System-Level Power and Timing Variability Characterization to Compute Thermal Guarantees

Pratyush Kumar and Lothar Thiele
Computer Engineering and Networks Laboratory
ETH Zurich, Switzerland
{kumarpr, thiele}@tik.ee.ethz.ch

ABSTRACT

With ever-increasing power densities, temperature management using software and hardware techniques has become a necessity in the design of modern electronic systems. Such techniques have to be validated and optimized with respect to the thermal guarantee they provide, i.e., a safe upper-bound on the peak temperature of the system under all operating conditions. The computation of such a guarantee depends on the power and timing characteristics of the system. In this paper, we present formalisms to capture such characteristics at the system-level and provide an analytical technique to compute a provably safe upper-bound on the peak temperature. The proposed characterization and analysis is general in that it considers an impulse-response-based thermal model, task-dependent power consumption, tasks with dynamic arrival patterns and variable resource demand, and a scheduling policy expressed as a hierarchical composition of several commonly used policies.

Categories and Subject Descriptors

C.4 [**Computer Systems Organization**]: Performance of systems—*Modeling techniques, Performance attributes*

General Terms

Design, Performance, Theory

Keywords

Power Variability Curve, Thermal Guarantee

1. INTRODUCTION

The power density of modern electronic systems continues to rise, leading to high on-chip temperatures. High temperatures are detrimental to the quality of the system: they can lead to transient reduction in system performance, unreliable timing discrepancies, or even long-term reliability reduction [1]. As a result, power dissipation and the consequent thermal implications are likely to be a significant, if not dominant, part of the design-effort and cost of future systems [2].

At design time, thermal management is performed through the use of appropriate packaging, heat sinks and additional active cooling solutions [3]. Due to the exponential rise in the cost of such physical-level solutions, they are often complemented during run-time by methods collectively referred to as Dynamic Thermal Management techniques [4]. These techniques include Dynamic Voltage Scaling (DVS) and various forms of system throttling, such as clock gating, cache throttling and so on.

The design of system-level thermal management is intricate: several different solutions, cutting across hardware and software, and design-time and run-time, must be evaluated for cost and effectiveness. Any such thermal management solution is only as good as the guaranteed upper-bound on the temperature of the system. If this bound meets the design guideline, the system can be deemed temperature-safe. Otherwise, a design revision must be performed.

Intuitively, the crucial step in the above iterative design process is the computation of the upper-bound on the peak temperature of the system. As we shall see later formally, temperature of the system is a state variable that evolves over time. Computation of the peak temperature requires us to analyze *traces* of power consumption. A first approach to compute the said bound is to collect a set of power traces of the system and to then simulate the thermal behavior of the system for these traces. This can be performed either in a software simulators, or more efficiently with hardware emulation [5]. The peak temperature noticed across all such simulations may be deemed the upper-bound on the peak temperature. Intuitively, the flaw with this approach is that the chosen set of power traces may not be fully representative of all possible power traces that the system may exhibit. Consequently, the obtained peak temperature may not be a true upper-bound. This is especially significant in today's multi-tasking systems where several tasks consuming different amounts of power are scheduled. In such systems timing non-determinism, such as variability in execution time, is generally observed. The number of possible power traces of such a system can potentially be very large, not all of which can be expected to be covered by a randomly chosen small set of sample traces.

On the other hand, it is also not desirable that the upper-bound on the temperature estimated using some approximate method is much larger than the actual peak temperature. An overly conservative bound, would unnecessarily incur a large cost on the cooling solution and/or sacrifice

CODES+ISSS'11, October 9–14, 2011, Taipei, Taiwan.

performance due to the use of aggressive DTM techniques. This presents the case for a more formal approach to characterize the power and timing properties of the system and a robust analysis technique to compute the thermal guarantee.

In this paper, we propose a formalism on a set of power traces called the Power Variability Curve (PVC). The PVC characterizes the maximum amount of the energy (integral of power) consumed by the system in any time interval of a given length, in any of the power traces considered. We show that a PVC, so computed, compactly represents the possible variability on a large set of power traces with respect to computing the required thermal guarantee. Using the PVC, we identify a power trace, the peak temperature of which provably upper-bounds the peak temperature amongst all power traces characterized by a PVC. We show that this property of the PVC holds true, independent of the thermal parameters, for the considered system model.

As the second step, we present an analytical technique to compute the Power Variability Curve (PVC). As inputs we use the power consumption of individual tasks and the timing properties of the system. Characterization and analysis of timing properties has been the core focus of the real-time systems community. We use models from one such stream of research, Real-Time Calculus (RTC) and Modular Performance Analysis (MPA) to represent the timing properties of the individual tasks and model the scheduling amongst tasks. We show how the use of RTC-MPA framework is a natural choice in the computation of the PVC.

The rest of the paper is organized as follows. We present examples to motivate the presented work in Section 2. We present the system model in Section 3 and an overview of our approach in Section 4. In Section 5, we formally define Power Variability Curve (PVC) and use it in Section 6 to compute the peak temperature of the system. In Section 7, we show how PVC can be computed using power consumption of the individual tasks and the timing properties as derived using Real-Time Calculus (RTC) and Modular Performance Analysis (MPA). We present experimental results on a case study in Section 8 that demonstrates the theoretical results. We compare our approach against existing work in Section 9, and conclude in Section 10.

2. MOTIVATING EXAMPLES

In this section, we consider two examples of systems executing two tasks each. The aim of these examples is to illustrate the difficultly in computing the thermal guarantee, even for simple systems.

For both examples, we consider a Fourier conductive heat transfer model given by the following differential equation,

$$C\frac{d\Theta}{dt} + G(\Theta - \Theta_{amb}) = P(t), \tag{1}$$

where C and G are thermal parameters, $P(t)$ is the power consumption of the system at time t, and $\Theta(t)$ is the temperature of the system at time t. The thermal parameters we use are the shown in Table 3(a).

2.1 Example 1

Consider a system executing two tasks T_A and T_B in round-robin fashion, i.e., jobs (or single instances) of tasks T_A and T_B execute alternately. Each task is characterized by a power consumption, a best-case execution time (BCET), and a worst-case execution time (WCET). These properties of the tasks are shown in Table 1.

Task	Power (W)	BCET (ms)	WCET (ms)
T_A	12	20	100
T_B	7	30	60

Table 1: Properties of two tasks for Example 1

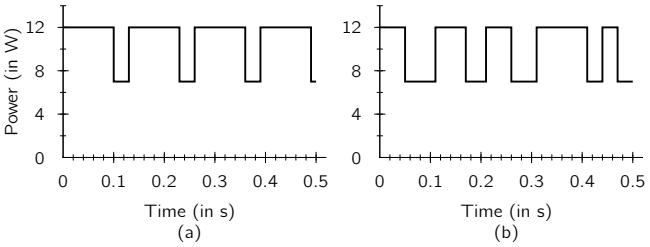


Figure 1: Power traces for Example 1

The execution times of the tasks can vary between their respective BCETs and WCETs. This non-determinism in the execution times implies a corresponding variability in the possible power traces of the system. Two such traces of length 500 ms ($= \tau$, the time horizon of interest) are shown in Figure 1(a) and (b). Naturally, infinitely many such traces can be drawn for the given task model. If any claim is made on a thermal guarantee of a system serving these two tasks, then it must hold true for the infinitely many possible power traces of the system.

For the considered Fourier model of heat dissipation, we can simulate the temperature of the system for either of the two power traces. We plot the corresponding temperature trace in Figure 2. For either case, the peak temperature is different. This illustrates that the execution time variability leads to variation in the peak temperature. A thermal guarantee for such a system must thus consider the entire range of possible variability.

2.2 Example 2

In the above example, the variability in the power traces arises due to the variable execution time of the tasks. A second cause for variability is the non-determinism in the arrival times of the instances of the tasks.

Consider two periodic tasks with jitter, T_X and T_Y, withe power consumption, period, maximum jitter and execution times as shown in Table 2. For simplicity, we consider tasks with constant execution times. The two tasks are scheduled

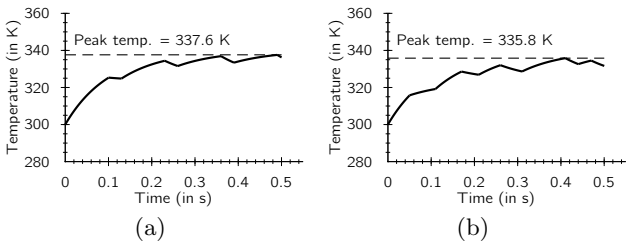


Figure 2: Temperature evolution for power traces shown in (a) Figure 1(a), and (b) Figure 1(b) of Example 1

Task	Power (W)	Period	Jitter	Exec. Time
T_X	20	100	30	30
T_Y	10	120	70	40

Table 2: Properties of two tasks for Example 2. All times are in ms.

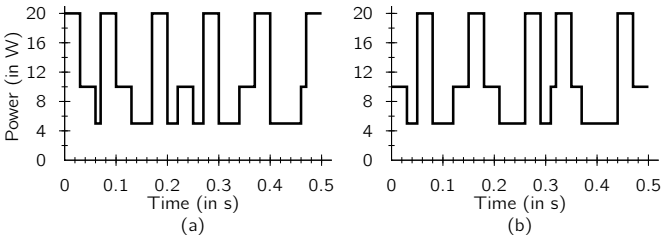


Figure 3: Two power traces for Example 2

using Rate Monotonic fixed priority scheduler, which gives higher priority to the task with shorter period, in this case task T_X. The power consumed when no tasks are executing, i.e., while the processor is idling, is $P_{idle} = 5$W.

The execution times of the tasks are not variable. However, the two periodic tasks can have different offsets and can exhibit variable jitter. Depending on these non-deterministic parameters, a corresponding variability in the power traces arises. Two example power traces of length 500 ms ($= \tau$) are shown in Figure 3. Again, for the considered thermal model, we plot the evolution of temperature with time for both the considered traces in Figure 4. Again, the peak temperature is different for either trace. This illustrates that variability in task arrival patterns also can influence the peak temperature of the system and consequently the thermal guarantee.

2.3 Random Simulations

As was discussed earlier, simulating of random power traces is a first approach to computing the thermal guarantee. For the above examples we numerically evaluate this approach. For either example, we generate 10,000 random traces of the system behavior. The quantities that are subject to variation, such as execution times in Example 1, and arrival times of jobs in Example 2, are uniformly and randomly varied in their respective ranges, throughout the traces. For each of the generated random traces, we simulate the thermal behavior of the system and note the peak temperature. We then identify, the maximum of the peak temperatures noted in the first n random traces, denoted as $\Theta_{max}(n)$. Since, the experiment involves generation of random numbers, we

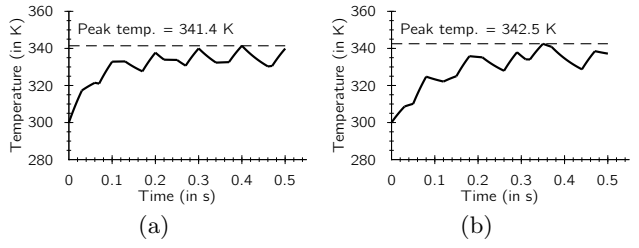


Figure 4: Temperature evolution for power traces shown in (a) Figure 3(a), and (b) Figure 3(b) of Example 2.

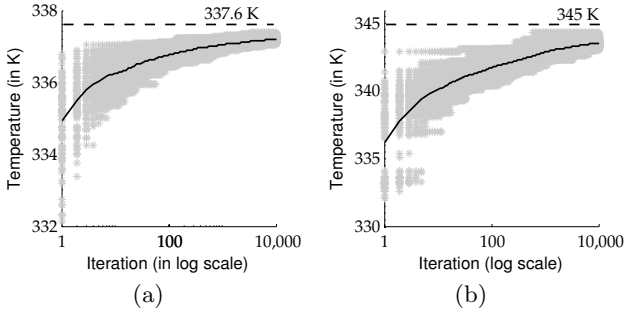


Figure 5: A plot of the peak temperature noted during random simulations for (a) Example 1 and (b) Example 2.

iterate over it 100 times and average the values of $\Theta_{max}(n)$ obtained across all the iterations. We denote the averaged value as $\Theta^{avg}_{max}(n)$. The interpretation of this function is thus: if we pick n random traces and simulate the thermal behavior of the system, then on an average we would obtain a maximum peak temperature of $\Theta^{avg}_{max}(n)$.

For the 100 random experiments with 10,000 iterations in each, we plot Θ_{max} and Θ^{avg}_{max} for either example in Figure 5. The grey dots denote Θ_{max}, whereas, the darker black curve plots the averaged value of Θ^{avg}_{max}. We also plot, using horizontal lines, bounds on the peak temperature which we compute using the analytical approach presented in Section 6. For either example, the function Θ^{avg}_{max} continues to increase, even after 10,000 random simulations. This indicates that even more random traces have to be generated and analyzed until a convincing upper-bound can be computed. Further, the obtained bounds with the proposed analysis, which are described in Section 6, provide a safe upper-bound: the grey dots come close to the horizontal lines, but never cross it.

The considered examples are simplistic in that they contain only two tasks. For systems with several tasks and more general scheduling techniques, such as hierarchical scheduling, the sources of non-determinism are likely to be much larger, and consequently the number of possible traces of system behavior would be exponentially larger. In such systems, it is expected that the number of random traces that have to be analyzed until a convincing upper-bound is identified will be larger, potentially making it practically infeasible. This motivates the need for an analytical approach to characterizing the potential variability and subsequently computing the thermal guarantee.

3. SYSTEM MODEL AND PROBLEM DEFINITION

In this section, we present the formal definition of the models we consider and state the specific problem that we solve in this paper.

3.1 Task and Power Model

We consider a system that serves several independent tasks denoted as $T_1, T_2, \ldots, T_n$. Each task has several instantiations of jobs, and is characterized by power and timing models.

We first discuss the power model. When serving tasks, the system consumes power, and consequently heats up. The power consumed by the system as a function of time is called

the power trace of the system, denoted as $P(t)$. This power can include both the dynamic power (due to switching of transistors) and the leakage power (due to static power dissipation). Typically, the power consumed by the system is task-dependent. For instance, a compute-intensive task may load the system to a large extent and consume a lot of power. For another task, DVS may be used to reduce the voltage (and frequency) to a smaller value, and thereby the system would consume a lower amount of power. To consider such systems, we assume that the power consumption is a function of the currently executing task. Given this, we refer to the power consumed by the system when a (job of a) task executes simply as the power consumption of the task. The power consumption of task T_i is denoted as ρ_i. When not executing any task, the system is said to be idle, and consumes a fixed, perhaps lower, power denoted as P_{idl}.

Each task is also characterized by a timing model. We use models from Real-Time Calculus (RTC) [6] to characterize the timing properties of the tasks. In RTC, each task T_i is characterized by its upper and lower arrival curves, denoted by α_i^u and α_i^l, respectively, which are defined as follows. Let the total amount of execution demanded by jobs (individual instances) of the task T_i that arrive in the time interval $[0, t]$ be $W_i(t)$. Then, the upper (lower) arrival curve $\alpha_i^u(\Delta)$ ($\alpha_i^l(\Delta)$), upper-bounds (lower-bounds) the amount of execution demanded by jobs of the task T_i that arrive in *any* time interval of length Δ, i.e.,

$$\alpha_i^l(\Delta) \leq (W_i(t+\Delta) - W_i(t)) \leq \alpha_i^u(\Delta), \quad \forall\, t \geq 0. \tag{2}$$

Arrival curves are general timing models of tasks: they can model several arrival patterns of jobs such as periodic with jitter, sporadic, and so on, along with execution time variability of individual jobs.

3.2 Thermal Model

The temperature of the system at time t is denoted by $\Theta(t)$. The temperature of the system when the power trace is $P(t)$ is denoted as $\Theta_P(t)$. The ambient temperature is considered to be a constant denoted by Θ_{amb}. (If the ambient temperature is variable, then the highest possible value is set as Θ_{amb}. This is a conservative step as temperature of the system is a monotonically increasing function of the ambient temperature.)

The thermal modelling of an electronic system is an ongoing research direction by itself. System-level studies, such as [7, 8, 9, 10, 11, 12] use approximate models such as the Fourier model described in (1). In this work, we consider an abstract model of the thermal system, using which several specific models, such as the Fourier model, can be expressed. However, we retain some of the assumptions considered in all the above papers such as (a) the active (power-consuming) part of the system is considered to be a point source of heat, and (b) only conductive heat transfer is considered.

Firstly, note that the thermal behavior of the system is generally governed by linear time-invariant (LTI) equations with power consumption as the input variable. As in standard LTI systems theory, the thermal behavior can be characterized by the impulse response of the temperature, i.e., the temperature due to a Dirac delta power source. We denote this impulse response as $h(t)$. We need an offset term, denoted as Θ_0, to denote steady-state conditions in the absence of any input, such as a non-zero ambient temperature. Thus, when the power trace is a Dirac delta function, the temperature of the system is given as below,

$$P(t) = \delta(0) \Rightarrow \Theta_P(t) = \Theta_0 + h(t). \tag{3}$$

We require that $h(t)$ is a non-increasing non-negative function. This assumption implies that the thermal effect of power consumed diminishes over time and additional power consumed never decreases the temperature. These are intuitive assumptions and are true for models we consider. For instance, for the Fourier heat model shown in (1), the impulse response is given as

$$h(t) = \frac{1}{C} \cdot e^{-t/(C/G)}, \tag{4}$$

with $\Theta_0 = \Theta_{amb}$. The above impulse response is indeed a monotonically decreasing positive function. Later in Section 8, we consider a more complex thermal model with distributed conduction and active cooling. As discussed in Section 8, the above assumption on the impulse response holds for that thermal model as well.

To conclude, we consider a thermal model that is given by a non-increasing impulse response $h(t)$ and a steady-state temperature Θ_0.

3.3 Scheduler Model

The system must arbitrate amongst the several tasks executing on the system. We use the Modular Performance Analysis (MPA) framework [13] to model the scheduler. MPA allows for specification of several commonly used scheduling policies such as fixed priority, Time Division Multiplexed Access (TDMA), Earliest Deadline First (EDF), First Come First Serve (FCFS), and their hierarchical compositions.

3.4 Problem Definition

Given is a system executing several tasks $T_1, T_2, \ldots, T_n$, each of which has repeated instantiations of jobs. Each task T_i is characterized by a power consumption ρ_i and upper- and lower-arrival curves, α_i^u and α_i^l. The thermal model is given by the impulse response $h(t)$ and the steady-state temperature Θ_0. The scheduling policy that is used to arbitrate between the tasks is also specified. The initial temperature of the system, i.e., at $t = 0$ is given to be $\Theta(0) = \Theta_0$. Also given is some finite time horizon of interest, denoted as τ, i.e., we are interested in the system behavior in the interval $[0, \tau]$.

Given such a system description, we are required to compute the upper-bound on the temperature of the system in the time interval $[0, \tau]$. This upper-bound must hold true for all possible valid traces of system behavior, as described in its model, throughout the time interval $[0, \tau]$.

4. OVERVIEW OF PROPOSED APPROACH

Having established the formal definition of the models and the problem, we now present an overview of the proposed approach to compute the thermal guarantee.

The approach is diagrammatically presented in Figure 6. There are three kinds of inputs to the system. Firstly the thermal model is characterized by the impulse response and steady-state temperature. Such inputs can be obtained by first obtaining the equivalent compact RC models [14] of the thermal system and then by analyzing the RC model using a standard circuit simulator. Alternately, specific thermal simulators such as Hotspot [15] can be used. Then the power model is captured by task-dependent power consumption

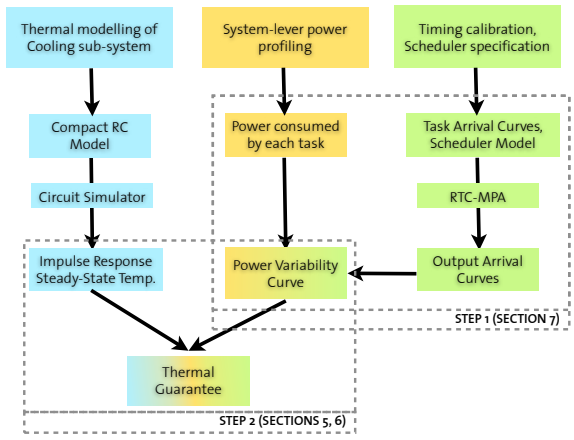


Figure 6: Overview of proposed method

values which can be computed using system-level profiling techniques such as the use of performance counters with simulators such as MPARM [16] and Wattch [17]. The third set of inputs is the timing properties of the system captured by the arrival curves of the tasks and the scheduler model. In all three input axes, we consider models of the respective properties with large modelling scope. This enables us to model realistically complex systems.

The proposed approach is divided into two steps as indicated in Figure 6. The common interface between the two steps is the Power Variability Curve (PVC), which is a formalism over a set of power traces. We first define and characterize PVC in Section 5. Then, we solve Step 2 in Section 6. In this step, with Power Variability Curve and the thermal properties as inputs, we compute the thermal guarantee. Having demonstrated Step 2, and thereby established the utility of the PVC formalism, we solve Step 1 in Section 7. In this step, the power consumption and arrival curves of each task and a specification of the scheduler are inputs to compute the Power Variability Curve. With both steps described, we solve the problem we set out to: Given the inputs of the thermal, power and timing models, compute an upper-bound on the peak temperature of the system.

5. POWER VARIABILITY CURVE

In this section, we formally define Power Variability Curve to compactly represent variability in a set of power traces. Later in Section 6, we show that this representation suffices to compute the upper-bound on the peak temperature, for the considered set of power traces.

5.1 Definition

The examples discussed in Section 2 illustrated that with task-dependent power, non-determinism in execution times and arrival patterns of tasks can lead to variability in the power traces. We would like to characterize the possible and allowable variability in such power traces, so as to be able to compute the thermal guarantee. To this end, we define the Power Variability Curve.

DEFINITION 1 (POWER VARIABILITY CURVE (PVC)). *Given is a set of power traces $\{P_1, P_2, \ldots P_n\}$ of length τ. We define $\phi(t)$ as the Power Variability Curve (PVC) of this set of power traces, such that*

$$\int_0^{\Delta} \phi(t)dt = \max_{\substack{i\in\{1,2,\ldots n\}\\ s\in[0,\tau]}} \left\{ \int_s^{s+\Delta} P_i(t)dt \right\}, \quad \Delta \in [0,\tau]. \quad (5)$$

In words, the Power Variability Curve (PVC) is such that the total energy consumed by the system in any time interval of length Δ, on any of the power traces considered, is no more than $\phi(\Delta)$. Given a PVC ϕ, a power trace P is said to *conform* to ϕ iff

$$\int_s^r P(t)dt \le \int_0^{r-s} \phi(t)dt, \quad 0 \le s \le r \le \tau. \quad (6)$$

For Examples 1 and 2 presented in Section 2, the shown power traces in Figures 1(a) and 3(a) are indeed the PVCs for the entire set of possible power traces of the systems described. All valid power traces of the systems described conform to their respective PVCs. In particular, this is true of the traces shown in Figures 1(b) and 3(b). For Example 1, intuitively, the power trace equals the PVC when all instances of task T_A execute for the WCET and all instances of task T_B execute for the BCET. For the second example, the power trace equals the PVC when the offset between the two tasks is 0, and both tasks exhibit maximum jitter (early arrival) during the arrival of the second instances of the tasks. We discuss later in Section 7 how to formally derive the PVCs.

Using the definition of the Power Variability Curve, a *set* of power traces is abstracted by a single curve, the PVC ϕ. The characterization provided by the PVC is very specific: it provides an upper-bound on the integral of power consumed in any interval of a given length. Of course, in such an abstraction, we lose some of the information about the individual power traces. However, as we shall see in the next section, this characterization is necessary and sufficient to compute the desired upper-bound on the peak temperature under the considered model assumptions.

6. COMPUTING THERMAL GUARANTEE USING POWER VARIABILITY CURVES

In this section, we present the core result of this work. Given a Power Variability Curve ϕ, we compute an upper-bound on the peak temperature of the system for any power trace that conforms to the considered PVC ϕ.

We being by first discussing how to compute the temperature of the system using the considered power and thermal models. Using standard linear systems theory, for an arbitrary power trace $P(t)$ we can express the temperature of the system at any time t using the convolution operator:

$$\begin{aligned}\Theta_P(t) &= \Theta_0 + (P \otimes h)(t) \\ &= \Theta_0 + \int_0^t P(u)\cdot h(t-u)du. \end{aligned} \quad (7)$$

We now define a specific power trace, called the flipped PVC trace, which is simply the PVC reversed in the interval $[0,\tau]$.

DEFINITION 2 (FLIPPED PVC TRACE). *Given is a Power Variability Curve ϕ of length τ. The flipped PVC trace, denoted as $\tilde{\phi}$, is a power trace that is defined as*

$$\tilde{\phi}(t) = \phi(\tau - t) \quad (8)$$

Using the defined flipped PVC trace, we now provide the core result of this work. The following result enables the computation of thermal guarantee of a system whose Power Variability Curve has been characterized.

THEOREM 1. *For any power trace, $P(t)$, conforming to a power variability curve ϕ, the temperature of the system at any time $s \in [0,\tau]$ is smaller than the temperature of the system at time τ with power trace equal to the flipped PVC trace, $\tilde{\phi}$, i.e.,*

$$\Theta_P(s) \leq \Theta_{\tilde{\phi}}(\tau), \quad \forall\ s \in [0,\tau]. \tag{9}$$

PROOF. Let $s \in [0,\tau]$ be any given time for which (9) has to be shown. Since, P conforms to PVC ϕ, we have

$$\int_0^{s-t} \phi(u)du \geq \int_t^s P(u)du, \quad \forall\ t \in [0,s]$$

$$\Rightarrow \int_t^s (\phi(s-u) - P(u))du \geq 0, \quad \forall\ t \in [0,s].$$

We define the left-hand side of the above inequality as a function, $f(t)$, on the domain $[0,s]$, i.e.,

$$f(t) = \int_t^s (\phi(s-u) - P(u))du, \quad t \in [0,s]. \tag{10}$$

The function f satisfies the following two properties:

$$f(t) \geq 0, \tag{11}$$

$$\frac{df(t)}{dt} = -(\phi(s-t) - P(t)). \tag{12}$$

From the thermal model of the system of (7), we have

$$\begin{aligned}
&\Theta_{\tilde{\phi}}(\tau) - \Theta_P(s) \\
=&(\Theta_0 + (\tilde{\phi} \otimes h)(\tau)) - (\Theta_0 + (P \otimes h)(s)) \\
=&\int_0^\tau \tilde{\phi}(t) \cdot h(\tau - t)dt - \int_0^s P(t) \cdot h(s-t)dt
\end{aligned}$$

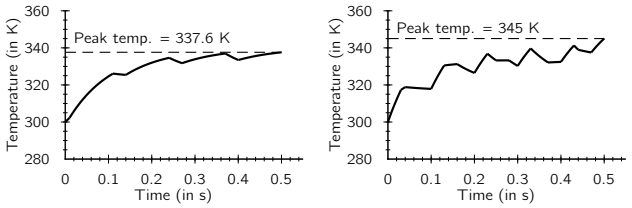
Since, the impulse response and the power variability curve are always non-negative we have,

$$\begin{aligned}
&\Theta_{\tilde{\phi}}(\tau) - \Theta_P(s) \\
\geq&\int_{\tau-s}^\tau \tilde{\phi}(t) \cdot h(\tau - t)dt - \int_0^s P(t) \cdot h(s-t)dt \\
=&\int_0^s \tilde{\phi}(t+\tau-s) \cdot h(s-t)dt - \int_0^s P(t) \cdot h(s-t)dt \\
=&\int_0^s (\phi(s-t) - P(t)) \cdot h(s-t)dt \\
=&-\int_0^s \frac{df(t)}{dt} \cdot h(s-t)dt
\end{aligned}$$

Integrating the above by parts, we have,

$$\begin{aligned}
&\Theta_{\tilde{\phi}}(\tau) - \Theta_P(s) \\
=&\int_0^s \frac{dh(s-t)}{dt} f(t)dt - h(s-t)f(t)\big|_0^s \\
=&\int_0^s \frac{dh(s-t)}{dt} f(t)dt - h(0)f(s) + h(s)f(0)
\end{aligned}$$

We show that the above value is always non-negative. We know that the impulse response, $h(t)$, is a non-increasing function. Thus, $\frac{dh(s-t)}{dt}$ is always non-negative for $t \in [0,s]$. Also, we know that $f(t) \geq 0$ for $t \in [0,s]$. Finally, $f(s) = 0$ and $h(s), f(0) \geq 0$. Using these, we know that the right hand side of (13) is non-negative and thus, $\Theta_{\tilde{\phi}}(\tau) \geq \Theta_P(s)$. □



(a) (b)

Figure 7: Temperature evolution for flipped PVC traces for (a) Example 1 and (b) Example 2

We revisit the two examples of Section 2. As discussed, the traces shown in Figures 1(a) and 3(a) are indeed the PVCs for the respective systems. We use this to compute the flipped PVC trace for each system and simulate the thermal behavior of the system for these traces. The corresponding temperature traces are shown in Figure 7. Firstly, observe that the peak temperature with the flipped PVC traces are obtained at time $t = \tau = 500$ms. Also, the peak temperature obtained with the flipped PVC traces upper-bounds the peak temperatures obtained with the thousands of random traces as shown in Figure 5. These plots thus numerically demonstrate Theorem 1, for the considered examples: the upper-bound on the peak temperature is obtained at time $t = \tau = 500$ms on the flipped PVC power trace. The simulation of thousands of traces has been improved upon by the simulation of a single trace.

The power trace considered in Theorem 1, namely the flipped PVC trace, is independent of the parameters of the thermal model. The result of the theorem relies only on the non-increasing and non-negative nature of the impulse response $h(t)$. This can be considered as a separation of concerns: the worst-case power trace that we should be prepared for is the same, independent of exact parameters of the thermal model. Architectural-level techniques can be explored to *reduce* this worst-case power trace.

7. COMPUTING THE PVC USING REAL-TIME CALCULUS

So far, we showed how to characterize the variability of power traces using Power Variability Curves and showed how such a characterization is useful to compute a provably safe upper-bound on the peak temperature. In the examples shown in Section 2, we understood that the variability in the power traces arises due to the timing uncertainties. Such uncertainties are dependent on the execution times of tasks, their arrival times, and also on the scheduling policies used. Study of such properties of systems has been the main focus of the real-time systems community. A result of one stream of research in this community is the theory of Real-Time Calculus (RTC) which along with Modular Performance Analysis (MPA) can be used to specify and analyze real-time properties of general systems. In this section, we use known results from RTC-MPA to demonstrate the computation of PVC.

7.1 Real-Time Calculus and Modular Performance Analysis

Thiele et. al., in [6], proposed Real-Time Calculus (RTC) building on existing results in Network Calculus [18]. RTC as a theory can be used to specify general models of the resource requirement of tasks (streams of jobs) and the service provided by a resource. Using operators that are part of the calculus, several quantities of interest can be computed, for instance worst-case delay and worst-case buffer requirement when the tasks are served by the resource. Modular Performance Analysis (MPA) [13] uses the theory of RTC to analyze models of real-time embedded systems. With the help of abstract components different resource sharing mechanisms can be expressed. Such components can be composed together to model and analyze realistic systems.

We can model our system with tasks characterized by the upper and lower arrival curves and the specified scheduling policy, in the RTC-MPA framework. We can then compute for each task, bounds on the amount of execution time it receives in time intervals of given length. For task T_i, the lower and upper *output* arrival curves, denoted as $\alpha_i^{\prime l}$ and $\alpha_i^{\prime u}$, respectively, lower- and upper- bound the amount of time a given task executes in any time interval of length Δ. Let $C(t)$ denote the amount of time a specific task executes for in the time interval $[0, t]$. Then, from the definition of the output arrival curves, we have

$$\alpha^{\prime l}(\Delta) \leq (C(t+\Delta) - C(t)) \leq \alpha^{\prime u}(\Delta), \quad \forall\, t, \Delta \geq 0. \quad (13)$$

7.2 Idle Task

We first begin by discussing how we model the idling of the system. According to the system model, when not executing any task, the processor is idle with power consumption P_{idl}. We interpret this as executing a dummy idle task, T_{n+1} with $\rho_{n+1} = P_{idl}$. The idle task must execute when, and only when, there are no other tasks in the system. We model this with two features. Firstly, the top-level scheduler of the system is a fixed priority preemptive scheduler, the lowest priority of which is assigned to the idle task. Secondly, the idle task is modelled to be always available, i.e., $\alpha^u_{n+1}(\Delta) = \Delta$. Given this modelling, the system idling is considered as simply executing this idle task and thus the system always executes some task.

7.3 The Energy Function

To compute the PVC, we first define the the *energy function* Φ which is given as

$$\Phi(\Delta) = \max_{(\Delta_1, \Delta_2, \ldots \Delta_n) \in \eta(\Delta)} \left(\sum_{i \;|\; T_i \in \mathbf{T}} (\rho_i \cdot \Delta_i) \right), \quad (14)$$

where,

$$\begin{aligned} &(\Delta_1, \Delta_2, \ldots \Delta_n) \in \eta(\Delta) \\ &\iff \Delta_i \in [\alpha_i^{\prime l}(\Delta), \alpha_i^{\prime u}(\Delta)],\ \forall i \in \{1, \ldots, n\} \\ &\qquad \wedge \quad \Delta_1 + \Delta_2 + \ldots + \Delta_n = \Delta. \end{aligned} \quad (15)$$

Let us understand the intuition behind the definition of the energy function. The value $\eta(\Delta)$ as defined in (15) gives the potentially infinite set of ordered pairs of possible execution times of tasks in any interval of length Δ, i.e., if $(\Delta_1, \ldots, \Delta_n)$ belongs to $\eta(\Delta)$, then it is plausible that in some time interval of length Δ, task T_i runs for Δ_i time, for $i \in \{1, \ldots, n\}$.

Algorithm 1 Computing $\Phi(\Delta)$

```
Φ(Δ) ← 0
δ ← 0
s ← 1
for each task T_i do
5:   Φ(Δ) ← Φ(Δ) + α'^l_i(Δ) × ρ_i
     δ ← δ + α'^l_i(Δ)
end for
while δ < Δ do
     j ← I(s)
10:  t ← min(α'^u_j(Δ) − α'^l_j(Δ), Δ − δ)
     Φ(Δ) ← Φ(Δ) + t × ρ_j
     δ ← δ + t
     s ← s + 1
end while
```

In any time interval of length Δ, for any $i \in \{1, \ldots, n\}$, the amount of time for which task T_i executes is in the range $[\alpha_i^{\prime l}(\Delta), \alpha_i^{\prime u}(\Delta)]$. The timing uncertainties allow for a variation, but the variation is bounded to a range as given by the output arrival curves. Further, having modelled the idle task as a dummy task, we know that the system is always executing a task. Thus, the sum of the execution times of all tasks in this interval of time must exactly equal Δ. Using these two facts, the set $\eta(\Delta)$ is as given in (15). Amongst all such ordered pairs $\eta(\Delta)$, we maximize the total energy consumed by all tasks in that interval of time and set $\Phi(\Delta)$ as this maximum value in (14).

Computing Φ as defined in (15), i.e., by identifying the potentially infinite set η, is not practical. For our specific model, where each task T_i has a power consumption ρ_i, we discuss an efficient method to compute Φ. We consider a function $\mathcal{I}$ where $\mathcal{I}(s)$ denotes the index of the task with the sth highest power consumption. For instance, if $j := \mathcal{I}(1)$, then task T_j is the task with the highest power consumption. Using this function, we show how to compute $\Phi(\Delta)$, for any given Δ, in Algorithm 1. The timing complexity of the algorithm is of the order of $O(n)$, where n is the total number of tasks in the system. It can be easily verified that the Φ computed in Algorithm 1, equals the energy function defined in (15).

THEOREM 2. *For a system, with energy function $\Phi(t)$ computed as in (14), the Power Variability Curve (PVC) is*

$$\phi(t) = \frac{d\Phi(t)}{dt} \quad (16)$$

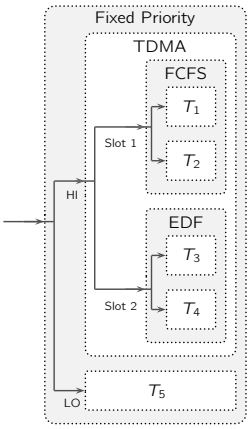
PROOF. This is easily proved by contradiction. Let ϕ not be a true Power Variability Curve (PVC). Then, there must be some power trace P such that for some s and Δ, we have

$$\int_s^{s+\Delta} P(t)dt > \int_0^{\Delta} \phi(t)dt = \int_0^{\Delta} d\Phi(t) = \Phi(\Delta). \quad (17)$$

Let Δ_i denote the amount of time for which task T_i executes for, in the trace P in the time interval $[s, s+\Delta]$, for $i \in \{1, \ldots, n\}$. From RTC-MPA, we know that for any valid trace the upper and lower output arrival curves bound the values of Δ_i, and $\sum_i \Delta_i = \Delta$. Thus $(\Delta_1, \ldots, \Delta_n) \in \eta(\Delta)$. Then from the definition of Φ given in (14), we have

$$\int_s^{s+\Delta} P(t)dt \leq \Phi(\Delta). \quad (18)$$

From (17) and (18) we have a contradiction. □



(a)

Task	Power (W)	BCET	WCET
T_1	4	7	16
T_2	9	18	20
T_3	5	4	8
T_4	12	5	7
T_5	2	-	-

TDMA Slot	Length
Slot 1	30
Slot 2	10

Task	Min. IAT	Max. IAT
T_1	80	100
T_2	55	75

Task	Period	Jitter	Rel. Deadline
T_3	60	10	50
T_4	45	5	60

(b)

Figure 8: (a) Hierarchical scheduling of tasks, and (b) Task parameters. All times shown are in ms.

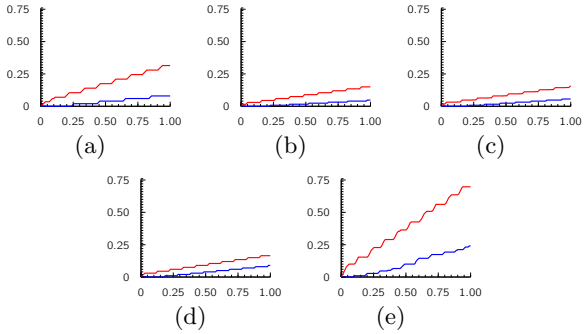


Figure 9: Output arrival curves for different tasks (a) T1, (b) T2, (c) T3, (d) T4, and (e) T5. Time in seconds is in the x-axis. Execution time in seconds is in the y-axis. The curves in (blue) red denote the (lower) upper output arrival curves.

Given output arrival curves for all tasks obtained from RTC-MPA analysis, computation of the Power Variability Curve is straight forward. We compute the energy function Φ using Algorithm 1 and differentiate it to obtain the PVC ϕ. The reason for this simplicity is apparent. The formalism of PVC characterizes all possible power traces of a system by using a single curve that gives the maximum amount of energy consumed by the system in any interval of a given length. Alternately stated, PVC bounds the energy consumed in the so-called interval domain. We have seen how this is sufficient to bound the peak-temperature of the system. Real-Time Calculus also provides results, in our case the output arrival curves, on the interval domain. This match allows for conveniently abstracting the potentially complex timing uncertainties of the system by using RTC-MPA to compute output arrival curves, and subsequently computing the PVC.

8. EXPERIMENTAL RESULTS

In this section, we present a case study of a system, where several tasks consuming different amounts of power are scheduled using a hierarchical scheduler. We model the system using Real-Time Calculus and Modular Performance Analysis and compute the timing properties of interest. Combining these values with the power consumption of the task, we compute the energy function Φ and subsequently the Power Variability Curve (PVC). From the PVC we obtain the flipped PVC trace. We consider two different thermal models, namely the Fourier conductive heat model and a model considering distributed conduction and active liquid cooling. For either model, we obtain the proven safe upper-bound on the peak temperature by simulating the obtained flipped PVC trace.

We first describe the task and scheduler considered. We consider a hierarchically scheduled system serving 4 tasks T_{1-4}, that is graphically described in Figure 8(a). On the top-most layer we have a preemptible fixed priority scheduler, with low priority (LO) given to the idle task, denoted as T_5, and high priority given to the actual tasks, T_{1-4}. The actual tasks are scheduled with Time Division Multiplexed Access (TDMA) with two slots. In the first TDMA slot, two sporadic tasks characterized by maximum and minimum inter-arrival time (IAT) are scheduled using First-Come-First-Serve (FCFS) policy. In the second slot two periodic tasks with jitter, T_3 and T_4, are scheduled using Earliest Deadline First (EDF) policy. Each instance of either task has a deadline equal to its arrival time plus a relative deadline. We describe all the timing characteristics of the tasks in Figure 8(b). Also shown is the power consumption of each of the tasks, including the idle task. We consider a time horizon τ of 1s. The aim of this case study is to illustrate the potential complexity of the model that can be analyzed by using the standard models and tools.

We model and analyze the above system using the Real-Time Calculus Toolbox available for MATLAB. In particular, this requires computing the arrival curves of the input streams of each of the tasks, and instantiation and interconnection of the different abstract components to model the resource sharing used. As a result of the analysis we obtain the output arrival curves for all the tasks. These curves are shown in Figures 9(a)-(e). At any x-coordinate the red and blue lines upper- and lower- bound the amount of execution that the task can receive in any time interval of length equal to the x-coordinate. Using these curves and the power consumption of the tasks, we compute the energy function Φ which is shown in Figure 10(a). Differentiating the energy function gives us the Power Variability Curve ϕ. Mirroring it about the time horizon τ gives us the flipped PVC trace which is as shown in Figure 10(b).

As discussed in Section 4, this marks the end of one step of the analysis process. We have fully and minimally characterized the variability in the tasks, the scheduler and the power consumption of the tasks, in the form of the flipped PVC trace. For the system model considered, the flipped PVC trace is the worst-case power trace independent of the parameters of the thermal model.

We now turn our attention to the thermal model. To highlight the general nature of the approach presented in this work, we present the results for two different thermal models. As the first thermal model, we consider the Fourier model of conductive heat dissipation as described in (1). The parameters for a typical processor are provided in Table 3(a). For this thermal model, the impulse response is obtained to be $h(t) = 33.3\ e^{-10t}$K/J, and $\Theta_0 = 300K$. With this

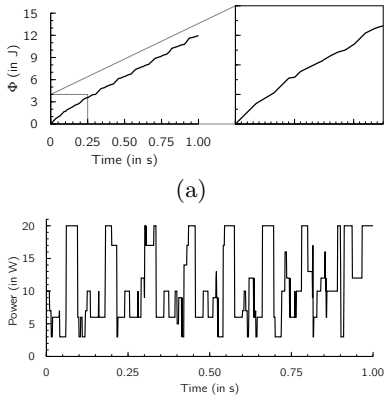


Figure 10: (a) Function Φ for the considered example. (b) Flipped PVC trace derived from Φ

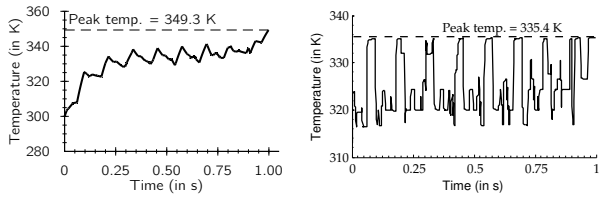


Figure 11: Temperature plot for flipped PVC trace for (a) the Fourier heat model, and (b) the thermal model with active cooling.

impulse response, we can thermally simulate the system for the obtained flipped PVC trace shown in Figure 10(b). The temperature of the system as a function of time is plotted in Figure 11(a). The upper-bound on the peak temperature is computed to be 349.3 K and, as expected, is obtained at $t = \tau = 1$s. This is the desired thermal guarantee.

Now we consider a more complex thermal model. We consider a system where the thermal model has 27 cells as shown in Figure 12(a). The active (power consuming) part of the chip is at the very centre of the system. On the top layer, we have a Cu layer which conducts heat to the ambient. On the bottom layer we have a channel etched into Si that provides active cooling, such as in [19]. The parameters

(a) Fourier model

Parameter	G	C	Θ_{amb}
Value	0.3 W/K	0.03 J/K	300 K

(b) Complex thermal model

Param.	G_{AmbCu}	G_{CuCu}	G_{CuSi}	G_{SiSi}
Value	0.03 W/K	0.34 W/K	0.08 W/K	0.11 W/K
Param.	G_{SiLq}	G_{LqLq}	Θ_{amb}	Θ_{liq}
Value	0.35 W/K	0.03 W/K	300 K	315 K
Param.	C_{Cu}	C_{Si}	C_{Lq}	
Value	5 mJ/K	0.5 mJ/K	0.3 mJ/K	

Table 3: Thermal parameters of the two thermal models considered in the paper.

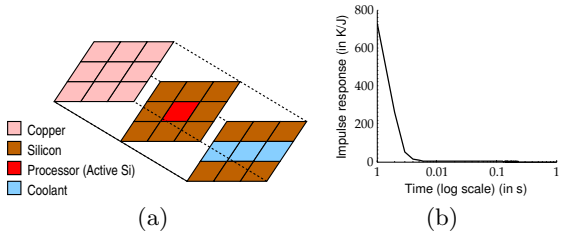


Figure 12: (a) Schematic of the general thermal model considered, and (b) its impulse response.

of the thermal model are shown in Table 3(b). The G terms denote the thermal conductivity between the different materials shown, the C terms denote the specific heat capacity of the different materials and the Θ terms denote the constant temperatures of the ambient and the liquid coolant. These parameters have been adapted from [20] and [12]. We make simplifying assumptions on the system: (a) a single liquid channel is considered, (b) the liquid is assumed to be always at a constant temperature Θ_{liq} throughout the channel, and (c) conduction is the only mode of heat exchange between cells. Though highly simplified, in the context of this work, the model is representative of the features of a thermal model with a complicated cooling sub-system. We analyzed the corresponding equivalent RC network on a circuit simulator and obtained the impulse response which is shown in Figure 12(b). The figure is plotted with a log x-axis, because for the parameters chosen, the impulse response falls rapidly, initially. As can be noted, the impulse response is monotonically decreasing and non-negative; the two assumptions that are required for the presented analysis to hold. The offset term Θ_0 was obtained by DC analysis of the conductance network to be 313.8 K.

As before, we simulate this thermal system for the flipped PVC trace and obtain the temperature as a function of time. This is shown in Figure 11(b). The upper-bound on the peak temperature is computed to be 335.4 K, and is again noted for $t = \tau = 1$s. This is the desired thermal guarantee.

9. RELATED WORK

Several research works have considered problems that simultaneously consider thermal aspects of the system along with real-time properties of the tasks. Some of them, such as [7] and [8], consider the problem of computing and then optimizing the peak temperature of the system under performance constraints of applications. While others, such as [9] and [10], attempt to maximize the responsiveness of tasks under a given constraint of peak temperature. The main difference between our work and these and several similar studies is the choice of the task model. In existing work, simple task arrival patterns such as periodic or a given trace of sporadic tasks are chosen. However, in reality such static characterization of tasks is not available. For instance, tasks exhibit bursts, at unknown times. Capturing and analyzing such aspects is crucial for obtaining the thermal guarantee. As an example, if a burst occurs while the processor is already hot, and the tasks are allowed to run, we are likely to see a high peak temperature. At the same time, to compute thermal guarantees, bounds must exist on the task arrival patterns, for instance bounds on the magnitude of the bursts

and their frequency. Arrival curves generalize task arrival patterns to represent variability in the resource demand of jobs, but retain bounds to compute the thermal guarantees.

To the best of our knowledge, only one other existing work considers arrival curves as the task model. In [11], the authors consider a workload conserving processor, serving a single stream of jobs that are characterized by an upper arrival curve, and aim to compute an upper-bound on the peak temperature of the system. The authors assume that all jobs consume the same amount of power and consequently scheduling is non-existent. The work presented here considerably generalizes this study. Firstly, we consider several tasks, each potentially consuming different amounts of power. Secondly we allow for a general scheduling of these tasks on the system, which can be non-workload conserving (by virtue of the defined idle task). In essence, the variability in the power consumption considered in [11] arises from the different patterns of the task arrivals. in our framework, in addition to the patterns of task arrivals, variability in power consumption occurs due to two additional factors: (a) different tasks consuming different amounts of power, and (b) the scheduler choosing the task to run amongst all the competing tasks. This generalizes the framework and truly captures the working of a realistic system.

10. CONCLUSIONS

Managing on-chip temperatures is likely to be an important aspect in the design of electronic devices. The efforts to achieve such thermal management will be on several fronts: improved VLSI design, system-level power optimizations, improved cooling solutions, run-time DTM techniques and so on. Both for high-end performance-hungry systems and for low-end resource constrained systems, design and optimization of such solutions, though challenging, is inevitable. Any such solution is only as good as the thermal guarantee it provides, i.e., the upper-bound on the temperature of the system. The computation of such a bound is complicated by the potentially infinite traces of behavior that a system can exhibit owing to timing non-determinism. To tackle this problem, in this work, we presented the formalism of Power Variability Curve (PVC) that captured the allowable variability in power traces with respect to computing the thermal guarantee. We showed that, independent of the thermal model (given some mild assumptions), the PVC captures the worst-case power consumption of the system. The PVC is an intermediate quantity that can be used to optimize the timing and power properties of the tasks in order to positively change the worst-case power trace, independent of the actual parameters of the thermal model. We showed how known results from Real-Time Calculus (RTC) and Modular Performance Analysis (MPA) can be used to compute the PVC for systems with task-dependent power consumption. The convenient match of using the interval domain in PVC formalism and RTC-MPA allowed for this straight forward coupling. An important advantage of our method is the use of general models to express the inputs which enables us to model realistically complex systems.

One direction forward is to extend this work to multi-core systems. The complexity of such an extension is due to the existence of multiple sources of power dissipation. In such systems, apart from time, space will also play a crucial role. Perhaps, formalisms on a *sets* of traces need be considered.

Acknowledgments

This work is supported by PRO3D project financed by the European Community FP7 programme (FP7-ICT-248776).

11. REFERENCES

[1] O. Semenov, A. Vassighi, and M. Sachdev, "Impact of self-heating effect on long-term reliability and performance degradation in CMOS circuits," *Device and Materials Reliability, IEEE Transactions on*, vol. 6, no. 1, 2006.

[2] S. Borkar, "Design challenges of technology scaling," *Micro, IEEE*, vol. 19, no. 4, pp. 23 –29, 1999.

[3] E. Wang and et. al., "Micromachined jets for liquid impingement cooling of VLSI chips," *Microelectromechanical Systems, Journal of*, vol. 13, no. 5, pp. 833 – 842, 2004.

[4] D. Brooks and M. Martonosi, "Dynamic Thermal Management for High-Performance Microprocessors," in *HPCA*, 2001.

[5] D. Atienza and et. al., "A fast hw/sw fpga-based thermal emulation framework for multi-processor system-on-chip," in *DAC*, pp. 618–623, 2006.

[6] L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," in *ISCAS*, vol. 4, pp. 101 –104 vol.4, 2000.

[7] N. Bansal, T. Kimbrel, and K. Pruhs, "Dynamic Speed Scaling to Manage Energy and Temperature," in *FOCS*, 2004.

[8] S. Murali and et. al., "Temperature-aware processor frequency assignment for MPSoCs using convex optimization," in *CODES+ISSS*, pp. 111–116, 2007.

[9] J.-J. Chen, S. Wang, and L. Thiele, "Proactive Speed Scheduling for Real-Time Tasks under Thermal Constraints," in *RTAS*, 2009.

[10] S. Wang and R. Bettati, "Reactive speed control in temperature-constrained real-time systems," *Real-Time Systems*, vol. 39, no. 1-3, 2008.

[11] D. Rai, H. Yang, I. Bacivarov, J.-J. Chen, and L. Thiele, "Worst-Case Temperature Analysis for Real-Time Systems," in *DATE*, 2011.

[12] P. Kumar and L. Thiele, "Thermally optimal stop-go scheduling of task graphs with real-time constraints," in *ASP-DAC*, pp. 123–128, 2011.

[13] E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse, "System architecture evaluation using modular performance analysis: a case study," *STTT*, vol. 8, no. 6, 2006.

[14] W. Huang, M. R. Stan, K. Skadron, K. Sankaranarayanan, S. Ghosh, and S. Velusamy, "Compact thermal modeling for temperature-aware design," in *DAC* (S. Malik, L. Fix, and A. B. Kahng, eds.), pp. 878–883, ACM, 2004.

[15] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. R. Stan, "Hotspot: A compact thermal modeling methodology for early-stage vlsi design," *IEEE Trans. VLSI Syst.*, vol. 14, no. 5, pp. 501–513, 2006.

[16] L. Benini, D. Bertozzi, A. Bogliolo, F. Menichelli, and M. Olivieri, "Mparm: Exploring the multi-processor soc design space with systemc," *VLSI Signal Processing*, vol. 41, no. 2, pp. 169–182, 2005.

[17] D. Brooks, V. Tiwari, and M. Martonosi, "Wattch: a framework for architectural-level power analysis and optimizations," in *ISCA*, pp. 83–94, 2000.

[18] J.-Y. L. Boudec and P. Thiran, *Network Calculus: A Theory of Deterministic Queuing Systems for the Internet*. Lecture Notes in Computer Science, Springer, 2001.

[19] T. Brunschwiler and et. al., "Interlayer cooling potential in vertically integrated packages," *Microsystem Technologies*, vol. 15, pp. 57–74, 2009. 10.1007/s00542-008-0690-4.

[20] A. Sridhar and et. al., "Compact transient thermal model for 3D ICs with liquid cooling via enhanced heat transfer cavity geometries," in *THERMINIC*, pp. 1 –6, 2010.