Worst-Case Temperature Analysis for Real-Time Systems

Devendra Rai, Hoeseok Yang, Iuliana Bacivarov, Jian-Jia Chen*, Lothar Thiele

Swiss Federal Institute of Technology (ETH), Zurich, Switzerland
*Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
firstname.lastname@tik.ee.ethz.ch

***Abstract*—With the evolution of today's semiconductor technology, chip temperature increases rapidly mainly due to the growth in power density. For modern embedded real-time systems, it is crucial to estimate maximal temperatures in order to take mapping or other design decisions to avoid burnout, and still be able to guarantee meeting real-time constraints. This paper provides answers to the question: When *work-conserving* scheduling algorithms, such as earliest-deadline-first (EDF), rate-monotonic (RM), deadline-monotonic (DM), are applied, what is the worst-case peak temperature of a real-time embedded system under all possible scenarios of task executions? We propose an analytic framework, which considers a general event model based on network and real-time calculus. This analysis framework has the capability to handle a broad range of uncertainties in terms of task execution times, task invocation periods, and jitter in task arrivals. Simulations show that our framework is a cornerstone to design real-time systems that have guarantees on both schedulability and maximal temperatures.**

***Keywords*-real-time systems; compositional analysis; worst-case peak temperature; thermal analysis**

I. Introduction

Power density has been continuously increasing in modern processors, leading to high on-chip temperatures. A system could fail if the operating temperature exceeds a certain threshold, leading to low reliability and even chip burnout. Therefore, at design time, temperature reduction has been achieved through appropriate packaging and active heat dissipation mechanisms. Moreover, at run time, dynamic thermal management (DTM) can be adopted to enhance system reliability and reduce packaging costs. A large area of research investigates DTM mechanisms, e.g., [1], [10], based on dynamic voltage/frequency scaling (DVFS), to reduce system power consumption and therefore, to prevent the system from overheating.

There have been many results in recent years about thermal management, including (1) thermal-constrained scheduling to maximize performance [1], [3], [4], [8], [9], [14], [18] or determine the schedulability of real-time systems under given temperature constraints [5], (2) peak temperature reduction to meet performance constraints [2], and (3) thermal control by applying control theory for system adaption [15].

Specifically, for thermal-constrained scheduling, in [1], an algorithm is developed to maximize the workload that can complete in a specified time window without violating thermal constraints. Formal analysis is later adopted in [5] to analyze schedulability of real-time tasks under thermal constraints. In [18], approximation algorithms are developed to minimize the completion time, while each task is restricted to execute at one speed. In [4], workload maximization is explored for discrete DVS speeds, under thermal constraints. Moreover, for reducing the peak temperature in the system in [2], the on-line algorithm for energy efficiency proposed in [17] is adopted. To model uncertainties of task execution, ambient temperature changes, or power variations, control theoretical approaches have been adopted, e.g., in [15].

Considering the available results on thermal management, one of the most fundamental questions for thermal estimation for real-time embedded systems is:

> *"What is the worst-case peak temperature of a real-time embedded system, under all possible scenarios of task executions?"*

The worst-case peak temperature can then be used as a reference for system designers either to decide whether thermal management mechanisms have to be applied or to evaluate the thermal effect of such techniques. Unfortunately, for non-trivial systems, to our best knowledge, the answer of this question is open. For example, for traditional scheduling algorithms, such as earliest-deadline-first (EDF), rate-monotonic (RM), deadline-monotonic (DM), real-time schedulability analysis is based on the critical instant of task releases to offer timing guarantees. However, for thermal investigations, even for simple arrival patterns, such as periodic tasks with jitter, there is no result about the critical instant which leads to the maximum peak temperature.

In this paper, we propose a method that at first constructs the worst-case task sequence, which can then be used to determine the corresponding worst-case peak temperature. The set of possible task arrival sequences is characterized by arrival curves as known from Real-Time Calculus [11] and Network Calculus [6]. An arrival curve bounds the maximal workload (in terms of accumulated processing times) that might arrive to the system for any interval length. As a special case, a set of periodic streams with bounded jitter and bursts can be modeled by arrival curves. Even though arrival curves constrain the possible workload injected to the system, there are infinitely many compliant workload traces in terms of initial phase, jitter, or burstiness. An exhaustive search to investigate the peak temperature corresponding to all possible workload traces constrained by an arrival curve is usually infeasible.

The proposed technique can be used to analyze the worst-case peak temperature for a given arrival curve under *work-conserving* real-time scheduling algorithms, including EDF, RM, and DM. For example, for EDF, RM, and DM, the critical instant for (timing) schedulability analysis is to release all tasks/events as early as possible without violating arrival curve constraints. By contrast, the main contribution of this paper shows that the workload trace that leads to the worst-case peak temperature at time instant τ is to release events in time interval $[\tau - \Delta, \tau)$ such that the processed computation in time interval $[\tau - \Delta, \tau)$ is maximized under arrival-curve constraints. For example, for periodic tasks with jitters, the worst-case peak temperature is to first warm up the system with periodic arrivals and then heat up the system with burst arrivals and jitters. Simulation results show that our technique

[1]978-3-9810801-7-9/DATE11/©2011 EDAA

can be applied to design real-time systems with timing and peak temperature guarantees.

II. System Model

Thermal Model. We base our work on the usual thermal model in the literature, e.g., [1], [2], [4], [14], [18]. In particular, the temperature follows a first-order linear differential equation,

$$C\frac{dT}{dt} = P - G(T - T_{amb}),$$

where C, P, G and T_{amb} denote thermal capacity, (temperature- and time-dependent) generated power, thermal conductance and ambient temperature, respectively. The temperature T as well as the generated power P are depending on time t, while the other quantities are supposed to be time-invariant. Note that, for notational simplicity, we will not present the time variable for T and P if the context is clear. Throughout this paper, all temperatures are given in the unit of absolute temperature (Kelvin, K). In addition, we model the temperature dependence of leakage power by means of a linear approximation [3], [7]. Therefore, the power consumption P is equal to $\phi T + \psi$, where ϕ and ψ are constants. As a result, we get the differential equation

$$C\frac{dT}{dt} = -(G - \phi)T + (\psi + GT_{amb}) \tag{1}$$

with time-dependent temperature T. For notational abbreviation, we can rewrite the above differential equation as follows:

$$\frac{dT}{dt} = -gT + h \text{ , with } g = \frac{G - \phi}{C} \text{ , } h = \frac{\psi + GT_{amb}}{C} \tag{2}$$

Now, we can determine the steady-state temperature with $\frac{dT}{dt} = 0$ as $T_\infty = h/g$. A closed-form solution to the above differential equation for constants g and h yields

$$T(t) = T(t_0) \cdot e^{-g(t-t_0)} + \int_{t_0}^{t} h \cdot e^{g\tau} d\tau \tag{3}$$

$$\Rightarrow \; T(t) = T_\infty + (T(t_0) - T_\infty) \cdot e^{-g\cdot(t-t_0)} \tag{4}$$

for time $t \geq t_0$, which has been widely used in the literature.

We distinguish between "active" and "idle" power component modes, i.e., P_a and P_i, characterized by different coefficients of the respective linear model. The active power P_a is the sum of leakage and dynamic power, and the idle power P_i can be considered as basically including just the leakage power.

$$P_a = \phi_a T + \psi_a \quad P_i = \phi_i T + \psi_i. \tag{5}$$

Therefore, (2) has piecewise constant coefficients

$$g_i = \frac{G - \phi_i}{C}, g_a = \frac{G - \phi_a}{C},$$
$$h_i = \frac{\psi_i + GT_{amb}}{C}, h_a = \frac{\psi_a + GT_{amb}}{C},$$

depending on the component mode.

Throughout this paper, we will implicitly consider a *proper* thermal model, in which the following reasonable conditions are satisfied:

- We have $g > 0$, i.e. , $G > \phi$ for both power modes.
- The steady-state temperature is not smaller in "active" mode than in "idle" mode, i.e. , we have $(T_\infty)_a \geq (T_\infty)_i$ or $\frac{\psi_a + GT_{amb}}{G - \phi_a} \geq \frac{\psi_i + GT_{amb}}{G - \phi_i}$.

Moreover, according to the thermal model, the component has the *thermal monotonicity* property.

Lemma 1: Suppose we consider two equal timed sequences of idle and active modes in a time interval from s to t. Then the sequence with the higher temperature at time s leads to a higher component temperature at time t.

Proof: If g_a is equal to g_i, this lemma simply comes from (3) by taking h as a time-dependent function. Otherwise, we can divide the interval $[s, t)$ into sub-intervals such that the component is only in one mode in one sub-interval. With the higher initial temperature at the beginning of a sub-interval, the temperature at the end of the sub-interval is also higher. By considering all the sub-intervals sequentially, we can reach the conclusion. ■

Computational Model. A component can process for $t - s$ time units in a time interval $[s, t)$. Tasks/events that have not been completed yet are queued. The only requirement towards the processing component is *work-conserving*, in which the processing component has to process an event if there is (at least) one event in the ready queue. There is no further assumption on the scheduling (queuing) discipline, may it be preemptive, non-preemptive, EDF, fixed priority, or any combination thereof.

The computation model follows the arrival curves in Network and Real-Time Calculus [6], [11]. For completeness, we will provide the required terms based on their definition in real-time calculus. We suppose that the component receives in time interval $[s, t)$ a cumulative workload (trace) of $R(s, t)$ time units, i.e. , in $[s, t)$ tasks with total workload of $R(s, t)$ arrive. The cumulative workload R is upper-bounded by the concept of an upper arrival curve α, where

$$R(s, t) \leq \alpha(t - s) \qquad \forall s < t \tag{6}$$

with $\alpha(0) = 0$. If this relation holds, we also say that a cumulative workload R complies to an arrival curve α. The arrival curve is assumed given in this paper, and the derivation is provided in the literature, e.g., [13]. A tight upper arrival curve is a monotonic increasing staircase function, i.e. , it has slope 0 almost everywhere. Moreover, the arrival curve is *sub-additive* if it satisfies

$$\alpha(x) + \alpha(y) \geq \alpha(x + y), \;\; \forall x, y \geq 0.$$

Note that in case of several independent workload functions R_j that need to be concurrently processed in a single component and that are bounded individually by arrival curves α_j, the accumulated workload can be bounded by $R(s, t) \leq \sum_{\forall j} \alpha_j(t - s) = \alpha(t - s)$. Therefore, the results in this paper will hold for several input event streams as well.

For work-conserving scheduling, the accumulated computing time $Q(s, t)$ of the component (for computing the incoming workload) in time interval $[s, t)$ is $Q(s, t) = \inf_{s \leq u \leq t}\{(t - u) + R(s, u)$, provided that at time s there is no buffered workload in the ready queue [11]. As a result, the upper bound on the accumulated computing time $\gamma(\Delta)$ for intervals of length Δ, with $Q(s, t) \leq \gamma(t - s)$, can be given as

$$Q(t - \Delta, t) \leq \gamma(\Delta) = \inf_{0 \leq \lambda \leq \Delta}\{(\Delta - \lambda) + \alpha(\lambda)\}. \tag{7}$$

Clearly, if α is a tight upper bound on R, then the above upper bound γ on Q is tight too. Throughout this paper, curve α is assumed sub-additive, which also leads to the sub-additivity of curve γ.

Note that, for peak temperature analysis, we implicitly assume the schedulability test for real-time systems has been passed, and do not intend to apply any thermal management to reduce the temperature.

Problem Definition. Now, we can formulate the worst-case peak temperature analysis problem:

> *Given is a work-conserving component characterized by a proper thermal model. The objective is to determine the peak temperature T^* for any cumulative workload R that complies to a given sub-additive arrival curve α.*

The most naïve solution to this problem is to state that T^* is upper-bounded by $(T_\infty)_a = \frac{h_a}{g_a}$, which simply ignores the arrival curve by assuming the component is always active. However, this is far beyond acceptable when the utilization is low. We would like to find the worst-case workload R that complies to the given arrival curve and that leads to the peak temperature T^*.

III. Thermal Analysis

In order to determine such an upper bound T^*, we will first construct worst-case accumulated computing time $Q^*(0,t)$ and then determine a corresponding feasible workload R^*.

A. Worst-Case Computing Time

The accumulated computing time $Q(s,t)$ for any fixed s with $s < t$ as well as its upper bound $\gamma(t-s)$ are monotonically increasing and their slopes at any time instant are either 1 or 0. Therefore, the mode function $S(t)$ defined as

$$S(t) = \frac{d\,Q(s,t)}{dt} \in \{0,1\} \tag{8}$$

determines the operating mode of the processing component. $S(t)=1$ or $S(t)=0$ denote that the processing component is in "active" or "idle" mode at time t, respectively. By combining (5) and (8), we can state that a component consumes P_i whenever $S(t)=0$ and P_a if $S(t)=1$.

As a first prerequisite, we will show in Lemma 2 that shifting an active processing mode closer to some time τ will always increase the temperature at time τ.

Lemma 2: For a given time instant τ, we consider two mode functions $S(t)$ and $\overline{S}(t)$ defined for $t \in [0,\tau)$. Mode function $S(t)$ only differs from $\overline{S}(t)$ in time interval $[\sigma, \sigma+2\delta)$ with $\delta > 0, \sigma \geq 0, \sigma + 2\delta < \tau$. In particular,

- *$S(t) = 1$ for all $t \in [\sigma, \sigma+\delta)$ ("active mode"),*
- *$S(t) = 0$ for all $t \in [\sigma+\delta, \sigma+2\delta)$ ("idle mode"), and*
- *$\overline{S}(t) = 1 - S(t)$ for all $t \in [\sigma, \sigma+2\delta)$.*

Then, with the same initial temperature at time 0 for both mode functions $S(t)$ and $\overline{S}(t)$, the temperature, denoted as $\overline{T}(\tau)$, at time τ for mode function $\overline{S}(t)$ is no less than the temperature, denoted as $T(\tau)$, at time τ for mode function $S(t)$. That is, $\overline{T}(\tau) \geq T(\tau)$.

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 $S(t) = 1$ for $t \in [\sigma, \sigma+\delta)$ and $S(t) = 0$ for $t \in [\sigma+\delta, \sigma+2\delta)$ we find

$$T(\sigma+2\delta) = \frac{h_i}{g_i} + (\frac{h_a}{g_a} - \frac{h_i}{g_i}) \cdot e^{-g_i\delta} + (T(\sigma) - \frac{h_a}{g_a}) \cdot e^{-(g_i+g_a)\delta}.$$

As $\overline{S}(t) = 0$ for $t \in [\sigma, \sigma+\delta)$ and $\overline{S}(t) = 1$ for $t \in [\sigma+\delta, \sigma+2\delta)$ we find

$$\overline{T}(\sigma+2\delta) = \frac{h_a}{g_a} + (\frac{h_i}{g_i} - \frac{h_a}{g_a}) \cdot e^{-g_a\delta} + (T(\sigma) - \frac{h_i}{g_i}) \cdot e^{-(g_i+g_a)\delta}.$$

For a proper thermal model, we have $\overline{T}(\sigma+2\delta) - T(\sigma+2\delta) = (\frac{h_a}{g_a} - \frac{h_i}{g_i}) \cdot [(1-e^{-g_a\delta}) \cdot (1-e^{-g_i\delta})] \geq 0$.

Moreover, for all $t \in [\sigma+2\delta, \tau)$, as $S(t) = \overline{S}(t)$, according the thermal monotonicity in Lemma 1, the condition $\overline{T}(\sigma+2\delta) \geq T(\sigma+2\delta)$ also implies that $\overline{T}(t) \geq T(t)$, which concludes the proof. $\blacksquare$

The following lemma shows that we obtain a higher temperature at time τ if the component is in the active mode at τ and has longer accumulated computing time in *any* interval ending at τ.

Lemma 3: For any given time instant τ, we consider two accumulated computing time functions Q resulting from mode function $S(t)$, and $\overline{Q}$ resulting from mode function $\overline{S}(t)$, with

$$\overline{Q}(\tau-\Delta, \tau) \geq Q(\tau-\Delta, \tau)$$

for all $0 \leq \Delta \leq \tau$. Then, with the same initial temperature at time 0, the temperature $\overline{T}(\tau)$ at time τ for mode function $\overline{S}(t)$ is no less than the temperature $T(\tau)$ at time τ for mode function $S(t)$.

Proof: Because of space limitation, only a sketch of the proof is provided. First note that because of (8), the condition $\overline{Q}(\tau-\Delta,\tau) \geq Q(\tau-\Delta,\tau)$ can be translated equivalently to $\int_{\tau-\Delta}^{\tau} \overline{S}(t)\,dt \geq \int_{\tau-\Delta}^{\tau} S(t)\,dt$. In other words, for $\overline{Q}$, the component is in the active mode at τ and has longer accumulated computing time in *any* interval ending at τ.

Now we will stepwise transform $S(t)$ into $\overline{S}(t)$ and, in each step, the temperature will not decrease because of Lemma 2. 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 $\tau = k_{\max}\delta$. We now execute the following algorithm:

1) Determine the smallest $1 \leq k_1 \leq k_{\max}$ such that $S(\tau - k_1\delta) \neq \overline{S}(\tau - k_1\delta)$. 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 $S(\tau - k_2\delta) = 1$. If such a k_2 exists, then change $S(t)$ for $t \in [\tau - k_2\delta, \tau - (k_2-1)\delta)$ from value 1 to value 0.
3) Change $S(t)$ for $t \in [\tau - k_1\delta, \tau - (k_1-1)\delta)$ from value 0 to value 1. Then continue with step 1.

Now, one can simply prove that after each iteration of the three steps, $T(\tau)$ increases until it reaches $\overline{T}(\tau)$ and therefore, the initial $T(\tau)$ was not larger than $\overline{T}(\tau)$. $\blacksquare$

Based on Lemma 3 we will present the first main result of this section. The following theorem provides a constructive method to determine the worst-case accumulated computing time Q^* for a work-conserving scheduling algorithm.

Theorem 4: Suppose that $T(t)$ is the temperature at time instant t for an arbitrary feasible workload trace that is bounded by the arrival curve α. When the scheduler is work-conserving and the thermal model is proper, the following statements hold:

- *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 at time τ, i.e. , $T^*(\tau) \geq T(\tau)$.*

- *If, in addition, $T(0) \leq (T_\infty)_i$ holds for the initial temperature, then for any feasible workload trace we have $T^*(\tau) \geq T(t)$ for all $0 \leq t \leq \tau$.*

Proof: At first, we show that $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ satisfies (7). 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 3 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 (7).

Now, let us prove the second item of the theorem by contradiction. Suppose now that there exists some time $\sigma \leq \tau$ where we have $T(\sigma) > T^*(\tau)$. From the first item in this theorem we know that the bound on $T(\sigma)$ is maximal 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 $T(0) = T^*(\tau-\sigma)$, then we have $T(\sigma) \leq T^*(\tau)$. Because of the thermal monotonicity in Lemma 1, the fact that $T(\sigma) > T^*(\tau)$ would require that $T(0) > T^*(\tau-\sigma)$.

As $T(0) \leq (T_\infty)_i$ and $(T_\infty)_i \leq (T_\infty)_a$, 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. ■

B. Worst-Case Accumulated Workload

Note that Theorem 4 only provides an upper bound $T^*(\tau)$ on the actual worst-case temperature and there may be no workload trace that leads to the critical accumulated computing time $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau-\Delta)$. Now, we will show that actually *there exists a single workload trace* $R^*(0,\Delta)$ for $0 \leq \Delta \leq \tau$ which (a) is compatible to the given arrival curve α and (b) results in $Q^*(0,\Delta)$.

We first determine a continuous accumulated workload function $R^*(0,\Delta)$, i.e. , which has slopes 1 and 0. It can be interpreted as the limit case of task arrivals with infinitesimally small inter-arrival time and infinitesimally small computation time. This condition will be relaxed in Lemma 6.

Theorem 5: The worst-case continuous workload function $R^*(0,\Delta) = Q^*(0,\Delta)$ for $0 \leq \Delta \leq \tau$ from Theorem 4

- *leads to the accumulated computing time $Q^*(0,\Delta)$,*
- *complies to the arrival curve α, and*
- *leads to the highest possible temperature $T^*(\tau) \geq T(t)$ for all $0 \leq t \leq \tau$ for any feasible workload trace with the same initial temperature $T^*(0) = T(0) \leq (T_\infty)_i$.*

Proof: For the first item we actually need to prove that $Q^*(0,\Delta) = \inf_{0 \leq u \leq \Delta}\{(\Delta - u) + Q^*(0,u)\}$ as $R^*(0,\Delta) = Q^*(0,\Delta)$. At first, we find that there exists a u' such that $(\Delta - u') + Q^*(0,u') = (\Delta - u') + Q^*(0,\Delta)$, namely $u' = \Delta$. Therefore, we only have to show that $(\Delta - u) + Q^*(0,u) \geq Q^*(0,\Delta)$ for all $0 \leq u \leq \Delta$. This condition is equivalent to $(\Delta - u) \geq Q^*(0,\Delta) - Q^*(0,u) = Q^*(u,\Delta)$. As the accumulated processing time in interval $[u,\Delta)$ can not exceed the available time $\Delta - u$, the first item is proven.

With $R^*(0,\Delta) = Q^*(0,\Delta)$, $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau-\Delta)$ and $\gamma(\Delta) = \inf_{0 \leq \lambda \leq \Delta}\{(\Delta - \lambda) + \alpha(\lambda)\}$ we find

$$\begin{aligned} R^*(x,y) &= \gamma(\tau - x) - \gamma(\tau - y) \\ &= \inf_{0 \leq \lambda \leq t-x}\{(t - x - \lambda) + \alpha(\lambda)\} \\ &\quad - \inf_{0 \leq \eta \leq t-y}\{(t - y - \eta) + \alpha(\eta)\} \\ &\leq \inf_{0 \leq u \leq (y-x)}\{((y-x) - u) + \alpha(u)\} \leq \alpha(y-x) \end{aligned}$$

where we use the fact that $x, y \leq \tau$, $x \leq y$, $\eta \leq \gamma$ as well as the subadditivity of α.

The third item is a simple consequence of Theorem 4 as (a) R^* leads to the accumulated computing time function Q^* and (b) Q^* leads to the highest temperature $T^*(\tau) \geq T(\tau)$ according to Theorem 4. ■

As a result of Theorem 5, the upper bound $T^*(\tau)$ determined through Theorem 4 is tight, i.e. , there exists a worst-case workload trace R^* that actually leads to $T^*(\tau)$. As it has been described, $R^*(0,\Delta)$ has slope 1 or 0 and corresponds to a continuous arrival of tasks. There are many possibilities to convert such a workload trace into one that has discrete task arrivals, but still (a) is compliant to the provided arrival curve α and (b) leads to the worst-case peak temperature at time τ. In the following, let us describe one of these possibilities.

Lemma 6: Let us suppose that for some constant c the given arrival curve α satisfies $\alpha(\Delta) = c \cdot \lceil \frac{1}{c}\alpha(\Delta) \rceil$ for all $\Delta \geq 0$, i.e. , the step size of $\alpha(\Delta)$ is an integer multiple of c. Suppose that the observation time τ is chosen such that $\gamma(\tau)$ according to (7) is a multiple of c as well. Then the worst-case accumulated workload $\hat{R}^*(0,\Delta) = c \cdot \lceil \frac{1}{c} R^*(0,\Delta) \rceil$ according to Theorem 5

- *is piecewise constant with a step size which is an integer multiple of c,*
- *complies to the arrival curve α according to (6), and*
- *leads to the highest possible temperature $T^*(\tau) \geq T(t)$ for all $0 \leq t \leq \tau$ for any feasible workload trace with the same initial temperature $T^*(0) = T(0) \leq (T_\infty)_i$.*

Proof: Let us first suppose without restricting the generality that $c = 1$. The first item is obvious from $\hat{R}^*(0,\Delta) = \lceil R^*(0,\Delta) \rceil$.

The second item can be shown as $\hat{R}^*(x,y) = \lceil R^*(0,y) \rceil - \lceil R^*(0,x) \rceil \leq \lceil R^*(0,y) - R^*(0,x) \rceil = \lceil R^*(x,y) \rceil \leq \lceil \alpha(y-x) \rceil = \alpha(y-x)$ for $x < y$.

In order to show the third item, we start from $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau-\Delta)$ in Theorem 4 and $\gamma(\Delta) = \inf_{0 \leq \lambda \leq \Delta}\{(\Delta - \lambda) + \alpha(\lambda)\}$ from (7). From the last equation one can observe that $\gamma(\Delta)$ has slope 1 or 0 and it has an integer value if it has slope 0. Therefore, if $\gamma(\tau)$ is integer as well, then we also find that $Q^*(0,\Delta)$ has slope 1 or 0 and it has an integer value if it has slope 0. If we can show that the accumulated workload function $\hat{R}^*(0,\Delta) = \lceil R^*(0,\Delta) \rceil = \lceil Q^*(0,\Delta) \rceil$ leads to the same worst-case accumulated processing time $Q^*(0,\Delta)$ as

$R^*(0,\Delta)$, the theorem would hold. Note that $Q^*(0,\Delta) = \inf_{0\le u\le\Delta}\{(\Delta-u)+Q^*(0,u)\}$ from Theorem 5. Because of the property of $Q^*(0,u)$ mentioned before, one can easily deduct that $\inf_{0\le u\le\Delta}\{(\Delta-u)+Q^*(0,u)\} = \inf_{0\le u\le\Delta}\{(\Delta-u)+\lceil Q^*(0,u)\rceil\} = \inf_{0\le u\le\Delta}\{(\Delta-u)+\hat{R}^*(0,u)\}$. ■

Note that $R^*(0,\Delta)$ does not necessarily contain the conventional critical instance scenario that is often used in real-time analysis in order to determine the worst-case timing behavior. For example, for periodic tasks with jitters, the worst-case peak temperature is to first warm up the system with periodic arrivals and then heat up the system with burst arrivals and jitters.

C. Computational Aspects

Theorem 4 implies a constructive method to determine the upper bound $T^*(\tau)$ for some time τ: Starting from a given arrival curve $\alpha(\Delta)$ for $0 \le \Delta \le \tau$ one can determine the function $\gamma(\Delta)$ for $0 \le \Delta \le \tau$ using (7). With $Q^*(0,\Delta) = \gamma(\tau)-\gamma(\tau-\Delta)$ for all $0 \le \Delta \le \tau$ and (8) one can determine the critical mode function $S^*(t)$, $0 \le t \le \tau$. It determines the critical trace of "idle" and "active" modes (5) which is used to solve the thermal model (2), i.e. , at time $t = \tau$. As a result, we can find $T^*(\tau)$.

There is one question remaining: How to choose an appropriate observation time τ such that a bound with an appropriate precision is determined? The following theorem provides an answer to this question.

Theorem 7: Suppose an arbitrarily feasible workload trace with the corresponding temperature $T(t)$ and initial temperature $T(0) \le (T_\infty)_i$. The worst-case peak temperature $T^* = max_{t\ge 0}T(t)$ is bounded by

$$T^* \in [T_i^*(\tau), T_a^*(\tau)],$$

for any $\tau > 0$. $T_i^*(\tau)$ and $T_a^*(\tau)$ denote the temperatures at time τ for the workload in Theorem 5 with initial temperatures $T_i^*(0) = (T_\infty)_i$ and $T_a^*(0) = (T_\infty)_a$, respectively. In addition, all observation times τ that satisfy the following relation guarantee a precision $T_a^*(\tau) - T_i^*(\tau)$ for estimating T^* with its upper bound $T_a^*(\tau)$

$$\tau \ge \frac{1}{\min\{g_i, g_a\}} \cdot \log\frac{(T_\infty)_a - (T_\infty)_i}{T_a^*(\tau) - T_i^*(\tau)}. \tag{9}$$

Proof: Because of space restrictions, only a sketch of the proof is given here. At first note, that for any trace with initial temperature $T(0) < (T_\infty)_i$ there exists always a feasible trace with initial temperature $T(0) = (T_\infty)_i$ (and therefore larger temperatures $T(t)$) as one can prepend the given one with an arbitrarily long idle mode. Following Theorem 5, we can construct a workload function that leads to the highest possible temperature T^* when using the same initial temperature $(T_\infty)_i$. Because of Lemma 1 and Theorem 4, we can easily see that $T_i^*(\tau) \le T^* \le T_a^*(\tau)$ because $T^*(0) = (T_\infty)_i$. For the bound on τ, we use (4) and find

$$\begin{aligned} T_a^*(\tau) - T_i^*(\tau) &= (T_a^*(0) - T_i^*(0)) \cdot e^{-\sum_k g_k \Delta t_k} \\ &\le (T_a^*(0) - T_i^*(0)) e^{-\min\{g_i, g_a\}\tau} \end{aligned}$$

where $g_k \in \{g_i, g_a\}$ denote the coefficient in the kth piecewise constant power section and Δt_k its length. ■

Following Theorem 7, we can determine a suitable observation time τ before the worst-case temperature simulation while guaranteeing a precision on the worst-case peak temperature bound.

	Video	Audio	Network
period (p)	[20, 90]ms	30ms	30ms
jitter	[20, 90]ms	10ms	10ms
min. inter-arrival	1ms	1ms	1ms
execution demand	6ms	3ms	2ms
deadline	period(p)	30ms	30ms

TABLE I
PARAMETERS OF THE SIMULATED SYSTEM.

G	C	$\phi_i = \phi_a$	ψ_i	ψ_a	T_{amb}	$(T_\infty)_a$	$(T_\infty)_i$
$0.3\frac{W}{K}$	$0.03\frac{J}{K}$	$0.1\frac{W}{K}$	$-25W$	$-11W$	$300K$	$395K$	$325K$

TABLE II
ARCHITECTURAL PARAMETERS.

IV. Experimental Analysis

System Description. A multi-processing video-conferencing system is considered, including a video codec, an audio codec, and a network process for communication management executing on an embedded processor. We use the period-jitter-delay model [12], our parameters being summarized in Table I. For the considered system, the video codec operation range varies from 12fps to 50fps (frames per second).This offers the possibility to investigate a large range of task invocation periods between 20ms and 90ms. The invocation periods of audio codec and network processes are 30ms. The relative deadline of each event stream is assumed identical to its invocation period.

We use the thermal and power consumption parameters as shown in Table II for demonstration. The base parameters were taken from [16], and these were set such that $(T^\infty)^a = 395K$ and $(T^\infty)^i = 325K$.

Peak Temperature Analysis. By setting $\tau = 1$s and both task invocation period and jitter equal to 20ms for the video codec, Figure 1 presents the results of transient temperatures of the system in the interval [0s, 1s) for 100 randomly generated workload traces that comply to the arrival curves, where (1) the *time critical instance* (the workload trace that fits to the critical instant for timing analysis by releasing the workload as early as possible) and (2) the *thermal critical instance* (worst-case workload trace generated by Theorem 5) are highlighted. All traces start from the steady-state idle temperature of 325K. As shown in Figure 1, the time critical instance has the higher transient temperature before its first idle time (around 0.31s) than the other traces. However, its temperature starts to decrease after that moment, and does not lead to the worst-case peak temperature. In contrast, the 100 random simulations might keep the system at higher temperature later on, but still under-estimate the worst-case peak temperature. Only thermal critical instance can reach the highest temperature at time $\tau = 1$s. As shown in Figure 1, the thermal critical trace first warms up the system with periodic arrivals and then heats up the system with burst arrivals and jitters.

However, $T_i^*(\tau)$ only gives the peak temperature of a feasible trace. As shown in Theorem 7, for estimating the worst-case peak temperature T^*, we also need the peak temperature at time τ, by starting at the steady-state temperature $(T_\infty)_a$ for

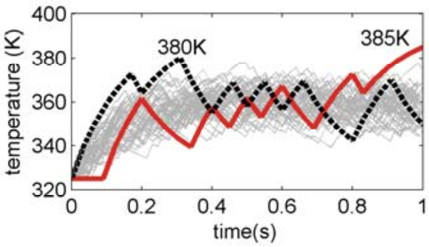

Fig. 1. Temperature (in time) for different traces (gray - 100 random traces; dashed black - time critical; red - temperature critical).

Thermal critical	Timing critical	Random Simulation
$385.00K$	$380.00K$	$378.00K$

TABLE III
PEAK TEMPERATURE FOR TRACES IN FIGURE 1 IN TIME INTERVAL $[0, 1)$S.

the trace generated by Theorem 5. Figure 2 demonstrates the temperature bound $[T_i^*(\tau), T_a^*(\tau)]$ by varying τ from 0.3s to 1.5s when task invocation period and jitter are both 20ms for the video codec. Note that, for different values of τ, the worst-case traces are also different. When τ is small, the bound is not precise. For example, the bound $[T_i^*(\tau), T_a^*(\tau)]$ is equal to $[381.44K, 390.91K]$ when $\tau = 0.3$s. However, when τ is equal to 1.5s, $T_i^*(\tau)$ and $T_a^*(\tau)]$ tend to converge, with the precision of the 1st digit of decimal point, i.e., 385.0K. As a result, we can conclude that the worst-case peak temperature is 385.0K for the invocation period of the video codec of 20ms and when the initial temperature $T(0)$ is not more than $(T_\infty)_i$.

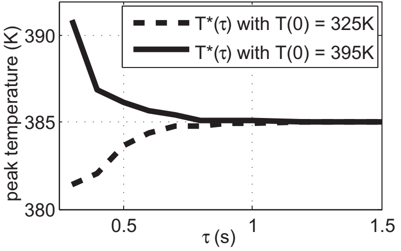

Fig. 2. Convergence of $T_i^*(\tau)$ and $T_a^*(\tau)$.

Worst-Case Temperature Analysis under Scheduling Nondeterminism. We would also like to analyze the effect of changes in invocation periods and jitter on both timing guarantees and worst-case peak temperature. We restrict ourselves to EDF scheduling in the reminder of this section, but the analysis is similar for any other work-conserving approach.

Figure 3 illustrates the dependence of the worst-case peak temperature and the schedulability of the system for different task invocation periods and jitters of the video codec. Note that τ is chosen carefully with precision loss less than 0.1K for the peak temperature analysis by applying Theorem 7. As shown in Figure 3, we observe that a larger jitter and a smaller invocation period will increase the worst-case peak temperature. This is because a larger jitter and a smaller invocation period both increase the size of a burst of active modes in the worst cases. If the jitter or the invocation period leads to an unacceptable temperature, designers can redesign the system by using proper parameters or adopt dynamic thermal management mechanisms.

V. Conclusion

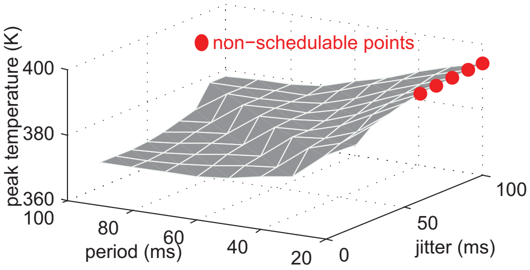

Fig. 3. Worst-case peak temperature for different set-ups of video codec.

We have presented an analytical approach to determine the worst-case peak temperature for any given work-conserving real-time system. The system is characterized by an upper bound on accumulated workload arriving from all task invocations, in any time interval. The analysis complexity comes from the non-determinism due to unknown initial phases, jitters, or bursts. It is shown, that the method (a) gives tight upper bounds on the worst-case peak temperature, (b) is constructive in the sense that the worst-case arrivals of tasks can be determined, and (c) provides bounds on the length of the observation interval for a given precision. With experimental results we show that our analysis method provides guarantees on (a) schedulability and (b) maximal system temperature. Direct impacts of such a method could be to avoid triggering dynamic thermal management, that is known to have high performance penalties, and to be a baseline reference for system designers when building a thermal critical system.

Acknowledgments

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

References

[1] N. Bansal, T. Kimbrel, and K. Pruhs. Dynamic speed scaling to manage energy and temperature. In *FOCS*, 2004.

[2] N. Bansal and K. Pruhs. Speed scaling to manage temperature. In *STACS*, 2005.

[3] T. Chantem, R. P. Dick, and X. S. Hu. Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. In *Design, Automation and Test in Europe*, 2008.

[4] T. Chantem, X. S. Hu, and R. P. Dick. Online work maximization under a peak temperature constraint. In *ISLPED*, pages 105–110, 2009.

[5] J.-J. Chen, S. Wang, and L. Thiele. Proactive speed scheduling for real-time tasks under thermal constraints. In *RTAS*, 2009.

[6] J.-Y. Le Boudec and P. Thiran. *Network Calculus — A Theory of Deterministic Queuing Systems for the Internet*, volume 2050 of *Lecture Notes in Computer Science*. 2001.

[7] Y. Liu, R. P. Dick, L. Shang, and H. Yang. Accurate temperature-dependent integrated circuit leakage power estimation is easy. In *DATE*, 2007.

[8] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, and G. D. Micheli. Temperature-aware processor frequency assignment for mpsocs using convex optimization. In *CODES/ISSS*, 2007.

[9] G. Quan, Y. Zhang, W. Wiles, and P. Pei. Guaranteed scheduling for repetitive hard real-time tasks under the maximal temperature constraint. In *CODES/ISSS*, 2008.

[10] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, 1(1):94–125, 2004.

[11] L. Thiele, S. Chakraborty, and M. Naedele. Real-time calculus for scheduling hard real-time systems. *ISCAS*, 4:101–104, 2000.

[12] E. Wandeler, A. Maxiaguine, and L. Thiele. Performance analysis of greedy shapers in real-time systems. In *DATE*, 2006.

[13] E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse. System Architecture Evaluation Using Modular Performance Analysis: A Case Study. *Int'l Journal on Software Tools for Technology Transfer (STTT)*, 8(6):649–667, Nov. 2006.

[14] S. Wang and R. Bettati. Delay analysis in temperature-constrained hard real-time systems with general task arrivals. In *RTSS*, 2006.

[15] Y. Wang, K. Ma, and X. Wang. Temperature-constrained power control for chip multiprocessors with online model estimation. In *ISCA*, pages 314–324, 2009.

[16] C.-Y. Yang, J.-J. Chen, L. Thiele, and T.-W. Kuo. Energy-efficient real-time task scheduling with temperature-dependent leakage. In *ACM/IEEE Conference of Design, Automation, and Test in Europe (DATE)*, 2010.

[17] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced CPU energy. In *Symposium on Foundations of Computer Science*, 1995.

[18] S. Zhang and K. S. Chatha. Approximation algorithm for the temperature-aware scheduling problem. In *ICCAD*, 2007.