TIGER:Thermal-Aware File Assignment in Storage Clusters

Ajit Chavan[†], Xunfei Jiang[†], Mohemmad I. Alghamdi[‡], Xiao Qin[†],Minghua Jiang[§], and Jifu Zhang[¶]
[†]Department of Computer Science and Software Engineering, Auburn University, Auburn, USA
[‡] Department of Computer Science, Al-Baha University, Kingdom of Saudi Arebia
[§] College of Mathematics and Computer Science, Wuhan Textile University, Wuhan, China
[¶]School of Computer Science and Technology, Taiyuan University of Science and Technology, Taiyuan, China

***Abstract*—In this paper, we present thermal-aware file assignment technique called TIGER for reducing cooling cost of storage clusters in data centers. TIGER first calculates the thresholds of disks in each node based on its contribution to heat recirculation in a data center. Next, TIGER assigns files to data nodes according to calculated thresholds. We evaluated performance of TIGER in terms of both cooling energy conservation and response time of a storage cluster. Our results confirm that TIGER reduces cooling-power requirements for clusters by offering about 10 to 15 percent cooling energy savings without significantly degrading I/O performance.**

I. Introduction

Thermal management for power-dense storage clusters can address cooling problems in today's data centers. In this paper, we show that thermal-aware file assignment policies can significantly reduce cooling cost of data center by lowering peak inlet temperatures of storage clusters.

The following three factors make thermal-aware file assignment desirabe and practical for storage clusters:

- the high cooling cost of large scale data centers,
- the rapid heat recirculation caused by data nodes in storage clusters, and
- the ability of file assignment policies to manage utilization of data nodes based on I/O access patterns.

Data nodes in storage clusters are typically configured with low power processors and RAID arrays containing multiple (4 to 32) disks. Modern storage systems account for almost 27% of total energy consumption [4]. Energy and cooling cost caused by data nodes motivate us to study file assignment solutions that can reduce inlet temperatures of data nodes.

The recirculation of hot air from the outlet of data nodes back to their inlets inevitably raise inlet temperatures and may cause hot spots [8], which forces computer room air conditions to continuously work at lower temperatures, increasing cooling cost. The goal of this study is to minimize heat recirculation and cooling cost, thereby increasing energy efficiency of data centers housing storage clusters.

Disks have non-negligible thermal impact on data nodes [2]. We developed thermal model to estimate the inlet temperature of storage servers based on processor and disks utilizations. We compared response time and cooling cost of storage systems managed by three data placement strategies, among which one can noticeably reduce cooling cost of storage systems in data centers. In this paper, we aim to develop a file placement scheme- TIGER - to offer tradeoffs between performance and thermal profile in storage clusters.

At the core of our TIGER approach, peak inlet temperatures of data nodes are reduced by the virtue of thermal-aware file assignment. The file assignment process relies on I/O loads thresholds that can be derived in two steps. First, TIGER applies cross-intereference coefficients to calculate contributions of each node to the heat recirculation of an entire storage clusters. Next, TIGER calculates the load thresholds of disks in each data nodes based on its contribution to heat recirculation.

978-1-4799-0218-7/13/$31.00 2013 IEEE

II. Modeling

A. Power Model

Clusters in a data center are comprised of both computing nodes and data nodes. The terms data nodes and storage nodes are used interchangeably throughout this paper. Let P_{comp} be the power consumed by computing nodes and $P_{storage}$ be the power consumed by storage nodes, which is nothing but summation of power consumed by each data node. Therefore, total power consumption P_C of cluster in data center can be calculated by:

$$P_C = P_{comp} + \sum_{i=1}^{N} P_i^{node}. \quad (1)$$

where, P_i^{node} is power consumption of i^{th} data node.

The power consumption P_i^{node} in equation 1 can be derived from (1) a fixed amount of power P^{base} consumed by node i's hardware (e.g., fans) other than processor and disks, (2) power P^{cpu} consumed by node i's CPU and power P^d consumed by disks residing in the node, which is summation of power consumption by each disk in residing in the node. Thus we can calculate P_i^{node} as:

$$P_i^{Node} = P_i^{base} + P_i^{CPU} + \sum_{j=1}^{D_i} P_{i,j}^d \quad (2)$$

where, $P_{i,j}^d$ is power consumed by j^{th} disk in i^{th} data node and D_i is total number of disks in i^{th} data node.

In what follows, we model the power consumption $P_{i,j}^d$ of disk j in storage node i. Disks have three modes of

operations: active, idle and sleep, each of which has a specific power requirement. We denote the power consumed by a single disk in the active, idle and in the sleep mode as $P^{d,active}, P^{d,idle}, P^{d,sleep}$, respectively. Power overhead is incurred when disks are transitioning among the mode (e.g., from the sleep mode to active or vice versa). We denote the power required to spin down a disk as $P_{S_{down}}$ and power needed to spin up the disks as $P_{S_{up}}$. Given a time interval T, let $t_{i,j}^{active}$, $t_{i,j}^{idle}$, and $t_{i,j}^{sleep}$ represent time periods when disk j in node i is active, idle, and sleep, respectively. We denote $N_{i,j}^{t}$ as the number of power-state transitions. Now, we model the disk power consumption $P_{i,j}^{d}$ as:

$$P_{i,j}^{d} = \frac{1}{T}\Big(t_{i,j}^{active} \times P_{i,j}^{d,active} + t_{i,j}^{idle} \times P_{i,j}^{d,idle} + t_{i,j}^{sleep} \times P_{i,j}^{d,sleep} + \frac{N_{i,j}^{t}}{2}\left(P_{S_{down}} + P_{S_{up}}\right)\Big) \quad (3)$$

B. Heat Recirculation Model

Handful of models have been proposed to characterize the heat recirculation in data centers [3] [6] [9]. All those models are well investigated and well validated and they predict the inlet temperature of nodes in cluster with reasonable accuracy. We used the model proposed by Gupta *et al.* [9], which characterized heat recirculation by a cross-interference matrix $A_{n\times n} = \alpha_{i,j}$, where $\alpha_{i,j}$ denotes fraction of its outlet heat node *i* contributes to node *j*. Therefore, according to the model proposed in [9], the vector of inlet temperature can be calculated by:

$$t_{in} = t_{sup} + \left[(K - A^{\intercal}K)^{-1} - K^{-1}\right] p. \quad (4)$$

where, t_{in} is the vector of inlet temperatures T_{in}, t_{sup} is vector of supply temperature of CRAC T_{sup}, p is vector of power consumption of each node P^{node}. $K_{n\times n}$ is a diagonal matrix of thermodynamic constants K_i:

$$K_i = \rho a_i c_p \quad (5)$$

where, ρ is the air density (in grams per cubic meter), a_i is the airflow rate (in cubic meters per seconds) of the node *i*, and c_p is the specific heat of the air (in joules per gram Kelvin).

C. Cooling Cost Model

Heat recirculation and node power consumption lead to an increase of inlet temperature. To control the raised inlet temperatures below redline, a cooling system is applied. Temperature of the air supplied by the cooling system is adjusted according to the maximum inlet temperature. Supply temperature T_{sup} affects the efficiency of the cooling system. The efficiency of cooling system is quantified in terms of *Coefficient of Performance* (COP) [5] [8] (see (6) below).

$$COP(T_{sup}) = 0.0068T_{sup}^{2} + 0.0008T_{sup} + 0.458 \quad (6)$$

COP increases when supply temperature goes up; increasing supply temperatures results in high cooling efficiency. (7) shows how to derive cooling cost from COP.

$$P_{AC} = \frac{P_C}{COP(T_{sup})} \quad (7)$$

where P_C is the total power consumed by the storage nodes in data center [8].

III. TIGER : Thermal Aware File Assignment For Data Centers

A. Basic Ideas

The goal of TIGER is place a set of *m* files to a group of *N* nodes in such a way to reduce cooling cost of a data center. The service time s_i and access rate λ_i of file f_i are provided by a file access predictor (see, for example, [1]). The algorithm is comprised of two phases. In the first phase, thresholds for disk utilization is determined (see Section III-B). In the second phase, files are assigned to storage nodes until the threshold is reached (see Section III-C).

In the process of calculating the disk utilization threshold, we take into account both performance and thermal management. To improve I/O performance, we apply a load balancing strategy to uniformly distribute I/O load among all the disks. When it comes to thermal management, we follow the principle that workload placed on the node should be inversely proportional to the contribution of the node in the heat recirculation in a data center. To place workload uniformly according to this principle, one has to ensure that all the nodes should contribute equally in heat recirculation. Achieving this goal may be difficult and; therefore, it normally useful to have a calibration phase, where we adjust the calculated threshold according to each node's contribution in the heat recirculation.

During the file assignment procedure, the list of nodes are sorted in the increasing order of their heat-recirculation contribution. For each node in the list, files are assigned to each disk on the node until the threshold has been reached. We keep doing this until either the node list is empty or there are no files remaining. If the node list is empty and there are some files remaining, then we will start from the first node in the node list and keep assigning files until either utilization reaches 90% or all files have been assigned.

B. Disk Utilization threshold calculation

Now we discuss how to calculate disk utilization threshold to be used in the second phase of our approach. Recall that the utilization threshold is introduced to guide the file assignment procedure, which affects node power consumption that make significant impacts on the outlet and inlet temperatures.

As mentioned earlier, we first calculate the threshold using the load balancing strategy. The utilization of the disk d_i is increased by u_i due to the allocation of file f_i. The utilization u_i is a product of service time s_i and access rate λ_i of the file. Therefore:

$$u_i = s_i * \lambda_i \quad (8)$$

Our file assignment algorithm aims to distribute the total utilization *U* generated by all the files to *D* disks. We use the greedy algorithm to uniformly balance load among all

the available disks. Disk utilization threshold U_{avg}^{Th} can be calculated using the following expression:

$$U_{avg}^{Th} = \frac{1}{D}\sum_{i=1}^{m} s_i \times \lambda_i \tag{9}$$

This average threshold can be adjusted according to each node's contribution in the heat recirculation of the data center. We characterize the heat recirculation as cross-interference coefficient. Then, the total contribution of node in the heat recirculation of a data center can be considered as sum of all the cross-interference coefficients of the node normalized over sum of all the cross-interference coefficients of all the nodes. Therefore,

$$S_i = \frac{\sum_{j=1}^{n} \alpha_{i,j}}{S_{total}} \tag{10}$$

where, S_{total} is sum of all the cross interference coefficients of all the nodes in a cluster.

Ideally, uniformly distributing workload makes all the nodes identical in terms of heat recirculation. Thus, we have:

$$S_i = S_{avg}, \quad \forall i \in N \tag{11}$$

where N is set of all nodes and $S_{avg} = \frac{1}{n}$.

Although the above expression shows the best case, (11) does not hold for most of the nodes in the data center. In real-world scenarios, a node's contribution to heat recirculation might be either higher or lower than the average contribution S_{avg}. This leads us to discuss the following two cases.

Case 1: $S_i > S_{avg}$

This case holds for most nodes that are nearer to the floor surface. We calculate the normalized difference between S_i and S_{avg} (see (12)) and decrease the threshold for the disks in node *i* by the normalized difference.

$$\Delta S = \frac{S_i - S_{avg}}{S_{avg}} \tag{12}$$

$$U_i^{Th} = U_{avg}^{Th} - \left(\Delta S \times U_{avg}^{Th}\right) \tag{13}$$

Case 2: $S_i < S_{avg}$

This case holds for most of the nodes nearer to the ceiling. As these nodes contribute less to the total heat recirculation of the data center, we place more workload on these nodes. We calculate the normalized difference (see (14)) between S_{avg} and S_i; the disk utilization threshold for these node is increased by the normalized difference.

$$\Delta S = \frac{S_{avg} - S_i}{S_{avg}} \tag{14}$$

$$U_{high}^{Th} = U_{avg}^{Th} + \left(\Delta S \times U_{avg}^{Th}\right) \tag{15}$$

$$U_i^{Th} = \begin{cases} U_{high}^{Th} & \text{if } U_{high}^{threshold} < 1.0 \\ 1.0 & \text{if } U_{high}^{threshold} > 1.0 \end{cases} \tag{16}$$

C. Tiger: Algorithm

The Tiger algorithm solves the thermal management problem by applying thermal-aware file assignment in data centers. Tiger relies on file access patterns and the amount of heat recirculation to make file placement decisions.

Algorithm 1: TIGER(file_info, node_info)

1: $U \leftarrow 0$
2: **for** $f_i \in m$ **do**
3: $\quad U \leftarrow U + s_i \times \lambda_i$
4: **end for**
5: $U_{avg}^{Th} \leftarrow \frac{1}{D}U$
6: $S_{total} \leftarrow 0$
7: **for** node $i = 1 \rightarrow N$ **do**
8: $\quad S_i \leftarrow \sum_{j=0}^{n} \alpha_{ij}$
9: $\quad S_{total} \leftarrow S_{total} + S_i$
10: **end for**
11: $S_{avg} \leftarrow \frac{1}{N}$
12: sort the nodes according to S_i
13: $k \leftarrow 0$
14: **for all** node $i \in$ sorted list **do**
15: $\quad S_i \leftarrow \frac{S_i}{S_{total}}$
16: $\quad$ **if** $S_i > S_{avg}$ **then**
17: $\quad\quad$ Calculate threshold using equation (13)
18: $\quad$ **end if**
19: $\quad$ **if** $S_i < S_{avg}$ **then**
20: $\quad\quad$ Calculate threshold using equation (16)
21: $\quad$ **else**
22: $\quad\quad U_i^{Th} \leftarrow U_{avg}^{Th}$
23: $\quad$ **end if**
24: $\quad$ **for all** disk $j \in D_i$ **do**
25: $\quad\quad$ **while** $U_j < U_i^{Th}$ **do**
26: $\quad\quad\quad$ assign file f_k to disk j
27: $\quad\quad\quad U_j \leftarrow U_j + (\lambda_k \times s_k)$
28: $\quad\quad\quad k \leftarrow k + 1$
29: $\quad\quad$ **end while**
30: $\quad$ **end for**
31: **end for**
32: **if** $k < m$ **then**
33: $\quad$ {still some files are remaining}
34: $\quad$ Start from the first node of the sorted list,
35: $\quad$ keep assigning files to the disk in the node until the utilization of the disks reaches 0.9
36: $\quad$ Repeat line 35 for consequent nodes in the sorted list until k=m.
37: **end if**

Prior to making any file placement decision, Tiger calculates the average disk utilization threshold U_{avg}^{Th} (see lines 2-5), thereby using the greed method to uniformly distribute I/O load among available disks. After the initial assignment is complete, Tiger computes three important factors (i.e., S_{avg}, S_i, and S_{total}, which are used to calibrate the disk utilization threshold of each node(see lines 6-11). Next, Tiger sorts the list of nodes in an ascending order of their heat recirculation

contribution S_i (see line 12). Tiger then picks the first node from the sorted node list, and adjusts the disk utilization threshold for all the disks in the selected node depending upon the values of S and S_{avg} (see lines 14-23). Finally, Tiger assigns files to each disk in the selected node until either the threshold is reached or the disk's free capacity becomes empty (lines 25-29). Tiger repeatedly performs steps 14-29 until all the files are placed to the disks.

If the node list is empty and there are some files remaining, then we will start from the first node in the node list and keep assigning files until either utilization reaches 90% or all files have been assigned(lines 34-36).

IV. Evaluation

A. Baseline Algorithms

To evaluate TIGER's system performance, we choose the following two baseline algorithms to compare against TIGER. The first one is a greedy load-balancing algorithm; the second one is the coolest Inlet algorithm.

1) The Greedy Load-balancing Algorithm: The greedy load balancing algorithm uniformly distribute I/O load among all available disks in data nodes. For fair comparisons, a prediction module offers the greedy algorithm with the service time s_i and access rate λ_i of each file (i.e., file f_i). The greedy algorithm calculates the total I/O load caused by requests accessing all the files. The algorithm then uniformly distributes the I/O load to all the disks.

2) Coolest Inlet [8]: This algorithm distributes workload based on inlet temperatures of nodes. It places more workload on the nodes with lower inlet temperature. For example, the threshold of nodes is inversely proportional to the inlet temperature of the nodes. The files are assigned to the disks upto its threshold, which is identical for all the disks in a node.

B. Experimental Setup

We use a simulator written in C for our simulation study. For most of the tests, the data center contains 2 rows of 5 racks each. A rack contains 5 chassis(or nodes), each of which contains six 1U RAID arrays. Every RAID array contains a RAID controller (no processor) and 4 hot swappable disks and draws 118 W power when no disks are attached. Therefore, we have:

$$P_a^{idle} = 118W \tag{17}$$

C. Thermal Impact of Energy Efficient Disks

1) Scenario 1: Figure 1 shows the results for the best case scenario. In this case, we assume that an efficient energy saving algorithm is used so that when the disks are not in active mode, they are spun down to the sleep mode. The power consumed by a disk have three components: power consumed by the disk in the active mode, power consumed by the disk in the sleep mode, and the power consumed by the disk to make transitions between different states. Therefore, Equation 3 is simplified to:

$$P_{i,j}^d = \frac{1}{T}\left(t_j^{active} \times P_{i,j}^{d,active} + t_j^{sleep} \times P_{i,j}^{d,sleep} + \frac{N_j^t}{2}\left(P_{S_{down}} + P_{S_{up}}\right)\right) \tag{18}$$

We observe from both Figure 1(a) and 1(b) that TIGER conserves more cooling energy than the other two algorithms. The difference in the performance is substantial (almost 15%) for data center utilization between 30-60% diminishing towards the two extreme ends. This is because, with data center utilization between 30-60%, there is great opportunity to unbalance the workload in order to achieve thermal benefits. Towards the both extreme cases, there is not much room available for unbalancing the workload to achieve thermal benefits.

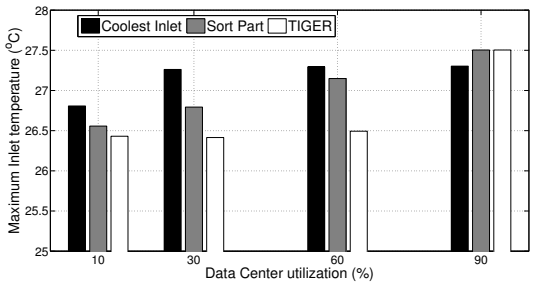
2) Scenario 2: Figure 2 shows the results for the case where no energy efficient techniques are used to spin up and spin down the disks. This is the worst case scenario in terms of energy savings. The disk would be either in one of the two states - active or idle. There are no transitions from the active mode to the sleep mode and vice versa. Then, no extra power is consumed for transitions. Therefore, Equation 3 becomes:

$$P_{i,j}^d = \frac{1}{T}\left(t_j^{active} \times P_{i,j}^{d,active} + t_j^{idle} \times P_{i,j}^{d,idle}\right) \tag{19}$$

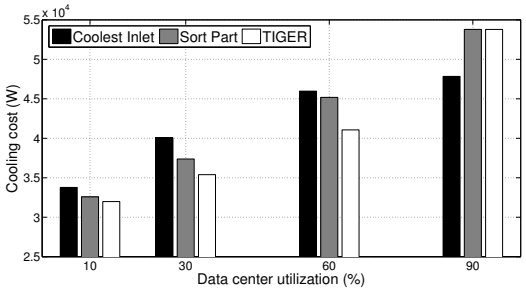
From Figure 2 we can see that though the TIGER outperforms the other two algorithms, the differences among the three solutions are very small. The pwe discrepancy between the active mode and the idle mode is almost negligible. Therefore, the distribution of power among the nodes in the data center does not vary much due to the workload distribution. Also, as the idle disks consume nearly equal power as the active disks, the overall power consumption in the data center is very high. This power feature results in high cooling cost for scenario 2.

V. Related Work

Thermal-aware workload placement strategies were proposed in recent studies [7] [8], which indicate that energy efficiency of CRAC can be improved by reducing the peak temperature in data center. For example, both a generic algorithm [8] and sequential quadratic programming approach [7] were developed to manage workload in a way to reduce the maximum inlet temperatures. Reducing the negative impact of heat recirculation is a new step towards saving cooling energy. For example, Moore *et al.* designed two approaches, ZBD and MinHR [5]. The ZBD scheme that uses paoching at where the effect of the heat recirculation is ovserved whereas MinHR manages workload in a way that each pod in a data center generates same amount of heat to minimize heat recirculation [5]. Wang *et al.* proposed a way of calculating heat generated by jobs, which are sorted in descending order of their hotness [10]. All the above strategies focused on computing nodes and used linear power model driven by CPU utilization. Unlike these techniques, our TIGER approach aims to reduce heat recirculation through file assignment.

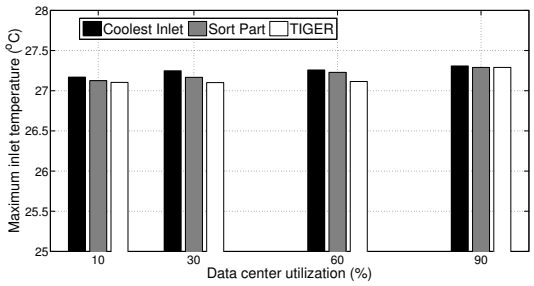


(a) Maximum T^{in}

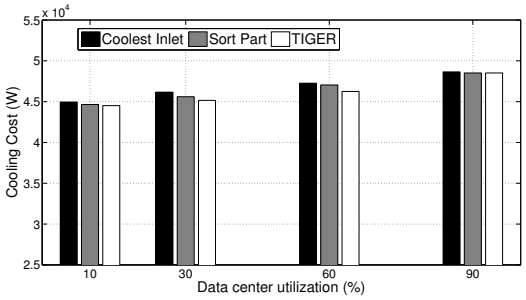


(b) Cooling Cost

Fig. 1: Comparison of algorithms under scenario 1: When disk is not active, it is always turned down in sleep mode.



(a) Maximum T^{in}



(b) Cooling Cost

Fig. 2: Comparison of algorithms under scenario 2: When disk is not active, it is always in idle state.

VI. Conclusion

In this paper, we proposed and implemented TIGER, a file assignment approach to reducing cooling energy requirements of data centers. TIGER first decides disk utilization threshold based on inlet temperatures of data nodes. Then, files are assigned to disks in each node provided that disk utilization is below the corresponding threshold. We applied cross-interference coefficients to estimate the recirculation of hot air from the outlets to the inlets of data nodes. We implemented TIGER in an HP server. Our experimental results confirm that TIGER is capable of offering about 10 to 15 percent cooling-energy savings without significantly degrading I/O performance.

Acknowledgment

This work is supported by the U.S. National Science Foundation under Grants CCF-0845257 (CAREER), CNS-0917137 (CSR), CNS-0757778 (CSR), CCF-0742187 (CPA), CNS-0831502 (CyberTrust), CNS-0855251 (CRI), OCI-0753305 (CI-TEAM), DUE-0837341 (CCLI), and DUE-0830831 (SFS).

References

[1] Rini T. Kaushik and Klara Nahrstedt. T: a data-centric cooling energy costs reduction approach for big data analytics cloud. In *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*, SC '12, pages 52:1–52:11, Los Alamitos, CA, USA, 2012. IEEE Computer Society Press.

[2] Y. Kim, S. Gurumurthi, and A. Sivasubramaniam. Understanding the performance-temperature interactions in disk i/o of server workloads. In *HPCA*, pages 176–186, 2006.

[3] Lei Li, Chieh-Jan Mike Liang, Jie Liu, Suman Nath, Andreas Terzis, and Christos Faloutsos. Thermocast: A cyber-physical forecasting model for data centers. In *Proc. KDD*, volume 11, 2011.

[4] A. Manzanares, X. Qin, X. Ruan, and S. Yin. Pre-bud: Prefetching for energy-efficient parallel i/o systems with buffer disks. *Trans. Storage*, 7(1):3:1–3:29, June 2011.

[5] 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.

[6] 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, 2008.

[7] 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.

[8] 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. *Parallel and Distributed Systems, IEEE Transactions on*, 19(11):1458 –1472, nov. 2008.

[9] Q. Tang, T. Mukherjee, S. K S Gupta, and P. Cayton. Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. In *Intelligent Sensing and Information Processing, 2006. ICISIP 2006. Fourth International Conference on*, pages 203–208, 2006.

[10] L. Wang, G. von Laszewski, J. Dayal, X. He, A.J. Younge, and T.R. Furlani. Towards thermal aware workload scheduling in a data center. In *Pervasive Systems, Algorithms, and Networks (ISPAN), 2009 10th International Symposium on*, pages 116 –122, dec. 2009.