Worst-Case Temperature Analysis for Different Resource Models

Lars Schor, Hoeseok Yang, Iuliana Bacivarov, and Lothar Thiele
Computer Engineering and Networks Laboratory
ETH Zurich, 8092 Zurich, Switzerland
firstname.lastname@tik.ee.ethz.ch

Abstract

The rapid increase in heat dissipation in real-time systems imposes various thermal issues. For instance, real-time constraints cannot be guaranteed if a certain threshold temperature is exceeded, as it would immediately reduce the system reliability and performance. Dynamic thermal management techniques are promising methods to prevent a system from overheating. However, when designing real-time systems that make use of such thermal management techniques, the designer has to be aware of their effect on both real-time constraints and worst-case peak temperature. In particular, the worst-case peak temperature of a real-time system with non-deterministic workload is the maximum possible temperature under all feasible scenarios of task arrivals. This article proposes an analytic framework to calculate the worst-case peak temperature of a system with general resource availabilities, which means that computing power might not be fully available for certain time intervals. The event and resource models are based on real-time and network calculus, and therefore, our analysis method is able to handle a broad range of uncertainties in terms of task arrivals and available computing power. Finally, we propose an indicator for the quality of the resource model with respect to worst-case peak temperature and schedulability.

1 Introduction

The miniaturization in the semiconductor industry and the demand for higher performance impose a major rise in heat dissipation per unit area, which in turn threats the reliability and performance of real-time systems. Moreover, nowadays, the thermal wall is recognized as one of the most significant barriers towards high performance systems [1], as high chip temperatures may lead to long-term reliability concerns and short-term functional errors. Providing guarantees on maximum temperature is therefore as important as functional correctness and timeliness.

As modern processors support several operation frequencies and deep power down states for power efficient design, dynamic thermal management (DTM) based on dynamic voltage/frequency scaling [2] is considered as a promising

technique to increase the system reliability. Despite their thermal effectiveness, these techniques cause a significant degradation of performance or lead to an expensive run-time overhead, both unacceptable in today's real-time systems where a certain response time must be guaranteed. Furthermore, the use of reactive thermal management techniques can even cause a potential violation of real-time constraints. As high temperatures can significantly reduce the system's performance, real-time constraints can only be guaranteed if the worst-case peak temperature of the system is incorporated in real-time analysis, at design-time.

Unfortunately, none of the previous work has studied the effect of general resource availabilities on the maximum temperature of a system under all possible scenarios of task executions, even though this knowledge is desirable for any modern real-time system. In [3], an analytic system-level analysis has been proposed, which calculates the worst-case peak temperature of a real-time system. However, as the work assumes full service availability, i.e., the processor is idle if and only if there is no workload to proceed, the method is not able to analyze DTM techniques. In order to close this gap, we describe and evaluate a method to determine an upper bound on the peak temperature of a real-time system with general resource availabilities under all possible scenarios of task executions. In contrast to a system with full service availability, a system with general resource availability might have no or only reduced computing power available for certain time intervals even if there is workload to proceed. This enables us, on the one hand, to study the effect of dynamic frequency scaling (DFS) or time division multiple access (TDMA) scheduling on the worst-case peak temperature, and, on the other hand, to calculate an indicator for the quality of the employed resource model with respect to the worst-case peak temperature. Throughout this paper, the term resource model is used to describe the availability of the processor's computing resources.

We apply real-time calculus [4], a formal method for schedulability and performance analysis of real-time systems, to temperature analysis. In particular, arrival curves are used to upper bound the workload that might arrive at the system in any time interval. In general, arrival curves can be derived either from application specifications or by profiling sufficient representative traces [5]. Although workload curves constrain the maximum possible workload arriving at the system, there are infinitely many traces that comply with a given event stream specification. Therefore, exhaustive search to compare the maximum temperature of all possible traces is usually infeasible. Due to dynamic resource management, the other uncertainty of the system is the availability of computing resources. Consequently, so-called service curves are used to upper and lower bound the computing resource provided by a processor in a given time interval.

This article is based on the work published in [6], which discusses general resource availabilities with the help of a case study. However, the prior work does not state any formal explanations on how to design a processing component so that the worst-case peak temperature is reduced and no real-time deadlines are missed. We extend the previous work by a method to derive a resource model so that the temperature is optimally reduced and no real-time deadline is missed. In other words, we calculate a lower bound on the peak temperature for a given set of tasks, each characterized by an arrival curve and a relative deadline. In

addition, this article gives a broader coverage of related work, the proofs of the stated methods are more detailed, and a richer set of experiments is carried out. Therefore, the contributions of this article can be summarized as follows:

- An upper bound on the peak temperature of a real-time system with non-deterministic workload and resource availability is determined.
- An indicator for the quality of the examined resource model with respect to worst-case peak temperature is formally derived.
- In various case studies, the proposed method is applied to four different resource models, namely full resource availability with frequency modulation, availability with bounded-delay, availability based on TDMA, and periodic availability.

The remainder of the article is organized as follows: First, related work is discussed in the next section. Afterwards, Section 3 introduces the computational, power, and thermal models considered in this article. The thermal analysis method and its formal proofs are induced in Section 4. In Section 5, an indicator for the quality of a resource model is proposed and finally, Section 6 presents case studies to highlight the viability of our method.

2 Related Work

As modern processors support several operation frequencies, dynamic power management methods like dynamic voltage and frequency scaling (DVFS) [2] are promising techniques to reduce the power consumption and to prevent the system from overheating. Therefore, thermal management has been a hot topic in research in recent years including thermal-constraint scheduling to maximize the performance [7–10] or peak temperature reduction to meet performance constraints [11, 12]. In [7], a DVFS control policy is proposed to maximize the workload of a uniprocessor system with discrete speed levels under thermal constraints. In [8], a convex optimization technique for temperature-aware frequency assignments is proposed. Several architectural-level techniques for thermal management on multicore processors are evaluated in [9] and thermal management techniques for unknown workload like load balancing or temperature aware random scheduling are discussed in [10]. Bansal et al. [11] explore peak temperature reduction by adapting an on-line algorithm for energy efficiency. Finally, thermal aware heuristics to reduce the maximum and average temperature are compared with power-aware heuristics in [12].

Estimating the temperature is typically a two-step procedure. First, the transient power dissipation of the system is determined by means of a power-aware simulator, either software-based [13, 14] or hardware-based [15]. Afterwards, the power dissipation is used to evaluate the transient temperature evolution in a thermal simulator like HotSpot [16]. However, due to the complexity of today's systems, it is difficult to identify corner cases that actually lead to the maximum temperature of the system under all feasible scenarios of task arrivals and consequently, simulation-based thermal analysis methods may lead to an undesired underestimation of the maximum temperature.

This drawback is addressed in [3], where the worst-case peak temperature of a real-time system is calculated from its system-level specification. However, as the work assumes work-conserving systems with full service availability, the method is not able to analyze DTM techniques. Based on this work, leaky bucket shapers are studied in [17] so that no job misses its real-time deadline and the peak temperature is optimally reduced. Finally, a method to calculate the worst-case peak temperature of a real-time system with multiple power sources and full service availability is proposed in [18]. In the work at hand, we study the effect of different resource models onto the worst-case peak temperature of a system. To this end, we restrict us to a uniprocessor system and omit the effect of multiple power sources.

3 System Model

This section introduces the models used to analyze the worst-case peak temperature.

3.1 Computational Model

The computational model of a processor is expressed using abstractions in real-time calculus [4]. Consequently, we suppose that in time interval $[s, t)$, events with a total workload of $R(s, t)$ time units arrive at the processing component. For any fixed s, one can think of $R(s, t)$ being a staircase function, that increases its value by its computation time when a task arrives. The arrival curve α upper bounds all possible cumulative workloads:

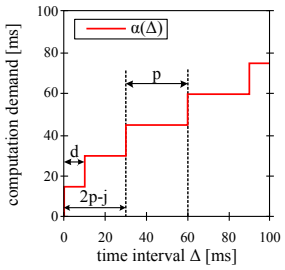
$$R(s,t) \leq \alpha(t-s) \quad \forall s < t \tag{1}$$

given $\alpha(0) = 0$. From the properties of $R(s, t)$ follows that a tight upper arrival curve is also a monotonically increasing staircase function, which is sub-additive, i.e., it satisfies $\alpha(s) + \alpha(t) \geq \alpha(s + t)$. In literature, a widely-used concept for workload specification is the standard event arrival pattern, defined by the parametric triple $(p,\ j,\ d)$, where p denotes the period, j the jitter, and d the minimum inter-arrival distance between events [19]. Figure 1a illustrates these parameters and the corresponding arrival curve is graphically depicted.

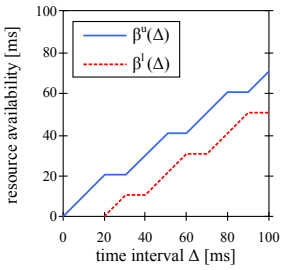
Suppose that the processor concurrently processes a set of independent tasks ν_k whose workload functions R_k are individually bounded by arrival curves α_k. Then the accumulated workload can be bounded by:

$$R(s,t) \leq \sum_{\forall k} \alpha_k(t-s) = \alpha(t-s). \tag{2}$$

The considered processor is abstracted by a resource model, that incorporates information about the computing resources as for example the processor speed. We characterize a resource model as follows. Events that have not yet been completed are queued in the component's ready queue while waiting for further resources. A processing component is given $W(s, t)$ computing resources in



(a) Arrival curve that corresponds to an event stream with a period p, a jitter j and a minimum event inter-arrival distance d.



(b) Service curve set for a periodically available resource as defined in [20].

Figure 1: Examples of a typical arrival curve and service curve.

time interval $[s, t)$. If there are no waiting or arriving tasks in $[s, t)$, the available resource $W(s, t)$ is wasted. Otherwise, it is used to process these tasks. A processing component that behaves according to the above explanation is called work-conserving. There are no further assumptions on the scheduling, and therefore, most traditional scheduling algorithms as, for example, earliest-deadline-first (EDF), rate-monotonic (RM), fixed-priority (FP), and deadline-monotonic (DM) can be analyzed by the proposed abstraction. The above described resource model leads to the concept of service curves. The resource availability W is upper and lower bounded by a pair of upper and lower service curves:

$$\beta^l(t-s) \leq W(s,t) \leq \beta^u(t-s) \quad \forall s < t \tag{3}$$

given $\beta^l(0) = \beta^u(0) = 0$. Figure 1b illustrates the concept of service curves based on a model with periodically available resources as defined in [20].

The accumulated computing time $Q(s, t)$ describes the amount of time units that a component is spending to process an incoming workload of $R(s, t)$. The accumulated computing time $Q(s, t)$ in time interval $[s, t)$ is:

$$Q(s,t) = \inf_{s \leq u \leq t} \{W(s,t) - W(s,u) + R(s,u)\} \tag{4}$$

provided that there is no buffered workload in the ready queue at time s [4]. Its upper bound can be determined as [21]:

$$Q(t-\Delta, t) \leq \gamma(\Delta) = \min\{\left((\alpha \otimes \beta^u) \oslash \beta^l\right)(\Delta), \beta^u(\Delta)\} \tag{5}$$

where $(f \otimes g)(\Delta) = \inf_{0 \leq \lambda \leq \Delta}\{f(\Delta - \lambda) + g(\lambda)\}$ is the min-plus convolution of functions f and g, and $(f \oslash g)(\Delta) = \sup_{\lambda \geq 0}\{f(\Delta + \lambda) - g(\lambda)\}$ is the min-plus deconvolution of functions f and g.

Because of (4) and (5), the accumulated computing time $Q(s, t)$ and its upper bound $\gamma(t - s)$ are monotonically increasing for any fixed s. The operation mode of a component can be expressed by the rate function, which describes the workload per time unit that a processor is processing:

$$S(t) = \frac{\mathrm{d}Q(s,t)}{\mathrm{d}t} \in [0, 1] \tag{6}$$

for any $s < t$. For instance, $S(t) = 1$ and $S(t) = 0$ denote that the processing component is fully utilized or idle, respectively.

3.2 Power Model

It is well known that the dynamic power consumption P_{dyn} growths quadratically with supply voltage v and linearly with operation frequency f, i.e., $P_{\text{dyn}} \propto v^2 \cdot f$. For the sake of simplicity, the supply voltage is assumed to be fixed value so that the model can be applied to DFS without loss of generality. With fixed v, the operation frequency f is proportional to the rate function $S(t)$. In addition, we model the temperature dependency of leakage power by means of a linear approximation [22]. Then, the total power consumption is:

$$P(t) = P_{\text{dyn}}(t) + P_{\text{leak}}(t) = \phi \cdot T(t) + \rho \cdot S(t) + \psi \tag{7}$$

where ϕ, ρ, and ψ are three constants that can be calculated by curve fitting the power consumption.

3.3 Thermal Model

A widely used duality to analyze the heat transfer of modern VLSI systems is to model the heat flow as current passing through a thermal resistance, the thermal difference as the corresponding voltage, and the heat capacity as an electrical capacity, see [23, 24]. In particular, temperature variations follow a first order linear differential equation of the form:

$$C \cdot \frac{\mathrm{d}T(t)}{\mathrm{d}t} = P(t) - G \cdot (T(t) - T_{\text{amb}}) \tag{8}$$

where C, P, G, and T_{amb} denote the thermal capacity, the generated power, the thermal conductance, and the ambient temperature, respectively. Rewriting (8) with (7) leads us to:

$$\frac{\mathrm{d}T(t)}{\mathrm{d}t} = -g \cdot T(t) + h(t) \tag{9}$$

with time-dependent temperature T,

$$g = \frac{G - \phi}{C}, \tag{10}$$

and

$$h(t) = \frac{\rho \cdot S(t) + \psi + G \cdot T_{\text{amb}}}{C}. \tag{11}$$

A closed-form solution of the above differential equation yields:

$$T(t) = T(t_0) \cdot e^{-g \cdot (t - t_0)} + \int_{t_0}^{t} e^{-g(t-\xi)} \cdot h(\xi)\, \mathrm{d}\xi. \tag{12}$$

As long as the system is processing with a constant rate $S(t) = l$, the above equation can be rewritten as:

$$T(t) = T^{\infty}_{[S(t)=l]} \cdot \left(1 - e^{-g \cdot (t - t_0)}\right) + T(t_0) \cdot e^{-g \cdot (t - t_0)} \tag{13}$$

where $T^{\infty}_{[S(t)=l]} = h_{[S(t)=l]}/g$ is the steady-state temperature in a single mode of slope $S(t) = l$ and $h_{[S(t)=l]} = \frac{\rho \cdot l + \psi + G \cdot T_{\text{amb}}}{C}$.

Throughout this article, we restrict our analysis to proper thermal models, in which the following reasonable conditions are satisfied:

- We have $g > 0$, i.e., $G > \phi$.
- The steady-state temperature is not smaller in a single mode of slope $S(t) = l$ than in a single mode of slope $S(t) = 0$, i.e., $T^{\infty}_{[S(t)=l]} \geq T^{\infty}_{[S(t)=0]}$ or $l \geq 0$.

Next, we will show that the thermal model satisfies the thermal monotonicity property.

Lemma 3.1 (Monotonicity). *Suppose that we consider two equal timed sequences of operation modes in a time interval from s to t. Then, the sequence with higher temperature at time s leads to a higher temperature at time t.*

Proof. We will prove this lemma by contradiction. Suppose that the above lemma is false, i.e., the sequence with higher temperature at time s has lower temperature at some time t. As temperature is a continuous function, there exists a time $\tilde{t}$ where the two sequences have equal temperatures. However, as both sequences have equal temperature derivations for the same temperature, the temperature at time t would be equal, as well, which contradicts the assumption. □

4 Thermal Analysis

In this section, we will first show how to construct the critical accumulated computing time $Q^*(0,t)$ that leads to the worst-case peak temperature T^*. Later, we show the tightness of this bound for a resource with full service availability and different bandwidth levels.

4.1 Worst-Case Computing Time

First, we will construct the critical accumulated computing time $Q^*(0,t)$ that leads to an upper bound on the maximum temperature T^* for a given arrival curve α and a given service curve β. As a prerequisite, we will show in Lemma 4.1 that allowing more power later in time (closer to observation time τ) will always increase the temperature $T(\tau)$ at observation time τ.

Lemma 4.1 (Shifting). *Given is a proper system model according to (7) and (9) as well as some time instance τ. In addition, we consider two mode functions $S(t)$ and $\overline{S}(t)$ defined for $t \in [0,\tau)$, which are only different in a time interval $[\sigma, \sigma + 2\delta)$ with $\sigma + 2\delta < \tau$. In particular, we have $S(t) = l_1$ for all $t \in [\sigma, \sigma + \delta)$ and $S(t) = l_2$ for all $t \in [\sigma + \delta, \sigma + 2\delta)$. $\overline{S}(t)$ is $\overline{l_1}$ for $[\sigma, \sigma + \delta)$ and $\overline{l_2}$ for $[\sigma + \delta, \sigma + 2\delta)$, where $\overline{l_1} = l_1 - \Delta$, $\overline{l_2} = l_2 + \Delta$, and $0 \leq \Delta \leq l_1$. In other words, we allow more power in the second time interval of $[\sigma, \sigma + 2\delta)$ while keeping the total power consumption the same. Then, if $\overline{T}(0) = T(0)$, we have $\overline{T}(\tau) \geq T(\tau)$, i.e., mode function $\overline{S}(\tau)$ results in a higher temperature at time τ when δ is small enough.*

Proof. As the mode functions satisfy $S(t) = \overline{S}(t)$ for all $t \in [0, \sigma)$ and $\overline{T}(0) = T(0)$, we clearly have $\overline{T}(\sigma) = T(\sigma)$. As $S(t) = l_1$ for $t \in [\sigma, \sigma + \delta)$ and $S(t) = l_2$ for $t \in [\sigma + \delta, \sigma + 2\delta)$, we find:

$$T(\sigma+2\delta) = T^{\infty}_{[S(t)=l_2]} \cdot \left(1 - e^{-g\delta}\right) + T^{\infty}_{[S(t)=l_1]} \cdot \left(1 - e^{-g\delta}\right) \cdot e^{-g\delta} + T(\sigma) \cdot e^{-2g\delta}. \quad (14)$$

Furthermore, as $\overline{S}(t) = \bar{l}_1$ for $t \in [\sigma, \sigma + \delta)$ and $\overline{S}(t) = \bar{l}_2$ for $t \in [\sigma + \delta, \sigma + 2\delta)$, we find:

$$\overline{T}(\sigma+2\delta) = T^{\infty}_{[S(t)=\bar{l}_2]} \cdot \left(1 - e^{-g\delta}\right) + T^{\infty}_{[S(t)=\bar{l}_1]} \cdot \left(1 - e^{-g\delta}\right) \cdot e^{-g\delta} + \overline{T}(\sigma) \cdot e^{-2g\delta}. \quad (15)$$

Then, for a proper thermal model, we have:

$$\overline{T}(\sigma + 2\delta) - T(\sigma + 2\delta) = \frac{\rho}{G - \phi} \cdot \Delta \cdot \left(1 - e^{-g\delta}\right)^2 \geq 0 \quad (16)$$

where we used the fact that $\Delta \geq 0$. As $S(t) = \overline{S}(t)$ for all $t \in [\sigma + 2\delta, \tau)$, we know from Lemma 3.1 that $\overline{T}(\sigma + 2\delta) \geq T(\sigma + 2\delta)$ also implies $\overline{T}(\tau) \geq T(\tau)$.□

The next lemma shows that we obtain a higher temperature at some time τ if in *any* interval ending at τ the component has larger accumulated computing time. This lemma provides the foundation for the main theorem.

Lemma 4.2 (Mode Functions Comparison). *Given is a proper thermal model as well as some time instance τ. In addition, we consider two accumulated computing time functions Q and $\overline{Q}$, which satisfy:*

$$\overline{Q}(\tau - \Delta, \tau) \geq Q(\tau - \Delta, \tau) \quad \textit{for all } 0 \leq \Delta \leq \tau. \quad (17)$$

Then, if $\overline{T}(0) = T(0)$ we have $\overline{T}(\tau) \geq T(\tau)$, i.e., mode function $\overline{S}(\tau)$ results in a higher temperature at time τ.

Proof. Because of (6), the condition of this Lemma translates equivalently to:

$$\int_{\tau-\Delta}^{\tau} \overline{S}(t)\,\mathrm{d}t \geq \int_{\tau-\Delta}^{\tau} S(t)\,\mathrm{d}t \quad \text{for all } 0 \leq \Delta \leq \tau. \quad (18)$$

Now, we will stepwise transform $S(t)$ into $\overline{S}(t)$ and in each step, the temperature will not decrease because of Lemma 4.1. In order to simplify the proof technicalities, we suppose discrete time, i.e., $S(t)$ and $\overline{S}(t)$ may change values only at multiples of δ. In other words, $S(t)$ and $\overline{S}(t)$ are constant for $t \in [k\delta, (k+1)\delta)$ for all $k \geq 0$. Let us define $\tau = k_{\max}\delta$. Now, we execute the following algorithm:

1. Determine the smallest $1 \leq k_1 \leq k_{\max}$ such that $S(\tau - k_1\delta) < \overline{S}(\tau - k_1\delta)$. If there is no such k_1, then $S(t) = \overline{S}(t)$ for all $0 \leq t \leq \tau$ and therefore, $\overline{T}(\tau) = T(\tau)$ and the algorithm stops.
2. Determine the smallest k_2 with $k_1 < k_2 \leq k_{\max}$ such that $S(\tau - k_2\delta) \neq 0$. In case k_2 does not exist, it is trivial that $\overline{T}(\tau) \geq T(\tau)$ and the algorithm stops. Otherwise,
   (a) set $\Delta = \min\{S(\tau - k_2\delta), \overline{S}(\tau - k_1\delta) - S(\tau - k_1\delta)\}$,
   (b) add Δ to $S(t)$ for $t \in [\tau - k_1\delta, \tau - (k_1 - 1)\delta)$, and
   (c) subtract Δ from $S(t)$ for $t \in [\tau - k_2\delta, \tau - (k_2 - 1)\delta)$.

3. If $S(\tau - k_1\delta) < \overline{S}(\tau - k_1\delta)$, continue with step 2. Otherwise, go to step 1.

Now, one can simply prove that, after each iteration, $T(\tau)$ does not decrease until it reaches $\overline{T}(\tau)$ and therefore, the initial $T(\tau)$ was not larger than $\overline{T}(\tau)$.□

Based on the above lemma, we will show the main result of this section. The following theorem provides a method to determine the critical accumulated computing time Q^* for any processing component whose computing power is bounded by a service curve set.

Theorem 4.3 (Critical Computing Time). *Given is a proper thermal model according to (7) and (9). The component has the computational model (4) with operation modes and power defined in (6) and (7), respectively. Then the following holds:*

- *Suppose that the accumulated computing time function $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ for all $0 \leq \Delta \leq \tau$ leads to temperature $T^*(\tau)$ at time τ. Then, $T^*(\tau)$ is an upper bound on the highest temperature $T^*(\tau) \geq T(\tau)$ for all feasible workload traces (5) that are bounded by arrival curve α and service curve β according to (3) and (4), respectively.*
- *If, in addition, $T(0) \leq T^{\infty}_{[S(t)=0]}$ holds in general, where $T^{\infty}_{[S(t)=0]}$ is the steady-state temperature of operation mode $S(t) = 0$, then for any feasible workload trace we find $T^*(\tau) \geq T(t)$ for all $0 \leq t \leq \tau$.*

Proof. At first, we show that $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ satisfies (5). We have $Q^*(t-\Delta,t) = Q^*(0,t) - Q^*(0,t-\Delta) = \gamma(\tau) - \gamma(\tau - t) - \gamma(\tau) + \gamma(\tau - t + \Delta) = \gamma(\tau - t + \Delta) - \gamma(\tau - t)$.

Now, we will show the first item of the theorem by contradiction. Suppose that there is an accumulated computing time function Q, which leads to a higher temperature $T(\tau)$ at time τ. Then, according to Lemma 4.2, there exists some $\Delta \leq \tau$ such that $Q^*(\tau - \Delta, \tau) < Q(\tau - \Delta, \tau)$. As we know that $Q^*(\tau - \Delta, \tau) = \gamma(\Delta) - \gamma(0) = \gamma(\Delta)$ we can conclude that $Q(\tau - \Delta, \tau) > \gamma(\Delta)$ and therefore, Q violates (5).

Let us prove the second item of the theorem by contradiction. Suppose that there exists some time $\sigma \leq \tau$ where we have $T(\sigma) > T^*(\tau)$. We know from the first item in this theorem that the bound on $T(\sigma)$ is maximized if $Q(\sigma - \Delta, \sigma) = \gamma(\Delta)$ for $0 \leq \Delta \leq \sigma$. As $Q^*(\tau - \Delta, \tau) = \gamma(\Delta)$, we can conclude that Q^* shifted by $\tau - \sigma$ and Q are identical, i.e., we have $Q(\sigma - \Delta, \sigma) = Q^*(\tau - \Delta, \tau)$ for $0 \leq \Delta \leq \sigma$. Therefore, if we would set $T(0) = T^*(\tau - \sigma)$, we have $T(\sigma) \leq T^*(\tau)$. Because of thermal monotonicity, $T(\sigma) > T^*(\tau)$ would require that $T(0) > T^*(\tau - \sigma)$.

As $T(0) \leq T^{\infty}_{[S(t)=0]}$ and $T^{\infty}_{[S(t)=0]} \leq T^{\infty}_{[S(t)\geq 0]}$, temperatures which are caused by Q^* satisfy $T^*(t) \geq T^*(0)$ for all times $0 \leq t \leq \tau$. This holds in particular for time $t = \tau - \sigma$, i.e., $T^*(\tau - \sigma) \geq T^*(0)$. As we have the same initial conditions for both scenarios with $T(0) = T^*(0)$, we find $T(0) \leq T^*(\tau - \sigma)$, which is a contradiction. □

Note that the worst-case scenario for the temperature is often completely different from the conventional critical instance scenario that is used in real-time analysis to determine the worst-case timing behaviour. For example, for periodic tasks with jitter, the worst-case scenario for temperature is to first warm up the system with periodic arrivals and then heat up the system with burst

arrivals and jitters. Moreover, because of (2), the result of Theorem 4.3 is also valid for a processor that concurrently processes a set of independent tasks.

The above outlined thermal analysis method can be extended to a chip with multiple power sources. In that case, the effect of a power source onto the component temperature has to be described by an impulse response. The impulse response only depends on the floor plan of the chip, but not on the actual workload. Then, Lemmas 4.1 and 4.2 have to be extended so that they do not only hold for a system with one power source, but also for an impulse response of a thermal RC circuit with multiple power sources. As a first step towards multiple power sources, the analysis method has been extended in [18] to a system with multiple power sources, but full service availability.

4.2 Tightness

Theorem 4.3 provides a method to calculate an upper bound T^* on the actual worst-case peak temperature of a real-time system. However, there might be no valid workload trace $R^*(0,\Delta)$ for $0 \leq \Delta \leq \tau$ that complies with the given arrival curve α and results in the critical accumulated computing time $Q^*(0,\Delta)$. Next, we will show that this is not the case for a resource with full service availability and different bandwidth levels B, i.e., $W(s,t) = B \cdot (t-s)$ with $0 \leq B \leq 1$.

Theorem 4.4 (Critical Accumulated Workload). *Suppose that the assumptions from Theorem 4.3 hold and the resource availability satisfies* $W(s,t) = B \cdot (t-s)$ *with* $0 \leq B \leq 1$ *and* $s \leq t$. *Furthermore, assume that the given arrival curve* α *satisfies* $\alpha(\Delta) = c \cdot \left\lceil \frac{1}{c} \cdot \alpha(\Delta) \right\rceil$ *for all* $\Delta \geq 0$, *i.e., the step size of* α *is an integer multiple of* c. *Then, the critical accumulated workload* $R^*(0,\Delta) = c \cdot \left\lceil \frac{1}{c} \cdot Q^*(0,\Delta) \right\rceil$

- *compiles with arrival curve* α *according to (1), and*
- *leads to the accumulated computing time* $Q^*(0,\Delta)$.

Proof. Without loss of generality, we suppose that $c = 1$. For the first item, we have to show that $R^*(s,t) \leq \alpha(t-s)$. From $W(s,t) = B \cdot (t-s)$ follows that $\beta(\Delta) = \beta^l(\Delta) = \beta^u(\Delta) = B \cdot \Delta$ for all $\Delta \geq 0$. This simplifies the computational model (5) to

$$Q(t-\Delta,t) \leq \gamma(\Delta) = (\alpha \otimes \beta)(\Delta) = \inf_{0 \leq \lambda \leq \Delta} \{\alpha(\Delta-\lambda) + B \cdot \lambda\}. \tag{19}$$

Then, we note that $\gamma(\Delta) \leq \alpha(\Delta)$ for all $\Delta \geq 0$. Now, we find $R^*(s,t) = R^*(0,t) - R^*(0,s) = \lceil Q^*(0,t) \rceil - \lceil Q^*(0,s) \rceil \leq \lceil Q^*(0,t) - Q^*(0,s) \rceil = \lceil Q^*(s,t) \rceil \leq \lceil \gamma(t-s) \rceil \leq \lceil \alpha(t-s) \rceil = \alpha(t-s)$ for all $s < t$.

Next, we show that the continuous workload function $\widehat{R}^*(0,\Delta) = Q^*(0,\Delta)$ leads to $Q^*(0,\Delta)$. As $\widehat{R}^*(0,\Delta) = Q^*(0,\Delta)$, we have to prove that $Q^*(0,\Delta) = \inf_{0 \leq u \leq \Delta} \{W(0,\Delta) - W(0,u) + Q^*(0,u)\}$. Because there exists a u' such that $W(0,\Delta) - W(0,u') + Q^*(0,u') = Q^*(0,\Delta)$, namely $u' = \Delta$, we only have to show that $W(0,\Delta) - W(0,u) + Q^*(0,u) \geq Q^*(0,\Delta)$ for all $0 \leq u \leq \Delta$. However, this condition is equivalent to $W(u,\Delta) \geq Q^*(u,\Delta)$ for all $0 \leq u \leq \Delta$. As the accumulated computing time in interval $[u,\Delta)$ cannot exceed the available time $W(u,\Delta)$, $\widehat{R}^*(0,\Delta)$ leads to $Q^*(0,\Delta)$. In order to show the second item, we have to prove that $R^* = \lceil Q^*(0,\Delta) \rceil$ leads to $Q^*(0,\Delta)$. From (19)

follows that $\gamma(\Delta)$ has either slope b or 0 and it has an integer value if its slope is 0. Consequently, if $\gamma(\tau)$ is integer, $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ has the same properties as $\gamma(\Delta)$ and we can deduce that $\inf_{0\le u\le\Delta}\{\Delta - u + Q^*(0,u)\} = \inf_{0\le u\le\Delta}\{\Delta - u + \lceil Q^*(0,u)\rceil\} = \inf_{0\le u\le\Delta}\{\Delta - u + R^*(0,u)\}$. □

As a result of Theorem 4.4, the upper bound T^* determined by Theorem 4.3 is tight under the conditions mentioned in Theorem 4.4, i.e., there exists a workload trace that leads to the critical accumulated computing time. Moreover, the tightness of the upper bound T^* is not restricted to resources with full service availability and different bandwidth levels. Once the service curves are known, the tightness can be proven by using the approach of Theorem 4.4 to show that the workload trace $R^*(0,\Delta)$ results in the critical accumulated computing time $Q^*(0,\Delta)$. Then, the remaining task is to prove that the workload trace $R^*(0,\Delta)$ complies with the given arrival curve α.

5 Optimal Resource Model

In this section, we use the result of Theorem 4.3 to calculate the resource availability, that optimally reduces the peak temperature for a given set of tasks, each characterized by an arrival curve and a relative deadline. We first consider a single stream of events with relative deadline D, i.e., each event has to complete its execution within D time units after its arrival. Afterwards, we extend the approach to a set of tasks with different arrival curves and deadlines, scheduled under EDF.

5.1 Schedulability Analysis

The maximum delay $d_{\max}$ experienced by any event on the event stream can be upper-bounded by [25]:

$$d_{\max} \le \sup_{\lambda\ge 0}\left\{\inf\left\{u \ge 0 : \alpha(\lambda) \le \beta^l(\lambda + u)\right\}\right\}. \tag{20}$$

In other words, the maximum delay is the maximum horizontal distance between the upper arrival curve $\alpha(\Delta)$ and the lower service curve $\beta^l(\Delta)$, i.e., the minimum available computing time in any time interval. Now, we can formulate the following scheduling constraint for a stream of events with relative deadline D:

$$\alpha(\Delta - D) \le \beta^l(\Delta) \quad \forall\Delta \ge 0. \tag{21}$$

5.2 Optimal Resource Availability

As higher uncertainty in resource availabilities results in higher peak temperatures, a necessary condition to optimally reduce the peak temperature is that the upper bound and lower bound service curves are identical, i.e., $\beta^l(\Delta) = \beta^r(\Delta) = \beta(\Delta)$ for all $\Delta \ge 0$, which means that there is no uncertainty in resource availability. Moreover, from the characteristics of the time interval domain follows that such a service curve has to be sub-additive, i.e.,

$\beta(s) + \beta(t) \geq \beta(s+t)$. The first lemma shows that we always obtain a lower temperature if we reduce the service curve.

Lemma 5.1 (Service Curve Comparison). *Given are two processors Θ_1 and Θ_2 with resource availabilities β_1 and β_2, respectively, which satisfy $\beta_1(\Delta) \leq \beta_2(\Delta)$ for all $\Delta \geq 0$. When the workload of both processors is upper bounded by the same arrival curve α, the peak temperature of processor Θ_2 is not less than the peak temperature of processor Θ_1.*

Proof. The condition $\beta^l(\Delta) = \beta^r(\Delta) = \beta(\Delta)$ for all $\Delta \geq 0$ simplifies the computational model (5) to:

$$Q(t-\Delta, t) \leq \gamma(\Delta) = (\alpha \otimes \beta)(\Delta) = \inf_{0 \leq \lambda \leq \Delta} \{\alpha(\Delta - \lambda) + \beta(\lambda)\}. \tag{22}$$

Suppose that $\gamma_1(\Delta) = (\alpha \otimes \beta_1)(\Delta)$ and $\gamma_2(\Delta) = (\alpha \otimes \beta_2)(\Delta)$, then we know from (22) that $\beta_1(\Delta) \leq \beta_2(\Delta)$ for all $\Delta \geq 0$ implies $\gamma_1(\Delta) \leq \gamma_2(\Delta)$ for all $\Delta \geq 0$. Now, let us prove the lemma by contradiction. Suppose that there exists a $\beta_1(\Delta) \leq \beta_2(\Delta)$ for all $\Delta \geq 0$, which leads to a higher peak temperature. Then, according to Lemma 4.2, there exists some $\Delta \leq \tau$ such that $Q_1^*(\tau - \Delta, \tau) > Q_2^*(\tau - \Delta, \tau)$. As we know that $Q^*(\tau - \Delta, \tau) = \gamma(\Delta) - \gamma(0) = \gamma(\Delta)$, we can conclude that $\gamma_1(\Delta) > \gamma_2(\Delta)$ for a certain Δ, which is a contradiction. □

Next, we will show that there is no service curve β that guarantees all deadlines and leads to a lower worst-case peak temperature than service curve β_{opt}, where $\beta_{\text{opt}}(\Delta) = ConvexHull(\alpha(\Delta - D))$ is the upper convex hull function of arrival curve α shifted by the relative deadline D.

Theorem 5.2 (Optimal Resource Availability). *Given is a proper thermal model according to (7) and (9). The component has the computational model (4) with the sub-additive service curve $\beta^u(\Delta) = \beta^l(\Delta) = \beta(\Delta)$ for all $\Delta \geq 0$. Operation modes and power are defined as in (6) and (7), respectively. Amongst all resource availabilities satisfying the schedulability constraint, the resource availability $\beta_{opt}(\Delta) = ConvexHull\,(\alpha(\Delta - D))$ for all $\Delta \geq 0$ optimally reduces the highest temperature of the processor.*

Proof. At first, we show that β_{opt} satisfies the schedulability constraint (21). From the definition of the upper convex hull follows that $\beta_{\text{opt}}(\Delta) = ConvexHull\,(\alpha(\Delta - D)) \geq \alpha(\Delta - D)$ and therefore, β_{opt} satisfies (21).

Next, let us prove by contradiction that there is no other service curve that has a lower peak temperature and, at the same time, satisfies the schedulability constraint. Suppose that there exists such a service curve $\beta'(\Delta)$. As it satisfies the schedulability constraints, $\beta'(\Delta) \geq \alpha(\Delta - D)$ and, since $\beta'(\Delta)$ leads to a lower temperature than $\beta_{\text{opt}}(\Delta)$, $\beta_{\text{opt}}(\Delta) \not\leq \beta'(\Delta)$ for all $\Delta \geq 0$, i.e., there exists a $\tilde{\Delta}$ such that $\beta_{\text{opt}}(\tilde{\Delta}) > \beta'(\tilde{\Delta})$. However, as $\beta_{\text{opt}}(\Delta)$ is the upper convex hull of $\alpha(\Delta - D)$, $\beta'(\Delta)$ is not sub-additive. □

The obtained result is similar to temperature reduction with leaky-bucket shapers as presented in [17]. However, while shapers are mainly a software solution that delays the execution of events to reduce the peak temperature, our approach is a mathematical indicator for the quality of a certain resource model with respect to worst-case peak temperature. In particular, it is impossible to design a resource model that leads to a lower peak temperature and guarantees all real-time deadlines.

5.3 Scheduling of Multiple Tasks

Next, we extend Theorem 5.2 to a set of tasks with different arrival curves and deadlines, scheduled under EDF. Suppose that ν_i denotes the i-th task within our task set. Every task ν_i is characterized by an arrival curve α_i and a relative deadline D_i. Then, we apply the concept of demand bound functions (DBF) [26] for schedulability tests, and obtain the DBF dbf as:

$$\mathrm{dbf}(\Delta) = \sum_{\forall i} \alpha_i(\Delta - D_i) \quad \text{for all } \Delta \geq 0 \tag{23}$$

and the optimal service curve by means of the DBF as:

$$\beta_{\mathrm{opt}}(\Delta) = ConvexHull(\mathrm{dbf}(\Delta)). \tag{24}$$

To summarize, in order to calculate the minimum peak temperature for a system with EDF policy, we calculate the optimal service curve β_{opt} that minimizes the peak temperature and still guarantees all deadlines. The procedure to be applied by system designers is as follows: First, determine the DBF of the system. Afterwards, β_{opt} is calculated as the upper convex hull of the DBF.

The proposed procedure can be used by system designers as an indicator for a qualitative assessment of their resource model with respect to maximum temperature. In particular, they can assess if it is worth investigating in a better, but probably more expensive, resource model to reduce the peak temperature.

6 Experimental Analysis

In this section, we compare the worst-case peak temperature results with the temperature of randomly generated workload traces and analyze the effect of reduced resource availability on the worst-case peak temperature. Finally, we provide hints on how to design a system, which is schedulable and meets temperature constraints at the same time. To this end, the MPA framework [27] that analyzes schedulability of real-time systems, is extended with the ability to calculate the worst-case peak temperature by means of the thermal analysis method proposed in Section 4.

6.1 Experimental Setup

Two different example applications executing on a uniprocessor system are evaluated in this section. The first example application consists of a single task with an invocation period of 200 ms and an execution demand of 50 ms. The second example is a multi-process video conferencing application, which includes a video codec, an audio codec, and a network process for communication management. For illustration purpose, we use the period-jitter-delay model with parameters summarized in Table 1. If not stated otherwise, the maximum jitter of the single task application and the video task is 20 ms. EDF policy is used to arbitrate between the processes. The system parameters are borrowed from [3] and

Table 1: Parameters of the video conferencing application.

	Video	Audio	Network
invocation period	50 ms	30 ms	30 ms
jitter	[20, 90]ms	10 ms	10 ms
minimum interarrival	1 ms	1 ms	1 ms
execution demand	6 ms	3 ms	2 ms
deadline	50 ms	30 ms	30 ms

Table 2: Thermal and power parameters of the considered example system architecture.

G	C	ϕ	ρ	ψ
$0.3\,\frac{\mathrm{W}}{\mathrm{K}}$	$0.03\,\frac{\mathrm{J}}{\mathrm{K}}$	$0.1\,\frac{\mathrm{W}}{\mathrm{K}}$	14.0 W	−25.0 W

summarized in Table 2. In all our experiments, we start simulation with initial temperature $T(0) = T^{\infty}_{[S(t)=0]} = 325\,\mathrm{K}$, calculated from the parameters given in Table 2. The observation time τ is set to 1.0 s in all experiments.

6.2 Peak Temperature Analysis

At first, we compare the transient temperature evolution of the thermal critical instance with the temperature of the timing critical instance and the temperature of 100 randomly generated workload traces. The thermal critical instance is the trace that leads to the worst-case peak temperature and the timing critical instance is the trace that releases the workload as early as possible. For this experiment, we assume a resource with full service availability.

Figure 2 presents the results of the transient temperature evolution for the video conferencing application with a jitter equals to 50 ms. The timing critical instance heats up the platform at the beginning leading to a peak temperature of 346.83 K. The 100 randomly generated workload traces might heat up the system the most when multiple frames arrive together. In particular, the highest observed temperature is 349.10 K for all traces. Finally, the thermal critical instance places the burst arrivals and jitters at the end around the observation time $\tau = 1\,\mathrm{s}$ so that a worst-case peak temperature of 350.39 K is observed.

6.3 Temperature Analysis for Different Resource Models

The proposed peak temperature analysis framework is applied to four different resource models, namely full resource availability with frequency modulation, availability with bounded-delay, availability based on time division multiple access (TDMA), and periodic availability. Besides peak temperature, the schedulability is analyzed by means of MPA and non-schedulable configurations are marked.

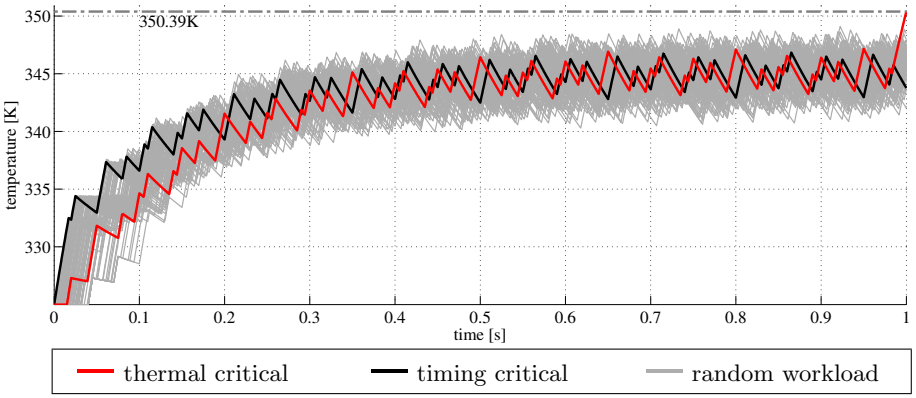
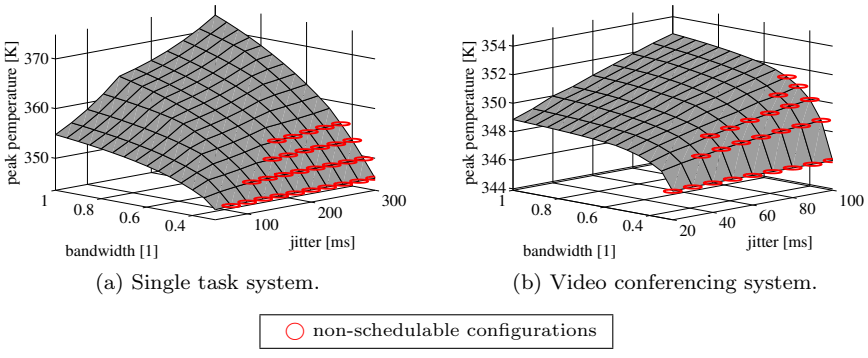


Figure 2: Transient temperature evolution of the thermal critical instance, the timing critical instance, and 100 randomly generated workload traces for the video conferencing application.

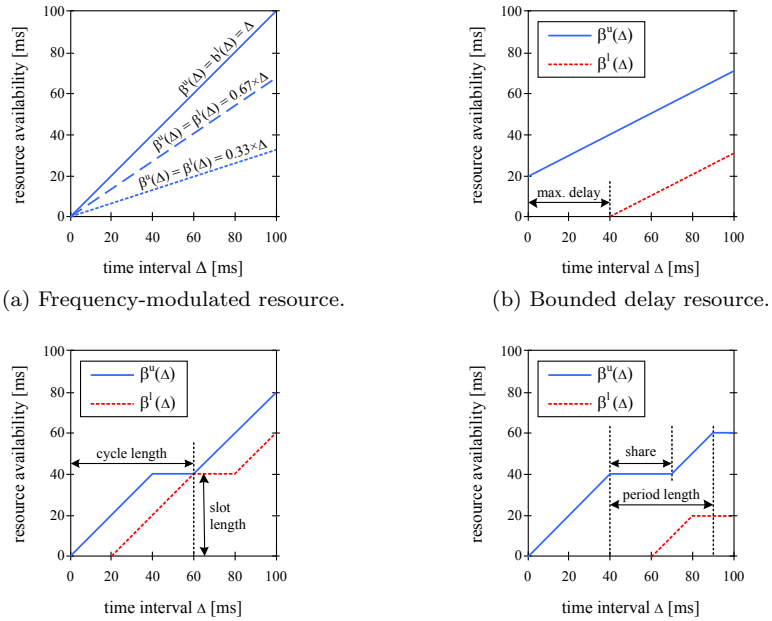
Resource with Frequency Modulation. In the first experiment, we examine a resource with full availability at different bandwidth levels in order to study the effect of frequency scaling on peak temperature. Figure 4a shows several service curves for different bandwidth levels B. The full service curve ($\beta(\Delta) = \Delta$) corresponds to full bandwidth utilization, i.e., the system is running at full operation frequency. The two partially utilized service curves ($\beta(\Delta) = 0.67\Delta$ and $\beta(\Delta) = 0.33\Delta$) correspond to a bandwidth utilization of 0.67 and 0.33, i.e., the system is running at an operation frequency of two-thirds and one-third, respectively. Figures 3a and 3b outline the worst-case peak temperature as a function of bandwidth level and jitter for single task and video conferencing application. For the single task application with a jitter of 50 ms, a reduction of the operation frequency by 50 % lowers the peak temperature by 4.23 K. Similarly, for a jitter of 300 ms, reducing the operation frequency by half lowers the peak temperature by 14.5 K. To interpret these results, we note that the steady-state temperature is independent of the operation frequency as frequency scales



(a) Single task system.

(b) Video conferencing system.

Figure 3: Worst-case peak temperature for a frequency-modulated resource as a function of both bandwidth and jitter.



(a) Frequency-modulated resource.

(b) Bounded delay resource.

(c) TDMA resource.

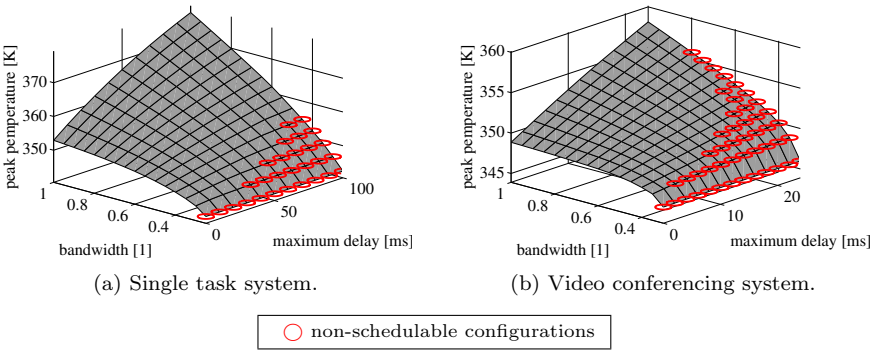
(d) Periodic resource.

Figure 4: Service curves for various resource models.

linearly with power consumption and inverse linearly with the execution time of a process. If a workload burst arrives, the component is continuously processing and the temperature increases towards $T^{\infty}_{[S(t)=l]}$, which in turn depends on the operation frequency.

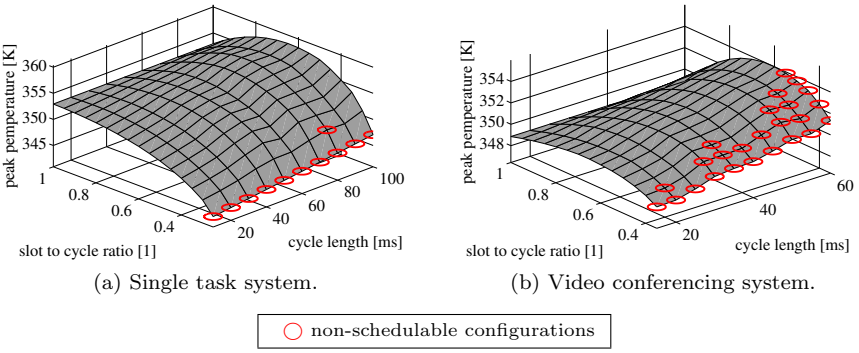
Resource with Bounded-Delay. A resource with bounded-delay generalizes the principle of a frequency-modulated resource by eliminating the need for full resource availability. To this end, a maximum delay d is introduced that models the availability. Figure 4b plots the service curve set for a resource with $d = 40\,\text{ms}$ and a bandwidth of 50 %. Figure 5 outlines the worst-case peak temperature as a function of bandwidth and maximum delay. The figure shows that the higher the maximum delay, the higher the peak temperature. Increasing the maximum delay leads to a higher uncertainty in resource availability so that the workload is buffered in the arrival queue and might cause a workload burst later. Moreover, the case study shows that bandwidth reduction might lead to non-schedulability for large maximum delays. Therefore, lowering peak temperature by bandwidth reduction is only applicable to systems with small maximum delay.

TDMA. A widely proposed method for temperature reduction is to place idle service intervals so that only a certain amount of time units are assigned to an application for computation. Then, in all other time units, the processor is forced to be in idle mode to reduce its temperature. This resource availability can be described by a TDMA scheme, where the processor is always available for S consecutive time units within every cycle C. We call C the cycle length and S the slot length. Figure 4c outlines a service curve set for a TDMA resource with



(a) Single task system. (b) Video conferencing system.

Figure 5: Worst-case peak temperature for a resource with bounded-delay as a function of both bandwidth and maximum delay.

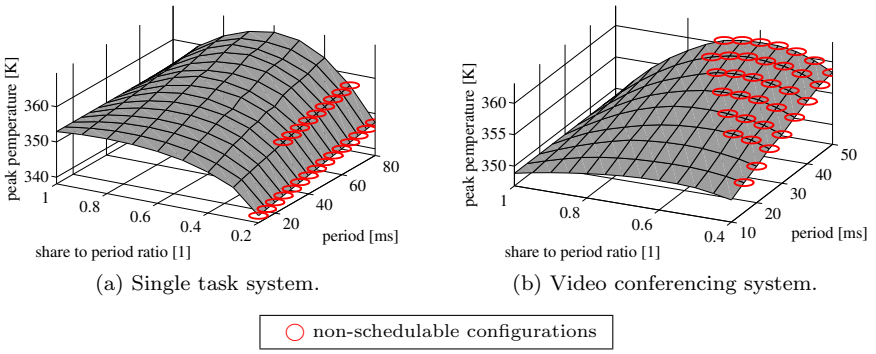


(a) Single task system. (b) Video conferencing system.

Figure 6: Worst-case peak temperature for a TDMA resource as a function of cycle length and slot to cycle ratio.

$C = 60\,\text{ms}$ and $S = 40\,\text{ms}$. The worst-case peak temperature dependency on the cycle length and slot to cycle ratio is slightly counterintuitive as longer idle intervals do not necessarily lead to a lower peak temperature as seen in Fig. 6. This characteristic can be explained by the buffering effect, as well. In case the resource is not fully available, workload that arrives during idle intervals is buffered in the queue and causes a workload burst later, leading to an increase of the peak temperature.

Periodic Availability. Last, we examine a periodically available resource as introduced in [20]. In particular, a share of s time units is allocated within every period of p time units. Similar to TDMA resources, the processor is supposed to be in idle mode in all other time units. Figure 4d outlines the service curve set for a periodically available resource with $s = 30\,\text{ms}$ and $p = 50\,\text{ms}$. The case study shows that a resource with periodic availability has similar peak temperature characteristics as a TDMA resource. The worst-case peak temperature is outlined in Fig. 7 as a function of period and share to period ratio. Again, assigning a shorter share does not necessarily lead to lower peak



(a) Single task system.

(b) Video conferencing system.

Figure 7: Worst-case peak temperature for a periodically available resource as a function of period and share to period ratio.

temperatures due to the same reasons as described for a TDMA resource.

6.4 Discussion

In Tables 3a and 3b, the peak temperatures of the previously discussed resource models are compared with the minimum possible peak temperature. The optimum peak temperature is calculated by means of the method presented in Section 5. Out of all resource configurations that have been previously evaluated, we only consider configurations that reduce the peak temperature the most and do not miss real-time deadlines. The optimal resource allocation model can be used as a guideline to optimize the parameters of a specific resource model. Even though there is only a minor difference between all peak temperature values, frequency modulation leads to the lowest peak temperature among all resource models. As the periodic resource model has the highest uncertainty in terms of resource availability, peak temperature reduction by means of a periodic resource is not as efficient as with the other evaluated resource models.

Besides guaranteeing that the system never exceeds a certain peak temperature, a second advantage of the proposed method is the evaluation time. On an Intel Core i7 M 620 processor, calculating the worst-case peak temperature took between 14.1 ms and 144.5 ms for the single task system and between 29.8 ms and 53.6 ms for the video conferencing system. Calculating the worst-case peak temperature for TDMA and periodically available resources takes the most time as computing $\gamma(\Delta)$ by means of (22) is more expensive than for resource models with service curves that are composed of one or two straight lines.

Even if it is not studied in this article, another scope of the proposed method is design space exploration. Consider, for example, a frequency-modulated resource with a certain critical temperature, i.e., exceeding this temperature might lead to functional errors. As tens of design candidates can be analyzed in one second, the proposed framework enables the identification of the optimal bandwidth that minimizes the latency amongst all bandwidths with a worst-case peak temperature lower than the critical temperature. As the proposed al-

Table 3: Peak temperature and evaluation time comparison.

(a) Single task system.

Resource model	Evaluation time	Peak temperature
frequency modulation ($B = 0.3$)	14.1 ms	344.8 K
bounded delay ($d = 5$ ms, $B = 0.3$)	15.6 ms	345.1 K
TDMA ($c = 10$ ms, $s = 3$ ms)	123.1 ms	345.3 K
periodic resource ($p = 10$ ms, $s = 3$ ms)	144.5 ms	346.7 K
optimum peak temperature	87.5 ms	343.3 K

(b) Video conferencing system.

Resource model	Evaluation time	Peak temperature
frequency modulation ($B = 0.4$)	29.8 ms	347.6 K
bounded delay ($d = 5$ ms, $B = 0.4$)	30.8 ms	348.2 K
TDMA ($c = 15$ ms, $s = 6$ ms)	44.9 ms	349.0 K
periodic resource ($p = 10$ ms, $s = 4$ ms)	53.6 ms	350.4 K
optimum peak temperature	51.0 ms	346.5 K

gorithm offers safe bounds, the system can safely execute this design without further involving other (dynamic) thermal management strategies.

7 Conclusion

In this article, we presented an analytic approach to determine the worst-case peak temperature of real-time systems with non-deterministic workload and general resource availability. Amongst others, the presented framework is able to deal with frequency modulated resources and temperature-dependent leakage power. The event and resource models are based on real-time calculus so that the framework is able to handle a broad range of non-determinism in terms of unknown jitter, burst, and provided computation. In particular, the accumulated workload arrival from all task invocations is characterized by an arrival curve, i.e., by an upper bound on the sum of task execution times arriving in any time interval. Similarly, we model the resource availability by service curves. By applying the proposed method to various resource models, we studied the effect of frequency modulation and periodic availability on the worst-case peak temperature.

Acknowledgments

This work was supported by EU FP7 projects EURETILE and PRO3D, under grant numbers 247846 and 249776.

References

[1] Hardavellas, N., Ferdman, M., Falsafi, B., and Ailamaki, A.: 'Toward Dark Silicon in Servers', *IEEE Micro*, 2011, 31, (4), pp. 6–15.

[2] Chandrakasan, A., Sheng, S., and Brodersen, R.: 'Low-Power CMOS Digital Design', *IEEE J. Solid-St. Circ.*, 1992, 27, (4), pp. 473–484.

[3] Rai, D., Yang, H., Bacivarov, I., Chen, J.-J., and Thiele, L.: 'Worst-Case Temperature Analysis for Real-Time Systems', *Proc. Design, Automation and Test in Europe (DATE)*, Grenoble, France, 2011, pp. 1–6.

[4] Thiele, L., Chakraborty, S., and Naedele, M.: 'Real-Time Calculus for Scheduling Hard Real-Time Systems', *Proc. IEEE Int. Symposium on Circuits and Systems (ISCAS)*, Geneva, Switzerland, 2000, pp. 101–104.

[5] Huang, K., Haid, W., Bacivarov, I., Keller, M., and Thiele, L.: 'Embedding Formal Performance Analysis into the Design Cycle of MPSoCs for Real-time Streaming Applications', *ACM Trans. Embedd. Comput. Syst.*, 2012, 11, (1), pp. 8:1–8:23.

[6] Schor, L., Yang, H., Bacivarov, I., and Thiele, L.: 'Worst-Case Temperature Analysis for Different Resource Availabilities: A Case Study', *Proc. Int. Workshop on Power and Timing Modeling, Optimization, and Simulation (PATMOS)*, volume 6951 of *LNCS*, Springer, 2011, pp. 288–297.

[7] Chantem, T., Hu, X. S., and Dick, R. P.: 'Online Work Maximization under a Peak Temperature Constraint', *Proc. Int. Symposium on Low Power Electronics and Design (ISLPED)*, San Fancisco, CA, USA, 2009, pp. 105–110.

[8] Murali, S., Mutapcic, A., Atienza, D., Gupta, R., Boyd, S., and De Micheli, G.: 'Temperature-Aware Processor Frequency Assignment for MPSoCs using Convex Optimization', *Proc. Int. Conf. on Hardware/Software Codesign and System Synthesis (CODES+ISSS)*, Salzburg, Austria, 2007, pp. 111–116.

[9] Donald, J., and Martonosi, M.: 'Techniques for Multicore Thermal Management: Classification and New Exploration', *Proc. Int. Symposium on Computer Architecture (ISCA)*, Boston, Massachusetts, USA, 2006, pp. 78–88.

[10] Coskun, A. K., Rosing, T. S., and Whisnant, K.: 'Temperature Aware Task Scheduling in MPSoCs', *Proc. Design, Automation and Test in Europe (DATE)*, Nice, France, 2007, pp. 1659–1664.

[11] Bansal, N., and Pruhs, K.: 'Speed Scaling to Manage Temperature', *Proc. Symposium on Theoretical Aspects of Computer Science (STACS)*, volume 3404 of *LNCS*, Springer, 2005, pp. 460–471.

[12] Xie, Y., and Hung, W.-l.: 'Temperature-Aware Task Allocation and Scheduling for Embedded Multiprocessor Systems-on-Chip (MPSoC) Design', *The Journal of VLSI Signal Processing*, 2006, 45, (3), pp. 177–189.

[13] Bartolini, A., Cacciari, M., Tilli, A., Benini, L., and Gries, M.: 'A Virtual Platform Environment for Exploring Power, Thermal and Reliability Management Control Strategies in High-Performance Multicores', *Proc. Great Lakes Symposium on VLSI (GLSVLSI)*, Providence, Rhode Island, USA, 2010, pp. 311–316.

[14] Thiele, L., Schor, L., Yang, H., and Bacivarov, I.: 'Thermal-Aware System Analysis and Software Synthesis for Embedded Multi-Processors', *Proc. Design Automation Conference (DAC)*, San Diego, CA, USA, 2011, pp. 268–273.

[15] Garcia del Valle, P., and Atienza, D.: 'Emulation-Based Transient Thermal Modeling of 2D/3D Systems-on-Chip with Active Cooling', *Microelectronics Journal*, 2010, 41, (10), pp. 1–9.

[16] Huang, W., Ghosh, S., Velusamy, S., Sankaranarayanan, K., Skadron, K., and Stan, M.: 'HotSpot: A Compact Thermal Modeling Methodology for Early-Stage VLSI Design', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2006, 14, (5), pp. 501–513.

[17] Kumar, P., and Thiele, L.: 'Cool Shapers: Shaping Real-Time Tasks for Improved Thermal Guarantees', *Proc. Design Automation Conference (DAC)*, San Diego, CA, USA, 2011, pp. 468–473.

[18] Schor, L., Bacivarov, I., Yang, H., and Thiele, L.: 'Worst-Case Temperature Guarantees for Real-Time Applications on Multi-Core Systems', *Proc. IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, IEEE, Beijing, China, 2012, pp. 87–96.

[19] Henia, R., Hamann, A., Jersak, M., Racu, R., Richter, K., and Ernst, R.: 'System Level Performance Analysis - The SymTA/S Approach', *IEE Proc. Computers and Digital Techniques*, 2005, 152, (2), pp. 148–166.

[20] Shin, I., and Lee, I.: 'Periodic Resource Model for Compositional Real-Time Guarantees', *Proc. Real-Time Systems Symposium (RTSS)*, Cancun, Mexico, 2003, pp. 2–13.

[21] Wandeler, E., Maxiaguine, A., and Thiele, L.: 'Performance Analysis of Greedy Shapers in Real-Time Systems', *Proc. Design, Automation and Test in Europe (DATE)*, Munich, Germany, 2006, pp. 444–449.

[22] Liu, Y., Dick, R. P., Shang, L., and Yang, H.: 'Accurate Temperature-Dependent Integrated Circuit Leakage Power Estimation is Easy', *Proc. Design, Automation and Test in Europe (DATE)*, Nice, France, 2007, pp. 1526–1531.

[23] Krum, A.: 'Thermal Management', Kreith, F. (Ed.): *The CRC Handbook of Thermal Engineering*, pp. 1–92, CRC Press, 2000.

[24] Skadron, K., Stan, M. R., Sankaranarayanan, K., Huang, W., Velusamy, S., and Tarjan, D.: 'Temperature-Aware Microarchitecture: Modeling and Implementation', *ACM Trans. Architec. Code Optim.*, 2004, 1, (1), pp. 94–125.

[25] Le Boudec, J.-Y., and Thiran, P.: 'Network Calculus: A Theory of Deterministic Queuing Systems for the Internet', volume 2050 of *LNCS*, Springer, 2001.

[26] Serreli, N., Lipari, G., and Bini, E.: 'The Demand Bound Function Interface of Distributed Sporadic Pipelines of Tasks Scheduled by EDF', *Proc. Euromicro Conf. on Real-Time Systems (ECRTS)*, Brussels, Belgium, 2010, pp. 187–196.

[27] Wandeler, E., Thiele, L., Verhoef, M., and Lieverse, P.: 'System Architecture Evaluation using Modular Performance Analysis: A Case Study', *Int. J. on Software Tools for Technology Transfer (STTT)*, 2006, 8, pp. 649–667.