Global Fan Speed Control Considering Non-Ideal Temperature Measurements in Enterprise Servers

†‡Jungsoo Kim
†DMC Research Center
Samsung Electronics
jungsoo9.kim@samsung.com

‡Mohamed M. Sabry ‡David Atienza
‡Embedded Systems Lab
EPFL
{mohamed.sabry,david.atienza}@epfl.ch

*Kalyan Vaidyanathan *Kenny Gross
*Physical Sciences Research Center
Oracle
{kalyan.vaidyanathan,kenny.gross}@oracle.com

***Abstract*—Time lag and quantization in temperature sensors in enterprise servers lead to stability concerns on existing variable fan speed control schemes. Stability challenges become further aggravated when multiple local controllers are running together with the fan control scheme. In this paper, we present a global control scheme which tackles the concerns on the stability of enterprise servers while reducing the performance degradation caused by the variable fan speed control scheme. We first present a stable fan speed control scheme based on the Proportional-Integral-Derivative (PID) controller by adaptively adjusting the PID parameters according to the operating fan speed and eliminating the fan speed oscillation caused by temperature quantization. Then, we present a global control scheme which coordinates control actions among multiple local controllers. In addition, it guarantees the server stability while minimizing the overall performance degradation. We validated the proposed control scheme using a presently shipping commercial enterprise server. Our experimental results show that the proposed fan control scheme is stable under the non-ideal temperature measurement system (10 sec in time lag and 1$^{\circ}C$ in quantization figures). Furthermore, the global control scheme enables to run multiple local controllers in a stable manner while reducing the performance degradation up to 19.2% compared to conventional coordination schemes with 19.1% savings in power consumption.**

I. Introduction

The soaring demand for computing has produced as collateral undesirable effect a surge in power consumption of servers [1]. Among various solutions to reduce the power consumption of computing servers, a variable fan speed control scheme is promising as it can save a significant amount of energy consumption by lowering the fan speed, as the power consumed by fans has a cubic relationship with fan speed, i.e., $P_{fan} \propto s_{fan}^3$ [2]. Recent enterprise systems have adopted variable-speed fans for cooling with a view to minimize energy usage, acoustics and vibration levels [8]. However, the new dynamic fan control concept is only conservatively used with simple single threshold or deadzone control schemes due to the stability concerns of using sophisticated control solutions in presently shipping commercial enterprise servers.

The reason why such simple controllers cannot work well in enterprise servers are mostly coming from non-ideal temperature measurement systems: 1) time lags between physical transducers and control firmware and 2) signal quantization. First, in most systems, there is a significant delay between the instantly sensed telemetry metrics and when they get through to the control algorithm that is embedded in the firmware that typically resides in the server's Service Processor or Baseboard Management Console (BMC) [3]. According to our measurements using presently shipping commercial enterprise servers, which is demonstrated in Fig. 1, it amounts to ∼10 sec[1]. It is due to the limited bandwidth of I^{2}C bus which has become a de-facto standard on the bus protocol used for temperature measurement systems [4]. Moreover, due to the increased number of temperature sensors in each new server platform, the time lag from bandwidth contention becomes even worse in newer generation servers. Second, as a result of the standardized usage of 8-bit A/D converters for physical sensors in systems, the reported readings are severely quantized. Signal quantization causes large uncertainties in computed metrics and jitter/inefficiencies in the feedback-control loops. Thus, we need to develop a stable variable fan control scheme which is robust to these non-ideal effects.

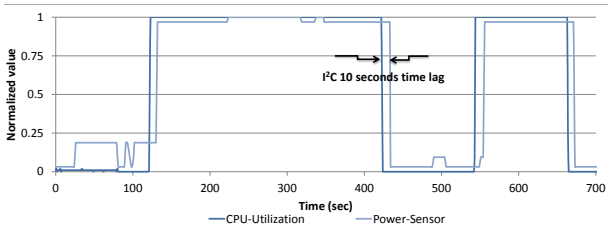

Fig. 1. Power sensor measurements following workload changes in a CPU. The observed change in temperature suffers from a 10 second lag due to the I^2C delay.

In addition to the foregoing stability challenges with variable fan speed controller, a more important challenge is arising now due to the existence of multiple local controllers in enterprise servers, e.g., CPU power management via dynamic voltage and frequency scaling (DVFS) and power gating [11], [12], and temperature-aware workload scheduling in the operating system (OS) [13], [14]. As an example, thermal designers use CPU temperatures as an input to fan control algorithm to keep

This work was supported in part by and ERO Research Grant from Oracle for ESL-EPFL, and the EC FP7 STREP GreenDataNet project (no. 609000).

[1]We obtain the data through private communication from our industrial partner from installed enterprise servers. However, we omit the family of the servers used in our experiments due to confidential reasons.

978-3-9815370-2-4/DATE14/©2014 EDAA

these temperatures inside a comfort zone window for reliability assurance. Meanwhile, processor designers implement P-state power management for the CPUs with CPU temperatures as input variables to achieve thermal capping. At the same time, OS developers work on intelligent thermally-aware workload scheduling to control the CPU temperature. If two or all three of these local controllers are active simultaneously in future servers, dynamic instability can (and most certainly will) ensue. In this work, we present a global control scheme for servers equipped with a variable fan speed control which assures stability while jointly optimizing the performance and the power consumption.

The contributions of this work are the following:

- Analysis of the temperature measurement lag and the quantization errors that lead to server systems instability in a variable fan speed controller environment.
- Designing a model-free fan speed controller which is resilient to the non-ideal effects in the temperature measurement systems of enterprise servers.
- Designing a low-complexity global coordination scheme to guarantee server operation stability when multiple independent local controllers are working together.

The rest of this paper is organized as follows. Section II reviews the related works. Section III describes the target server system and its temperature model. Section IV explains the new fan controller design, which is robust to non-ideal temperature measurement effects. Section V explains our global coordinating solution. Section VI validates the proposed control scheme, followed by conclusions in Section VII.

II. Related Work

Many works have proposed control solutions for the variable fan speed control [2], [5]–[8]. However, all of them ignore the non-ideal effects in the temperature measurement systems. In addition, the effectiveness of the solutions depends on the accuracy of the models as the control solutions are derived from the models. In order to make the control solution stable accounting for the non-ideal effects as well as model-free, we present a fan speed control solution based on Proportional-Integral-Derivative (PID) control [9] that is robust with respect to the challenges of signal quantization and system bus latencies. PID is one of the most widely used control solutions due to its simplicity while guaranteeing stability, accuracy, settling time, and overshoot (SASO) criterion by simply adjusting PID parameters. However, simply applying the PID control solution is not sufficient to a variable fan speed control as the relationship between temperature and fan speed is highly nonlinear and vulnerable to the measurement quantization [10].

Multiple control knobs in servers are jointly manipulated to achieve further power savings while satisfying thermal limits. In [6], [14], they present proactive thermal-aware workload management solutions to distribute the workload among cores while minimizing the cooling energy costs of fan subsystems. They compare the effectiveness of multiple control actions, i.e., ratio of temperature reduction to energy increase against multiple control actions (e.g., migrating workload vs. increasing fan speed when a thermal emergency happens), and then, select the control action that yields the best efficiency. However, it can lead to huge performance degradation as it does not take into account the impact to the performance degradation because each controller locally decides the action. [15] presents a coordinated control scheme which theoretically guarantees the stability and the accuracy when multiple local controllers are running together in a system while achieving the desired power and performance control objectives. However, it only focuses on manipulating variables in computing parts while not adjusting the fan speed of servers. Of particular note, we compare the proposed solutions with [6] in the experimental section as [14] is similar with [6] and [15] does not account for the fan speed control scheme.

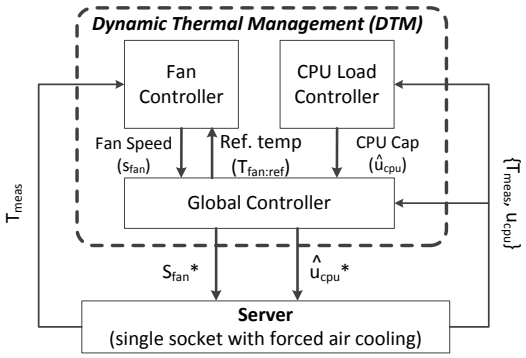

Fig. 2. Proposed target computing/server system

III. Target server systems

A. System overview

Fig. 2 shows the target system and its controller. In this work, we focus on the computing and the cooling subsystems of servers, i.e., CPU and fan. For the sake of simplicity, we target a server consisting of N_{core} cores assuming that running workloads are perfectly balanced among the cores, which implies that multiple fans in a server run at the same speed. Temperature sensors are located at each core and deliver measured temperatures to the dynamic thermal management (DTM) unit, quantized and time-lagged, due to the underlying signal acquisition standards (i.e., Analog-to-Digital Converter (ADC) and I^2C bus). The target server model is developed on the basis of a presently shipping enterprise server. All temperature sensors in the server use an 8-bit ADC. According to our measurements, the time lag on the temperature measurement in this system amounts to 10 sec (cf. Fig 1).

The role of the proposed controller is to jointly determine the optimal fan speed, i.e., s^*_{fan}, and maximum allowable CPU utilization (so called CPU cap), i.e., $\hat{u}^*_{cpu}$, so as to maintain the operating temperature of CPU within a safe operating region, e.g., $<80^\circ C$, while minimizing performance degradation which happens when the required performance level is higher than the CPU cap. The controller largely consists of two parts: 1) multiple local and 2) a global controllers. In the target architecture, we have two local controllers, namely fan speed (s_{fan}) and CPU cap ($\hat{u}_{cpu}$) controllers. In this work, we focus on developing a stable fan controller robust to the non-ideal

temperature measurement while simply using low-complexity CPU capper using a deadzone-like scheme. In a deadzone-like CPU capper, there are two threshold values, i.e., T_{th}^{low} and T_{th}^{high}. The CPU cap, i.e., $\hat{u}_{cpu}$, is only increased when the measured temperature, T_{meas}, is higher than T_{th}^{high} while lowering $\hat{u}_{cpu}$ when T_{meas} is lower than T_{th}^{low}. Regarding the fan speed controller, we adopt a PID control scheme to make the junction temperature track a reference trajectory, i.e., $T_{fan}^{ref}(t)$.[2] Due to the different time constants between CPU die and heat sink (cf. Section III-B), we adjust $\hat{u}_{cpu}$ more frequently than s_{fan}. The independent local decisions are fed into a global controller which gives an optimal control action by coordinating the local decisions. We explain the designs of the local fan speed and the global controllers in Sections IV and V, respectively.

B. Power and temperature modeling

The server's total power consumption (P_{tot}) can be modeled as the sum of CPUs (P_{cpu}) and fans (P_{fan}) power consumptions, i.e., $P_{tot} = P_{cpu} + P_{fan}$. P_{cpu} is proportional to CPU utilization ($u_{cpu} \in [0, 1]$) and modeled as follows [16], [17]:

$$P_{cpu} = P_{cpu}^{static} + P_{cpu}^{dyn} \cdot u_{cpu} \tag{1}$$

where P_{cpu}^{static} and P_{cpu}^{dyn} are the static and the maximum dynamic power consumption of CPU. P_{fan} has a cubic relationship with the fan speed, i.e., $P_{fan} \propto s_{fan}^3$.

The temperature of the target system can be modeled using a well known duality between thermal and electrical phenomena [18]. Hence, we use the temperature model presented in [6]. Using the model, the temperature of the heat sink at the time $(t + \Delta t)$, i.e., $T_{hs}(t + \Delta t)$, is calculated as follows:

$$T_{hs}(t + \Delta t) = T_{hs}^{ss} + \left(T_{hs}(t) - T_{hs}^{ss}\right) \cdot e^{-\frac{\Delta t}{R_{hs} C_{hs}}} \tag{2}$$

where R_{hs} and C_{hs} represent the heat sink thermal resistance and capacitance. R_{hs} is inversely proportional to the fan speed. T_{hs}^{ss} is the steady-state T_{hs} which is calculated as follows:

$$T_{hs}^{ss} = T_{amb} + R_{hs} \cdot P_{cpu} \tag{3}$$

where T_{amb} is the ambient temperature. The thermal time constant of the heat sink is much larger than that of the CPU die, even at the highest fan speed (i.e., smallest heat sink time constant). Thus, we can calculate the junction temperature of the CPU die, T_j by solving the differential equation for the thermal RC network assuming that T_{hs} is constant. Table I summarizes the parameters and corresponding values for modeling the power and the temperature in the target system.

TABLE I
DESIGN PARAMETERS USED IN POWER AND TEMPERATURE MODELING

CPU	P_{max}	160W
	P_{idle}	96W
	Die thermal time constant	0.1 sec
Fan	Fan power per socket	29.4W
	Max fan speed per socket	8500rpm
	Fan sample interval	1 sec
	Heat sink thermal resistance in K/W	$R_{hs} = 0.141 + \frac{132.51}{V^{0.923}}$ V: fan speed (rpm)
	Heat sink thermal time constant at max air flow	60 sec

IV. Robust fan speed controller design

In this section, we present a robust fan speed controller design based on a PID control scheme which assures control stability even under the non-ideal effects associated with the temperature measurement subsystem. We first explain the basics of a PID control scheme and an efficient way of the parameter tuning. Then, we enhance the conventional PID-based fan speed controller with two solutions which make the controller resilient to the time lag and the quantization error while reducing server performance degradation.

A. Basics of PID-based fan speed controller

We periodically adjust the fan speed at every fan speed control step $\Delta t_{control}^{fan}$. In particular, the (k+1)-th decision period, the fan speed controller based on the proposed PID control scheme adjusts the fan speed, i.e., $s_{fan}(k+1)$, as follows:

$$\begin{aligned} s_{fan}(k+1) &= s_{fan}^{ref} + K_P \cdot \Delta T_{cpu}(k) \\ &\quad + K_I \cdot \sum_{i=1}^{k} \Delta T_{cpu}(i) \\ &\quad + K_D \cdot (\Delta T_{cpu}(k) - \Delta T_{cpu}(k-1)) \end{aligned} \tag{4}$$

where s_{fan}^{ref} is the offset of the fan speed. K_P, K_I, and K_D are the coefficients of proportional, integral and derivative gains, respectively. $\Delta T_{cpu}(k)$ is the temperature error which is the difference between the measured value at the k-th time period, i.e., $T_{meas}(k)$, and the reference temperature T_{fan}^{ref}.

The first term is used for linearization as the PID control theory works well only for linear plant models [9]. The second term is a proportional portion which reacts to ΔT_{cpu} to quickly reach the required value. The third term is an integral portion which is required to eliminate the steady-state error. The last term is a derivative portion which adjusts s_{fan} to reduce the overshoots. The three parameters need to be carefully decided by jointly considering stability, accuracy, settling time, and overshoot (SASO) [9]. Among various tuning solutions, we used *Ziegler-Nichols closed-loop tuning* method [21], which tunes the parameters by measuring two parameters, namely *ultimate gain value* (K_u) and the ultimate period of oscillation (P_u). K_u is accomplished by finding the value of the proportional-only gain that causes the control loop to oscillate indefinitely at steady state. The ultimate period, P_u, is the time required to complete one full oscillation while the system is at steady state. Thus, we determine the PID parameters as follows [21]:

$$K_P = 0.6 \cdot K_u \tag{5}$$

$$K_I = K_P \cdot (2/P_u) \tag{6}$$

$$K_D = K_P \cdot (P_u/8). \tag{7}$$

[2] We also used a single threshold and a deadzone fan controllers. However, they are not stable due to the non-ideal effects in the temperature measurement systems. Thus, we simply used PID controller in this work.

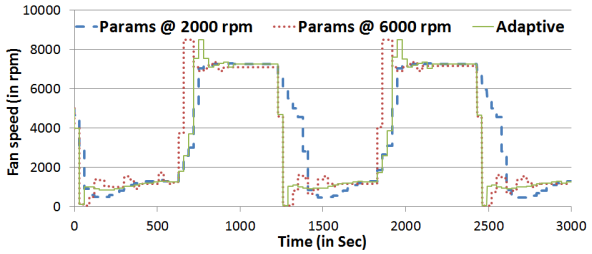

Fig. 3. Comparisons of fan speed traces obtained by applying an adaptive and a conventional PID control scheme with a set of PID parameters obtained when the fan speed is 2000 rpm and 6000 rpm, respectively.

B. Adaptive PID control scheme

A set of PID parameters obtained through the tuning solution is only optimal for a linear target system. However, the target system (temperature and the fan speed relationship) is nonlinear as the thermal resistance also varies nonlinearly with respect to fan speed as shown in Table I. Thus, a set of PID parameters obtained in one fan speed region may not work well in other fan speed regions. Fig. 3 shows traces of the fan speed and temperature when CPU load periodically alternates between 0.1 and 0.7 using PID parameters obtained from fan speeds of 2000 and 6000 rpm. When using a set of parameters at 2000 rpm, the system is stable while the convergence time is very slow, i.e., 210 sec. It is due to the low K_P value since the temperature is more sensitive to the fan speed in this region. On the other hand, the fan speed control with the parameters obtained from 6000 rpm shows that the convergence time becomes faster, while it becomes unstable, especially at the lower fan speed range as the set of PID parameters obtained at 6000rpm is located outside the range of ensuring the stability (i.e., it was obtained based on the less sensitive temperature/fan speed relationship).

In order to solve the problem of using a single set of PID parameters in a target server system, we introduce a new adaptive PID control scheme which dynamically adjusts the set of the parameters according to the operating fan speed. First, we obtain the sets of parameters in multiple fan speed regions. Note that the number of regions depends on the error of the piecewise linearization. In our work, two regions, i.e., 2000 and 6000 rpm, are enough to linearize the relationship within 5% error for the considered enterprise server systems.

Then, at runtime, a set of PID parameters is adjusted according to the measurement time latency and the operating fan speed at every fan speed decision period. It first finds the adjacent two regions which satisfies $s_{fan}^{ref(i)} \leq s_{fan}^{ref} \leq s_{fan}^{ref(i+1)}$ where $s_{fan}^{ref(i)}$ is the *reference fan speed* of the i-th operating region. Note that s_{fan}^{ref} is arranged in increasing order, i.e., $s_{fan}^{ref(i)} < s_{fan}^{ref(i+1)}$. Next, the parameters are determined as the weighted sum of the parameters in the two consecutive regions with a weight which is proportional to the distance between s_{fan} and $\{s_{fan}^{ref(i)}, s_{fan}^{ref(i+1)}\}$ as follows:

$$K_{P/I/D}(k) = \Big(1 - \alpha(k)\Big) \cdot K_{P/I/D}^{(i)} + \alpha(k) \cdot K_{P/I/D}^{(i+1)} \quad (8)$$

$$\alpha(k) = \frac{s_{fan}(k) - s_{fan}^{ref(i)}}{s_{fan}^{ref(i+1)} - s_{fan}^{ref(i)}}. \quad (9)$$

When the operating region is changed, s_{fan}^{ref} in Eqn. (4) is updated and $\sum_{i=1}^{k} \Delta T_{cpu}(i)$ is set to zero. The solid line in Fig. 3 shows the traces of temperature and fan speed when we use an adaptive PID control scheme. As shown in the traces, the target systems becomes stable while the convergence time is also drastically improved compared to the case of using PID parameters at 2000 rpm.

C. Quantization error elimination scheme

To eliminate the oscillation caused by the quantization of the temperature measurement, we propose a quantization elimination scheme that enforces no change in fan speed (i.e., s_{fan}). In particular when the temperature measurement error is less than the size of the quantization step, i.e., $|T_Q|$, the quantization is adapted as follows:

$$S_{fan}(k+1) = s_{fan}(k) \text{ when } |T_{ref}^{fan} - T_{meas}(k)| < |T_Q|. \quad (10)$$

V. Global Controller Design

In this section, we present a new global control scheme that coordinates multiple local control actions to guarantee server system operation stability while jointly minimizing the performance degradation and the energy consumption.

A. Rule-based global coordination approach

In order to guarantee the system stability when multiple local controllers are running together in a system, we propose a global control scheme that dynamically selects only one control action at a time affecting the system because the stability of each controller has been proven at the local controller design. The suitable selection among multiple local control actions varies according to operating conditions. To deal with the issue, we propose a rule-based coordination scheme as presented in Table II, which adjusts the control variables, i.e., $\{\hat{u}_{cpu}, s_{fan}\}$, by considering performance as the primary system behavior concern. Note that the rule can be changed according to the metric that we need to satisfy. However, in enterprise servers equipped with a variable fan speed controller, performance degradation is the most critical concern as the power consumption has already been reduced significantly in recent years [6] while the performance can be significantly degraded if the fan speed is not properly set. Thus, we use the rule biased for improving the performance throughout this paper. As shown in Table II, we can largely classify the operating condition into 9 cases according to the relative values between the current control variables and the next local control actions. First, when only one variable is changed in local controllers, we just change the parameter. Second, when s_{fan} is determined to set higher, we adjust s_{fan} while leaving $\hat{u}_{cpu}$ unchanged because the adjustment of the fan speed happens infrequently, as explained in Section III, which leads to greater performance degradation until the next fan speed decision period once the fan speed sets too low. Finally, when s_{fan} controller decides to set lower fan speed, it increases $\hat{u}_{cpu}$ when it requires a higher value, otherwise, we lower the fan speed.

TABLE II
A RULE-BASED COORDINATION

		Fan speed		
		$s_{fan}(k+1) < s_{fan}(k)$	$s_{fan}(k+1) = s_{fan}(k)$	$s_{fan}(k+1) > s_{fan}(k)$
CPU cap	$u_{cpu}(k+1) < u_{cpu}(k)$	$s_{fan} \downarrow$	$u_{cpu} \downarrow$	$s_{fan} \uparrow$
	$u_{cpu}(k+1) = u_{cpu}(k)$	$s_{fan} \downarrow$	-	$s_{fan} \uparrow$
	$u_{cpu}(k+1) > u_{cpu}(k)$	$u_{cpu} \uparrow$	$u_{cpu} \uparrow$	$s_{fan} \uparrow$

B. Predictive adjustment of the set-point temperature of fan controllers

The performance degradation can be reduced by lowering the reference temperature of a fan controller (T_{ref}^{fan}) because it makes the CPU junction temperature lowered by setting the fan speed higher while it increases the power consumed by fans (cf. Section III-B). Thus, we need a solution to judiciously adjust T_{ref}^{fan} to jointly reduce both performance degradation and power consumption. Our studies outline two observations:

- When CPU utilization is low, attenuate T_{ref}^{fan} to cope with any unexpected abrupt increase of the CPU utilization.
- When CPU utilization is high, amplify T_{ref}^{fan} to lower the temperature increase caused by the unexpected increase on CPU utilization.

Based on the observations above, we linearly scale T_{ref}^{fan} according to the predicted CPU utilization. Furthermore, in order to filter out the noise term in the CPU utilization, we used a moving average filter for the prediction [19].

C. Single-step fan speed scaling

To further reduce the performance degradation, especially, caused by abrupt spikes on required CPU utilization, we present a single-step fan speed scaling solution which sets the fan speed to the maximum when the measured performance degradation is higher than a predefined threshold value. As analyzed in [20], the performance spike in server workloads is much faster than the settling time of controllers. In adjusting the fan speed, it takes $N_{trans}^{fan} \cdot t_{interval}^{fan}$ where N_{trans}^{fan} is the number of decision periods until the fan speed reaches to a steady-state. Thus, the performance of the servers can be severely degraded during the long transient time period. The single-step fan speed scaling scheme can guarantee that the performance degradation during the settling time is no larger than the predefined value. Once the maximum fan speed is set, we lower the fan speed to reach the lowest possible fan speed which enables to run required CPU utilization without any temperature violation.

VI. EXPERIMENTAL RESULTS

A. Setup

We developed our simulation environment to model the system characteristics of actual commercial enterprise server. Table I summarizes the parameters used in this simulation. We used synthetic workload traces which alternates between 0.1 and 0.7 while imposing a random Gaussian noise to further validate the robustness of the propose control scheme in realistic CPU variation characteristics. Considering the normal control interval in commercial enterprise servers, we set the CPU and fan control time constants ($\Delta t_{control}^{cpu}$ and $\Delta t_{control}^{fan}$) to 1 sec and 30 sec, respectively.

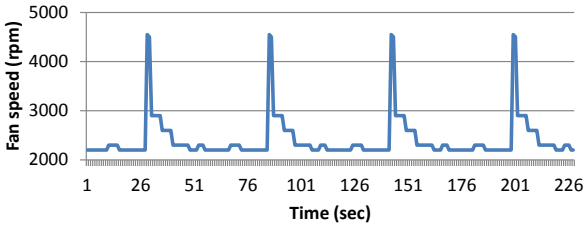

Fig. 4. Measured of fan speed and temperature under a stable workload

First, we demonstrate the stability of the proposed control scheme under the dynamic workload scenario (in Section VI-B), and then, compare the performance (in Section VI-C) and the energy consumption with the following solutions:

- *w/o coordination*: baseline which uses the fan speed and CPU load controllers without any coordination
- *E-coord*: energy-aware coordination scheme in [6]
- *R-coord(@ $T_{ref}^{fan} = 75°C$)*: rule-based coordination while using a fixed reference temperature with $75°C$, for the fan speed controller (in Section V-A)
- *R-coord+A-T_{ref}^{fan}*: rule-based coordination with adaptively changing the T_{ref}^{fan} from 70 to $80°C$ according to the CPU utilization (in Section V-B)
- *R-coord+A-T_{ref}+SS^{fan}*: additively applying the single-step fan speed control scheme (in Section V-C)

For fair comparison, we use the proposed fan speed control scheme in all solutions. Note that the system becomes unstable if we directly adopt the fan speed control scheme presented in [6] as it is designed without the long I^2C delay concern.

B. Stability analysis

Fig. 4 shows the measured fan speed in the target server adopting a deadzone fan speed control scheme under a fixed workload. It demonstrates that the fan speed becomes oscillatory due to the effects caused by the non-ideal temperature measurement. In Fig. 3, we have already shown that the variable fan speed control in the server becomes stable as we use the proposed adaptive PID-based fan control scheme.

To further validate the stability of the proposed global coordination scheme, we performed a simulation while running the proposed fan speed control scheme along with the CPU load controller (in Section III under time-varying CPU utilization. Fig. 5 shows the varying CPU utilization (solid line and left Y-axis) and the fan speed (dotted line and right Y-axis). As shown in the figure, even with time-varying CPU utilization, the proposed control solution provides stable fan speed control.

C. Performance benefits

The second column of Table III shows the performance comparisons in terms of the fraction of the deadline violations caused by the thermal emergency. As shown in the table, *E-coord* can cause a huge increase in the deadline violation

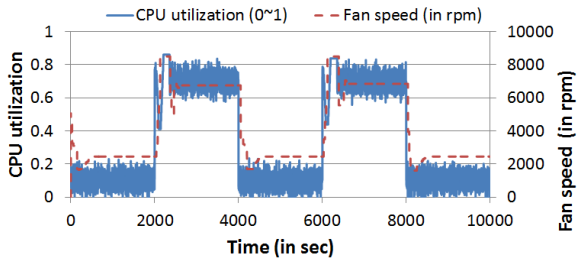

Fig. 5. Traces of fan speed with the dynamic CPU load and noise (standard deviation is set to 0.04).

TABLE III
COMPARISONS OF THE PERFORMANCE AND THE POWER CONSUMPTION

Solution	Deadline violation (%)	Norm. Fan energy consumption
w/o coordination (***baseline***)	26.12	1
E-coord [6]	44.44	0.703
R-coord(@ $T_{ref}^{fan} = 75°C$)	14.14	1.075
R-coord+*A*-T_{ref}^{fan}	11.42	0.801
R-coord+*A*-T_{ref}^{fan}+SS^{fan}	6.92	0.804

as it does not take into account the impact of the performance degradation when it decides the control action when a thermal emergency happens. On the contrary, *R-coord*(@ $T_{ref}^{fan} = 75°C$) can reduce the percentage of the deadline violation by up to 12% compared to the baseline scheme as the rule-based coordination scheme is designed to minimizing the performance degradation by increasing the fan speed first when multiple control actions are conflicted when the thermal emergency happens. Further improvement can be obtained by using the predictive adjustment of T_{ref}^{fan} because the scheme lowers T_{ref}^{fan} when a predicted CPU load is low, which enables to reduced the performance degradation caused by unexpected CPU load spike. The single-step fan speed scaling provides additional 4.5% reduction in the terms of the performance degradation by further reducing the performance degradation until the fan speed reaches its desired point by setting the fan speed is set to its maximum when the measured performance degradation is higher than a certain threshold value.

D. Energy consumption evaluation

The third column of Table III shows the the energy consumed by fans when adopting four different solutions. The values are normalized with respect to the uncoordinated case. As we use the rule-based coordination scheme with a fixed T_{ref}^{fan}, the energy consumption is slightly increased compared to the baseline case, as we set T_{ref}^{fan} to low value in order to prevent the performance degradation caused by unexpected spike in the CPU load. The energy consumption can be reduced as we adaptively adjust T_{ref}^{fan} according to the predicted CPU load by up to 20% as we increase T_{ref}^{fan} when the predicted CPU load is high. The single-step fan speed scaling scheme leads to the slight increase in the energy consumption as it set the fan speed to its maximum when the performance degradation is higher than a threshold value. *E-coord* can yield even 10% lower energy consumption compared to the proposed solution as it is originally designed to minimize the energy consumption. However, the performance degradation is unacceptably high.

VII. CONCLUSION

In this paper we have presented a stable control scheme for enterprise servers that dynamically adjusts the fan speed while jointly minimizing the performance degradation and the power consumption. We have first presented a stable PID fan speed controller that is robust to non-ideal temperature effects, i.e., measurement time lag and signal quantization. Then, we have presented a global control scheme which coordinates multiple local control actions via a low-complexity rule-based management scheme while minimizing the performance degradation caused by the variable fan speed control. We have validated the proposed control scheme by developing simulation environment modeling presently shipping commercial enterprise servers. The experimental results show that the proposed fan speed control scheme is robust to the long measurement time lag and the temperature quantization while reducing the performance degradation and the power consumption by up to 19.2% and 19.06%, respectively, compared to a conventional fan speed control solution.

REFERENCES

[1] L. A. Barroso and U. Holzle, "The datacenter as a computer: an introduction to the design of warehouse-scale machines," in *Synthesis Lectures on Computer Architecture* 4.1, 2009.
[2] D. Shin, *et al.*, "Energy-optimal dynamic thermal management for green computing," in *Proc. ICCAD*, 2009.
[3] Z. Haihong, *et al.*, "Remote Management with the Baseboard Management Controller in Eighth-Generation Dell PowerEdge Servers," in *Magazine of Dell Power Solutions*, 2004.
[4] Semiconductors, Philips, "The I2C-bus specification," in *Philips Semiconductors*, 2000.
[5] Z. Wang, *et al.*, "Optimal fan speed control for thermal management of servers," in *Proc. IPAC*, 2009.
[6] R. Ayoub, *et al.*, "JETC: joint energy thermal and cooling management for memory and CPU subsystems in servers," in Proc. *HPCA* 2011.
[7] C. S. Chan, *et al.*, "Fan-speed-aware scheduling of data intensive jobs," in *Proc ISLPED*, 2012.
[8] M. Zapater, *et al.*, "Leakage and temperature aware server control for improving energy efficiency in data centers," in *Proc. DATE*, 2013.
[9] J. L. Hellerstein, *et al.*, "Feedback control of computing systems," in *Wiley. com*, 2004.
[10] Y. Okuyama, "Robust stabilization and PID control for nonlinear discretized systems on a grid pattern," in *Proc. ACC*, 2008.
[11] J. Kim, *et al.*, "Program phase-aware dynamic voltage scaling under variable computational workload and memory stall environment," in *IEEE TCAD*, 2011.
[12] D. Meisner, *et al.* "Power management of online data-intensive services," in *Proc. ISCA*, 2011.
[13] J. Kim, *et al.*, "Correlation-aware virtual machine allocation for energy-efficient datacenters," in *Proc. DATE*, 2013.
[14] R, Ayoub, *et al.*, "Temperature-aware dynamic workload scheduling in multisocket cpu servers," in *IEEE TCAD*, 2011.
[15] X. Wangi, and Y. Wang, "Co-con: Coordinated control of power and application performance for virtualized server clusters," in *Proc. IWQoS*, 2009.
[16] D. Economou, *et al.*, "Full-system power analysis and modeling for server environments," in *Proc. WMBS*,2006.
[17] M. Pedram and I. Hwang, "Power and performance modeling in a virtualized server system," in *Proc. ICPPW*,2010.
[18] W. Huang, *et al.*, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," in *IEEE TVLSI*, 2006.
[19] A. K. Coskun, *et al.*, "Utilizing predictors for efficient thermal management in multiprocessor SoCs," in *IEEE TCAD*, 2009.
[20] A. Bhattacharya, *et al.*, "The Need for Speed and Stability in Data Center Power Capping," in *Proc. IGCC*, 2012.
[21] D. Valerio, *et al.*, "Tuning of fractional PID controllers with ZieglerNichols-type rules," in *Signal Processing*, 2006.