A Unified Methodology for Scheduling in Distributed Cyber-Physical Systems

QINGHUI TANG
Texas Instruments
SANDEEP K. S. GUPTA
Arizona State University
and
GEORGIOS VARSAMOPOULOS
Arizona State University

A Distributed Cyber-Physical System (DCPS) may receive and induce energy-based interference from and to its environment. This paper presents a model and an associated methodology that can be used to: i) schedule tasks in DCPSs to ensure that the thermal effects of the task execution are within acceptable levels; and ii) verify that a given schedule meets the constraints. The model uses coarse discretization of space and linearity of interference. The methodology involves characterizing the interference of the task execution and fitting it into the model, then using the fitted model to verify a solution or explore the solution space.

Categories and Subject Descriptors: C.4 [**Performance of Systems**]: Design studies; C.4 [**Performance of Systems**]: Modeling techniques; D.4.1 [**Operating Systems**]: Process Management—*Scheduling*

General Terms: Design; Management; Performance; Theory; Verification

Additional Key Words and Phrases: abstract heat flow model, thermal management, thermal awareness, scheduling, cyber-physical systems, sustainable computing, green computing

1. INTRODUCTION

In recent years, research and technologies developed in the fields of computer science and engineerign have made vast progress. Pervasive and autonomous computing have been increasingly introduced to real applications. Smart and multi-functional electronic devices with various computation capabilities are used in all aspects of daily life. Driven by new demands and applications, computing is changing human life with higher mobile, ubiquitous, uninterrupted, and embedded capabilities [Adelstein et al. 2005]. Therefore, a new kind of computing system has emerged, known as Cyber-Physical System (CPS, also *DCPS* if distributed) [Sztipanovits 2007; Lee 2008; Wolf 2009], which is the integration

This work was supported in part by grants from Intel Corporation, Science Foundation Arizona and National Science Foundation (CNS #0649868, #0649868, #0831544, #0834797, and #0855277). Qinghui Tang is now with Texas Instruments, Austin, TX (this work was done when the author was with the Impact Laboratory at ASU). E-mail: jeffrey.tang@ti.com. Sandeep K. S. Gupta and Georgios Varsamopoulos are with the Impact Laboratory, School of Computing and Informatics, Ira A. Fulton School of Engineering, Arizona State University, Tempe, AZ 85287-8809. E-mail: {sandeep.gupta,georgios.varsamopoulos}@asu.edu. `http://impact.asu.edu/`.
Permission to make digital/hard copy of all or part of this material without fee for personal or classroom use provided that the copies are not made or distributed for profit or commercial advantage, the ACM copyright/server notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or a fee.
© 2010 ACM 1529-3785/2010/0700-0001 $5.00

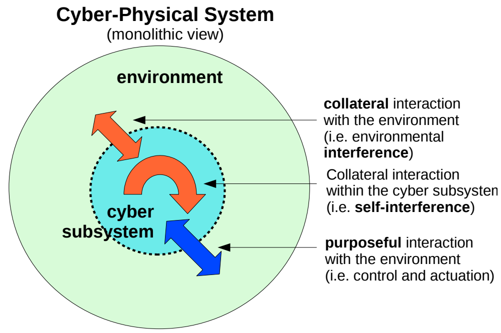

Fig. 1: Conceptual depiction of an Environmentally-Coupled Cyber-Physical system (ECCPS). The ECCPS may have a purposeful interaction with the environment, i.e. controlling the environmental conditions through sensors and actuators. However, being tightly coupled with its surrounding environment, it will also have a collateral interaction with itself and the environment, referred to as *self* and *environmental interference*. The work in this paper focuses on characterizing the interference and taking it into consideration when scheduling computation.

of computing systems with physical processes and a physical environment. Two major motivating factors behind the notion of CPS is the need to design and produce reliable and sustainable computing systems that work in harmony with their surroundings.

As shown in Figure 1, CPS may be used to monitor or control physical environments. This work, however, focuses on the type of CPSs that are tightly coupled with the surrounding environment so that they affect each other's operation and condition *beyond* the intended interaction, usually through the exchange of some form of energy (heat, sonic, electromagnetic etc.). Such a system is referred to as an ***Environmentally Coupled Distributed Cyber-Physical System*** (ECDCPS).

The concept of ECDCPS is not entirely new; for example, it shares similarities with *embedded systems*. A type of embedded DCPS are biosensor networks implanted in human tissue, aimed for monitoring and possibly controlling biological functions [Schwiebert et al. 2001]. An embedded biosensor network may *interfere* with body functions other than the ones that the network is intended for.

The notion of interference extends beyond the interference from and to the environment. For example, large data centers consolidate computing and data storage into one room. Each computing server in a data center produces heat which is recirculated in the room and *interferes* with the thermal levels and operation of potentially all servers, effectively increasing the energy consumption [Tang et al. 2008].

A CPS's interference with the environment is not restricted to heat only; also, it is not limited to embedded computing applications or large scale computing sites. For example, a power grid can be considered as a DCPS that is coupled with the environs it crosses, affecting the surrounding life and landscape (through noise and E/M radiation) and being affected by the weather. In general, if the DCPS generates a form of disturbance to the hosting environment, or if its operations can potentially be disturbed by the environment, it can be examined as an ECDCPS. Such an ECDCPS should be able to minimize the effects of this interference and achieve an enduring "balance" with the environment.

The work in this paper builds on the basic idea that the interference principally derives from the operation of the DCPS, which in turn can be managed through scheduling of the operation's tasks among the distributed nodes (the model allows for accounting for external agents). The paper provides a framework, i.e. an abstract heat flow model and a generic methodology, on how to identify and characterize such interference and how to use this interference to formulate a scheduling problem that either minimizes the interference or keeps it with specified constraints. The scheduling problem formulation can then be used to either find a schedule that meets the constraints or verify that a given schedule does so.

1.1 Related work

From a scheduling perspective, CPSs have been considered almost entirely as time-constrained or resource-constrained systems [Jiang et al. 2008; Xue et al. 2009; Porter et al. 2009]. To that extent, it is suggested to enhance computing architecture abstractions so as to include time [Lee 2006; 2008]. With the exception of control-oriented studies [Zhang et al. 2008], CPS scheduling work does not take the physical interactions and sensed information into account.

Another active topic of research is the modeling and specification of CPS. Compared to related work on scheduling, research on modeling is closer to the physical aspects of CPSs: [Bujorianu et al. 2009] provide a specification logic for the behavior of human-centric CPSs in environments with uncertainty; [Ilic et al. 2008] propose a high-level modular model of the dynamics of electric power systems (i.e., power plants and grids); in a similar manner, [Sentilles et al. 2008] propose another modular model oriented towards embedded control systems; lastly, [Tan et al. 2009] propose a spatio-temporal event model that allows physical and computing events to be represented in an integrated manner.

The above efforts focus on modeling the intended operation and interaction of a CPS and its components. Modeling the *casual* (or *collateral*) interference [Gill and Niehaus 2006] among the components of a CPS, or its interference with the environment has received little attention. [Sun et al. 2007] propose a methodology to verify the non-interference of components in an power grid. Another related effort on characterizing the physical interference is the *thermal mapping* work of the Weatherman project [Moore et al. 2006].

In most of the previous work on thermal awareness, reducing interference was an independent and local-view goal for each node separately, i.e. it is not done from a perspective of the entire system. For example, the work of Multicore Thermal Management [Donald and Martonosi 2006] tries to achieve thermal management through changing voltage or migrating process among multiple cores without considering what happens at other systems. Another similar work [Springer et al. 2006] tries to minimize the execution time while satisfying time and energy constraints based on computing clusters capable of voltage and frequency scaling (DVFS).

This paper complements the related work on CPS by providing an abstract heat flow model that describes thermal interference in a CPS and using that model to derive an interference-aware scheduling and verification methodology. This paper draws lessons learned from previous work [Tang et al. 2005; Tang et al. 2006; Tang et al. 2007; Tang et al. 2008], where thermal-aware scheduling problems in ECDCPS were studied. This paper proposes a methodology that builds on the common aspects of the previous work efforts, and contrasts the differences to show how diverse the field of thermal-aware scheduling in ECDCPS is.

1.2 Overview of intellectual merits

This work proposes an abstract heat flow model and a generic methodology that consider the scheduling of computational tasks in a system from a cyber-physical point-of-view, i.e., it considers both the computational performance and the thermal interference in a unified manner. The model organizes the representation of interference by discretizing the interference space of the CPS and by providing a linear mapping between the interfering points. The methodology involves four steps: i) characterizing the correlation between a task's execution and its resulting power consumption as a function; ii) characterizing the interference using the aforementioned abstract heat flow model; iii) formalizing the objective function and constraints; and iv) exploring the solution space and find the best schedule, in terms of minimizing the effects of interference.

The methodology is then applied to two separate and diverse domains: a) temporal scheduling of communication in an in-vivo biosensor network, and b) spatial scheduling of computation in a data center. In the problem of scheduling the biosensor communication, the scheduling approach reduces the thermal impact to human tissues and achieves more balanced power consumption. In the problem of scheduling computation in a data center, the approach reduces the heat recirculation and lowers the cooling energy requirements. Lowering the energy requirements in data centers, including lowering the cooling requirements, has been a sought-after objective of green computing. To summarize, the overall contributions of this paper are:

—An *abstract heat flow model* for ECDCPS; the model can be used to identify the potential hot spots and estimate the thermal performance of the ECDCPS in a more efficient and faster manner. The abstract model is then used to define the scheduling problem.

—Formalization of an *unified methodology* of analyzing and designing an ECDCPS with minimized thermal interference, based on the *abstract heat flow model*.

—*Revisiting* of two thermal-aware application issues, namely, thermal aware communication scheduling of biosensor networks [Tang et al. 2005] and thermal-aware task placement within data centers [Tang et al. 2008], to demonstrate the capabilities of the abstract heat flow model and the methodology.

The proposed methodology aims to provide an improvement in designing and verifying environment-friendly and sustainable CPSs. It will allow the development of the CPS with capabilities such as low overhead, energy-efficiency, and interference minimization. Minimal interference and good energy-efficiency are cornerstones of *green computing*.

The rest of the paper is organized as follows: Section 2 provides term definitions and a description of the problem. Sections 3 and 4 describe the abstract heat flow model and the unified methodology for thermal-aware scheduling, respectively. Section 5 describes the application of the model and unified methodology to the example applications of leader rotation in tissue-embedded biosensor networks, and batch scheduling in data centers, and discusses the similarities and differences of the two problem domains. Lastly, Section 6 concludes the paper and provides future research directions.

2. SYSTEM MODEL AND PROBLEM DESCRIPTION

This section presents some general concepts and definitions pertinent to ECDCPS, and provides a description of the problems caused by interference, discussing the background and the reasons for our focus on thermal related applications.

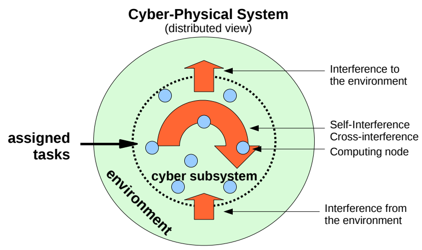

Fig. 2: Depiction of an environmentally coupled distributed cyber-physical system (ECDCPS) and the notion of *cross-interference*, which is specific to the distributed nature of the system. The execution of tasks by the ECDCPS creates interference to and from the nodes, noted as self-interference and cross-interference, and to the environment.

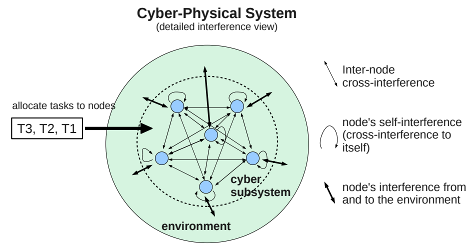

Fig. 3: The interference is decomposed and discretized to cross-interference among each pair of nodes, each node to itself and each node with the environment. With this decomposition, it is easier to model, analyze and predict the effects of interference.

2.1 System model and definitions

Hosting environment (in short, ***environment***) is defined as the physical place where a CPS is installed and operating, as well as the natural pattern of activities conducted in the environment. The environment can be a natural or a constructed area. ***Control volume*** is the three-dimensional space of the environment which the interference can impact, including the contained objects.

Although the notion of CPS may prompt toward environment monitoring and controlling, the authors stress the notion that CPS may be a system whose operation may casually or collaterally affect or be affected by the environment. Therefore an ***environmentally coupled CPS*** (ECCPS) is defined to be any CPS whose primary operation is not to control the environment but nevertheless has a considerable impact to it, i.e. in some perspective it is *invasive*. *Distributed* ECCPSs (i.e., ECDCPSs) consist of distinct and autonomous computing ***nodes***. This work considers an ECDCPS whose nodes are physically static with respect to themselves and their immediate environment.

Interference is the casual[1] exchange of some form of energy among the nodes and the environment, which has the potential to affect the operation of the ECDCPS or the conditions of the environment. Considering the structural decomposition of an ECDCPS, the interference can be classified to the following types:

—***Cross-interference***: this is defined as the interference among nodes of a ECDCPS. One example of interference is the increased probability of failure of an overheated caused by heat arriving from other servers. Another example, mentioned in [Fulford-Jones et al. 2004; Dutta et al. 2005], is that radio transmission can cause interference in circuitry. A third example is from [Arora et al. 2004], where sensors with different modalities are deployed to sense magnetic, acoustic, seismic, and optical data. The existence and operation of one optical sensor may influence the reading of another magnetic sensor.

[1]This paper is limited to the casual interference effects caused by the normal operation of individual nodes and normal environmental conditions. The interference and impacts due to abnormal and unpredictable operation and behavior, such as hardware malfunctions, calamities and other *criticalities* [Mukherjee and Gupta 2009] are not considered.

—***Self-interference***: the cross-interference of a node onto itself. The operation of a node may degrade its own performance or lead to a loss of functionality. In monolithic, centralized CPSs, the cross-interference reduces to self-interference only, as there is only one component.

—***Environment interference***: the interference of the system to the environment and vice versa. Environment interference is distinguished into *interference to the environment* and *interference from the environment*.

Condition variable is an observable, and desirably predictable, quantity that describes one aspect of the system's or environment's condition. Examples of condition variables are temperature, pressure or sound level. Condition variables are associated with a spatial point in the control volume. A model that describes how a condition variable's values are affected with respect to the distributed energy consumption of the ECDCPS is referred to as ***interference model***, and a condition variable is then referred to as ***interference effect*** variable. Also, there are certain ***safety constraints*** that are associated with a condition variable (usually expressed as upper or lower value limits).

The ECDCPS is assigned to execute a set of ***tasks***. The tasks can be assigned to any node and at any time within the given ***computational constraints***. The execution of each task causes interference, which can vary depending on the location and time of the execution. In addition to the constraints, there usually is an ***optimization objective***. Depending on whether the optimization is oriented toward computing performance or toward energy efficiency, a schedule will have to optimize a metric such as *throughput*, *makespan* (i.e., the length of the schedule), *total energy consumed*, or *average power*.

2.2 Problem description: thermal-aware spatio-temporal scheduling

Although the operation of an ECDCPS creates interference in various forms, including sonic, mechanical, thermal, and electromagnetic, this paper focuses on heat-related interference, as thermal properties are important design considerations in any computing system. The interference-aware scheduling problem in this paper can be defined as:

> **Given an ECDCPS and a set of tasks to be executed, how can a spatio-temporal schedule of those tasks be verified not to exceed the acceptable limits of thermal condition variables? Further, how can a spatio-temporal schedule be identified so that the performance is maximized while the operation of the ECDCPS satisfies the performance or thermal safety constraints?**

The above problem involves solving important subproblems, including:

—How can the heat generated by nodes when they run the application's tasks be modeled?

—How can the interference caused by the execution of a task at a node at a specific time be characterized and modeled?

—How can the execution of tasks be related to their effect on condition variables?

To answer the above questions, the work in this paper proposes an abstract thermal interference model and a unified methodology for ECDCPS in Section 3 below.

3. AN ABSTRACT MODEL FOR SCHEDULING IN ECDCPS

As mentioned previously, the ECDCPS is composed of a set of distributed nodes from 1 to n: CPS $= \{\text{node}_1, \text{node}_2, \ldots, \text{node}_n\}$, at specific locations $\text{loc}_1, \text{loc}_2, \ldots, \text{loc}_n$. These

Table I: Nomenclature

Symbol	Definition
CPS	The system's set $\{\text{node}_1, \ldots, \text{node}_i, \ldots, \text{node}_n\}$ of n CPS nodes
$\{loc_i\}$	The set of the corresponding n locations of the nodes
C	The set $\{T_1, \ldots, T_j, \ldots, T_m\}$ of m tasks
A_j	The node i, start and end time $(t_{j,\text{s}}, t_{j,\text{f}})$ assignment of task T_j, i.e. $\langle T_i, \text{node}_i, (t_{j,\text{s}}, t_{j,\text{f}})\rangle$.
$\mathcal{S}$	The *task* schedule $\{A_j\} = \{\langle T_i, \text{node}_i, (t_{j,\text{s}}, t_{j,\text{f}})\rangle\}$ of all task assignments
$G_i(T_j)$	The function $G : \text{CPS} \times C \rightarrow \mathcal{G}$ that maps an assignment to the power dissipation function $G_{i,j}(t)$ of node i when running task T_j
$\mathcal{S}_G(C)$	The *power* schedule $\{\langle G_i(T_j), (t_{j,\text{s}}, t_{j,\text{f}})\rangle\}$ that results from $\mathcal{S}(C)$
$\mathcal{G}$	The set of all possible power dissipation functions $G(t)$
$\mathcal{F}$	The set of all possible condition variable functions $f(t)$
F	The interference effect function $F : \left(\mathcal{G} \times \mathbb{R}^3\right)^n \times \mathbb{R}^3 \rightarrow \mathcal{F}$, which maps the power functions from each node and their locations to the function of the condition variable at a specific location

nodes may be heterogeneous in nature, but of compatible instruction set architectures and system call APIs. There are m incoming tasks $C = \{T_1, T_2, \ldots, T_m\}$ to be assigned. Let G : CPS$\times C \rightarrow \mathcal{G}$ be the mapping between task execution at nodes and the energy consumption rate. For example, a node i will consume energy at a rate of $G(\text{node}_i, T_j) = G_{i,j}(t)$ during the execution of task T_j (t is time); the power consumption $G_{i,j}(t)$ depends both on the hardware characteristics and the task's computational characteristics.

A schedule $\mathcal{S}$ is a set of spatio-temporal assignments A_j among tasks and nodes, i.e.:

$$\mathcal{S}(C) = \left\{ A_j = \langle T_j, \text{node}_i, (t_{j,\text{s}}, t_{j,\text{f}})\rangle \mid i \in \{1 \ldots n\}, j \in \{1 \ldots m\} \right\},$$

where $(t_{j,\text{s}}, t_{j,\text{f}})$ is the pair of start and finish times. Using the power consumption functions $G_{i,j}(t)$, the power dissipation of the schedule can be denoted as:

$$\mathcal{S}_G(C) = \left\{ \left\langle G_{i,j}(t), (t_{j,\text{s}}, t_{j,\text{f}}) \right\rangle \mid A_j \in \mathcal{S} \right\}.$$

The consumed energy will be manifested almost entirely as heat in the environment, primarily because of the electrical resistance of the ECDCPS's circuitry and secondarily because of electromagnetic absorption.

The *condition variable* (i.e., *interference effect* variable) is energy-related, whose distribution inside the control volume, including the temperature of each node, depends on the overall power consumption of each node and on how energy is propagated in the control volume. We denote as $F : \left(\mathcal{G} \times \mathbb{R}^3\right)^n \times \mathbb{R}^3 \rightarrow \mathcal{F}$ the generic *interference effect* function that maps the power consumption of the n nodes, each at some Euclidean coordinate loc_i, to the function of a condition variable $f(t)$ (e.g. temperature) at a point (x, y, z) the Euclidean control volume. Note that the interference effect function combines both the interference model, i.e. how the energy is distributed in the control volume, and the conversion from energy to the chosen condition variable.

The *computational performance constraints* can be either formulated as brute-force constraints, e.g. "$t_f(A_j) - t_s(A_j) < 1$ s" if a task j's runtime must be less than one second, or as constraints on objective functions such as the *makespan*. The formulation of the *safety constraints* depend on the definition of F; in the most general case, they can be expressed as a polytope $\Pi \subset \mathbb{R} \times \mathbb{R}^3$ which poses limits to the values of F at each location.

These definitions are generalized, allowing for continuous time or space and delays of heat transfer, but they are difficult to use "as is". They should be adapted to the nature of the scheduling problem and its associated CPS. For example, if the power consumption is constant at each node, and if we are not interested in the exact coordinates of the nodes, then F may be defined as a discrete-space time-invariant function that maps a power vector ($P = \langle P_1, P_2, \ldots, P_n \rangle$) to a converging (i.e., $t \rightarrow \infty$) temperature distribution of k select points (i.e., $F : \mathbb{R}^n \rightarrow \mathbb{R}^k$). In this case, the constraints can be expressed as two real vectors in $\mathbb{R}^k$ giving the lower and upper limits for F.

Given the well-known general trade-off between performance and energy, the common practice is to constraint one of the two and optimize the other, e.g., pose an upper constraint to the schedule's makespan and minimize the total energy consumption. The optimization function of the schedule is denoted as $H(\mathcal{S})$ and conventionally it is formulated as an expression to be minimized (as opposed to be maximized).

With the above definitions, a scheduling problem can be formally expressed in Pinedo $\boldsymbol{\alpha \mid \beta \mid \gamma}$ notation[2] [Pinedo 2008] as

$$\mathrm{R}n \;\Big|\; \forall (x, y, z), F(\mathcal{S}_G, \{\mathrm{loc}_i\}, x, y, z) \in \Pi \;\Big|\; H(\mathcal{S}), \tag{1}$$

where Rn is the "unrelated machines in parallel" computational environment [Pinedo 2008].

Small example: consider a small marine wireless sensor network of three nodes (n=3, $i \in \{1, 2, 3\}$) at certain coordinates loc_i. There are three perpetual roles (m=3, $j \in \{1, 2, 3\}$)to be assigned with the power function being $G_{i,j}(t) = i \cdot j$ watts (i.e., time-invariant). Assume an isotropic propagation model with a path loss exponent of 3; the condition variable is the power density. Then

$$F(\mathcal{S}_G, \{\mathrm{loc}_i\}, x, y, z) = \sum_{A(T_j, \mathrm{node}_i) \in \mathcal{S}} \frac{i \cdot j}{4\pi \ \mathrm{distance}^3(\mathrm{loc}_{x,y,z}, \mathrm{loc}_i)}.$$

For marine life safety, the power density levels must be under a certain threshold Th, for example 0.1 W/m^2. We are asked to find a schedule that respects these levels and minimizes the average power consumed. The scheduling problem then translates to:

$$\mathrm{R}_3 \;\Big|\; \text{exclusive}; \forall (x, y, z), F(\mathcal{S}_G, \{\mathrm{loc}_i\}, x, y, z) < \mathrm{Th} \;\Big|\; \sum\nolimits_{A(T_j, \mathrm{node}_i) \in \mathcal{S}} G_i(T_j),$$

where the *exclusive* keyword means that no node can run two or more tasks at the same time.

3.1 The abstract heat flow model (AHFM)

Most of the time, the thermal performance of a distributed system is of interest only at certain key locations (for example, the inlets of the server nodes). Therefore, a detailed and high-granularity temperature evaluation for every point inside control volume is not necessary; the dissipation and exchange of heat inside a distributed system can be represented as a discrete, light-weight *flow model*.

3.1.1 *Discretization of space: points of interest.* A straight-forward first-step is to discretize the model with respect to the *points of interest*. A conceptual model is shown in Figure 4: the scheduler decides how the tasks are assigned, the tasks' execution at the nodes produces heat which is then "fed" as input to the interference effect function, which in turn can be calculated for specific chosen points of interest, such as the nodes themselves or other sensitive physical entities or locations. In this model, one can distinguish a column of heat sources on the left (called ***victors***), and a column of heat receptors on the right (called ***victims***). The model also makes it easy to distinguish between cross-interference, shown in the upper part, and environmental interference, shown in the lower part.

[2]The notation $\alpha \mid \beta \mid \gamma$ is read as *computational environment class* | *constraints* | *optimization objective*.

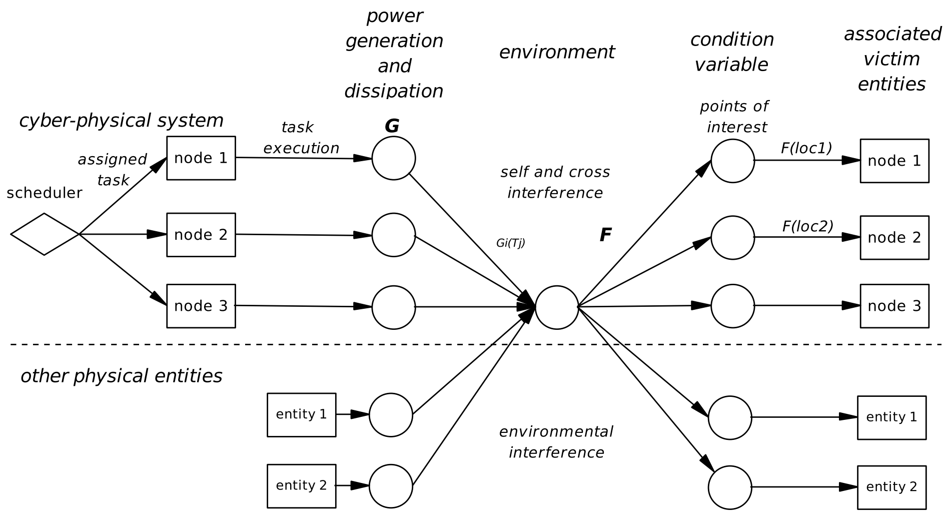

Fig. 4: A heat flow model of an Environmentally Coupled Distributed Cyber-Physical System as it is derived from the definitions in §3: the nodes consume energy when performing assigned tasks, according to the power function G, then it will be distributed and manifested as the condition variable (temperature) according to the interference F at select discrete points.

It may be possible that function F does not have an exact analytical representation. Conventionally, time-consuming numerical/computational methods are used to calculate the value of F *for one instance* of its arguments, an approach which is unsuitable for on-line decision making such as scheduling.

Numerical methods such as finite-difference time-domain (FDTD) or computational fluid dynamics (CFD) are frequently used to thermally evaluate electrical and computer systems. These methods can thoroughly and accurately simulate the thermal properties of the control volume, and can obtain the temperature distribution with a high accuracy at a fine resolution. Nevertheless, these numerical methods have the *limitation* of discretizing the entire control volume into small grids in order to achieve a smaller granularity. Consequently, a large and complicated control volume can result in a very long solution time, that can take days or even weeks.

The discussion in the subsection below leverages the coarse space discretization of the points-of-interest, and the conservation of energy to design a linear heat flow model.

3.1.2 *Discretization of function F: linear heat flow.* The main purpose of a thermal model is to describe how heat is propagated. Knowing that heat is contributed independently by each node of the ECDCPS, and potentially from other known physical entities (e.g. the sun), the heat produced by each node or entity is divided into parts and redistributed through the environment. The heat at any particular point is the sum of those portions of heat from each source. These portion coefficients can be organized into an *interference matrix* $\boldsymbol{D} = \{d_{ik}\}$, where a coefficient d_{ik} denotes what portion of the heat from victor i goes to victim k. Note that $\sum_i d_{ik}=1$ only when the ECDCPS is a closed system.

Figure 5 shows a modification of the conceptual model that captures the division and aggregation of heat. For each victor, the heat (P_i) is split according to the interference

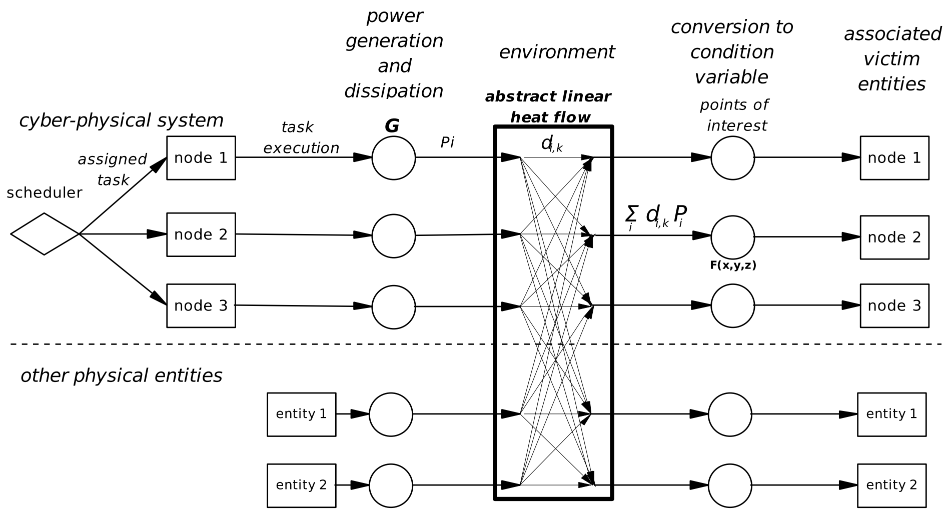

Fig. 5: The proposed Abstract Heat Flow Model of an ECDCPS: Heat distribution is internally modeled as a linear distribution matrix $\boldsymbol{D}$. Each matrix element represents how much heat is transferred from one select *victor* point to another *victim*. The heat at any victim point is the weighted sum of incoming heat.

coefficients d_{ik}. The heat Q_k that aggregates at a victim k is the weighted sum of the heat portions from each heat source:

$$Q_k = \sum_i d_{ik} P_i.$$

With the heat determined, one can calculate any heat-based condition variables. For example, a *temperature rise* can be obtained from heat using the formula

$$\Delta T_k = \text{specific heat} \times \text{mass} \times Q_k = \varphi_k \; Q_k,$$

where φ_k is the heat capacity of the victim point. The matrix $\boldsymbol{D}$ relates with the interference effect function as:

$$F(P, k) = \varphi_k Q_k = \varphi_k \sum_i d_{ik} P_i.$$

There are several possible scenarios regarding the flow of heat among distributed nodes and cooling systems. First, the system may be open or closed, which affects the interference matrix $\boldsymbol{D}$. Secondly, the focus of interest may be on the effect on the nodes themselves, or on the effect on the environment. Depending on what the focus is, various victim entities may be omitted.

The heat flow rates of our abstract model are deduced from numerical simulations or real measurements. Our model abstracts the system at a higher level, considering the thermal effects due to the components' characteristics and the properties of the performed tasks. The computational benefit of this model is that it can be reused to explore the effects of many schedules, thus it can be used to search for a good schedule or to verify a given one.

The model is highly customizable; and it can easily identify the potential thermal effect on the system, such as the occurrence of hot spots due to insufficient cooling or due to continuous long time heat dissipation, or an over-cooled system engaging in wasting cooling energy. The model can be integrated with scheduling, and can be used to analyze the

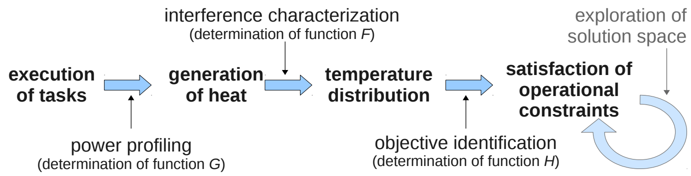

Fig. 6: Process of the unified methodology. The last step can be iterated when exploring the solution space.

change of the system performance under various scheduling policies and thermal effects. The following section proposes the unified methodology on how to quantify the AHFM, and how to integrate it with finding a schedule as per the problem in §2.2.

4. METHODOLOGY FOR PROACTIVE THERMAL-AWARE SCHEDULING IN ECDCPS

This section presents a unified methodology to numerically specify the ECDCPS in the model above, which is divided in three steps:

(1) Determine the task-to-power function G.
(2) Characterize the interference effect function F.
(3) Either
  (a) Verify that a schedule satisfies the constraints, or
  (b) Identify an optimal schedule that satisfies the constraints by:
    i. Formalizing the objective function H.
    ii. Exploring the schedule space to find an optimal schedule.

The overall process of the proposed unified methodology is depicted in Figure 6.

4.1 Characterization the power function G

Function G can be derived by experimental profiling. The profiling process can be either a measurement and validation process of an existing Cyber-Physical System, or an analytical estimation using previously published information. For instance, in the example biosensor application (Section 5.1), the power consumption profile of implanted sensors is based on available data from published literature. The power consumption profiles of our data center application are obtained through extensive measurements of data center servers.

4.2 Characterization of the interference effect function F

Generally, there are two ways to characterize F: i) by use of analytical models from the literature, or ii) by profiling of the interference matrix $\boldsymbol{D}$.

4.2.1 *Use of analytical models.* In many cases, it is possible to use analytical models to specify F. For example, radio propagation, heat convection and conduction models can be used to specify the propagation of energy, and then convert the propagation to a condition variable. This option is used in the biosensor network example of §5.1.

4.2.2 *Determination of the interference matrix* $\boldsymbol{D}$. In determining the interference coefficients, the **first** step is to identify the points of interest, i.e. the victors and victims. For example, if all the nodes of the ECDCPS, then there are n^2 coefficients. The **second** step is to profile the spatio-temporal behavior of interference, i.e. how heat originating from one point spreads throughout the space and time. This can be done in three ways:

—*Experimentally*, (if the heat sources can be controlled) by performing a set of profiling experiments. In each experiment, all heat sources but one are kept constant. The selected heat source variates in output while condition variable measurements are taken at the points of interest. Experiments are repeated for each heat source.

—*Observationally*, (if the heat sources cannot be controlled) by observational measurements. In each observation, each reading is considered an instance of the same model.

—*Numerically (by simulation)*, if the system (e.g. a prototype) is not readily available. This option emulates the mechanics of the above two ways. The advantage of this option is that there is no need for an actual system, at the cost of being forced to use computationally intensive methods.

Once the cycle of experiments, observations or simulations is done, the data will be fitted to the AHFM with regression methods. After this one-step process, the AHFM can be used to explore all scheduling scenaria.

4.3 Verifying or optimizing

4.3.1 *Verifying the safety.* With the knowledge of the G and F functions, one can verify whether a schedule C satisfies the constraints by examining whether all the elements of the vector $F(G(C))$ fall within the polytope Π of the constraints.

4.3.2 *Optimizing the performance.* The functions G and F can be used to find a schedule that optimizes (or near-optimizes) the performance while not exceeding the environmental condition variable constraints. This is done in two steps.

—*Specifying the objective function H:* Once the G and F functions are specified, then objective function H can be specified.

—*Putting it all together and finding an optimal solution:* With all functions G, F and H specified, the problem can now be expressed in Pinedo notation (Eq.1), which translates in the following optimization formulation:

$$\begin{aligned} \textbf{given}\ & \text{a set of tasks } C, \\ \textbf{minimize}\ & H(\mathcal{S}(C)) \\ \textbf{such that}\ & F(\mathcal{S}_G, \{\text{loc}_i\}, x, y, z) \in \Pi. \end{aligned} \tag{2}$$

There are various numerical or other algorithmic methods for solving this formulation, including quadratic programming, branch-and-bound, or a genetic algorithm instance, depending on the exact formulation of the problem.

4.4 Complexity of the methodology

The time length of the profiling of function G depends on the product of how many different types of nodes there are and how many different types of tasks there are, i.e. in the worst case it is $|\text{CPS}| \times |C|$, where each profiling takes as long as the duration of task T_j on node$_i$. However, the process can be speeded up if reasonable extrapolations are performed.

Determination of function F is arguably the slowest part, which may take up to several weeks to complete, depending on the method used (analytical, or AHFM fitting). Specifically for AHFM, the methodology requires at least |Victors| sets of experiments or simulations. Once the data is collected, however, applying multiple linear regression on them to yield the interference matrix $\boldsymbol{D}$ is a rather fast process (nominally, in the order of minutes for MATLAB on a contemporary desktop system).

Determination of function H is probably the fastest step, as it can be easily expressed once the above functions are specified. Once the problem is formalized and numerically specified, numeric methods may take from a few seconds up to several hours to complete. Note that finding an optimal schedule is NP-hard in general.

From the above discussion, the methodology is rather tedious to perform, and it is not recommended when there is only one instance of scheduling to be solved. The methodology is highly recommended when the systems are rather simple and there is a vast number of problem instances to be solved; in that case, the functions G, F, and H can be reused and only numeric methods need to be re-applied.

4.4.1 *Loss of accuracy and introduction of errors.* There are two discretizations introduced in this section. The fist one is the discretization of the control volume by selecting the *points of interest*. This is not a discretization in the rigid sense of discretizing the underlying model and domain. In fact, the underlying heat propagation model remains as a continuous-space model; this step merely selects a certain number of points against which to compute the value of the heat propagated. For this reason, this discretization does not affect the accuracy of the model, which is exact for the points of interest. However, if the propagation model is *approximated*, then there will be loss of accuracy due to the approximation, which is orthogonal to the selection of the points of interest.

The second discretization is the formulation of the interference effect function F. This discretization is based on the assumption that the interference effect at a point of interest is a linear, weighted sum of the heat produced by the ECDCPS nodes and by other non-ECDCPS environmental entities. This assumption in turn is based on the law of energy conservation. The error introduced in this step is that of a linear approximation. This error can be recorded in the characterization of the function F (Step 2 of the unified methodology) and taken into consideration when evaluating the studied system.

Additionally, errors may be introduced in the profiling of the function G. As values of G are used as arguments to F, these errors are scaled according to the accuracy of F. In conclusion, errors in the AHFM are introduced because of representation approximations, and not because of sampling or discretization.

4.4.2 *Efficient approaches.* One of the greatest threats to the scalability of the methodology is the discretization of the domain, which may i) lead to state space explosion, and ii) render numerical methods unsuitable.

In order to tackle the state space explosion, a layered "divide-and-conquer" approach can be used, where a coarse model is used at the top layer, and increasingly finer models are used at the lower layers. One approach to tackle the incompatibility of numerical methods and discrete domain, is to explore solutions in the continuous space, and then discretizing to a nearby feasible solution.

The above approach does not address the problem of getting trapped at local minima. In such case, randomized numerical approaches should be used.

5. EXAMPLE APPLICATIONS OF THE UNIFIED CYBER-PHYSICAL METHODOLOGY

This section presents the application of the unified cyber-physical methodology on two CPS examples: An implanted biosensor network, and an HPC data center. Although the discussion and formulations in these two examples are based on published results from previous research work [Tang et al. 2005], their organization and presentation under the prism of the unified methodology is new.

5.1 Example A: thermal-aware communication scheduling of biosensor networks

This subsection revisits the research work of [Tang et al. 2005] under the prism of the unified cyber-physical methodology. The outcome of that work was a new technique for rotating the cluster leadership to minimize the heating effects on human tissues. To achieve energy consumption balance and maximize network life time, each sensor takes the leadership in turn to communicate with the base station. Different leadership sequences lead to different temperature rises. For example, a 4-node cluster could have two different leadership sequences, such as $(3, 2, 1, 4)$ or $(4, 3, 1, 2)$, and the two sequences can result in different temperature distributions due to the different leadership history.

Importance of the problem. Networking multiple biosensors into an application-specific medical treatment, such as health monitoring, diagnostics, and prosthesis, is a promising approach, which will require research with a different perspective to resolve novel and challenging problems [Schwiebert et al. 2001]. One of the problems is potential thermal hazard resulting from power consumption of implanted sensors. This is an ECDCPS application problem; there is a thermal interaction between the implanted distributed sensors and the tissue environment.

5.1.1 *Application domain and problem description.* A *smart biosensor node* contains one or more sensors, a processor, and a transceiver. A small group of biosensor nodes form a cluster. Although there can be several clusters in the network, we assume that all nodes in the control volume are members of a single cluster; also, the base station is assumed to know the location of the nodes. This is a reasonable assumption: surgically implanted sensors are are monitored after implantation for precise location.

In a cluster-based protocol, the cluster leader collects information from all the non-leader nodes in the cluster and transmits it to the base station. Therefore, the cluster leader consumes energy faster than the non-leader nodes. The *working period* of each node is the period in which the node is the acting leader. The primary reason for rotating the cluster leader is the additional heating of the tissue surrounding the leader because of the power dissipation of the sensor and the tissue's exposure to RF waves. Rotation also provides the previous leader with a cool-down period.

The following subsection describes the adaptation of the unified methodology to the specifics of this problem. The thermal effect of this application can be modeled with AHFM because: i) the sensors are working collaboratively in a distributed manner, ii) the power consumption and the resulting thermal interference is observable and predictable.

5.1.2 *Applying the unified cyber-physical methodology.* The thermal-aware scheduling problem in this biosensor applications is ***how to schedule a leader rotation sequence to reach a minimum temperature increase and to minimize the heat effects on the tissue, given a control volume, the known locations of implanted sensors, and the related properties***.

The section below summarizes our research results according to the steps proposed in unified methodology in Section 4.

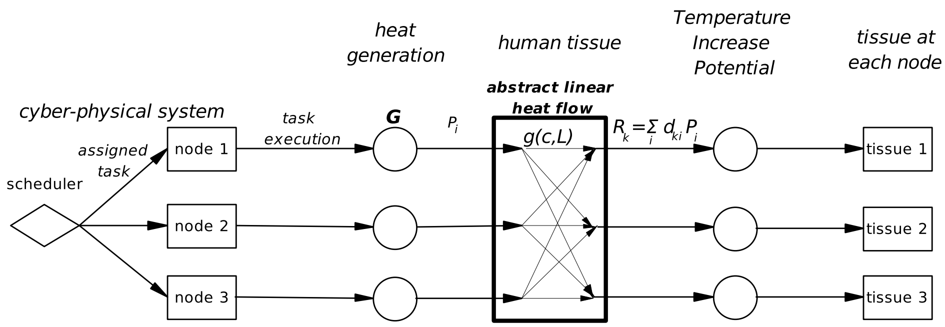

Fig. 7: Abstract heat flow model of a biosensor network. Due to homeostasis, the generated heat by the tissue is removed by blood and radiant cooling. The biosensor nodes are the only ones that produce extra heat.

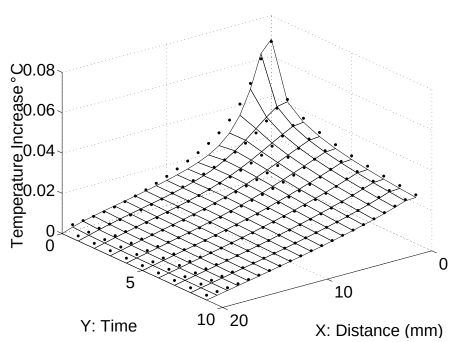

Fig. 8: Time-Space function. X-axis: distance from the heating source (leader node). Y-axis: time elapsed since the previous leader tenure (Source: [Tang et al. 2005]).

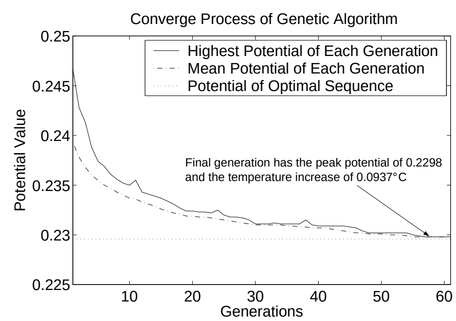

Fig. 9: Results of the genetic algorithm for each generation. Solid line is the performance of GA, dash line is the optimal performance. After 60 generations, GA performance is very close to the optimal. (Source: [Tang et al. 2005])

Defining the power consumption G of a task execution. In this application, all nodes are assumed to be identical. We use a simple power model defined as follows:

$$G(\text{Leader}) = 1\,\text{W}, \quad G(\text{unassigned}) = 0\,\text{W}.$$

The exact values in watts are not of significance in this formulation, as the most efficient schedule would be the same for nearly any reasonable pair of values of G.

Defining the interference effect function F. The temperature increase of any point close to the leader depends on both the Euclidean distance δ from the leader node, and the time τ elapsed since the leader's previous leadership tenure. We denote this relationship as the time-space function $f(\tau, \delta)$, whose numeric instances can be obtained using FDTD, as shown with the mesh lines in Fig. 8 (f itself has no analytical expression). Fitting a rational function $g(\tau, \delta)$ on $f(\tau, \delta)$ yields:

$$g(\tau, \delta) = \frac{1}{14.8827 + 8.8633\delta^{0.5} + 3.1134\tau^2 - 1.5933\tau^{2.5} + 0.2471\tau^3}. \tag{3}$$

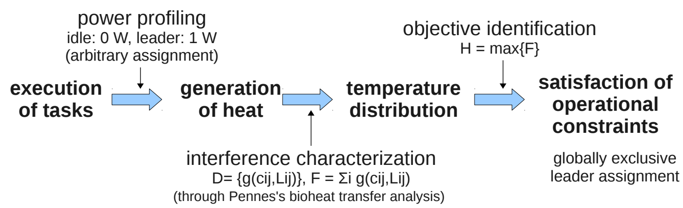

Fig. 10: Application of the methodology to the biosensor application example.

Each pair of sensor nodes, e.g., node i and node j, has different positions inside the rotation sequence, referred to as *rotation distance* c_{ij}, which is the time elapsed between the leadership tenures of node i and node j. Each pair of nodes, e.g., node i and node j, has a fixed Euclidean distance between them, denoted as *location distance* L_{ij}. Then $g(c_{ij}, L_{ij})$ give the temperature increase if they were the only two nodes in the network. We thus define the distribution matrix $\boldsymbol{D}$ as:

$$\boldsymbol{D} \equiv \{g(c_{ij}, L_{ij})\} \approx \{f(c_{ij}, L_{ij})\}. \tag{4}$$

The total interference on one node is the sum of the individual interferences from all other heating sources, so F at a specific node can be defined as:

$$F(G, j) \equiv \sum_{i=1}^{N} d_{ij} \approx \sum_{i=1}^{N} g(c_{ij}, L_{ij}). \tag{5}$$

Formalizing the objective function H. The objective is to find a leader rotation schedule that minimizes the maximum value of temperature rise $F(G, j)$:

$$H(S) \equiv \max_j \left\{F(G, j)\right\} = \max_j \left\{\sum_{i=1}^{N} d_{ij}\right\}.$$

Putting it all together and finding a solution. In order to ensure that the leadership rotation is fair, the input task set is of the same cardinality as the CPS node set. Also, as the nodes are of equal capabilities (this is a homogeneous environment), each leadership task has the same length. The problem is to find an assignment sequence of leadership that minimizes the maximum temperature rise. Let index_i be the node's turn in the leadership rotation, and the distance c_{ij} be defined as $c_{ij} = (\text{index}_i - \text{index}_j + N \mod N)$; then the optimal leader rotation scheduling problem can be formalized as:

$$P_N \;\Big|\; \text{globally exclusive} \;\Big|\; \max_j \left\{\sum_i d_{ij} \equiv \sum_i g(c_{ij}, L_{ij})\right\}, \tag{6}$$

where *globally exclusive* denotes that only one task may run at any time throughout the entire CPS.

Applying the Genetic Algorithm (GA) optimization approach to find a near-optimal rotation sequence yields efficiency results as shown in Fig. 9, combining TIP with GA significantly shortens the time for finding an near-optimal solution that has a performance very close to the optimal sequence.

5.1.3 *Research results.* Prior results from this work have been published in [Tang et al. 2005]. The contributions of that research are:

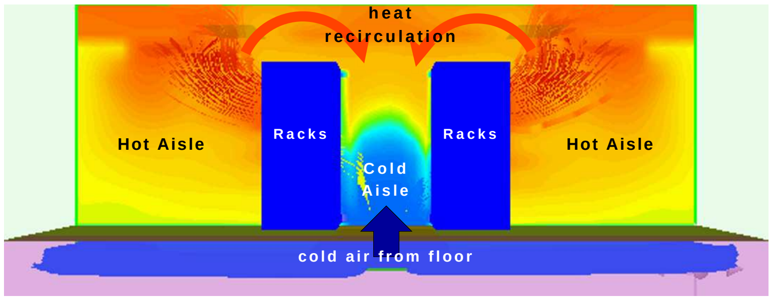

Fig. 11: Demonstration of heat recirculation: heated air in the hot aisle loops around the equipment to enter the air inlets. (Source: [Mukherjee et al. 2009])

—The modeling of thermal effects of implanted biosensors. By using a radiation model based on RF powering and the properties of the communication antenna, as well as a power dissipation model of the sensor circuitry, one can calculate the specific absorption rate (SAR) and the temperature increase in homogeneous tissue.

—The identification of factors that minimize thermal effects of a cluster-based implanted sensor network. We show the necessity of leadership rotation, and the importance of considering the leadership history and the sensor locations to lower temperature increase and to prevent potential thermal hazards to the tissue.

—The development of a Temperature Increase Potential (TIP) based Genetic Algorithm (GA) for fast computation of a minimal temperature increase rotation sequence.

5.2 Example B: task placement in data centers

During the past few years, with the prevailing usage of data centers for data processing, data storage, and communications networking, the heat dissipation density of data centers increases exponentially. Improperly designed or operated data centers may either suffer from overheated servers and potential system failures, or from over-cooled systems and paying extra utilities costs. The goal of this work is to find task placements (i.e., server assignments) so as to lower energy costs, reduce system failure rates, and consequently, optimize computing resources and minimize business expenditures.

Results of our research on thermal management of data centers have been published in [Tang et al. 2007; Tang et al. 2008; Tang et al. 2006]. The proactive scheduling of jobs can increase the energy efficiency of data centers beyond traditional energy-saving techniques. This section presents the work through the perspective of the unified methodology.

5.2.1 *Application and problem description.* A typical data center is laid out in a hot-aisle/cold-aisle arrangement, with the racks installed on a the raised floor which features perforated tiles. The air conditioners, normally referred to as *computer room air conditioner* (CRAC) or *heating ventilation air conditioner* (HVAC), deliver cold air under the elevated floor. This is referred to as *cool air*. The cool air enters the racks from their front side, picks up heat while flowing through these racks, and exits from the rear of the racks. The heated exit air forms hot aisles behind the racks, and is extracted back to the air conditioner intakes, which, in most cases, are positioned above the hot aisles. Each rack consists of several chassis, and each chassis accommodates several computational devices (servers

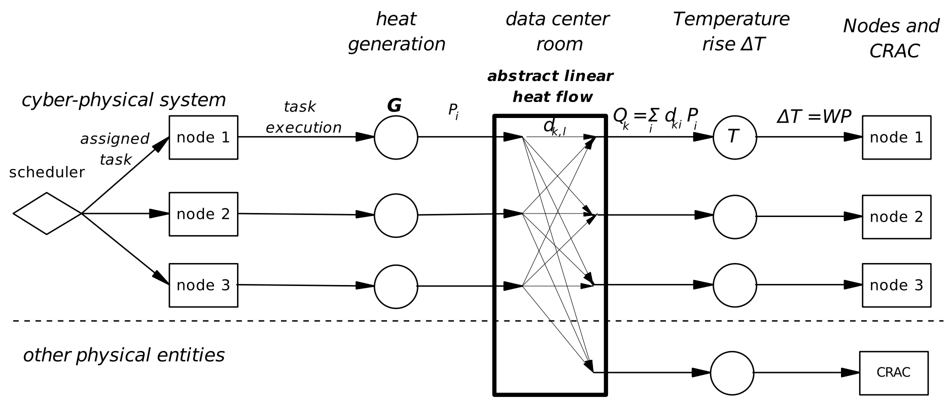

Fig. 12: Abstract heat flow model of a data center. The computing nodes are the only heat sources in the room, whereas the CRAC is the only heat sink.

or networking equipment).

The *recirculation* of hot air, i.e. *thermal interference*, from the air outlets of the IT equipment back into their air inlets (see Fig. 11) increases the inlet temperatures and can cause the appearance of hot spots [Bash and Forman 2007; Sharma et al. 2002]. Heat recirculation forces data center operators to operate their CRACs to supply cold air at a *much* lower temperature than the redline (although in the ideal case of no recirculation it could be equal to the redline temperature). Lowering CRAC's output temperature forces it to operate at a worse *coefficient of performance*, i.e. the ratio of the removed heat over the energy required to do so, which considerably increases the cooling cost.

The projected problem is ***how to assign (i.e., place) an incoming HPC job to the data center servers so that the requirement for cool air supply is minimized, thus allowing for the supply of warmer and cheaper cool air by the CRACs***. The following section describes the adaptation of the unified methodology to the specifics of the data center application.

The impact of thermal interference concerns only the nodes of the ECDCPS, the data center servers in this case, and not other entities, as opposed to the biosensor application, where the thermal interference has an impact to the tissue, i.e. the hosting environment.

5.2.2 *Applying the unified methodology.* The data center consists of n nodes (enclosures) with q servers (i.e., blade servers) each. There are m tasks submitted to the system, with Req(j) servers for each task j.

Profiling the function G. If a node i expends a_i power when idle, and each blade server expends b_{ij} power when running a specific task j, the power consumption of a chassis with q blades running task j is modeled as:

$$p_i = qb_{ij} + a_i,$$

where b_{ij} is the liner coefficient of server i's power when running task T_j. This is a *linear* power model with respect to CPU utilization. The profiling results in the ASU Fulton HPC data center [Mukherjee et al. 2007] suggest a linear trend in the *system-wide power consumption* with respect to CPU utilization for the systems measured at the data center. This power consumption includes the power of other components including memory and I/O. The research suggests that variation in other component utilization, such as disk I/O rate or memory I/O rate, in these systems does not significantly vary the power consumption.

Since a node has multiple blade servers, each server may be assigned to a separate task. Let $\boldsymbol{C}$ be the *task allocation matrix*[3], where each element c_{ij} denotes that c_{ij} servers are allocated to task T_j on node i. Then, the power consumption of node i is:

$$G_i() \equiv p_i = \sum_j c_{ij} b_{ij} + a_i \quad \Rightarrow \quad \boldsymbol{p} = \boldsymbol{C} \odot \boldsymbol{B} + \boldsymbol{a}, \tag{7}$$

where $\odot$ stands for the row-for-row product of two matrices, yielding a vector. With a very small error, all the electric power is assumed to convert into heat, i.e. $Q_i \equiv p_i$.

Defining the thermal interference function F. In this application example, the heat interference is the recirculation of heat. Previous work has allowed to yield a matrix $\boldsymbol{A} = \{\alpha_{ij}\}$ of coefficients denoting that α_{ij} heat from node i goes to node j. This means that the incoming heat to the servers, $\boldsymbol{Q}_{\text{in}}$ is the redistribution of the output heat $\boldsymbol{Q}_{\text{in}}$ (assuming that the cooling system matches the heat produced):

$$\boldsymbol{Q}_{\text{in}} = \boldsymbol{A}\boldsymbol{Q}_{\text{out}}.$$

However, the output heat equals, the input heat plus the electric power converted into heat ($Q_i = p_i$ from above).

$$\boldsymbol{Q}_{\text{out}} = \boldsymbol{p} + \boldsymbol{Q}_{\text{in}}.$$

These two equations describe the heat feedback (recirculation) process. Substituting the second into the first, one gets:

$$\begin{aligned} &\boldsymbol{Q}_{\text{in}} = \boldsymbol{A}(\boldsymbol{p} + \boldsymbol{Q}_{\text{in}}) \;\Rightarrow\; \boldsymbol{A}^{-1}\boldsymbol{Q}_{\text{in}} = \boldsymbol{p} + \boldsymbol{Q}_{\text{in}} \;\Rightarrow\; \boldsymbol{A}^{-1}\boldsymbol{Q}_{\text{in}} - \boldsymbol{Q}_{\text{in}} = \boldsymbol{p} \Rightarrow \\ &\boldsymbol{Q}_{\text{in}} = [\boldsymbol{A}^{-1} - \boldsymbol{I}]^{-1}\boldsymbol{p} = \boldsymbol{D}\boldsymbol{p}, \quad \text{with } \boldsymbol{D} \equiv [\boldsymbol{A}^{-1} - \boldsymbol{I}]^{-1}. \end{aligned} \tag{8}$$

While $\boldsymbol{D}\boldsymbol{p}$ gives the heat rate arriving at a node i, the chosen condition variable is the temperature rise ΔT_{in} at the air inlet of each node, caused by this heat arrival rate. To compute ΔT_{in}, the heat rate vector $\boldsymbol{D}\boldsymbol{p}$ has to be converted into temperature. The thermodynamic parameters that convert heat into temperature are organized into a vector $\boldsymbol{\varphi} = \langle \varphi_i \rangle$. Then, $F(G, k)$ can be defined as:

$$F(G) \equiv \Delta \boldsymbol{T}_{\text{in}} = \boldsymbol{\varphi}\boldsymbol{Q}_{\text{in}} \overset{(8)}{=} \boldsymbol{\varphi}\boldsymbol{D}\boldsymbol{p} \overset{(7)}{=} \boldsymbol{\varphi}\boldsymbol{D}\,[\boldsymbol{C}\odot\boldsymbol{B} + \boldsymbol{a}]\,. \tag{9}$$

Formalizing the objective function H. Since the goal is to reduce the maximum temperature rise, the objective function is defined as the maximum of the vector $\Delta \boldsymbol{T}_{\text{in}}$.

$$H \equiv \max \Delta \boldsymbol{T}_{\text{in}}. \tag{10}$$

Putting it all together and finding a solution. Recall from above that the task allocation matrix $\boldsymbol{C} = \{c_{ij}\}_{n\times m}$ denotes how many servers from each node i are allocated to task j. The sum of server assignments to tasks per node *may not* exceed the available processors on that node, i.e.,

$$\sum\nolimits_{j=1}^{m} c_{ij} \leqslant q, \forall i = 1 \ldots n,$$

[3]This is unrelated to the c_{im} in Section 5.1.

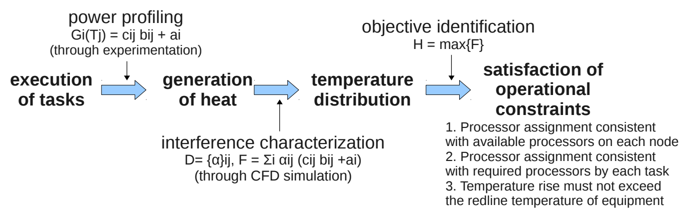

Fig. 13: Application of the methodology to the data center application example.

and the sum of processor assignments per task *must* be equal to the number of processors it requires, i.e.,

$$\sum\nolimits_{i=1}^{n} c_{ij} = \mathrm{Req}(j), \forall j = 1 \ldots m.$$

The safety constraint associated with this ECDCPS is the *redline* temperature, T_{red} of the computing equipment. This is an upper limit value provided by the equipment manufacturer and it represents the highest operating temperature recommended by the manufacturer. The redline temperatures are organized into a vector:

$$\boldsymbol{T}_{\mathrm{red}} = \langle T_{\mathrm{red}}, T_{\mathrm{red}}, \ldots, T_{\mathrm{red}} \rangle.$$

The problem of minimizing the temperature rise can be now stated as:

$$\text{Find a task placement matrix } \{c_{ij}\}_{n\times m} \text{ in:}$$

$$P_n \left| \begin{array}{c} T_{\mathrm{supply}} + \Delta \boldsymbol{T} < \boldsymbol{T}_{\mathrm{red}}; \\ \sum_{j=1}^{m} c_{ij} \leqslant m, \forall i = 1 \ldots n; \\ \sum_{i=1}^{n} c_{ij} = \mathrm{Req}(j), \forall j = 1 \ldots m \end{array} \right| \max\left\{ \boldsymbol{\varphi D} \left(\boldsymbol{C} \odot \boldsymbol{B} + \boldsymbol{a} \right) \right\}. \tag{11}$$

5.3 Similarities and dissimilarities of the example applications

In the application of thermal-aware sensor networks, radiation of wireless communication and the power dissipation of implanted sensors will heat the surrounding tissues; therefore the optimization goal is to reduce the temperature increase and lower the risk of thermal hazard. The formalized problem is how to find the leadership sequence that minimizes the peak temperature. In other words, finding the best sequence is a *task placement in a temporal manner*: which sensor runs the leadership task first, which one goes second, etc.

In the case of thermal management of data centers, the non-optimized heat dissipation of servers will increase the cooling energy costs and reduce the servers' reliability. Tasks can be scheduled to achieve better temperature distributions and lower the energy costs. Therefore, our optimization goal is to improve the thermal distribution (e.g., to lower the inlet temperature of servers), minimize the energy costs and maximize the computing capability. This is more like *task placement in a spatial manner*: which part of data center dissipates heat at a higher rate, which part of data center dissipates heat at a lower rate, etc.

In both applications, the DCPSs present thermal effects to the surrounding environment and the systems themselves. In addition, both of them require certain heat models to evaluate the correlations between the temperature change and system performance.

The similarities between the two applications, such as thermal interference to the environment and using scheduling to manipulate the change of temperatures, motivated us to

Table II: Comparisons of Thermal-aware Applications: Biosensor Networks and Data Centers

	Thermal problem for biomedical application	**Thermal Management of Data center**
Application scenario	The wireless radiation and the power consumption of sensor operations increase the temperature of surrounding tissues.	The heat generated by high density servers increases energy cost of data center operation.
Heat transfer mechanism	Convection, conduction and radiation.	Convection
Original numerical simulation	Finite Difference Time Domain (FDTD)	Computational Fluid Dynamics (CFD)
Power consumption profile $G_i()$	Obtained from literature and estimation	Obtained through measurement and profiling
Abstract heat model to characterize interference F	Time-space function	Abstract model of cross interference
Objective function H	To minimize the temperature rise of implanted sensors by finding a leadership rotation sequence which minimizes the temperature rise.	To minimize the energy cost through dynamically assigning task among server nodes.
Task scheduling	Spatio-temporal (time and node assignment)	Spatial-only (node assignment)

explore a unified methodology of thermal aware scheduling for ECDCPS. We have established a generalized approach to model and reduce the thermal interference in ECDCPS.

The most obvious difference between the two applications is the physical granularity: tiny sensor networks versus large server farms. Furthermore, finding the best sequence is a temporal domain task scheduling, whereas assigning tasks among server nodes is spatial domain task scheduling[4]. The dissimilarities of the two applications provided us the opportunity to validate the applicability of our approach. In this paper, it is shown that our approach can be successfully applied on thermal management of different distributed systems regardless of the physical sizes of the systems, or the manner of assigning tasks.

Table II compares similarities between the two applications. It is also shown how to apply the unified methodology to two applications.

As one can see, both of the problems present thermal effects to the surrounding environment or the system itself. The temperature change of the environment would change the normal operation and the performance of the distributed system. In both applications, it is required to conduct extensive, time-consuming numerical simulation to evaluate thermal performance. Similarly, the thermal evaluation process can be accelerated by adopting proper abstract heat model and find the solution with heuristic algorithms. The approach and methodology presented in the aforementioned two problems could be applied to the other thermal-related distributed system applications.

[4]A recent work combines spatial and temporal scheduling in data centers [Mukherjee et al. 2009].

6. CONCLUSIONS

This work addressed a special type of Cyber-Physical Systems referred to as Environmentally Coupled Cyber-Physical Distributed Systems. The focus is on thermal interference problems because heat dissipation is the a critical issue for electronic devices, and it has immediate influence on the environment by an ECDCPS.

We presented an abstract heat flow model and a unified methodology of thermal-aware scheduling for ECDCPS. We applied this methodology on two distinct application domains and showed how this methodology helps in identifying the thermal interference and its effects and how to relate them to functional and operational requirements. Our achievements in two vastly different applications confirm that the proposed approach has wider applicability to minimize the interference of environmentally coupled distributed systems.

7. FUTURE WORK

Future work is divided into two axes: formalization and usability. Future work in formalization involves improving the methodology's abstractions and identifying a language-based formal specification scheme. Future work in usability involves extending the methodology to other energy domains, automating the process of the methodology, and enhancing the methodology to devise online algorithms.

7.1 Formalization

Formalization efforts pertain on developing a DCPS theory, i.e. the theoretical background necessary to design, model, specify, verify and validate DCPS systems. This includes: i) proper concepts and abstractions, ii) an empirical or experimental modeling methodology, iii) a formal specification scheme toward verifying and validating the DCPS systems.

7.1.1 *Language-based formal model specification of DCPS.* Although the unified methodology provides the basis to verify DCPS systems, the verification cannot be done rigorously before a formal model specification scheme is available. Ongoing work is looking into using the Abstract Architecture Description Language (AADL) from the Software Engineering Institute (SEI) for this purpose. It is extensible since new language constructs can be incorporated as an *annex* to the language. The challenge of this research work is to identify the proper AADL constructs to specify the interferences and the criticality management. An research effort of using AADL to develop a design tool for body-area networks and devices is given in [Banerjee et al. 2010].

Figure 14 shows a sample envisioned AADL model for Data Centers. As shown in the figure, there are two primary extensions required: i) the capability to describe the physical components; and ii) the capability to specify the physical characteristics of the computing component. The physical component description is intended to specify the energy interference type (*heat* in case of data centers) and the threshold energy levels for the physical environment. For example, the redline temperature of the servers can act as a threshold temperature for the data centers. The physical characteristics of the computing entities are intended to describe the interference rules. The interference from the servers to environment depends on their resource usage.

7.2 Improving the usability of the methodology

7.2.1 *Studying the unified methodology in other energy domains.* In this paper introduced and used an abstract heat flow model to characterize the thermal interference in

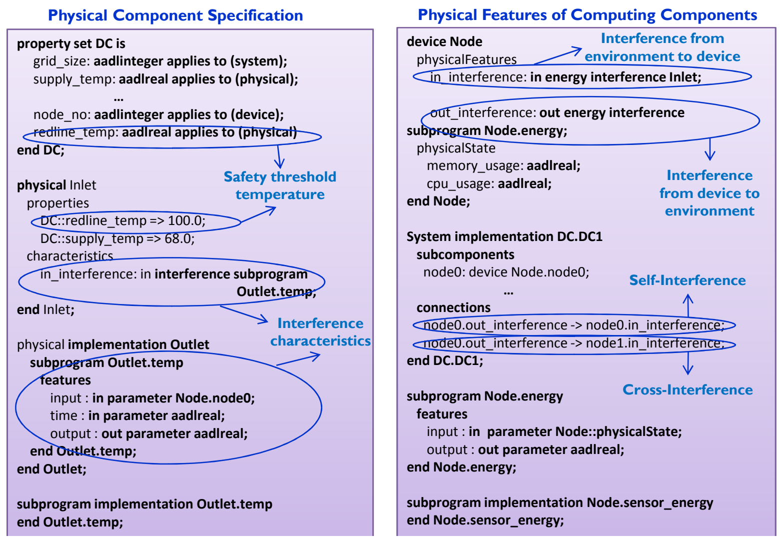

Fig. 14: Sample *envisioned* AADL Specification of Data Centers. AADL needs to be augmented with the constructs to specify: i) the physical components; and ii) the physical characteristics and interferences of the computing entities.

distributed systems. In general, it should be easy to characterize other energy-based (e.g. sonic or electromagnetic) interference problems. It may be however more difficult to use the model on other types of interference, e.g. biological interference in systems of organism or biomass exchange, or "social interference" involved with human activities.

7.2.2 *Going from offline to online scheduling.* The solutions presented in the examples in this paper consider the search of a schedule in an *offline* manner. A more pragmatic solution scheme should allow for online decision making, especially one that considers a task arrival process and changes in the node availability.

7.2.3 *Automating the process of the methodology.* A sustainable CPS is a CPS that is also self-managing and self-configuring. CPSs have to deal with the ever-changing nature of their hosting environment, as well with the uncertainty in node availability. For example, a data center may experience some layout or infrastructure change frequently, being equipped with new devices, and frequent profiling process would bring severe interruption of data center operation. As it is impossible to pre-configure the CPS for all possible scenarios, a CPS should be able to dynamically adapt its operation to the conditions. For that matter, it should have the ability to automatically apply the unified methodology in order to be able to characterize the interference and its effects. Automating the unified methodology involves automating all steps.

Automated, non-interruptive profiling. In the unified methodology, the process of characterizing interference is based on offline profiling to characterize interference among distributed nodes. In practice, a profiling process for a large data center may take several days. A special profiling schedule may not be able to be inserted into the busy operation schedule of a data center and cannot be risked on the patient's tissue. Therefore, future work is needed to study the online profiling without interrupting the normal operation of the distributed systems. An idea is to obtain a power and recirculation model from built-in sensor and from operating system statistics logs, with the objective to waive the requirement for extensive simulation and manual effort in obtaining the power model.

Automated, non-intrusive interference characterization. This is arguably the most difficult step to automate. Typically, characterizing the interference requires testing for all possible operation states. In DCPS, there is a state space explosion because of all the possible combinations of operation states in the node set. Therefore, efficient techniques must be employed that can build the model without needing to test all possible states, e.g., taking advantage of the linearity of the interference or of possible symmetries.

Acknowledgments

The authors thank Naveen Tummala for his contribution to biosensor thermal scheduling, Tridib Mukherjee for his contribution to and data center equipment profiling, and the anonymous reviewers along with the special issue guest editors, Georgios E. Fainekos, Eric Goubault, Franjo Ivancic and Sriram Sankaranarayanan, for their helpful suggestions to improve the quality of this paper.

REFERENCES

Adelstein, F., Gupta, S. K. S., Richard, III, G. G., and Schwiebert, L. 2005. *Fundamentals of Mobile and Pervasive Computing*. McGraw-Hill.

Arora, A., Dutta, P., Bapat, S., Kulathumani, V., Zhang, H., Naik, V., Mittal, V., Cao, H., Demirbas, M., Gouda, M., Choi, Y., Herman, T., Kulkarni, S., Arumugam, U., Nesterenko, M., Vora, A., and Miyashita, M. 2004. A line in the sand: A wireless sensor network for target detection, classification, and tracking. *Computer Networks 46,* 5 (Dec), 605–634. Military Communications Systems and Technologies.

Banerjee, A., Kandula, S., Mukherjee, T., and Gupta, S. K. S. 2010. BAND-AiDe: A tool for cyber-physical oriented analysis and design of body area networks and devices. *ACM Transactions on Embedded Computing Systems (TECS), Special issue on Wireless Health Systems, (minor revision submitted for review)*.

Bash, C. and Forman, G. 2007. HPL-2007-62 cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. Tech. Rep. HPL-2007-62, HP Laboratories Palo Alto. Aug.

Bujorianu, M. C., Bujorianu, M. L., and Barringer, H. 2009. A unifying specification logic for cyber-physical systems. In *MED '09: Proceedings of the 2009 17th Mediterranean Conference on Control and Automation*. IEEE Computer Society, Washington, DC, USA, 1166–1171.

Donald, J. and Martonosi, M. 2006. Techniques for multicore thermal management: Classification and new exploration. *SIGARCH Comput. Archit. News 34,* 2, 78–88.

Dutta, P., Grimmer, M., Arora, A., Bibyk, S., and Culler, D. 2005. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *The Fourth International Conference on Information Processing in Sensor Networks (IPSN'05)*.

Fulford-Jones, T. R. F., Wei, G.-Y., and Welsh, M. 2004. A portable, low-power, wireless two-lead ekg system. In *In Proceedings of the 26th IEEE EMBS Annual International Conference*. San Francisco.

Gill, C. D. and Niehaus, D. 2006. Towards system software platforms for cyber-physical systems, Position Paper. In *NSF Cyber-Physical Systems Workshop*.

Ilic, M., Xie, L., Khan, U., and Moura, J. 2008. Modeling future cyber-physical energy systems. In *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, 2008 IEEE*. 1–9.

Jiang, W., Xiong, G., and Ding, X. 2008. Energy-saving service scheduling for low-end cyber-physical systems. In *Young Computer Scientists, 2008. ICYCS 2008. The 9th International Conference for*. 1064–1069.

Lee, E. A. 2006. Cyber-physical systems: Are computing foundations adequate? (Position Paper). In *NSF Cyber-Physical Systems Workshop*.

Lee, E. A. 2008. Cyber physical systems: Design challenges. In *ISORC '08: Proceedings of the 2008 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing*. IEEE Computer Society, Washington, DC, USA, 363–369.

Moore, J., Chase, J., and Ranganathan, P. 2006. Weatherman: Automated, online and predictive thermal mapping and management for data centers. In *Autonomic Computing, 2006. ICAC '06. IEEE International Conference on*. 155–164.

Mukherjee, T., Banerjee, A., Varsamopoulos, G., Gupta, S. K. S., and Rungta, S. 2009. Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers. *Computer Networks 53,* 17, 2888 – 2904. Virtualized Data Centers.

Mukherjee, T. and Gupta, S. K. S. 2009. CRET: A Crisis Response Evaluation Tool to improve Crisis Preparednesss. In *Proceedings of the International Conference on Technologies for Homeland Security*. IEEE.

Mukherjee, T., Varsamopoulos, G., Gupta, S. K. S., and Rungta, S. 2007. Measurement-based power profiling of data center equipment. In *Proc. IEEE Conference on Clustered and Grid Computing (Cluster 2007), Workshop on Green Computing (GreenCom'07)*. Austin, TX.

Pinedo, M. L. 2008. *Scheduling: Theory, Algorithms, and Systems*, 3 ed. Springer, Chapter 2. Deterministic Models: Preliminaries, 13–19.

Porter, J., Karsai, G., and Sztipanovits, J. 2009. Towards a time-triggered schedule calculation tool to support model-based embedded software design. In *EMSOFT '09: Proceedings of the seventh ACM international conference on Embedded software*. ACM, New York, NY, USA, 167–176.

Schwiebert, L., Gupta, S. K. S., and Weinmann, J. 2001. Research challenges in wireless networks of biomedical sensors. In *MobiCom '01: Proceedings of the 7th annual international conference on Mobile computing and networking*. ACM, New York, NY, USA, 151–165.

Sentilles, S., Vulgarakis, A., TomášBureš, Carlson, J., and Crnković, I. 2008. A component model for control-intensive distributed embedded systems. In *CBSE '08: Proceedings of the 11th International Symposium on Component-Based Software Engineering*. Springer-Verlag, Berlin, Heidelberg, 310–317.

Sharma, R. K., Bash, C. E., and Patel, C. D. 2002. Dimensionless parameters for evaluation of thermal design and performance of large scale data centers. In *Proceedings of the American Institute of Aeronautics and Astronautics (AIAA)*. 3091.

Springer, R., Lowenthal, D. K., Rountree, B., and Freeh, V. W. 2006. Minimizing execution time in MPI programs on an energy-constrained, power-scalable cluster. In *ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming (PPoPP'06)*. 230–238.

Sun, Y., McMillin, B., Liu, X. F., and Cape, D. 2007. Verifying noninterference in a cyber-physical system the advanced electric power grid. In *QSIC '07: Proceedings of the Seventh International Conference on Quality Software*. IEEE Computer Society, Washington, DC, USA, 363–369.

Sztipanovits, J. 2007. Composition of cyber-physical systems. In *Engineering of Computer-Based Systems, 2007. ECBS '07. 14th Annual IEEE International Conference and Workshops on the*. 3 –6.

Tan, Y., Vuran, M. C., and Goddard, S. 2009. Spatio-temporal event model for cyber-physical systems. In *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*. IEEE Computer Society, Washington, DC, USA, 44–50.

Tang, Q., Gupta, S. K. S., and Varsamopoulos, G. 2007. Thermal-aware task scheduling for data centers through minimizing heat recirculation. In *IEEE Cluster*.

Tang, Q., Mukherjee, T., Gupta, S. K. S., and Cayton, P. 2006. Sensor-based fast thermal evaluation model for energy efficient high-performance datacenters. In *Int'l Conf. Intelligent Sensing & Info. Proc. (ICISIP2006)*. 203–208.

Tang, Q., Tummala, N., Gupta, S. K. S., and Schwiebert, L. 2005. Communication scheduling to minimize thermal effects of implanted biosensor networks in homogeneous tissue. *IEEE Transactions on Biomedical Engineering 52,* 7 (July), 1285–1294.

Tang, Q., Varsamopoulos, G., and Gupta, S. K. S. 2008. Thermal-aware task scheduling for data centers through minimizing peak inlet temperature. *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Power-Aware Parallel and Distributed Systems (TPDS/PAPADS) 19,* 11 (Nov.), 1458–1472.

Wolf, W. 2009. Cyber-physical systems. *Computer 42,* 3 (Mar.), 88 –89.

Xue, C., Xing, G., Yuan, Z., Shao, Z., and Sha, E. 2009. Joint sleep scheduling and mode assignment in wireless cyber-physical systems. In *Distributed Computing Systems Workshops, 2009. ICDCS Workshops '09. 29th IEEE International Conference on*. 1–6.

Zhang, F., Szwaykowska, K., Wolf, W., and Mooney, V. 2008. Task scheduling for control oriented requirements for cyber-physical systems. In *Real-Time Systems Symposium, 2008*. 47–56.