2012 Sixth IEEE/ACM International Symposium on Networks-on-Chip

An Optimal Control Approach to Power Management for Multi-Voltage and Frequency Islands Multiprocessor Platforms under Highly Variable Workloads

Paul Bogdan, Radu Marculescu
Department of ECE
Carnegie Mellon University
Pittsburgh, USA 15213
{pbogdan,radum}@ece.cmu.edu

Siddharth Jain
Department of EE
Indian Institute of Technology
Kanpur, India 208016
sidjain@iitk.ac.in

Rafael Tornero Gavila
Departament d'Informàtica
Universitat de València
46100 Burjassot, Valencia, Spain
rafael.tornero@uv.es

***Abstract*—Reducing energy consumption in multi-processor systems-on-chip (MPSoCs) where communication happens via the network-on-chip (NoC) approach calls for multiple voltage/frequency island (VFI)-based designs. In turn, such multi-VFI architectures need efficient, robust, and accurate run-time control mechanisms that can exploit the workload characteristics in order to save power. Despite being tractable, the linear control models for power management cannot capture some important workload characteristics (e.g., fractality, non-stationarity) observed in heterogeneous NoCs; if ignored, such characteristics lead to inefficient communication and resources allocation, as well as high power dissipation in MPSoCs. To mitigate such limitations, we propose a new paradigm shift from power optimization based on linear models to control approaches based on fractal-state equations. As such, our approach is the first to propose a controller for fractal workloads with precise constraints on state and control variables and specific time bounds. Our results show that significant power savings (about 70%) can be achieved at run-time while running a variety of benchmark applications.**

Keywords - Networks-on-Chip; Power Management; Fractal Workloads; Finite Horizon Optimal Control

I. INTRODUCTION

Networks-on-Chip (NoCs) have been recently proposed as a promising communication paradigm able to overcome various communication issues (e.g., increased interconnect delays, high power consumption) in highly integrated multi-core systems. In contrast to traditional bus-based designs, the NoC paradigm enables various processing elements (PEs) communicate by routing packets instead of wires [4][5][17][22]. As such, NoCs not only offer a high degree of scalability and reusability, but also represent the main driver for achieving tera-scale computing [15].

Integrating many cores operating at high frequencies in order to accommodate complex applications [29][33] leads to heterogeneous workloads, higher power consumption, and temperature fluctuations within die [6][8][19][26]. Such problems cannot be predicted or corrected at the design and manufacturing stages so, in order to sustain the increasing computational demands, it is essential to enhance the multi-core platforms with smart power management policies, which can enable per core/tile control of power consumption while satisfying various performance levels [16][19].

Towards this end, in this paper, we formulate the problem of optimal power management for multi-domain platforms where communication happens via a globally asynchronous locally synchronous (GALS) NoC architecture. The goal of our on-line control algorithm is to determine the *optimal* operating frequencies for both PEs and routers that belong to separate voltage and frequency islands (VFIs) such that the performance constraints, typically measured by queues utilization (or occupancy) values, are met despite the high variability exhibited by real computational workloads. Queues utilization is used as a performance measure because it is directly related to the waiting times of packets in the queues and so it is proportional with packet latency[1].

Generally speaking, several types of control techniques have been used in many engineering applications; they would then naturally appear as good candidates to control the MPSoC behavior as well [12][25][34][40]. For instance, the feedback-based control approaches compute a set of *control actions* meant to bring the system into a desired state with no constraints on the magnitude of the control signal. In many situations, however, such control signals can take exceedingly high values which make them (physically) unfeasible. In addition, the feedback-based control strategies have the drawback that only a limited number of design parameters can be found from the closed-loop pole locations. An alternative approach is to consider the problem of *finite horizon optimal control* with a predefined reference which finds the best sequence of control actions over a fixed time interval (horizon); this set of control actions can bring the system (characterized by differential equations) to the desired reference at the end of the control interval.

The new optimal control approach proposed in this paper is also finite horizon in nature. As opposed to existing approaches, it allows to directly optimize a certain performance metric subject to *fractal* (i.e., fractional derivatives) state equations (i.e., for queue utilization) and bounded control signals (i.e., operating frequencies). In other words, the proposed fractal controller is able to provide the optimal control signals (i.e., operating frequencies), if they exist, for a given performance level.

[1] We note that our mathematical formulation for power management can be extended for other performance metrics as well.

978-0-7695-4677-3/12 $26.00 © 2012 IEEE
DOI 10.1109/NOCS.2012.32

IEEE computer society

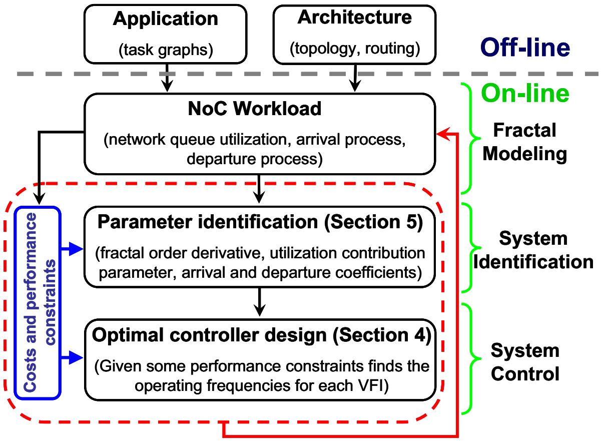
Application
(task graphs)
Architecture
(topology, routing)
Off-line
On-line
NoC Workload
(network queue utilization, arrival process, departure process)
Fractal Modeling
Parameter identification (Section 5)
(fractal order derivative, utilization contribution parameter, arrival and departure coefficients)
System Identification
Optimal controller design (Section 4)
(Given some performance constraints finds the operating frequencies for each VFI)
System Control
Costs and performance constraints

Figure 1. Overview of our methodology. We first input the information about application (e.g., task graphs), architecture (e.g., topology, routing) and the set of costs and performance constraints that have to be met by the NoC platform. Building on NoC workload measurements (i.e., queue utilization, packet arrival and departure times), we build a fractal model of the queue dynamics by estimating the fractional order of the time derivative, the utilization parameter and the arrival/departure coefficients. Next, we use this identified model to design an optimal controller which determines the VFI optimal operating frequencies such that the performance constraints are satisfied.

We note that existing control-based approaches for power management [2][12][25][34][40] rely on the assumption that real workloads can be modeled via linear state-space models without explicitly including the timing constraints in problem formulation, i.e., they work under the assumption of having an infinite time horizon to control the system. As shown later in the paper, such approaches can provide inefficient solutions that actually may consume more power in real situations.

In summary, our novel contribution to state-of-the-art power management of MPSoCs is threefold:

- First, we propose a fractal-based state description of the dynamics of queues interfacing neighboring VFIs (see the top part of Figure 1). For completeness, we also describe a strategy for estimating the parameters of the fractal model as shown in the mid-section of Figure 1.
- Second, we formulate the power management problem as a *constrained finite horizon fractional control* problem, which brings the utilization of queues in the multicore platforms as closely as possible to some predefined reference values, while minimizing the individual operating frequency of both PEs and routers (see Figure 1). Of note, the controller we synthesize *does* account for the high variability observed in computational workloads and ensures that the operating frequencies of processors and routers remain within a predefined interval.
- Third, using Lagrange optimization and calculus of variations, we derive the optimality conditions that need to be satisfied by all operating frequencies across the VFIs in order to reach the desired performance level across the entire multicore platform.
- Finally, we evaluate the practical impact of all these solutions using realistic benchmarks and applications.

The remainder of the paper is organized as follows. Section II reviews several power and thermal management techniques proposed to date for multicore platforms and highlights our novel contribution. In Section III, we discuss the main concepts specific to fractional calculus and explore how these ideas can be related to the observed characteristics of real world processes. Section IV presents the fractional optimal control problem and summarizes the optimality conditions for the run time power management algorithm we propose. Section V describes how the parameters of the proposed fractal model can be identified (at run time) from real traces and presents the experimental results we obtained while evaluating this approach. Finally, Section VI concludes the paper by summarizing our main contribution.

II. Related Work and Novel Contribution

The power and/or thermal management methodologies proposed to date for multicore systems focus on either balancing the power/thermal profile via task or thread manipulation (i.e., allocation, migration, scheduling) [10][11][13][28][37], optimizing power consumption via clock gating [20], or dynamic voltage and frequency scaling (DVFS) [3][18][24][30]. For instance, while building on a DVFS-based approach, Sharifi et al. [30] propose a joint technique for thermal and energy management which optimizes for energy efficiency. Arjomand et al. [3] propose a thermal-aware heuristic for regular 3D NoCs which, for a predefined mapped application, seeks to find the voltage and frequency of all cores such that the power consumption is reduced. Along the same lines, Coskun et al. [10] propose an autoregressive moving average (ARMA) model for predicting the core temperature and allocating the application tasks to the cores while trying to balance the thermal profile. Coskun et al. [11] also describe a thermal-aware job scheduling that reduces the frequency of hot spots and spatial thermal gradients. In contrast, a task migration strategy for thermal management is presented in [13]. Along the same lines, a dynamic thermal management for 2D MPSoCs is proposed in [24] which scales the frequency for a hot core such that its temperature is kept below a given threshold.

Several research studies propose control-theoretic approaches to optimize for power and/or thermal profiles [2][12][23][25][34][40]. For example, while trying to minimize the power consumption in multiple clock domain multicores, the pioneering work by Wu et al. [39][40] models cores as queuing systems and proposes a Proportional-Integrator-Derivative (PID) controller which scales the operating frequency (for each clock domain) such that the queue utilization is kept close to a reference value. Ogras et al. [25] propose a feedback-based control mechanism to control the speed of each VFI while describing the NoC traffic through a linear state-space equation [35]. Along the same lines, Garg et al. [12] investigate the performance of a small world distributed control approach and propose a custom feedback control strategy which seeks to minimize the implementation cost of global communication, while sacrificing some performance. Adopting a similar linear system representation, Alimonda et al. [2] propose a feedback control DVFS of data-flow

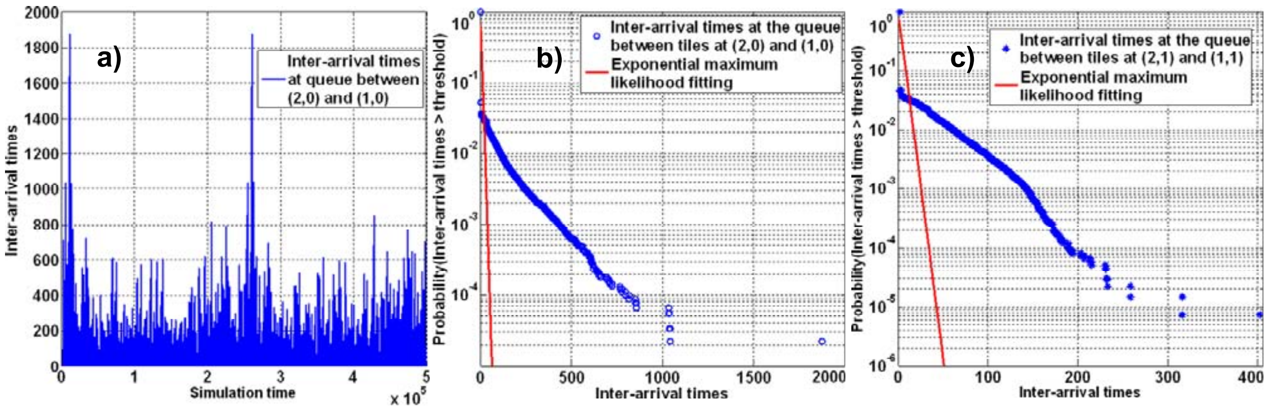

Figure 2. a) Packet inter-arrival times at queue between tiles (2,0) and (1,0) a 4×4 mesh NoC running a 16-node multithreaded online transaction processing (*oracle*) application. b) Comparison between the probability of inter-arrival times to exceed a given threshold and the maximum likelihood exponential fitting for the inter-arrival times at queue between tiles (2,0) and (1,0) obtained from a 4×4 mesh NoC running a 16-node multithreaded online transaction processing (*oracle*) application. c) Comparison between the probability of inter-arrival times to exceed a given threshold and the maximum likelihood exponential fitting for the inter-arrival times at queue between tiles (2,1) and (1,1) obtained from a 4×4 mesh NoC running a 16-node multi-threaded scientific (*ocean*) application. This significant departure from exponential assumption shows that the network control and power management cannot be done properly using the classical linear system theory. Such experimental characteristics invalidate the use of classical control theory for regulating bursty workloads, and motivate our fractional calculus approach to power management.

applications mapped on MPSoCs. Zanini et al. propose a predictive model for balancing the thermal profile of the multicore platform by controlling the operation frequencies.

Before getting into the details of our new approach, we note that in all this previous work, the authors assume that either the workloads are stationary (and therefore use techniques like task migration/remapping which are based on average values of the variables that need to be optimized), or the NoC traffic can be characterized by exponentially distributed inter-arrival times between successive packet arrivals at various queues in the network. Unfortunately, the real applications can rarely be characterized by exponential-type distributions [9]. For instance, Figure 2.a shows the time series of inter-arrival times between successive packets received at a queue located between tiles (2,0) and (1,0) in a 4×4 mesh NoC under XY wormhole routing. In this experiment, the packets consist of 15 flits, queues have 50 slots, and inter-arrival times are recorded while running a multi-threaded transaction processing application.

By looking at data in Figure 2.a, we can easily note that the inter-arrival times exhibit not only a high variation, but also a non-stationary type of behavior which cannot be properly captured by the classical linear system theory models [9]. This observation can be directly proved by checking the validity of the exponential assumption as shown in Figure 2.b and Figure 2.c. More precisely, we observe that the empirical probability distribution of inter-arrival times between successive packets does *not* follow the exponential assumption and it can be better fit by a heavy tail distribution instead.

The existence of heavy tails in the dynamics of queue utilization suggests that the average value of any variable characterizing a metric of interest in the system (e.g., queue utilization) at time t does *not* depend only on its previous value at time t-1, but instead, due to the long term memory of the traffic, it depends on the weighted sum of several values at prior moments in time t-1, t-2, etc. This is precisely the point where correlations observed in the inter-arrival times of packet communication can and should be exploited via fractal-based models to control and optimize the overall multi-core platform operation [7]. From a mathematical perspective, the fractal model is based on fractional (i.e., non-integer) derivatives which are described next.

III. Basics On Fractional Calculus

Since its inception, fractional calculus has found many applications in physics (e.g., dielectric polarization, heat transfer phenomena, etc.) and engineering (e.g., control, bio-engineering, etc.) [27]. More recently, the calculus of variations has been extended to systems characterized by fractional dynamical equations [1].

Simply speaking, the fractional (or fractal) calculus is based on techniques for differentiation and integration of arbitrary orders [21][27]. Unlike classical (i.e., integer order) calculus, fractional derivatives allow us to directly incorporate the dynamical characteristics (fractal behavior) of any target process $x(t)$ (e.g., queue utilization in a network) through a weighted sum denoting the contribution of the previous events $x(\tau)$, for $\tau \in [0,t]$:

$$\frac{d^{\alpha}x(t)}{dt^{\alpha}} = \frac{1}{\Gamma(n-\alpha)}\frac{d^{n}}{dt^{n}}\int_{0}^{t}\frac{x(\tau)}{(t-\tau)^{\alpha-n+1}}d\tau \qquad (1)$$

where α is the fractional order of the derivative and $\Gamma(n\text{-}\alpha)$ is the Gamma function [21], n is an integer and $n\text{-}1<\alpha<n$. This continuous time definition of a fractional derivative can also be written in a discretized form as follows:

$$\frac{d^{\alpha}x(t)}{dt^{\alpha}} = \lim_{h\to 0}\frac{1}{h^{\alpha}}\sum_{j=0}^{\left[\frac{t-a}{h}\right]}(-1)^{j}C_{j}^{\alpha}\,x(t-jh) \qquad (2)$$

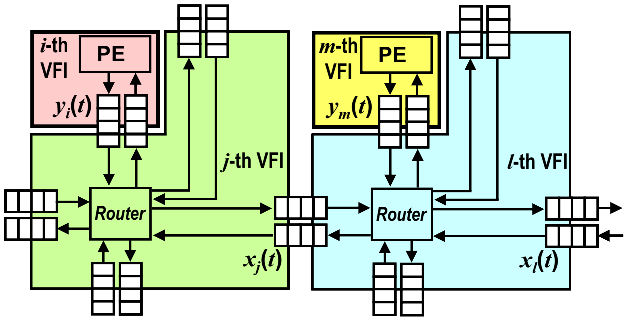

Figure 3. Representation of *j*-th and *l*-th neighboring VFIs where each PE is set to run, if necessary, at its own frequency. The $x_j(t)$ variable represents the utilization of the interface queue between the *j*-th and *l*-th VFIs. The $y_i(t)$ represent the utilization values of the interface queue between the *i*-th PE and *j*-th VFI. Note that various colors of the tiles in the above NoC imply that each island can run at a certain frequency.

where *h* is the time increment, $[(t\text{-}a)/h]$ represents the integer part of the ratio between $(t\text{-}a)$ and h. Equations (1) (continuous) and (2) (discrete) capture directly the role of the power law distributions observed in packet inter-arrival times (i.e., the term $(t\text{-}\tau)^{\alpha\text{-}n+1}$); they allow not only for a more accurate description of the dynamics of queue utilization $x(t)$, but also for a better optimization; this issue is discussed next.

IV. Power Management In Fractal Workloads

A. Problem Formulation

Next, we formulate the power management as an optimal control problem which takes into account the fractal characteristics of the NoC workload. As shown in Figure 3, we consider a VFI-based MPSoC architecture consisting of N_{PE} PEs, N_r routers, and N_j^q queues interfacing the router in the *j*- th VFI with other routers in the neighboring VFIs.

The goal of our nonlinear control problem is to find, for a given starting time (t_i) and a final time (t_f), the *optimal assignment* of operating frequencies for the PEs, routers, and queues, which minimize the quadratic costs of queues utilization with respect to a predefined reference, as well as the operating frequency of each VFI (this would implicitly minimize also the power consumption):

$$min \int_{t_i}^{t_f}\left\{\sum_{i=1}^{N_{PE}}\left[\frac{w_i}{2}\left(y_i(t)-y_i^{ref}\right)^2+\frac{z_i}{2}f_i^{\,2}(t)\right]+\right.$$
$$\left.+\sum_{j=1}^{N_r}\left[\frac{r_j}{2}f_j^{\,2}(t)+\sum_{k=1}^{N_j^q}\left[\frac{q_k}{2}\left(x_k(t)-x^{ref}\right)^2\right]\right]\right\}dt \qquad (3)$$

subject to the constraints given in Eq. (4) through Eq. (6) :

$$\frac{d^{\alpha_i}y_i(t)}{dt^{\alpha_i}}=a_i(t)y_i(t)+b_i(t)f_i-c_i(t)f_j, \qquad (4)$$

$$0<y_i^{min}\le y_i(t)\le y_i^{max}<1, \qquad i=1,..,N_{PE}$$

where $y_i(t)$ and y_i^{ref}, for i = 1,…,N_{PE}, are the actual utilization and the utilization reference of the queue between the *i*-th PE and its corresponding router, $x_k(t)$ and x_k^{ref} for k = 1,.., N_j^q are the actual and the reference utilization of the *k*-th queue located between the routers in *j*-th and *l*-th VFIs.

In (3), w_i, z_i, r_j and q_k are some positive weighting coefficients. In (4), α_i is the fractional order which depends on the fractal dimension characterizing the queue utilization process $y_i(t)$, $a_i(t)$ represents the weighting coefficient of the utilization $y_i(t)$, b_i and $c_i(t)$ reflect the contributions of the writing frequency (f_i) and the reading frequency (f_j), y_i^{min} and $y_i^{max}$are the admissible lower and upper bounds on the queue utilization $y_i(t)$. Of note, the optimal controller allows the designer to set individual weighting coefficients (i.e., w_i, z_i, r_j and q_k) in (3) for each of the NoC components such that the major power consumption elements can have a higher impact on the overall cost function.

The next set of constraints is meant to characterize the utilization of queues between neighboring VFIs:

$$\frac{d^{\alpha_k}x_k(t)}{dt^{\alpha_k}}=a_k(t)x_k(t)+b_k(t)f_k-c_k(t)f_l, \qquad (5)$$

$$0<x_k^{min}\le x_k(t)\le x_k^{max}<1, \qquad k=1,..,N_j^q.$$

where α_k is the fractional order characterizing the queue utilization process $x_k(t)$, $a_k(t)$ represents the contribution of utilization $x_k(t)$ to the entire queues utilization dynamics, $b_k(t)$ and $c_k(t)$ are the coefficients of the writing frequency (f_k) and the reading frequency (f_l) respectively, x_k^{min} and x_k^{max} are the lower and upper bounds on the utilization $x_k(t)$.

Note that the cost function in (3) maintains *all* NoC queues at specific utilization references (see the squared differences between $y_i(t)$ and y_i^{ref} or $x_k(t)$ and x_k^{ref}), while the control inputs (i.e., operating frequencies) are prevented from taking exceedingly large values which would induce a too high power consumption. The role of the optimal controller is to select the *minimum* operating frequencies for which the performance constraints are satisfied. Moreover, in order to prevent the nonlinear controller from selecting an unacceptable range of operating frequencies, we also impose the following constraints:

$$f_i^{\min}\le f_i\le f_i^{\max}, i=1,...,N_{PE}, \;\; f_j^{\min}\le f_j\le f_j^{\max}, j=1,...,N_r \qquad (6)$$

where f_i^{min} and f_i^{max} are the lower and upper frequency bounds at which each PE can run, f_j^{min} and f_j^{max} are the lower and upper frequency bounds for each router.

Note that we introduce two indices *i*- for the PEs and *j*- for the routers and implicitly two variables (i.e., f_i and f_j) such that the operating frequencies of the PEs are decoupled from the router frequencies; this way, we avoid setting the PE to a small frequency which may affect the computational performance requirements, or setting the router to a too high frequency when it may not be necessary. Since considering a single VFI for each router would introduce further complexity (due to a larger number of mixed-clock queues), we limit ourselves at considering that the N_r consists of just a few VFIs and include more constraints to reflect the fact that neighboring routers are operating at the same frequency.

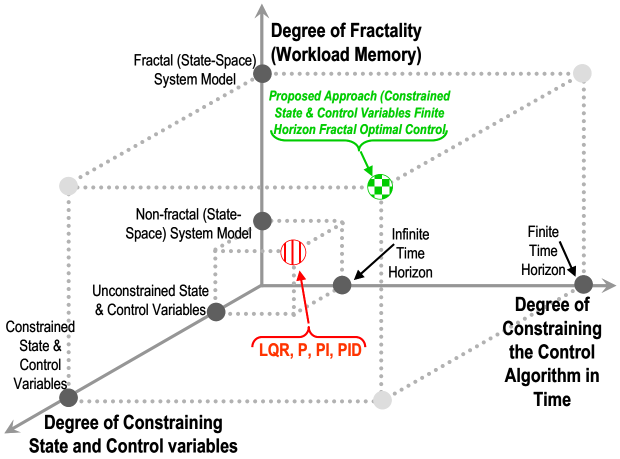
Degree of Fractality (Workload Memory)
Fractal (State-Space) System Model
Proposed Approach (Constrained State & Control Variables Finite Horizon Fractal Optimal Control)
Non-fractal (State-Space) System Model
Infinite Time Horizon
Finite Time Horizon
Unconstrained State & Control Variables
LQR, P, PI, PID
Constrained State & Control Variables
Degree of Constraining the Control Algorithm in Time
Degree of Constraining State and Control variables

Figure 4. Power management problem seen from three different perspectives: Degree of fractality of state-space model (Z-axis), degree of constraining the state and control variables (Y-axis), and scale (finite vs. infinite) of time constraints (X-axis). The red (striped) dot corresponds to an infinite time horizon optimal control problem with no constraints on state and control variables relying on a non-fractal state-space model. Particular cases of this problem are linear quadratic regulator (LQR) and proportional-integral-derivative (PID) controller. Our approach (the green chess dot) refers to a *finite-horizon optimal* control problem which operates with constraints on both state and control variables and relies on a fractal state-space model.

B. Significance of Fractal Power Management Problem

To put our power management approach into a broader perspective, we look at the optimal control problem defined in equations (3), (4), (5), and (6) from three different perspectives (see Figure 4), namely: *i*) the degree of fractal behavior exhibited by the system (state-space) model (Z-axis in Figure 4), *ii*) the degree of constraining imposed on the state and control variables (Y-axis), and *iii*) the scale (infinite vs. finite) of the timing constrains imposed on the control algorithm (X-axis).

The classical optimal control refers to infinite time horizon approaches that rely on first order derivative state-space models with no constraints imposed on state and control variables (see the striped red dot in Figure 4). We note that a linear quadratic regulator (LQR) approach as in [12][25] is a particular case of our problem formulation by considering $\alpha_i = \alpha_k = 1$, for $i = 1,\ldots,N_{PE}$, $k = 1,\ldots,N_j^q$, $t_f = \infty$ and imposing no constraints on the state variables in (4) and (5), and control signals in (6).

By the same token, a proportional-integral-derivative (PID) type of controller as in [39][40] can be also obtained as a particular case of our approach by considering $z_j = r_j = 0$ and $\alpha_i = \alpha_k = 1$, for $i = 1,\ldots,N_{PE}$ and $k = 1,\ldots,N_q$, infinite time horizon $t_f = \infty$ and imposing no constraints on the state variables in (4) and (5), and control signals in (6). So, it becomes clear now that the newly proposed problem formulation (shown as the chessboard green dot in Figure 4) is more general than either classical LQR- or PID-type of approaches. Given its generality, our new formalism allows to handle the high variability that occurs in real workloads.

Next, we derive the necessary and sufficient conditions for optimal control using convex optimization concepts; we also show how the constrained finite horizon fractal control problem can be solved efficiently via linear programming.

C. New Algorithm for Optimal Controller Synthesis

To solve the power management problem, we use concepts from the optimization theory and first construct the Lagrangian functional, L(y_i, f_i, λ_i, x_k, f_j, $\gamma_{k,j}$), as follows:

$$L(y_i, f_i, \lambda_i, x_k, f_j, \gamma_{k,j}) = \int_{t_i}^{t_f} \left\{ \sum_{i=1}^{N_{PE}} \left[\frac{w_i (y_i(t) - y_i^{ref}(t))^2}{2} + \frac{z_i f_i^2(t)}{2} + \lambda_i \left(\frac{d^{\alpha_i} y_i(t)}{dt^{\alpha_i}} - a_i(t) y_i(t) - b_i(t) f_i(t) + c_i(t) f_j(t) \right) \right] + \sum_{j=1}^{N_r} \sum_{k=1}^{N_j^q} \left[\frac{r_j f_j^2(t)}{2} + \frac{q_k (x_k(t) - x_k^{ref}(t))^2}{2} + \gamma_{k,j} \left(\frac{d^{\alpha_k} x_k(t)}{dt^{\alpha_k}} - a_k(t) x_k(t) - b_k(t) f_k + c_k(t) f_l \right) \right] \right\} dt \qquad (7)$$

where $y_i(t)$ and $x_k(t)$ denote the queue utilization variables, f_i is the frequency associated with the *i*-th PE, f_j is the frequency associated with the *j*-th router, λ_i is the Lagrange multiplier associated with the constraint imposed for the queue between the PE and the router, and $\gamma_{k,j}$ are the Lagrange multipliers associated with the constraints on the queue between neighboring routers in different VFIs.

For completeness, we also have to add some boundary constraints on the utilization of mixed clock queues:

$$y_i(t = t_i) = y_i^0 \quad y_i(t = t_f) = y_i^{ref} \quad i = 1, \ldots, N_{PE} \qquad (8)$$
$$x_k(t = t_i) = x_k^0 \quad x_k(t = t_f) = x_k^{ref} \quad j = 1, \ldots, N_r, k = 1, \ldots, N_j^q$$

These conditions are required in order to satisfy a certain performance level from the computation standpoint.

By expanding the Lagrangian in (7) via the Taylor formula and considering that it attains its minimum in the vicinity of $\tau = 0$, i.e., $\frac{\partial L}{\partial \tau} = 0$; we obtain the next relations:

$$\frac{\partial L}{\partial y_i} + {}_{t}D_{t_f}^{\alpha_i} \frac{\partial L}{\partial\, {}_{t_i}D_t^{\alpha_i} y_i} = 0 \qquad \frac{\partial L}{\partial f_i} = 0 \qquad i = 1, \ldots, N_{PE} \qquad (9)$$
$$\frac{\partial L}{\partial x_k} + {}_{t}D_{t_f}^{\alpha_k} \frac{\partial L}{\partial\, {}_{t_i}D_t^{\alpha_k} x_k} = 0, \quad \frac{\partial L}{\partial f_j} = 0 \quad j = 1, \ldots, N_r, k = 1, \ldots, N_j^q$$

where ${}_{t}D_{t_f}^{\alpha_i}$ and ${}_{t_i}D_t^{\alpha_k}$ represent the fractional derivatives operating backward and forward in time, respectively.

In order to solve the equations in (9), we discretize the interval [t_i, t_f] into N intervals of size $(t_f - t_i)/N$ and use the formula in (2) to construct a linear system which can be solved using LU decomposition. In short, the algorithm of the optimal controller synthesis works as follows:

a) ***Step 1****:* From the system identification module, read the coefficients α_i, a_i, b_i, c_i for $i=1,\ldots,N_{PE}$ characterizing the dynamics of queues between PEs and routers, and α_k, a_k, b_k, c_k for $j = 1,\ldots,N_r$, $k = 1,\ldots,N_j^q$ describing the dynamics of queues between neighboring routers in different VFIs;

b) ***Step 2****:* For a fixed number of discrete steps N, compute the coefficients obtained after discretizing the fractional derivatives in (4), (5) and (9) using the formula in (2) and construct a linear system, where the unknown variables are represented by the operating frequencies (i.e., f_i, f_j) and Lagrange multipliers (i.e., λ_i and $\gamma_{k,j}$);

c) ***Step 3****:* Solve the linear system in (9) and find the operating frequencies for each VFI in the NoC architecture.

V. Experimental Setup and Results

To evaluate our fractal optimal control algorithm, we consider a combination of trace-driven and cycle-accurate simulation of a VFI-based NoC architecture. From an application perspective, we consider four 16-node multi-threaded commercial workloads (i.e., *Apache HTTP server v2.0* from SPECweb99 benchmark [33], on-line transaction processing consisting of TPC-C v3.0 workload on *IBM DB2 v8 ESE* and *Oracle 10gExterprise Database Server*, blocked sparse Cholesky factorization and *ocean* simulation) obtained from a FLEXUS-based shared-memory 16-processor environment consisting of cycle accurate models of out-of-order processors and cache hierarchy [36][38].

From an architectural perspective, we consider a 4×4 mesh NoC employing XY wormhole routing with mixed clock queues of 10 flit size and packets consisting of 15 flits. In this setup, we consider that the execution of a set of applications is divided into control intervals of 20μs length. Of course, the proposed control algorithm can also work with larger time intervals but the model parameters need to be estimated for the new time scale.

A. New Power Management Methodology

For each control interval, the parameter identification module (PIM) estimates the fractional exponent α_k in (5) in two stages: First, it computes the workload variation coefficients for $log_2(m)$ resolution scales based on queue utilizations (m is the size of the control interval). Second, it performs a linear regression between the resolution scales and the variation coefficients. Because this approach relies only on variation coefficients at various time scales (as opposed to the entire time series which would be needed for any linear regression method [14]), the PIM module not only reduces the computational complexity from a $O(N^3)$ order to a linear order $O(N)$, but it also enables an on-line estimation procedure with minimum memory overhead (i.e., it does not require to store all queue utilizations and can be done iteratively whenever there is a change in queue occupancy).

Next, the PIM module estimates parameters a_k, b_k, and c_k from arrival (A_k), departure (D_k) and queue utilization (X_k) processes by solving a 10×3 linear system with three unknowns a_k, b_k, and c_k.

After the identification step is completed, and in parallel with the application computations, the fractal optimal controller implemented in the Power Manager (PM) module reads these parameters and solves the linear system defined by the equations (9) to determine the optimal operating frequencies for the PEs and routers (for the next interval of 20µs) that can ensure a predefined performance level specified in terms of queue utilization references.

B. Hardware Complexity of the Power Manager

To illustrate the hardware requirements of the PM module, we consider that the optimal controller solves the optimality conditions in (9) for 30 discretization steps. Implementing the controller in Verilog and synthesizing it using the Xilinx's SXT/Isim on a Virtex4 FPGA (Device: XC5VLX30, Package: FF324), for controlling a large multi-VFI platform with 80 queues we would need about 19440 registers, 19120 LUTs, 80 RAM/FIFOs and 480 DSP48Es in a basic, un-optimized, FPGA design. This looks very reasonable for all practical purposes; with some additional optimization, these requirements can be reduced even further. Alternatively, the linear program solver in (9) can be implemented in software.

C. Power Management under Real Workloads

Next, we apply the optimal controller for $N = 30$ discrete steps and a 4×4 mesh NoC running an *Apache HTTP* webserver application; the objective is to determine the operating frequencies such that the queues utilization remains below 0.1. The reason for bounding the maximum utilization of the queues at a reference value of 0.1 is two-fold: First, a small utilization of the queues implies smaller packet waiting times in buffers and, implicitly, smaller source-to-destination latencies. Second, a small queue utilization, also minimizes the chances of thermal hotspot buildup; this can improve the chip reliability significantly.

Note also that the number of discrete steps (N) chosen for discretizing equations (4), (5) and (9) influences the precision of computing the operating frequencies. Thus, when less precision is needed in terms of operating frequency, we can use a smaller N value (e.g., 5 to 10).

For illustration purposes, we consider the case of $N = 30$ discrete steps and solve the linear system describing the dynamics of the same large multi-VFI platform with 80 queues in less than 1µs. This clearly illustrates that our approach is suitable for online power management of future multicore platforms.

Figure 5.a shows the utilization of a few queues at tiles (0,0), (0,2), (1,1), (1,2), (2,2), (2,1), and (3,3). We can observe that the optimal controller is able to bring the utilization of these tiles below the reference value of 0.1. Figure 5.b) shows the operating frequencies of the tiles at (0,0), (0,2), (1,1), (1,2), (2,2), (2,1), and (3,3) needed to attain the imposed reference values. Moreover, Figure 5.c) shows the utilization of the queues without and with fractal optimal control. The optimal controller is able to keep the utilization of all queues below 0.1 by adjusting the operating frequencies of all PEs and routers.

For completeness, we have also applied the optimal controller to a 4×4 mesh NoC running Cholesky factorization with a maximum reference value of 0.4 allowed for utilization of all queues. Note that, by imposing a value of 0.4 for queue utilization in this experiment, the impact on average packet latency is less than 4%, while the power savings are about 40%.

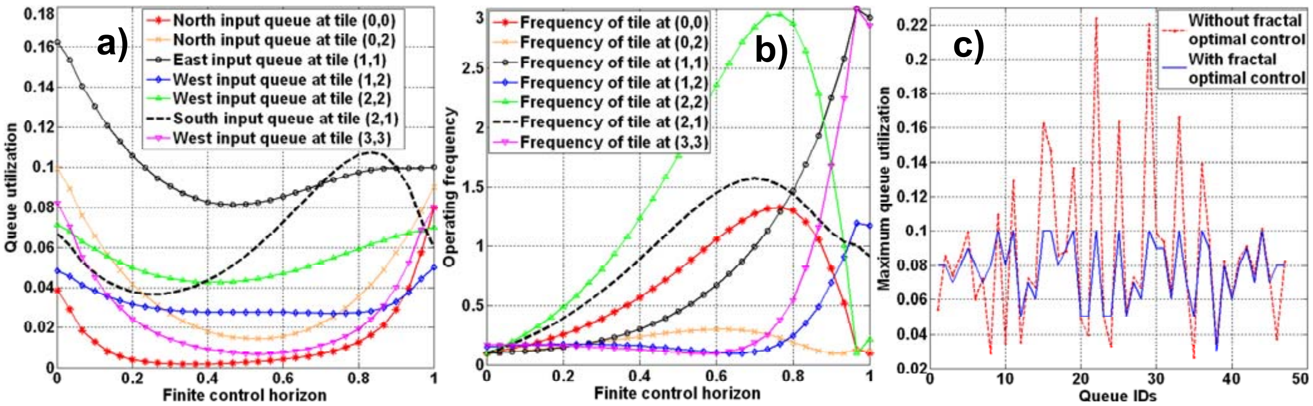

Figure 5. a) Utilization of the queues at tiles (0,0), (0,2), (1,1), (1,2), (2,2), (2,1), and (3,3) for a 4×4 mesh NoC running Apache HTTP webserver application. b) The variation of the operating frequencies for the routers at (0,0), (0,2), (1,1), (1,2), (2,2), (2,1), and (3,3) necessary to reach the reference values imposed on the utilization of all queues. c) Comparison between the utilization of all queues in the uncontrolled case and the queue utilization for the case of the fractal optimal controller described in Section IV.C.

D. Comparison Between our Approach and Classical Approaches for Power Management

To illustrate the difference our fractal controller (FC) can make in terms of power savings, we consider next a 4×4 NoC running one of the burstiest applications among all the benchmarks we tested (namely, the *Apache HTTP* webserver) and compare the power consumption observed under three different PM approaches, i.e., PM with PID, PM with LQR, and PM with FC, against a baseline platform consisting of a homogeneous NoC running at 3GHz. More precisely, tin order to better understand the mathematical underpinnings of this comparison, for the PM based on PID control (as in [40]) and for the PM based on LQR (as in [12]), the model behind the controller is based on integer order differential equations that characterize the queues utilization at run time. In contrast, for the FC approach, the queue utilization is modeled using fractional calculus as discussed in Section IV.

As we can see from data in Table 1, the PM based on PID turns out to oscillate because it sometimes selects exceedingly high frequencies. Consequently, an approach based on PID control cannot deal with the kind of burstiness a real workload like *Apache HTTP* exhibits. The PM based on LQR control does a better job and reduces the overall power consumption of the system to approximately 70% of the power consumption of the baseline platform. In contrast, the PM based on FC makes the platform consume only about 30% of the dynamic power consumption of the baseline. This shows that the PM based on fractional control performs significantly better than classical control approaches for power management like PID or LQR which can get trapped in local minima for such bursty workloads.

Of note, the power savings for PM with LQR can further decrease, if no bounds on maximum frequency are set. However, the fractal optimal controller allows to find the optimal solution; this enables the highest amount of power savings to be achieved (as shown, almost 2X power savings compared to LQR).

TABLE I. NORMALIZED POWER CONSUMPTION VALUES TO THE HOMOGENEOUS BASELINE PLATFORM AND THREE PM APPROACHES: PID, LQR, AND FC WHILE RUNNING A BURSTY APPLICATION (APACHE HTTP).

No PM (baseline) Normalized power consumption	**PM with PID**	**PM with LQR**	**PM with FC**
1	cannot stabilize	0.70	0.30

Similarly, by performing other experiments with the *Cholesky factorization* benchmark, we observe that our approach leads to 40% power savings compared to the homogeneous baseline where PEs and routers run at 3GHz. Finally, by applying the proposed algorithm and comparing the savings with NoC architectures running at 3GHz leads to 50% and 20% power savings for *ocean* simulation and *on-line transaction processing* application, respectively.

VI. CONCLUSION

In this paper, we have addressed the problem of power management in multicore architectures where computational workloads are highly complex and exhibit fractal characteristics. Towards this end, we have proposed a new modeling approach based on the dynamics of queue utilization and fractional differential equations. This fractal model is used to formulate an optimal control problem for dynamic power management which determines the necessary operating frequencies such that the NoC queues reach and remain at their target reference values for bursty workloads.

ACKNOWLEDGMENTS

The authors would like to thank to anonymous reviewers for their constructive comments and helpful suggestions. The material presented herein was supported by the NSF under CCF-0916752 grant and GRC under 2011-HJ-2168 and 2012-HJ-2232 grants. R. T. Gavila has been supported by the Spanish MICINN, European Commission FEDER funds, and University of Valencia under grants Consolider-Ingenio 2010, CSD2006-00046, TIN2009-14475-C04-04, and V SEGLES program.

REFERENCES

[1] O. P. Agrawal et al., "Fractional Optimal Control Problems with Several State and Control Variables," *J. of Vibration and Control*, vol. 16 (13) , pp. 1967-1976, May, 2010.

[2] A. Alimonda et al., "A Feedback-based Approach to DVFS in Data-Flow Applications," *IEEE Trans. Comp.-Aided Des. Integ. Cir. Sys*., 28, 11, pp. 1691-1704, 2009.

[3] M. Arjomand and H. Sarbazi-Azad, "Voltage-Frequency Planning for Thermal-Aware, Low-Power Design of Regular 3-D NoCs," 23*rd Intl. Conf. on VLSI Design*, pp.57-62, January 2010.

[4] D. Atienza et al., "Network-on-Chip Design and Synthesis Outlook," *Integr. VLSI J.*, vol. 41, Issue 3, pp. 340-359, May 2008.

[5] L. Benini and G. De Micheli, "Networks on chip: a new SoC paradigm," *IEEE Computer*, vol. 35, no. 1, January 2002.

[6] D. Bertozzi et al., "Energy-Reliability Trade-off for NoCs," in Networks on chip, Kluwer, 2003.

[7] P. Bogdan and R. Marculescu, "Statistical Physics Approaches for Network-on-Chip Traffic Characterization," 7*th IEEE/ACM intl. conf. on Hardware/software codesign and system synthesis* (CODES+ISSS '09), pp. 461-470, 2009.

[8] P. Bogdan and R. Marculescu, "Workload characterization and its impact on multicore platform design," 8*th IEEE/ACM/IFIP intl. conf. on Hardware/software codesign and system synthesis* (CODES/ISSS'10), pp. 231-240, 2010.

[9] P. Bogdan and R. Marculescu, "Non-Stationary Traffic Analysis and Its Implications on Multicore Platform Design," *IEEE Trans. Comp.-Aided Des. Integ. Cir. Sys*., vol. 30(4), pp. 508-519, April 2011.

[10] A. K. Coskun et al., "Proactive Temperature Balancing for Low Cost Thermal Management in MPSoCs," *IEEE/ACM Intl. Conf. on Computer-Aided Design* (ICCAD), pp. 250-257, November 2008.

[11] A.K. Coskun et al., "Dynamic thermal management in 3D multicore architectures," *Design, Automation and Test in Europe* (DATE'09), pp. 1410-1415, April 2009.

[12] S. Garg et al., "Custom Feedback Control: Enabling Truly Scalable On-Chip Power Management for MPSoCs," 16*th ACM/IEEE Intl. Symp. on Low Power Electronics and Design* (ISLPED '10), pp. 425-430, October 2010.

[13] Y. Ge et al., "Distributed Task Migration for Thermal Management in Many-core Systems," 47*th Design Automation Conf.* (DAC '10), pp. 579-584, June 2010.

[14] P. Van Den Hof, *System Identification*, Elsevier, 2004.

[15] J. Howard et al., "A 48-Core IA-32 Message-Passing Processor with DVFS in 45nm CMOS," *Intl. Solid-State and Circuits Conf.* (ISSCC), pp. 108 - 109, February 2010.

[16] C. Isci et al., "An Analysis of Efficient Multi-Core Global Power Management Policies: Maximizing Performance for a Given Power Budget," 39*th IEEE/ACM Intl. Symp. on Microarchitecture* (MICRO'39), pp. 347 - 358, December 2006.

[17] A. Jantsch and H. Tenhunen, Networks-on-Chip, 2003.

[18] H. Jung and M. Pedram, "Supervised Learning Based Power Management for Multicore Processors," *IEEE Trans. Comp.-Aided Des. Integ. Cir. Sys.*, vol. 29, no. 9, pp. 1395-1408, September 2010.

[19] A. Lungu et al.,"Multicore Power Management: Ensuring Robustness via Early-Stage Formal Verification," 7*th IEEE/ACM intl. conf. on Formal Methods and Models for Codesign* (MEMOCODE'09), pp. 78-87, July 2009.

[20] N. Madan et al., "A Case for Guarded Power Gating for Multi-Core Processors," 17*th IEEE Intl. Symp. on High-Perf. Comp. Arch*., pp. 291-300, February 2011.

[21] B. Mandelbrot, *Gaussian, Self-Affinity and Fractals*, Springer, 2002.

[22] R. Marculescu and P. Bogdan, "The Chip Is the Network:Toward a Science of Network-on-Chip Design," *Foundations and Trends in Electronic Design Automation*, Vol. 2, No 4, pp 371-461, 2009.

[23] A.K. Mishra et al.,"CPM for CMPs:Coordinated Power Management for Chip-Multiprocessors," *Intl. Conf. for High Performance Computing, Networking, Storage and Analysis* (SC '10), pp. 1-12, November 2010.

[24] R. Mukherjee and S.O. Memik. "Physical Aware Frequency Selection for Dynamic Thermal Management in Multi-core Systems," *IEEE/ACM Intl. Conf. on Computer-Aided Design* (ICCAD), pp. 547-552, November 2006.

[25] U.Y. Ogras et al.,"Design and Management of Voltage-Frequency Island Partitioned Networks-on-Chip," *IEEE Trans. on VLSI Syst*., vol. 17, Issue 13, pp. 330-341, 2009.

[26] P. P. Pande et al., "Sustainability Through Massively Integrated Computing: Are We Ready to Break the Energy Efficiency Wall for Single-Chip Platforms?" *Design, Automation & Test in Europe Conf.* (DATE'11), pp. 1-6, March 2011.

[27] I. Podlubny, *Fractional Differential Equation. An Introduction to Fractional Derivatives*, Academic Press, 1999.

[28] K.K. Rangan et al., "Thread Motion: Fine-Grained Power Management for Multi-Core Systems," *SIGARCH Comput. Archit. News*, vol. 37, Issue 3, pp. 302-313, June 2009.

[29] P. Ranganathan, "From Microprocessors to Nanostores: Rethinking Data-Centric Systems," *IEEE Computer,* vol 44, Issue 1, pp. 39-48, January 2011.

[30] S. Sharifi et al., "Hybrid Dynamic Energy and Thermal Management in Heterogeneous Embedded Multiprocessor SoCs," *Asia and South Pacific Design Automation Conf.* (ASPDAC '10), pp. 873-878, January 2010.

[31] S. Somogyi, et al., "Spatial Memory Streaming," 33*rd annual international symposium on Computer Architecture* (ISCA '06), pp. 252-263, July 2006.

[32] Standard Performance Evaluation Corporation, Inc. SPECweb99 Benchmark. http://www.spec.org/osg/web99.

[33] C. Yen-Kuang et al., "Convergence of Recognition, Mining, and Synthesis Workloads and Its Implications," *Proceedings of IEEE*, vol. 96, Issue 5, pp. 790 – 807, May 2008.

[34] F. Zanini et al., "A Control Theory Approach for Thermal Balancing of MPSoC," *Asia and South Pacific Design Automation Conf.* (ASPDAC '09), pp. 37 – 42, January 2009.

[35] K. Zhou, J.C. Doyle, K. Glover, *Robust and OptimalControl*, 1996.

[36] T. F. Wenisch et al., "Simflex: Statistical Sampling of Computer System Simulation," *IEEE Micro*, vol.26, no.4, pp.18-31, July - August 2006.

[37] J. A. Winter et al., "Scalable Thread Scheduling and Global Power Management for Heterogeneous Many-Core Architectures," 19*th intl. conf. on Parallel Architectures and Compilation Techniques* (PACT '10), pp. 29 – 40, September 2010.

[38] S. C. Woo et al., "The SPLASH-2 Programs: Characterization and Methodological Considerations," *Intl. Symp. on Comp. Arch*., pp. 24-36, June 1995.

[39] Q. Wu et al., "Formal Online Methods for Voltage/Frequency Control in Multiple Clock Domain Microprocessors," 11*th Intl. Conf. on Architectural Support for Programming Languages and Operating System* (ASPLOS'04), pp. 248-259, October 2004.

[40] Q. Wu et al., "Formal Control Techniques for Power-Performance Management," *IEEE Micro*, vol.25, no.5, pp. 52-62, September - October 2005.