Real-Time Syst (2013) 49:730–762
DOI 10.1007/s11241-013-9188-y

Real-time worst-case temperature analysis with temperature-dependent parameters

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

Published online: 30 July 2013

Abstract With the evolution of today's semiconductor technology, chip temperature increases rapidly mainly due to the growth in power density. Therefore, for modern embedded real-time systems it is crucial to estimate maximal temperatures early in the design in order to avoid burnout and to guarantee that the system can meet its real-time constraints. This paper provides answers to a fundamental question: What is the worst-case peak temperature of a real-time embedded system under all feasible scenarios of task arrivals? A novel thermal-aware analytic framework is proposed that combines a general event/resource model based on network and real-time calculus with system thermal equations. This analysis framework has the capability to handle a broad range of uncertainties in terms of task execution times, task invocation periods, jitter in task arrivals, and resource availability. The considered model takes both dynamic and leakage power as well as thermal dependent conductivity into consideration. Thorough simulation experiments validate the theoretical results.

Keywords Real-time systems · Real-time analysis · Formal worst-cast temperature analysis · Temperature-dependent leakage power · Resource availability

1 Introduction

Power density has been continuously increasing in modern embedded processors, leading to high on-chip temperatures. A system could simply fail if the operating

H. Yang (✉) · I. Bacivarov · D. Rai · L. Thiele
ETH Zurich, Zurich, Switzerland
e-mail: hyang@ethz.ch

J.-J. Chen
Karlsruhe Institute of Technology, Kalsruhe, Germany
e-mail: j.chen@kit.edu

temperature exceeds a certain threshold, leading to an overall low reliability and even chip burnout. Therefore, at design time, temperature reduction has been achieved through appropriate packaging and active heat dissipation mechanisms. On the other hand, at run time, dynamic thermal management can be adopted to enhance the system reliability and reduce the packaging cost.

For modern CMOS processors, switching activity (contributing to dynamic power mostly) and the leakage current (contributing to static power mostly) are the two major sources of power consumption. There are many research results available that relate to mechanisms for Dynamic Thermal Management (DTM) (Skadron et al. 2004; Brooks and Martonosi 2001; Huang et al. 2000) based on dynamic voltage/frequency scaling (DVFS), to reduce the power consumption in the system. In particular, dynamic voltage scaling (DVS) (Bansal et al. 2004) varies supply voltages leading to different execution speeds and thus different power consumption. On the other hand, chip temperatures are not only related to energy, but also to the time pattern of the power consumption, i.e. when the system consumes power. As a result, there is a large body of work available in the area of temperature-aware scheduling mechanisms, e.g., Kumar et al. (2006), Gomaa et al. (2004).

However, most of the above approaches implicitly assume that thermal management is essential, without checking whether it is indeed necessary. The drawback of such mechanisms is that either they are too expensive (i.e., packaging), they may cause a degradation of performance (i.e., DVFS) or substantial run-time overhead, which is unacceptable in today's embedded real-time applications. In situations where the operating system needs to be light-weight, implementing thermal-aware routines in scheduling comes with additional high cost. Therefore, system-level mechanisms at design-time become more and more interesting for future embedded systems. These methods can be simple such as appropriate mapping of tasks or using particular resource sharing mechanisms, e.g. servers or shapers (Kumar and Thiele 2011).

In all of these cases, the fundamental question is:

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

Unfortunately, for non-trivial systems, we could not find any research results that can answer this question. For example, traditional 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 with jitter there is a lack of results about the critical instant which leads to the maximum peak temperature.

We illustrate the complexity of the above fundamental question by means of a simple example. Let us assume that we have a work-conserving component which processes just a single event stream which is periodic with jitter: period 120 ms, jitter 240 ms, computing time 30 ms, minimal inter-arrival time 30 ms. The thermal model of the processing component is equivalent to that used in Sect. 5. In order to explain the problem we are going to solve, let us compare five simple methods for computing the temperature:

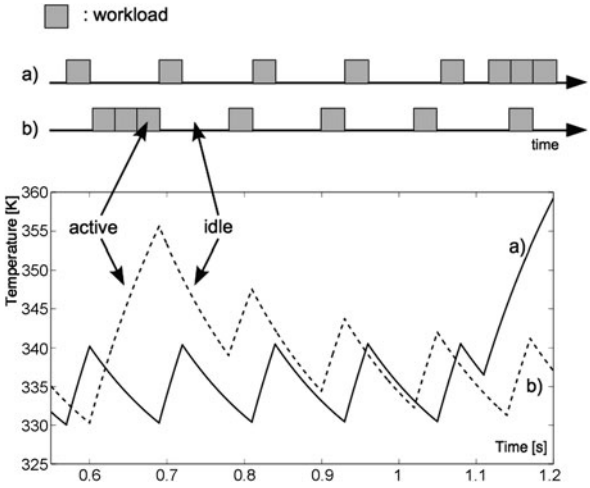

Fig. 1 Simple example that shows (**a**) the worst case workload trace and (**b**) a constructed workload trace as well as the corresponding temperature changes

1. We use the average workload of the tasks, i.e. utilization $U = 0.25$, and compute the corresponding average power of utilization 0.25 according to (6). The resulting steady state temperature T is 335.08 K.
2. We use a set of traces with random jitter that comply to the specification. For 100 independent runs with 1.2 s trace length we find an average, standard deviation and maximum temperature of 350.98 K, 1.75 K and 354.46 K, respectively.
3. We use the well known critical instant, i.e. starting from an arbitrary but fixed time all task instances are released as early as possible. The peak temperature observed in this method is 351.63 K with a trace length of 1.2 s.
4. We try to construct a reasonable critical instance as follows: We suppose that for an unlimited time, the task is periodic. At some time, the task shows its maximal jitter, i.e. a maximally dense workload pattern. In this case, the simulation shows a maximal temperature of 355.62 K.
5. We use the method developed in this paper and get a tight bound on the maximal temperature of 359.22 K with a trace length of 1.2 s.

It appears that none of the obvious methods is able to determine the maximal temperature even for such a very simple workload. The following Fig. 1 illustrates the workload traces corresponding to options (4) and (5) as well as the corresponding temperature traces with an initial temperature of 319.31 K.

In this paper, we propose an accurate system-level analytic technique which offers temperature-guarantees for real-time systems. When no information of the workload is given, the worst-case temperature can only be predicted assuming a fully stressed case, which results in the worst-case estimate of 402.33 K. This is clearly too pessimistic and far from *tight* estimation. Information on the workload enables a tight peak temperature estimation as will be shown in what follows. We consider general event arrivals modeled by arrival curves in Real-Time Calculus (Thiele et al. 2000) and Network Calculus (Le Boudec and Thiran 2001). An arrival curve provides an upper bound (and a lower bound) on the workload that might arrive to the system in any interval lengths. Usually, these curves can be derived by profiling sufficiently

representative traces or by analyzing the specification of an application. Even though arrival curves constrain the possible workload injected to the system, there are infinitely many traces that comply to the provided bound on the workload, in terms of initial phase, jitter, or burstiness.

Another uncertainty we have in the system is the availability of the computing resource. A processor is not always completely available for computation in modern embedded systems due to dynamic resource management like dynamic frequency modulation. The same is true if active-idle schemes are used in order to reduce its peak temperature, i.e. switching the processor on and off using a predefined pattern. Thus, the computing resource provided by a processor within a given time interval can also be constrained by an upper and lower bound, denoted as service curve. An exhaustive search to check all possible combinations of workload traces and computation availabilities to determine the peak temperature of the system is infeasible.

As in most studies related to temperature analysis and simulation, cooling and heating of the system is modeled by means of Fourier's law, i.e. the law of heat conduction modeled by a linear differential equation.

The thermal behavior of processing architectures is usually modeled by considering heat conductances and capacitances of micro-architectural elements. But it has been shown recently that leakage power and thermal conductances are temperature dependent. In contrast to Fourier's law on cooling and heating, the resulting differential equation is no longer linear. Based on this generalized and more accurate thermal model, we use the technique proposed in this paper to analyze the maximum peak temperature for given characterizations of the workload and the available computing resource under *workload-conserving* real-time scheduling algorithms, such as earliest-deadline-first (EDF), rate-monotonic (RM), deadline-monotonic (DM). The schedulability of a system can also be analyzed in the proposed technique in combination with existing Real-Time Calculus based schedulability analysis techniques (Thiele et al. 2000).

The contributions of this paper can be summarized as follows:

- Based on a characterization of the task arrival variability as well as the availability of the computation resource, an upper bound on the peak temperature for any workload-conserving scheduling discipline is determined.
- The peak temperature analysis is applied to three commonly used power and temperature models that consider temperature dependent leakage and conductances into account.
- Extensive experimental studies validate the proposed analysis framework and show its applicability.

This paper is structured as follows. Section 2 summarizes the related work in thermal management and system-level thermal analysis. While Sect. 3 presents our basic thermal and computational system model, Sect. 4 introduces the thermal analysis method and mathematically proves its validity for any work conservative system. Finally, Sect. 5 illustrates our model with some basic experiments and shows the validity of our method.

2 Related work

There have been many results in recent years about thermal management, including directions for (1) thermal-constrained scheduling to maximize the performance (Hung et al. 2005; Murali et al. 2007; Bansal et al. 2004; Zhang and Chatha 2007; Wang and Bettati 2006a, 2006b, 2008; Chantem et al. 2008, 2009; Quan et al. 2008; Chen et al. 2009) or determine the schedulability of real-time systems under given temperature constraints, (2) peak temperature reduction to meet performance constraints (Bansal and Pruhs 2005; Chen et al. 2007; Fisher et al. 2009), and (3) thermal control by applying control theory for system adaption (Fu et al. 2009, 2010; Wang et al. 2009).

Specifically, for thermal-constrained scheduling, in Bansal et al. (2004), an algorithm was developed to maximize the workload that can complete in a specified time window without violating imposed thermal constraints, in which the analysis is later adopted in Chen et al. (2009) to analyze the schedulability of frame-based real-time tasks, in which tasks have the same period. In Wang and Bettati (2006b, 2008), a reactive speed control scheme was proposed and schedulability tests were presented for frame-based real-time tasks. In Wang and Bettati (2006a), delay analysis is performed under reactive speed control for general task arrivals. In Zhang and Chatha (2007), approximation algorithms were developed to minimize the completion time, while each task is restricted to execute at one speed. Chantem et al. (2009) explore the maximization of workload for systems with discrete DVS speeds, under thermal constraints.

Moreover, for reducing the peak temperature, Bansal and Pruhs (2005) explore the peak temperature reduction by adopting the on-line algorithm for energy efficiency proposed by Yao et al. (1995). Fisher et al. (2009) design an algorithm for deciding the execution speeds on multicore systems to reduce the peak temperature while still guaranteeing the real-time property for global multiprocessor scheduling. Chen et al. (2007) analyze the peak temperature reduction by applying the optimal energy scheduling for periodic tasks. To model uncertainties of task execution, ambient temperature changes, or power variations, control theoretical approaches have been adopted in Fu et al. (2009, 2010), Wang et al. (2009). However, most of the above approaches implicitly assume that thermal management is essential without checking whether it is indeed necessary.

On the other hand, estimating system temperature is typically based on two main classes of tools. The first class is the thermal simulation of the system. HotSpot (Skadron 2004; Huang et al. 2009) is a popular thermal simulator that considers single core systems at micro-architecture level. The instantaneous power/performance profile is extracted via a power/performance simulator coupled with the thermal simulator, like for instance the system coupling Wattch (Brooks et al. 2000) with HotSpot (Huang et al. 2009). The computational efficiency of the thermal simulation is improved by allowing dynamic adaption of temporal and spatial granularity in ISAC (Yang et al. 2007). However, still these simulation methods have the coverage limitation, i.e., it is infeasible to simulate all possible cases to find out the worst-case peak temperature. The second class of tools is based on steady-state analysis, see e.g. Yang et al. (2010), where the system thermal equation is approximated by its

steady state behavior. While these methods work well for estimates of average power consumption, they fail to capture dynamic variations in power and temperature. In summary, these methods can either offer only information about average temperature in the system, or due to the incomplete coverage problem and nondeterministic workload behavior, they are not able to capture the worst case temperature.

As a thermal model of integrated circuits, first-order linear differential equation of RC electronic circuit have been used traditionally (Murali et al. 2007; Wang and Bettati 2006a, 2006b, 2008; Zhang and Chatha 2007), as shown in Sect. 3.3. This traditional model, however, turned out to be inaccurate (Liu et al. 2007) as some parameters are not constant to temperature changes and these affect the accuracy more significantly in deep sub-micron technology. Two parameters are temperature-varying in the model: leakage power and thermal conductance. It has been demonstrated a quadratic dependency (Liao et al. 2005; Liu et al. 2007). However, according to Liu et al. (2007), within the operating temperature ranges of current circuits, the leakage power can be accurately estimated by a linear model, approximation also used in this paper. On the other hand, the variable thermal conductance makes thermal differential equation more complicated. To best our knowledge, none of previous researches have considered temperature dependent leakage as well as conductances in system-level temperature analysis.

In contrast to the results described in Rai et al. (2011), this paper takes both temperature dependent parameters (leakage and conductance) into account in the thermal model. This results in a modified non-linear differential equation which substantially changes the analysis. Furthermore, we extend the resource availability to general service curve. Moreover, general properties are derived under which the main results hold. As a result of this generalization, results are shown for the most widely used power and temperature models. From an architectural point of view, the analysis technique is now applicable to resource varying architectures such as frequency modulated processors or on-off schemes. Finally, extensive experimental results are available that validate the proposed approach.

3 System model

This section introduces the models and methods that allow for the worst-case temperature analysis of a single computing component. The models take into account variability in the task arrivals, workloads, and computing resources, as well as temperature dependent leakage power and thermal conductance.

3.1 Computational model

The computational model of a processing component follows the ideas of network and real-time calculus. We suppose that the component receives in time interval $[s, t)$ a cumulative workload of $W(s, t)$ time units, e.g. in $[s, t)$ tasks with a total workload of $W(s, t)$ arrive. $W(s, t)$ is a stair case function for any fixed s, i.e. it has slope 0 almost everywhere and when a task arrives it jumps by its computation time. Similarly, the component is characterized by the availability of its computing resource, i.e. $R(s, t)$ time units are available for task processing in time interval $[s, t)$.

Incoming task workloads are stored in a queue until they are processed by the computing resource. If there are no waiting or arriving tasks in $[s,t)$, then the available resource $R(s,t)$ is wasted. Otherwise, it is used to process incoming and waiting tasks. For example, a component can process for $R(s,t)$ time units in time interval $[s,t)$.

According to the above explanation, the processing semantics is work conserving, i.e. the processing component has to process on available tasks if it has resources available. There are no further assumptions on the scheduling (queuing) discipline, i.e. it may be preemptive, non-preemptive, EDF, fixed priority, or any combination thereof. It can easily be verified, that the above requirement of work conservation leads to the following accumulated processing time $Q(s,t)$ in interval $[s,t)$, i.e. the accumulated time a component is spending to operate on incoming (and queued) workload

$$Q(s,t) = \inf_{s \le u \le t} \big\{ R(s,t) - R(s,u) + W(s,u) \big\} \tag{1}$$

where we suppose that at time s there are no buffered tasks, see also Le Boudec and Thiran (2001), Thiele et al. (2000).

We are interested in determining the upper bound on the temperature under any possible workload that is bounded by some event stream characteristics, e.g. periodic with arbitrary initial phase, periodic with jitter, sporadic, or bursty. To this end, we suppose that the cumulative workload W is upper bounded using the concept of an arrival curve

$$W(s,t) \le \alpha(t-s) \quad \forall s < t \tag{2}$$

where $\alpha(0) = 0$, see for example Le Boudec and Thiran (2001). A tight upper arrival curve is a monotonically increasing staircase function, i.e. it has slope 0 almost everywhere. Furthermore, it is sub-additive, i.e. it satisfies $\alpha(a) + \alpha(b) \ge \alpha(a+b)$ for all $a, b \ge 0$. Note that in case of several independent workload functions W_i that are bounded individually by arrival curves α_i and that need to be concurrently processed in a single component, the accumulated workload can be bounded by

$$W(s,t) \le \sum_{(i)} \alpha_i(t-s)$$

As a consequence, the results in this paper will hold for components with several event inputs as well.

In a similar way, the resource availability R can be upper and lower bounded using a pair of upper and lower service curves

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

where $\beta^l(0) = \beta^u(0) = 0$.

We can now determine an upper bound on the accumulated computing time according to Le Boudec and Thiran (2001), Wandeler et al. (2006) by using (2) and (3)

$$Q(t-\Delta, t) \le \gamma(\Delta) = \min\big\{ (\alpha \otimes \beta^u) \oslash \beta^l, \beta^u \big\} \tag{4}$$

where $(f \otimes g)(\Delta) = \inf_{0 \le \lambda \le \Delta}\{f(\Delta - \lambda) + g(\lambda)\}$ and $(f \oslash g)(\Delta) = \sup_{\lambda \ge 0}\{f(\Delta + \lambda) - g(\lambda)\}$.

Narratively speaking, the maximum amount of the workload processed during the time interval Δ is bounded by $\gamma(\Delta)$. When there is more workload available than the provided resource during the interval, it is bounded by the maximum available resource ($\min\{\ldots, \beta^u\}$). If not, $\gamma(\Delta)$ is determined by considering the combination of workload and resource ($(\alpha \otimes \beta^u) \oslash \beta^l$). The accumulated workload (that is not processed previously) does affect γ and this is bounded by the minimum resource availability ($\oslash\beta^l$).[1] In case of full resource availability, this γ is known to be $\alpha \otimes \beta^u$ according to the Real-Time and Network Calculus (Le Boudec and Thiran 2001; Wandeler et al. 2006).

Because of (1) and (4), the accumulated computing time $Q(s,t)$ for any fixed s as well as its upper bound $\gamma(t-s)$ are monotonically increasing. We now can define the rate function $S(t)$ which represents the rate (in workload per time unit) by which the computing resource is processing:

$$S(t) = \frac{dQ(s,t)}{dt} \tag{5}$$

If the computing resource is always fully available, then we find $R(s,t) = \beta^u(t-s) = \beta^l(t-s) = t - s$ as in any time interval of length $t - s$ the computing resource can fully operate on available tasks. In this case, $Q(s,t)$ always has either slope 1 or 0. In other words, the rate function satisfies $S(t) \in \{0, 1\}$, i.e. $S(t) = 1$ and $S(t) = 0$ denote that the processing component is in 'active' and 'idle' mode at time t, respectively.[2] However, such a simplified 'active/idle' mode classification is no longer valid if we allow for general resource availabilities. In this case, the rate function can take any value $0 \le S(t) \le 1$.

Now, we need to characterize the mapping of the operation modes to the corresponding power consumption.

3.2 Power models

A well accepted model for the frequency and voltage dependency of the dynamic power consumption P is $P \propto v^2 f$ where v and f denote the supply voltage and operating frequency in CMOS circuits, respectively. That is, the dynamic power consumption of a single operation mode is determined by the combination of v and f which may be chosen reactively at run-time in modern processor architectures. The processor frequency f is proportional to the execution rate $S(t)$.

For the sake of simplicity, the supply voltage v is assumed to be constant in the following discussions and therefore, the dynamic power P depends linearly on f which is proportional to the rate function $S(t)$.

[1] This is exactly where the work-conserving assumption is effective. That is, without this work-conserving assumption, arbitrarily many workload can be delayed and accumulated regardless of resource availability making γ unbounded.

[2] As $S(t)$ implies an operating mode at moment t, it is not a continuous function.

In addition, we may model the temperature dependence of leakage power by means of a linear approximation (Yang et al. 2010; Liu et al. 2007), which finally yields

$$P(t) = \phi T + \rho S(t) + \psi, \tag{6}$$

where $\phi T + \psi$ and ρ are the leakage and dynamic power, respectively. That is, ϕ is a temperature-dependent coefficient for the leakage, while the dynamic power in the active status is ρ.

We also will consider a different, well accepted model which applies to the case where $S(t) \in \{0, 1\}$. If $S(t) = 0$, then the processor is in 'idle' mode with power P^i, if $S(t) = 1$, then the processor is in the 'active' mode with power P^a:

$$P(t) = \begin{cases} P^i & \text{if } S(t) = 0 \\ P^a & \text{if } S(t) = 1 \end{cases} \quad \text{with } P^i = \phi^i T + \psi^i,\ P^a = \phi^a T + \psi^a \tag{7}$$

3.3 Thermal models

In order to show the versatility of our approach, we will consider three well accepted power-temperature models based on various forms of linear and quadratic differential equations, see also Murali et al. (2007), Wang and Bettati (2006a, 2006b, 2008), Zhang and Chatha (2007).

The first two models we consider are based on the following differential equation

$$C\frac{dT}{dt} = P - G \cdot (T - T_{amb}) \tag{8}$$

where C, P, G and T_{amb} denote the thermal capacity, the generated power, the thermal conductance, and the ambient temperature, respectively.

Active/idle model In this case, the system is either in active or idle mode based on the rate $S(t)$, i.e. $P(t) \in \{P^i, P^a\}$, see (7).

The steady-state temperatures with rates $S(t) = 0$ and $S(t) = 1$ can be determined as

$$T_0^\infty = \frac{GT_{amb} + \psi^i}{G - \phi^i}, \qquad T_1^\infty = \frac{GT_{amb} + \psi^a}{G - \phi^a}$$

A closed-form solution of (8) yields

$$T(t) = T^\infty + \left(T(t_0) - T^\infty\right) \cdot e^{-\frac{G-\phi}{C} \cdot (t - t_0)} \tag{9}$$

as long as the system is in a constant mode (active or idle) for $t \geq t_0$ and T^∞ is the corresponding steady-state temperature.

Continuous mode model Considering the continuous power model (6), we derive from (8) the differential equation

$$C\frac{dT}{dt} = \left(\phi T + \rho S(t) + \psi\right) - G(T - T_{amb}) \tag{10}$$

In addition, we consider an additional temperature-dependent parameter, namely the thermal conductance of silicon. It is reported that the thermal resistance (reciprocal of conductance) can be linearly approximated (Walkey et al. 2001), thus the temperature-dependent conductance can be modeled as follows:

$$G(T) = \frac{1}{R_0 + R_1 T} \tag{11}$$

Combining (10) and (11), we derive the following differential equation

$$\frac{dT}{dt} = \frac{T_{amb} - T}{L_0 + L_1 T} + MT + N \tag{12}$$

where $L_0 = CR_0$, $L_1 = CR_1$, $M = \phi/C$, and $N = (\rho S(t) + \psi)/C$.

Linear model Often, the physical computing device contains several interacting thermal layers such as silicon and copper. In this case, linear models based on matrix linear differential equations are well established, see e.g. Skadron et al. (2004), Skadron (2004), Huang et al. (2009). Based on the physical structure of the device, one can derive the corresponding impulse response function $h(t)$ which leads to the closed-form solution

$$T(t) = T_0^{\infty} + \int_0^t S(u) \cdot h(t-u)du \tag{13}$$

where T_0^{∞} denotes the steady-state temperature at constant rate $S = 0$. Note again, that $S(t)$ could be replaced in some monotonic power function $P(S(t))$.

3.4 Soundness of models

An *active/idle model* is called *proper* if it satisfies the following two properties:

- In order to guarantee a stable thermal model, we require that $G > \phi^i$ and $G > \phi^a$.
- We also require that the steady-state temperature in the active mode is larger than that in the idle mode, i.e. $\frac{GT_{amb}+\psi^i}{G-\phi^i} < \frac{GT_{amb}+\psi^a}{G-\phi^a}$.

We call a *continuous mode model* proper, if the temperature converges to a certain value for constant execution rate $S(t)$. We can determine potential steady-state temperatures under constant rate $S(t)$ by setting $\frac{dT}{dt} = 0$ in (12) as

$$\frac{1}{2L_1M}\left(1 - L_0M - L_1N \pm \sqrt{(L_0M + L_1N - 1)^2 - 4L_1M(L_0N + T_{amb})}\right)$$

As a consequence, the following condition needs to hold for a *proper continuous mode model*:

$$(L_0M + L_1N - 1)^2 \geq 4L_1M(L_0N + T_{amb}) \tag{14}$$

More precise insight on the *proper* continuous mode model is given in Fig. 2. From the two real potential steady-state temperatures only the lower one is stable.

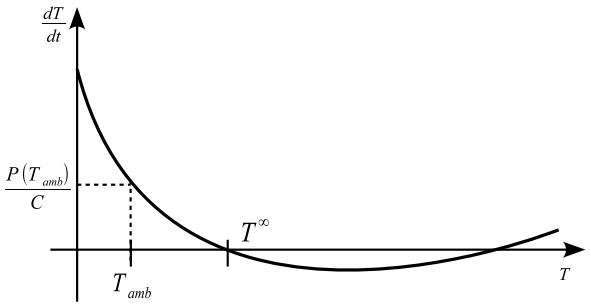

Fig. 2 Illustration of $\frac{dT}{dt}$ in (12) for a constant operation mode

Therefore, we can conclude that the steady-state temperature for a constant rate $S(t)$ is

$$T^{\infty} = \frac{1}{2L_1M}\left(1 - L_0M - L_1N - \sqrt{(L_0M + L_1N - 1)^2 - 4L_1M(L_0N + T_{amb})}\right). \tag{15}$$

Finally, we call a *linear model* proper, if the corresponding impulse response function $h(t)$ is strictly monotonically decreasing, i.e. $h(s) > h(t)$ for all $s > t$.

3.5 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 W that complies to a given sub-additive arrival curve α and any resource availability R bounded by the service curves β^u and β^l.*

The most naive solution to this problem is to state that T^* is upper-bounded by T_1^{∞} in a fully-utilized mode with $S(t) = 1$, which simply ignores the arrival/service curve by assuming the component is always fully utilized and there is always the full computing resource available. However, this is far beyond acceptable when the utilization is low. Therefore, we would like to determine a tight upper bound on T^*, and if possible, even determine a workload trace W and a resource availability R that leads to the peak temperature T^*.

4 Thermal analysis

In order to determine such an upper bound on the peak temperature T^*, we will at first show how to construct a worst case computing time sequence. This result will then be used to determine the desired upper bound. Finally, we will discuss the tightness of this bound and computational aspects.

4.1 Worst-case computing time

As a main prerequisite for constructing the peak temperature, we need two properties of the underlying power and thermal models, namely *monotonicity* and *shift*. These

two properties will be shown based on the generic form of the thermal differential equation

$$\frac{dT}{dt} = H(S, T) \tag{16}$$

where the dependence of $H(S, T)$ on the execution rate S reflects the time-varying power consumption. In other words, H already incorporates the power generation sequence.

We will now show that the generic thermal models as defined above satisfy the thermal *monotonicity* property.

Lemma 1 (Monotonicity) *Suppose we consider two solutions of* (16) *with different initial temperatures* T_0 *at time* s. *Then the solution* $T(t)$ *with the higher initial temperature will at no time* $t \geq s$ *be smaller than the solution with lower initial temperature*.

Proof Let us suppose that the above theorem is false. Then the temperature trace with the higher initial temperature at s has the lower temperature at some time t. As a consequence, the two temperature traces cross in between, i.e. there exists a time $s < t_0 \leq t$ where the temperatures of the two sequences are equal. As the two sequences have equal derivative $H(S, T)$ for equal temperatures, their temperature at t will be equal, which contradicts the assumption. □

In terms of the various thermal models that have been described in the previous section, one can interpret the above lemma as follows: If the power traces are the same, then a higher initial temperature does not lead to a smaller final temperature.

As the next prerequisite in constructing an upper bound on the temperature, we will show in Lemma 2 under what conditions a time shift of the rate function S leads to temperature increase.

Lemma 2 (Shifting) *Given is a differential equation of the form* (16). *We consider two different execution rate traces in the time interval* $[0, 2\delta)$, *namely*

$$S(t) = \begin{cases} S_1 & \text{if } 0 \leq t < \delta \\ S_2 & \text{if } \delta \leq t < 2\delta \end{cases}$$

$$\overline{S}(t) = \begin{cases} S_1 - \sigma & \text{if } 0 \leq t < \delta \\ S_2 + \sigma & \text{if } \delta \leq t < 2\delta \end{cases}$$

for some rate shift $\sigma \geq 0$. *Then the corresponding temperature traces* $T(t)$ *and* $\overline{T}(t)$ *satisfy*

$$\overline{T}(2\delta) > T(2\delta)$$

for $T(0) = \overline{T}(0) = T_0$ *and* $\delta \to 0$ *if either*

$$H(S_1 - \sigma, T_0) + H(S_2 + \sigma, T_0) > H(S_1, T_0) + H(S_2, T_0) \tag{17}$$

or

$$H(S_1-\sigma,T_0)+H(S_2+\sigma,T_0)=H(S_1,T_0)+H(S_2,T_0)$$
$$\wedge\quad H(S_1-\sigma,T_0)\cdot\frac{\partial H(S_2+\sigma,T)}{\partial T}(T_0)>H(S_1,T_0)\cdot\frac{\partial H(S_2,T)}{\partial T}(T_0) \quad (18)$$

Proof Neglecting higher order terms in δ, we can write

$$T(2\delta)=T_0+\delta\big[H(S_1,T_0)+H\big(S_2,T_0+\delta H(S_1,T_0)\big)\big]$$
$$\overline{T}(2\delta)=T_0+\delta\big[H(S_1-\sigma,T_0)+H\big(S_2+\sigma,T_0+\delta H(S_1-\sigma,T_0)\big)\big]$$

The condition $\overline{T}(2\delta)>T(2\delta)$ is satisfied if

$$H(S_1-\sigma,T_0)+H\big(S_2+\sigma,T_0+\delta H(S_1-\sigma,T_0)\big)$$
$$>H(S_1,T_0)+H\big(S_2,T_0+\delta H(S_1,T_0)\big)$$

This condition directly leads to (17,18) if we use the following relations

$$H\big(S_2,T_0+\delta H(S_1,T_0)\big)=H(S_2,T_0)+\delta H(S_1,T_0)\frac{\partial H(S_2,T)}{\partial T}(T_0)$$
$$H(S_2+\sigma,T_0+\delta H(S_1-\sigma,T_0)=H(S_2+\sigma,T_0)+\delta H(S_1-\sigma,T_0)$$
$$\times\frac{\partial H(S_2+\sigma,T)}{\partial T}(T_0)$$

which hold for $\delta\to 0$. □

Based on this lemma, we will show later that the temperature at some measuring time gets larger if we 'shift' some 'execution rate' σ towards it. The lemma itself uses two execution rate traces $S(t)$ and $\overline{S}(t)$. These two cases correspond to the above mentioned 'shift' towards larger time instances. If (17) and (18) are satisfied, then we have a higher temperature at some measuring time after we shift the execution rate σ closer to it. We now show that the shift lemma holds for the simple active/idle mode.

Lemma 3 *Suppose that we consider the active/idle model according to* (7), (8) *with*

$$H(0,T)=P^i-G\cdot(T-T_{amb}),\qquad H(1,T)=P^a-G\cdot(T-T_{amb})$$

Then with $S_1=1$, $S_2=0$ *and* $\sigma=1$, *the condition* (18) *in lemma* 2 *is satisfied if the model is proper.*

Proof Condition (18) is equivalent to

$$H(0,T_0)+H(1,T_0)=H(1,T_0)+H(0,T_0)$$
$$\wedge\quad H(0,T_0)\frac{\partial H(1,T)}{\partial T}(T_0)>H(1,T_0)\frac{\partial H(0,T)}{\partial T}(T_0)$$

which is equivalent to

$$\begin{aligned}&\big(\phi^i T_0 + \psi^i - G\cdot(T_0 - T_{amb})\big)\big(G-\phi^a\big)\\ &< \big(\phi^a T_0 + \psi^a - G\cdot(T_0 - T_{amb})\big)\big(G-\phi^i\big)\\ \Leftrightarrow\quad &\frac{\psi^i + GT_{amb}}{G-\phi^i} < \frac{\psi^a + GT_{amb}}{G-\phi^a}\end{aligned}$$

if the model is proper. The last relation is true as the active/idle model is proper. □

From the above lemma, we can conclude that the shift lemma holds for the active/idle model if it is *proper*. In other words, exchanging active and idle modes such that the active mode gets closer to the measurement time increases the temperature.

Now, we will show a similar condition for the more complex continuous mode model.

Lemma 4 *Suppose that we consider the continuous model according to* (12) *with*

$$H(S,T) = \frac{T_{amb} - T}{L_0 + L_1 T} + MT + N(S)$$

Then the condition (18) *in Lemma* 2 *is satisfied if*

$$\sigma > 0 \quad \wedge \quad T_0 < T^{\infty}$$

Proof Due to the linearity of $N(S)$, we have $H(S_1 - \sigma, T_0) + H(S_2 + \sigma, T_0) = H(S_1, T_0) + H(S_2, T_0)$. Then we find that (18) is equivalent to

$$\begin{aligned}&\left(\frac{T_{amb} - T}{L_0 + L_1 T} + MT + N(S_1 - \sigma)\right)\cdot\frac{-L_0 - L_1 T_{amb}}{(L_0 + L_1 T)^2}\\ &> \left(\frac{T_{amb} - T}{L_0 + L_1 T} + MT + N(S_1)\right)\cdot\frac{-L_0 - L_1 T_{amb}}{(L_0 + L_1 T)^2}\end{aligned}$$

If $T_0 < T_{min} = M/2L$ then this condition is equivalent to

$$\begin{aligned}&\frac{T_{amb} - T}{L_0 + L_1 T} + MT + N(S_1 - \sigma) < \frac{T_{amb} - T}{L_0 + L_1 T} + MT + N(S_1)\\ &\Leftrightarrow\quad N(S_1 - \sigma) < N(S_1) \quad\Leftrightarrow\quad \sigma > 0\end{aligned}$$

□

In summary, the above lemma states that one gets a higher temperature if we shift the rate σ towards the measuring time in case of the continuous model.

Finally, we show that the shift-property as defined in Lemma 2 also holds for the linear model.

Lemma 5 (Shift) *Given is a linear temperature model of the form* (13). *We consider two different execution rate traces in the time interval* $[0, 2\delta)$, *namely*

$$S(t) = \begin{cases} S_1 & \text{if } s \le t < s + \delta \\ S_2 & \text{if } s + \delta \le t < s + 2\delta \end{cases}$$

$$\overline{S}(t) = \begin{cases} S_1 - \sigma & \text{if } s \le t < s + \delta \\ S_2 + \sigma & \text{if } s + \delta \le t < s + 2\delta \end{cases}$$

for some rate shift $\sigma > 0$. *It the model is proper, then the corresponding temperature traces* $T(t)$ *and* $\overline{T}(t)$ *satisfy*

$$\overline{T}(s + 2\delta) > T(s + 2\delta)$$

for $T(s) = \overline{T}(s)$ *and* $\delta \to 0$.

Proof From (13) we can derive by simple algebraic transformations

$$\overline{T}(s + 2\delta) - T(s + 2\delta) = \sigma \left(\int_s^{s+\delta} h(t - u)du - \int_{s+\delta}^{s+2\delta} h(t - u)du \right)$$

Therefore, $\overline{T}(s + 2\delta) > T(s + 2\delta)$ is equivalent to

$$\int_0^{\delta} h\big((t - s) - u\big)du > \int_0^{\delta} h\big((t - s) - \delta - u\big)du$$

which is satisfied if $h(t) > h(t - \delta)$ for all δ, i.e. if the model is proper. □

The next lemma shows for all considered power and temperature models, 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 of this section.

Lemma 6 (Worst-case computing time) *Given is a proper thermal model, i.e. a model that satisfies the shift condition as defined in Lemmata 2 and 5, as well as some time instance* τ. *In addition, we consider two accumulated computing time functions* Q *and* $\overline{Q}$ *which satisfy*

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

for all $0 \le \Delta \le \tau$. *Then, if* $\overline{T}(0) = T(0)$ *we have* $\overline{T}(\tau) \ge T(\tau)$, *i.e. the temperature at time* τ *is not higher if we use the accumulated computing time function* Q *instead of* $\overline{Q}$.

Proof First note that because of (5), the condition of the lemma translates equivalently to the following condition on the corresponding rate functions S and $\overline{S}$:

$$\int_{\tau-\Delta}^{\tau} \overline{S}(t)\,dt \ge \int_{\tau-\Delta}^{\tau} S(t)\,dt$$

The following algorithm performs a stepwise transformation of $S(t)$ into $\overline{S}(t)$ using the elementary rate shift operations in Lemmata 2 and 5. As a result, we can show that in each step the temperature will increase. 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 $k \ge 0$. 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 $S(\tau - k_1\delta) < \overline{S}(\tau - k_1\delta)$. 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 such a k_2 does not exist, then $\overline{T}(\tau) \geq T(\tau)$ holds and the algorithm stops. Otherwise, execute the following steps:
   (a) Set $\sigma = \min\{S(\tau - k_2\delta), \overline{S}(\tau - k_1\delta) - S(\tau - k_1\delta)\}$.
   (b) For all i starting from k_2 down to $k_1 + 1$ change $S(t)$ as follows: $S(t) := S(t) - \sigma$ for $t \in [\tau - i\delta, \tau - (i-1)\delta)$ and $S(t) := S(t) + \sigma$ for $t \in [\tau - (i-1)\delta, \tau - (i-2)\delta)$.
3. If $S(\tau - k_1\delta) < \overline{S}(\tau - k_1\delta)$ is still true, continue with step 2. Otherwise, go to step 1.

Note that σ is always positive in step 2(a) as required in Lemmata 3, 4 and 5. Now, we can simply prove using Lemma 2 that after each execution of step 2, $T(\tau)$ decreases until it reaches $\overline{T}(\tau)$. Therefore, the initial $T(\tau)$ was not larger than $\overline{T}(\tau)$. □

Based on the above Lemma 6 we will show the first main result of this section. The following Theorem 1 provides a constructive method to determine the worst-case accumulated computing time Q^* for a work-conserving component.

Theorem 1 (Worst-case temperature) *Given is a work-conserving processing component with the computational model* (1)*, one of the power models defined in Sect.* 3.2*, and a thermal model as described in Sect.* 3.3*. The thermal model is supposed to be proper according to Sect.* 3.4*. 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*, where* γ *is defined in* (4)*. Then* $T^*(\tau)$ *is an upper bound on the highest temperature* $T(\tau)$ *for all feasible workload traces that are bounded by the arrival curve* α *and service curve* β *according to* (2) *and* (3) *respectively.*
- *If in addition* $T(0) \leq T_0^\infty$ *holds, where* T_0^∞ *is a steady-state temperature for the constant rate function* $S(t) = 0$*, 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 (4). We have $Q^*(t - \Delta, t) = Q^*(0, t) - Q^*(0, t - \Delta) = \gamma(\tau) - \gamma(\tau - t) - \gamma(\tau) + \gamma(\tau - t + \Delta) = \gamma(\tau - t + \Delta) - \gamma(\tau - t)$. Therefore, we have to show that $\gamma(\tau - t + \Delta) - \gamma(\tau - t) \leq \gamma(\Delta)$ for all $0 \leq t \leq \tau$ which is satisfied if $\gamma(a + b) \leq \gamma(a) + \gamma(b)$ for all $a, b \geq 0$.

It is well known from network and real-time calculus that tight upper arrival/service curves are sub-additive and tight lower service curves are superadditive, i.e. they satisfy $\alpha(a) + \alpha(b) \geq \alpha(a + b)$, $\beta^u(a) + \beta^u(b) \geq \beta^u(a + b)$ and $\beta^l(a) + \beta^l(b) \leq \beta^l(a + b)$, see also Thiele et al. (2000). Using these properties in (4) results in $\gamma(a) + \gamma(b) \leq \gamma(a + b)$.

Now, we will show the first item of the theorem by contradiction. Suppose that there is an accumulated computing time Q which leads to a higher temperature $T(\tau)$ at time τ. Then according to Lemma 6 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 $Q(\tau - \Delta, \tau) > \gamma(\Delta)$. Such a computing time function Q would violate (4).

Now, let us prove the second item of the theorem by contradiction. To this end, we denote as $T(t)$ the temperature caused by some accumulated computing time Q. Suppose now that there exists some time $\sigma \leq \tau$ where we have $T(\sigma) > T^*(\tau)$. For an upper bound on $T(\sigma)$, e.g. $T^*(\sigma) \geq T(\sigma)$, we also would find $T^*(\sigma) > T^*(\tau)$. Suppose that we construct such an upper bound using the first item in the theorem, i.e. we choose as a worst case accumulated computing time $Q(\sigma - \Delta, \sigma) = \gamma(\Delta)$ for $0 \leq \Delta \leq \sigma$.

As $Q^*(\tau - \Delta, \tau) = \gamma(\Delta)$, we can now conclude that Q^* shifted by $\tau - \sigma$ and Q are equal, i.e. we have $Q(\sigma - \Delta, \sigma) = Q^*(\tau - \Delta, \tau)$ for $0 \leq \Delta \leq \sigma$. Because of the monotonicity of the thermal model (see Lemma 1), the assumption $T^*(\sigma) > T^*(\tau)$ would require that the initial temperature $T(0)$ (as used for the worst case computing time Q) is larger then the temperature at time $\tau - \sigma$ when using computing time Q^*.

As the thermal models are supposed to be proper, temperatures in all scenarios, i.e. with all computing time functions, are always larger or equal than the minimum of the initial temperature and T_0^∞. As we have $T(0) \leq T_0^\infty$, temperatures are always larger or equal $T(0)$. This contradicts the above requirement that the initial temperature $T(0)$ is larger than some temperature that occurs using Q^*. □

As a result of the above theorem, we can now describe a method to determine an upper bound on the component temperature $T^*(\tau)$ at some time $t = \tau$:

- We start with a given bound on the cumulative workload, i.e. the arrival curve α, and with given upper and lower bounds on the resource availability β^u and β^l, see (2) and (3).
- We determine the upper bound on the accumulated computing time γ based on (4). Using γ, we can determine the worst case computing time function Q^* according to Theorem 1 and the corresponding rate function S^* in (5).
- Corresponding to the chosen power and temperature model, we solve the temperature equation (8), (10), or (13). The model has to satisfy the conditions of the corresponding Lemma 3, 4 or 5, i.e. the models should be proper according to Sect. 3.4. The solution may be done analytically or numerically with an appropriate initial temperature $T(0)$ according to Theorem 1. The temperature at $t = \tau$ is $T^*(\tau)$.

There are still two questions that need to be answered: Under what conditions is the above bound tight? What is a reasonable time τ such that the upper bound $T^*(\tau)$ holds for arbitrary long runs of the system, i.e. $T(t) \leq T^*(\tau)$ for all $t \geq 0$? The next two sections will answer these questions.

4.2 Tightness

Note that Theorem 1 only provides an upper bound $T^*(\tau)$ on the actual worst-case temperature. In other words, there may be no *single* trace that leads to the critical accumulated computing time $Q^*(0, \Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$. Now, we will show that

in the case of maximal resource availability there exists a single trace $W^*(0, \Delta)$ for $0 \leq \Delta \leq \tau$ which (a) is compatible to the given arrival curve α and (b) results in the worst case accumulated computing time $Q^*(0, \Delta)$. Maximal resource availability results in $R(s,t) = t - s$, i.e. in any time interval of length $t - s$ the computing resource is fully available. As a result we find $\beta^u(\Delta) = \beta^l(\Delta) = \Delta$.

We first determine a continuous accumulated workload function $W^*(0, \Delta)$, i.e. which has slopes 1 and 0. It can be interpreted as the limit case of task arrivals with infinitesimally small inter-arrival times and infinitesimally small computation times.

Theorem 2 (Tightness) *Suppose that the assumptions from Theorem 1 hold and the resource availability satisfies $R(s,t) = t - s$ for all $t \geq s$. Then, the continuous workload function $W^*(0, \Delta) = Q^*(0, \Delta)$ for $0 \leq \Delta \leq \tau$*

- *leads to the accumulated computing time $Q^*(0, \Delta)$ according to the computational model (1),*
- *complies to the arrival curve α according to (2), and*
- *leads to the highest possible temperature $T^*(\tau) \geq T(\tau)$ for any feasible workload trace.*

Proof With the condition $\beta^l(\Delta) = \beta^u(\Delta) = \Delta$, the computational model in (1), (4) can be simplified to

$$Q(s,t) = \inf_{s \leq u \leq t} \big\{(t-u) + \alpha(s,u)\big\} \tag{19}$$

$$Q(t-\Delta, t) \leq \gamma(\Delta) = \inf_{0 \leq u \leq \Delta} \big\{(\Delta - u) + \alpha(u)\big\} \tag{20}$$

Thus, 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 $W^*(0, \Delta) = Q^*(0, \Delta)$. At first, we find that there exists a u' such that $(\Delta - u') + Q^*(0, u') = Q^*(0, \Delta)$, 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 service $\Delta - u$, the first item is proven.

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

$$\begin{aligned} W^*(a,b) &= \gamma(\tau - a) - \gamma(\tau - b) \\ &= \inf_{0 \leq \lambda \leq t-a} \big\{(t - a - \lambda) + \alpha(\lambda)\big\} - \inf_{0 \leq \eta \leq t-b} \big\{(t - a - \eta) + \alpha(\eta)\big\} \\ &\leq \inf_{0 \leq u \leq (b-a)} \big\{\big((b-a) - u\big) + \alpha(u)\big\} \leq \alpha(b-a) \end{aligned}$$

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

The third item is a simple consequence of Theorem 1 as (a) W^* leads to the accumulated computing time function Q^* and (b) Q^* leads to the highest temperature $T^*(\tau) \geq T(\tau)$. □

As has been mentioned above, $W^*(0, \Delta)$ 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, which is compliant to the provided arrival curve α and which leads to the worst-case temperature. In the following, let us describe one of these possibilities.

Lemma 7 (Worst-case workload) *Let us suppose that the conditions of Theorem 2 hold. Furthermore, 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 (4) is a multiple of c as well. Then the worst-case accumulated workload $\hat{W}^*(0, \Delta) = c \cdot \lceil \frac{1}{c} W^*(0, \Delta) \rceil$*

- *is piecewise constant with a step size which is an integer multiple of c,*
- *complies to the arrival curve α according to (2) and*
- *leads to the highest possible temperature $T^*(\tau) \geq T(t)$ for all $0 \leq t \leq \tau$ for any feasible workload trace.*

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

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

In order to show the third item, we start from $Q^*(0, \Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ in Theorem 1 and $\gamma(\Delta) = \inf_{0 \leq \lambda \leq \Delta}\{(\Delta - \lambda) + \alpha(\lambda)\}$ from (20). 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 $\hat{W}^*(0, \Delta) = \lceil W^*(0, \Delta) \rceil = \lceil Q^*(0, \Delta) \rceil$ leads to the same worst-case accumulated processing time $Q^*(0, \Delta)$ as $W^*(0, \Delta)$, the theorem would hold. Note that $Q^*(0, \Delta) = \inf_{0 \leq u \leq \Delta}\{(\Delta - u) + Q^*(0, u)\}$ from (20) in Theorem 2. Because of the property of $C^*(0, u)$ mentioned before, one can easily deduct that $\inf_{0 \leq u \leq \Delta}\{(\Delta - u) + Q^*(0, u)\} = \inf_{0 \leq u \leq \Delta}\{(\Delta - u) + \lceil Q^*(0, u) \rceil\} = \inf_{0 \leq u \leq \Delta}\{(\Delta - u) + \hat{W}^*(0, u)\}$. □

Note that $W^*(0, \Delta)$ does not necessarily represent the conventional critical instant scenario that is often used in real-time analysis in order to determine the worst-case timing behavior.

As a result of Theorem 2 and Lemma 7, the upper bound $T^*(\tau)$ determined through Theorem 1 is proven to be tight under the conditions mentioned in Theorem 2, i.e. there exists a worst case workload trace W^* that actually leads to $T^*(\tau)$ when a system resource is fully available all the time.

4.3 Computational aspects

As has been mentioned already, Theorem 1 implies a constructive method to determine the upper bound $T^*(\tau)$ for some time τ: Starting from a given arrival curve $\alpha(\Delta)$ for $0 \leq \Delta \leq \tau$ one can determine the function $\gamma(\Delta)$ for $0 \leq \Delta \leq \tau$ using (4).

With $Q^*(0,\Delta) = \gamma(\tau) - \gamma(\tau - \Delta)$ for all $0 \leq \Delta \leq \tau$ and (5) one can determine the critical mode function $S^*(t)$, $0 \leq t \leq \tau$. It determines the critical distribution of operating modes (6) which is used to solve the thermal model, i.e. at time $t = \tau$ we 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? For simplicity and without restricting the generality, we suppose for the remainder of this section, that the initial temperature satisfies $T(0) = T_0^\infty$, i.e. it equals the steady state temperature of an idle system with constant rate $S = 0$.

As a simple approach, it is possible to determine an upper bound on the precision, given an observation time τ. This does not solve the original question but can be used as a basis for some heuristics, e.g. double the observation time τ until the desired precision is guaranteed. Given τ, we first determine the upper bound $T_0^*(\tau)$ for initial temperature T_0^∞ and the bound $T_1^*(\tau)$ for initial temperature T_1^∞, i.e. the steady state temperature of an active system with constant rate $S = 1$. Due to the monotonicity of the power/temperature mode according to Lemma 1, we can conclude that $\lim_{t\to\infty} T^*(t) \in [T_0^*(\tau), T_1^*(\tau)]$.

For the active/idle model and the continuous mode model, it is possible to determine the observation time τ explicitly, given a desired precision.

Lemma 8 *We are given computational, temperature and power models according to Theorem 1 and $T(0) = T_0^\infty$, i.e. the steady state temperature for constant rate $S = 0$. Then*

$$\lim_{t\to\infty} T^*(t) \in \left[T^*(\tau), T^*(\tau) + \Delta T^*\right]$$

where $T^*(\tau)$ denotes the upper bound determined using Theorem 1 and observation time τ and where ΔT^* denotes a bound on the corresponding precision. Given ΔT^*, the following expressions provide a lower bound on the observation time τ that achieves this precision:

$$\tau \geq \frac{C}{G - \max\{\phi^i, \phi^a\}} \ln\left(\frac{T_1^\infty - T_0^\infty}{\Delta T^*}\right) \tag{21}$$

holds for the active/idle model and

$$\tau \geq \frac{1}{-g} \ln\left(\frac{T_1^\infty - T_0^\infty}{\Delta T^*}\right) \quad \text{with } g = \frac{-L_0 - L_1 T_1^\infty + L_1(T_{amb} - T_1^\infty)}{(L_0 + L_1 T_1^\infty)^2} + M \tag{22}$$

holds for the continuous mode model where N is computed for $S = 1$.

Proof At first note, that due to the monotonicity of the power/temperature mode according to Lemma 1, we find $\lim_{t\to\infty} T^*(t) \in [T_0^*(\tau), T_1^*(\tau)]$. In order to be independent of the actual rate function $S(t)$, we first determine the worst case rate function, i.e. which leads to the worst precision.

Suppose that we integrate (16) for a small time step δ and two different initial temperatures T_1 and T_2. With the resulting temperatures T_1' and T_2' we obtain

$$\frac{T_2' - T_1'}{T_2 - T_1} = 1 + \delta \frac{H(S, T_2) - H(S, T_1)}{T_2 - T_1}$$

For the active/idle model, we simply find that this term is $1 - \delta/C \cdot (G - \phi)$ where $\phi \in \{\phi^i, \phi^a\}$. This term is maximal for $\phi = \max\{\phi^i, \phi^a\}$, i.e. the constant rate associated to ϕ. For the continuous mode model we obtain the term $1 + \delta(M - \frac{L_0 + L_1 T_{smb}}{(L_0 + L_1 T_1)(L_0 + L_1 T_1)})$. As larger rates lead to a higher temperature gradient $H(S, T)$, the term is maximal for constant rate $S = 1$.

Now, let us prove (21). Using (9), we find

$$\Delta T^* = \left(T_1^\infty - T_0^\infty\right) e^{\frac{\phi - G}{C}\tau}$$

which directly leads to (21).

The proof for the continuous mode model follows. Let $T(T_s, \tau)$ be the resulting temperature at τ with starting temperature T_s for $S = 1$. When the starting temperature is T_1^∞, the temperature stays at T_1^∞. Thus,

$$\Delta T^* = T\left(T_1^\infty, \tau\right) - T\left(T_0^\infty, \tau\right) = T_1^\infty - T\left(T_0^\infty, \tau\right)$$

As it is hard to get the closed form solution for (12), we approximate $T(T_0^\infty, \tau)$ with an asymptotic line on dT/dt as shown in Fig. 3. With $F = \frac{d^2T}{dt^2}$, the approximated temperature $\tilde{T}$ holds following:

$$\frac{d\tilde{T}}{dt} = F\left(T_1^\infty\right)\left(\tilde{T} - T_1^\infty\right)$$

Since $\frac{dT}{dt} \geq \frac{d\tilde{T}}{dt}$ for $T, \tilde{T} \leq T_1^\infty$, $T(T_0^\infty, \tau) \geq \tilde{T}(T_0^\infty, \tau)$, $\tilde{T}$ converges to T_1^∞ slower than T. Using the solution for $\tilde{T}$ in form of (9), we have

$$\Delta T^* \geq T_1^\infty - \tilde{T}\left(T_0^\infty, \tau\right) = T_1^\infty - \left(T_1^\infty + \left(T_0^\infty - T_1^\infty\right) e^{g\tau}\right)$$

where g is $F(T_1^\infty)$. Some algebraic transformations with $g = \frac{-L_0 - L_1 T_1^\infty + L_1 (T_{amb} - T_1^\infty)}{(L_0 + L_1 T_1^\infty)^2}$ $+ M$ lead to (22). □

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

5 Experimental analysis

In this section, we will compare our worst-case analysis results with a random simulation of a basic real-time system. For simplicity, we illustrate our techniques assuming

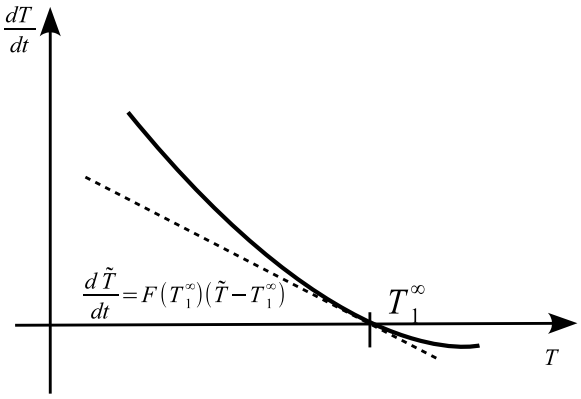

Fig. 3 Approximated temperature $\tilde{T}$ which has the same stable temperature T_1^∞ but converges slower

a set of processes that are periodic with jitter. Furthermore, we investigate the impact of different task invocation periods and jitter in task arrivals, and observe their relationship to the maximal temperature. Impact of variable resource availability on the temperature is also studied by allowing different service curves.

To ensure repeatability of our experiments and further investigations, our code has been integrated into the MPA-RTC toolbox and is available on-line at http://www.mpa.ethz.ch/rtctoolbox.

5.1 Benchmarks and basic configuration

In addition to the simple example from Sect. 1, a multi-processing video-conferencing system is considered, where three processes are executing on an ARM embedded processor. The system includes a video codec, an audio codec, and a network process which manages the communication, and for illustration it has been configured with a period-jitter-delay model, see Wandeler et al. (2006), Wandeler and Thiele (2008), with parameters summarized in Table 1.

In the example, the video codec operates in a range varying from 12 frames per second (fps) to 50 fps, being able to provide different output video qualities. This offers us the possibility to investigate a large range of invocation periods between 20 ms and 90 ms. For illustration purposes we set the audio codec to operate at a similar sampling rate, with an invocation period of 20 ms, which in the real system means just pre-processing (buffering) audio samples. Finally, the network process will be invoked as well with a period of 20 ms. We assumed the deadline of each task identical to its invocation period. For the exact meaning of all parameters in Table 1, please refer to Wandeler and Thiele (2008). System parameters are summarized in Table 2. Coefficients for the temperature dependent thermal resistance (R_0 and R_1) are taken from Walkey et al. (2001) and properly scaled to have the same thermal conductance as Yang et al. (2010) at 300 K. Power parameters are borrowed from (Rai et al. 2011) and scaled down to be proper according to (14). In all our experiments we start simulations with the initial temperature $T(0) = T_0^\infty = 319.31$ K, calculated from parameters given in Table 2 and the observation time interval $\tau = 1.2$ s, see Lemma 8. For the power mode in Eq. (6) and the continuous thermal model in Eq. (12) have been used for the simulation.

Table 1 Parameters of the video conferencing application

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

Table 2 Thermal and power parameters of the considered embedded system architecture

R_0	R_1	C	ϕ	ρ	ψ
$0.052\frac{1}{W}$	$0.0123\frac{1}{WK}$	$0.0218\frac{J}{K}$	$0.07\frac{W}{K}$	$9.8W$	$-17.5W$

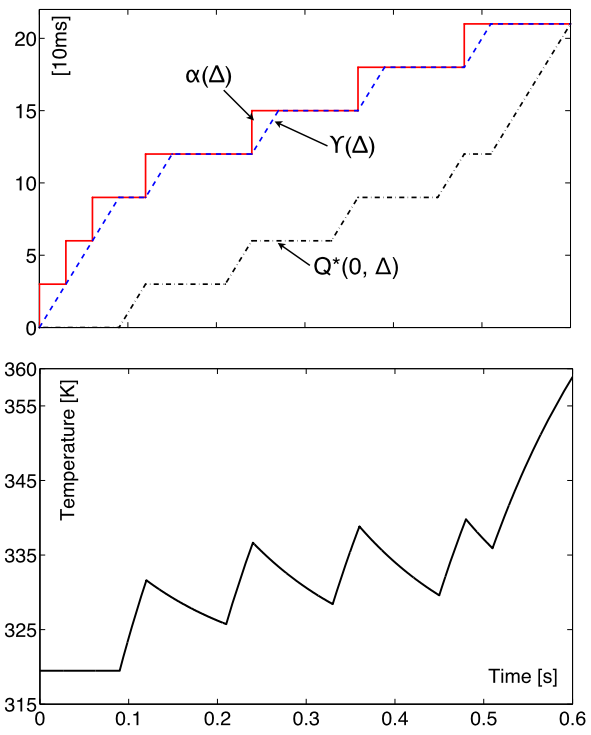

Fig. 4 Relevant analysis quantities related to the simple example from Sect. 1

5.2 Peak temperature analysis

At first, let us look again at the simple example from Sect. 1, i.e. a *single* task stream with period 120 ms, jitter 240 ms, computation time 30 ms and minimal inter-arrival time 30 ms on the fully serviced processor. The following Fig. 4 shows all relevant quantities, i.e. $\alpha(\Delta)$ according to (2), $\gamma(\Delta)$ according to (4), the worst-case accumu-

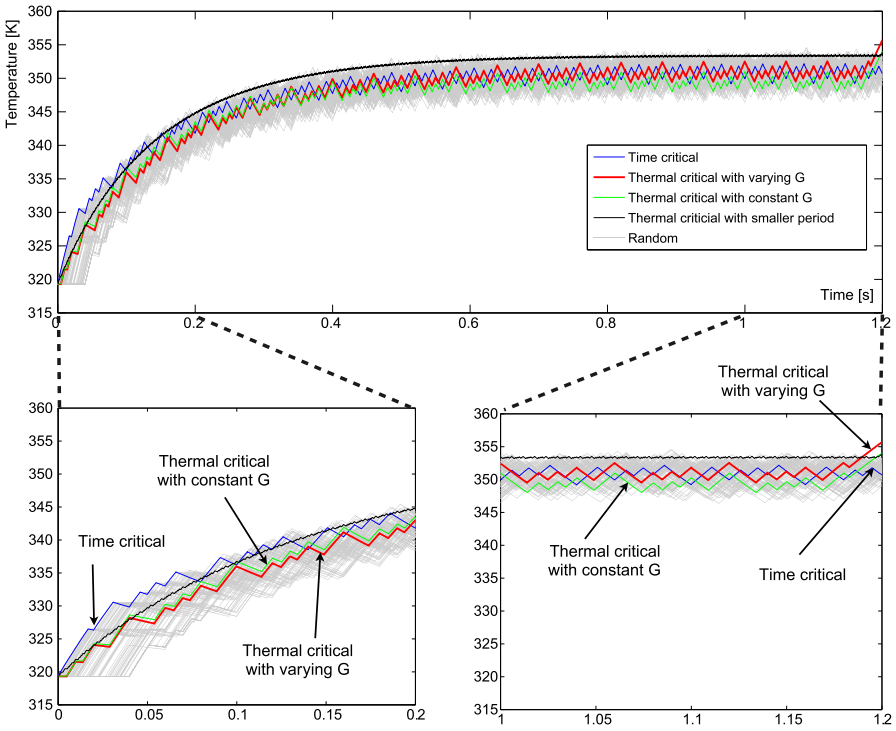

Fig. 5 Comparison between worst-case temperature estimate and other traces: (**a**) time critical instance, (**b**) thermal critical instance with temperature dependent thermal conductivity, (**c**) thermal critical instance with constant conductivity, (**d**) thermal critical instance with infinitesimally small period/jitter (varying conductivity), and (**e**) 100 random simulations

lated computing time $Q^*(0, \Delta)$ from Theorem 1 and the observation interval length $\tau = 0.6$ s. It is intuitively shown that the workload is placed as late as possible upto τ causing the bursty shot at the end. A part of the corresponding temperature trace and workload according to Lemma 7 has already been shown in Fig. 1, but using the observation interval $\tau = 1.2$ s.

The video conferencing example described in Table 1 is also analyzed in terms of peak temperature. The invocation period of video task is fixed to 50 ms in this experiment. For the comparison to a temperature simulation of random traces, we use 100 randomly generated task arrivals that conform to the workload specification. Figure 5 shows the correctness of the computed upper bound according to Theorem 1 and the limited coverage of random simulations.

Five different transient temperatures in the interval [0 s, 1.2 s) are drawn in Fig. 5: (a) the time critical instance (the workload trace that fits to the critical instant for timing analysis by releasing the workload as early as possible at the beginning), (b) the thermal critical instance with constant G, (c) the thermal critical instance (generated by Theorem 1) with variable G of (11), (d) the thermal critical instance with varying G and infinitesimally small period/jitter, and (e) 100 randomly generated workload traces. All traces start from the stead-state idle temperature $T_0^\infty = 319.49$ K. For (d),

Table 3 Bounds on worst-case peak temperature analysis for different τ values

τ	0.3 s	0.6 s	0.9 s	1.2 s	2.0 s
$T_{l=0}(\tau)$	350.794 K	354.853 K	355.535 K	355.652 K	355.681 K
$T_{l=1}(\tau)$	366.318 K	357.573 K	356.004 K	355.732 K	355.681 K

we change the period and jitter infinitesimally small while keeping the utilization the same.

The time critical instance has higher transient temperature before its first idle time than other traces. However, its temperature starts to decrease after that moment, and does not lead to the worst-case peak temperature. The highest temperature observed is 350.721 K. In contrast, the 100 random simulations might keep the system at higher temperature later on, but still does not capture the worst-case peak temperature. The highest peak temperature is 354.4 K for random simulations. As shown in Fig. 4, the thermal critical trace first warms up the system with periodic arrivals and then heats up the system with burst arrivals and jitters at the end around τ and the resultant worst-case transient temperature 355.652 K in Fig. 5.

How the bursty shot affect the worst-case peak temperature is well illustrated in Fig. 5(d). We make the period and jitter value infinitesimally small in (d), while keeping the total utilization of workload as the same as others. This eventually makes the trace equivalent to the average workload removing the bursty shot at the end. This is well shown on the temperature trajectory in Fig. 5(d) resulting in the worst-case temperature of 353.862 K.

The effect of temperature variant thermal conductivity is presented in the difference between (b) and (c). Note that we take the mean value of varying G as a constant value using $G_{const} = \frac{G(T_{amb})+G(T_1^\infty)}{2}$ and (10) is used to calculate the transient temperature. In case of constant conductivity, the worst-case temperature is far underestimated as 354.03 K.

The worst-case temperature $T^*(\tau)$, however, only gives the peak temperature of a feasible trace. As shown in Lemma 8, for estimating the worst-case peak temperature T^*, we also need the peak temperature at time τ, by starting at the possible highest temperature $T_{l=1}^\infty$ for the same trace. Table 3 demonstrates the temperature bound $[T_0^*(\tau), T_1^*(\tau)]$ by varying τ from 0.3 s to 2.0 s when task video task period and jitter are both 20 ms. Note that, for different values of τ, the worst-case traces are also different. When τ is small, the bound is not precise. For instance, the bound is [354.853 K, 357.573 K] in case of $\tau = 0.6$ s. However, when τ is getting larger, the bound tends to converge, with the precision of the 4th digit of decimal point, i.e., 355.6810 K. As a result, we can conclude that the worst-case peak temperature is 355.6810 K for the invocation period of the video task of 20 ms and when the initial temperature is not more than T_0^∞.

5.3 Worst-case temperature analysis under scheduling nondeterminism

In this subsection, we analyze the effect of changes in invocation periods and jitter on the maximal temperature. In addition, we provide hints on how to design a system

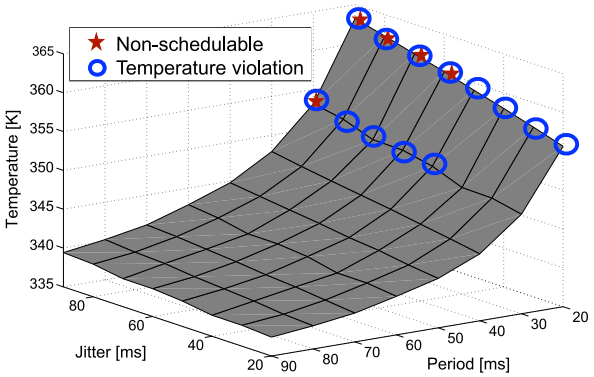

Fig. 6 Worst-case temperature function of both task invocation period and jitter. *Star markers* represent non-schedulable sequences, while *circle markers* highlight the cases that violate the temperature constraint of 350 K

which is schedulable and meets temperature constraints at the same time. As the maximal temperature depends on task arrivals but not on the scheduling policy, e.g. earliest deadline first (EDF), fixed priority, preemptive or non-preemptive, we restrict ourselves to EDF in the reminder of this section. Note that $\tau = 1.2$ s is carefully chosen to make precision loss less than 0.1 K according to Lemma 8.

5.3.1 Temperature-aware QoS optimization

The proposed analytic framework can be used to study the influence of critical design parameters at early design stages. As an example, for the video codec in Table 1, a shorter task period provides a higher quality of service (more frames per second). But, in general, shorter periods also lead to higher peak temperature as shown in the results in Fig. 6. Designers can quickly investigate the effect of such parameter changes by the proposed analysis framework and check if the current system configuration violates temperature constraints. In Fig. 6, an invocation period of 20 ms for instance would imply a high temperature of 360.18 K when the maximum jitter is 60 ms. For an invocation period of 40 ms, for instance, the video quality will be lower, but also the peak temperature will be as low as 346.09 K with the same jitter bound.

5.3.2 Changing task jitter

Besides varying the invocation period, we vary the maximal jitter in task arrivals. As shown in Fig. 6, we observe that a large jitter will increase the worst-case temperature. This is an expected qualitative behavior, since a large jitter increases the size of a burst of active modes in the power profile, thus inducing a higher temperature. If such a jitter leads to an unacceptable temperature, designers can redesign the system such at it reduces the jitter by introducing traffic shapers or other resource servers, see e.g. Wandeler et al. (2006).

5.4 Schedulability analysis

In addition, we run an EDF schedulability test by means of the RTC toolbox (Wandeler and Thiele 2008). It verifies that all output arrival curves satisfy the deadline,

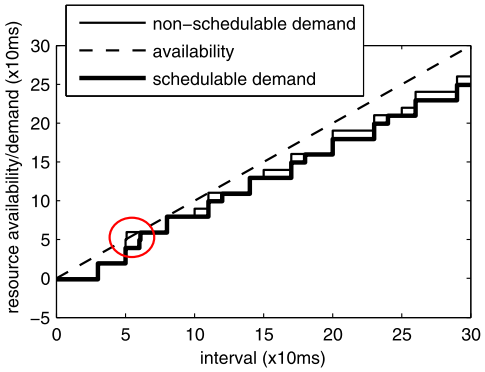

Fig. 7 Two examples of system traces and schedulability tests. *Thin solid line* represents a non-schedulable sequence that *crosses* the availability line at the location indicated by the *circle*. The *thick solid line* corresponds to a system which is schedulable

i.e. $\sum_i \alpha(\Delta - d_i) \leq \beta(\Delta), \forall \Delta$, with d_i the relative deadline, see e.g. Wandeler and Thiele (2006).

Figure 6 includes the results of such a schedulability test. For instance, for periods of more than 30 ms, independent of jitter, the system is schedulable. When the video task has the period of 30 ms, it is only schedulable when the maximum jitter is less than or equal to 50 ms, but only for very small jitter. Note that with a maximum acceptable temperature of 350 K, temperature constraints are met for all schedulable systems. All other configurations are not schedulable and they violate the temperature bound of 350 K.

For illustration, Fig. 7 provides an example of a demand bound function for the case where the system is schedulable, i.e. the sum of arrival curves shifted by the corresponding deadlines is smaller that the unit function Δ. Therefore, our analysis can be used as a tool to design systems which are at the same time schedulable and in a temperature-safe region.

5.5 General resource availability

The proposed technique can be also applied to more general system where the resource may not be always *completely* available for computation. Most modern embedded processors, for instance, support several operation frequencies and deep power down modes for power efficiency. In this subsection, we analyze the effect of different resource availabilities on the worst-case temperature.

Resource availability can be described using *service curves* as explained already in Sect. 3.1. Figure 8 shows several service curves as used in the experiments. The full service model ($\beta^u(\Delta) = \beta^l(\Delta) = \Delta$) as well as two lower resource availability curves ($\beta(\Delta) = 0.67$ or $0.33 \cdot \Delta$) are shown on the left-hand side. These service curves may correspond to operation frequencies of 100 %, 67 %, and 33 %, respectively. They may also be interpreted as the result of a general processor sharing (GPS) scheme where the processing resource is multiplexed between several consumers. The right hand side of Fig. 8 represents the service curve of a TDMA (time division multiple access) scheduling discipline. A TDMA resource is characterized by two variables: Cycle length C and Slot length S. For the service curve in the figure, for instance, we chose $C = 100$ ms and $S = 80$ ms, i.e. the processor is available for consecutive time units of 80 ms during every cycle of 100 ms.

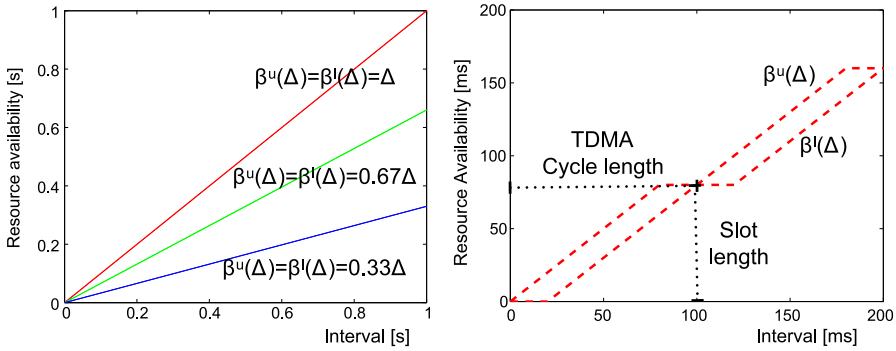

Fig. 8 Various resource availabilities: Frequency modulated processors (*left*) and TDMA (*right*)

For this set of computational models, we use the video conferencing example again, but with a different configuration. The period and maximum jitter of the video task are 60 ms and 20 ms, respectively. The observation time τ is set to 1.2 s. It is worth noting that the same supplied voltage is assumed for all resource availability configurations.

The first graph of Fig. 9 provides the comparison of the full service model to the partially utilized ones (67 % and 33 %). The steady-state temperatures in case of arbitrarily high workload are as follows: $T^{\infty} = 402.37$ K for $S = 1$, $T^{\infty} = 367.76$ K for $S = 0.67$, and $T^{\infty} = 340.63$ K for $S = 0.33$. On the other hand, the peak temperatures for the corresponding three service curves are 341.05 K, 339.54 K, and 338.15 K, respectively. The differences in the peak temperatures are not significant compared to that of the steady-state temperatures. When the resource is not sufficient to process the injected workload, the remaining workload is buffered in the waiting queue for further service. This buffering, in turn, causes a pervasive distribution of workload. In contrast, we have some *idle* intervals to cool down the system in case of a high processing rate. Moreover, the schedulability test fails in case of 33 % maximal frequency.

Supposing we need to keep the critical temperature 340 K unviolated for this system, the naive pessimistic analysis drives us to have a dynamic thermal management technique enabled without guaranteeing the real-timeliness. It has also been shown that none of the frequency modulation downto 33 % and 67 % guarantees the safe peak temperature. On the other hand, the estimates out of the proposed technique confirm that 66 % resource availability is a good compromise between the real-time and the peak temperature guarantees. In case of the critical temperature of 350 K, no thermal management technique is needed.

The bottom part of Fig. 9 shows the transient temperature changes when the TDMA scheduling discipline is adopted. We test two configurations: TDMA1 with $C = 100$ ms $S = 80$ ms and TDMA2 with $C = 50$ ms, $S = 40$ ms, i.e. half the cycle length but the same utilization. In both cases, only 80 % of the processing resource is consumed for processing the workload. Intuitively, the TDMA schemes may lead to a lower peak temperature in comparison to a full resource availability. However, TDMA1 not only shows a higher peak temperature (346.32 K) than the full service

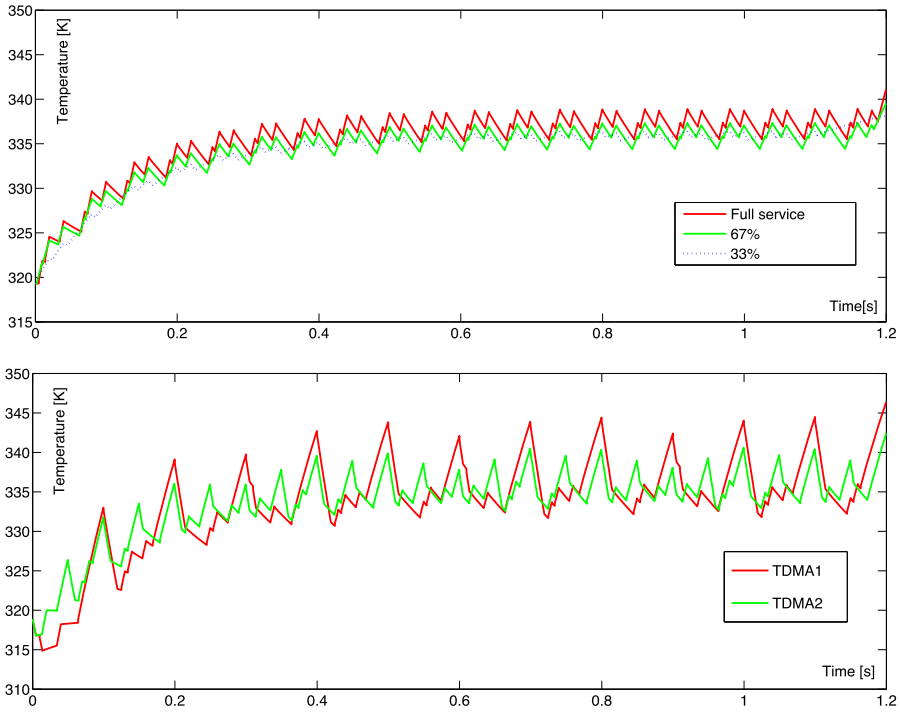

Fig. 9 Transient temperature changes of the video conferencing example. Period and maximum jitter are set to 60 ms and 20 ms, respectively

model but also fails to pass the schedulability test. Again, this is due to the previously mentioned buffering effect. During the idle service interval, the arrived workload is buffered in the queue causing a bursty resource utilization later on. If we reduce the idle service interval as in TDMA2, the peak temperature is reduced to 342.45 K and the system becomes schedulable.

In summary, two resource variations are analyzed in terms of peak temperature: reduce the *rate* of the resource and place idle service intervals properly (TDMA). The analysis framework presented in the paper can be used as a tool to choose proper resource parameters while guaranteeing schedulability and temperature bounds.

6 Conclusion

We have presented an analytical approach to determine the worst case temperature for any given work-conserving real-time system with general resource availability. We consider several power-temperature models that consider temperature dependent power consumption, leakage and thermal conductivity. They lead to non-linear differential equations that describe the relation between power consumption and temperature. The analysis method is able to deal with static and dynamic power consumption and takes into account its temperature dependence. The accumulated workload arriving 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. Analogously to the workload, we model the resource availability by service curves.

The complexity of the analysis comes from the fact that task arrivals and resource availability may be non-deterministic, i.e. due to unknown initial phases, jitter, burst, and provided computation. It is shown, that the method (a) gives tight upper bounds on the maximal temperature, (b) is constructive in the sense that the worst case arrival of tasks can be determined and (c) provides bounds on the length of the observation interval for a given precision.

As shown in the experimental results, the analysis method is a cornerstone to design real-time systems that give guarantees on (a) the schedulability *and* (b) on the maximal temperature. A direct impact of such a capability could be to avoid triggering *reactive control mechanisms* for hardware cooling, that are known to have high performance penalties, especially for high performance systems. Instead, one can formally analyze the influence of servers and traffic shapers in real-time systems, in terms of timing as well as maximal temperature.

Acknowledgements The work described in this paper has been funded by EU FP7 project EURETILE under grant number 247846 and partially supported by the TRANSCEND Strategic Action from NanoTera.ch.

References

Bansal N, Pruhs K (2005) Speed scaling to manage temperature. In: STACS

Bansal N, Kimbrel T, Pruhs K (2004) Dynamic speed scaling to manage energy and temperature. In: FOCS

Brooks D, Martonosi M (2001) Dynamic thermal management for high-performance microprocessors. In: HPCA '01: proceedings of the 7th international symposium on high-performance computer architecture. IEEE Computer Society, Washington, p 171

Brooks D, Tiwari V, Martonosi M (2000) Wattch: a framework for architectural-level power analysis and optimizations. In: ISCA '00: proceedings of the 27th annual international symposium on computer architecture. ACM, New York, pp 83–94

Chantem T, Dick RP, Hu XS (2008) Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. In: Design, automation and test in Europe

Chantem T, Hu XS, Dick RP (2009) Online work maximization under a peak temperature constraint. In: Henkel J, Keshavarzi A, Chang N, Ghani T (eds) ISLPED. ACM, New York, pp 105–110

Chen J-J, Hung C-M, Kuo T-W (2007) On the minimization of the instantaneous temperature for periodic real-time tasks. In: IEEE real-time and embedded technology and applications symposium

Chen J-J, Wang S, Thiele L (2009) Proactive speed scheduling for real-time tasks under thermal constraints. In: RTAS. IEEE Computer Society, Los Alamitos, pp 141–150

Fisher N, Chen J-J, Wang S, Thiele L (2009) Thermal-aware global real-time scheduling on multicore systems. In: RTAS

Fu X, Wang X, Puster E (2009) Dynamic thermal and timeliness guarantees for distributed real-time embedded systems. In: RTCSA. IEEE Computer Society, Los Alamitos, pp 403–412

Fu Y, Kottenstette N, Chen Y, Lu C, Koutsoukos X, Wang H (2010) Feedback thermal control for real-time systems. In: RTAS

Gomaa M, Powell MD, Vijaykumar TN (2004) Heat-and-run: leveraging SMT and CMP to manage power density through the operating system. In: ASPLOS-XI: proceedings of the 11th international conference on architectural support for programming languages and operating systems. ACM, New York, pp 260–270

Huang M, Renau J, Yoo S-M, Torrellas J (2000) A framework for dynamic energy efficiency and temperature management. In: International symposium on microarchitecture

Huang W, Skadron K, Gurumurthi S, Ribando RJ, Stan MR (2009) Differentiating the roles of IR measurement and simulation for power and temperature-aware design. In: ISPASS, pp 1–10

Hung W-L, Xie Y, Vijaykrishnan N, Kandemir MT, Irwin MJ (2005) Thermal-aware task allocation and scheduling for embedded systems. In: ACM/IEEE conference of design, automation, and test in Europe

Kumar P, Thiele L (2011) Cool shapers: shaping real-time tasks for improved thermal guarantees. In: Proc of design automation conference (DAC 2011). ACM, San Diego

Kumar A, Shang L, Peh L-S, Jha NK (2006) HybDTM: a coordinated hardware-software approach for dynamic thermal management. In: DAC, pp 548–553

Le Boudec J-Y, Thiran P (2001) Network calculus—a theory of deterministic queuing systems for the Internet. Lecture notes in computer science, vol 2050. Springer, Berlin

Liao W, He L, Lepak K (2005) Temperature and supply voltage aware performance and power modeling at microarchitecture level. IEEE Trans Comput-Aided Des Integr Circuits Syst 24(7):1042–1053

Liu Y, Dick RP, Shang L, Yang H (2007) Accurate temperature-dependent integrated circuit leakage power estimation is easy. In: DATE. EDA Consortium, San Jose, pp 1526–1531

Murali S, Mutapcic A, Atienza D, Gupta R, Boyd S, Micheli GD (2007) Temperature-aware processor frequency assignment for mpsocs using convex optimization. In: CODES/ISSS

Quan G, Zhang Y, Wiles W, Pei P (2008) Guaranteed scheduling for repetitive hard real-time tasks under the maximal temperature constraint. In: Gebotys CH, Martin G (eds) CODES/ISSS. ACM, New York, pp 267–272

Rai D, Yang H, Bacivarov I, Chen J-J, Thiele L (2011) Worst-case temperature analysis for real-time systems. In: Design, automation and test in Europe

Skadron BK (2004) HotSpot: thermal modeling. http://lava.cs.virginia.edu/HotSpot/index.htm [Online]. Available: http://lava.cs.virginia.edu/HotSpot/index.htm

Skadron K, Stan MR, Sankaranarayanan K, Huang W, Velusamy S, Tarjan D (2004) Temperature-aware microarchitecture: modeling and implementation. ACM Trans Archit Code Optim 1(1):94–125

Thiele L, Chakraborty S, Naedele M (2000) Real-time calculus for scheduling hard real-time systems. In: ISCAS, vol 4, pp 101–104

Walkey D, Smy T, MacElwee T, Maliepaard M (2001) Linear models for temperature and power dependence of thermal resistance in si, inp and gaas substrate devices. In: Seventeenth annual IEEE symposium on semiconductor thermal measurement and management, pp 228–232

Wandeler E, Thiele L (2006) Interface-based design of real-time systems with hierarchical scheduling. In: RTAS '06: proceedings of the 12th IEEE real-time and embedded technology and applications symposium. IEEE Computer Society, Washington, pp 243–252

Wandeler BE, Thiele L (2008). Real-Time Calculus (RTC) toolbox. http://www.mpa.ethz.ch/Rtctoolbox [Online]. Available http://www.mpa.ethz.ch/Rtctoolbox

Wandeler E, Maxiaguine A, Thiele L (2006) Performance analysis of greedy shapers in real-time systems. In: DATE, pp 444–449

Wang S, Bettati R (2006a) Delay analysis in temperature-constrained hard real-time systems with general task arrivals. In: RTSS

Wang S, Bettati R (2006b) Reactive speed control in temperature-constrained real-time systems. In: Euromicro conference on real-time systems

Wang S, Bettati R (2008) Reactive speed control in temperature-constrained real-time systems. Real-Time Syst 39(1–3):658–671

Wang Y, Ma K, Wang X (2009) Temperature-constrained power control for chip multiprocessors with online model estimation. In: Keckler SW, Barroso LA (eds) ISCA. ACM, New York, pp 314–324

Yang Y, Gu Z, Zhu C, Dick RP, Shang L (2007) Isac: integrated space-and-time-adaptive chip-package thermal analysis. IEEE Trans Comput-Aided Des Integr Circuits Syst 26(1):86–99

Yang C-Y, Chen J-J, Thiele L, Kuo T-W (2010) Energy-efficient real-time task scheduling with temperature-dependent leakage. In: ACM/IEEE conference of design, automation, and test in Europe (DATE)

Yao F, Demers A, Shenker S (1995) A scheduling model for reduced CPU energy. In: Symposium on foundations of computer science

Zhang S, Chatha KS (2007) Approximation algorithm for the temperature-aware scheduling problem. In: ICCAD

Hoeseok Yang is a postdoctoral researcher at Computer Engineering and Networks Laboratory (TIK) in ETH Zurich, Switzerland. He received the B.S. and Ph.D. degrees in computer science and engineering from the Seoul National University, Seoul, Korea, in 2003 and 2010, respectively. His current research interests include the design and optimization methodology of embedded systems and performance/thermal analysis of multiprocessor systems-on-chip.

Iuliana Bacivarov received the electrical engineering degree in 2002 from the National Polytechnic Institute of Bucharest, Romania. In 2002–2003, she received a master's degree in microelectronics integrated systems design from the Université Joseph Fourier in Grenoble, France, as well as a master's degree in quality and reliability engineering from the National Polytechnic Institute of Bucharest. She received her Ph.D. degree in microelectronics from the National Polytechnic Institute of Grenoble in 2006. She has been a post-doctoral researcher at the Computer Engineering and Networks Laboratory of ETH Zurich since 2006. Her research interests include design, analysis, and optimization of MPSoC.

Devendra Rai is a Ph.D. student at ETH, Zurich, and is interested in the general area of multiprocessing, including design of frameworks, and related thermal analysis. Devendra Rai earned an M.S. from the University of Virginia (2009), and is a member of the Eta Kappa Nu (HKN) honor society.

Jian-Jia Chen joined Department of Informatics at Karlsruhe Institute of Technology (KIT) in Germany as a Juniorprofessor for Institute for Process Control and Robotics (IPR) in 2010. He received his Ph.D. degree from Department of Computer Science and Information Engineering, National Taiwan University, Taiwan in 2006. He received his B.S. degree from the Department of Chemistry at National Taiwan University 2001. After finishing the compulsory civil service in Dec. 2007, between Jan. 2008 and April 2010, he was a postdoc researcher at Computer Engineering and Networks Laboratory (TIK) in ETH Zurich, Switzerland. His research interests include real-time systems, embedded systems, energy-efficient scheduling, power-aware designs, temperature-aware scheduling, and distributed computing. He received Best Paper Awards from ACM Symposium on Applied Computing (SAC) in 2009 and IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA) in 2005.

Lothar Thiele joined ETH Zurich, Switzerland, as a full professor of computer engineering in 1994, where he currently leads the Computer Engineering and Networks Laboratory. He received his DiplomIngenieur and Dr.-Ing. degrees in Electrical Engineering from the Technical University of Munich in 1981 and 1985 respectively. His research interests include models, methods and software tools for the design of embedded systems, embedded software and bioinspired optimization techniques. Lothar Thiele is associate editor of IEEE Transaction on Industrial Informatics, IEEE Transactions on Evolutionary Computation, Journal of Real-Time Systems, Journal of Signal Processing Systems, Journal of Systems Architecture, and INTEGRATION, the VLSI Journal. In 1986 he received the "Dissertation Award" of the Technical University of Munich, in 1987, the "Outstanding Young Author Award" of the IEEE Circuits and Systems Society, in 1988, the Browder J. Thompson Memorial Award of the IEEE, and in 2000–2001, the "IBM Faculty Partnership Award". In 2004, he joined the German Academy of Sciences Leopoldina. In 2005, he was the recipient of the Honorary Blaise Pascal Chair of University Leiden, The Netherlands. Since 2009 he is a member of the Foundation Board of Hasler Foundation, Switzerland. Since 2010, he is a member of the Academia Europaea.