2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications

Energy-Performance Modeling and Optimization of Parallel Computing in On-Chip Networks

Shuai Zhang, Zhiyong Liu, Dongrui Fan, Fonglong Song, Mingzhe Zhang
State Key Laboratory of Computer Architecture
Institute of Computing Technology, Chinese Academy of Sciences
{zhangshuai, zyliu, fandr, songfenglong, zhangmingzhe} @ict.ac.cn

***Abstract*—This paper discusses energy-performance trade-off of networks-on-chip with real parallel applications. First, we propose an accurate energy-performance analytical model that conduct and analyze the impacts of both frequency-independent and frequency-dependent power. Second, we put together the communication overhead, memory access overhead, frequency scaling, and core count scaling to quantify the performance and energy consumed by NoCs. Third, we propose a new energy-performance optimization method, by choosing a pair of frequency and core count to get optimal energy or performance. Finally, we implement eight PARSEC parallel applications to evaluate our model and the optimization method. The experiment result confirms that our model predicts NoCs energy and performance well, and selects correct frequency level and core count for most parallel applications.**

Keywords—Energy; Performance; NoC

I. Introduction

Over the last few years, energy has become a first concern to processor designers and manufacturers [8, 20]. Computing represents a significant source of electricity energy and creates a critical cost problem for sustainability [1]. As semiconductor technology has entered deep submicron (DSM) sizes, low threshold voltage results in an exponential increase of the sub-threshold leakage current, leakage power has become a considerable proportion of the total power for current silicon technologies (e.g. 30%~40% of the total power of multi-core is leakage power.[41]). DVFS (Dynamic Voltage/Frequency Scaling) technique is widely chosen to manage dynamic power in multiprocessor systems[7, 13, 31], and DPM (Dynamic Power Management) technique is used to cut off the static power (including leakage power) by making the idle cores sleep or shut down. However, decreasing frequency level and turning off the idle cores both damage the processor performance, so we need to build a smart policy to choose the proper frequency level and the number of active cores to balance the energy and performance tradeoff.

With future CMPs likely supporting many threads and their cores supporting a number of voltage and frequency levels, it's hard to find an optimal energy-performance operating point for a parallel application, in particular at run-time. Synchronization and communication constitute the overheads growing rapidly as we increase the number of cores. On the other hand, the beneficial effect of the increased cache capacity may yield an increase in performance as the core number scales. Uncertainty penetrates in the relationship among performance, energy, parallel overhead, shared cache capacity, core number, and voltage/frequency levels. It is not obvious which voltage and frequency level should be applied, in combination with the appropriate number of cores. An accurate prediction of energy consumption and parallel performance would improve our understanding and enable optimization of energy-performance tradeoff. Similarly, such model is also lacking in the literature of NoCs designs.

The Networks-on-chip (NoCs) is a critical element of a multi-core architecture. On an 8-core processor, for example, even under conservative assumptions, an interconnection can consume the power equivalent of one core [15]. Paper [12] shows that as technology scales, NoCs leakage power becomes increasingly significant from 2.5% of total power at 180nm technology to a 60% at 70nm if frequency is kept invariant. If assuming frequency doubles each process generation, NoCs leakage still rises to 27% at 70nm. Hence the leakage power plays a more important role in total power of NoCs. This has led to many researches on modeling [9, 11] and optimizing [8, 10, 11] the power consumption of interconnection networks. The number of NoC nodes scales as the processor cores, therefore, the frequency and node count also influence the energy-performance tradeoffs of NoCs.

However, to the best of our knowledge, there doesn't exist a model can be adopted to analyze the impacts of all the factors of frequency-independent power, parallel overhead, off-hip memory access overhead, number of nodes and frequency on energy-performance tradeoffs of NoCs. In this paper, we attempt to build such an accurate model based on a classic 2D-Mesh NoC and real applications. The main contributions are:

- We develop an accurate energy-performance model for NoCs. Compared to the traditional models, we conduct both parallel overhead and frequency-independent power into energy-performance model. For the first time, we put together the communication overhead, memory access overhead, frequency scaling, core count scaling and frequency-independent power to quantify the performance and energy consumed by NoCs.
- Furthermore, We propose a new energy-performance optimization method, by manipulating the frequency

978-0-7695-5022-0/13 $26.00 © 2013 IEEE
DOI 10.1109/TrustCom.2013.107

level and core number to get optimal energy or performance with constraints. The experiment results confirm that our model predicts NoCs energy and performance behavior well, and we exactly select correct optimal frequency level and core number.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 presents the chip-level power model and NoCs power model. Section 4 describes the performance model of the chip. Section 5 presents the new energy-performance model and defines two optimization problems. Section 6 introduces the applications and simulation platforms. Section 7 presents and discusses our evaluation results and Section 8 concludes the paper.

II. RELATED WORK

There is a large collection of literatures on the energy and performance of multicore processors. In [3], Amdahl's law is extended to describe energy efficiency. But it's aimed at how to improve the energy efficiency from homogeneous to heterogeneous multicore processor, ignoring the impact of frequency scaling. Paper [14] proposed an energy optimization by frequency scaling, but it doesn't consider to shutdown idle cores to reduce leakage power, so the optimal solution is limited.

Models in [3, 6, 7, 9, 10] don't consider the frequency-independent power and parallel overhead, while [5, 8] consider both leakage power (including frequency-independent part) and performance models based on parallel efficiency (without parallel overhead). [36, 37] present the frequency-independent power but not including parallel overhead. Energy-performance models in [4, 9, 25, 27, 28, 32] consider parallel overhead but not including the frequency-independent power. [8] proposes an accurate power model, however it's hard to implement under the precise non-linear relation between voltage and frequency. Besides, some studies on multi-core chip assume the leakage power is a fixed ratio of the total power [5, 24], other works assume the leakage power is in direct proportion to frequency [10], but the frequency-independent power is eliminated under these two assumptions. In [8], it also proposes an energy-performance model based on a given parallel overhead. However, it doesn't consider the parallelism of applications and off-chip memory access. While in [24], it considers the parallelism of applications and use DPM and DVFS to get minimal energy, but it assumes the leakage power is a fixed ratio of the total power, so the frequency-independent power disappears.

In the recent researches, paper [25] focuses on power-aware speedup, to build up an energy-performance model with frequency-independent power and parallel overhead. It divides the workload into on-chip and off-chip parts.

So far, there doesn't exist an energy-performance model involving all the factors of communication overhead, memory access overhead, frequency scaling, core number scaling, frequency-independent power, and utilize both DVFS and DPM technologies to obtain optimal energy or performance. Therefore, we need to propose such a model to involve all the factors above.

III. POWER MODELING

A. On-Chip Power Model

In a computer system, the total power can be divided into two parts: frequency-dependent power and frequency-independent power [2, 33]:

$$P = P_{dep}(f) + P_{ind} \quad (1)$$

The frequency-dependent power $P_{dep}(f)$ consists of on-chip devices which are dependent on the on-chip clock, such as processor cores, on-chip network and cache. While frequency-independent power P_{ind} is mainly consumed by I/O device, disk, memory and off-chip lower-level cache, which can be removed and have a different clock rate from the chip.

In this subsection, our goal is to find out the frequency-independent expression in the power equation. Total power consumption of a multi-core chip is generally written as the sum of dynamic and static power [32, 40]:

$$P = P_d + P_d = ACV^2f + VI_{leak} \quad (2)$$

Where A is the faction of gate activity, C is the total capacitance, V is the supply voltage, f is the operating frequency, and I_{leak} is the leakage current.

$$f_{max} = \eta (V - V_{th})^{\alpha} / V \quad (3)$$

Equation (3) shows the dependency of maximum operating frequency on supply voltage, where α is an experimentally derived constant from 1 to 2 [36, 37]. The frequency-independent power is presented at $f \to 0$, as the voltage $V \to V_{th}$. Thus, we can write down frequency-independent power as follow:

$$P_{ind} = P(f = 0) = V_{th} I_{leak}(V_{th}) \quad (4)$$

However, the direct proportional relation between

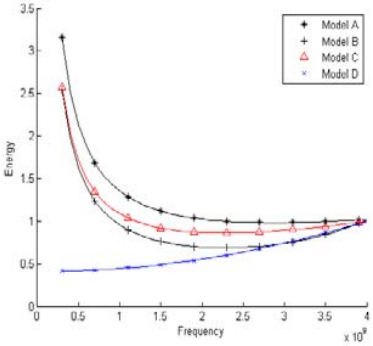

Figure 1. Each model is normalized to the energy at 3GHz. (65nm, traffic load=0.02 flits/cycle/port)

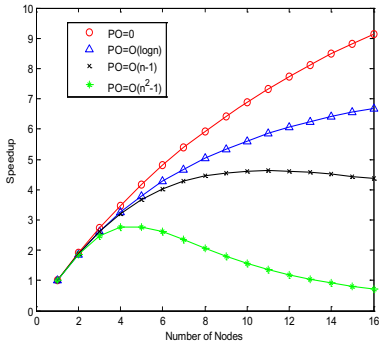

Figure 2. Speedup of three parallel overhead models.

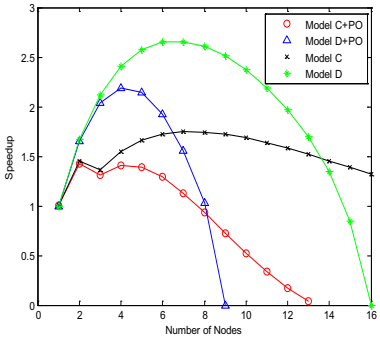

Figure 3. Speedup with energy budget, the budget is the energy consumption for N=1.

frequency and voltage is widely used in modeling power with voltage/frequency scaling. If we use

$$V = \beta_3 f \tag{5}$$

to replace (3). we obtain the frequency-independent power $P = P_{ind} = 0$, which is obviously out of practice.

Therefore, the only way to write out the accurate expression of frequency-independent power is using the nonlinear relation in (3). In the performance model, it is frequency but not voltage influences the performance directly. However, it's hard to express voltage as a function of frequency from (3). In the case of super-threshold DVFS, an approximately linear relationship of frequency and supply voltage can be writen as [30, 36]:

$$V = \beta_1 + \beta_2 f \tag{6}$$

Where $\beta_1 = V_{min}$ and $\beta_2 = (V_{max} - V_{min}) / f_{max}$. Equation (5) is mostly equivalent to (3). If $f \to 0$, we can get the same frequency-independent power as (4). To comparing the accuracy of different models, we list them as follows:

Model A: Power model based on (3), with α=1.

Model B: Power model based on (3), with α=2.

Model C: Power model based on (6).

Model D: Power model based on (5), voltage is assumed in direct proportion to frequency, which is widely used for DVFS.

We validate NoCs power simulator ORION2.0 [14] to evaluation these four models. From Fig.1, we observe that power curve of Model C runs between that of Model A and B all the time, but the curve of Model D deviates much from the trend of other Models, particularly at low frequency. Fig.1 shows Model C is an accurate simplified power model, it maintains the same trend with the nonlinear model of (3).

B. 2D-Mesh NoCs Power Model

Generally, NoCs power increases super-linearly as the number of nodes scale. This is because that the increase in both number of control wires and queuing latches are linear [11]. For simplicity, in this paper, the number of router buffers does not change with the node count, thus total power is almost linear increase with the node count. Since wires dissipate less leakage power compared to the routers, the proportion of leakage power is quite different for routers and links. We assume a 2D-Mesh NoC consists of only routers and links, their power can be represented as follows:

$$P_i^{(r)}(\sigma_i^{(r)}, f) = A(\sigma_i^{(r)})\, C^{(r)}\, V^2(f) f + V(f)\, I_{leak}^{(r)}(f) \tag{9}$$

$$P_j^{(l)}(\sigma_j^{(l)}, f) = A(\sigma_j^{(l)})\, C^{(l)}\, V^2(f) f + V(f)\, I_{leak}^{(l)}(f) \tag{10}$$

Where $\sigma_i^{(r)}$, $C^{(r)}$, $I_{leak}^{(r)}$ are respectively traffic load, active capacitance and leakage for router on the ith node, $\sigma_j^{(l)}$, $C^{(l)}$, $I_{leak}^{(l)}$ are respectively traffic load, active capacitance and leakage for jth link in the network. We use curve fitting to obtain a linearly increase relation between the power active factor A and traffic load σ, through inputting various traffic load in ORION2.0. Note that, there is also a considerable increase in the network traffic as number of cores grows, at the meantime, increasing cache leads to higher hit rates and less traffic on the interconnect. From simulation, we observe the traffic load grows very mildly with node count increasing. Hence, we use an average traffic load to replace that of each router and link. The total power is the sum of routers and links. If each link is bidirectional, the total power of 2D-Mesh NoC can be written as:

$$\begin{aligned} P(\sigma, f, n) &= \sum_{i=1}^{n} P_i^{(r)}(\sigma_i^{(r)}, f) + \sum_{j=1}^{4\left\lfloor n-\sqrt{n} \right\rfloor} P_j^{(l)}(\sigma_j^{(l)}, f) \\ &= nP_i^{(r)}(\sigma^{(r)}, f) + 4\left\lfloor n-\sqrt{n} \right\rfloor P_i^{(l)}(\sigma^{(l)}, f) \end{aligned} \tag{11}$$

Where $\sigma^{(r)}$ is the average value of $\sigma_i^{(r)}$, and $\sigma^{(l)}$ is the average value of $\sigma_j^{(l)}$.

IV. SPEEDUP MODELING

In this section, our goal is to describe overhead-bound multi-core performance in simple formulas.

We assume frequency scaling is used over the whole chip, voltage and frequency can be scaled continuously. Our analytical model assumes the network is built out of homogeneous nodes. The workloads are size-fixed multithread applications.

Extensions of Amdahl's Law have been developed for modeling *uncore* components, such as the interconnection network and last-level cache (LLC). Furthermore, some research consider the parallel penalty by data sharing and thread synchronization[22]. The speedup of an application can be written as:

$$S = \frac{T_s}{T_s(1-p) + T_s \cdot p / n + T_{com} + T_{llc} + T_{mem}} \tag{12}$$

Where p is the fraction of parallel execution and n is the number of cores. T_s is the total sequential execution time running on a single core. T_{com} is the overhead corresponding to the communication and T_{llc} is the overhead of thread synchronization and contention in the shared last-level cache. T_{mem} is the overhead of off-chip memory access. As the discussions in [21, 22, 23], generally T_{com} is an increasing function of n and clearly there is:

$$T_{com}(n=1) = 0 \tag{13}$$

This communication overhead is not only highly dependent on input size of workload, but also dependent on the core count, network topology and traffic pattern. Typically, Hennessy [19] gives the complexity on the number of cores of four famous applications: the communication overhead of LU Kernel, Barnes and Ocean scale as the $O(n^{1/2})$, and FFT scales as $O(n)$. Another work [18] shows that T_{com} scales as $O(n\log n)$ for quick sort. For varies topologies, T_{com} of hypercube network has a complexity of $O(\log n)$ [22]. For 2-D mesh NoCs, [17] presents the lower bounds on complexity of different traffic patterns: one-to-all broadcast is $O(\log n)$, all-to-all broadcast and one-to-all scatter are $O(n)$, and all-to-all scatter is $O(n^2)$. Since we only consider a 2-D mesh network on chip, the communication overhead of all traffic patterns should be considered. Therefore, we conduct three basic communication overhead models: sub-linear $O(\log n)$, linear $O(n)$, and super-linear $O(n^2)$.

Madhavan [29] concludes that the LLC overhead should scale sub-linearly with the number of coes. Therefore, the parallel overhead T_{PO} also ought to be presented as three basic models above, where $T_{PO} = T_{com} + T_{llc}$.

The overhead of off-chip memory access T_{mem} is independent on the number of cores, it only relates cache misses from LLC. In [26], if LLC doubles in capacity, the miss rate will decrease by 30%. [35] presents the relation between cache capacity and miss rate: $miss\_rate = \beta C^{\alpha}$, where β is positive and α is a negative constant, C represents cache capacity. Assuming each core has a C-size sharing cache, the memory overhead can be written as: $T_{mem} \propto \beta (nC)^{\alpha} = \beta C^{\alpha} n^{\alpha} = \lambda n^{\alpha}$, where $\lambda = \beta C^{\alpha}$ is a positive constant. For a fixed-size workload, the total sequential execution time T_s is a constant value. After a little rearranging on (12), the speedup can be expressed as:

$$S = \frac{1}{1 - p + p / n + w_{PO} + w_{mem}} \tag{14}$$

The parallel overhead fraction $w_{PO} = c\{\log n, n\text{-}1, n^2\text{-}1\}$, c is the coefficient of parallel overhead, which is a constant for a given application. [28] shows that no matter what parallel overhead model, the attainable speedup must decrease over some core count. Fig.2 also shows how these different parallel overhead models effect the performance as the core number scales. When frequency scaling is adopted over the chip, the speedup of multi-core should satisfy:

$$S(f, n) = \frac{f}{F \cdot (1 - p + p / n + w_{PO}) + f \cdot w_{mem}} \tag{15}$$

Where F is a reference frequency.

In order to predict the performance with varying number of cores, it is necessary to investigate on how to make an appropriate choose to fit the real applications out of the three parallel overhead models.

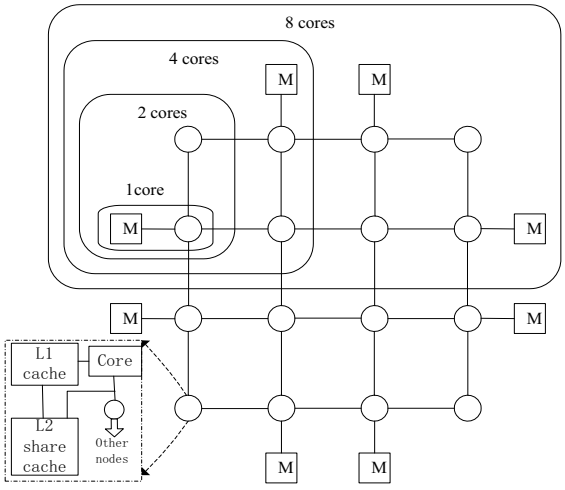

Figure 4. NoC Architecture

V. ENERGY-PERFORMANCE ANALYSIS AND OPTIMIZATION FOR NOCS

A. NoCs Energy-Performance Optimization Problems

Assuming t_n is the total execution time on n cores, the total energy consumed by NoCs is written as:

$$E = P(\bar{\sigma}, f, n) \cdot t_n$$

We rearrange (15) to present frequency:

$$f = S \cdot F \cdot (1 - p + p / n + w_{PO}) / (1 - S \cdot w_{mem})$$

The parallel fraction and parallel overhead are both dependent on applications, thus the frequency is a function of node count, target speedup and applications: $f = f(S, n, App)$.

The total execution time $t_n = T/S$, where T is the execution time on a single core with reference frequency. Hence the total energy can be represented as:

$$E = P(\sigma(App), f(S, n, App), n) \cdot T/S = P(S, n, App) \cdot T/S$$

In our study, we assume that the processor supports only a single application running on the chip. For a specified application, we define the energy optimization Problem I:

$$\text{Min } E = E(S, n, App)$$

$$\text{s.t. } f = S \cdot F \cdot (1 - p + p / n + w_{PO}) / (1 - S \cdot w_{mem}) < f_{max}$$

$$S \geq S_{target}$$

We also present another optimal problem, to obtain the maximum speedup with an energy budget, Problem II:

$$\text{Max } S = S(S, n)$$

$$\text{s.t. } f = S \cdot F \cdot (1 - p + p / n + w_{PO}) / (1 - S \cdot w_{mem}) < f_{max}$$

$$E(S, n, App) \leq E_{target}$$

Fig.3 illustrates the speedup under a energy target. For Model C+PO in Fig.3, there are two similar maximum speedup values at N=2 and N=4. We should choose the solution with less number of nodes for saving on-chip resource.

B. Optimal Energy-Performance Management

Based on the analysis above, we propose a method to establish the energy-performance model for a real parallel application on a given NoC.

After optimization, we obtain the optimal number of nodes and frequency pair (n_E, f_E) for Problem I and (n_S, f_S) for Problem II. The method is as follow:

Step 1: Execute the application for m times, $m \geq 3$. For the ith times, record speedup S_i, number of nodes n_i, traffic load σ_i and number of off-chip memory access r_i, $1 \leq i \leq m$, $n_i \leq n_{i+1}$

Step 2: Fitting curves $r_i \propto n_i^{\alpha}$ to the number of off-chip memory access, obtain the constants α.

Step 3: Fitting curves $1/(1\text{-}p_1\text{+}p_1/n_i\text{+}\ c_1 \log n_i + \lambda_1 n_i^{\alpha})$ to S_i, $1 \leq i \leq m$, obtain the constants p_1, c_1, λ_1;

Fitting curves $1/(1\text{-}p_2\text{+}p_2/n_i\text{+}\ c_2(n_i\text{-}1) + \lambda_2 n_i^{\alpha})$ to S_i, $1 \leq i \leq m$, obtain the constants p_2, c_2, λ_2;

Fitting curves $1/(1\text{-}p_3\text{+}p_3/n_i\text{+}\ c_3(n_i^2\text{-}1) + \lambda_3 n_i^{\alpha})$ to S_i, $1 \leq i \leq m$, obtain the constants p_3, c_3, λ_3;

Step 4: Figure out speedups of n_i nodes with 3 overhead models:

While $i \le m$ **do** $S_i^1 = 1/(1-p_1+p_1/n_i+ c_1 \log n_i + \lambda_1 n_i^{\alpha})$, $i \leftarrow 1$;
While $i \le m$ **do** $S_i^2 = 1/(1-p_2+p_2/n_i+ c_2(n_i-1) + \lambda_2 n_i^{\alpha})$, $i \leftarrow 1$;
While $i \le m$ **do** $S_i^3 = 1/(1-p_3+p_3/n_i+ c_3(n_i^2-1) + \lambda_3 n_i^{\alpha})$;

Step 5: Figure out the coefficients of determination of speedups with 3 overhead models:

$R_j^2 = \sum (S_i^j - \sum S_i/m) / \sum (S_i - \sum S_i/m)$, $1 \le i \le m$;
Find $j = \arg_{j=1,2,3} \max R_j^2$;
If $j = 1$, **then** $w_{PO} = c_1 \log n$, $w_{mem} = \lambda_1 n^{\alpha}$, $p = p_1$;
If $j = 2$, **then** $w_{PO} = c_2(n-1)$, $w_{mem} = \lambda_2 n^{\alpha}$, $p = p_2$;
If $j = 3$, **then** $w_{PO} = c_3(n^2-1)$, $w_{mem} = \lambda_3 n^{\alpha}$, $p = p_3$;
Obtain the speedup model $S = 1/(1-p + p/n + w_{PO} + w_{mem})$;

Step 6: If input S_{target}, **then**
solve Problem I and return $(n_E, f_E) = \arg_{n,f} \min E$;
If input E_{target}, **then**
solve Problem II and return $(n_S, f_S) = \arg_{n,f} \max S$;

VI. IMPLEMENTATION

In order to verify the analytical model in section 5, we conduct detailed simulation of parallel applications running on a CMP with a scalable NoCs. In this section, we present the architecture, the applications and the simulation platform.

A. Architecture

We assume a homogenous 16-core CMP, each core simulates a Sparc-III-plus processor. Table.1 lists some parameters in configuration, technology parameters refer to ITRS [39]. The cores share an on-chip L2 cache with MESI coherence protocol. The architecture of NoC is shown in Fig.4. The number of cores scales from 1 to 16, which must be power of 2 as GEMS [34] simulator supports. We assume 2 memory controllers in each direction. The frames in Fig.4 show how to shut down idle nodes (both idle cores and routers) when applications are executed on 1, 2, 4, 8 cores.

For the simplicity, we assume global voltage/frequency scaling for the whole chip. The frequency can be scaled from 0 to 4GHz. The supply voltage and threshold voltage are both refer to Orion2.0 LVT parameters at 65nm technology. The voltage can be scaled from 2x threshold voltage to 1.1V.

B. Applications

We use eight parallel applications from PARSEC2.1 [38]. They are *blackscholes*, *bodytrack*, *ferret*, *fluidanimate*, *streamcluster*, *swaptions*, *x264*, and *vips*. Each application skips its initialization part and reserves a little sequential code. We select medium input packages for simulation to ensure a reasonable simulation time for all the applications. In our experiment, we do not change the problem size. Table 2 lists the applications and their features. Parallel overhead (PO) model indicates the growth rate of the fraction of PO, and *c* indicates the coefficient of PO.

Technology Parameters	
Process Technology	65nm
Clock Frequency Scale	0-4GHz
Voltage Scale	0.195V-1.2V
Supply Voltage	1.2V
Threshold Voltage	0.195V
Multi-core Processor	
CMP Size Scale	(1,2,4,8,12,16) cores
L1 Cache	16KB Response latency: 3 cycles Request latency: 2 cycles
L2 Cache	4MB Shared on chip, 16banks, Latency:150 cycles
Memory	8 banks off chip, 4GB

Table 1 Architecture Configuration

C. Simulator and power model

We conduct the CMP on a generally used simulator Simics + GEMS, and evaluate energy of NoC by Orion2.0. Simics runs applications on virtual processor cores, and GEMS models memory, cache and NoCs except disk and processor cores. We also add some statistical sampling code in GEMS to obtain the traffic load of each node.

VII. EXPERIMENTAL ANALYSIS

In section 3, we assume an arbitrary parallelizable application to explore the power-performance tradeoffs, and conclude that Model C is simple and accurate, which is appropriate to be an predictable power model. In section 5, we propose an energy-performance model based on Model C and involve the parallel overhead into performance speedup. By the method in section 5, we establish an energy-performance prediction and optimization mechanism. In this section, firstly, we evaluate our energy-performance model by PARSEC applications; Secondly, we fulfill energy-performance optimization for these applications.

A. Energy Optimization

We choose three performance targets to see whether there exists minimum energy consumption as the number of core scales: (1) A *low* performance target, equivalent to half the fastest possible execution on one core, and normalize all energy measurements to the energy with n=1. (2) A *intermediate* performance target, equivalent to the fastest possible execution on one core, and also normalize all energy

Applications	Domain	Data sharing	Exchange	parallel overhead model	Parallel fraction	*c*	w_{PO} (N=4)	w_{PO} (N=16)	Average traffic load(flits/cycle/port)
blackscholes	Financial analysis	Low	Low	N2	0.977	2.6e-5	0.00042	0.0066	0.02
bodytrack	Computer Vision	High	Medium	N2	0.905	2.3e-5	0.00036	0.0058	0.03
ferret	Similarity search	High	High	N	0.940	0.012	0.048	0.192	0.07
fluidanimate	Animation	Low	Medium	Logn	0.940	0.024	0.048	0.096	0.03
streamcluster	Data mining	Low	Medium	N2	0.951	1.3e-3	0.0208	0.332	0.03
swaptions	Financial analysis	Low	Low	N2	0.989	2.6e-5	0.00041	0.0066	0.02
vips	Media processing	Low	Medium	N2	0.990	1.1e-3	0.0052	0.281	0.06
X264	Media processing	High	High	Logn	0.930	0.05	0.1	0.2	0.07

Table 2: Applications used in the experiments.

measurements to the energy with n=1. (3) A *high* performance target, equivalent to the fastest possible execution on two cores, and normalize all energy measurements to the energy with n=2. The results are shown in Fig.5, the plots from the top to the bottom are corresponding to *low*, *intermediate*, *high* performance targets respectively. It illustrates the energy consumptions of various numbers of cores for each application. Fig.5 shows our proposed Model C+PO keeps well consistency with the simulation results. From observation in Fig.5, model C without parallel overhead can track the simulation energy for applications with low or medium communication such as *blackscholes*, *bodytrack*, *swaptions*, while it cannot track the energy for *ferret*, *fluidanimate*, *x264*, *vips* and *streamcluster*, which have medium to high communication. For those communication-intensive applications, we observe that model C without parallel overhead has an increasing deviation. Model D+PO cannot track the simulation and our model with a reasonable accuracy.

For the applications with low PO at both small and large core count, such as *blackscholes*, *bodytrack* and *swaptions*, the predictions with Model C without PO are partly accurate, because PO does not play a major role in the performance of these applications. Moreover, for the applications with high PO at large core count (*ferret*, *fluidanimate*, *x264*, *streamcluster* and *vips*), the predictions without PO deviate much from those of simulation, while predictions with Model D+PO seem partly accurate because the PO becomes more and more dominant as the core count scales.

We notice that, there exist two deviations: ferret with a low performance target and *X264* with an intermediate performance target. The reason is our model follows a linear relation, which is not as accurate as (3) in some cases.

With a low performance target (top plot in Fig.5), optimal energy is acquired at a small core count. However, with a high performance target (bottom plot in Fig.5), optimal energy is acquired at a large core count. The optimal energy consumption inclines to move to a larger core count as we increase the performance constraints. In Fig.6, the blue lines link the predicted optimal number of cores for different models. The optimal number of cores with Model C+PO is almost the same as simulation for all applications, however the predicted optimal number of cores with Model C and Model D+PO seems not accuracy enough compared to the simulation.

In Fig.6, we plot the energy consumption of real execution (by simulation) with the optimal number of cores and frequency derived from each model. We also compare against the executions on only one core and all the cores. The blue dots indicate the optimal number of core of each model. Model C without PO and Model D+PO not only occupy more cores, but also consume more energy than the simulation and Model C+PO prediction.

B. Performance Optimization

In the second experiment, we assess the performance optimization for each application. Fig.7 shows that Model C+PO also tracks the simulation (ideal) better than other models. And blue dots in Fig.8 shows Model C+PO predicts

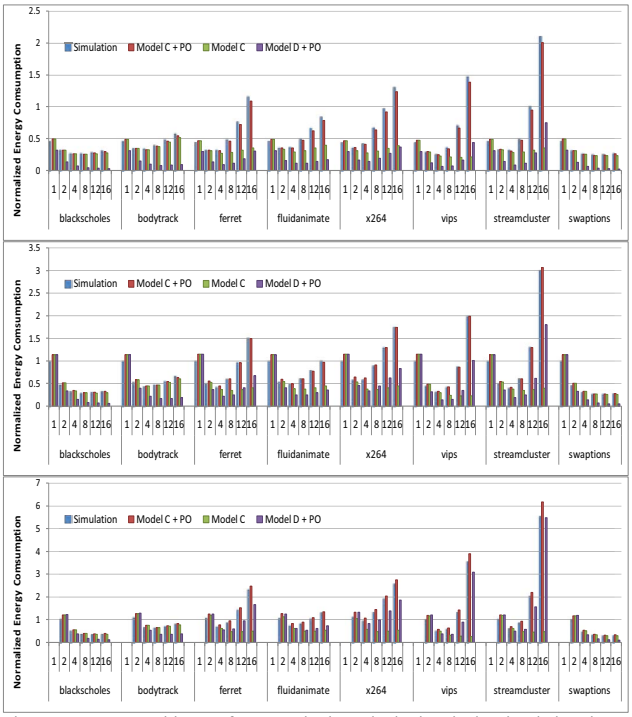

Figure 5. Energy with a performance budget, the budget is the simulation time with n=1. Each value is normalized to the simulation energy with n=1.

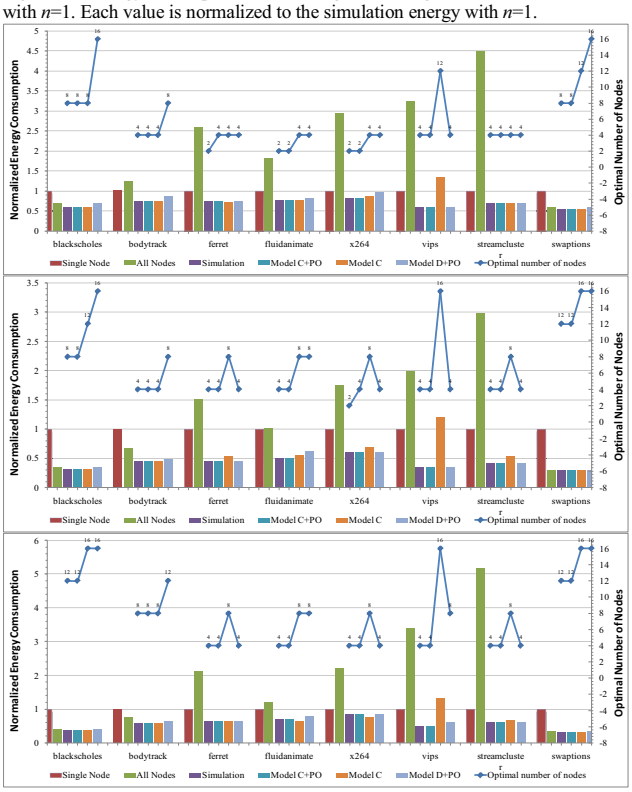

Figure 6. Energy consumption at optimal frequency and core count of each model, the performance budget is the execution time with n=1. Each value is normalized to the simulation energy with n=1.

all optimal core counts accurately with simulation, while Model C and Model D+PO fail in those applications with high PO and large coefficient *c*.

In Fig.8, We compare the optimal performance against simulation and each model and two configurations that execution on only one core and all the cores.

Overall, the result of our evaluation shows, by using Model C+PO,

- We greatly cut down the search space of the optimal pair of core number and frequency levels.
- Our proposed model tracks the simulation (ideal) much better than other models, both at low and high core count with variant constraints.
- For the energy optimization, our optimal solution exploits less cores with less energy consumption. And for the performance optimization, though each model predicts performance close to the simulation, our optimal solution with Model C+PO occupies less cores for execution.

VIII. CONCLUSIONS

In this paper, we have explored energy-performance issues of NoCs. We develop an analytical model to study the energy-performance tradeoffs with considering the frequency-independent factors in both energy and performance models: frequency-independent power in the energy model and frequency-independent off-chip memory access overhead in the performance model. For the first time, we put together the communication overhead, memory access overhead, frequency scaling, core count scaling and frequency-independent power to quantify the performance and energy consumed by NoCs, and utilize both DVFS and DPM technologies to obtain optimal number of cores and frequency.

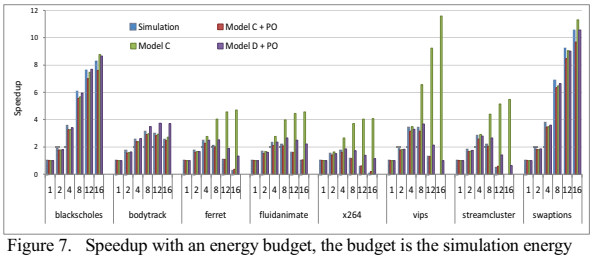

Figure 7. Speedup with an energy budget, the budget is the simulation energy with n=1. Each value is normalized to the simulation time with n=1.

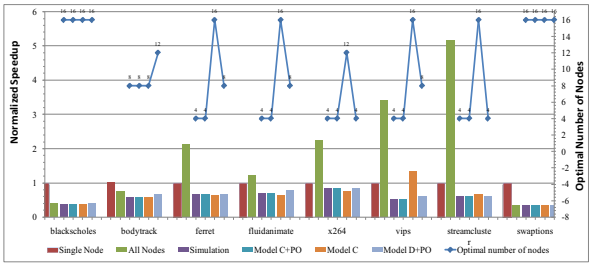

Figure 8. Performance at optimal frequency and core count of each model, the budget is the energy with n=1. Each value is normalized to the speedup with n=1.

Our optimization method avoids falling into a large search space of core number and frequency levels and saving plenty of time on application execution. We also present a method to establish the accurate energy-performance model for a real parallel application running on an NoC step by step.

Finally, we implement eight PARSEC parallel applications to evaluate our model and optimization method. The experiment result confirms that our model predicts NoCs energy and performance behavior well, and we exactly select correct optimal frequency level and core count for most parallel applications.

ACKNOWLEDGMENT

We thank Da Wang, Lin Wang for suggestions and comments. The work in this paper is supported by the National Grant Fundamental Research 973 Program of China (2011CB302501), National Science Fundamental for Distinguished Young Scholars of China (60925009), Foundation for Innovative Research Groups of the National Natural Science Foundation of China (60921009), National Natural Science Foundation of China (61173007, 61100013, 61100015, 61202059, 61161160566, 61020106002).

REFERENCES

[1] Thomas D.Burd and Robert W.Brodersen.“Energy Efficient CMOS Microprocessor Design,” Proceedings of the 28th Annual Hawaii International Conference on System Sciences. Hawaii, USA, 1995, 1:298-305

[2] Yingmin Li, Benjamin Lee, David Brooks , Zhigang Hu, Kevin Skadron. “Impact of Thermal Constraints on Multi-Core Architectures Thermal and Thermomechanical,” Proceedings 10th Intersociety Conference on Phenomena in Electronics Systems(ITHERM ’06). San Diego, CA, 2006: 8-139

[3] Dong Hyuk Woo and Hsien-Hsin S.Lee. “Extending Amdahl’s Law for Energy-Efficient Computing in the Many-Core Era,” Computer, 2008, 41(12):24-31

[4] Sangyeun Cho, Rami G.Melhem. “Corollaries to Amdahl’s Law for Energy.” Computer Architecture Letters, 2008, 7(1):25-28

[5] Hadi Esmaeilzadeh, Emily BlemRenée, Karthikeyan Sankaralingam, Doug Burger. “Dark Silicon and the End of Multicore Scaling,” Proceedings of the ISCA2011, San Jose, CA, 2011:365-376

[6] Jian-Jia Chen, Lothar Thie. “Expected system energy consumption minimization in leakage-aware dvs systems,” Proceeding of the 13th ISLPED 08 (2008)

[7] Dakai Zhu. “Reliability-Aware Dynamic Energy Management in Dependable Embedded Real-Time Systems,” ACM Transactions on Embedded Computing Systems (TECS), December 2010

[8] J. Li and J.F. Martínez. “Power-performance considerations of parallel computing on chip mutliprocessors,” In ACM Trans. on Architecture and Code Optimization, Vol. 2, No. 4, Dec. 2005

[9] Jeong-Gun Lee, Eungu Jung, and Wook Shin. “An Asymptotic Performance/Energy Analysis and Optimization of Multi-core Architectures,” In Proceedings of the 10th International Conference on Distributed Computing and Networking. ICDCN 2009

[10] Omid Azizi, Aqeel Mahesri, Benjamin C. Lee, Sanjay J. Patel, Mark Horowitz. “Energy-Performance Tradeoffs in Processor Architecture and Circuit Design: A Marginal Cost Analysis,” Proceedings of the 37th annual international symposium on Computer architecture. ISCA 2010

[11] Li Shang, Li-Shiuan Peh and Niraj K. Jha. “Dynamic Voltage Scaling with Links for Power Optimization of Interconnection Networks,” In Proceedings of the 9th International Symposium on High-Performance Computer Architecture (HPCA), Anaheim, CA, January 2003

[12] Xuning Chen and Li-Shiuan Peh. “Leakage Power Modeling and Optimization in interconnection network,” In Proceedings of the International Symposium on Low Power and Electronics Design (ISLPED), Seoul, Korea, August 2003

[13] Jian-Jia Chen and Tei-Wei Kuo. “Procrastination Determination for Periodic Real-Time Tasks in Leakage-Aware Dynamic Voltage Scaling Systems,” in IEEE/ACM International Conference on Computer-Aided Design (ICCAD), San Jose, CA, USA, Nov. 5-9, 2007

[14] A B Kahng, Bin Li, Li-Shiuan Peh, Kambiz Samadi. “Orion 2.0: A fast and accurate noc power and area model for early-stage design space exploration,” DATE,2009

[15] R. Kumar, V. Zyuban. “Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads and Scaling,” ISCA 2005

[16] Xiaobo Fan,Carla Schlatter Ellis,Alvin R.Lebeck. “Memory controller policies for DRAM power management,” Proceedings of ISLPED 2001

[17] Jaros, J., Ohlidal, M., Dvorak, V. “Complexity of Collective Communications on NoCs,” In 5th International Conference on Parallel Computing in Electrical Engineering ,PARELEC 2006

[18] Isaac D. Scherson, Peter F. Corbett. “Communications overhead and the expected speedup of multidimensional mesh-connected parallel processors. Journal of Parallel and Distributed Computing 11 (1991) 86-96.

[19] John L. Hennessy, David Patterson. Computer Architecture A Quantitative Approach.Appendix H:Large Scale Multiprocessors and Scientific Applications.

[20] Tilak Agerwala, Siddhartha Chatterjee. “Computer architecture: Challenges and opportunities for the next decade,” IEEE Micro 25, 2005, 25(1): 58-69

[21] Vaclav Dvorak. “Communication Performance of Mesh and Ring-Based NoCs,” In Seventh International Conference on Networking, ICN 2008

[22] M. Barton and G. Withers. “Computing Performance As A Function Of The Speed, Quantity, And Cost Of The Processors,” In Proceedings of Supercomputing 1989

[23] Tommaso Cucinotta. “Optimum Scalability Point for Parallelisable Real-Time Components,” In 32nd IEEE Real-Time Systems Symposium (RTSS 2011), Vienna, Austria, November 29 - December 2, 2011

[24] Sangyeun Cho and. Rami G. Melhem. “On the Interplay of Parallelization Program Performance and Energy Consumption”

[25] R. Ge and K.W. Cameron. “Power-Aware Speedup,” In Proc. IPDPS, 2007

[26] C. K. Chow, “Determination of Cache's Capacity and its Matching Storage Hierarchy,” IEEE Trans. Computers, vol. c-25, 1976

[27] J. Li and J.F. Martínez. “Dynamic power-performance adaptation of parallel computation on chip multiprocessors,” HPCA 2006

[28] Gabriel Loh. “The Cost of Uncore in Throughput-Oriented Many-Core Processors,” In Workshop on ALTA, 2008

[29] Madhavan Manivannan, Ben H. H. Juurlink, Per Stenström. “Implications of Merging Phases on Scalability of Multi-core Architectures,” International Conference on Parallel Processing, ICPP 2011

[30] Kim N.S., Austin T., Blaauw D., Mudge T., Flautner K., Hu J.S., Jane. Irwin M., Kandemir M., Narayanan V. “Leakage current: Moore"s law meets static power,” IEEE Computer, 2003

[31] Ravindra Jejurikar. Cristiano Pereira. Rajesh K. Gupta. “Leakage aware dynamic voltage scaling for real-time embeded systems,” In Proc. Design. Automation Conf. (DAC), pages 275–280, 2004.

[32] R. Ge, X. Feng, and K. W. Cameron, “Modeling and Evaluating Energy-Performance Efficiency of Parallel Processing on Multicore Based Power Aware Systems,” In Proceedings of the 5th Workshop on High-performance, Power-aware Computing (HPPAC), 2009.

[33] Xiaobo Fan, Carla S. Ellis, Alvin R. Lebeck. “The synergy between power-aware memory systems and processor voltage Scaling,” In Workshop on Power-Aware Computing Systems, Dec. 2003

[34] GEMS: http://research.cs.wisc.edu/gems/

[35] B. L. Jacob, P. M. Chen, S. R. Silverman, and T. N. Mudge, “An Analytical Model for Designing Memory Hierarchies,” IEEE Trans. on Computers, vol. 45, no 10, October 1996

[36] N. S. Kim,and T. Mudge. “Total Power-Optimal Pipelining and Parallel Processingunder Process Variations in Nanometer Technology,” ICCAD 05

[37] Jungseob Lee, Nam Sung Kim. “Analyzing Potential Throughput Improvement of Power- and Thermal-Constrained MulticoreProcessors by Exploiting DVFS and PCPG,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Feb. 2012

[38] PARSEC: http://parsec.cs.princeton.edu/

[39] ITRS: http://www.itrs.net/

[40] Trevor Mudge. “Power: a first-class architectural design constraint,” Computer, vol. 34, no. 4, pp. 52-58, April 2001

[41] Srikanth Balasubramanian. “Power delivery for high performance microprocessors,” Proceedings of the 13th international symposium on Low power electronics and design, ISLPED '08