Thermal Effects on Real-Time Systems

Youngwoo Ahn
Department of Electrical & Computer Engineering
Texas A&M University
College Station, TX 77840
ayw@ece.tamu.edu

Riccardo Bettati
Department of Computer Science
Texas A&M University
College Station, TX 77843
bettati@cs.tamu.edu

***Abstract*— In our research, we study how real-time systems are affected by thermal management to satisfy the temperature constraint. In temperature-constrained real-time systems, deadline guarantees must be met without exceeding safe temperature level of the processor. While processor speed control is the most popular method of thermal management of systems, it eventually makes the task delays longer. In our study, we describe how to find the worst case execution considering speed control in temperature-constrained environment. With the worst case execution scenario, we study how the simple reactive speed scaling scheme can improve the processor utilization compared with any constant-speed scheme. For aperiodic tasks, it is briefly reviewed how the naive application of slack stealing leads to missed deadlines and the design-time slack allocation is proposed. A queueing model is presented to analyze the response time provided to aperiodic jobs and validated with results from a discrete-event simulator.**

I. INTRODUCTION

In recent years, power density in processors has increased exponentially. Because of the high power density, modern processors are suffering from the stress of high temperature caused by the high levels of energy consumption. Since the reliability of a circuit is closely related to the operating temperature and high temperature might result in the processor's failure and the system thereby, temperature is becoming a big concern in system design.

There have been many approaches to manage the thermal problem. Thermal management through packaging (like advanced circuit design or improved air flows), and active cooling (such as FAN control) have been widely adopted for most computing systems in personal desktop and server environments. However, these approaches are inappropriate for many high-performance embedded systems, such as emerging systems-in-package and many other devices where the requirements for packaging and the operating environment make the deployment of these traditional approaches for thermal management difficult. In such cases, system temperature must be controlled through control of power input. The power consumption by a processor is the main source of the increase of temperature in a system and a number of *dynamic* thermal management approaches for processors have been proposed, including Clock Throttling or Clock Gating, Dynamic Voltage Scaling (DVS), and architecture-level thermal control mechanism such as local gating of sub components inside the processor. We categorize all these schemes as *Dynamic Speed Scaling* schemes.

There is a lot of literature on dynamic speed scaling for real-time systems. Most work, however, focuses on using dynamic speed scaling for lowering energy consumption of the system rather than maintaining safe temperature levels. That is, energy-aware systems focus on how to keep the *average* power consumption level low while temperature-aware systems focus on lowering *peak* power consumption. In the latter case, dynamic speed scaling is used to obtain the maximum performance while at the same time guaranteeing timing and temperature constraints.

II. SPEED SCALING

In our research, we focus on the dynamic speed scaling for a study of temperature-constrained real-time systems. To meet the delay constraints for real-time tasks, we run the processor at a higher speed. To meet the temperature constraints, we run the processor at a lower speed. In our study, we apply a very simple dynamic speed scaling scheme, which we call Reactive Speed Scaling (RSS): Whenever the CPU is busy, it is allowed to run at high speed S_H until it reaches a maximum critical temperature T_c at a safe margin from the junction temperature. Once T_c is reached, the CPU continues at a reduced *equilibrium speed* S_E, which keeps the temperature at or below T_c.

For thermal analysis, we apply the well-known model used in [1], [2]: the rate of heating or cooling is proportional to the difference in temperature between the object and the environment. There are assumptions in our analysis that environmental temperature is fixed and scaled to zero. Another arguable assumption is that the CPU's temperature is changed only by its speed and execution.

III. TASK MODEL

A. Periodic Tasks

In our research, we start with the analysis of periodic tasks first and proceed to aperiodic and sporadic tasks. Fig. 1 shows the case for periodic task sets. Even for a single periodic task shown in Fig. 1, increased temperature makes the response of the task get longer and the guarantee of deadline constraint unpredictable.

A *critical instance* of a task is a time instance which is such that the job in the task released at the instance has the maximum response time of all jobs in the task. In a fully preemptable system with RSS, there are two factors

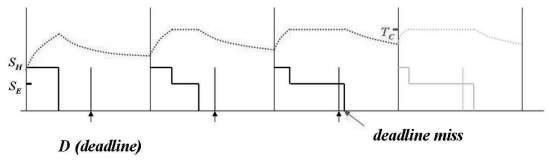

Fig. 1. Deadline misses of a periodic task under thermal constraint

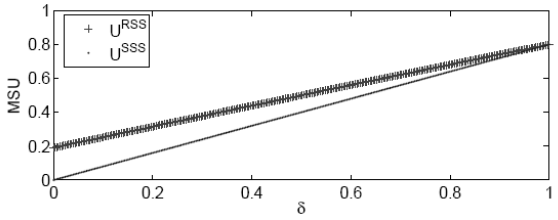

Fig. 2. Evaluation of Maximum Schedulable Utilization

that affect the critical instance. Besides the preemptions by higher-priority jobs, there is an effect by the temperature at the instance. With any speed scaling, once the temperature hits the threshold, the speed will drop to no higher than the equilibrium speed. Thus, the response time of a job is affected by the temperature at its arrival. For the schedulability analysis, the worst-case preemption from the higher-priority tasks and the maximal temperature at the time instance should be obtained. Define the critical instance as *thermal* critical instance. By looking into the *thermal* critical instance, we compute a Maximum Schedulable Utilization (MSU)and compare it with that of Constant Speed Scaling (CSS). Fig. 2 shows the comparison of MSU according to various ratios of deadline over the length of period, δ.

B. Aperiodic Tasks

In Fig. 3, we see the execution of the mixed task set of hard real-time periodic and soft real-time aperiodic tasks. Simply following the conventional way of handling aperiodic jobs, like the slack stealing server, may not guarantee the deadline constraints of hard periodic tasks because of increase caused in CPU temperature. While securing slack just in *time* is enough for traditional slack stealing approach for real-time systems only with delay constraints, slack in *clock-cycles* is required not to miss deadlines of systems with RSS. That is, the prediction of thermal variation and the computation of the available clock cycles under the thermal profile is necessary. In our study, we also suppose the worst case execution condition for running aperiodic jobs and try to get a proper model to analyze the performance of aperiodic server in thermally constrained environment. The comparison of response times by the analytical model and the simulation is shown in Fig. 4. For the analytical model and its analysis, we modify M/G/1 queueing model to compute the average response time of aperiodic jobs. As it can be seen from the Fig. 4, the computed average response times are acceptable with very small errors when they are compared with the results from simulations on discrete-event simulator.

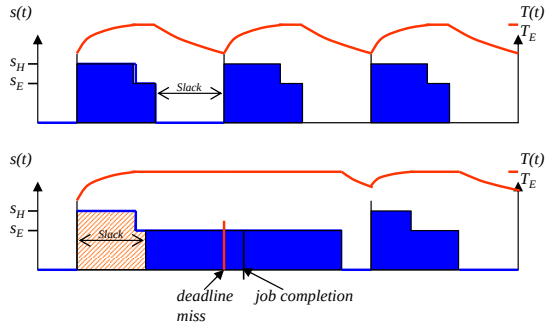

Fig. 3. Traditional Slack Stealer causes deadline misses in temperature constrained environment.

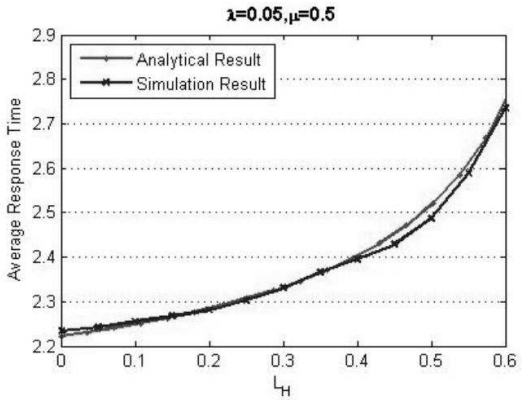

Fig. 4. Average response time of aperiodic jobs based on Simulation and Queueing Analysis - with the variation in length of CPU's high speed running. P=10.0, s_H=1.0, s_E=0.8.

IV. CONCLUSION

Especially for real-time systems, the performance analysis is really important to make the system to guarantee various constraints including delays in time, system resources, and some physical limits like the temperature. Due to high power density of modern microprocessors and difficulty in power dissipation in dense packaging of systems-in-package, thermal issues are becoming more critical. In our research, we describe a model for the analysis of real-time systems with speed scaling. To meet the temperature constraints, we applied a simple reactive speed-scaling scheme, which could easily be implemented using the thermal management facilities on modern microprocessors. Even though the speed scaling scheme is very simple, the schedulability analysis for real-time systems and the examination of the performance of an aperiodic server can be really complex due to the non-linear characteristic of the temperature variation. Here in our study, we suggest a new concept, *thermal* critical instance, to find the maximum schedulable utilization of real-time systems with thermal constraints. To predict the average response time of aperiodic jobs in real-time system with thermal constraint before simulations, we also did some analytical modelings. They showed very trustful analytical results for the worst execution cases.

REFERENCES

[1] N. Bansal and K. Pruhs, "Speed scaling to manage temperature," in *Symposium on Theoretical Aspects of Computer Science*, 2005.
[2] J. E. Sergent and A. Krum, *Thermal Management Handbook*, McGraw-Hill, 1998.