Compile-Time Energy Optimization for Parallel Applications in On-Chip Multiprocessors^{*}

Juan Chen¹, Huizhan Yi¹, Xuejun Yang¹, and Liang Qian²

¹ School of Computer, National University of Defense Technology,
Changsha 410073, P.R.China
{juanchen, huizhanyi, xjyang}@nudt.edu.cn

² Interdisciplinary Research Centre in Materials, University of Birmingham, UK
liangqian99@gmail.com

Abstract. Energy consumption is becoming one of the key optimization objects in on-chip multiprocessor. Minimizing the energy consumption without parallel performance loss is concerned. In this paper, we focus on a DVFS-enabled on-chip multiprocessor architecture, which allows dynamically adjusting each processor's voltage/frequency or shut down unused processors so to obtain energy savings. A detailed analytical model is provided and validated by experiments. Experimental results show energy saving can be up to 10.34% without performance loss.

1 Introduction

Many embedded systems using rechargeable batteries are strongly constrained by power, such as satellite systems, hand-held devices, and solar powered systems. Low-power technology is critical for these systems. In our article, we focus on parallel compilation technique that optimizes the energy consumption of parallel applications.

We adopt a DVFS-enabled (dynamical voltage/frequency scaling-enabled) on-chip multiprocessor architecture to support this low-power parallel compilation technique. Such a DVFS-enabled on-chip multiprocessor architecture meets current high-performance and low-power demands through two main techniques: *on-chip multi-processing* and *dynamical voltage/frequency scaling (DVFS)*. *On-chip multi-processing* exploits the high-level and coarse-grain parallelism in parallel programs.

DVFS is mainly driven by the demand of energy reduction during parallel program execution. Current DVFS-enabled processors include Transmeta's Crusoe [1] and Intel's XScale [2]. They adjust processor voltage/frequency in terms of execution load imbalance. Considering the high-performance and low-power demands, we expect these two techniques (on-chip multi-processing and *DVFS*) to be co-exist in the future architectures.

Considering such an architecture, we explore load imbalance existing in parallel applications to adjust multiprocessors' voltage/frequency. For a given parallel application, there exist serial regions and parallel regions. In the serial execution process, only one of the processors runs and other processors are idle, which causes serious load imbalance.

^{*} This work was supported by the National High Technology Development 863 Program of China under Grant No. 2002AA1Z2101 and No. 2004AA1Z2210.

We can shut down idle processors during the serial executions for energy savings. In parallel execution processes, load imbalance still cannot be fully avoided even if program has been well parallelized. During this parallel region execution process, the parallel execution time depends on the processor with the largest workload and other light-load processors must wait this processor. At this time, reducing these light-load processors frequencies will save energy consumption without performance loss.

We must note that during the transition from one serial region to next parallel region, it takes some extra energy consumption to bring a processor from the power-down state to the active state. And during the transition from one parallel region to next parallel region, each processor's voltage/frequency maybe needs re-adjusting, among which switching overheads must be considered.

We develop an analytical model to make a research on such an energy optimization approach for parallel applications, and then do the experiments to validate our model. Experimental results show energy saving can be up to 10.34% without performance loss.

I. Kadayif also did the related work. In [3], he found loop nests usually need fewer processors than the maximum and his strategy determines each loop nest to use different number of processors. In [4], he provided an integer linear programming method to solve this problem. In his later work [5], a new energy reduction method is provided. Each processor voltage/frequency is set to different level for energy savings. Our previous energy optimization approach [6] combines shutting-down technique and DVFS technique. We exploited the workload imbalance from serial regions and parallel regions and validated our approach through simulation. The major difference between our current work and [6] is that we build a detailed analytical model instead of simulation experiments. And our analytical model avoids the performance loss while the approach in [6] allows some performance loss.

The rest of the paper is organized as follows: Section 2 illustrates our analytical model for energy optimization. Section 3 gives the detailed experimental results and evaluations. Section 4 makes the conclusions.

2 Analytical Model for Energy Optimization

This analytical model is based on a DVFS-enabled on-chip multiprocessor architecture, which consists of eight DVFS-enabled processors. Each processor can change its voltage from 0.9V to 2.05V and change its frequency from 200 MHz to 1 GHz. Each processor has its own L1 cache and L2 cache and they share an off-chip DRAM memory. All the inter-processor communications occur through reading from and writing into this shared off-chip DRAM memory. Multithread parallel applications are executed on this architecture.

Two main energy optimization methods are: *DVFS* and *shutting down*. We allow each processor can continuously adjust its voltage/frequency for energy savings. Switching overheads are taken into accounts carefully. Besides, the unused processor can be placed into a power-down state [7][8]. When the processor/cache is placed into power-down state, the energy consumption is reduced significantly. *Power reduction factor* defines the proportion of the power-down state power dissipation to the active state power dissipation. The re-synchronization penalty is incurred when a processor/cache pair is activated from power-down state. We assume 50-cycle re-synchronization cycle and $10 - \mu J$ energy cost.

Our energy optimization problem is stated as follows: given a parallel application with N_s serial regions, N_p loop regions and a DVFS-enabled on-chip multiprocessor with P processors, find the optimal voltage and frequency modes for each regions, then the whole energy consumption is optimized without performance loss. Model parameters are shown in Table 1, where $1 \leq i \leq N_s$ and $1 \leq j \leq N_p$.

Table 1. Model parameters

Name	Meanings
ts_i	the execution time for the i -th serial region
IC_i^s	the instruction cycles for the i -th serial region (profiling data)
ES_i	the energy consumption for the i -th serial region
t_{syn}	the re-synchronization time penalty (from power-down to active state)
f_{max}	the maximum clock frequency used during the active period
$P_{acti}(f)$	the active power dissipation with variable frequency f
P_{stat}	the power dissipation when the processor is shut down
E_{syn}	the energy consumption for one re-synchronization
$tps_{j,k}$	the execution time of the k -th processor for the j -th parallel region
tp_j	the longest execution time among the P processors for the j -th parallel region
$IC_{j,k}^{ps}$	instruction cycles for the k -th processor during the j -th parallel region (profiling data)
EP_j	the energy consumption for j -th parallel region
$t_{switch}(V_{j-1,k}, V_{j,k})$	the execution time switching cost from voltage $V_{j-1,k}$ to $V_{j,k}$
$f_{j,k}$	the optimal frequency of the k -th processor for the j -th parallel region
$V_{j,k}$	the optimal voltage of the k -th processor for the j -th parallel region
$E_{switch}(V_{j-1,k}, V_{j,k})$	the energy switching cost in switching form voltage $V_{j-1,k}$ to $V_{j,k}$

We use normalized clock frequency f^* and normalized operating voltage V^* instead of f and V , where $f^* = f / f_{max}$, $V^* = V / V_{max}$. Then,

$$V^* = k_1 + k_2 \cdot f^* \quad (1)$$

where the constants $k_1 = V_{th} / V_{max}$ and $k_2 = 1 - k_1$. This simple relationship closely matches industrial data [9]. From today's technology, k_1 is approximately 0.3, and k_2 equals to 0.7.

According to [10], power dissipation can be calculated by¹ formula (2), where A is the switching activity factor, C is the load capacitance, V is the supply voltage and f is the clock frequency.

$$P = ACV^2 f \quad (2)$$

Then the energy consumption consumed by P processors for the N_s serial regions and N_p parallel regions at the maximum frequency is:

$$E_{ori} = \sum_{i=1}^{N_s} (P \cdot ts_i \cdot ACV_{max}^2 f_{max}) + \sum_{j=1}^{N_p} (P \cdot tp_j \cdot ACV_{max}^2 f_{max}) \quad (3)$$

¹ For 130 nm technology, static power dissipation due to leakage current has been ignored.

For all serial regions, we shut down all unused ($P-1$) processors. For all parallel regions, we use the optimal frequency/voltage values ($f_{j,k}$ and $V_{j,k}$) instead of the maximum ones ($f_{\max}$ and $V_{\max}$). After that, the total energy consumption is :

$$E_{opt} = \sum_{i=1}^{N_s} (ts_i \cdot ACV_{\max}^2 f_{\max} + (P-1) \cdot ts_i \cdot P_{stat}) + \sum_{j=1}^{N_p} \sum_{k=1}^P (tp_j \cdot ACV_{j,k}^2 f_{j,k}) \quad (4)$$

where

$$f_{j,k} = f_{j,k}^* \cdot f_{\max} \quad k = 1, \dots, P, j = 1, \dots, N_p \quad (5)$$

$$V_{j,k} = V_{j,k}^* \cdot V_{\max} \quad (6)$$

$$f_{j,k}^* = \frac{tps_{j,k}}{tp_j} \quad (7)$$

$$V_{j,k}^* = k_1 + k_2 \cdot f_{j,k}^* \quad (8)$$

where $ts_i, tps_{j,k}$ is calculated by $IC \cdot CPI \cdot f^{-1}$. So we require profiling data IC_i^s and $IC_{j,k}^{ps}$. Assume β represents power reduction factor, then

$$P_{stat} = \beta \cdot ACV_{\max}^2 \cdot f_{\max} \quad (9)$$

we assume $\beta = 0.1$, $V_{\max} = 2.05V$, $f_{\max} = 1GHz$.

Fig. 1. Illustration for ideal pre-activation strategy for serial regions

In fact, there exist two kinds of switching overhead: one is re-synchronization penalty; the other is voltage/frequency switching overhead. A processor activation call brings a processor from the power-down state to the active state. In common, performance loss will exist as shown in Fig. 1(b), we use an ideal pre-activation strategy to avoid the performance loss as Fig. 1(c) shows.

Although performance loss can be avoided by ideal pre-activation strategy, re-synchronization energy overhead is inevitable. So, the total energy consumption is:

$$E_{opt}^* = \sum_{i=1}^{N_s} (ts_i \cdot ACV_{\max}^2 f_{\max} + (P-1) \cdot ((ts_i - t_{syn}) \cdot P_{stat} + E_{syn})) + \sum_{j=1}^{N_p} \sum_{k=1}^P (tp_j \cdot ACV_{j,k}^2 f_{j,k}) \quad (10)$$

where $E_{syn} = 10 \text{ \& micro; J}$, $t_{syn} = 50 \text{ cycles}$.

Another switching overhead is from DVS, which is illustrated in Fig. 2. Fig. 2(a) illustrates the normal activities of all the processor at the maximum frequency/voltage, where idleness is inevitable due to imbalance. DVFS can eliminate idleness, but switching time yields performance loss as Fig. 2(b) shows. In Fig. 2(c), optimal voltage/frequency for each processor is re-calculated for no performance loss.

Fig. 2. Illustration for voltage/frequency switching overheads

Switching energy and switching time are calculated by:

$$E_{switch}(V_{j-1,k}, V_{j,k}) = (1-u) \cdot c \cdot |V_{j-1,k}^2 - V_{j,k}^2| \quad j = 2, \dots, N_p \quad (11)$$

$$t_{switch}(V_{j-1,k}, V_{j,k}) = \frac{2c}{I_{MAX}} \cdot |V_{j-1,k} - V_{j,k}| \quad j = 2, \dots, N_p \quad (12)$$

Equation (11) and (12) are taken from [11], and are considered to be an accurate modeling of these switching overhead. The variable c is the voltage regulator capacitance and u is the energy-efficiency of the voltage regulator. I_{MAX} is the maximum allowed current. Here we use $c = 10 \mu\text{f}$, $I_{MAX} = 1\text{A}$, $u = 90\%$.

Based on formula (5), (7), we can easily get a new optimal frequency which considering switching time:

$$f_{j,k} = \frac{tps_{j,k}}{tp_j - t_{switch}(V_{j-1,k}, V_{j,k})} \cdot f_{\max} \quad (13)$$

Based on formula (13), we can deduce the optimal frequency $f_{j,k}$ using formula (12), (6), (8) and (5). And $V_{j,k}$ can be calculated easily.

$$f_{j,k} = \begin{cases} \frac{tps_{j,k}}{tp_j} \cdot f_{\max} & j = 1 \\ \frac{tps_{j,k} \cdot f_{\max}}{tp_j - \frac{2c \cdot k_2 \cdot V_{\max}}{I_{MAX}} |f_{j-1,k} - f_{j,k}|} & j = 2, \dots, N_p \end{cases} \quad k = 1, \dots, P \quad (14)$$

Considering voltage/frequency switching overhead, the total energy consumption can be optimized into:

$$E_{opt}^* = \sum_{i=1}^{N_s} (ts_i \cdot ACV_{\max}^2 f_{\max} + (P-1) \cdot ((ts_i - t_{syn}) \cdot P_{stat} + E_{syn})) \\ + \sum_{j=1}^{N_p} \sum_{k=1}^P (tp_j \cdot ACV_{j,k}^2 f_{j,k}) + \sum_{j=1}^{N_p-1} \sum_{k=1}^P (1-u) \cdot c \cdot |V_{j,k}^2 - V_{j+1,k}^2| \quad (15)$$

where $V_{j,k} / f_{j,k}$ is achieved through the previous formulas.

According to the above analyses, we can optimize the energy consumption without performance loss as long as $ts_i > t_{syn}$.

3 Experiments

The whole compiler implementation framework consists of three phases: profiling phase, frequency/voltage scaling phase and optimization phase. In profiling phase, some necessary data are profiled, and then the optimal voltage/frequency for each region is calculated based on the above formulas. In the second phase, some compiler-directed commands are inserted into the codes to shut down the unused processors and adjust voltage/frequency at the right points. In the third phase, processor reuse optimization is applied. We assume the processor assignment for each parallel region is done independently of the other ones. As a result of this, as we move from one parallel region to another one, the same frequency perhaps is set for the different processor, which causes some unnecessary overhead for changing processor. We reuse as many processors as possible to reduce this overhead. For example, we assume frequency level $f_{j,k}$ is set for processor k during the j -th parallel region. If the next adjoined region is also a parallel region and we still need to assign $f_{j,k}$ to one processor, we still use processor k . It is necessary to re-schedule load assignment for each processor. Finally, the processors are reused as much as possible and unnecessary energy consumption is avoided.

We use a representative array-intensive program—sparse matrix-vector product (SMVP)—to validate the effective of our energy optimization approach. SMVP calculates the product of a sparse matrix and a vector. In common, this sparse matrix is big and only a few non-zero elements. Due to the number of non-zero elements in all rows are quite different, load imbalance always occurs [6].

The mp_simplesim simulator [12] combined with energy analytical model is used to validate our energy optimization model. We implemented multithread SMVP

Table 2. Descriptions for the sparse matrices

Sparse Matrix	Size	Descriptions
orsirr1	1030	Oil reservoir simulation for a 21*21*5 irregular grid
orsirr2	886	Oil reservoir simulation for a 21*21*5 irregular grid
sherman2	1080	A black oil, impes simulation, 35*11*13 grid
sherman3	5005	A fully implicit black oil simulator, 16*23*3 grid

Table 3. Profiling cycles for each serial region and parallel region

Sparse Matrix	Cycles for serial region	Cycles for the 1st parallel region / Cycles for the 2nd parallel region							
		P1	P2	P3	P4	P5	P6	P7	P8
orsirr1	75620	531381/ 531680	531743/ 531916	540119/ 540320	531736/ 531993	531295/ 531531	539931/ 539572	531449/ 531580	539697/ 539661
orsirr2	90160	393208/ 393354	393557/ 393492	400844/ 400983	393620/ 393688	393480/ 393688	400383/ 401101	393571/ 393457	400499/ 400526
sherman2	72590	580267/ 583660	588839/ 592406	580181/ 583581	588839/ 592078	580181/ 583140	588839/ 591924	580454/ 583175	588905/ 589542
sherman3	72600	1250209/ 12546222	12543121/ 12547473	12544017/ 12508724	12544815/ 12548586	12547937/ 12547802	12511428/ 12549846	12548979/ 12552057	12546366/ 12551027

Table 4. DVS results for each parallel region

Sparse Matrix	The optimal voltage/frequency during two successive parallel regions								
		P1	P2	P3	P4	P5	P6	P7	P8
orsirr1	1st parallel region	2.0268V/ 983.82MHz	2.0277V/ 984.49MHz	2.05V/1GHz	2.0277V/ 984.48MHz	2.0266V/ 983.66MHz	2.0495V/ 999.65MHz	2.027V/ 983.95MHz	2.0489V/ 999.22MHz
	2nd parallel region	2.0271V/ 984.02MHz	2.0277V/ 984.45MHz	2.05V/1GHz	2.0279V/ 984.59MHz	2.0267V/ 983.74MHz	2.0481V/ 998.67MHz	2.0268V/ 983.83MHz	2.0483V/ 998.8MHz
orsirr2	1st parallel region	2.0227V/ 980.95MHz	2.0239V/ 981.82MHz	2.05V/1GHz	2.0241V/ 981.98MHz	2.0236V/ 981.63MHz	2.0483V/ 998.85MHz	2.024V/ 981.86MHz	2.0488V/ 999.14MHz
	2nd parallel region	2.0223V/ 980.7MHz	2.0229V/ 981.08MHz	2.0496V/ 999.73MHz	2.0235V/ 981.55MHz	2.0235V/ 981.53MHz	2.0501V/ 1GHz	2.0227V/ 981MHz	2.048V/ 998.6MHz
sherman2	1st parallel region	2.029V/ 985.33MHz	2.0498V/ 999.89MHz	2.0287V/ 985.19MHz	2.0498V/ 999.89MHz	2.0287V/ 985.19MHz	2.0498V/ 999.89MHz	2.0294V/ 985.65MHz	2.05V/1GHz
	2nd parallel region	2.0288V/ 985.24MHz	2.05V/1GHz	2.0286V/ 985.11MHz	2.0492V/ 999.47MHz	2.0276V/ 984.4MHz	2.0489V/ 999.22MHz	2.0277V/ 984.47MHz	2.0434V/ 995.39MHz
sherman3	1st parallel region	2.0446V/ 996.26MHz	2.0493V/ 999.53MHz	2.0494V/ 999.6MHz	2.0495V/ 999.67MHz	2.0499V/ 999.92MHz	2.0457V/ 997.01MHz	2.05V/1GHz	2.0497V/ 999.79MHz
	2nd parallel region	2.0493V/ 999.54MHz	2.0495V/ 999.64MHz	2.0451V/ 996.55MHz	2.0496V/ 999.72MHz	2.0495V/ 999.66MHz	2.0498V/ 999.83MHz	2.05V/1GHz	2.0499V/ 999.92MHz

Table 5. Energy saving results

Sparse Matrix	Serial energy before DVFS (J)	Parallel energy before DVFS (J)	Total energy before DVFS (J)	Serial energy after DVFS (J)	Parallel energy after DVFS (J)	Total energy after DVFS (J)	Serial energy savings	Parallel energy savings	Energy savings
orsirr1	0.0047	0.0667	0.0714	0.0011	0.0651	0.0662	76.60%	2.40%	7.28%
orsirr2	0.0056	0.0495	0.0551	0.0013	0.0481	0.0494	76.79%	2.83%	10.34%
sherman2	0.0045	0.0730	0.0775	0.0010	0.0716	0.0726	77.78%	1.92%	6.32%
sherman3	0.0045	1.5502	1.5547	0.0010	1.5471	1.5481	77.78%	0.20%	0.42%

running on mp_simplesim. Four different sparse matrices are used: orsirr1, orsirr2, sherman2, and sherman3 from Harwell-Boeing sparse matrix set [13][14]. Table 2 provides the sparse matrix descriptions. Fig. 4 in [6] gives the non-zero element distributions of each row for these four sparse matrices.

We choose the same SMVP parallel program for four sparse matrices, which consists of one serial region followed by two successive parallel regions. The serial region implements the initialization; sparse matrix is divided into two parts and two parallel regions complete sparse matrix-vector product. Table 3 shows the detailed profiling data for these regions. And Table 4 shows the detailed DVS results.

The detailed energy saving result is showed in Table 5. From statistics results, the total energy saving can be up to 10.34%. For each input matrix set, serial regions provides much more energy saving than parallel regions. That is because SMVP application has been well parallelized, and the opportunity for DVFS is little.

4 Conclusions

We provide a DVFS-enabled on-chip multiprocessor architecture, which allows dynamically adjusting each processor voltage/frequency or shut down some processors to obtain energy savings. Our energy optimization approach adjusts each processor's voltage/frequency in terms of load imbalance. We use pre-activation strategy to effectively avoid performance loss. Besides, voltage/frequency switching costs are also considered. Experimental results show our energy optimization approach is successful in reducing the energy consumption of the parallel applications. The maximum energy saving can be up to 10.34%.

References

- [1] http://www.transmeta.com/crusoe_docs/tm5900_databook_040204.pdf.
- [2] Intel XScale Technology. <http://www.intel.com/design/intelxscale/>.
- [3] I. Kadayif, M. Kandemir and M. Karakoy. An Energy Saving Strategy Based on Adaptive Loop Parallelization. In the *Proceedings of the 39th Design Automation Conference (DAC 2002)*, New Orleans, LA, USA, June 10-14, 2002. p.195-200.
- [4] I. Kadayif, M. Kandemir and U. Sezer. An Integer Linear Programming Based Approach for Parallelizing Applications in On-Chip Multiprocessors. In the *Proceedings of the 39th Design Automation Conference (DAC 2002)*, New Orleans, LA, USA, June 10-14, 2002. p. 703-708.
- [5] I. Kadayif, M. Kandemir, N. Vijaykrishnan, M. J. Irwin and I. Kolcu. Exploiting Processor Workload Heterogeneity for Reducing Energy Consumption in Chip Multiprocessors. In the *Proceedings of DATE'04*. pp.1158-1163. Feb 2004.
- [6] Juan Chen, Yong Dong, Xue-jun Yang, Dan Wu. A Compiler-Directed Energy Saving Strategy for Parallelizing Applications in On-Chip Multiprocessors. In the *Proceedings of Fourth International Symposium on Parallel and Distributed Computing*. France, July 4-6 2005. IEEE Computer Society Press, pp. 147-154.
- [7] A. Chandrakasan, W. J. Bowhill, and F. Fox. Design of High-Performance Microprocessor Circuits. IEEE Press, 2001.
- [8] W. Zhang, N. Vijaykrishna, M. Kandemir, M. J. Irwin, D. Duare, and Y. Tsai. Exploiting VLIW schedule slacks for dynamic and leakage energy reduction. In the *Proceedings of the 34th Annual International Symposium on Microarchitecture*, Austin, TX, December 2001.
- [9] K. Nowka et al. A 0.9V to 1.95V Dynamic Voltage-Scalable and Frequency-Scalable 32b PowerPC Processors. In the *Proceedings of International Solid-State Circuits Conference (ISSCC)*, IEEE Press, 2002, pp. 340-341.
- [10] J. L. Hennessy and D. A. Patterson. Computer Architecture: A Quantitative Approach. Elsevier Science Pte Ltd., third edition, 2003.
- [11] T. Burd and R. Brodersen. Design issues for dynamic voltage scaling. In the *Proceedings of International Symposium on Low Power Electronics and Design (ISLPED-00)*, June 2000.
- [12] Naraig Manjikian. Multiprocessor Enhancements of the SimpleScalar Tool Set. In *ACM Computer Architecture News*, Vol.29 No. 1, March 2001, pp. 8-15.
- [13] I. S. Duff, R. G. Grimes and J. G. Lewis: User's Guide for the Harwell-Boeing Sparse Matrix Collection, Tech. Report TR-PA-92-96, CERFACS, Toulouse Cedex, France, Oct. 1992.
- [14] <http://www.cise.ufl.edu/research/sparse/HBformat/HB/>