the IT facilities working. And cooling cost is the cost of cooling systems that lower down the temperature in data centers. Studies have been conducted on reducing either the computing cost or cooling cost in order to build energy-efficient data centers.

2.1.1 Computing Cost

A lot of research have been done in reducing computing cost of data centers [14] [100] [119]. For instance, CMPs are widely used in data centers, and the frequency/voltage of CPU cores could be adjusted in order to save power consumption. Mishra *et al.* proposed a two-tier feedback-based control scheme, in which the first-tier is comprised of a global power manager to allocate power targets to individual islands according to workloads and the second-tier consists of local controllers that adjust island power through changing the voltage and frequency as a response of workload requirements [82]. A power-efficient scheme for erasure-coded storage clusters–ECS2–was proposed, which aims to offer high energy efficiency with marginal reliability degradation [56].

Popular strategies to reduce computing cost include redistributing workload and powering off idle disks or data nodes. For example, an energy-efficient strategy was proposed which specifies a subset of disks as cache disks and dispatches workloads to these cache disks while making the other disks spin down [38]. Another strategy introduced a Popular Data Concentration (PDC) technique that migrates frequently accessed data to a subset of disks [89]. Then the other disks which are not accessed frequently could be transitioned to low-power mode, and the total computing cost of these data nodes could be reduced.

Many researchers concentrate on resource management and task scheduling in data centers to decrease computing energy consumption [13] [23] [24] [70] [112]. For instance, Beloglazov and Buyya proposed an energy-efficient resource management system for virtualized Cloud data centers [24]. In this system, VMs are consolidated according to the utilization of resources, and virtual network topologies are built between VMs and thermal status of computing nodes to save energy. This management system reduces the operational costs of data centers and provides the required Quality of Service (QoS). Beloglazov *et al.* also demonstrated an architectural framework (including resource provisioning and allocation algorithms) and principles for energy-efficient Cloud computing [23]. Experimental results show that their Cloud computing model has immense potential in energy saving and energy efficiency improvement under dynamic workload scenarios. In addition, Aksanli *et al.* demonstrated an adaptive job scheduler that utilizes the prediction of solar and wind energy production [13]. This job scheduler improves the energy efficiency by three times. Lee and Zomaya pointed out that under-utilized resources account for a large amount of energy use and resource allocation strategies could be applied to achieve high energy efficiency [70]. They proposed two task consolidation heuristics methods that aim to maximize resource utilization and take into account of both active and idle energy consumption. Experimental results illustrated the energy saving capability of their heuristics.

With the growing of data center density and size, designers should take into account of both energy costs and carbon footprint. Altering the usage patterns of data centers is believed to be a practical method to affect demand response. Chiu *et al.* pointed out that shifting computational workloads across geographic regions to match electricity supply may help balance the electric grid [37]. They proposed a symbiotic relationship between data centers and grid operators and a low cost workload migration mechanism. Ren and He proposed an online algorithm, called COCA (optimizing for COst minimization and CArbon neutrality), for minimizing operational cost in data centers while satisfying carbon neutrality without long-term future information [92]. COCA enables distributed server-level resource management: each server autonomously adjusts its processing speed and optimally decides the amount of workloads to process. Analysis of trace-based simulation studies show that COCA reduces cost by more than 25% (compared to state of the art) while resulting in a smaller carbon footprint.

Furthermore, network facilities are also investigated in order to reduce the energy consumption of data centers. The architecture of a Data Center Network (DCN) affects its scalability, however, its power consumption is a main contributor to its energy cost. Hammadi and Mhamdi classified existing DCNs as switch-centric and server-centric networks, and conduct literature review of existing technologies in energy saving and renewable energy approaches [55].

2.1.2 Cooling Cost

Cooling cost is an unignorable component of the total energy consumption for a data center. Increasing studies are investigating strategies to save cooling cost [22] [76] [94]. Generally speaking, there are seven categories of strategies for saving cooling cost of data centers [39]. Major strategies include managing airflow in data centers; locating cooling systems as close as to IT equipments; using dynamic control to fit with the thermal load of data centers; and maintaining a higher operating temperature.

A novel approach was proposed to model the energy flows in a data center and optimize its operations [76]. Overall sustainability of data center operations could be improved through a holistic approach. In this approach, predictions of renewable energy and IT demands were conducted and an IT workload management plan was generated. This management plan schedules IT workloads and allocates IT resources depending on cooling efficiency and power supply. Experimental results show that this approach saves both recurring power cost and the use of non-renewable energy.

However, constraints exist in optimizing energy consumption of data centers, such as the threshold for income temperatures, the capacity and response time. To balance the performance and the temperature constraint, a coupled thermal-performance model and a cooling-aware workload placement strategy were proposed [94]. This thermal-performance model leads to a power saving of 21% and the data placement strategy gains energy saving of 8%.

Research demonstrated the efficiency of workload management strategies in reducing outlet temperature of data nodes [83], minimizing heat recirculation [109], or decreasing inlet temperature which leads to a reduction of cooling cost of data centers [110]. For instance, a thermal-aware task scheduling algorithm, XInt, was proposed to minimize heat recirculation by balancing the workloads within a homogeneous data center [110]. In this work, researchers discovered that cooling costs highly depend on peak inlet temperatures. In order to lower cooling power, they designed a task assignment policy, MPIT-TA, which minimizes the peak inlet temperature through task assignment. Their simulation results show that MPIT-TA saves at least 20% of cooling energy.

After analysing Energy Inefficiency Ratio of SPatial job scheduling (a.k.a. job placement) algorithms, also referred as SP-EIR, a coordinated cooling-aware task placement and cooling management algorithm, Highest Thermostat Setting (HTS), was developed [22]. HTS is aware of dynamic behavior of the Computer Room Air Conditioner (CRAC) units and dispatches tasks to reduce the cooling demands from the CRACs. Dynamic updates of the CRAC thermostat settings based on the cooling demands could decrease the total energy consumption.

2.2 Thermal Modeling

Temperature is a major contributor to cooling cost, and studies have been conducted to reduce heat generation or speed up heat dissipation. Heat sinks and heat pipes are investigated to promote heat dissipation. For instance, a heat sink model associated with one of IEEE EMC challenging problems was used to study three different grounding configurations [77]. A new simulation model for an Intel P4 CPU heat sink was proposed and analyzed. Then, an optimal design of the CPU heat sink could be performed in order to minimize the radiated emission from the CPU heat sink. Besides heat sinks, heat pipes are also applied to transfer heat from hot to cold regions. Researchers present a time- and temperature-aware methodology that uses additional heat pipes [50]. A thermal model was developed to simulate effects of metal interconnects on heat distribution. Results show that, by deploying additional heat pipes, their methodology gains a 5% to 7% decrease in temperature variation through-out and 2 to 3 degree reduction in hotspot temperature.

Besides deploying heat sinks and heat pipes, another solution is to reduce the heat generation of data nodes. To characterize the thermal behaviour of data nodes, studying the thermal impact of IT components on each data node is an important approach. There have been extensive research investigating the thermal behaviors of CPU, disk, memory, and network cards, and researchers find that these components make key contribution to the outlet temperature of a data node.

2.2.1 CPU Models

CPU had been identified as a resource that greatly contributes to energy-consumption in data centers. Studies analyzed and modeled the power consumption of processors [26] [28] [45]. Thermal impacts of processors are also widely studied. For example, HotSpot was proposed to estimate the temperatures of CPUs, which could accurately and fast predict the temperatures of CPUs at the micro-architecture level [103]. This model is based on an equivalent circuit of thermal resistances and capacitances that correspond to micro-architecture blocks and essential aspects of the thermal package. In order to model the thermal behaviour of different types of CPU, the micro-architecture need to be studied and sophisticated models should be generated. Few work has been done to model CPU temperatures at coarse-grained level.

2.2.2 Disk Models

There have been studies investigating thermal characteristics of disks. An early work proposed a thermal model to predict the transient temperature of an IBM's fixed disk drive [46]. Another work introduced a three-dimensional transient temperature model, which estimates disk temperatures under frequent seeking operations [107]. A comprehensive model which takes into account of five components (internal drive air, spindle motor, the base and cover of the disk, the voice-coil motor, and disk arms) of a hard drive disk was demonstrated to predict disk temperature [54]. Researchers also studied the impacts of seek time and inter-seek time on disk temperature [65], and they found that either increasing the inter-seek time or decreasing the seek time could decrease the disk temperature.

Previous disk temperature models took into account of heat dispatching and disk activities at a fine-grained perspective. Detailed specifications are need to model the temperature of a new disk. The disk temperature under particular workload cannot be estimated by simply using previous modeling approaches. To address this problem, we conducted studies on disk temperatures by considering the thermal impacts of disk utilizations on disk temperatures, and proposed thermal models for both hard drive disks and solid state disks [58] [59].

2.2.3 Memory Models

Approaches were also proposed to coordinate processors and memory to improve system performance and/or power efficiency during memory thermal emergency [73] [74]. An adaptive core gating (DTM-ACG) and coordinated DVFS (DTM-CDVFS) schemes as well as a thermal model were designed to predict DRAM temperatures [74]. Experimental results demonstrate that these two schemes exhibit 6.7% and 15.3% of improvements in terms of performance. DTM-CDVFS also reduces the processor power rate by 15.5% and system (including processor and memory) energy by 22.7%. Besides that, a DRAM thermal model was proposed and validated with measurement on an instrumented server platform. Experimental results illustrate that their model reflects the dynamic DRAM temperature changes; the average temperature difference between estimated and measured values is less than 1 ℃.

2.3 Solid State Disk (SSD)

Solid state disk (SSD) is an emerging storage technology with high I/O performance and energy efficiency. There is a high potential to widely apply SSDs in large-scale cluster storage systems. SSDs are more expensive than traditional hard drives [85], but they perform better than traditional hard drive disks in random reads and writes [44] [75] [78]. Meanwhile, with the increasing density of flash-based SSDs, reliability, endurance, and performance are all declining [52]. Growing studies are conducted to improve the reliability and performance of SSDs [31] [61] [102].

To improve both performance and energy efficiency, hybrid SSD devices may be employed to build large storage systems. Recently, Chang proposed an SSD-based hybrid storage system that combines MLC flash-based and SLC flash-based SSDs [30]. Their experimental results demonstrate that compared with MLC-flash-based SSD storage, the hybrid system can gain significant improvements in terms of throughput and energy savings. Oh *et al.* proposed a cost-effective and reliable SSD host cache solution–SRC (SSD RAID Cache) [86]. In this solution, cost-effectiveness is ensured by using multiple low-cost SSDs and reliability is enhanced by RAID-based data redundancy.

Apart from hybrid SSDs, hybrid storage systems that combine HDDs and SSDs have also been proposed to make a good trade-off between performance and cost. For example, Chen *et al.* designed a hybrid storage system – Hystor – in which hot data is stored in SSDs to optimize system I/O performance [35]. All data accesses are periodically recorded and analyzed by a *monitor* module. When any data becomes hot, it will be moved to a SSD to reduce data access time. Wu *et al.* developed a hybrid page/block architecture along with an advanced replacement policy called BPAC to exploit both temporal and spatial locality [115]. Mao *et al.* proposed a hybrid parity-based disk array architecture (HPDA), where SSDs and HDDs are integrated in a RAID system to improve the performance and reliability of the RAID [79]. Balakrishnan *et al.* proposed Diff-RAID, a parity-based redundancy solution that unevenly distributes and balances the parity across SSDs to improve the reliability of storage systems [21]. Schall *et al.* investigated the performance and energy efficiency of SSDs and HDDs in I/O-intensive database applications [97]. Although hybrid storage systems can offer good performance and reliability, less attention has been paid to the thermal characteristics of hybrid storage devices that have significant impacts on the energy costs of cooling systems in future data centers.

2.4 Thermal Management

Improving energy efficiency becomes increasingly important for data centers. Techniques or strategies reducing energy cost of cooling systems make a major contribution to advance energy-efficient data centers. Growing research focuses on thermal management to build energy-efficient data centers [47] [64] [87] [99] [118]. Thermal-aware resource management strategies are proposed for balancing temperature distribution in data centers in order to save energy consumption.

2.4.1 CPU Thermal Management

A handful of temperature-aware load balancing strategies which considering the thermal impacts of processors were proposed [67] [95]. For instance, a customized threshold is set to limit CPU temperatures [96]. If the CPU temperatures exceed the threshold, the CPU's voltage and frequency will be dynamically adjusted to conserve CPU energy consumption at the cost of increasing execution time. Sharma *et al.* demonstrated a thermal-load-balancing framework to dynamically distribute workloads across data nodes in a data center [101]. Their simulation results show that equipment reliability can be improved by placing an asymmetric workload and uniformly distributing temperature in data centers.

Ayoub *et al.* stated a multi-tier approach for significantly reducing the cooling costs associated with fan subsystems without compromising the system performance [17]. Fan speed is managed by intelligently allocating the workload at the core level as well as at the CPU socket level. At the core level, a proactive dynamic thermal management scheme is proposed and a new predictor is also introduced to utilize the band-limited property of the temperature frequency spectrum.

2.4.2 Memory Thermal Management

Energy consumption and thermal behaviour of memory are also investigated. A joint energy, thermal and cooling management technique (JETC) was proposed to reduce the cooling and memory energy cost of each server [18]. JETC takes into account of thermal and power states of CPU and memory, thermal coupling between CPU, memory and fan speed to make energy efficient decisions. CPU and memory actuators are used to make decisions. The memory actuator decreases the energy cost of memory by performing cooling aware clustering of memory pages to a subset of memory modules. The CPU actuator reduces cooling energy by lowering down the hot spots between and within the CPU sockets and minimizing the effects of thermal coupling. Their experimental results show that employing JETC leads to 50.7% average energy reduction in cooling and memory subsystems with less than 0.3% performance overhead.

A Coordinated Management of Energy, Thermal, and Cooling (CoMETC) technique was proposed to minimize cooling and memory energy of server machines [19]. State-of-the-art solutions decouple the optimization of cooling costs and energy consumption of CPU and memory subsystems. This leads to suboptimal solutions because of thermal differences between CPU and memory and the non-linearity in energy costs of cooling. CoMETC decreases the memory operational energy by clustering active memory pages to a subset of memory modules while accounting for thermal and cooling aspects. Simultaneously, CoMETC removes hotspots between and within the CPU sockets and reduces the impacts of thermal coupling with memory.

2.4.3 Storage Thermal Management

Energy consumption and power management of storage systems are widely investigated [27] [72] [93] [98]. After studying write policies [66], cache and prefetching techniques are proposed to save energy consumption for disks. For instance, Song *et al.* proposed a data prefetching scheme, in which the amount of data prefetched for each video stream is dynamically adjusted for the bit-rates of streams and the power characteristics of different disks [104].

As is known that the biggest power consumer in data centers is the storage system. Disk drives are lowly utilized and there is large space for savings power consumption of disks. A methodology that quantitatively estimates the performance impact for power savings was proposed [117]. By taking into consideration the effects of propagation delay, the correctness and efficiency of the proposed analytical methodology was verified in their experiments driven by production server trances.

A large fraction of the power budget in data centers is consumed by storage systems. Enterprise storage systems are not widely deployed with power-saving solutions. The traditional way that spins down disks is ineffective because idle periods are too short for industry workloads. By analyzing block-level traces from 36 volumes in an enterprise data center for one week, Narayanan *et al.* made conclusions that significant idle periods exist and can be further increased by modifying the read/write patterns using write off-loading [84]. Write off-loading allows write requests on spun-down disks to be temporarily redirected to other persistent storage in the data center. Experimental results show that spinning down disks when they are in idle state could save 28-36% of energy, while write off-loading further increases the savings to 45-60%.

Disk power consumption could be saved by turning a disk drive into a low power mode during idle times. Problem exist that future job arrivals is unaware, thus future disk activities could not be predicted. By exploring ranges and trade offs of possible power savings and performance within a set of enterprise storage traces, Riska and Smirni demonstrated the difficulty of obtaining significant power savings in traces where overall utilization is less than 5% and explored the feasibility of popular schemes such as workload shaping for power savings [93]. They proposed a proactive autonomic algorithm that provides suggestion on when and for how long a power savings mode should be activated by given an acceptable performance degradation target. Their experimental results show the robustness of the algorithm.

Bostoen *et al.* studied alternative methods that reduce disk access time, conserve space, or exploit energy-efficient storage hardware in dynamic power management [27]. Previous energy-conservation techniques do not consider the fundamental trade-offs between power, capacity, performance, and dependability. They stimulated an integration of different power-reduction techniques in new energy-efficient file and storage systems.

However, previous load balance strategies have not fully considered disks as an important thermal impact to the outlet temperature. In this dissertation, we will study the thermal impacts of disks and propose thermal-aware management strategies to save energy consumption, especially cooling cost.

2.4.4 NoC Thermal Management

Nowadays, three-dimensional network-on-chip (3D NoC), which integrates NoC and die-stacking 3D IC technology, achieves lower latency, higher network bandwidth, and lower power consumption. However, with the increment of dies stack vertically, the raise of length of heat conduction path and power density per unit area cannot be ignored. Chao *et al.* found that routers of NoC have comparable thermal impact as processors [32]. Their research shows that NoC contributes significantly to overall chip temperature. They proposed a traffic- and thermal-aware run-time thermal management (RTM) scheme, which ensures both thermal safety and less negative performance impact from temperature regulation. Based on their simulation experiments, the RTM scheme is effective and can be combined with thermal-aware mapping techniques to achieve higher run-time thermal safety.

Though three-dimensional Network-on-Chip (3D NoC) has been proposed to solve complex on-chip communication issues, however, thermal problem become another big issue because of the larger power density and the heterogeneous thermal conductance in different silicon layer of 3D NoC [36]. When a device is thermal-emergent, Dynamic Thermal Management (DTM) techniques will be triggered. However, these reactive DTM schemes result in significant system performance degradation. Thus, they proposed a temperature prediction model and a proactive DTM with vertical throttling (PDTM-VT) scheme, which is managed by the distributed Thermal Management Unit (TMU) on each NoC node. Based on their prediction temperature model, the TMU can manipulate devices to avoid thermal-emergent. According to their experimetal results, the prediction error of the proposed temperature prediction model is less than 0.25% compared with real measurement within 50ms. Furthermore, a 11.84% - 23.18% reduction of thermal-emergent nodes and a 0.47% - 47.90% improvement of network throughput can be observed when PDTM-VT is used.

2.4.5 Predictive Thermal Management

Besides traditional dynamic thermal management techniques which making actions after emergency, predictive thermal management strategies have also been studied [48] [91]. A performance-effective Dynamic Thermal Management (DTM) system for multimedia applications was demonstrated to reduce energy consumption [105]. In this study, a predictive DTM algorithm was developed to efficiently use response mechanisms. The experimental results show that the DTM algorithm performs significantly better than existing reactive DTM algorithms. Another group of researchers built a software structure for Internet services (C-Oracle) [91]. In this study, the system chooses the best reaction by predicting and evaluating temperature and performance impacts of various thermal management reactions. C-Oracle effectively deals with thermal emergencies without unnecessary performance degradation. In addition, an energy-saving framework that provides energy estimation before data is transmitted was proposed [60]. Experimental results show that this frame work would choose the most energy-efficient data transmission strategy from given candidate strategies by using related runtime information.

2.5 Data Compression

Data compression techniques have been widely applied to achieve high space efficiency in storage systems and shorten data retrieval time [68] [15]. Compression techniques are able to reduce data sizes; however, existing compression techniques introduce extra CPU overhead. In addition, compression ratios of a particular method may vary greatly for different file types.

Cannane and Williams proposed a semi-static phrase-based scheme called XRAY [29]. 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 illustrate that their method performs 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) [42]. 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%.

2.6 Summary

One objective of this dissertation is to propose thermal-aware management strategies to save energy cost of data centers. To reduce energy consumption, efforts were placed on improving performance or decreasing temperatures in data centers. In the first section, we introduced main methods in building energy-efficient data centers. In the second section, thermal models of components in data nodes were investigated. We observed that previous thermal models predict disk temperature at a fine-grained level. If detailed specifications and properties of a disk are not available, it is impossible to model the disk temperature. In addition, solid state disks have become increasingly popular in data storage. In the third section, we presented related work on solid state disks. Then, previous thermal management strategies were stated in the forth section. Finally, data compression methods are discussed in the fifth section.

Chapter 3

Preliminary Thermal Models

There have been a lot of studies on constructing thermal models for data centers. Some generate models to estimate thermal behaviours of CPUs, disks, memories, and network cards. Others model the outlet temperature of data nodes by taking into account of air recirculation in data centers. However, thermal behaviour of disks and their impacts on data nodes have not been fully explored.

In this chapter, we generate the thermal profile of a storage server containing three hard disks. The profiling results show that disks have comparable thermal impacts as processing and networking elements to overall storage node temperature. Then, we develop a thermal model to estimate the outlet temperature of a storage server based on processor and disk utilizations.

The rest of this chapter is organized as follows. In Section 3.1, a group of experiments are presented for evaluating the thermal impact of both CPU and disks on outlet temperatures. In Section 3.2, we propose a thermal model for predicting the outlet temperature under four types of workloads: combinations of CPU and disks are either idle or fully utilized. The thermal model is validated against data acquired by an infrared thermometer as well as build-in temperature sensors on disks. Then, case studies of applying the thermal model to analyse real problems is presented in Section 3.3. Finally, Section 3.4 concludes this chapter by summarizing the main contributions of the chapter.

3.1 Thermal Impacts of Disk I/O

To characterize the impacts of CPU and disks on the inlet/outlet temperatures of a data node, we conduct a number of experiments on a Linux server. In these experiments, CPU temperatures are detected by software lm-sensors [3] and disk temperatures are collected by software hddtemp [1]. The inlet and outlet temperature are acquired by an infrared thermometer.

3.1.1 Testbed

The testbed used in these experiments is equipped with four Intel(R) Xeon 2.4 GHz CPU, 2.0 GBytes RAM, and three 160 GBytes SATA disks deployed in a disk array. The configuration parameters are summarized in Table 3.1.

Table 3.1 Testbed Configuration

Hardware	**Software**
4 × Intel(R) Xeon 2.4 GHz CPU X3430 1 × 2.0 GBytes of RAM 3 × WD 160 GBytes Sata disk (WD1600AAJS-75M0A0 [7])	Ubuntu 10.04 Linux kernel 2.6.32

3.1.2 Impact of CPU and Disks on Inlet/Outlet Temperatures

Outlet temperatures of a node are determined by various factors, including CPU and disk temperatures, mother-board temperatures, and inlet temperatures. The CPU factor has been addressed in prior studies (see, for example, [110] [95] [96]). Unfortunately, the thermal impact of disk I/O on data nodes remains an open issue.

Table 3.2 Experiment Configuration

Experiments	**Utilization(%)**		**Power (W)**
	CPU	**Disk**	
1	0	0	73
2	100	0	135
3	0	100	85
4	100	100	142

To investigate the relationship between CPU/disks and the inlet/outlet temperatures, we conduct four experiments, in which a combination of high (100%) and low (0%) utilizations of CPU and disks are considered. The configuration details are shown in Table 3.2. In these experiments, CPU and I/O workloads are generated by stress [5] and postmark [63], respectively. The power consumption of the testbed is measured by a power meter. The temperatures of the four cores and three disks in the testbed are presented in the rest of this section.

Low CPU and Low Disk Utilization

In the first experiment, we place both CPU and disks in the idle mode. Fig. 3.1 shows that disk and CPU temperatures keep the same. The node's inlet temperature varies slowly from 24.8 ℃ to 30.6 ℃, which leads the outlet temperature to vary accordingly. When the outlet temperature goes up, the inlet temperature also increase due to heat recirculation. On average, the difference between the inlet and outlet temperatures is 3.8654 ℃, ranging anywhere between 3.2 ℃ and 5.0 ℃. In this case, the discrepancy between inlet and outlet temperatures can be expressed as a constant. Thus, we have:

$$T_{diff1}(t) = 3.8654 \tag{3.1}$$

High CPU and Low Disk Utilizations

In the second experiment, we keep CPU extremely busy (i.e., CPU utilization approaches to 100%) while placing disks in the idle mode. Fig. 3.2 shows that the CPU temperature goes up fast; it increases 20 ℃ in 4 minutes. On the other hand, the disk temperatures do not change much. The difference between the inlet and outlet temperatures increases slowly from 4.6 ℃ to 6.6 ℃ in the first 600 seconds, and then maintain at a constant value in the next 1200 seconds. We denote inlet and outlet temperature difference as T_{diff2}, where t refers to the time at which the data node has run under 100% CPU and 0% disk utilizations.

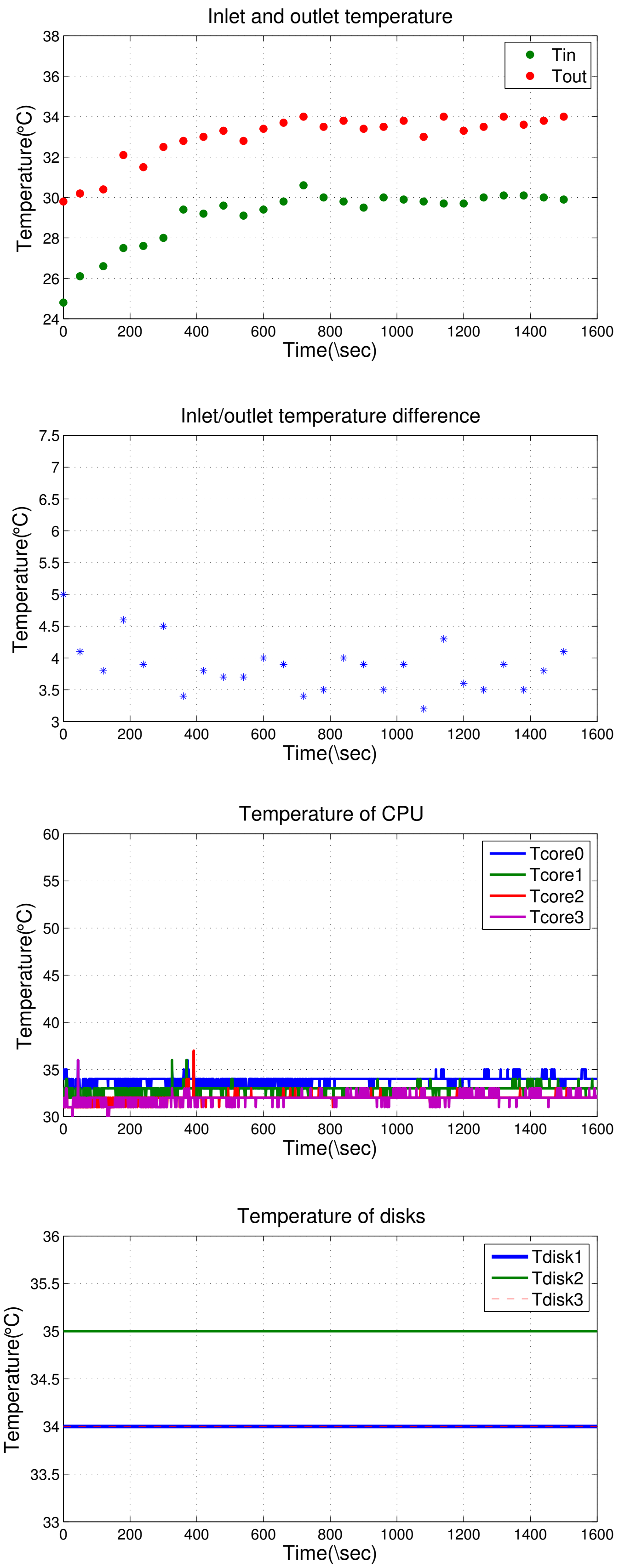


Figure 3.1: Temperature evaluation under the low CPU and low disk utilizations.

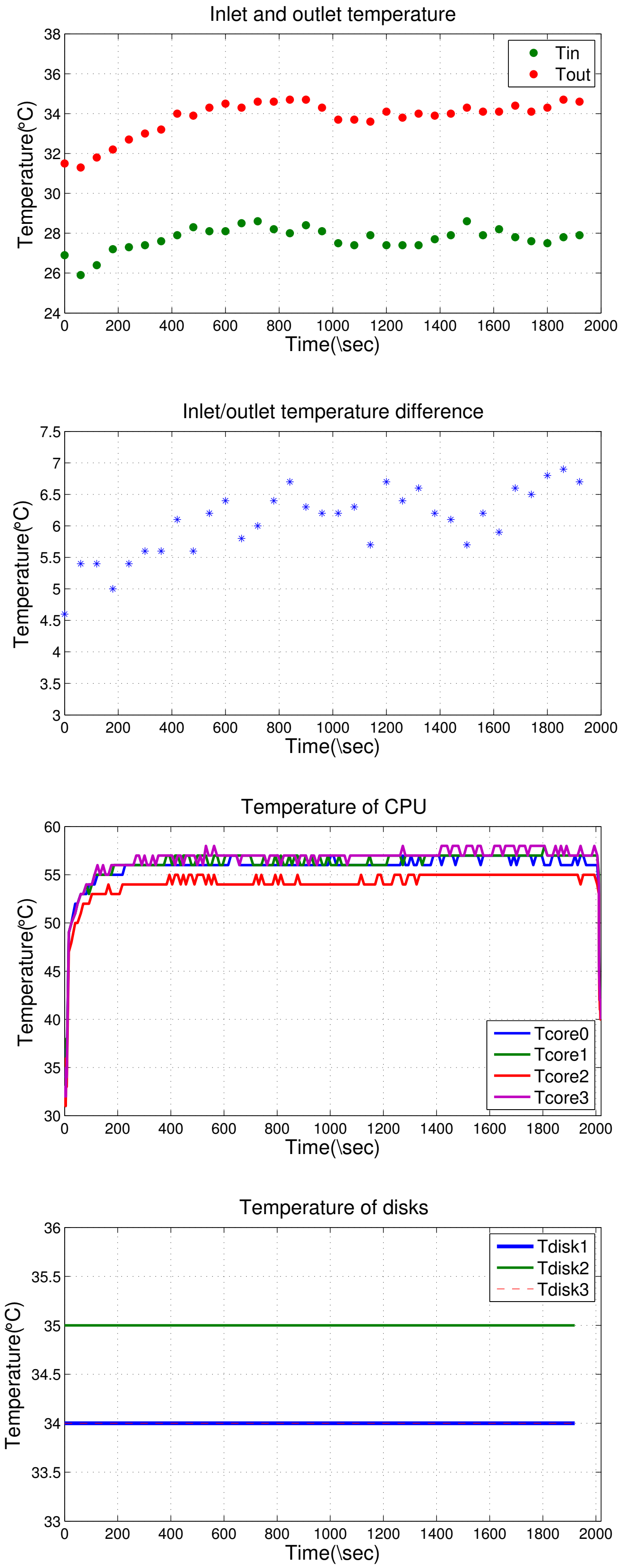


Figure 3.2: Temperature evaluation under the high CPU and low disk utilizations.

Thus, we have:

$$T_{diff2}(t) = \begin{cases} 0.0023 * t + 4.8818, & if\ t \leq 600 \\ 6.2692, & if\ t > 600 \end{cases} \tag{3.2}$$

Low CPU and High Disk Utilizations

In the third experiment, we keep a low CPU utilization while increasing disk utilization up to approximately 100%. We run three tasks, each of which imposes I/O-intensive load on the disk. We observe from Fig. 3.3 that CPU temperature frequently fluctuates between 31 ℃ and 35 ℃ , because the three I/O-intensive tasks require the CPU resource to issues I/O requests. Nevertheless, the CPU utilization remains fairly low. After completing the tasks, CPU returns to the idle status and its temperature decreases to the normal value. In this case, the thermal impact of CPU is negligible. In contrast, disk temperatures slowly increase at the rate of around 2 ℃ per 1000 seconds. The difference between inlet and outlet temperature can be expressed by (3.3).

$$T_{diff3}(t) = \begin{cases} 0.0001 * t + 4.6086, & if\ t \leq 1000 \\ 4.7086, & if\ t > 1000 \end{cases} \tag{3.3}$$

High CPU and High Disk Utilization

In the final experiment, we push both CPU and disks utilizations up to 100%. We observe that the CPU temperature increases 20 ℃ at the beginning and goes back to the original value after 1500 seconds when CPU-intensive tasks are completed. Therefore, we focus on the data collected before 1500 seconds. The inlet and outlet temperature difference falls in the range from 4.3 ℃ to 7.5 ℃ . In the first 660 seconds, the temperature difference increase very fast and then do not fluctuate much. Thus, we conclude from the experiment that CPU and disks significantly affect outlet temperatures, and the discrepancy between

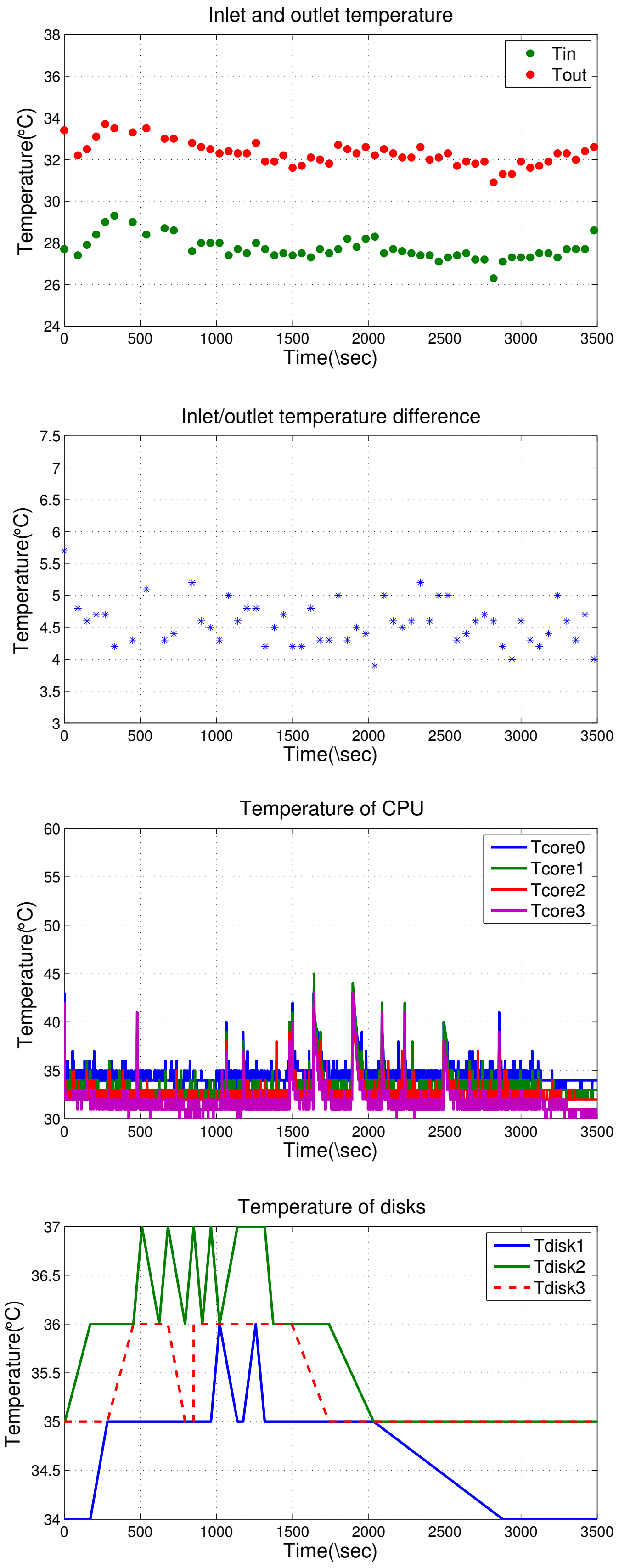
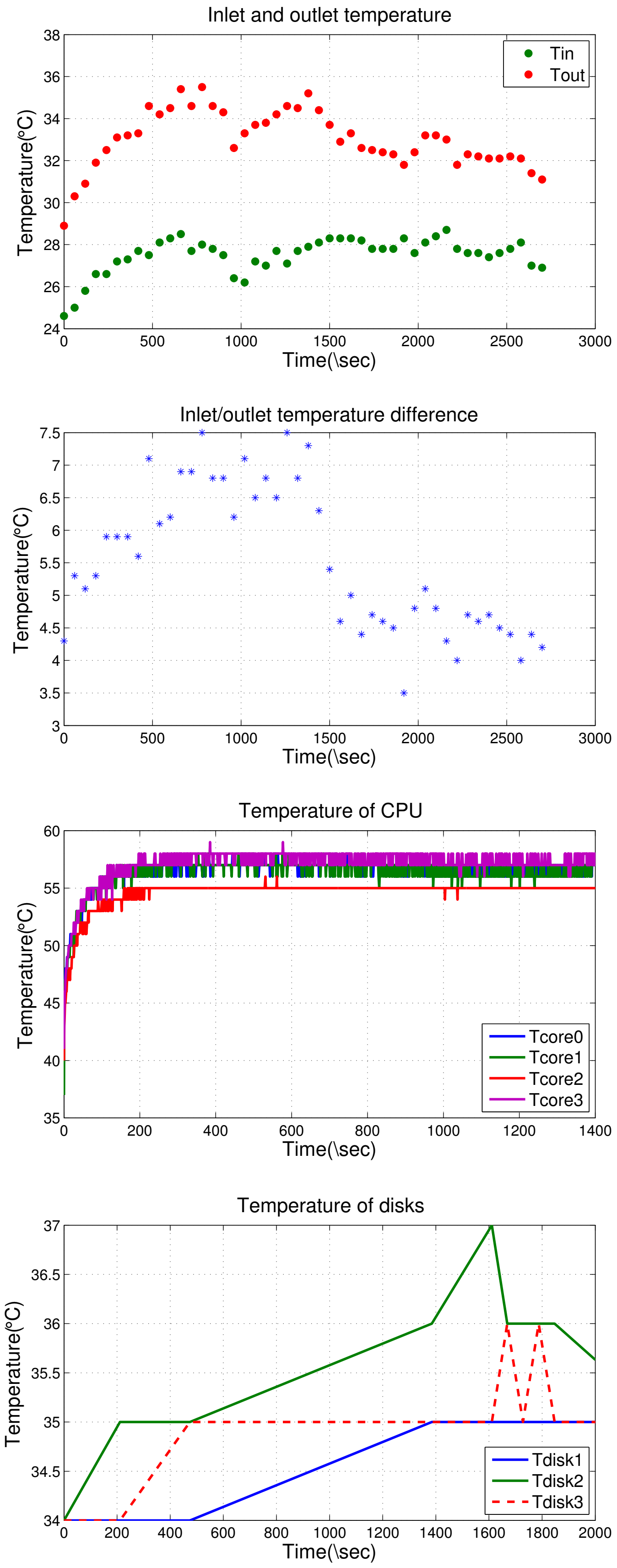


Figure 3.3: Temperature evaluation under the low CPU and high disk utilizations.



Inlet and outlet temperature

Inlet/outlet temperature difference

Temperature of CPU

Temperature of disks

Figure 3.4: Temperature evaluation under the high CPU and high disk utilizations.

inlet and outlet temperature can expressed as (3.4).

$$T_{diff4}(t) = \begin{cases} 0.0014 * t + 5.3720, & if\ t \leq 660 \\ 6.8923, & if\ t > 660 \end{cases} \tag{3.4}$$

Fig. 3.4 also shows that the average cold-start time for the three disks is more than 1200 seconds, much larger than the cold-start time of CPU (i.e., CPU cold-start time is 100 seconds).

3.2 Thermal Models

It is extremely challenging to model the energy consumption relationship between computing and cooling systems. The cooling cost depends not only on cooling setting (*e.g.*, inlet temperatures and cooling equipment placement), but also on heat dissipated by computing facilities. CPU and disks are two major types of components and heat contributors in data nodes. In this section, we develop a thermal model that aim to estimate outlet temperatures by considering the impacts of CPU and disks. Moreover, by combining a coefficient of performance (COP, for short) model that predicts cooling costs by CRAC supply temperature [83], our model can be used to predict the impact of CPUs and disks on cooling cost.

3.2.1 Framework

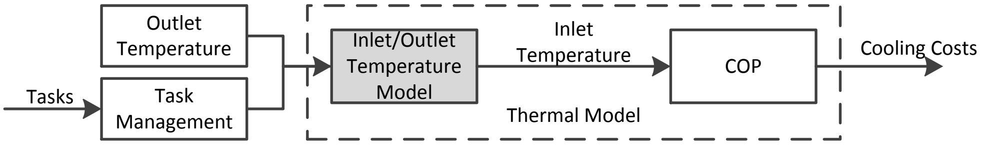


Figure 3.5: Framework of proposed solution.

Fig. 3.5 displays our thermal-modeling framework, which consists of two components, namely, inlet/outlet-temperature model and COP model. The inlet/outlet-temperature model builds up the relationship between inlet and outlet temperatures by profiling analysis. In addition, given an outlet temperature, our model estimates inlet temperatures under certain workloads. The COP model computes cooling costs by taking into account inlet temperatures offered by the inlet/outlet-temperature model. The main contributions of this framework are: (1) a thermal model that characterizes the relationship between inlet and outlet temperatures of a data node and (2) cooling cost estimation for data center designers.

3.2.2 An Inlet/Outlet Temperature Model

Considering CPU and disk utilizations, we classify workloads of a node into four basic types (i.e., see Section 3.1.2 for a combination of high and low utilizations of CPU and disks). During any time period, the workload of a node can be decomposed into a number of sub-period, in which the node runs under one of the four basic types. Thus, in each sub-period, the discrepancy between inlet and outlet temperatures is modeled by incorporating the four basic workload types.

$$T_{diff}(t) = \begin{cases} T_{diff1}(t),\ if\ U_{CPU} = 0 \quad, U_{disk} = 0 \\ T_{diff2}(t),\ if\ U_{CPU} = 100, U_{disk} = 0 \\ T_{diff3}(t),\ if\ U_{CPU} = 0 \quad, U_{disk} = 100 \\ T_{diff4}(t),\ if\ U_{CPU} = 100, U_{disk} = 100 \end{cases} \tag{3.5}$$

Given workloads and a number of sub-period $\mathfrak{T} = \{t_1, ...t_n\}$, we derive the outlet temperature from (3.1)-(3.4) as:

$$T_{diff}(\mathfrak{T}) = \frac{\sum_{i=1}^{n} T_{diff}(t_i)}{|\mathfrak{T}|} \tag{3.6}$$

3.2.3 The COP Model

The energy cost of a node is contributed by the energy consumption of the node and the cooling cost. We use COP (i.e., the Coefficient Of Performance model), described in [83], to calculate the cooling cost.

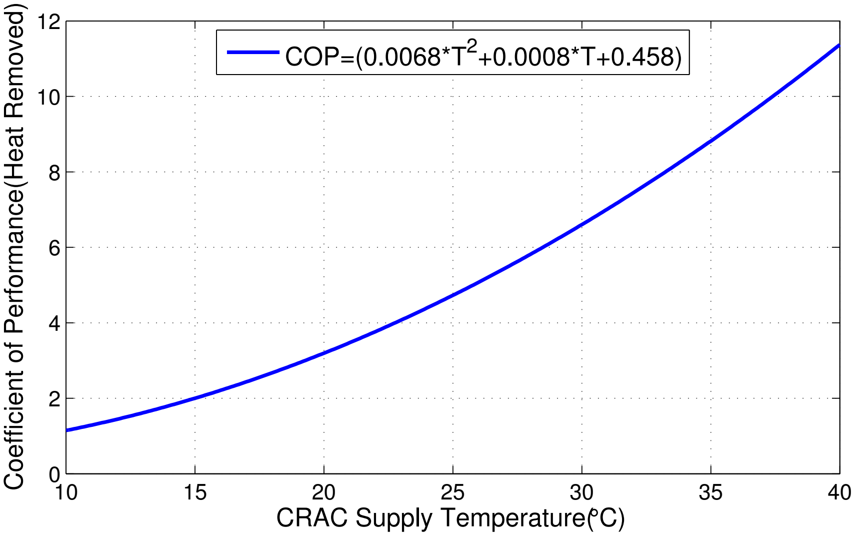


Figure 3.6: Coefficient of the performance curve for the chilled-water CRAC units at the HP Labs Utility Data Center [83].

Fig. 3.6 plots COP values that increase with the supply temperature of CRAC. A large COP value indicates a high energy efficiency.

$$COP(T) \;=\; 0.0068 * T^2 + 0.0008 * T + 0.458 \tag{3.7}$$

In 3.7, COP is defined as the ratio of heat removed to the energy cost of the cooling system for heat removal. T refers to the supply temperature of CRAC. The cooling power P_{AC} can be derived from COP using (3.8).

$$P_{AC} = \frac{P_C}{COP(T)}, \tag{3.8}$$

where P_C is the computing energy power.

3.3 Case Studies

In order to demonstrate the application of our thermal model, we conduct three case studies, representing three typical access patterns of applications. We use the same testbed (see Section 3.1) to perform the case studies. We keep all the three disks busy in the high-disk utilization cases. Let us consider the following access patterns (see Fig. 3.7) in our case studies:

- Pattern 1: In the *Computing After Reading* pattern, applications first load data from disks, then process the loaded data using CPU resources.
- Pattern 2: In the *Computing Then Writing* pattern, applications perform CPU-intensive computation first, followed by write-intensive activities to output data to disks.
- Pattern 3: In the *Computing and Reading/Writing in Parallel* pattern, applications concurrently impose both CPU-intensive and I/O-intensive load to the node.

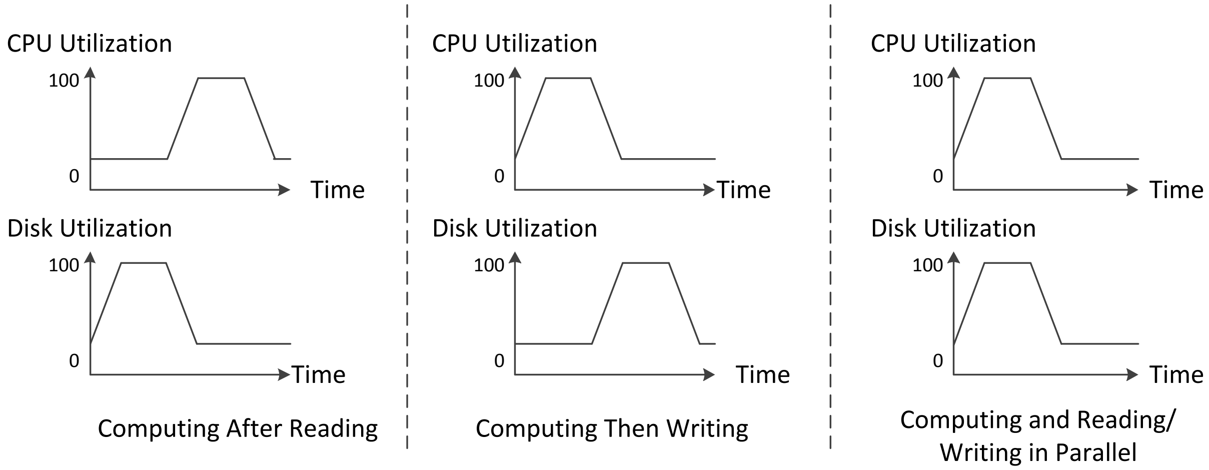


Figure 3.7: Three typical access patterns.

Since the cold-start phase of disks is longer than that of CPU, we consider two scenarios in each case study. The first scenario represents cases where that the execution time of I/O tasks is smaller than the cold start phase. In this scenario, the cold-start issue significantly affects outlet temperatures. The second scenario represents case where the execution time of I/O tasks is much longer than the cold-start time. In the second scenario, the cold-start issue becomes negligible. In the case studies, P_C is the node's power consumption.

Impact of the Cold-Start Phase

We set the execution time of both CPU- and I/O-intensive tasks to 10 minutes, which is smaller than the cold start phase of disks. During the period of 10 minutes, the difference between inlet and outlet temperatures under the four basic workload types are:

$$T_{diff1}(600) = 3.8654 \ (℃)$$
$$T_{diff2}(600) = 6.2618 \ (℃)$$
$$T_{diff3}(600) = 4.6686 \ (℃)$$
$$T_{diff4}(600) = 6.2120 \ (℃)$$

After processing the CPU- and I/O-intensive tasks for 20 minutes in each case study, we evaluate the differences between inlet and outlet temperatures as follows.

Access Pattern 1. Disks are kept in the busy status in the first phase; $T_{diff3}(600)$ denotes the inlet/outlet-temperature difference. The increase of difference between inlet and outlet temperatures is $T_{diff3}(600)$-$T_{diff1}(0)$, which is 0.8032 ℃. Since the cold-start time for disks are longer than 10 minutes, the disk temperature remains unchanged in the second phase. In this case, if the increase of the inlet/outlet-temperature difference in the first phase is considered as the increase in the inlet temperature for the second phase, and then this increment should be accumulated to the second phase. Therefore, the overall inlet/outlet-temperature difference can be derived as:

$$\begin{aligned} &T_{pattern1}(1200) \\ &= \frac{T_{diff3}(600) + T_{diff3}(600) - T_{diff1}(0) + T_{diff2}(600)}{2} \\ &= 5.8668 \ (℃) \end{aligned}$$

Access Pattern 2. We obtain an average difference between inlet and outlet temperatures (i.e., 5.4652 ℃) after running the test for 20 minutes. $T_{diff2}(600)$ is the temperature increment in the first phase, in which CPU is busy. Then, in the second phase, the CPU temperature falls down to the normal value in the first 10 seconds; the CPU temperatures in the second phase can be considered as a constant. The inlet/outlet temperature difference in the second phase can be calculated by $T_{diff3}(600)$. The average difference of inlet/outlet temperature is described below:

$$\begin{aligned} T_{pattern2}(1200) &= \frac{T_{diff2}(600) + T_{diff3}(600)}{2} \\ &= 5.4652 \text{ (℃)} \end{aligned}$$

Access Pattern 3. the inlet and outlet temperature difference increases from 3.8654 ℃ to 6.2120 ℃ in the first phase. In the second phase, the CPU temperature drops down quickly; whereas the disk temperature slowly decreases. The increasing and decreasing rates of disk temperature are slow; no difference is observed in a 10-minute period. Hence, we use T_{diff4} and T_{diff1} to calculate $T_{pattern3}(1200)$ as:

$$\begin{aligned} T_{pattern3}(1200) &= \frac{T_{diff4}(600) + T_{diff1}(600)}{2} \\ &= 5.0387 \text{ (℃)} \end{aligned}$$

Theoretically, cooling costs under these three patterns can be reflected by the inlet-outlet-temperature difference. To precisely evaluate cooling costs, we use the COP model that takes inlet temperatures as an input and produces cooling energy consumption. The inlet temperatures in the case studies are calculated in the way that identical outlet temperatures will be produced after the CPU- and I/O-intensive tasks are executed. For example, the inlet temperatures under the aforementioned access patterns are 24.1 ℃, 24.5 ℃ and 25.0 ℃ with outlet temperature being 30 ℃.

According to the COP model, the COP values of these access patterns are:

$$COP_{pattern1} = COP\ (24.1) = 4.4268$$

$$COP_{pattern2} = COP\ (24.5) = 4.5593$$

$$COP_{pattern3} = COP\ (25.0) = 4.728$$

Given power (see Table Table 3.2) of the node, we derive the energy dissipation as:

$$P_{POWER1} = 135 * 600 + 85 * 600 = 132,000(J)$$

$$P_{POWER2} = 135 * 600 + 85 * 600 = 132,000(J)$$

$$P_{POWER3} = 142 * 600 + 73 * 600 = 129,000(J)$$

The cooling costs calculated by the COP model are:

$$P_{AC1} = \frac{P_{POWER1}}{COP_{pattern1}} = 29,818(J)$$
$$P_{AC2} = \frac{P_{POWER2}}{COP_{pattern2}} = 28,952(J)$$
$$P_{AC3} = \frac{P_{POWER3}}{COP_{pattern3}} = 27,284(J)$$

From the above analysis, access patten 3 saves the cooling cost of patterns 1 and 2 by 2,534 J and 1,668 J, respectively. The total energy cost, including computing and cooling energy consumption, are shown below:

$$P_{TOTAL1} = P_{POWER1} + P_{AC1} = 161,818(J)$$

$$P_{TOTAL2} = P_{POWER2} + P_{AC2} = 160,952(J)$$

$$P_{TOTAL3} = P_{POWER3} + P_{AC3} = 156,284(J)$$

We observe that access pattern 3 leads to the lowest energy. Pattern 3 makes it possible to increase CRAC temperature to lower cooling cost. This observation motivates us to propose a thermal-aware workload management that minimizes the total energy consumption by data placement optimization (see Chapter 6).

To validate the accuracy of the model, we manually measure the inlet and outlet temperatures of the node by using an infrared thermometer. We collect 20 temperature samples in each case study. We compare inlet-outlet-temperature differences obtained from our model against the real-world measurement. Table 3.3 shows that the precision-errors of our model for the three case studies are 2.28 %, 3.74%, and 4.84%, respectively. The precision is calculated by dividing an average difference between real measurement and simulation results by real measurement.

Table 3.3 Thermal Model Validation

	Case Study 1	**Case Study 2**	**Case Study 3**
Precision Error (%)	2.28	3.74	4.84

Negligible Cold-Start Phase is Insignificant

if the execution time of CPU- and I/O-intensive tasks are sufficiently long, impact of the cold-start phase becomes negligible. Now, we extend the model to consider cases where the cold-start phase can be ignored. We set the execution time of the tasks to be 60 minutes (totally 120 minutes), T_{diff} of the basic workload types are given below:

$$T_{diff1}(3600) = 3.8654(℃)$$

$$T_{diff2}(3600) = 6.2692(℃)$$

$$T_{diff3}(3600) = 4.7086(℃)$$

$$T_{diff4}(3600) = 6.8923(℃)$$

The average inlet-outlet-temperature differences under the three access patterns are:

$$T_{pattern1}(7200) = 5.4889(°C)$$

$$T_{pattern2}(7200) = 5.4889(°C)$$

$$T_{pattern3}(7200) = 5.3789(°C)$$

We can obtain the total energy costs of these cases as:

$$P_{TOTAL1} = 1,610,000(J)$$

$$P_{TOTAL2} = 1,610,000(J)$$

$$P_{TOTAL3} = 1,570,000(J)$$

The results show that compared with patterns 1 and 2, pattern 3 offer 40,000 J savings in energy.

3.4 Summary

Energy efficiency and thermal management of storage systems must be urgently addressed, because energy consumption and cooling costs of large-scale storage systems in data centers have been increasing in the past decade. Recent studies show that cooling costs contribute a significant portion of the operational cost of data centers. Thermal management techniques have been applied to reduce the energy consumption in cooling systems, thereby significantly improving the energy efficiency of data centers. Thermal models play a key role in thermal management; however, traditional thermal models for data centers do not take into account disk utilizations. In this chapter, we developed a thermal model to investigate thermal impacts of hard disks on storage systems. We showed how to apply the thermal model to estimate the outlet temperature of a storage server based on processor and disk utilizations.

The proposed thermal model offers the following two benefits. First, the model makes it possible to reduce thermal monitoring cost. Thermal management of hard disks in storage systems helps to cut cooling cost and boost system reliability. Monitoring temperatures is a key issue in thermal management techniques; however, it is prohibitively expensive to acquire and set up a huge number of sensors in a large-scale data center. Our model is an alternative to monitoring temperatures of storage systems. Second, our thermal model enables data center designers to make intelligent decisions on thermal management during the design phase.

Chapter 4

Advanced Thermal Models

In the previous chapter, we have learn that disk has comparable impact as processor on outlet temperature. In the preliminary experiments, inlet and outlet temperatures of the data node are detected by using an infrared thermometer. By monitoring the disk temperature with its inner temperature sensor, we observe that disk temperatures increase only 1-2 ℃ when disks are fully used. If the disk is not heavily loaded, no significant difference appears compared with the disk stay in idle state.

In this chapter, to achieve higher accuracy of accuracy of temperature models, we deploy external temperature sensors [6] to monitor temperature and collect the temperature data with MiniGoose [4]. We conduct several groups of experiments to study the thermal behaviour of disks and the CPU under various utilizations. Furthermore, we also investigate their impacts on the temperature of the data node.

This chapter is organized as follows. Section 4.1 states the testbed for all the experiments presented in this chapter. Section 4.2 shows approaches of modeling temperatures of a hard drive disk and a solid state disk. Section 4.3 introduces a thermal modeling approach for CPU. Section 4.4 demonstrates a outlet temperature model by taking into account of inlet temperature and workloads. Section 4.5 evaluates the thermal models by comparing the estimate values with real measurements. Finally, Section 4.6 concludes the chapter and summarizes major contributions of this chapter.

4.1 Testbed

The testbed used in this chapter is equipped with a Celeron(R) 2.2 GHz CPU, 1.0 GBytes RAM, and a 500 GBytes SATA disk. External temperature sensors and MiniGoose are applied to monitor the disk, inlet and outlet temperatures. The configuration parameters are summarized in Table 4.1.

Table 4.1 Testbed Configuration for Advanced Thermal Models

Hardware	**Software**
1 × Intel(R) Celeron(R) 450@2.2GHz 1 × 1.0 GBytes of RAM 1 × WD 500 GBytes Sata disk (WD5000AAKS-75M0A0 [8])	Ubuntu 10.04 Linux kernel 2.6.32

4.2 Thermal Models of Disks

To study the thermal characteristics of HDDs (hard drive disks) and SSDs (solid state disks), we investigate the thermal behaviours of a Western Digit hard drive disk (WD5000AAKS [8]) and an Intel SSD (SSDSA2M080G2GC [2]). The specifications of these two disks are shown in Table 4.2. The Intel SSD has faster sequential read rate than the Western Digit HDD, but slower sequential write rate. In addition, the Intel SSD consumes much less energy than the Western Digit HDD both in idle and active states.

Table 4.2 Specifications of the Two Disks

	WD5000AAKS	**Intel SSD**
Capacity(GB)	500	80
Sequential Read(MB/s)	126	250
Sequential Write (MB/s)	126	70
Power(Idle)	8.75 W	75 mW
Power(Active)	9.5W	150 mW

Throughout the rest of this section, the following four features are measured to study disk thermal characteristics in the context of cluster storage systems.

1. Steady Temperature: The temperature of a disk that stays in a steady state.

2. Temperature Increment: The difference between an initial temperature and a steady temperature when a disk is active.

3. Heat-up Time: A time interval during which a disk is heating up from its initial temperature to a steady temperature when the disk is active.

4. Cool-down Time: A time interval during which a disk is cooling down from a steady temperature to the disk's initial temperature.

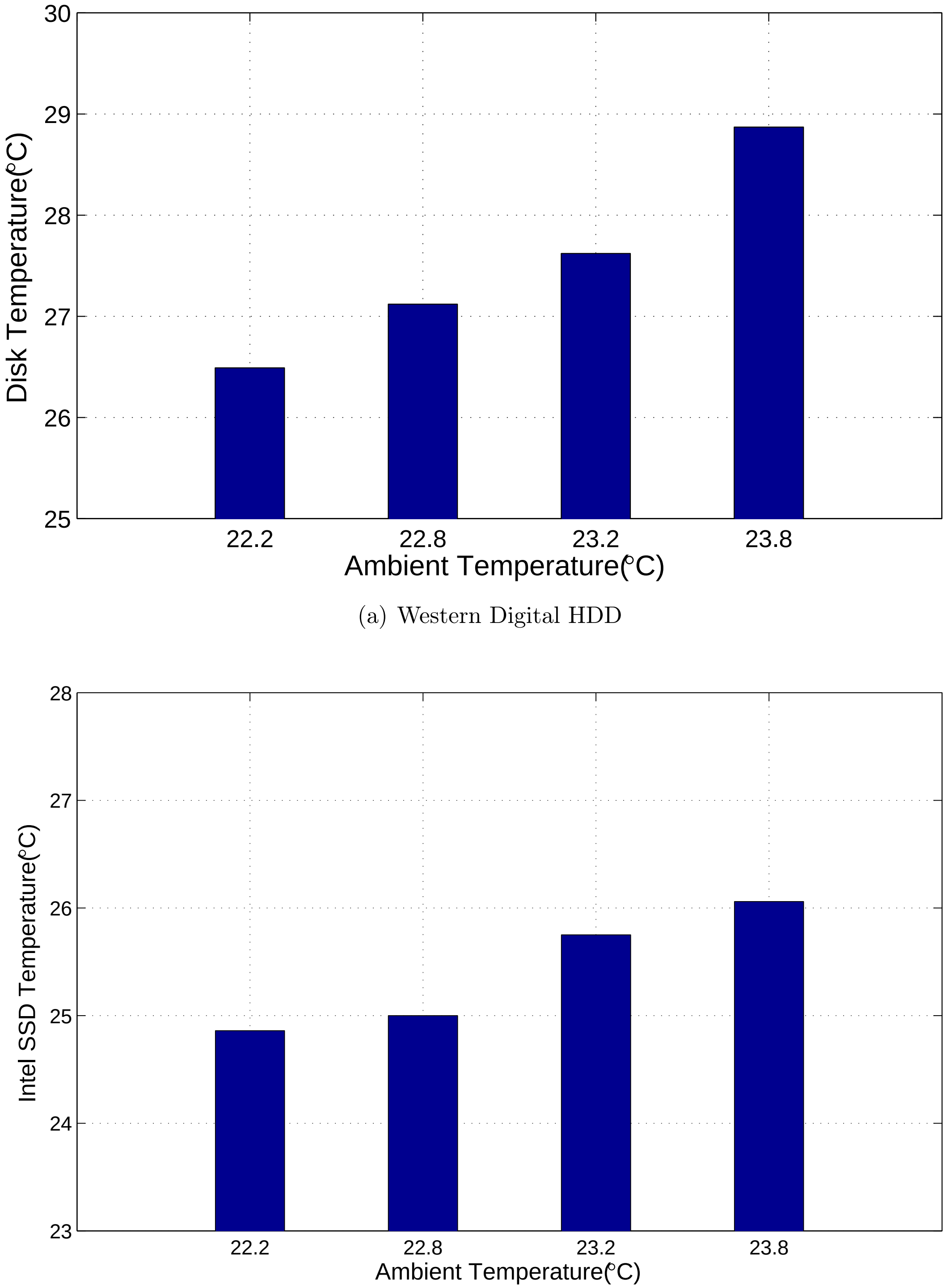
4.2.1 Ambient Impacts on Disk Temperatures

Evidence shows that ambient temperatures have impacts on processor temperatures [103]; however, little attention has been given to the impact of ambient temperatures on disk temperatures. In the first step toward the coarse-grained thermal model, we conduct a group of experiments to study the thermal impacts of ambient temperatures on disks.

Fig. 4.1 shows disk temperatures during an idle period when the computer room temperature is set to 22.2 ℃ , 22.8 ℃ , 23.2 ℃ , 23.8 ℃ . We observe that the ambient temperature does affect the temperature of the disks that are sitting idle. As shown in Fig. 4.1(a), when the ambient temperature is 22.2 ℃, the disk temperature of the Western Digital hard drive disk is 26.49 ℃. An ambient temperature of 23.8 ℃ makes the disk temperature increase to 28.87 ℃. An increase of 1.6 ℃ in ambient temperature leads to an increment of 1.97 ℃ on disk temperature. While for the Intel SSD, as shown in Fig. 4.1(b), its temperatures are 24.86 ℃ , 25.0 ℃ , 25.75 ℃ , 26.06 ℃ , respectfully. It worth noting that, in idle state, the temperature of the Intel SSD is lower than that of the Western Digital HDD under various ambient temperatures. This result suggests that ambient temperature has directly impact on the disk temperatures.

4.2.2 Various Number of Transactions

We control disk utilization by varying the number (i.e., 1000, 2000, and 5000) of I/O transactions issued by Postmark. We set the computer room temperature to 23.2 ℃ , and use Postmark to launch three I/O-intensive tasks. Each task start running when the disk is sitting idle until a steady state, with the initial disk temperature is 27.62 ℃ for the Western



(a) Western Digital HDD

(b) Intel SSD

Figure 4.1: Disk temperatures are affected by ambient temperatures.

Digital HDD and 25.75 ℃ for the Intel SSD. Table 4.3 shows the features of the three tasks. The number of files is set to 100, and file sizes are in a range between 1.E+6 and 1.E+8 Byte. All the other parameters of Postmark are set to the default values.

Table 4.3 Configurations of Tasks with Various Number of Transactions

	Task1	**Task2**	**Task3**
File Number	100	100	100
Transactions	1,000	2,000	5,000
File Size(Byte)	1.E+6 - 1.E+8	1.E+6 - 1.E+8	1.E+6 - 1.E+8

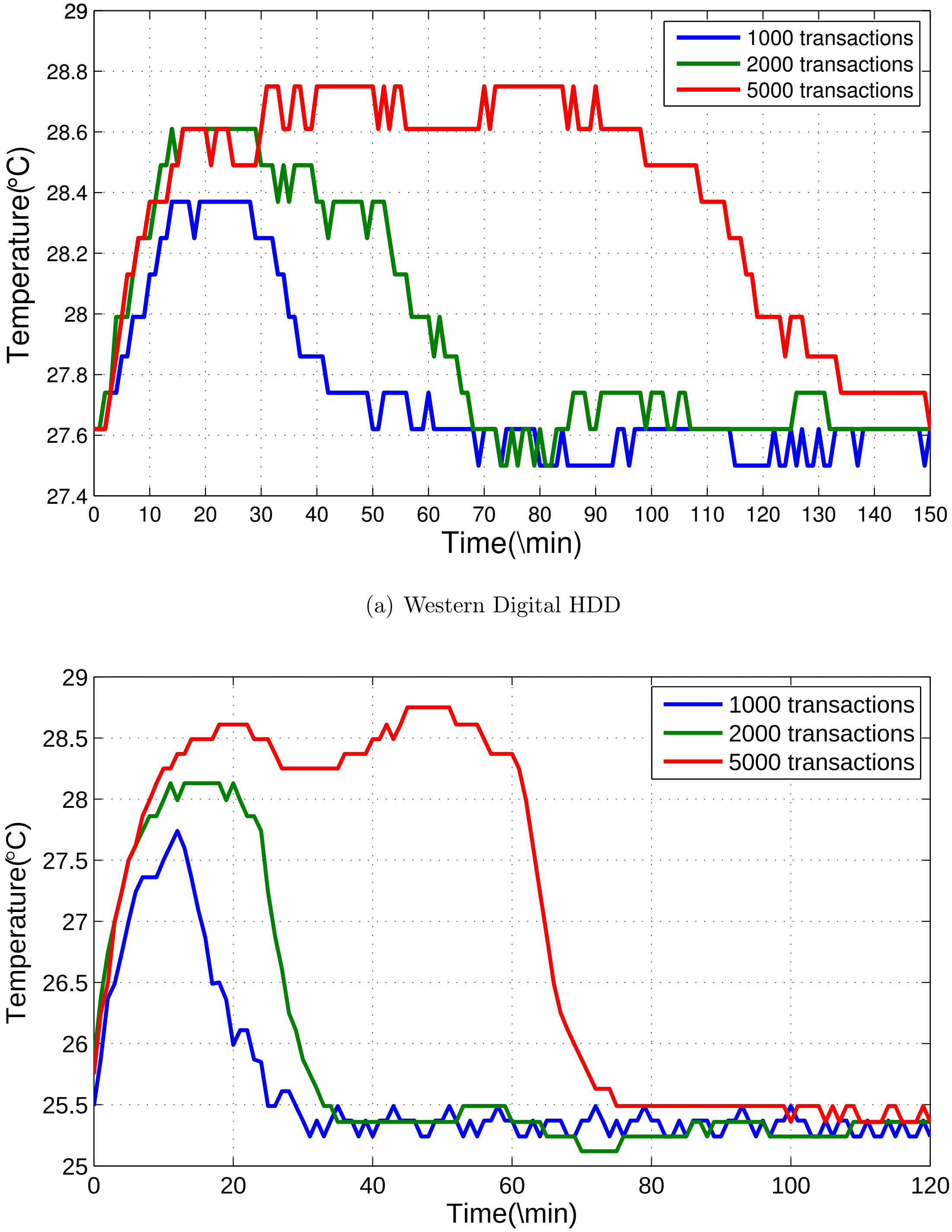
The execution time of running these three tasks on the two disks are shown in Table 4.4. We observe that, when tasks are running, the utilizations of both disks are 100%. By comparing the the execution time, we could make a conclusion that the Intel SSD performs better than the Western Digital HDD.

Table 4.4 Execution Time of Running Three Different Tasks

Disk Type	**Execution Time(s)**		
	Task1	**Task2**	**Task3**
Western Digital HDD	905	2115	5649
Intel SSD	803	1504	3733

The temperature of the two disks are shown in Fig. 4.2. The Western Digital HDD's temperatures of running these tasks are shown in Fig. 4.2(a). When assigning 5000 transactions to the hard drive disk, its peak temperature is 28.75 ℃ ; when running 2000 transactions, it peak temperature is 28.61 ℃ . We observe that disk temperature goes up gradually and it takes about 30 minutes for the disk to heat up to the peak temperature or cool down from the peak temperature to its initial temperature. And the difference between the initial temperature and the peak temperature is around 1.13 ℃ .

The experimental results of running these three tasks on Intel SSD are shown in Fig. 4.2(b). The steady temperature of the Intel SSD in idle state is around 25.75 ℃ . When it is fully utilized, its temperature goes up very fast. While running 1000 or 2000 transactions on Intel SSD, the peak temperature is not the same as running 5000 transactions. When running



(a) Western Digital HDD

(b) Intel SSD

Figure 4.2: Disk temperature of running different tasks.

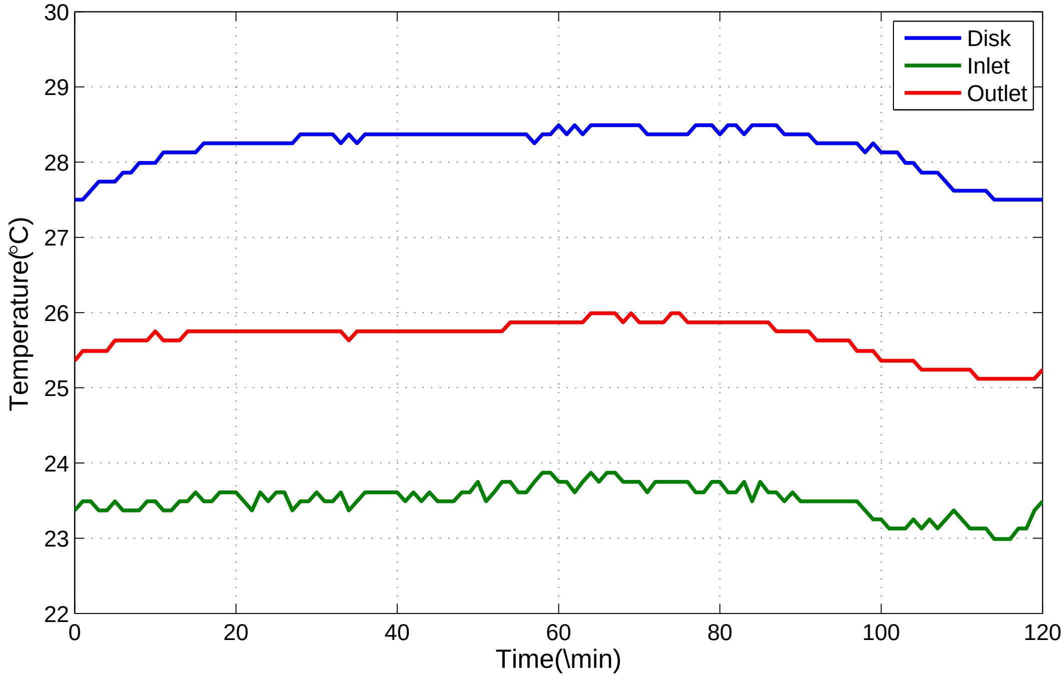
5000 transactions, the Intel SSD's temperature could be heated up to 28.75 ℃ . Compared with its initial steady temperature, there is an increment of 3.0 ℃ . Thus, when analysing the thermal characters of the Intel SSD, we would better consider the experimental results of running 5000 transactions which ensures that the disk has been heated up to its steady temperature in busy state. The Intel SSD's heat-up stage is 20 minutes, and cool-down stage is a little shorter than 20 minutes. Both of heat-up stage and cool-down stage for the Intel SSD are shorter than that of the Western Digital HDD.

A comparison of temperatures with these two disks running 5000 transactions in the chassis is shown in Fig. 4.3. We observe that inlet temperatures in both experiments keep fluctuate between 23 ℃ and 24 ℃ . The temperature of the Western Digital hard drive disk increases less than 1.5 ℃ . However, for the Intel SSD, we find a significant increase of the disk temperature for 3.0 ℃ . On average, the Intel SSD steady temperature when fully utilized is higher than that of the Western Digital HDD. In addition, Intel SSD's execution time is obviously shorter than that of the Western Digital HDD.

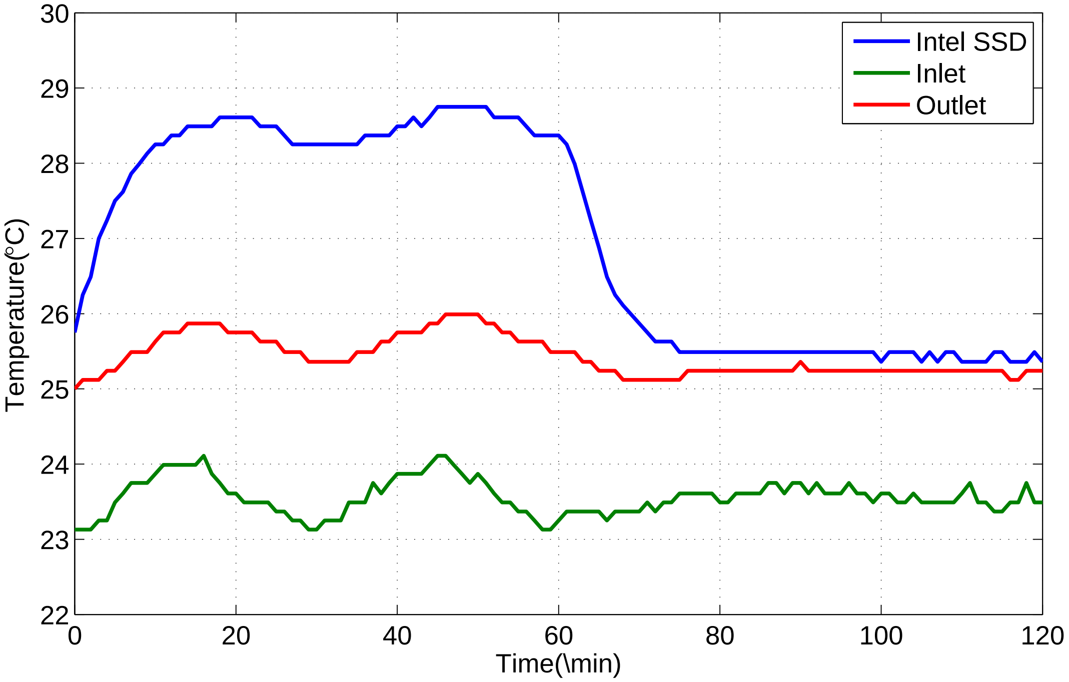
We summarize the execution time and heat-up, cool-down time of these two disks(see Fig. 4.4) to make a better comparison. We observe that Western Digital HDD costs about 42% more time than Intel SSD to finish the task, which dues to SSD's significant fast read rate. And the HDD needs more time to heat-up or cool-down than Intel SSD.

Fig. 4.5 show the comparison of temperature data for these two disks. The HDD's initial temperature is about 1.87 ℃ higher than that of Intel SSD. However, its peak temperature and steady temperature in active state are less than that of the Intel SSD. From all of the above, we conclude that Intel SSD is more sensitive to the disk activity, and it heats up and cools down faster than the Western Digital HDD.

Let us consider heat up stage (the first 30 minutes) of running $Task2$ on the Western Digital HDD. To formalize the disk thermal profile, we fit two models to the data in its heat up stage. First, we use a polynomial model to fit the disk temperature T_{disk} as a function of time t as $T_{disk}(t) = \omega * t^2 + \theta * t + \lambda$. Then we fit a logarithmic model to represent the



(a) Western Digital HDD



(b) Intel SSD

Figure 4.3: Thermal characteristics of running 5000 transactions.

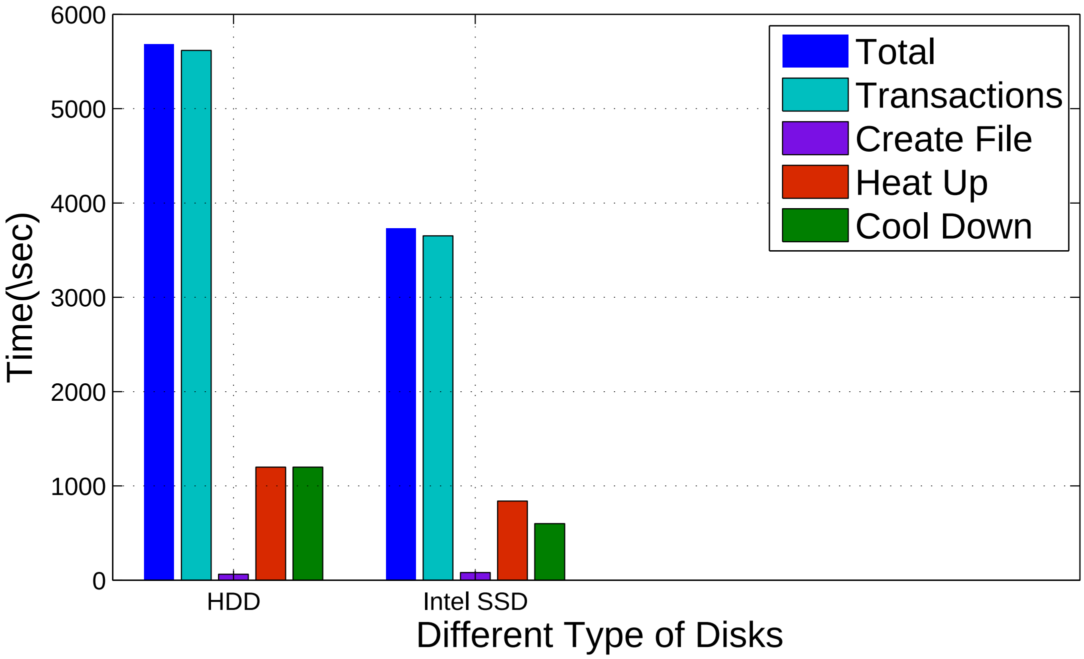


Figure 4.4: Time comparison of disks running 5000 transactions.

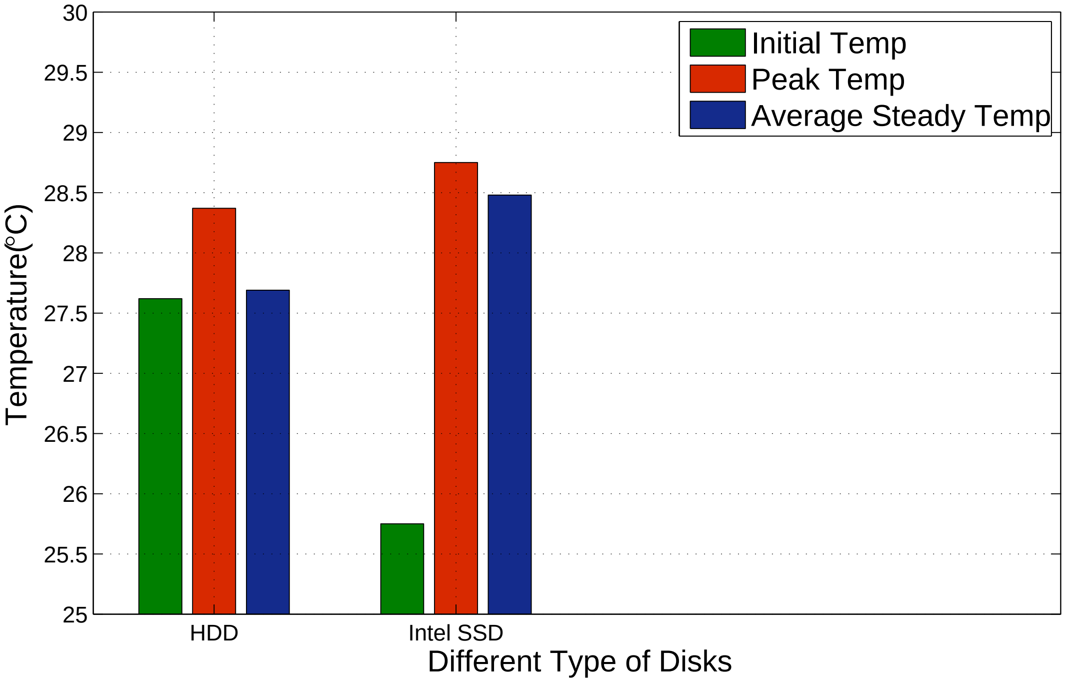


Figure 4.5: Temperature comparison of disks running 5000 transactions.

disk temperature as $T_{disk}(t) = \alpha * ln(t) + \beta$. The detailed parameters of these two models are shown in Table 4.5.

Table 4.5 Parameters for Fitting Polynomial and Logarithmic Models to Disk Temperature as a Function of Time

Disk Utilization(%)	**Polynomial Fit**			**Logarithmic Fit**	
	ω	θ	λ	α	β
100	-0.002	0.09	27.62	0.3506	27.51

To validate the accuracy of these two models, we compare temperature obtained from these models with those measured from the real-world disk. As shown in Fig. 4.6, for the heat up stage, the estimate values offered by polynomial model are very close to the real measurements with a precision error of 0.15% and standard deviation of 0.12%. And the logarithmic model gains a precision error of 0.25% and standard deviation of 0.19%.

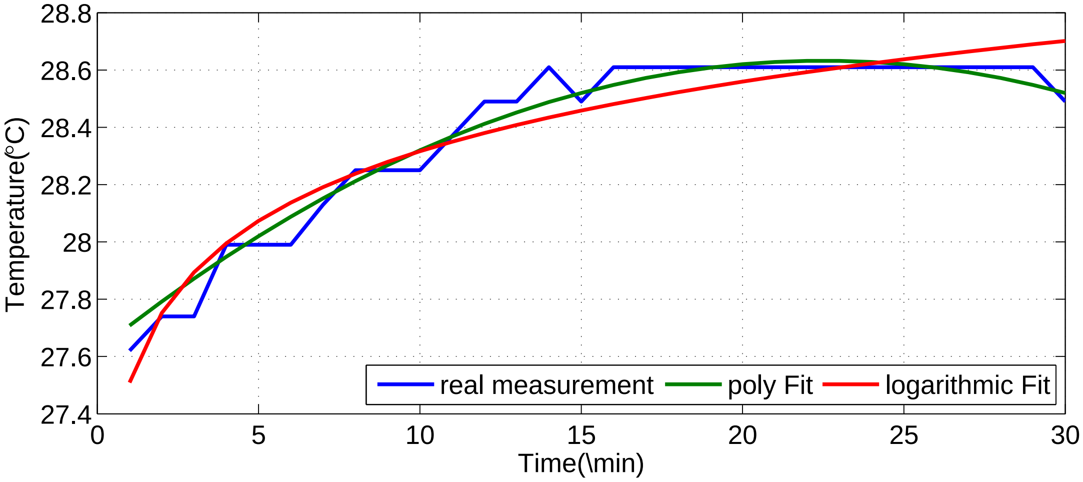


Figure 4.6: Comparison of estimated disk temperatures with real measurements.

For the steady stage (the disk running at full utilization but its temperature stays unchanged), we use a constant value 28.7 ℃ to represent the disk temperature. And for the cool down stage (disk state change from active to idle), we use the same process to model the disk temperature. And we observe that in the cool down stage polynomial model ($T_{disk}(t) = -0.0003*t^2-0.0101*t+28.61$) has a much better precision error than logarithmic model ($T_{disk}(t) = -0.2430 * ln(t) + 28.862$). Here in these two models, t represent the time in minutes that the disk stays in cool down stage. For the Intel SSD, we could apply the same approach to model its temperature in the heat up or cool down stage.

4.2.3 Disk Temperatures under Various Utilizations

We first conduct five experiments on the Western Digital HDD to study the thermal impacts of disk utilizations on disk temperatures. In each experiment, we assign one task to the disk and let the task start while the disk is sitting idle in a steady state. The ambient temperature is 23.2 ℃ and initial temperature of the disk is 27.62 ℃ . The number of files and file sizes are the same as shown in Table 4.3.

We alter disk utilization by varying the write block size and buffering setting of Postmark. If buffer is enabled, then buffered *stdio* function calls should be used instead of the lower level raw system calls [63]. All the other parameters of Postmark are set to their default values. The disk utilization is periodically assessed by the *iostat* utility program. The experiment setting are summarized in Table 4.6.

Table 4.6 Postmark Configurations of Experiments on Disks

Scenarios	1	2	3	4	5
Buffer Enabled	N	N	N	N	Y
Write Block Size(KB)	16	32	64	128	256

The utilizations and temperatures of the Western Digital HDD in these five experiments are shown in Fig. 4.7 and Fig. 4.8. Fig. 4.7 exhibits that increasing write block size leads to higher average disk utilization and shorter execution time. As shown in Fig. 4.8, the disk temperatures explore three stages (heat up, steady, and cool down stages) and large write block size results in higher disk temperature discrepancy.

The average disk utilizations in these five experiments are summarized in Table 4.7. The results indicate that we are able to generate different disk utilization by choosing an appropriate write block size with Postmark.

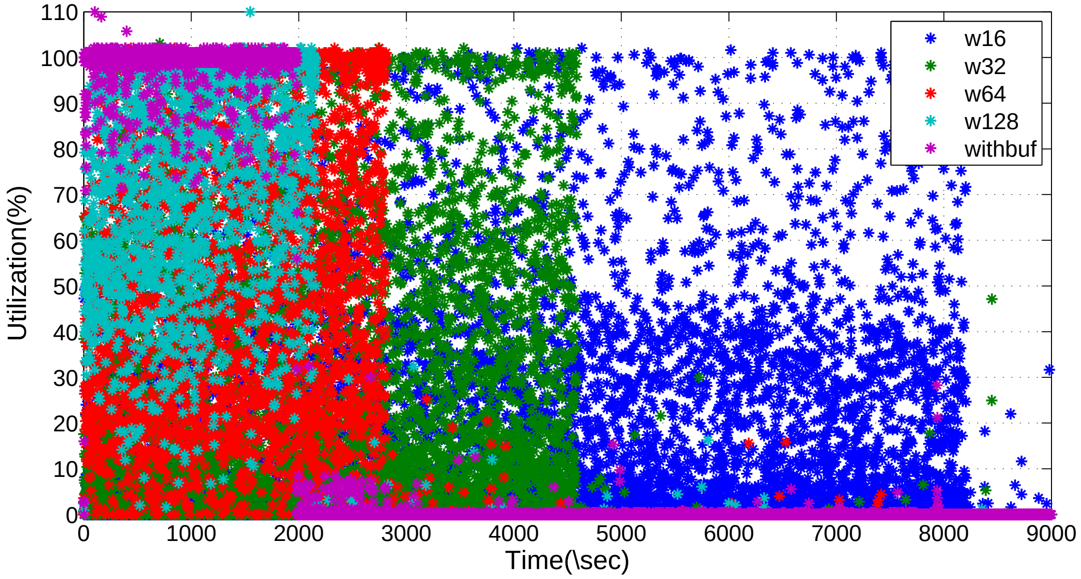


Figure 4.7: Western digital HDD's utilizations with various write block sizes.

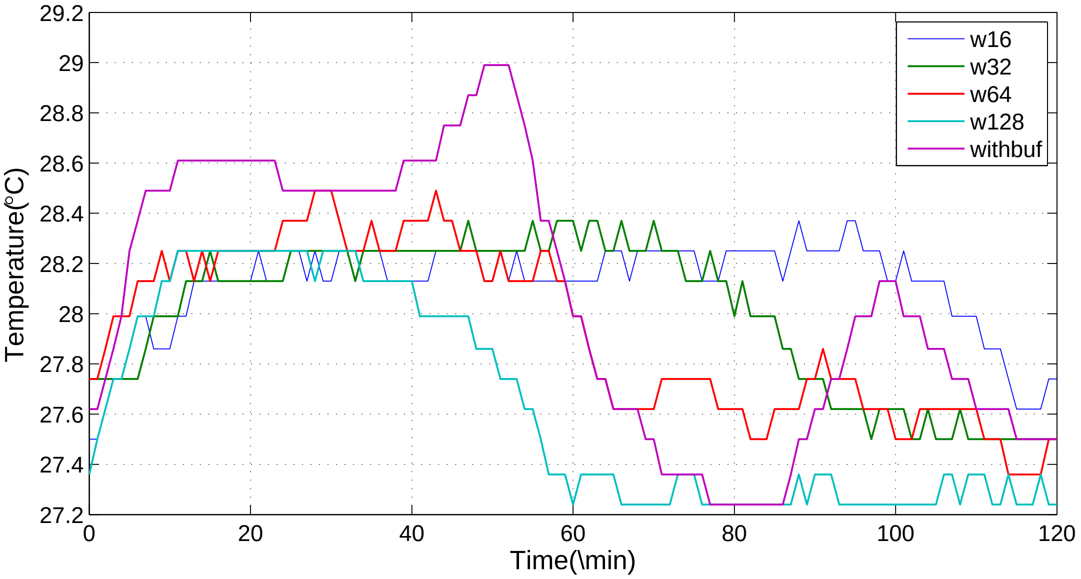


Figure 4.8: Western digital HDD's temperatures with various write block sizes.

Table 4.7 Western Digital HDD's Utilizations under Various Write Block Sizes

Scenarios	1	2	3	4	5
Average Util(%)	14.24	28.91	53.49	80.57	100

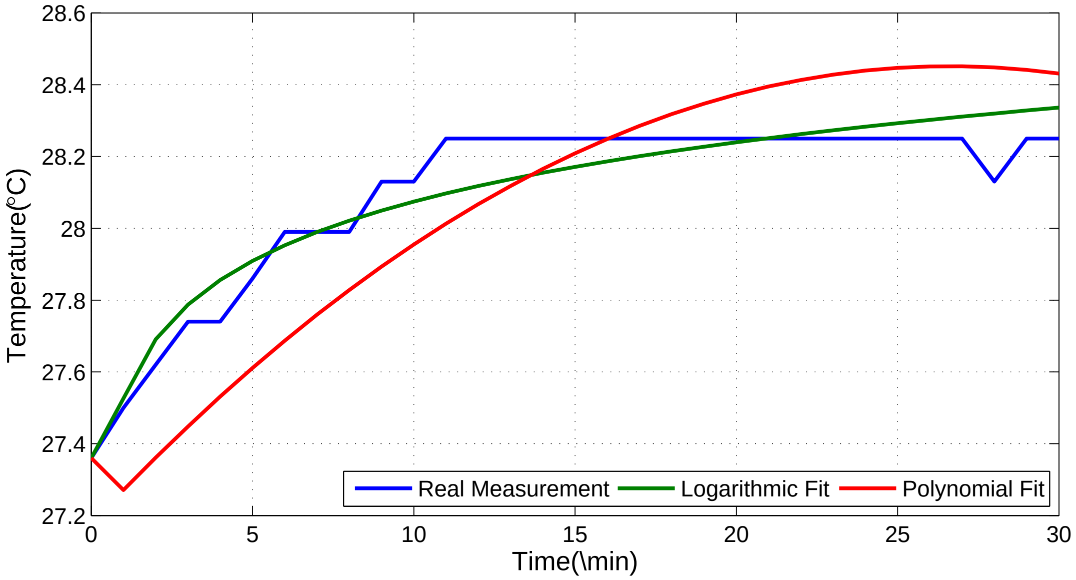


Figure 4.9: Western digital HDD's temperature model validation (write block size: 128 Byte).

We use polynomial models and logarithmic models to fit the disk temperatures during the heat up stage under different disk utilizations. A comparison of disk temperature in the heat up stage under the utilization of 80.57% estimated by these two models and the real measurement is shown in Fig. 4.9. The precision errors are 0.61% for the polynomial model and 0.21% for the logarithmic model. Here, the logarithmic model fits the disk temperature better than the polynomial model.

Our findings show that the polynomial and logarithmic models can successfully demonstrate the disk temperatures during the heat up stage under various disk utilization. Table 4.8 summarizes the important parameters determined in our parameterization process. The average precision error for polynomial fitting is 0.47%, and for logarithmic fitting is 0.20%. Hence, we conclude that the logarithmic model exhibits better curve fitting performance than that of the polynomial model for estimating disk temperatures.

Then we run five experiments on the Intel SSD with the task configurations are the same as Table 4.6. Write block sizes for this group of experiments are also set to 16, 32, 64,

Table 4.8 Parameters for Fitting Polynomial and Logarithmic Models to Western Digital HDD's Temperature under Various Utilizations

Utilization	**Polynomial Fit**				**Logarithmic Fit**		
	ω	θ	λ	err(%)	α	β	err(%)
14%	-0.0018	0.0486	27.63	1.15	0.2130	27.50	0.18
29%	-0.0007	0.0392	27.56	0.17	0.1983	27.56	0.19
53%	-0.0001	0.0257	27.68	0.27	0.1918	27.73	0.17
80%	-0.0018	0.0958	27.44	0.61	0.2382	27.53	0.21
100%	-0.0020	0.0900	27.62	0.15	0.3506	27.51	0.25

128, and 256 Bytes respectively. Without buffering, the disk utilizations are different while setting different write block sizes.

A comparison of the disk utilizations of the Western Digital HDD and the Intel SSD is shown in Fig. 4.10. The average disk utilizations for the experiments running on the Western Digital HDD are 14.24%, 28.91%, 53.49%, 80.57% while setting write block size to 16 Byte, 32 Byte, 64 Byte, and 128 Byte. And for Intel SSD, the disk utilizations are 11.00%, 30.57%, 52.90%, and 78.20%, respectively. Disk utilizations of these two disks are very close when they are set to the same write block size without buffering. And we observe that higher write block size leads to higher average disk utilization for both disks.

For the Western Digital HDD, the executing time is 8481 seconds when write block size is 16 Bytes; while the write block size is 32 Bytes, the executing time is 4760 seconds; when write block size is set to 64 Bytes, the running time of the task is 2973 seconds; and setting write block size to 128 Bytes results in a task executing time of 2313 seconds. We could draw a conclusion that larger write block size(/higher disk utilization) would result in shorter execution time. For the Intel SSD, it is also the same that larger write block size results in shorter execution time.

For these five experiments with different write block sizes, the initial temperature(/steady temperature in idle state) of the Western Digital HDD is about 28 ℃. While for Intel SSD, its initial temperature is 25.75 ℃. Under different disk utilizations, the highest temperatures that the disks stay steadily are different. Peak disk temperatures of these experiments could be summarized as Fig. 4.11. From this figure, we could observe that big write block size

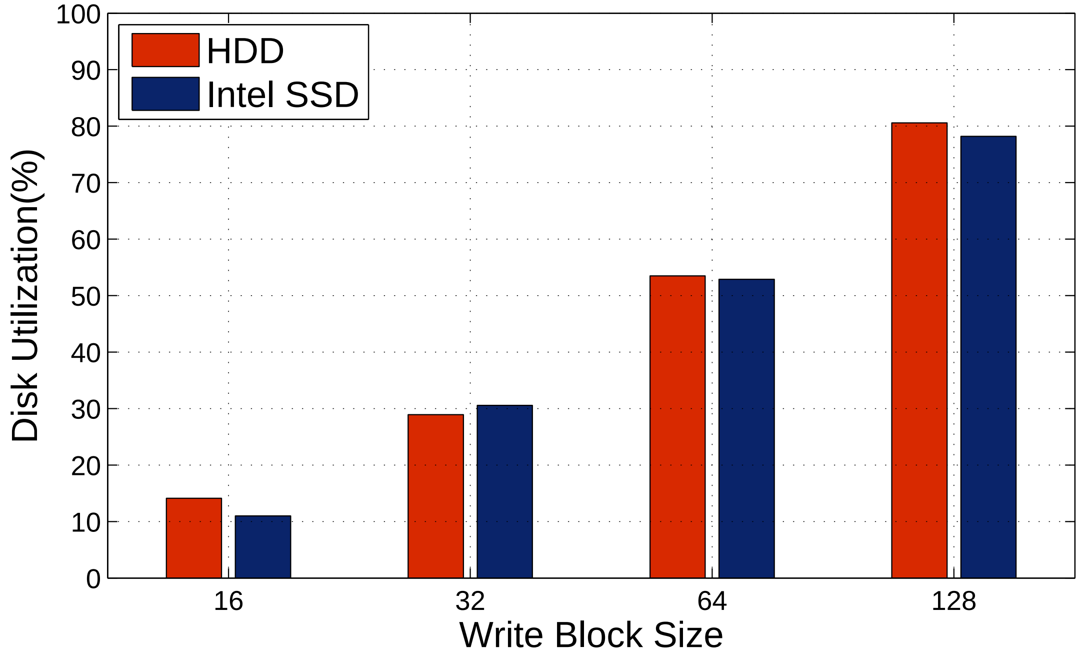


Figure 4.10: Disk utilizations under different write block sizes.

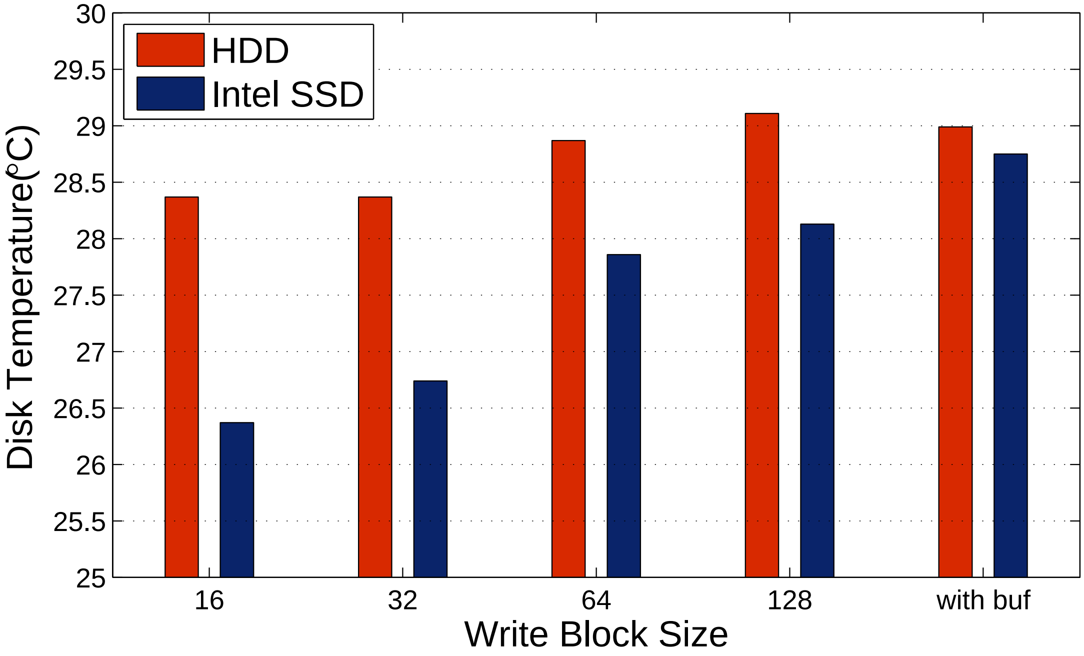


Figure 4.11: Peak disk temperatures under different write block sizes.

results in high peak disk temperature. Both the HDD and SSD scenarios share a similar trend in the sense that a large write block size leads to high disk utilization, which in turn gives rise to high disk temperature. Thus, we could have a conclusion that disk utilization has a positive impact on disk temperature.

4.2.4 Different Number of Disks

One disk may have a marginal impact on the outlet temperature of a data node; however, multiple disks have profound impact on the thermal behavior of the data node. Our goal is to investigate how do multiple disks affect outlet temperatures. The testbed used in this set of experiments includes an Intel(R) Xeon 2.4 GHz CPU, 2.0 GBytes RAM. We vary the number of disks in a data node from one to four. We test an I/O-intensive task that issues 2000 transactions on each disk; the write buffer is enabled to make disks maintain high utilization.

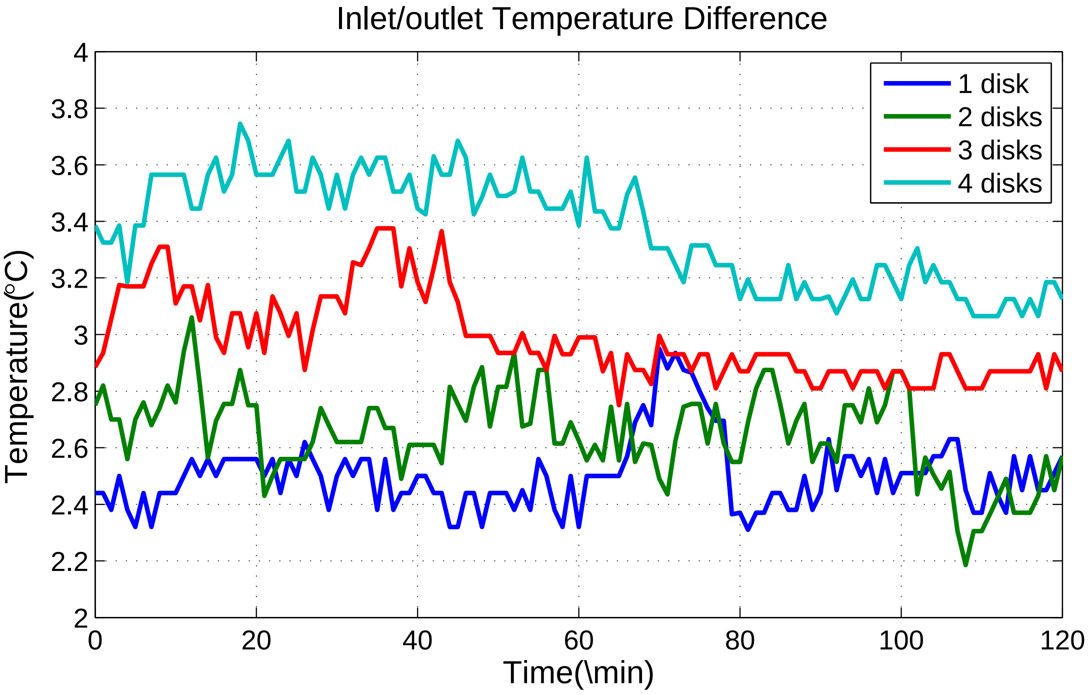


Figure 4.12: Inlet/outlet temperature differences in the cases of different numbers of disks.

Fig. 4.12 illustrates that when the disks are sitting idle, the initial differences between inlet and outlet temperature are 2.4 ℃ , 2.8 ℃ , 2.9 ℃ , and 3.4 ℃ for one, two, three, and four disks, respectively. Compared with the one-disk case, the four-disk case has a larger inlet/outlet difference. On average, a disk contributes about 0.33 ℃ increment of outlet temperature, which takes almost more than 10% of the difference between inlet and outlet temperature. If more than 16 disks are deployed in a data node, such a discrepancy between inlet and outlet temperatures will be more pronounced. The peak values of inlet/outlet temperature differences are 2.6 ℃ , 3.1 ℃ , 3.3 ℃ , and 3.7 ℃ for the one-disk, two-disk, three-disk, and four-disk cases, respectively. We conclude that increasing the number of disks in a data node can widen the gap between inlet and outlet temperatures of data nodes.

4.3 Thermal Model of CPU

We use the interior temperature sensor in CPU to monitor CPU temperature. To study the thermal behaviour of CPU, we first let the CPU remain in idle state with the CPU temperature is around 40 ℃ . Then we run whetstone – a float computation benchmark – to generate different experiment scenarios. In these scenarios, we make small modification to the original whetstone benchmark to achieve different CPU utilizations by setting various number of loops (i.e, 4000, 8000, 10000, 11900, 11950, and 12000). The CPU utilizations of these experiments are shown in Fig. 4.13, and the CPU temperatures are shown in Fig. 4.14.

As shown in Fig. 4.13, when different number of loops are set, CPU utilizations are relatively steady around specific values in the whole CPU active phases. In Fig. 4.14 shows that CPU temperatures could also be categorised into three stages: heat up stage, steady stage, and cool down stage. In the heat up stage, the CPU temperature goes up very quickly. In the steady stage, CPU temperature remains the same with the CPU running at a stable utilization. In the cool down state, CPU temperature cools down to its original temperature

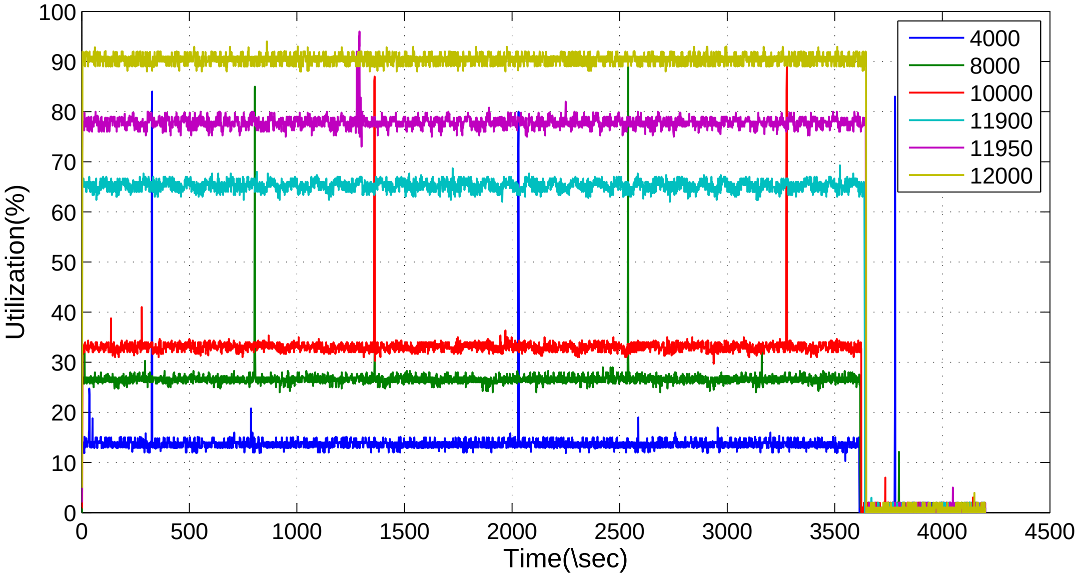


Figure 4.13: CPU utilization under different scenarios.

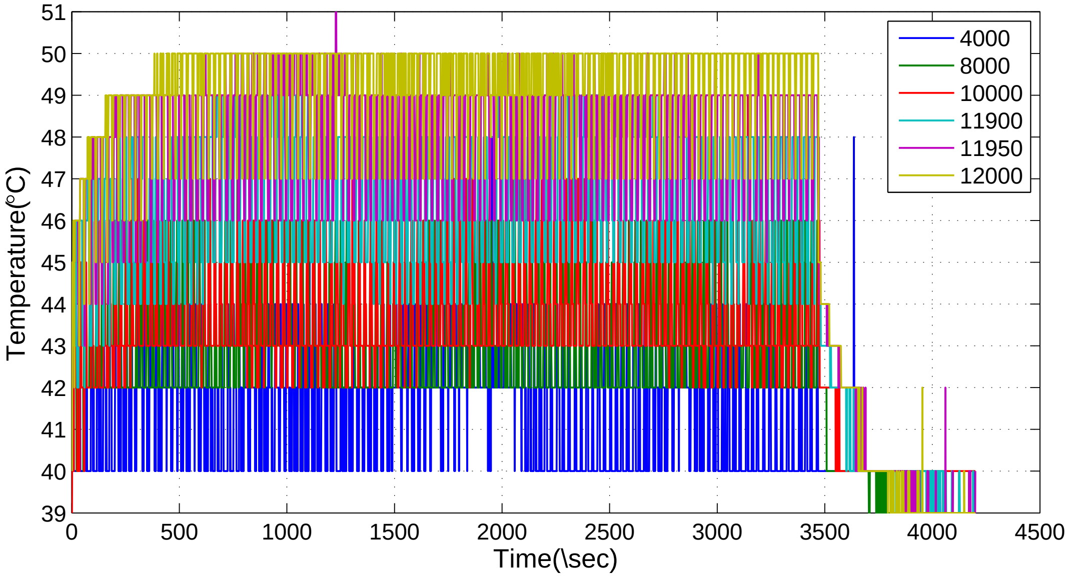


Figure 4.14: CPU temperature under different scenarios.

which is equal to the CPU temperature in idle state because CPU-intensive workload has been finished.

From the above two figures, we could conclude that increasing loop number leads to higher CPU utilization and CPU temperature. In all these experiments when CPU is active, CPU temperatures go up very quickly in the first 600 seconds (or 10 minutes), and then CPU temperature remains steady. We conclude that the heat up time for CPU is around 10 minutes, and the cool down time for CPU is less than 10 minutes. CPU was cooled down to its original temperature fast than been heated up.

Table 4.9 CPU Utilizations and Temperatures in the Steady Stage under Various Number of Loops

Scenarios	1	2	3	4	5	6
Loop Number	4000	8000	10000	11900	11950	12000
Average Utilization(%)	13.8	26.7	33.1	65.2	77.9	90.5
Average Temperature(℃)	41.3	42.9	43.7	46.7	48	49.1
Max Temperature(℃)	48	50	49	49	51	50
Min Temperature(℃)	40	40	42	44	46	47

For better comparison, we summarize the average CPU utilizations and temperatures as shown in Table 4.9. The average CPU temperatures in the steady stage are 41.3 ℃ , 42.9 ℃ , 43.7 ℃ , 46.7 ℃ , 48.0 ℃ , and 49.1 ℃ , respectively. The maximum CPU temperatures in steady stage are from 49 to 51 ℃ , and the minimum CPU temperatures in steady stage are increasing when loop number increases.

We use a polynomial model $T_{CPU}(t) = \rho * t^2 + \mu * t + \nu$ and a logarithmic model $T_{CPU}(t) = \gamma * ln(t) + \delta$ to capture the characteristics of CPU temperatures during the heat up stage under a wide range of CPU utilization. In these two models, t is the time in seconds during which CPU is running under a specific utilization. A comparison of CPU temperatures estimated by these two models and the real measurements when loop number is 12000 is shown in Fig. 4.15. The precision error of the polynomial model is 1.87%, which is higher than that of the logarithmic model (1.31%). The logarithmic model fits the curve

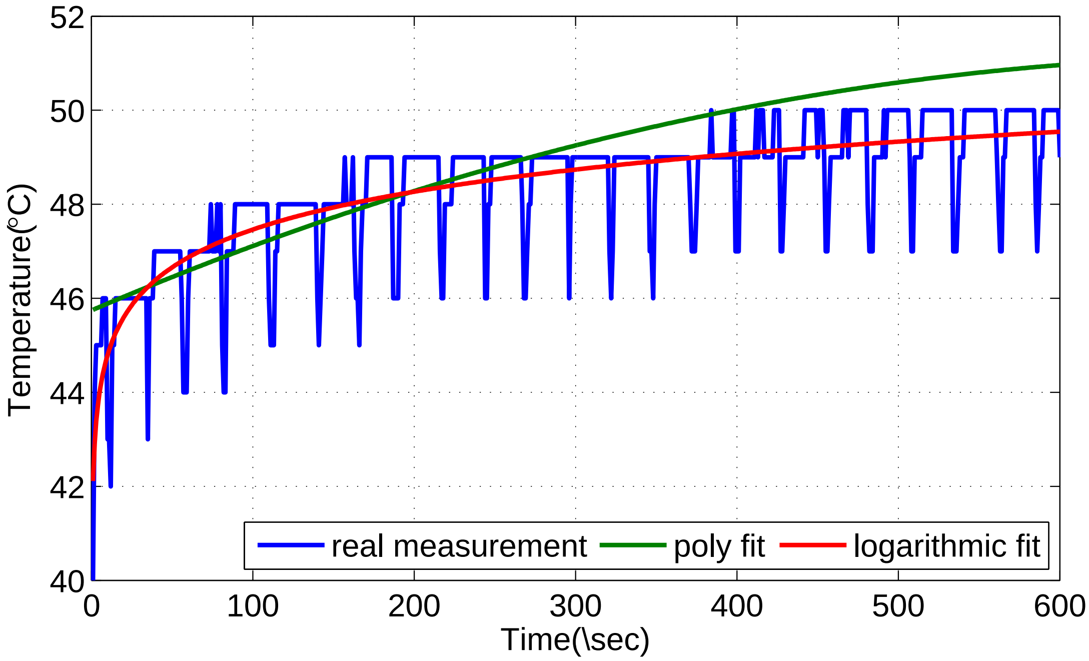


Figure 4.15: CPU temperature model validation (12000LOOPS).

of CPU temperature better than the polynomial model when CPU is in the heat up stage. For the other CPU utilizations, we also observe that logarithmic models achieve better curve fitting performance than that of the polynomial models in most cases. Thus, logarithmic models are selected for estimating CPU temperatures.

4.4 Thermal Model of a Data Node

To study the impact of CPU and disk on outlet temperature, we analyse the experiment results of running the modified whetstone benchmark with 4000 iterations. We use a linear model $T_{diff} = a + b * x + c * y$ to demonstrate the discrepancy between inlet and outlet temperatures. In this model, T_{diff} is the outlet temperature, x is the CPU temperature, and y is the disk temperature. The parameters are shown in Table 4.10. And through $T_{outlet} = T_{inlet} + T_{diff}$, the outlet temperature of a data storage node can be estimated.

Fig. 4.16 shows the comparisons between the estimated temperatures and measured ones after running the whetstone benchmark. The validation results confirm that the model

Table 4.10 Parameters for the Linear Model to Estimate Outlet Temperatures as a Function of CPU and Disk Temperatures.

	a	b	c
Linear Fit	4.842	0.0773	-0.2232

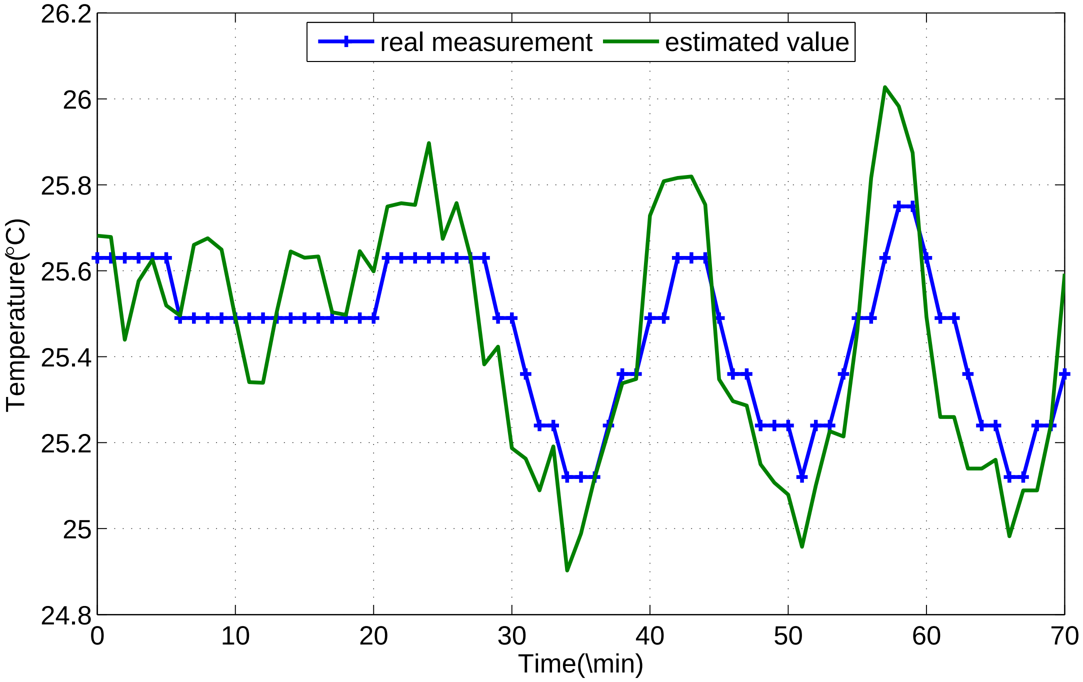


Figure 4.16: Validation of the outlet temperature model.

can be successfully applied to estimate outlet temperatures derived from CPU and disk temperatures. The precision error of this model is as low as 0.5%. We also validate the result of running the whetstone benchmark with other number of iterations; the precision errors of the other experiment results are all below 0.5%.

4.5 Evaluation of Temperature Models

To verify the CPU, disk and outlet temperature models, we conduct an experiment by running the WordCount benchmark on a given folder. This folder is composed of files randomly generated by Postmark and locates in the Western Digital HDD. All these files sum up to 10 GB. As shown in Fig. 4.17, the CPU and disk utilizations are relatively steady when the benchmark is running. The average CPU utilization is 92.48%, and the average disk utilization is 18.60%.

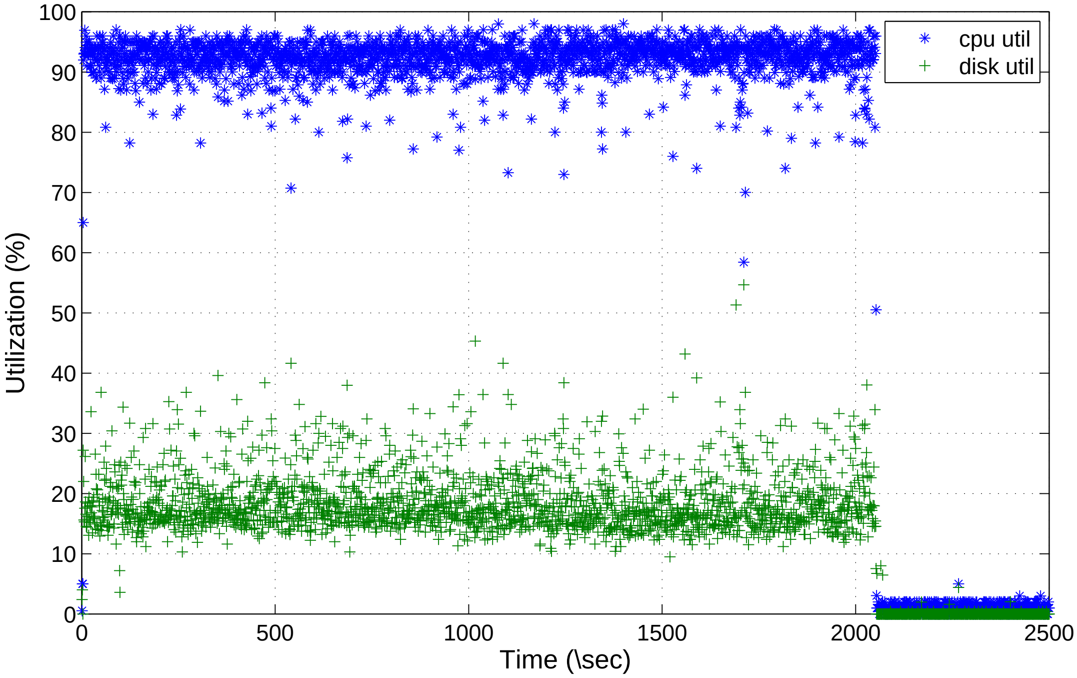


Figure 4.17: CPU and disk utilizations for running WordCount.

Now we demonstrate a way of applying our proposed models to estimate disk temperature. Three main steps are involved to derive estimated disk temperatures for a specific disk utilization:

1. to choose several time stamps and record disk temperatures under different disk utilizations;

2. to build a disk temperature model as a function of disk initial temperature, ambient temperature, disk utilization; and

3. to estimate the disk temperature under a specific utilization using the model built in the above step.

The above procedure allows us to estimate disk temperatures using ambient temperatures and disk utilization.

The following shows the details of the disk-temperature estimation procedure. We obtain the disk temperatures from our preliminary experiments when the disk utilization is 14.24%, 28.91%, 53.49%, 80.57%, and 100%, respectively. Then, six time stamps (i.e., 5, 10, 15, 20, 25, 30 minutes) are chosen and the disk-temperature equations are applied to estimate the disk temperature under each time stamp. With the estimated temperature data, we apply the logarithmic model to fit the disk temperature at these six time stamps when the disk utilization becomes 18.60%. The results are summarized in Table 4.11.

Table 4.11 Estimated Disk Temperatures under a Specific Utilization

Time	**Real Measurement**					**Estimation**
	14.24%	**28.92%**	**53.49%**	**80.57%**	**100%**	**(18.60%)**
5	27.83	27.88	28.04	27.92	28.20	27.85
10	27.99	28.02	28.17	28.07	28.39	28.00
15	28.07	28.09	28.25	28.17	28.50	28.07
20	28.14	28.15	28.30	28.24	28.58	28.13
25	28.18	28.19	28.35	28.29	28.64	28.17
30	28.22	28.23	28.39	28.34	28.69	28.21

Using the estimated disk temperatures under disk utilization of 18.60%, we derive the disk-temperature model as follow:

$$T_{disk}(t) = 0.2 * ln(t) + 27.53, \tag{4.1}$$

The above model can be used to predict disk temperatures during the heat up stage when the disk utilization is 18.60%. According to the same process, we could generate the model for estimating the CPU temperature under the utilization of 92.48% as follow:

$$T_{CPU}(t) = 1.27 * ln(t) + 42.01, \tag{4.2}$$

The comparison of estimate CPU and disk temperature with the real measurements are shown in Fig. 4.18 and Fig. 4.19.

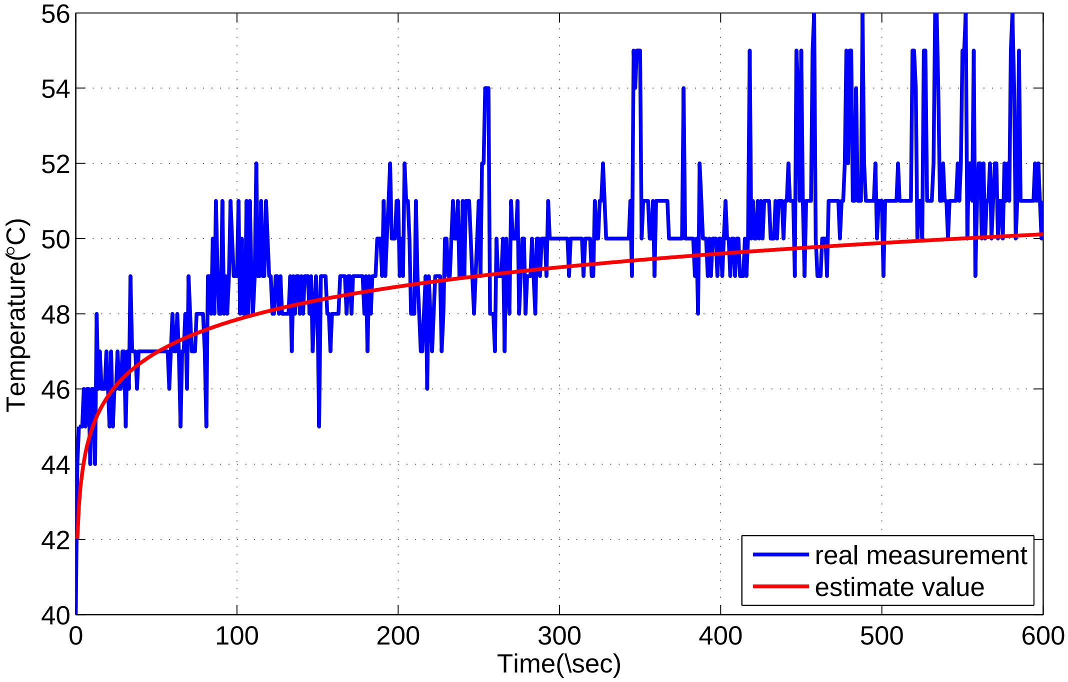


Figure 4.18: CPU temperature model validation for WordCount.

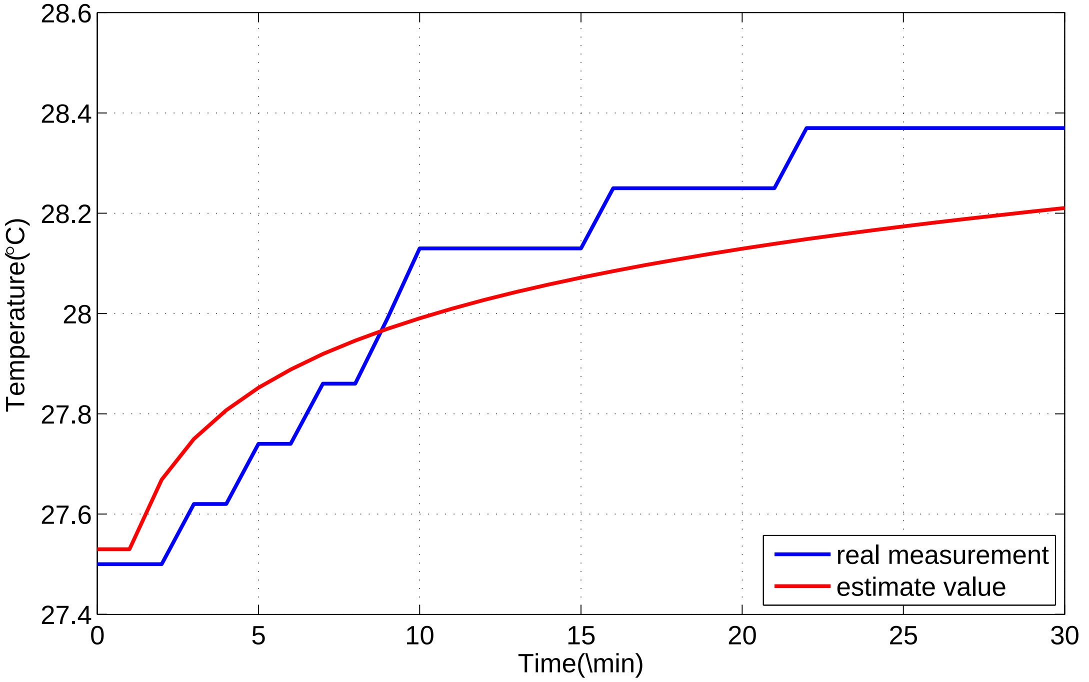


Figure 4.19: Disk temperature model validation for WordCount.

The precision of the CPU and disk models are 1.52% and 0.48%. Then, we apply the same outlet temperature model to this experiment, and we found that the outlet temperature gains a precision error of 3.77%.

4.6 Summary

The goal of our study is to build a thermal model to estimate the outlet temperature of a storage server (a.k.a., data node) based on processor and disk utilizations. Thermal models play a key role in thermal management; however, traditional thermal models for data centers do not take into account disk utilizations. In this chapter, we developed a thermal model to investigate thermal impacts of hard disks on data nodes in storage clusters. Our thermal models were developed at a coarse-grained level without the knowledge of detailed specification of data nodes. Our experimental results show that our modeling approach could predict the temperature of both disk and CPU with high accuracy. Furthermore, we presented how to apply the thermal model to estimate the outlet temperature of a storage server under certain processor and disk utilizations.

In this chapter, We make the following contributions:

1. we generated the thermal profile of a storage server. The profiling results are obtained by running I/O-intensive workloads imposed by Postmark and CPU-intensive workloads by running Whetstone. When the disk and CPU are running under various load scenarios, we monitor their temperatures as well as the inlet and outlet temperatures of the data node with temperature sensors.

2. we built a thermal modeling approach for estimating temperatures of CPU and disk under giving workloads.

3. we built an outlet temperature model by considering the thermal impacts of inlet temperature, CPU and disk temperatures.

Our method enables data storage systems to save thermal monitoring costs. In addition, our thermal models enable data center designers to make intelligent decisions on thermal management during the design phase. Thermal management of storage systems helps to cut cooling costs and boost system reliability. Monitoring temperatures is a key issue in thermal management techniques; however, it is prohibitively expensive to acquire and set up a huge number of sensors in a large-scale data center. Our modeling method is an alternative to monitoring temperatures of storage systems.

Though most of the experiments in this chapter were conducted under the ambient temperature of 23.2 ℃. Our proposed approach can be applied to a data storage environment with various CRAC supply temperatures. Thus, when the CRAC supply temperature changes, we may need to conduct the profiling experiments, which allow us to assign specific parameter values to our model.

Chapter 5

Thermal-aware Task Scheduling

Now we propose a thermal-aware energy-efficient task scheduling system, where task schedulers are introduced for dispatching incoming workloads. As has been verified in Chapter 3 that scheduling tasks of computing and read/write in parallel could save energy than the other two patterns. Thus, in our task scheduling system, we keep CPUs and disks as busy as possible. The system consists of two components: a centralized thermal-aware task scheduler that maintains a global task waiting list and a candidate node list that contains the data nodes that are not fully utilized; and sub-schedulers that are installed in every data node to maintain tasks assigned to them. The centralized task scheduler is responsible for dispatching workloads according to properties of the tasks. In the process of task scheduling, thermal issues are considered to avoid hot spots in data centers.

This chapter is organized as follows. First of all, Section 5.1 introduces the framework of our thermal-aware task scheduling system. Then, the performance and efficiency of our task scheduling system are presented in Section 5.2. Finally, Section 5.3 concludes the contribution of our thermal-aware task scheduling system.

5.1 Framework

The framework of our management system for task scheduling is shown in Fig. 5.1. It shows a data storage system with n nodes, and on top of the storage system, a thermal-aware task scheduling system manages the workload assigned to the storage system. On each data node, we deploy a sub-system, in which a monitor is applied to detect the utilization and temperature of the components in this data node. Our system schedules the workload so that the components (i.e., CPU, disk, and etc) work as hard as possible with the outlet temperature does not exceed a threshold.

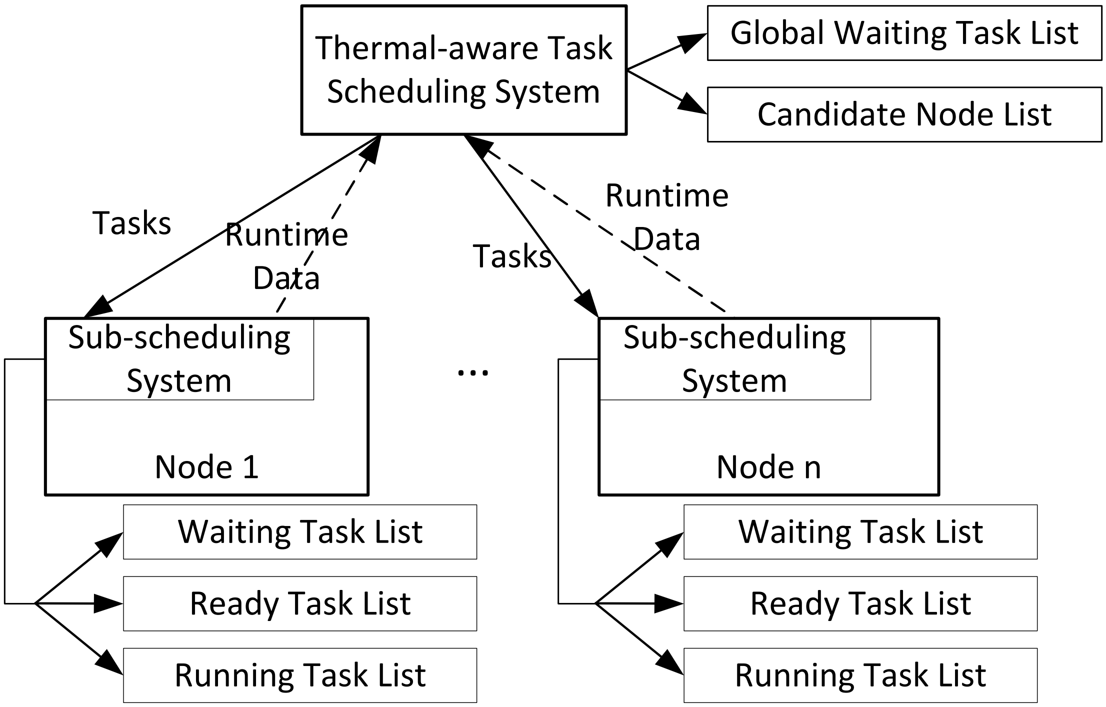


Figure 5.1: The framework of thermal management system for task scheduling.

The thermal-aware task scheduling system maintains two lists:

- a global waiting task list,
- a candidate node list.

The global waiting task list holds the tasks assigned. The candidate node list maintains the data node that are in idle state or relatively light loaded.

In the global task list, the tasks are arranged in coming time ascending order. The scheduling systems monitor the behaviours of all data nodes, and assign the tasks in global task list to the candidate data nodes. Before assigning a task to a candidate data node, the runtime information is fetch from the monitor of the data node, and temperature models are applied to estimate the thermal impact of the task on the candidate data node. The task would be assigned to this candidate data node if it would not introduce hot spot.

The sub-scheduling system also maintains three task lists on each data node:

- a waiting list,
- a ready list,
- and a running list.

The waiting list holds the tasks that must be executed in this particular data node, the ready list holds the tasks that would be executed immediately. The running list holds the tasks that are running on the data node. The sub-scheduling system manges these three task lists. It maintains the runtime information of the running tasks, and lunches tasks from the ready list on each node. While the ready list is empty, it introduces tasks from the waiting list when these tasks would not lead the outlet temperature exceed a threshold.

Whenever a new task comes, the scheduling system will first check if the task is node-relative. Here, a node-relative task refers to a task that need be executed on a particular data node which hold the related data to complete the task.

If the task is not node-related, the task would be pushed into the global waiting task list. And then the system will dispatch the tasks in the global waiting task list to the candidate data nodes. When a task in the global waiting list is dispatched to a data node, this task would be moved to the ready list of this data node.

If the task is node-related, system will check the monitor information from the destination data node, and estimate the CPU and disk utilization that the new task will lead to. With the thermal models introduced in the previous section, how the outlet temperature would be impact could be estimated. If the outlet temperature will not exceed the threshold, the task would be put into the ready list of the data node, and been executed immediately. However, if the outlet temperature is estimated to exceed the threshold, the new task will be added to the waiting list of the data node. Until the system find that the new task will not drive the outlet temperature to exceed the threshold, the task will be moved to the ready list.

If the waiting list of a data node is too long that the tasks could not be finished in an expected time period, the system will choose some candidate data nodes, and move some of these tasks in the waiting list of the current data node to the candidate data nodes. When determining which task to move in the waiting list, some rules are applied:

1. choose CPU-intensive task first, and then I/O-intensive task;

2. choose the task whose associated data could also be accessed in the candidate data nodes;

3. choose the task that the size of its associated data is smaller than other tasks.

After determining which task to be moved and where the task should be moved to, scheduling system checks if the destination data node have the relative data. If the destination data node has the required data, the task could be moved directly to the waiting list of the destination node. If it has't, then a new task will be generated to move the data from the original data node to the destination data node. After the data movement, the task will then be moved to the destination data node's waiting list.

5.2 Experiments

To evaluate the performance of our task scheduling system, we conduct two groups of experiments, which resemble various real-world workload scenarios. Table 4.1 shows the parameters of a small-scale storage cluster of four data nodes. And throughout these experiments, we set the outlet temperature threshold for each data node to 27 ℃.

For tasks without any preferred data nodes, it is flexible for our task scheduler to dispatch the tasks to any candidate nodes. While selecting the best candidate data node to assign tasks, the scheduler should address the following issue. The scheduler may assign tasks to the least loaded data nodes or data nodes with the highest utilization. For comparison purpose, we consider the following three scheduling policies:

- Distribute Evenly (DE): to evenly schedule tasks to all the data nodes in the first-in-first-out order, thereby well balancing load among the nodes.

- Distribute based on Utilization (DU): to schedule tasks to as many as data nodes while keeping active nodes' utilization at a high level.

- Distribute to Minimum Active Nodes(DMN): to schedule tasks in a way to minimize the number of active data nodes.

5.2.1 CPU-intensive Workload

In the first group of experiments, we consider CPU-intensive workload. A total of ten CPU-intensive tasks are running Whetstone on the cluster. These CPU-intensive tasks lead to various CPU utilizations. The configuration and average utilization for each task are summarized in Table 5.1.

Table 5.1 Task Configurations of CPU-intensive Workloads

Tasks	**Task 1**	**Task 2**	**Task 3**	**Task 4**	**Task 5**
LOOPS(#)	4000	8000	10000	11820	11850
Avg Util(%)	13	25	32	44	52
Tasks	**Task 6**	**Task 7**	**Task 8**	**Task 9**	**Task 10**
LOOPS(#)	11900	11930	11980	12020	12050
Avg Util(%)	64	72	85	96	100

Let us consider a baseline task scheduler that assigns all tasks to a single data node, thereby making use of the least number of active data nodes. We conduct experiments to assign all the ten tasks to one of the four available data nodes, and the tasks are sequentially executed on the node. The average time to complete the ten tasks scheduled by this baseline approach is 6131 seconds.

Table 5.2 lists the three task scheduling strategies under the CPU-intensive workload conditions. The DE strategy evenly assigns tasks to the four data nodes. For instance, on data node 1, tasks 1 and 5 are concurrently executed; task 9 is running after the completion

Table 5.2 Task Scheduling Schemes for CPU-intensive Workloads.

Strategies	**Node 1**	**Node 2**	**Node 3**	**Node 4**
DE	Task 1, 5, 9	Task 2, 6, 10	Task 3, 7	Task 4, 8
DU	Task 1, 8, 9	Task 2, 7, 10	Task 3, 6	Task 4, 5
DMN	Task 1, 2, 3, 8	Task 4, 5, 9	Task 6, 10	Task 7

of task 1 and 5. When the DU strategy is in charge of the scheduling, tasks 1 and task 8 are executed simultaneously on node 1, where the CPU utilization is as high as 98%. After completing tasks 1 and 8, node 1 start running task 9. With the DU strategy in place, each node keeps a high CPU utilization, while ensuing that its CPU is not overloaded. When it comes to the DMN strategy, new tasks are scheduled to minimize the number of active data nodes. Thus, tasks 1, 2, and 3 are all assigned to data node 1.

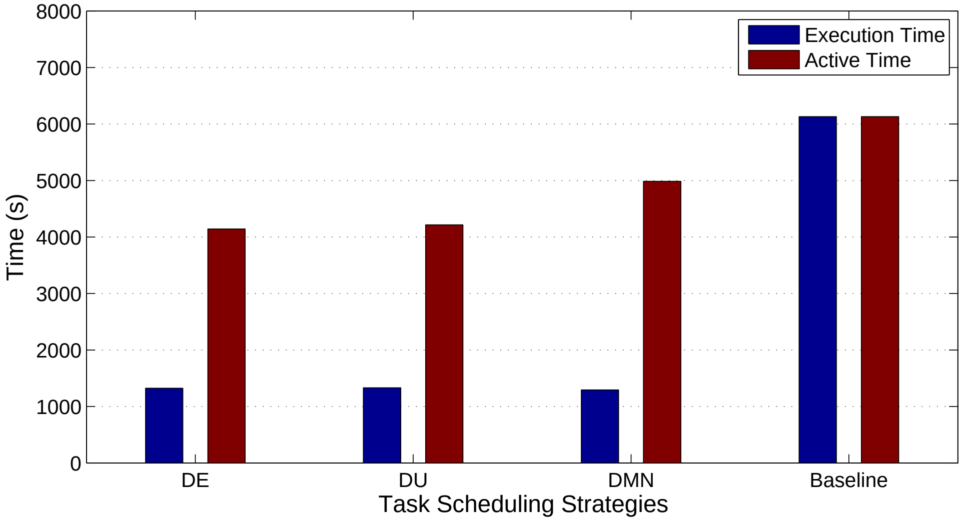


Figure 5.2: Execution time and active time of data nodes under CPU-intensive Workloads.

Fig. 5.2 reveals the performance of the three scheduling strategies. Execution times are referred to the time spent in completing all submitted tasks; active times are defined as the accumulation of time intervals in which the four data nodes are staying in the active state. Experimental results show that the outlet temperatures of the data nodes do not exceed the specified threshold.

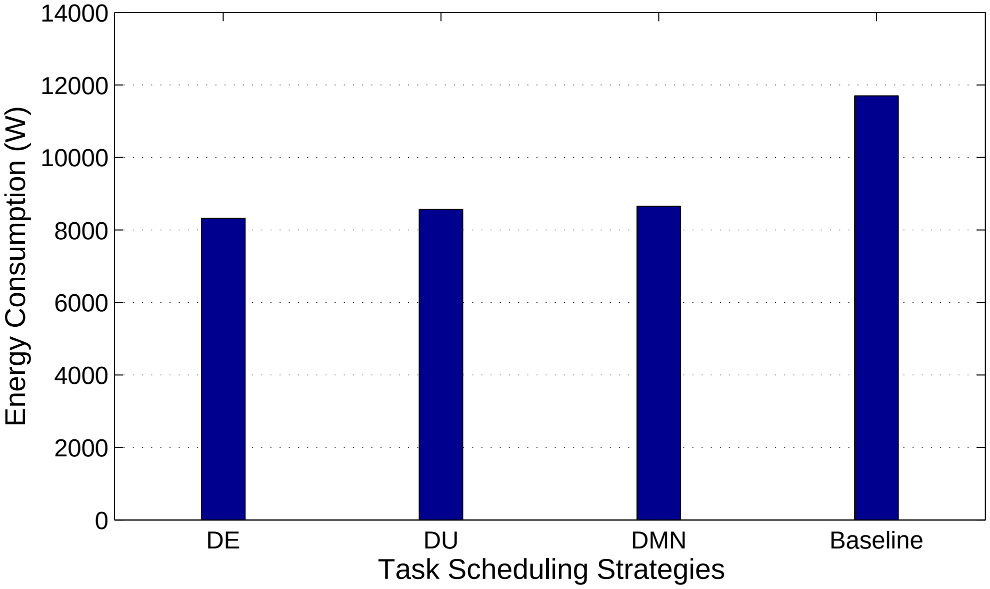


Figure 5.3: Energy consumption of data nodes under CPU-intensive Workloads.

Fig. 5.3 compared the baseline scheme with the three evaluated strategies in terms of energy consumption.

Among all the four scheduling strategies, the baseline one exhibits the longest execution time and consumes the most energy. By comparing the three evaluated strategies, the DMN strategy achieves the best performance, whereas DE delivers the highest energy efficiency. For example, DE saves the energy consumption of the other strategies by 3.8%, and DE also conserves the energy consumption of the baseline scheme by 28.9%. Thus, we could conclude that the DE strategy is the best scheduler for CPU-intensive load on storage clusters.

5.2.2 I/O-intensive Workload

In the second group of experiments, we assigned ten I/O-intensive tasks to the cluster. Each task generates 50 files and issues 200 transactions. We change the write block size to vary the disk utilization of each data node. The characteristics of these I/O-intensive tasks are shown in Table 5.3.

Table 5.3 Task Configurations of I/O-intensive Workloads.

Tasks	**Task 1,2**	**Task 3,4**	**Task 5,6**	**Task 7,8**	**Task 9,10**
Write Block Size(Byte)	16	32	64	128	256
Avg Util(%)	14	29	54	81	100

A baseline scheme assigns all the tasks to a single data node. We compare the aforementioned scheduling strategies with the baseline one. Table 5.4 shows the three task scheduling schemes.

Table 5.4 Task Scheduling Schemes for I/O-intensive Workloads.

Strategies	**Node 1**	**Node 2**	**Node 3**	**Node 4**
DE	Task 1, 5, 9	Task 2, 6, 10	Task 3, 7	Task 4, 8
DU	Task 1, 8, 9	Task 2, 7, 10	Task 3, 6	Task 4, 5
DMN	Task 1, 2, 3, 4	Task 5, 6	Task 7, 9	Task 8, 10

Fig. 5.4 shows the performance of the evaluated scheduling strategies. The results reveal that regardless of the tested schedulers, the outlet temperatures are kept below the pre-defined threshold. And the energy consumption of the cluster managed by the three strategies compared with the baseline one can be found in Fig. 5.5.

Not surprisingly, the baseline strategy is outperformed by the three other schedulers in terms of execution time and energy consumption. The utilization-based scheduler is superior to the other three schemes in performance. The most energy efficient scheduler is the one (i.e., DE) that evenly distribute the load across all the four data nodes; this energy-efficient scheduler save the energy consumption of the baseline and the other schemes by 10.8% and 3.4%, respectively. Again, DE is the best scheduler for I/O-intensive workload.

In summary, under both CPU-intensive and I/O-intensive workload conditions, evenly distributing load across active data nodes is very energy efficient.

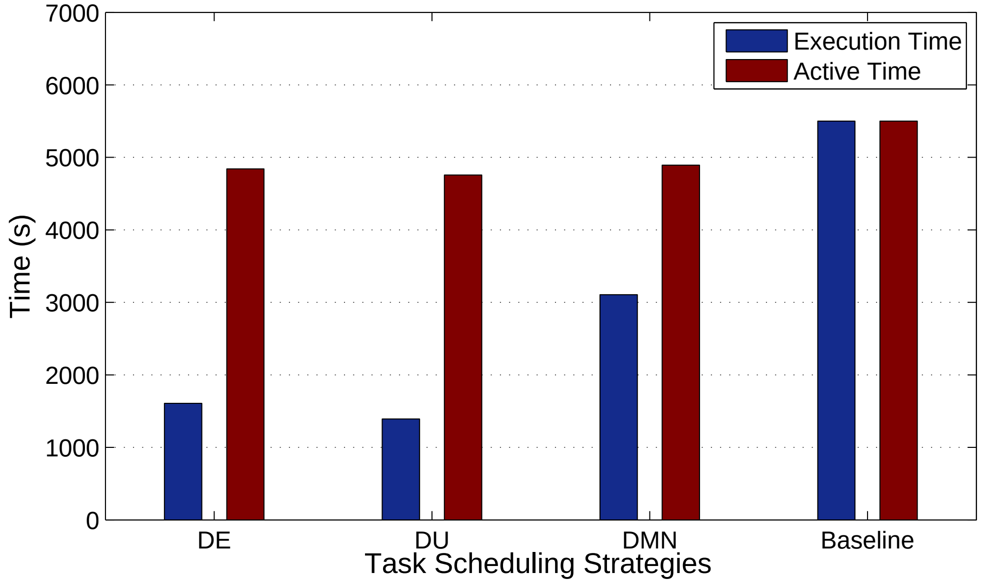


Figure 5.4: Execution time and active time of data nodes under I/O-intensive workloads.

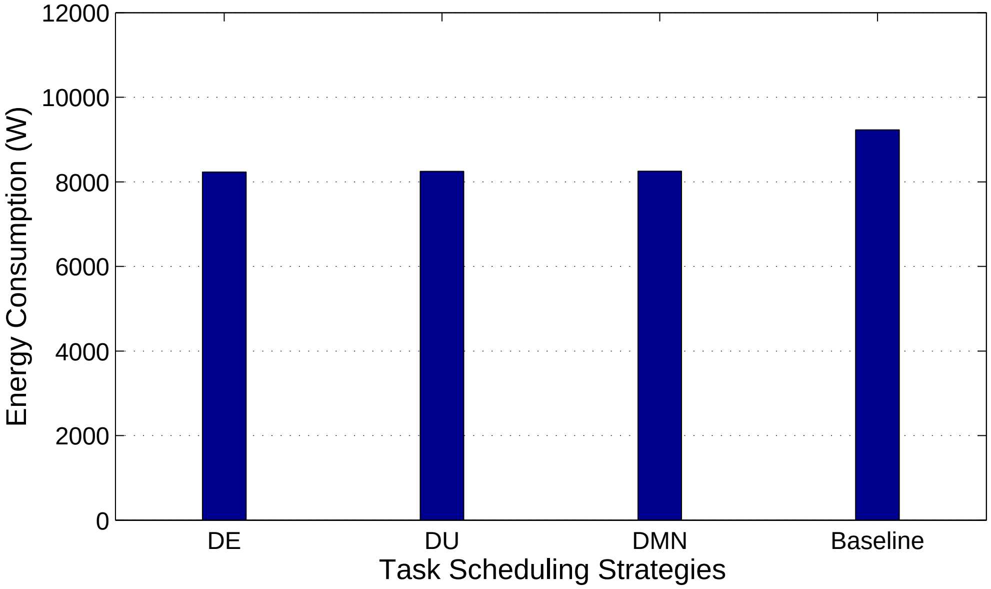


Figure 5.5: Energy consumption of data nodes under I/O-intensive workloads.

5.3 Summary

Energy-aware task scheduling policies were proposed to redistribute workloads in order to minimize the energy consumption of computing infrastructures. We incorporated our thermal models into a thermal-aware management system that distributes tasks to ensure data nodes thermal and energy friendly. This system is integrated into a task scheduler that dispatches and redistributes tasks in a way that all the data nodes' outlet temperatures are below a given threshold. Three strategies were considered in the process of determining which candidate data node should be selected. Through experiments of dispatching CPU-intensive and I/O-intensive workloads, we made a conclusion that evenly distributing the workload across active data nodes is more energy-efficient than the other two strategies.

Chapter 6

Thermal-aware Data Placement

Our evidence shows that disks have non-negligible thermal impacts on the temperature of data nodes (see Chapter 3 and Chapter 4). In this chapter, we demonstrate that data placement strategies can significantly affect thermal performance of data nodes. Firstly, we study the thermal impacts of data placement strategies in a homogeneous environment in Section 6.1. Then, in Section 6.2, we consider the thermal impacts of data placement in a hybrid data storage system. Finally, Section 6.3 concludes this chapter.

6.1 Homogeneous Disk Arrays

After developing a thermal model for a single disk, we are in position to investigate thermal behaviors of multiple disks. Nowadays, a single data node used to have multiple disks. For instance, a single Teradata equipment is able to support more 100 disks. To study how multiple disks deployed in a single data node would affect each other and how they would affect the outlet temperature of a data node, we conduct two groups of experiments. The number of disks in these two groups of experiments are set to two and three, respectively. In this study, we use the internal disk sensors to monitor the disk temperatures because the temperature sensors are not able to applied to the disks in a disk array.

6.1.1 The Two-Disk Case

In the first group of experiments, two disks are configured in the data node. In this data placement study, we use the same testbed described in Chapter 3. It is noteworthy that both disks are placed inside the node's chassis rather than an external disk array. These two disks are of the same type. Compared with disk 2, disk 1 is kept closer to the fan. The initial disk temperature of disk 1 is 36 ℃ , and the initial disk temperature of disk 2 is 38 ℃ . Two I/O-intensive tasks driven Postmark are running on the two disks. We leverage Postmark to create 100 files, the size of which ranges anywhere between 1 to 100 MBytes. Each of the two tasks issues a total of 2,000 transactions.

Table 6.1 The Two-Disk Scenarios

	Disk 1	**Disk 2**
Scenario 1	Task 1	Task 2
Scenario 2	Task 1 & 2	
Scenario 3		Task 1 & 2

We set up three scenarios summarized in Table 6.1. In scenario 1, the two tasks are keeping both disks busy. In scenarios 2 and 3, the two tasks are accessing on one disk while keeping another disk idle.

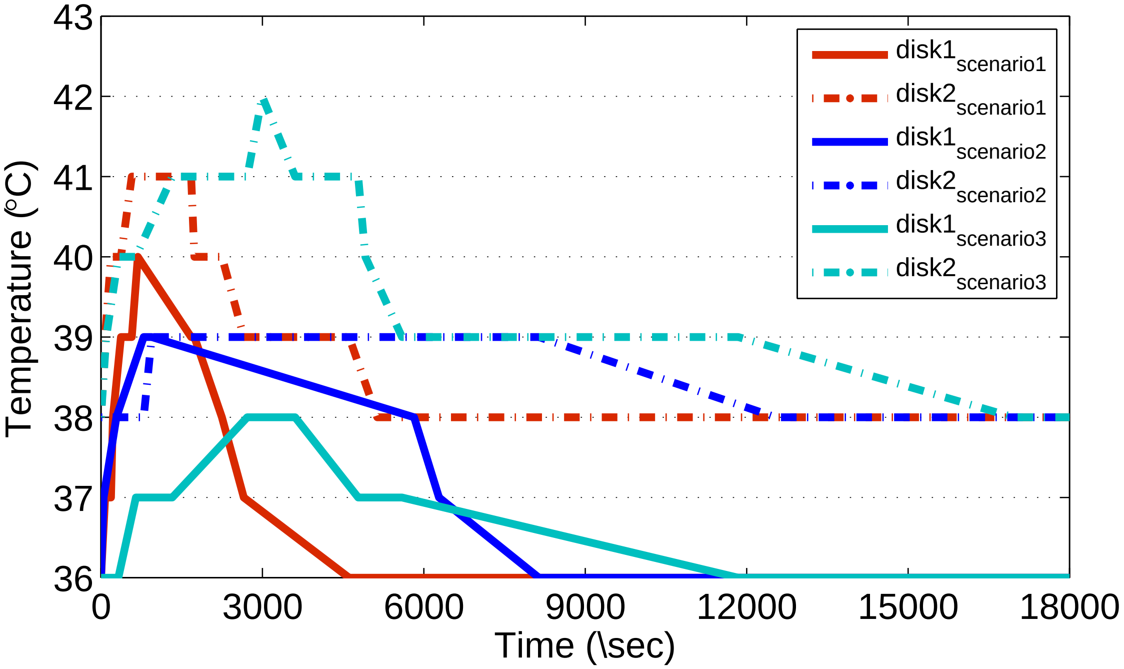


Figure 6.1: Thermal impacts of data placement in the two-disk case.

Fig. 6.1 shows the disk temperatures in the three tested scenarios. In scenario 1, the temperature of disk 1 increases by 4 ℃ , and disk 2 increases by 3 ℃. In scenario 2, after running for a few minutes, the temperature of disk 1 increases by 3 ℃ , and the temperature of disk 2 increases by 1 ℃. In scenario 3, the temperature of disk 2 increases by 4 ℃ , and the temperature of disk 1 increases by 2 ℃ as well.

Table 6.2 Peak Average Disk Temperatures and Total Task/Application Execution Times

Scenarios	**Peak Average Temperature (℃)**	**Execution Time(s)**	
		Task	**Application**
Scenario 1	40.5	4,136	2,250
Scenario 2	39.0	10,632	5,323
Scenario 3	40.0	7,948	3,981

Table 6.2 compares the execution times and peak average temperatures of the two disks tested in the three scenarios. Task execution time is the sum of the two tasks' execution times; application execution time is the maximum execution time of the two tasks involved in the application. We observe that scenario 3 results in the shortest accumulative active disk time (i.e., 3,981 seconds) compared with scenario 1 (i.e., 4,136 seconds) and scenario 2 (i.e., 5,323 seconds), concluding that disks tested in scenario 3 may consume the least energy. Evenly distributing requests issued by the application to the two disks (see scenario 1) produces a high average disk temperature. However, scenario 1 exhibits smaller application execution time than those of scenarios 2 and 3. More interestingly, issuing requests to disk 1 that is closer to the fan in the chassis (see scenario 2) gives rise to the lowest average disk temperature. This result reveals that scenario 2 is more thermal friendly than the other two scenarios.

6.1.2 The Three-Disk Case

We deploy three disks inside a disk-array chassis connecting to the HP server. The testbed is shown in Table 6.3. The disk-array chassis has a fan to cool down disks. We use postmark to initially create 100 files, the size of which ranges from 1 to 100 MBytes. Three postmark tasks issue 1,000 requests to the disks. Ten scenarios (see Table 6.4) are investigated in this group of experiments. In the first scenario, the three tasks are accessing the three disks. In the next three scenarios, the three tasks are sharing a single disk. And for the other scenarios, different task assignments are examined.

Table 6.3 Testbed Configuration for Three-Disk Case

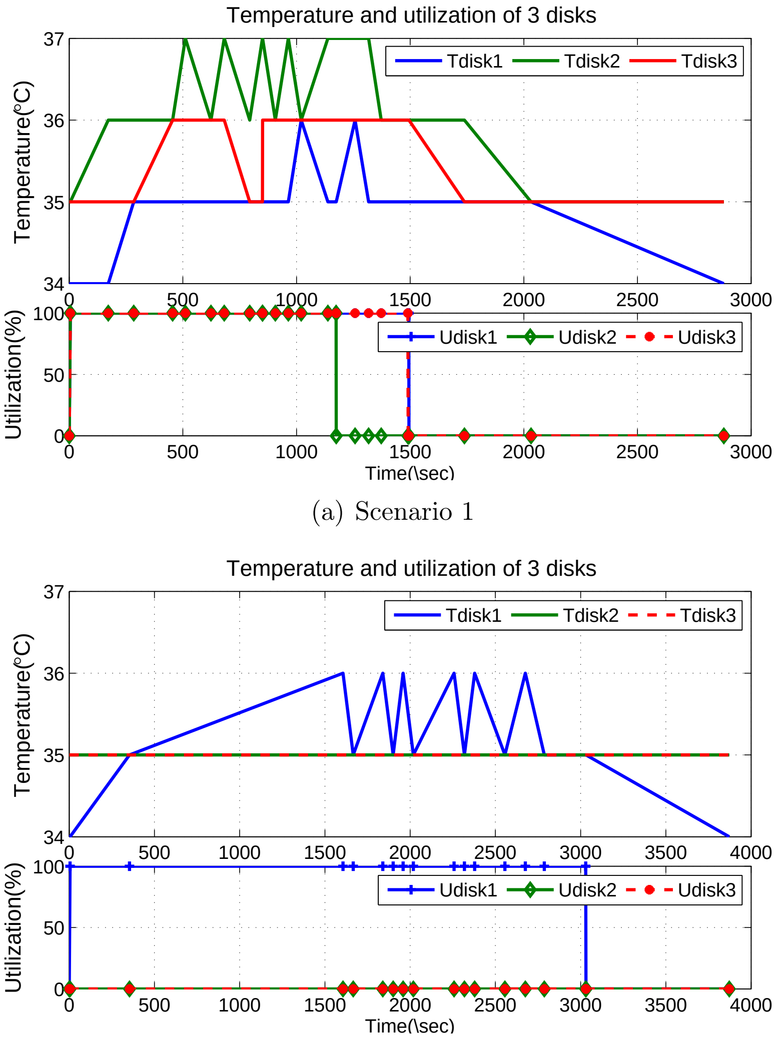
Hardware	**Software**
4 × Intel(R) Xeon 2.4 GHz CPU X3430	Ubuntu 10.04
1 × 2.0 GBytes of RAM	Linux kernel 2.6.32
3 × WD 160 GBytes Sata disk	lm-sensors [3]
(WD1600AAJS-75M0A0 [7])	hddtemp [1]

Table 6.4 The Three-Disk Scenarios

	Disk 1	**Disk 2**	**Disk 3**
Scenario 1	Task 1	Task 2	Task 3
Scenario 2	Task 1 & 2 & 3		
Scenario 3		Task 1 & 2 & 3	
Scenario 4			Task 1 & 2 & 3
Scenario 5	Task 1 & 2	Task 3	
Scenario 6	Task 1 & 2		Task 3
Scenario 7	Task 1	Task 2 & 3	
Scenario 8	Task 1		Task 2 & 3
Scenario 9		Task 1 & 2	Task 3
Scenario 10		Task 3	Task 1 & 2

Fig. 6.2 and fig. 6.3 plots the disk utilization and temperature of the first four scenarios examined in the three-disk case. Fig. 6.4 and fig. 6.5 show other scenarios of task assignment. The peak average temperatures of three disks, the task/application executing times and the estimated cooling cost of each scenario are summarized in Table 6.5, where task execution time is the sum of the three tasks' execution times; application execution time is the maximum execution time of the three tasks within the application.

We observe that evenly distributing tasks to the disks (i.e., scenario 1) leads to higher temperatures on average than forcing all the tasks to share a single disk, however, it takes 1,500 seconds (the shortest time) to complete all the I/O requests. Fig. 6.2(a) shows that the temperatures of disk 1 and 2 increase by 2 ℃; the temperature of disk 3 increases by



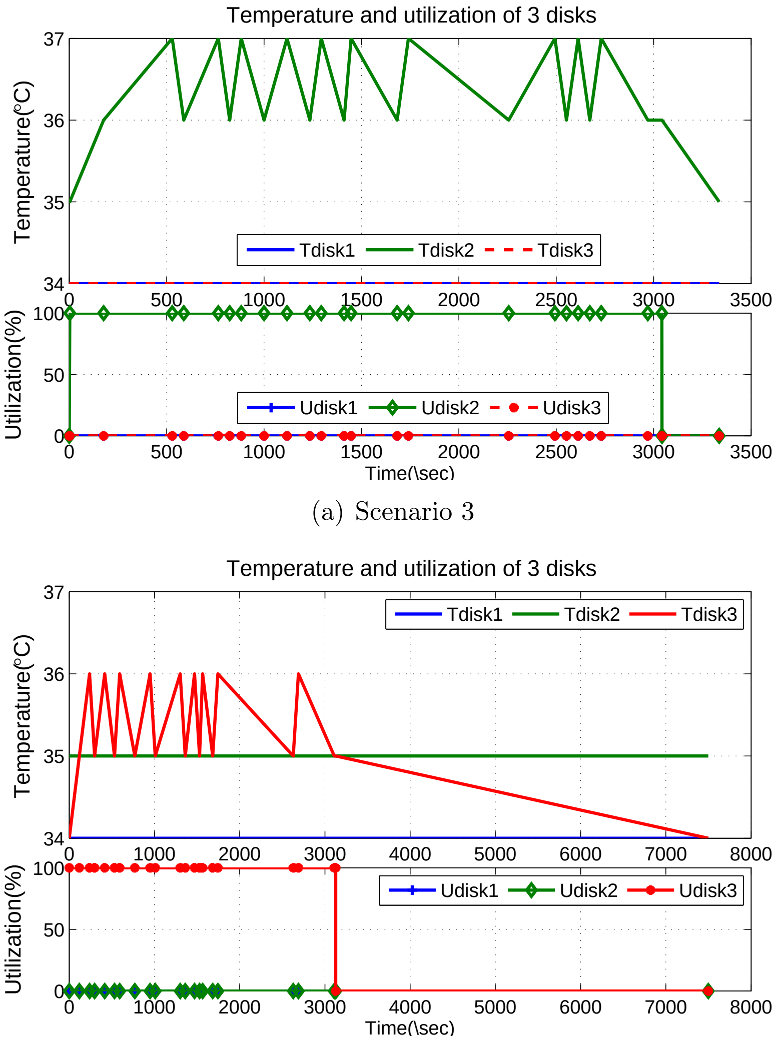
(a) Scenario 1

(b) Scenario 2

Figure 6.2: Thermal impacts of data placement in the three-disk case.

1 ℃. When the three tasks are sharing one disk, the disk temperature increases by 2 ℃ , whereas temperatures of the other two disks remain unchanged. We conclude that sharing a disk among multiple tasks can maintain low disk temperatures at the cost of increased I/O processing time (e.g., from 1,500 to 3,000 seconds).

In both scenarios 5 and 6, two tasks are issuing I/O requests to disk 1 and the third task is sending I/O requests to another disk. The task execution times in these two scenarios are 2,616 and 4,271 seconds, respectively. The long execution time of scenario 6 keeps the three disks in a higher temperature than the initial state. Fig. 6.4(a) shows that the temperature of



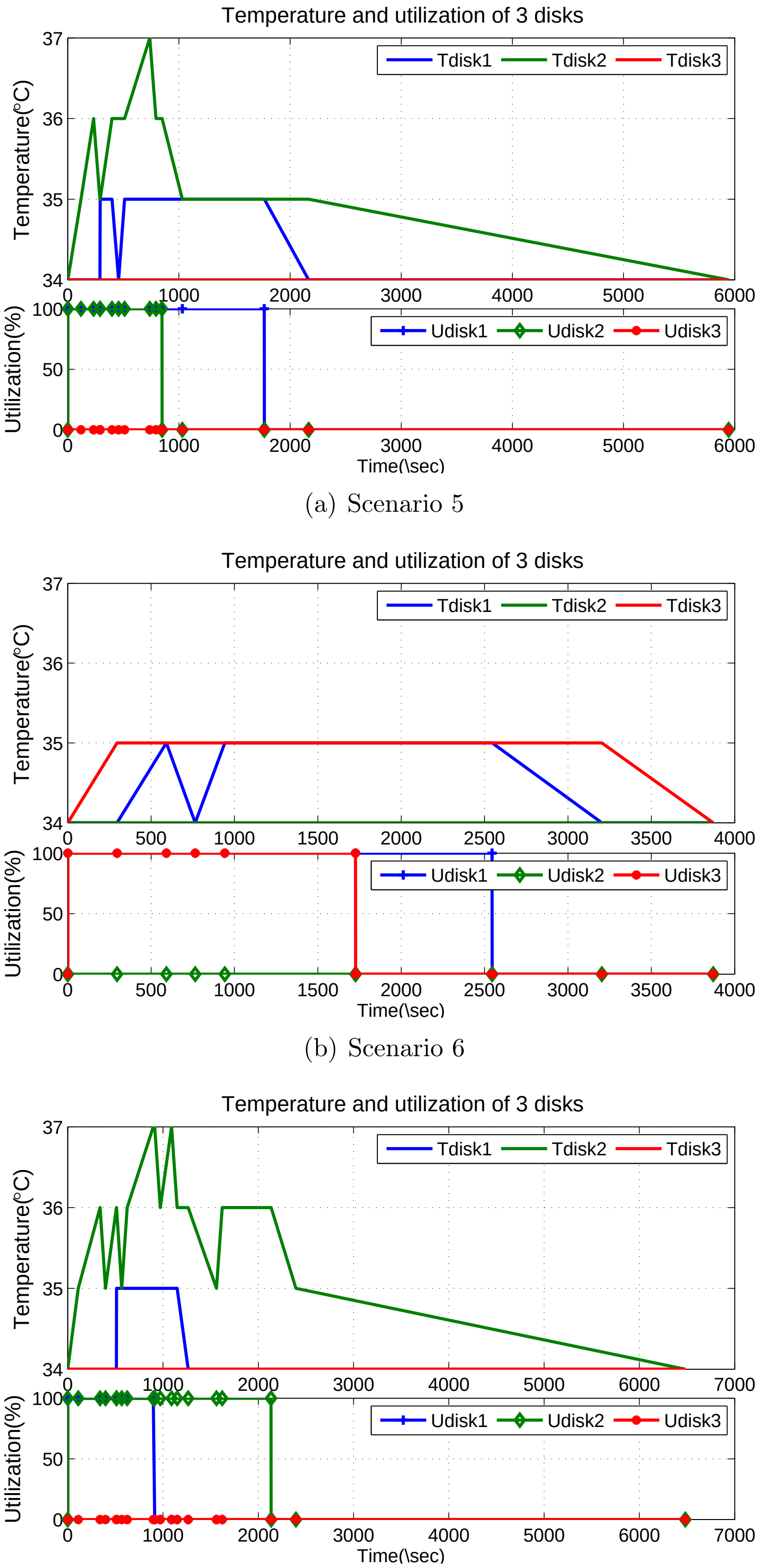
(a) Scenario 3

(b) Scenario 4

Figure 6.3: Thermal impacts of data placement in the three-disk case(2).

disk 1 increases by 3℃, and the temperature of disk 2 increases by 1℃. Fig. 6.4(b) indicates that the temperatures of disks 1 and 3 both increase by only 1℃.

In scenarios 7 and 9, two tasks are assigned to disk 2 and the third one is allocated to the third disk. The execution times of these three tasks are very close. Figs. 6.4(c) and 6.5(b) show that the temperature of disk 2 increases by 3℃. The temperature of disk 1 in scenario 7 rises by only 1℃; however, the temperature of disk 3 in scenario 9 goes up by 2℃. The disks lead to higher energy consumption in scenario 7 than in scenario 9.

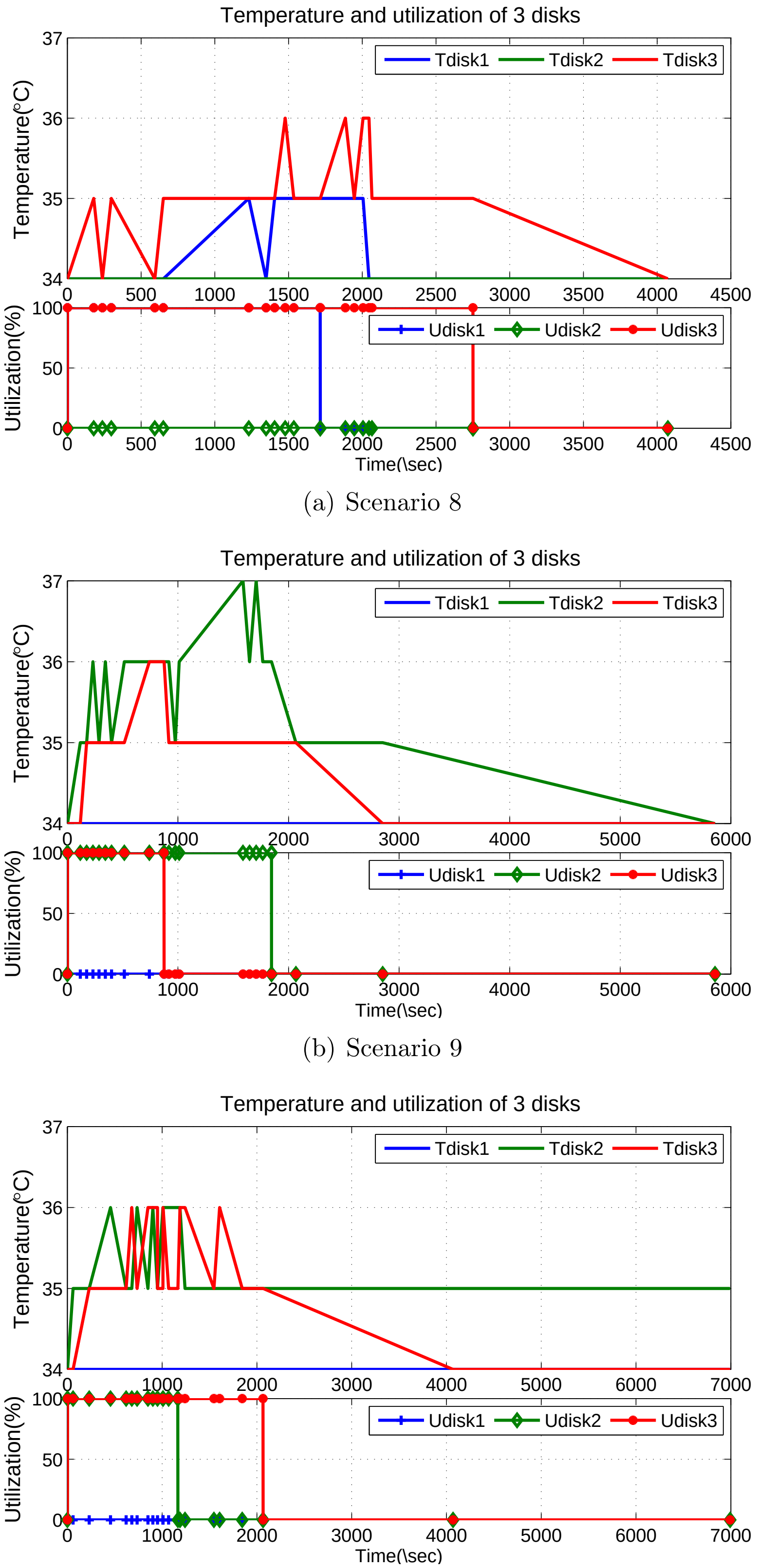


(a) Scenario 5

(b) Scenario 6

(c) Scenario 7

Figure 6.4: Thermal impacts of data placement in the three-disk case(3).



(a) Scenario 8

(b) Scenario 9

(c) Scenario 10

Figure 6.5: Thermal impacts of data placement in the three-disk case(4).

Table 6.5 Peak Average Disk Temperatures, Execution Times and Estimated Cooling Costs.

Scenarios	**Peak Average Temp(°C)**	**Execution Time(s)**		**Cooling Cost(J)**
		Task	**Application**	
Scenario 1	36.35	4144	1500	23,655
Scenario 2	35.33	3010	3010	48,527
Scenario 3	35.00	3024	3024	48,671
Scenario 4	35.00	3126	3126	50,065
Scenario 5	35.34	2616	1768	28,469
Scenario 6	34.67	4271	2551	41,169
Scenario 7	35.34	3032	2134	34,340
Scenario 8	35.00	4466	2751	44,370
Scenario 9	35.33	2717	1846	29,723
Scenario 10	35.35	3227	2063	33,244

When it comes to scenarios 8 and 10, disk 3 handles requests from two tasks, and another disk deals with the requests from the third task. The task execution time of the scenario 8 is much longer than that of scenario 10. Let us consider the first 4,000 seconds during the testing process. Figs. 6.5(a) and 6.5(c) illustrate that the average temperature of the three disks in scenario 10 is higher than that in scenario 8. These results confirm that assigning tasks to a disk sitting in the middle can give rise to high disk temperatures and low energy efficiency.

From Table 6.5, we observe that the cooling cost of scenario 1 is the least and cooling cost of scenario 4 is the most. From the above experiments, we conclude that though evenly distribute tasks have the highest peak average temperature because a load balancing strategy which makes disks stay in high temperatures for less time offers better overall performance, and it is more energy-efficient.

6.1.3 Data Placement Strategy

The previous subsection shows evidence that outlet temperatures affected by disks vary greatly among the tested cases. In the three-disk case, we chose to evaluate ten scenarios out of many other possibilities. For example, one possible scenario might be that the workload is composed of tasks that are of different disk utilizations or of different execution times. And to provide large storage capacity, one may increase the number of disks in each data node. Manually measuring all possible scenarios is a time-consuming and impractical process. A promising solution is to use real measurements collected in simple disk configurations, and to model the thermal characteristics of other complicated scenarios.

Our results suggest that disk temperatures significantly affect the outlet temperatures of a node. Disk temperatures in turn depends on data placement and I/O activities. These observations motivate us to study thermal-aware data placement strategies, which aim to migrate data among disks in order to minimize the cooling costs.

Let us consider a storage cluster containing a large number of data nodes. Encouraged by our experimental results presented in the previous sections, we propose a thermal-aware data placement strategy that is composed of two stages:

- Initial stage: place data sets in data nodes in a way that all the nodes have very similar outlet temperatures.

- Redistribution stage: migrate data according to temperature distribution measured by sensors and predicted by our models.

In the initial stage, a large amount of data must be written into data nodes of a storage cluster. A straightforward strategy is to evenly distribute data across all the data nodes in the system. Data nodes of a storage cluster can be configured in two ways. The first strategy is designed for storage clusters where nodes have the same number of disks deployed. In this strategy, more amounts of data is placed on disks whose temperature in the idle state is higher than other disks. The second strategy is tailored for heterogeneous storage clusters where nodes have different number of disks. In this case, data nodes equipped with more disks should handle a less amount of data in order to reduce heat stress.

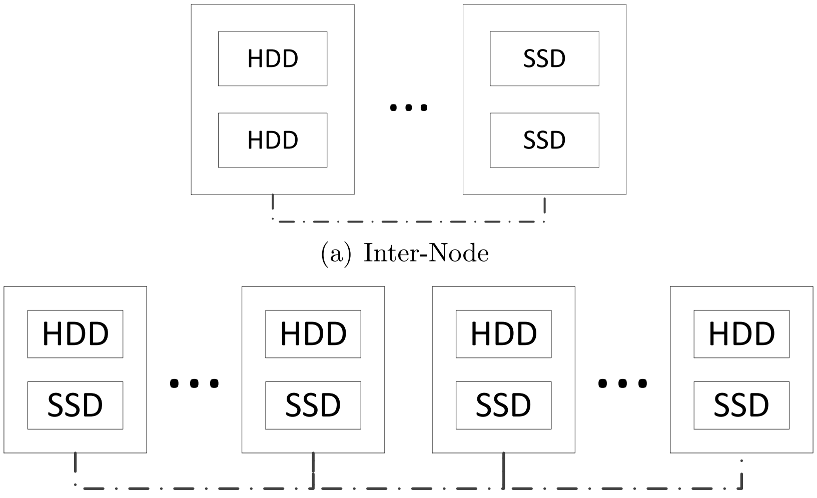
After the initial stage of a storage cluster, the data access patterns are likely to change dynamically. For example, some data sets are accessed more frequently than the other data. The storage cluster tends to exhibit unbalanced outlet temperatures of the data nodes. To balance thermal stresses, the data placement mechanism migrates hot data sets from nodes with high outlet temperatures to those with low outlet temperatures. The data redistribution process is triggered by a threshold of outlet temperatures. For instance, when the maximum outlet temperature is 25% higher than the average temperature of all the nodes, the data redistribution process begins. To maintain high I/O performance, our mechanism delays the redistribution process until the nodes involved in the migration procedure have very large I/O load.

6.2 Hybrid Storage Clusters

After studying the thermal impacts of data placement strategies on homogeneous storage systems, we are in position to investigate thermal behaviors of hybrid disks in the context of cluster storage systems, each of which is comprised of a number of storage nodes. Thanks to good I/O performance offered by SSDs, future cluster storage systems are likely to be powered by a large number of hybrid disks containing both HDDs and SSDs. In this section, we pay attention to the thermal behaviors of two types of hybrid storage clusters. We show that data placement is an efficient approach to minimize negative thermal impacts of a hybrid storage cluster for high-performance clusters.

6.2.1 System Configuration of Hybrid Storage

In this part of study, we build two types of hybrid cluster storage systems, namely, inter-node and intra-node hybrid cluster storage systems (see Fig. 6.6). In an *inter-node hybrid cluster storage system*, there are two types of storage nodes – SSD-enabled nodes and HDD-enabled nodes. All disks in an SSD-enabled node are solid state disks, whereas all disks in an HDD-enabled node are hard drives. In an *intra-node hybrid cluster storage* *system*, each node contains both solid state disks and hard drives. Intra-node hybrid cluster storage systems are homogeneous systems in the sense that all the nodes share an identical configuration. In contrast, inter-node hybrid systems are heterogeneous systems because some nodes are equipped with SSDs while others are comprised of HDDs.



(a) Inter-Node

(b) Intra-Node

Figure 6.6: Two types of hybrid cluster storage systems.

6.2.2 Case Studies

We investigate HDD-first and SSD-first data placement strategies, in which data would be distributed to either HDDs or SSDs. By using the HDD-first strategy, one of the HDDs will be randomly selected if both HDDs and SSDs are available; while the SSD-first strategy will choose SSDs at first. In our evaluation, the inter-node hybrid storage cluster is comprised of 128 SSD-enabled nodes and 128 HDD-enabled nodes. The intra-node hybrid storage cluster has 256 nodes. We make use of Postmark to resemble 128 I/O-intensive tasks, in each of which 1,000 files are created and 5,000 I/O requests are issued. We set the outlet temperatures of nodes to 40 ℃.

Inter-Node Hybrid Storage Cluster

In an inter-node hybrid storage cluster (see Fig. 6.6(a)), the I/O tasks will be evenly issued to the HDD-enabled nodes by the HDD-first strategy. In this case, the requests can be completed within 88 minutes based on our preliminary experiments. According to the HDD temperature model, the working HDD temperature increases to 28.40 ℃ . The temperature of another HDD in the node remains 27.50 ℃ . The temperatures of both SSDs residing in SSD-enabled nodes remain unchanged (i.e., 25.75 ℃). We define the average value of two disk temperatures as the disk temperature of a storage node. The discrepancy between inlet and outlet temperatures of HDD-enabled nodes is $T_{diff}(27.95) = 2.10$; the discrepancy between inlet and outlet temperatures of SSD-enabled nodes is $T_{diff}(25.75) = 1.43$. Therefore, if the inlet temperatures of HDD-enabled and SSD-enabled nodes are 37.90 ℃ and 38.57 ℃ respectively, we could get the same outlet temperature of 40 ℃ . Since our preliminary experiments show that there is about 8 ℃ difference between the inlet temperature and the air-conditioner supply temperature, the air-conditioner supply temperatures should be set to 29.9 ℃ for HDD-enabled nodes and 30.57 ℃ for SSD-enabled nodes in order to gain the same outlet temperature of 40 ℃ .

The power consumptions of a HDD-enabled and a SSD-enabled node are 66.25 W and 48.9 W in idle state. The COP model (see Fig. 3.6 in Section 3.7) indicates that the COP values of HDD-enabled and SSD-enabled nodes are 6.56 and 6.84. Let's consider the power consumption of this inter-node cluster. The mechanical power consumptions are 353,760 J for a HDD-enabled node and 258,129 J for an SSD-enabled node. Using the COP values, we estimate that the cooling costs with respect to HDD-enabled and SSD-enabled nodes are 53,917 J and 37,362 J. Therefore, the total energy consumption incurred by the inter-node hybrid storage cluster and its cooling system is 90,064,864 J.

By using the SSD-first strategy, the I/O requests will be evenly handled by SSD-enabled nodes. In this case, the requests can be finished within 62 minutes based on preliminary results. The temperature of the active SSD is 28.75 ℃ , whereas the other SSD and HDDs remain at 25.75 ℃ and 27.50 ℃ . At HDD-enabled nodes, the difference between inlet and outlet temperatures is 1.96 ℃ ; such temperature difference at SSD-enabled nodes is 1.88 ℃ . Thus, The inlet temperatures of HDD-enabled and SSD-enabled nodes are nearly 38.04 ℃ and 38.12 ℃ . And the supply temperatures are 30.04 ℃ for HDD-enabled nodes and 30.12 ℃ for SSD-enabled nodes. Using the same method, we could calculate the total power consumption of this case is 63,139,305 J. The SDD-first strategy could save 42.64% power consumption than the HDD-first strategy in the Inter-node Hybrid Storage Cluster.

Intra-Node Hybrid Storage Cluster

In an intra-node hybrid storage cluster, the I/O requests will be processed by HDDs in 128 nodes under the HDD-first strategy. The other 128 nodes will remain idle. If the SSD-first strategy is applied, the only difference from the HDD-first case is that the I/O requests will be executed on SSDs rather than HDDs.

Due to the space limitation, we do not present the intermediate results that can be calculated in a similar way. The total energy consumption is 90,022,885 J under the HDD-first strategy, and 63,137,638 J under the SSD-first strategy. The SSD-first strategy reduces the energy consumption by 42.58%.

We observe that the total energy consumption of the HDD-first strategy on an inter-node hybrid cluster is the maximum one, and using the SSD-first strategy on intra-node hybrid cluster results in the minimum total power consumption. In the same hybrid architecture, the SSD-first strategy will save more power than the HDD-first strategy. We conclude that keeping SSD active in the intra-node hybrid storage cluster can achieve the best energy efficiency.

6.3 Summary

In this chapter, we first studied the impact of data placement on the cooling cost and thermal performance of storage system, and proposed a thermal-aware energy-efficient data placement strategy. Then, we built two types of hybrid storage clusters, namely, inter-node and intra-node hybrid storage clusters. Compared with the HDD-first strategy, SSD-first strategy is an efficient approach to minimize negative thermal impacts of hybrid storage clusters for cluster computing.

Chapter 7

Predictive Thermal-aware Energy-efficient Data Transmission

Growing data transmission has become a crucial type of workload in data centers. This chapter presents a novel Predictive Thermal-Aware Management System (PTMS) that is able to reduce the energy cost of storage systems by appropriately selecting data transmission methods. We evaluate the energy consumption 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 to 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.

This chapter is organized as follows: Section 7.1 introduces related information about data transmission in data centers; Section 7.2 shows preliminary results of applying three data transmission strategies in transferring two different types of datasets; Section 7.3 is the framework of our predictive thermal-aware management system; Section 7.4 presents the efficiency of our PTMS system in transferring two large datasets; finally, Section 7.5 concludes this chapter.

7.1 Introduction

A data transmission between two data nodes is composed of three phases: pre-transmission phase; transmission phase; and after-transmission phase. In the first phase, data are read from disk to cache on the original data note. Then in the second phase, data are transformed from original data node to destination data node through network. In the last phase, data are written to the destination disk.

Frequent data transmission contributes to a large portion of energy consumption of data centers. Data placement strategies and data reuse methods are proposed to reduce the energy cost of potential data movements. A new trend of decreasing energy consumption of data transmission is to compress the data before transforming it. In a preliminary work, thermal behaviour of data compression has been investigated [60]. Our predictive thermal management system aims at decreasing the thermal impact of these data transmissions and reducing the total energy cost of data nodes in data centers.

We study three transmission strategies in transforming various data resources:

- Direct Transmission
- Archived Transmission
- Compressed Transmission

In Direct Transmission (DT for short), data are transferred over the network directly, without archive or compression. In the transmission phase, the original data is transferred.

In Archived Transmission (AT for short), data are firstly archived as a single file in the pre-transmission phase, and then transferred through network. After the archived data reach the destination data node, the data should be un-archived on destination data node and then be written to disk in the after-transmission phase.

In Compressed Transmission (CT for short), data are firstly compressed into a single file in the pre-transmission phase, then transferred through network, and finally be decompressed and written to disk at the destination data node in the after-transmission phase.

Data compression has been claimed as an efficient solution to save energy consumption in high-end servers and data centers [68]. Compared with the direct data transmission, compressed data transmission leads to smaller volume of data transformed through network.

However, compressed data transmission generates extra workload on CPU which drive the CPU working under a relative high utilization.

In data centers of current business companies (e.g., Google, Amazon, Facebook), there are more download data transmission than upload data transmission. Download process is composed of transferring data from data nodes in data centers to customer clients. So we focus on reducing energy cost of data transmission from data centers perspective.

7.2 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.

The testbed consists of two Linux servers connected through the fast Ethernet. Table 7.1 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.

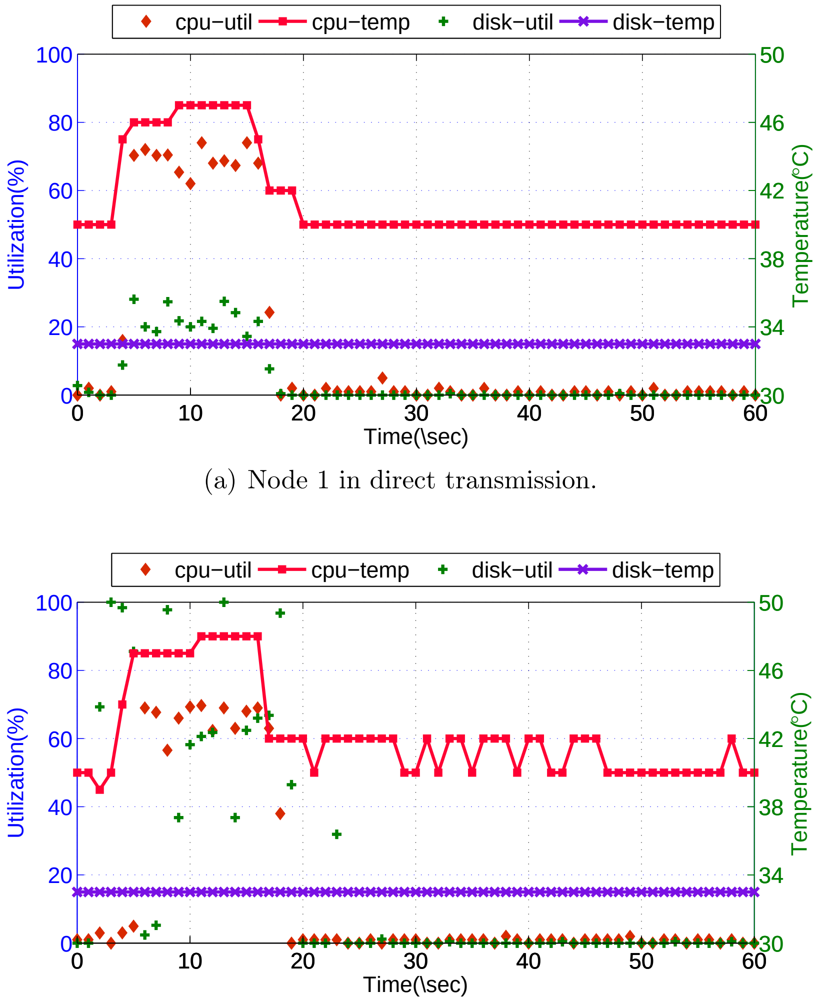
Table 7.1 Testbed Configuration for Data Transmission

	Node 1	**Node 2**
CPU	Intel(R) Celeron(R) 450@2.2GHz	
Network	1 GigaBit Ethernet network card	
Disk	WD-500GB Sata disk([8])	WD-160GB Sata disk([7])
Operating System	Ubuntu 10.04(lucid) Linux kernel 2.6.32-43	Ubuntu 10.04(lucid) Linux kernel 2.6.32-38

We transfer two real-world datasets between the two storage nodes, the results of which are presented as following. Three data-transmission strategies (DT, AT, and CT) are examined in this preliminary experiment.

Transferring A Single Text File

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

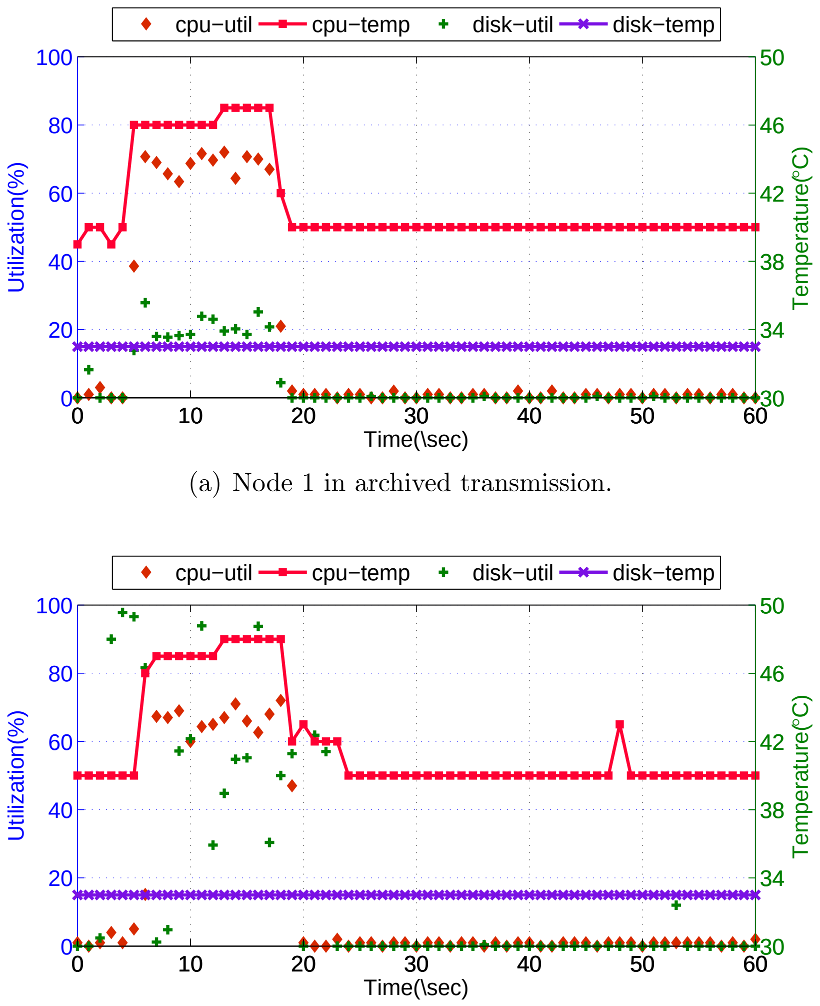


(a) Node 1 in direct transmission.

(b) Node 2 in direct transmission.

Figure 7.1: Performance of transferring 1 text file in direct transmission.

Fig. 7.1, 7.2, and 7.3 display 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) [59]. Staying in the active state for a short period (e.g., less than one minute) has no significant impact on the disk temperature.

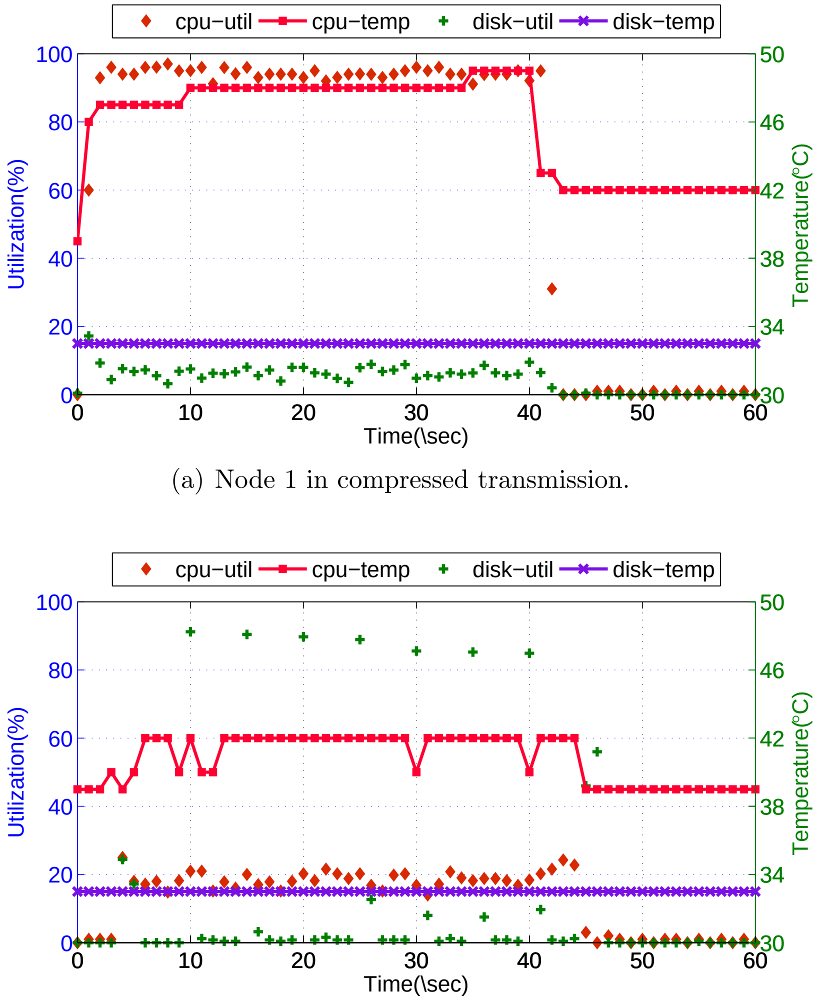


(a) Node 1 in archived transmission.

(b) Node 2 in archived transmission.

Figure 7.2: Performance of transferring 1 text file in archived transmission.

Figs. 7.1(a), 7.2(a), and 7.3(a) show that node 1's CPU utilization and temperature increase 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.



(a) Node 1 in compressed transmission.

(b) Node 2 in compressed transmission.

Figure 7.3: Performance of transferring 1 text file in compressed transmission.

Figs. 7.1(b), 7.2(b), and 7.3(b) 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 7.2 summarizes the execution times and file size, as well as compression ratios. The temperatures and utilizations of CPU and disks are also summarized in Table 7.2. 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 among the three test strategies.

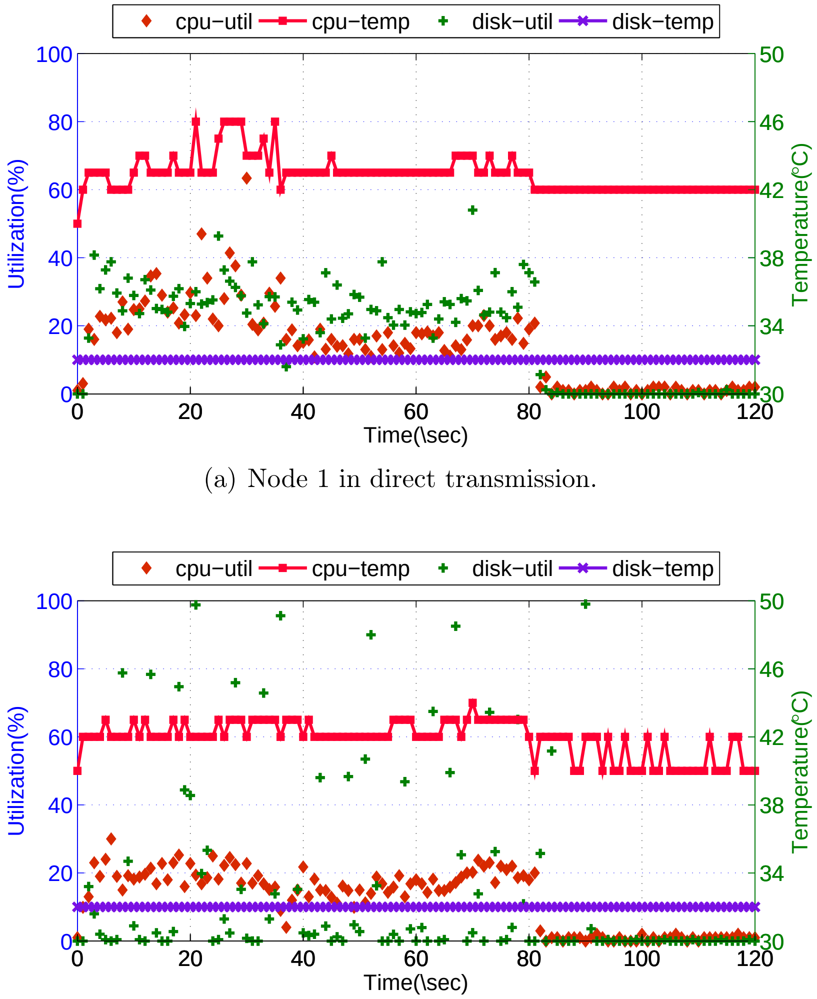
Table 7.2 Summary of Single Text File Transmission

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}**(℃)**	47	48	47	48	49	43
MAX T_{Disk}**(℃)**	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	

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.

Transferring Source Code Files

We evaluate a second case where Linux source code files are transferred between two storage nodes. Fig. 7.4, 7.5, and 7.6 reveal temperatures and utilizations of CPUs and disks where the three data transfer strategies are adopted.

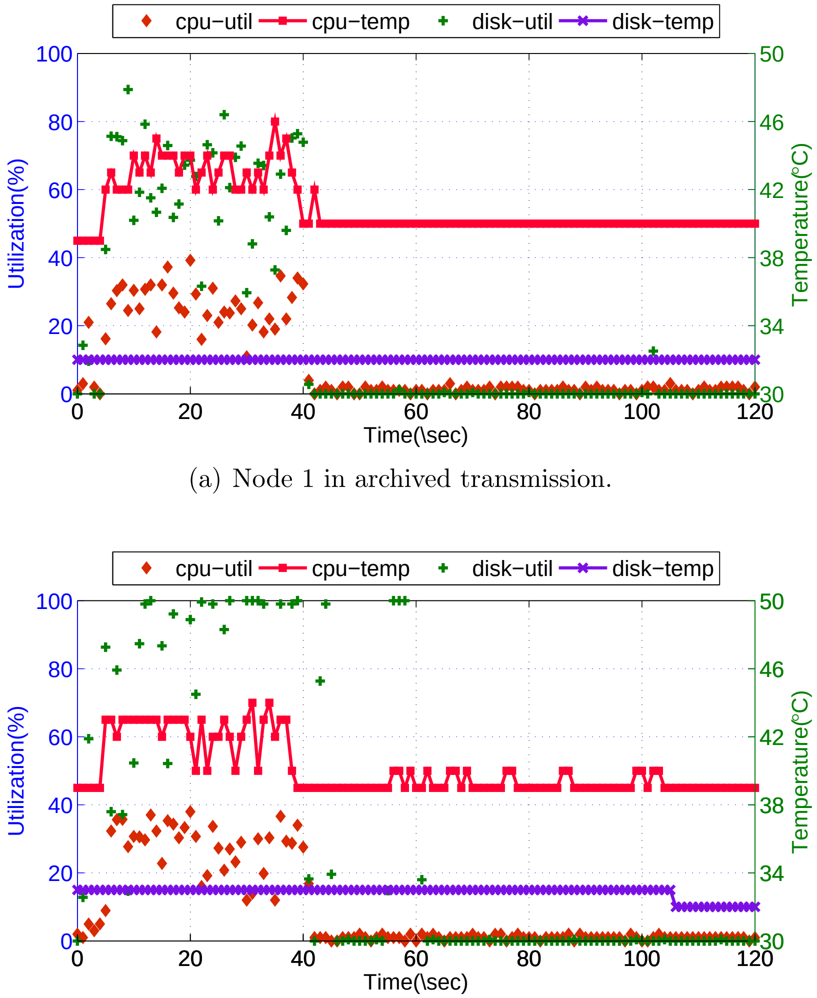


(a) Node 1 in direct transmission.

(b) Node 2 in direct transmission.

Figure 7.4: Performance of transferring Linux kernel files in direct transmission.

For transferring the Linux kernel files with direct transmission (DT), as shown in Fig. 7.4, the time to finish the data transmission is a little longer than 80 seconds. Both data nodes have relatively low CPU utilization for the entire transmission procedure, with CPU utilization of data node 1 is between 10% to 40% and of node 2 is between 20% to 30%. Besides that, on data node 1, the CPU utilization in the first 40 seconds is about 10% higher than the following 40 seconds. And for data node 2, we could observe the same trend for CPU utilization. However, for disk, node 1 maintains a utilization between 20% to 40% and node 2 maintains a widely distributed utilization (from 0% to 100%). The temperature of these two data nodes are also different: the CPU temperature of data node 1 reaches 46℃, while for data node 2, its CPU temperature is heated up to 44℃.

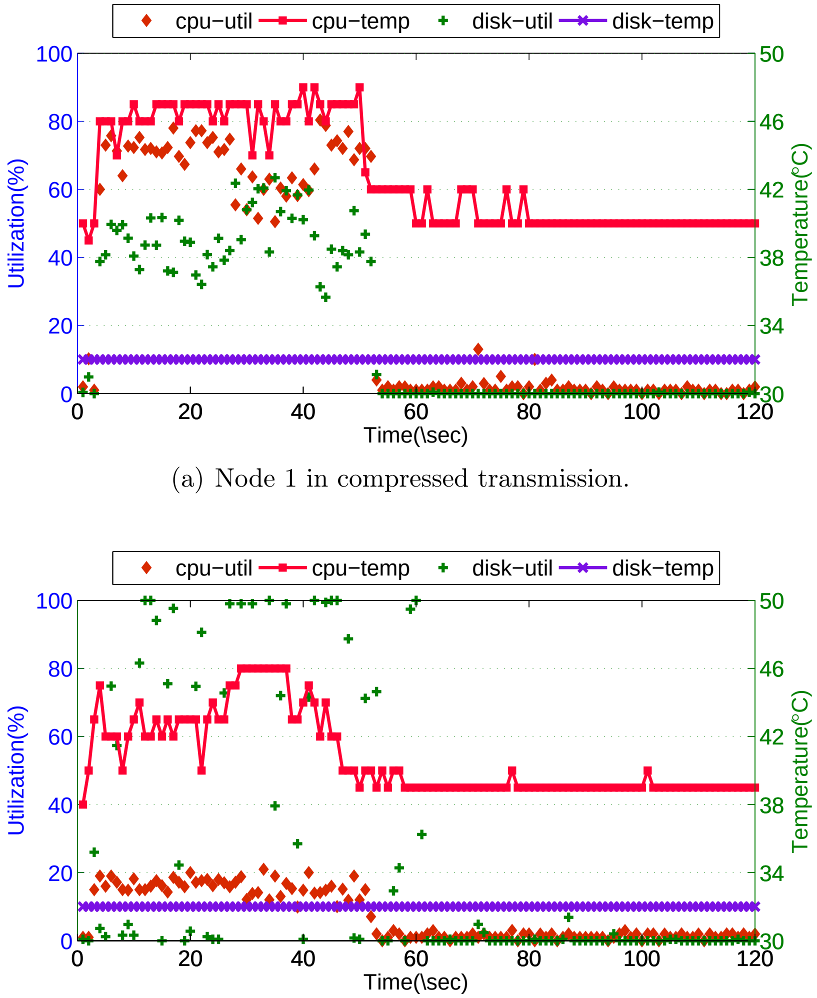


(a) Node 1 in archived transmission.

(b) Node 2 in archived transmission.

Figure 7.5: Performance of transferring Linux kernel files in archived transmission.

Transferring the Linux kernel files with archived transmission (AT), as shown in Fig. 7.5 cost about only 40 seconds, which is half of the time to transferring these files with using direction transmission (DT). This dues to the archival process reduces the data size to be transferred through the network. Different from using DT, CT results in high disk utilization on data node 1 (from 30% to 90%) and higher average disk utilization on data node 2 (from 20% to 100%). The CPU utilizations on both data nodes seem to be almost the same as using DT strategy.



(a) Node 1 in compressed transmission.

(b) Node 2 in compressed transmission.

Figure 7.6: Performance of transferring Linux kernel files in compressed transmission.

As shown in Fig. 7.6, the time cost to transfer the same data files is about 60 seconds with compressed transmission (CT) strategy. The CPU utilizations in this experiment is totally different from the previous two experiments. We observe a very high CPU utilization (from 50% to 80%) on data node 1 during the data transmission, and the CPU temperature is heated up to 48℃. On data node 2, though the CPU utilization remains between 10% and 20%, the CPU temperature is also higher than using the other two transmission strategies. The disk utilization of data node 1 is between 30% and 60%. While for data node 2, disk utilization is from 0% to 100%. Compare with using DT strategy, the disk utilization of data node 2 is more concentrated on higher value.

Table 7.3 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}**(℃)**	46	44	46	44	48	46
MAX T_{Disk}**(℃)**	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	

For better comparison, we summarize the detailed results as shown in Table 7.3. 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 AT is much shorter than that of DT. 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 group of experiments, 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. 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.

Motivation of the Predictive Thermal-aware 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 7.2); whereas AT is the most energy-efficient strategy to transfer a large number of small files (see Section 7.2). 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. Automatically selecting an appropriate method is critical to save energy on data transmissions.

To address this problem, we design a predictive thermal-aware management system or PTMS. There are two phases incorporated in PTMS. 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 PTMS are illustrated in the next section.

7.3 Framework of Predictive Thermal-aware Management System

Fig. 7.7 shows the framework of predictive thermal-aware management system (PTMS for short). It displays a storage system equipped with n data nodes. The PTMS is applied on each node. The monitor module gathers runtime parameters related to data transmissions, file metadata, and storage nodes (e.g., temperatures and utilizations). When a data transmission request is detected, the module forwards the request to the method selector, which chooses a thermal friendly data-transfer strategy for the transmission.

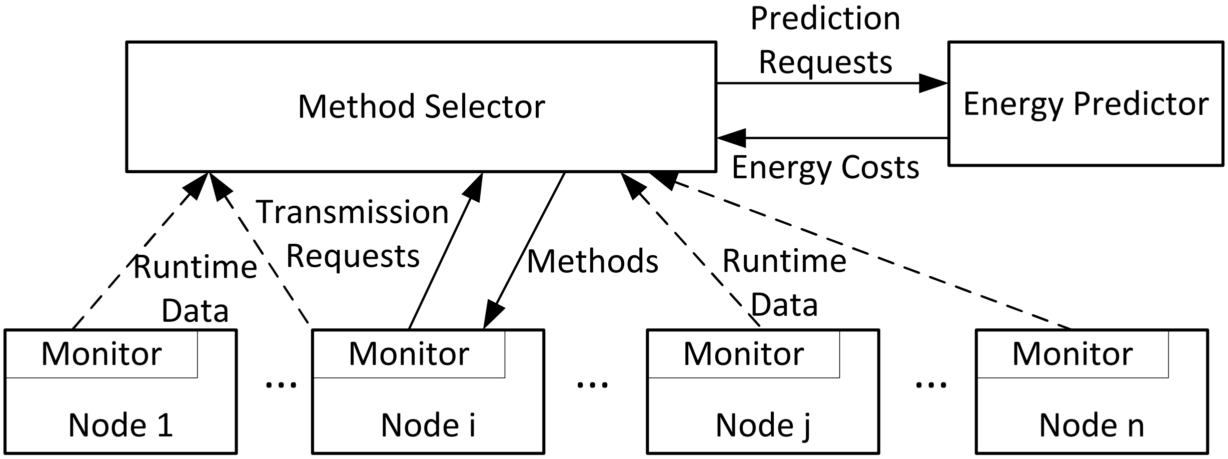


Figure 7.7: The framework of the predictive thermal-aware management system.

The Method Selector not only maintains candidate data-transfer strategies, but it also judiciously chooses the best strategy to reduce thermal impact and the energy cost. Fig. 7.7 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. 7.8, provides the energy estimates of data transmissions handled by a particular strategy. In our predictive thermal management system, the predictor is focused on the energy consumption (including both computing energy cost and cooling cost) of data nodes in the storage system. So, before estimate the energy cost, the data transmission type should be determined. There are mainly three types of data transmission from the perspective of data centers: upload, download and data transmission in data centers. The performance models and energy models proposed in PEAM system are used [60].

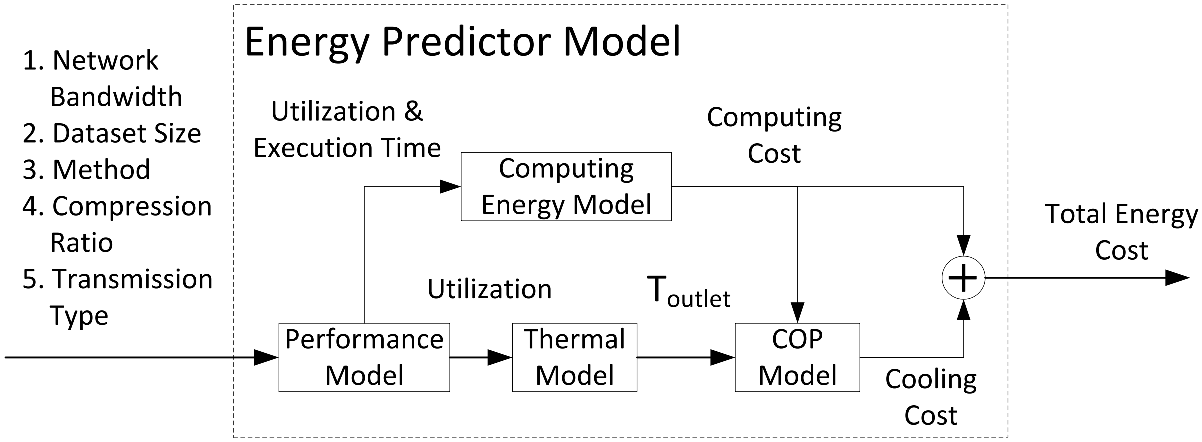


Figure 7.8: Framework of the energy predictor module.

7.3.1 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

$$U_{CPU} = g(Method, R_{compression}), \tag{7.1}$$

$$U_{disk} = h(Method, R_{compression}), \tag{7.2}$$

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

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

$$\begin{aligned} T_{execution} =& k(size, Method, R_{compression}, Bandwidth) \\ =& T_{read} + T^{Method}_{pre-proc} + T_{send} \\ &+ T_{receive} + T^{Method}_{after-proc} + T_{write} \end{aligned} \tag{7.3}$$

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.

7.3.2 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:

$$T_{CPU}(t) = f_{CPU}(T^{CPU}_{i}, T_A, U_{CPU}, t), \tag{7.4}$$

where T_i^{CPU} 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. The disk temperature can be modelled as:

$$T_{disk}(t) = f_{disk}(T_i^{disk}, T_A, U_{disk}, t), \tag{7.5}$$

where T_i^{disk} 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 outlet temperature model proposed in the previous chapter to quantify the thermal impact of CPU and disk activities on outlet temperatures.

$$T_{outlet} = T_{inlet} + a + b * T_{CPU} + c * T_{disk}, \tag{7.6}$$

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

7.3.3 Computing Energy Power Model

We use (7.7) 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. $P_{component}^{max}$ and $P_{component}^{idle}$ are the power when the component works in full capacity and is in the idle state, respectively.

$$P_C = P_i + \Sigma(U_{component} * (P_{component}^{max} - P_{component}^{idle})) \tag{7.7}$$

With the computing cost P_C and cooling power P_{AC}(see Chapter 3) in place, we can express the overall power as:

$$P_{Total} = P_C + P_{AC}, \tag{7.8}$$

7.4 Results

Massive amount of data are uploaded to and downloaded from data centers. For instance, 72 hours of videos are uploaded to Youtube every minute; 350GB data are uploaded to Facebook every minute; 15,000 tracks are downloaded from iTune every minute [25]. Uploading and downloading a large amount of data consume considerable energy and time; even worse, energy cost of data centers is rising dramatically with the increasing amount of data.

To evaluate the energy efficiency of our predictive thermal-aware management system designed for data centers, we conduct two sets of experiments.

In the first group of experiments, a pair of data nodes are transferring a dataset that contains hundreds of ASCII files generated by Postmark. The dataset's size is 1GB; the file size of each is anywhere between 1M to 100M. Among all the transferred files, small files are accessed more frequently than large files. It is important to study the energy consumption caused by transferring small files. For example, a report shows that there are 500 millions of files saved every 48 hours on Dropbox as of May, 2012 [40]. A majority of Dropbox users use their free space to store small files. In most cases, uploaded files to the Dropbox servers are small in size.

We compare the performance of the four data transmission strategies (i.e., DT, AT, CT, and PTMS) transferring the two datasets. Fig. 7.9 shows the energy cost of Node 1 that transfers the first dataset to Node 2. We observe that compared to the other strategies, AT consumes less energy for both nodes 1 and 2 when the ASCII files are transmitted. CT is the least energy-efficient scheme among all the evaluated strategies.

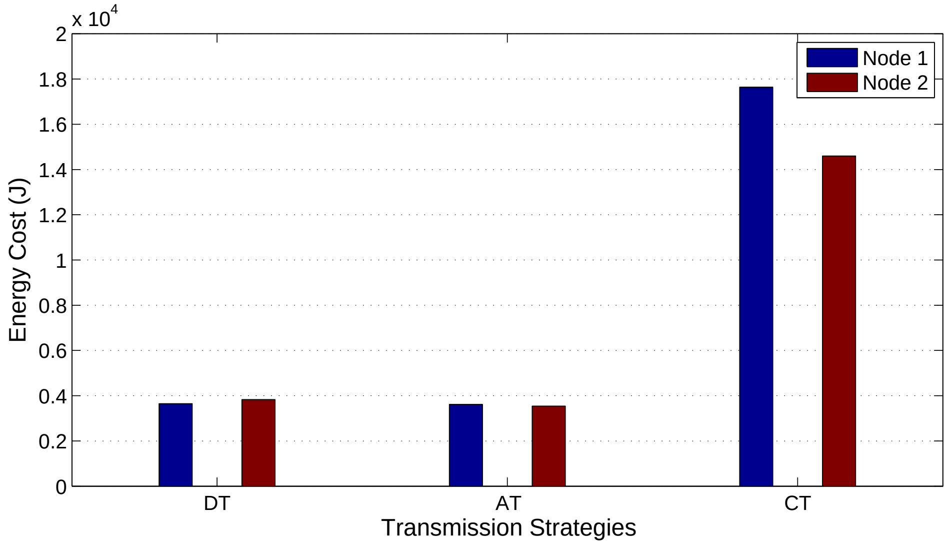


Figure 7.9: Energy cost of data nodes in transferring the ASCII files.

Now we are in a position to evaluate the energy efficiency of our PTMS. Fig. 7.10 shows the energy cost of the four strategies under different transmission types. Not surprisingly, CT consumes more energy transferring this dataset than the other strategies. This is mainly because data compression or/and decompression cost extra CPU time and energy. Regardless of the transmission types, PTMS is the best one among all the tested strategies.

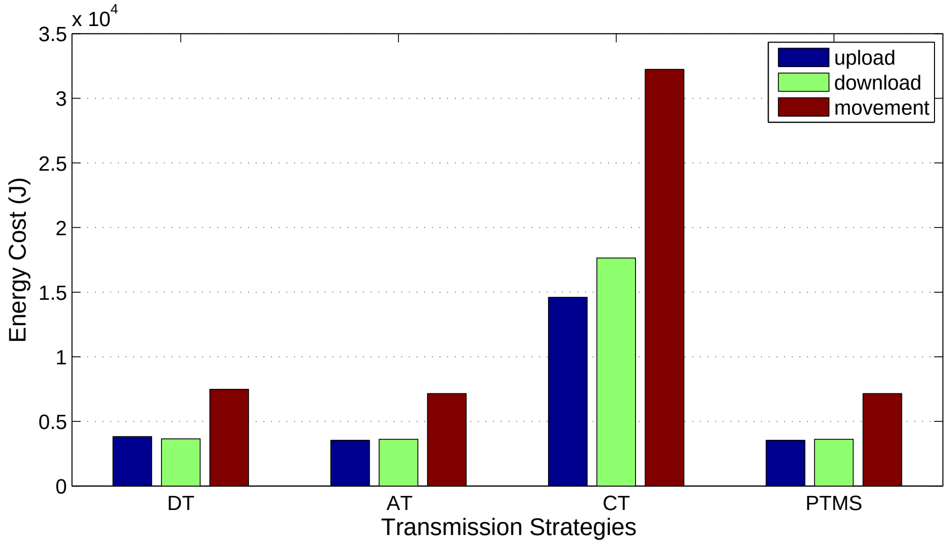


Figure 7.10: Energy cost of transferring the ASCII files under different transmission types.

To resemble real-world cases where large files are transferred, in the second experiment group we choose to use a dataset of 60 GB Human Genome sequences. This dataset is available at NIH's (National Institutes of Health) NCBI website[1]. Each sequence file contains the DNA sequence of an entire chromosome. Most of the files in this dataset are larger than 3GB.

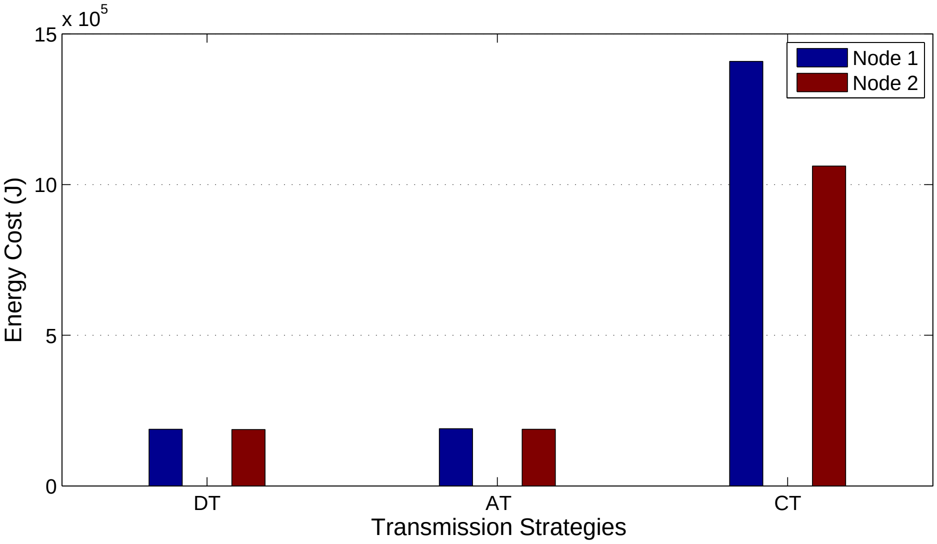


Figure 7.11: Energy cost of data nodes in transferring the Human Genome dataset.

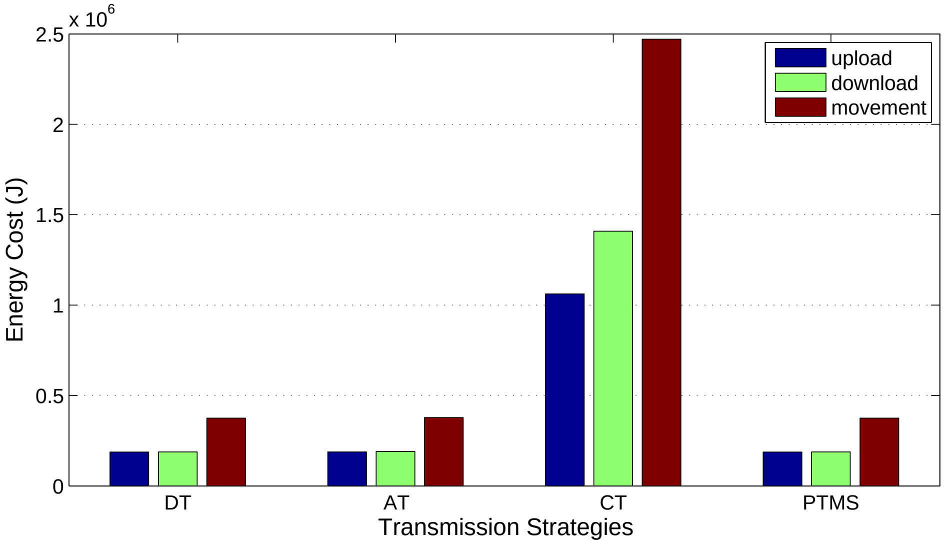


Figure 7.12: Energy cost of transferring the Human Genome dataset under different transmission types.

Fig. 7.11 shows the energy incurred by transferring the Human Genome dataset between nodes 1 and 2. Fig. 7.12 depicts the energy cost of transferring the Human Genome dataset

[1] ftp://ftp.ncbi.nih.gov/genomes/H_sapiens

with the four strategies under different transmission type. We observe that regardless of data nodes, AT and PTMS outperform the other two strategies. The experimental results suggest that PTMS noticeably conserves energy for all the other three data transmission types.

7.5 Summary

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 chapter, we first characterized the thermal and performance behaviours of three data transmission methods. A thermal-aware data transmission strategy is proposed, where data transmission is divided into three camps: uploads, downloads, and migrations within a data center. We implemented the thermal-aware data transmission strategy in a predictive thermal-aware management system or PTMS, which is conducive to estimating data nodes' energy consumption that guides the management of data transmissions. Among all the candidate data transmission policies, PTMS dynamically chooses the most appropriate one that meets the needs of a wide range of data-intensive applications coupled with various data transmission patterns. Our experimental results show that our system performs better than simply selecting any one among the candidate methods for data transmission in terms of energy efficiency.

Chapter 8

Conclusion

In this dissertation, we demonstrated a thermal modeling approach that investigates thermal impacts of both CPUs and disks in data nodes. Then, the model is used to estimate the outlet temperatures of data nodes based on CPU and disk utilization. In addition, we incorporated our thermal models into thermal management strategies, which make data nodes thermal and energy friendly. The first strategy is integrated into a scheduler to dispatch and redistribute I/O tasks in a way to ensure that all the data nodes' outlet temperatures are lower than a predetermined threshold. Following are a two-stage data placement strategy in homogeneous data storage systems and a SSD-first data placement strategy in hybrid storage systems. The last one is a thermal-aware data transmission strategy, where data transfers are divided into three camps: uploads, downloads, and migrations within a data center. We implemented the thermal-aware data transmission strategy in a predictive thermal-aware management system or PTMS, which is conducive to estimating data nodes' energy consumption that guides the management of data transmissions. Among all the candidate data transmission policies, PTMS dynamically chooses the most appropriate one that meets the needs of a wide range of data-intensive applications coupled with various data transmission patterns.

8.1 Main Contributions

Energy consumption of data centers has increased dramatically in recent years. Computing costs of IT facilities and cooling costs of air conditioner systems contribute a large portion of the total energy consumption of data centers. There are urgent needs to build energy-efficient data centers; growing attention has been paid to reducing cooling costs of data centers. The temperatures of data nodes in data centers have been identified as key factors to cooling costs. Thus, modeling the temperatures of data nodes plays an important role in estimating their energy consumption, which could be used to guide the development of energy-efficient workload management.

8.1.1 Thermal Modeling of Disk Temperatures

Thermal behaviors of disks are not fully studied, and disks have not been taken into account as an important contributor to outlet temperatures of data nodes. My preliminary experimental results show that disks make noticeable impacts to outlet temperatures (i.e., deploying an additional disk contributes 0.3 ℃ to the outlet temperature). In addition to traditional hard drives, solid-state disks (SSD) are investigated in my dissertation research. Compared with hard disk drives, solid-state disks have higher read/write performance and lower energy consumption. Solid-state disks are more temperature-sensitive to disk activities than hard drives. We proposed a thermal model that incorporates both hard drives and solid-state disks.

8.1.2 Thermal Modeling of CPU Temperatures

Thermal behavior of CPU has been studied a lot; however, previous research models the CPU temperatures in a fine-grained level. For instance, deep research is conducted to study how micro-architecture would impact the thermal behaviors of processors. In addition, CPU frequency and voltage are also considered as important contributions to CPU temperature. We investigate the thermal characters of processors in a coarse-grained level by considering the utilizations. Relations between CPU utilization, temperature, and energy consumption have been built. We implemented two types of models, namely the polynomial and logarithmic models, to predict CPU temperatures. Experimental results show that the logarithmic models (with the precision error less than 1%) have better performance than the polynomial models.

8.1.3 Thermal Modeling and Energy Consumption of Data Nodes

Modeling the temperature of data nodes is a critical step prior to energy consumption estimation of data nodes, especially the cooling cost for data centers. With the thermal models in place, outlet temperatures of data nodes can be predicted under particular workloads without deploying temperature sensors. Combining these models enables administrators to set up an appropriate supply temperature, which substantially reduces cooling cost of data centers. Cooling cost is derived from computing cost of data nodes and the COP (Coefficient of Performance) model, which is a function of cooling systemsŠ supply temperatures. The total energy cost of a data center is the summation of its computing cost and cooling cost.

8.1.4 Thermal-aware Task Scheduling System

Dispatching tasks plays a significant important role in load balancing and reducing the energy consumption of data storage systems. Conventional task scheduling strategies distribute tasks for decreasing the computing cost of data nodes in storage systems. New trends are brought up by considering the reduction of cooling cost of data nodes. With energy consumption of data nodes estimated by the thermal models, We proposed a task scheduling strategy, which keeps the outlet temperatures of data nodes well balanced. My task scheduling strategy not only selects the best data node that the task should be assigned to in terms of total energy costs of storage systems, but also ensures that the outlet temperatures of data nodes do not exceed a pre-determined threshold, which protects computing resources from working in a high temperature environment.

8.1.5 Data Placement in Homogeneous Disk Arrays

Evidence has shown that disks have non-negligible impacts on data nodes. In modern data centers, a single data node usually supports multiple disks. For instance, a Teradata equipment is able to house more than 100 disks. Data placement can significantly affect the thermal behaviors of data nodes. The thermal impacts and energy consumption of data nodes with various data placement schemes motivated me to build a new data placement strategy. My data placement strategy contains two-stage schemes: in the initial stage, data are distributed evenly inside the data center; and then in the redistribution stage, data are migrated according to outlet temperature distributions.

8.1.6 Data Placement in Hybrid Cluster Storage Systems

Hybrid cluster storage systems could be classified into two categories: inter-node and intra-node hybrid systems. In an intra-node hybrid cluster storage system, each node contains both solid state disks (SSDs) and hard drive disks (HDDs). In an inter-node hybrid system, some nodes are equipped with SSDs while others are comprised of HDDs. The performances and thermal behaviors of hard drive disks and solid state disks are explored, and SSD-first strategy is proposed to minimize the negative thermal impacts of hybrid storage clusters.

8.1.7 Predictive Thermal-aware Management System (PTMS)

By investigating the thermal impacts and energy consumption of applying several potential data transmission strategies, We developed the PTMS system that chooses the most energy-efficient data transmission strategy for data storage systems. PTMS is composed of three components: an energy cost predictor, a method selector, and monitors. The energy cost predictor estimates the energy consumption of data transmission by giving the size of data to be transferred, compression ratio, bandwidth of network, and the like. The method selector chooses the best data transmission method in terms of energy efficiency. The monitor collects run-time information of each data node.

8.2 Future Work

8.2.1 Considering Ambient Temperatures

Ambient temperatures are a major factor affecting disk temperatures. Modeling the impact of ambient temperatures on disk temperatures is still in its infancy. I plan to conduct experiments to study the thermal behavior of disks with different workload conditions under various ambient temperatures. Besides postmark, I will also consider running continuous read and write benchmarks to study disk thermal behaviour.

8.2.2 Data Storage Nodes Equipped with Multiple Disks

My preliminary findings suggest that deploying an additional disk leads to an increment of about 0.3 ℃ of outlet temperatures. Each of my current tested node houses no more than four disks. Real-world data nodes may be equipped with more than 64 disks. I will further investigate the impacts of the large number of disks on outlet temperatures.

8.2.3 Heterogeneous Data Centers

My current research focuses on task scheduling and data placement on homogeneous data centers. With the rapid development of technology, heterogeneous data centers are becoming popular. When a data center is expanded, new instruments are deployed, which makes the data center heterogeneous in nature. I intend to design new scheduling and data placement algorithms tailored for heterogeneous data centers.

8.2.4 Thermal Models for Hadoop Clusters

Hadoop clusters, which support the processing of large data sets in a distributed computing environment, have been widely used in modern data. Hadoop enables the distribution of workload among thousands of data nodes with continuous operation even if some of the data nodes fail. Each data file in the Hadoop system has three replicas. I plan to develop new thermal models to capture thermal behaviors of Hadoop clusters. My new model will incorporate various data placement strategies designed for Hadoop clusters.

8.2.5 Energy-aware Hadoop Distributed File System

I have investigated thermal-aware data transmission inside a data center. In the future, I plan to study the energy-efficient data management in Hadoop clusters. In the Hadoop system, there are usually three replicas for each data block. Thus, when a file is imported into the Hadoop distributed file system or HDFS, three copies of the file are created. I will develop an energy-efficient HDFS, which can manage replicas in a way which reduces energy consumption. In addition, I will develop an energy-efficient data transmission mechanism to efficiently transfer massive amounts of data between clients and HDFS.

8.2.6 Address Big Data Challenges

Big Data is a collection of large and complex data sets that are difficult to be processed by traditional data management tools. My long-term goal is to address big data challenges such as data processing, storage, and transferring. Among a wide variety of big data applications, I will be focusing on genomics and biological research. I plan to start this research by investigating two genomics and bioinformatics applications running on a Hadoop cluster. These applications are drivers for my future parallel computing studies that are focused on data analytics. Data placement of massive amounts of data will be addressed in my future research while these data-intensive applications are being developed.

8.2.7 Security Issue of Data Storage Systems

A traditional method to ensure the security of data is encryption. However, there is a new trend that hackers send continuous requests to data servers to make these servers extra hot until there are down. I plan to conduct research by applying thermal-aware management strategies to distribute the workload and control the responses to user requests to ensure the security of data servers in data storage systems.

Bibliography

[1] hddtemp. `http://manpages.ubuntu.com/manpages/natty/man8/hddtemp.8.html`.

[2] Intel ssd sa2m080g2gc. `http://download.intel.com/newsroom/kits/ssd/pdfs/X25-M_34nm_DataSheet.pdf`.

[3] lm-sensors. `http://www.lm-sensors.org/`.

[4] Minigoose-ii. `http://www.itwatchdogs.com/datasheets/MiniGoose_II_User_Manual_v1_05.pdf`.

[5] stress. `http://www.unixref.com/manPages/stress.html`.

[6] Temperature sensor. `http://www.itwatchdogs.com/datasheets/Temperature%20Sensor%20datasheet%20(v1.06).pdf`.

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

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

[9] Whetstone. `http://www.netlib.org/benchmark/whetstones`.

[10] Global data center energy demand forecasting. Technical report, Datacenter Dynamics, 2011.

[11] What happens on facebook in each day? `http://visual.ly/what-happens-facebook-each-day`, 2012.

[12] Accenture and WSP. Cloud computing and sustainability: The environmental benefits of moving to the cloud. Technical report, Accenture, 2010.

[13] B. Aksanli, J. Venkatesh, L. Zhang, and T. Rosing. Utilizing green energy prediction to schedule mixed batch and service jobs in data centers. *SIGOPS Oper. Syst. Rev.*, 45(3):53–57, Jan. 2012.

[14] M. Al Assaf, X. Jiang, M. Abid, and X. Qin. Eco-storage: A hybrid storage system with energy-efficient informed prefetching. *Journal of Signal Processing Systems*, 72(3):165–180, 2013.

[15] A. R. Alameldeen and D. A. Wood. Adaptive cache compression for high-performance processors. *SIGARCH Comput. Archit. News*, 32(2):212–, Mar. 2004.

[16] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor. Storage modeling for power estimation. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, SYSTOR '09, pages 3:1–3:10, New York, NY, USA, 2009. ACM.

[17] R. Ayoub, K. Indukuri, and T. Rosing. Temperature aware dynamic workload scheduling in multisocket cpu servers. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 30(9):1359–1372, Sept 2011.

[18] R. Ayoub, R. Nath, and T. Rosing. Jetc: Joint energy thermal and cooling management for memory and cpu subsystems in servers. In *High Performance Computer Architecture (HPCA), 2012 IEEE 18th International Symposium on*, pages 1–12, Feb 2012.

[19] R. Ayoub, R. Nath, and T. S. Rosing. Cometc: Coordinated management of energy/thermal/cooling in servers. *ACM Trans. Des. Autom. Electron. Syst.*, 19(1):1:1–1:28, Dec. 2013.

[20] M. Baile. The economics of virtualization: Moving toward an application-based cost mode. Technical report, VMware, November 2009.

[21] M. Balakrishnan, A. Kadav, V. Prabhakaran, and D. Malkhi. Differential raid: Rethinking raid for ssd reliability. *Trans. Storage*, 6(2):4:1–4:22, July 2010.

[22] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta. Cooling-aware and thermal-aware workload placement for green hpc data centers. In *Proceedings of the International Conference on Green Computing*, GREENCOMP '10, pages 245–256, Washington, DC, USA, 2010. IEEE Computer Society.

[23] A. Beloglazov, J. Abawajy, and R. Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 28(5):755 – 768, 2012. <ce:title>Special Section: Energy efficiency in large-scale distributed systems</ce:title>.

[24] A. Beloglazov and R. Buyya. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, CCGRID '10, pages 826–831, Washington, DC, USA, 2010. IEEE Computer Society.

[25] S. Bennett. What happens on line in 60 seconds? `http://www.mediabistro.com/alltwitter/online-60-seconds_b46813`, July 25, 2013.

[26] W. L. Bircher and L. K. John. Complete system power estimation using processor performance events. *IEEE Trans. Comput.*, 61(4):563–577, Apr. 2012.

[27] T. Bostoen, S. Mullender, and Y. Berbers. Power-reduction techniques for data-center storage systems. *ACM Comput. Surv.*, 45(3):33:1–33:38, July 2013.

[28] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: A framework for architectural-level power analysis and optimizations. *SIGARCH Comput. Archit. News*, 28(2):83–94, May 2000.

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

[30] L.-P. Chang. Hybrid solid-state disks: combining heterogeneous nand flash in large ssds. In *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, ASP-DAC '08, pages 428–433, Los Alamitos, CA, USA, 2008. IEEE Computer Society Press.

[31] Y.-H. Chang, C.-K. Hsieh, P.-C. Huang, and P.-C. Hsiu. A caching-oriented management design for the performance enhancement of solid-state drives. *Trans. Storage*, 8(1):3:1–3:21, Feb. 2012.

[32] C.-H. Chao, K.-Y. Jheng, H.-Y. Wang, J.-C. Wu, and A.-Y. Wu. Traffic- and thermal-aware run-time thermal management scheme for 3d noc systems. In *Networks-on-Chip (NOCS), 2010 Fourth ACM/IEEE International Symposium on*, pages 223–230, May 2010.

[33] F. Chen, J. Grundy, J.-G. Schneider, Y. Yang, and Q. He. Automated analysis of performance and energy consumption for cloud applications. In *Proceedings of the 5th ACM/SPEC International Conference on Performance Engineering*, ICPE '14, pages 39–50, New York, NY, USA, 2014. ACM.

[34] F. Chen, J. Grundy, Y. Yang, J.-G. Schneider, and Q. He. Experimental analysis of task-based energy consumption in cloud computing systems. In *Proceedings of the 4th ACM/SPEC International Conference on Performance Engineering*, ICPE '13, pages 295–306, New York, NY, USA, 2013. ACM.

[35] F. Chen, D. A. Koufaty, and X. Zhang. Hystor: making the best use of solid state drives in high performance storage systems. In *Proceedings of the international conference on Supercomputing*, ICS '11, pages 22–32, New York, NY, USA, 2011. ACM.

[36] K.-C. Chen, S.-Y. Lin, and A.-Y. Wu. Design of thermal management unit with vertical throttling scheme for proactive thermal-aware 3d noc systems. In *VLSI Design, Automation, and Test (VLSI-DAT), 2013 International Symposium on*, pages 1–4, April 2013.

[37] D. Chiu, C. Stewart, and B. McManus. Electric grid balancing through lowcost workload migration. *SIGMETRICS Perform. Eval. Rev.*, 40(3):48–52, Jan. 2012.

[38] D. Colarelli and D. Grunwald. Massive arrays of idle disks for storage archives. In *Proceedings of the 2002 ACM/IEEE conference on Supercomputing*, Supercomputing '02, pages 1–11, Los Alamitos, CA, USA, 2002. IEEE Computer Society Press.

[39] T. G. Consortium. 7 strategies to optimize data centre cooling. `http://www.biztechmagazine.com/article/2011/01/keep-your-cool/`, Jan. 2011.

[40] J. Constine. Dropbox is now the data fabric tying together devices for 100m registered users who save 1b files a day. `http://techcrunch.com/2012/11/13/dropbox-100-million/`, Nov. 2012.

[41] G. Cook. How clean is your cloud? Technical report, Greenpeace International, April 2012.

[42] 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.

[43] W. Deng, F. Liu, H. Jin, C. Wu, and X. Liu. Multigreen: Cost-minimizing multi-source datacenter power supply with online control. In *Proceedings of the Fourth International Conference on Future Energy Systems*, e-Energy '13, pages 149–160, New York, NY, USA, 2013. ACM.

[44] P. Desnoyers. Analytic models of ssd write performance. *Trans. Storage*, 10(2):8:1–8:25, Mar. 2014.

[45] T. Diop, N. E. Jerger, and J. Anderson. Power modeling for heterogeneous processors. In *Proceedings of Workshop on General Purpose Processing Using GPUs*, GPGPU-7, pages 90:90–90:98, New York, NY, USA, 2014. ACM.

[46] 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.

[47] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder. Temperature management in data centers: Why some (might) like it hot. *SIGMETRICS Perform. Eval. Rev.*, 40(1):163–174, June 2012.

[48] D. Essary and A. Amer. Sustainable predictive storage management: On-line grouping for energy and latency reduction. In *Proceedings of the 4th Annual International Conference on Systems and Storage*, SYSTOR '11, pages 9:1–9:11, New York, NY, USA, 2011. ACM.

[49] A. Fanara, J. Abelson, A. Bailey, K. Crossman, R. Shudak, A. Sullivan, M. Vargas, and M. Zatz. Report to congress on server and data center energy efficiency. Technical report, U.S. Environmental Protection Agency, August 2007.

[50] K. P. Ganeshpure, I. Polian, S. Kundu, and B. Becker. Reducing temperature variability by routing heat pipes. In *Proceedings of the 19th ACM Great Lakes Symposium on VLSI*, GLSVLSI '09, pages 63–68, New York, NY, USA, 2009. ACM.

[51] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel. The cost of a cloud: Research problems in data center networks. *SIGCOMM Comput. Commun. Rev.*, 39(1):68–73, Dec. 2008.

[52] L. M. Grupp, J. D. Davis, and S. Swanson. The bleak future of nand flash memory. In *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, FAST'12, pages 2–2, Berkeley, CA, USA, 2012. USENIX Association.

[53] S. K. S. Gupta, A. Banerjee, Z. Abbasi, G. Varsamopoulos, M. Jonas, J. Ferguson, R. R. Gilbert, and T. Mukherjee. Gdcsim: A simulator for green data center design and analysis. *ACM Trans. Model. Comput. Simul.*, 24(1):3:1–3:27, Jan. 2014.

[54] 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.

[55] A. Hammadi and L. Mhamdi. Review: A survey on architectures and energy efficiency in data center networks. *Comput. Commun.*, 40:1–21, Mar. 2014.

[56] J. Huang, F. Zhang, X. Qin, and C. Xie. Exploiting redundancies and deferred writes to conserve energy in erasure-coded storage clusters. *Trans. Storage*, 9(2):4:1–4:29, July 2013.

[57] I. E. Insights. Annual it spending by western european utilities to reach 12.7 billion by 2017, Aug 2013.

[58] X. Jiang, M. Al Assaf, J. Zhang, M. Alghamdi, X. Ruan, T. Muzaffar, and X. Qin. Thermal modeling of hybrid storage clusters. *Journal of Signal Processing Systems*, 72(3):181–196, 2013.

[59] X. Jiang, M. Alghamdi, J. Zhang, M. Assaf, X. Ruan, T. Muzaffar, and X. Qin. Thermal modeling and analysis of storage systems. In *Performance Computing and Communications Conference (IPCCC), 2012 IEEE 31st International*, pages 31–40, 2012.

[60] X. Jiang, J. Zhang, M. Alghamdi, X. Qin, M. Jiang, and J. Zhang. Peam: Predictive energy-aware management for storage systems. In *Networking, Architecture and Storage (NAS), 2013 IEEE Eighth International Conference on*, pages 105–114, July 2013.

[61] X. Jimenez, D. Novo, and P. Ienne. Ph&oelig;nix: Reviving mlc blocks as slc to extend nand flash devices lifetime. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '13, pages 226–229, San Jose, CA, USA, 2013. EDA Consortium.

[62] P. Jones. Industry census 2012: Emerging data center markets. `https://www.datacenterdynamics.com/blogs/industry-census-2012-emerging-data-center-markets`, October 2012.

[63] J. Katcher. Postmark: A new file system benchmark. *System*, (3022):1–8, 1997.

[64] A. Kaur and S. Kinger. Temperature aware resource scheduling in green clouds. In *Advances in Computing, Communications and Informatics (ICACCI), 2013 International Conference on*, pages 1919–1923, Aug 2013.

[65] 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.

[66] R. Koller, L. Marmol, R. Rangaswami, S. Sundararaman, N. Talagala, and M. Zhao. Write policies for host-side flash caches. In *Proceedings of the 11th USENIX Conference on File and Storage Technologies*, FAST'13, pages 45–58, Berkeley, CA, USA, 2013. USENIX Association.

[67] J. Kong, S. W. Chung, and K. Skadron. Recent thermal management techniques for microprocessors. *ACM Comput. Surv.*, 44(3):13:1–13:42, June 2012.

[68] R. Kothiyal, V. Tarasov, P. Sehgal, and E. Zadok. Energy and performance evaluation of lossless file data compression on server systems. In *Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, SYSTOR '09, pages 4:1–4:12, New York, NY, USA, 2009. ACM.

[69] Y. Lee and A. Zomaya. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, 60(2):268–280, 2012.

[70] Y. C. Lee and A. Y. Zomaya. Energy efficient utilization of resources in cloud computing systems. *J. Supercomput.*, 60(2):268–280, May 2012.

[71] L. Li, C.-J. M. Liang, J. Liu, S. Nath, A. Terzis, and C. Faloutsos. Thermocast: a cyber-physical forecasting model for datacenters. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '11, pages 1370–1378, New York, NY, USA, 2011. ACM.

[72] Z. Li, K. M. Greenan, A. W. Leung, and E. Zadok. Power consumption in enterprise-scale backup storage systems. In *Proceedings of the 10th USENIX Conference on File and Storage Technologies*, FAST'12, pages 6–6, Berkeley, CA, USA, 2012. USENIX Association.

[73] J. Lin, H. Zheng, Z. Zhu, E. Gorbatov, H. David, and Z. Zhang. Software thermal management of dram memory for multicore systems. *SIGMETRICS Perform. Eval. Rev.*, 36(1):337–348, June 2008.

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

[75] R.-S. Liu, C.-L. Yang, C.-H. Li, and G.-Y. Chen. Duracache: A durable ssd cache using mlc nand flash. In *Proceedings of the 50th Annual Design Automation Conference*, DAC '13, pages 166:1–166:6, New York, NY, USA, 2013. ACM.

[76] Z. Liu, Y. Chen, C. Bash, A. Wierman, D. Gmach, Z. Wang, M. Marwah, and C. Hyser. Renewable and cooling aware workload management for sustainable data centers. *SIGMETRICS Perform. Eval. Rev.*, 40(1):175–186, June 2012.

[77] J. Lu and F. Dawson. Emc computer modeling techniques for cpu heat sink simulation. *Magnetics, IEEE Transactions on*, 42(10):3171–3173, Oct 2006.

[78] T. Luo, S. Ma, R. Lee, X. Zhang, D. Liu, and L. Zhou. S-cave: Effective ssd caching to improve virtual machine storage performance. In *Proceedings of the 22Nd International Conference on Parallel Architectures and Compilation Techniques*, PACT '13, pages 103–112, Piscataway, NJ, USA, 2013. IEEE Press.

[79] B. Mao, H. Jiang, S. Wu, L. Tian, D. Feng, J. Chen, and L. Zeng. Hpda: A hybrid parity-based disk array for enhanced performance and reliability. *Trans. Storage*, 8(1):4:1–4:20, Feb. 2012.

[80] R. Miller. Facebook′s $1 billion data center network. `http://www.datacenterknowledge.com/archives/2012/02/02/facebooks-1-billion-data-center-network/`, February 2012.

[81] M. P. Mills. The cloud begins with coal: Big data, big networks, big infrastructure, and big power. `http://www.tech-pundit.com/wp-content/uploads/2013/07/Cloud_Begins_With_Coal.pdf?c761ac&c761ac`, August 2013.

[82] A. K. Mishra, S. Srikantaiah, M. Kandemir, and C. R. Das. Coordinated power management of voltage islands in cmps. *SIGMETRICS Perform. Eval. Rev.*, 38(1):359–360, June 2010.

[83] 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.

[84] D. Narayanan, A. Donnelly, and A. Rowstron. Write off-loading: Practical power management for enterprise storage. In *Proceedings of the 6th USENIX Conference on File and Storage Technologies*, FAST'08, pages 17:1–17:15, Berkeley, CA, USA, 2008. USENIX Association.

[85] D. Narayanan, E. Thereska, A. Donnelly, S. Elnikety, and A. Rowstron. Migrating server storage to ssds: Analysis of tradeoffs. In *Proceedings of the 4th ACM European Conference on Computer Systems*, EuroSys '09, pages 145–158, New York, NY, USA, 2009. ACM.

[86] Y. Oh, J. Choi, D. Lee, and S. H. Noh. Improving performance and lifetime of the ssd raid-based host cache through a log-structured approach. In *Proceedings of the 1st Workshop on Interactions of NVM/FLASH with Operating Systems and Workloads*, INFLOW '13, pages 5:1–5:8, New York, NY, USA, 2013. ACM.

[87] E. Pakbaznia, M. Ghasemazar, and M. Pedram. Temperature-aware dynamic resource provisioning in a power-optimized datacenter. In *Proceedings of the Conference on Design, Automation and Test in Europe*, DATE '10, pages 124–129, 3001 Leuven, Belgium, Belgium, 2010. European Design and Automation Association.

[88] A. Pavlo, E. Paulson, A. Rasin, D. J. Abadi, D. J. DeWitt, S. Madden, and M. Stonebraker. A comparison of approaches to large-scale data analysis. In *Proceedings of the 2009 ACM SIGMOD International Conference on Management of data*, SIGMOD '09, pages 165–178, New York, NY, USA, 2009. ACM.

[89] E. Pinheiro and R. Bianchini. Energy conservation techniques for disk array-based servers. In *Proceedings of the 18th annual international conference on Supercomputing*, ICS '04, pages 68–78, New York, NY, USA, 2004. ACM.

[90] E. Pinheiro, W.-D. Weber, and L. A. Barroso. Failure trends in a large disk drive population. In *Proceedings of the 5th USENIX conference on File and Storage Technologies*, pages 2–2, Berkeley, CA, USA, 2007. USENIX Association.

[91] 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.

[92] S. Ren and Y. He. Coca: Online distributed resource management for cost minimization and carbon neutrality in data centers. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '13, pages 39:1–39:12, New York, NY, USA, 2013. ACM.

[93] A. Riska and E. Smirni. Autonomic exploration of trade-offs between power and performance in disk drives. In *Proceedings of the 7th International Conference on Autonomic Computing*, ICAC '10, pages 131–140, New York, NY, USA, 2010. ACM.

[94] A. Sansottera and P. Cremonesi. Cooling-aware workload placement with performance constraints. *Perform. Eval.*, 68(11):1232–1246, Nov. 2011.

[95] 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.

[96] 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.

[97] D. Schall, V. Hudlet, and T. Härder. Enhancing energy efficiency of database applications using ssds. In *Proceedings of the Third C* Conference on Computer Science and Software Engineering*, C3S2E '10, pages 1–9, New York, NY, USA, 2010. ACM.

[98] P. Sehgal, V. Tarasov, and E. Zadok. Optimizing energy and performance for server-class file system workloads. *Trans. Storage*, 6(3):10:1–10:31, Sept. 2010.

[99] A. Shah, V. Carey, C. Bash, C. Patel, and R. Sharma. Exergy analysis of data center thermal management systems. In Y. Joshi and P. Kumar, editors, *Energy Efficient Thermal Management of Data Centers*, pages 383–446. Springer US, 2012.

[100] M. Sharifi, H. Salimi, and M. Najafzadeh. Power-efficient distributed scheduling of virtual machines using workload-aware consolidation techniques. *J. Supercomput.*, 61(1):46–66, July 2012.

[101] 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.

[102] J.-Y. Shin, M. Balakrishnan, L. Ganesh, T. Marian, and H. Weatherspoon. Gecko: A contention-oblivious design for cloud storage. In *Proceedings of the 4th USENIX Conference on Hot Topics in Storage and File Systems*, HotStorage'12, pages 4–4, Berkeley, CA, USA, 2012. USENIX Association.

[103] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, 1(1):94–125, Mar. 2004.

[104] M. Song, Y. Lee, and E. Kim. Saving disk energy in video servers by combining caching and prefetching. *ACM Trans. Multimedia Comput. Commun. Appl.*, 10(1s):15:1–15:21, Jan. 2014.

[105] 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.

[106] Statista. Number of monthly active facebook users worldwide from 3rd quarter 2008 to 1st quarter 2014 (in millions), 2014.

[107] 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.

[108] 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.

[109] 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.

[110] 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.

[111] P. Thibodeau. Data centers use 2% of u.s. energy, below forecast, August 2011.

[112] W. Tian, Q. Xiong, and J. Cao. An online parallel scheduling method with application to energy-efficiency in cloud computing. *J. Supercomput.*, 66(3):1773–1790, Dec. 2013.

[113] N. Vasic, T. Scherer, and W. Schott. Thermal-aware workload scheduling for energy efficient data centers. In *Proceedings of the 7th international conference on Autonomic computing*, ICAC '10, pages 169–174, New York, NY, USA, 2010. ACM.

[114] J. Whitney and J. Kennedy. Is cloud computing always greener? Technical report, Natural Resources Defense Council, October 2012.

[115] G. Wu, X. He, and B. Eckart. An adaptive write buffer management scheme for flash-based ssds. *Trans. Storage*, 8(1):1:1–1:24, Feb. 2012.

[116] T. Xie and Y. Sun. Understanding the relationship between energy conservation and reliability in parallel disk arrays. *J. Parallel Distrib. Comput.*, 71:198–210, February 2011.

[117] F. Yan, X. Mountrouidou, A. Riska, and E. Smirni. Quantitative estimation of the performance delay with propagation effects in disk power savings. In *Proceedings of the 2012 USENIX Conference on Power-Aware Computing and Systems*, HotPower'12, pages 5–5, Berkeley, CA, USA, 2012. USENIX Association.

[118] G.-W. You, S.-W. Hwang, and N. Jain. Ursa: Scalable load and power management in cloud storage systems. *Trans. Storage*, 9(1):1:1–1:29, Mar. 2013.

[119] M. Zapater, J. L. Ayala, and J. M. Moya. Leveraging heterogeneity for energy minimization in data centers. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (Ccgrid 2012)*, CCGRID '12, pages 752–757, Washington, DC, USA, 2012. IEEE Computer Society.

[120] Z. Zhang and S. Fu. Characterizing power and energy usage in cloud computing systems. In *Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science*, CLOUDCOM '11, pages 146–153, Washington, DC, USA, 2011. IEEE Computer Society.