PPW with Dark Silicon and Thermal Model

In the coming many-core era, due to the tight power budget, power efficiency is cricital for multicore processor design. In the previous work by Dong Hyuk Woo, evaluation of energy efficiency on the basis of performance and power (PPW) models is developed, which shows the tendency of PPW with number of cores. However, due to the dependency of leakage current on temperature and the prevalence of DVFS and dark silicon in multicore processor, the speedup potential given by existing extending methods of Amdahl's law no longer works out. In this paper, We implement dark silicon and thermal model into the evaluation of PPW to provide an alternative suitable with the dark silicon era.

In this section, we first present the Amdahl's law and performance-per-watt's traditional expression in multi-core processor. Then the static power modeling and thermal modeling is introduced, with which we can write the revised expression of performance-per-watt in multi-core processor.

A. Amdahl's law

Amdahl's law put an upper bound for the speedup that a multi-core processor can achieve by parallelization as:

$$Perf = \frac{1}{(1-f)+f/n} \tag{1}$$

where n is the number of cores, and f is the fraction of a program's execution time that is parallelizable (0¡f¡1). Noted that, (1) is purely theroretical for it doesn't consider any constraints such as power budget.

The average power consumption of the many-core processor is as follows:

$$W = \frac{P_1 \times (1-f) + P_n \times f/n}{(1-f)+f/n} \tag{2}$$

P_1 is the power consumption during the sequential computation process, P_n is the power consumption during the parallel computation process.

$Perf/W$ of a many-core processor is expressed as:

$$\begin{aligned}\frac{Perf}{W} &= \frac{1}{(1-f)+f/n} \times \frac{(1-f)+f/n}{P_1 \times (1-f) + P_n \times f/n} \\ &= \frac{1}{P_1 \times (1-f) + P_n \times f/n}\end{aligned} \tag{3}$$

P_1 is composed of power consumption of 1 active core and $n-1$ idle cores, in previous works, due to lack of consideration of thermal constraints, P_1 is expressed as $1+(n-1)k$, in which k stands for the fraction of power the processor consumes in idle state ($0 \leq k \leq 1$). For parallel computation phase, P_n consists of power consumption of n active cores, which is previously defined as n.

B. Static power modeling

It is widely acknowledged that the total power of a chip is composed of dynamic and static power. The dynamic power is dependent on the activities of the chip, therefore it's easily estimated by methods implemented with performance counter. Yet the static power p_s of the chip is caused by leakage current I_{leak} as:

$$p_s = V_{dd} I_{leak} \tag{4}$$

Due to the non-linear relationship between I_{leak} and temperature, static power is also sensitive to temperature, which makes it hard to obtain.

I_{leak} is composed of many components, including subthreshold current, gate current, reverse-biased junction leakage current, et cetera. Among which, subthreshold current and gate current are the main parts of leakage current, therefore I_{leak} can be approximated as:

$$I_{leak} = I_{sub} + I_{gate} \tag{5}$$

Noted that I_{gate} is cause by tunneling between the gate terminal and the other three terminals, does not depend on temperature and can be considered as a technology-dependent constant. Yet I_{sub} is considered to be highly related to temperature, and can be modeled in the commonly accepeted MOSFET transitor model BSIM 4 as:

$$I_{sub} = K v_T^2 e^{\frac{V_{GS}-V_{th}}{\eta v_T}} (1 - e^{\frac{-V_{DS}}{v_T}}) \approx K v_T^2 e^{\frac{V_{GS}-V_{th}}{\eta v_T}} \tag{6}$$

C. Thermal modeling

To estimate the power consumption of a IC chip, we first divide the chip and its package into multiple blocks called thermal nodes. Then, the thermal resistance and capacitance among these thermal nodes is computed. With above mentioned information, the thermal mdoel for a chip with n total thermal nodes can be generated:

$$\begin{gathered} GT(t) + C\frac{dT(t)}{dt} = BP_{T,t} \\ Y(t) = LT_t \end{gathered} \tag{7}$$

where $T(t) \in \mathbb{R}^n$ is the temperature vector (distinguished from T_p, which is a scalar representing temperature at only one place), representing temperatures at n places of the chip and package; $G \in \mathbb{R}^{n\times n}$ and $C \in \mathbb{R}^{n\times n}$ contain equivalent thermal resistance and capacitance information respectively; $B \in \mathbb{R}^{n\times l}$ stores the information of how powers are injected into the thermal nodes; $P(T,t) \in \mathbb{R}^l$ is the power vector, which contains power consumptions of l components on chip, including both dynamic power vector P_d and static power vector P_s, i.e., $P(T,t) = P_s(T,t) + P_d(t)$, reminding that static power $P_s(T,t)$ is actually a function of temperature T; $Y(t) \in \mathbb{R}^m$ is the output temperature vector, containing only temperatures of thermal nodes that the user is interested in, for example, thermal nodes on the chip only (excluding

This work is supported in part by National Natural Science Foundation of China under grant No. 61404024, in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars, State Education Ministry.

package thermal nodes); $L \in \mathbb{R}^{m\times n}$ is the corresponding output selection matrix which selects the m chip temperatures from $T(t)$.

By applying thermal model, the effect of temperature on cores can be introduced. Through ergodic method, the distribution of light core with the maximum PPW for number of light core from 1 to n can be specified.

I. New Method

This work aims to find the optimal number and distribution of active cores for a fixed set of cores to achieve the maximum performance-per-watt, with consideration of thermal constraints. In order to estimate the power consumption of a multi-core chip for different number of active cores and various active core distributions, we first propose an iteration based full-chip power estimation method, which is accurate but very time-consuming. Furthermore, a non-iteration based method is proposed, which can implement local linearization to avoid time-consuming iterations. Additionally, a greedy based method can be integrated into the non-iteration based power estimation method to achieve further acceleration.

A. Finding maximum PPW by minimizing chip temperatures

As discussed above, the maximum PPW for a multicore chip corresponds to the minimum $P_1 \times (1-f) + P_n \times f/n$, let $P = P_1 \times (1-f) + P_n \times f/n$ to simplify notation. Noted for dark silicon systems, which are extremely temperature limited, we focus on the major problem of thermal limits, and other constraints can be added with minor modification if needed. The task of finding the maximun PPW can be formulated as the following optimization problem

$$\begin{aligned} &\text{minimize } \|P\|_1 \\ &\text{subject to} = \begin{cases} card(P) = n_a, \\ T_c \preceq T_{th}, \end{cases} \end{aligned} \tag{8}$$

where $card(P)$ stands for the cardinality or the size of the vector P, which is defined as the number of nonzero components in P. In our case, $card(P) = n_a$ means there are n_a active cores.

B. Iteration based leakage-aware power estimation

The steady state power estimation can be achieved by estimating the steady state temperature first, which is calculated using model (x) by neglecting the differential term $C\frac{dT(t)}{dt}$, leading to

$$Y(t) = L^T G^{-1} B P \tag{9}$$

Then the steady state power can be deduced with (4), (5), and (6). Because of the dependency of static power on temperature, it's not straightforward to compute the static power and temperature of next steady state based on current static power and temperature. Iteration method can be implemented to solve such problem, the computation flow is shown in Fig.x.

The initial value of $P_s^0(T, t+h)$ is a guess we provide based on the process technology. The temperature distribution $T(t+h)^0$ can be calculated with such guess. $P_s^1(T, t+h)$, the static power of next time step is updated with $T(t+h)^0$. Next, the temperature distribution $T(t+h)^1$ can be derived from $P_s^1(T, t+h)$, which concludes one iteration loop. Such iteration goes on until the convergence test is satisfied as $\| P_s^i(T, t+h)$-$P^{(}i-1)_s(T, t+h) \| < \epsilon$. Finally, the static power and temperature of steady state is outputted.

The iteration based method can produce an accurate outcome providing the ϵ is chosen to be small enough, yet the computing time is a serious problem, especially when the number of cores is large enough.

C. Local linearized thermal model

The major difficulty of calculating leakage-aware power estimation comes from the nonlinear thermal model shown in (x), which is caused by the nonlinear relation between subthreshold current and temperature.

To reduce the long computing time caused by iteration method, the leakage current I_{leak} can be linearized to eliminate the non-linearity between p_s and temperature, thus accelerating the computation.

Taylor expansion is performed on the original I_{leak} model at a expansion point T_0. Thus the linearized relation of I_{leak} and temperature is obtained as:

$$\begin{aligned} I_{sub} =& K(\frac{k}{q})^2 e^{(}\frac{q(V_{GS}-Vth)}{\eta k T_{p0}} \\ &\times (T_{p0}^2 + (2T_{p0} - \frac{q(V_{GS}-V_{th})}{\eta k})(T_p - T_{p0})) \\ &+ o[(T_p - T_{p0})^2], \end{aligned} \tag{10}$$

where $o[(T_p - T_{p0})^2]$ is the remainder. By ignoring the remainder, the linearized I_{sub}, denoted as I_{lin} can be expressed as:

$$\begin{aligned} I_{lin} =& K(\frac{k}{q})^2 e^{(}\frac{q(V_{GS}-Vth)}{\eta k T_{p0}} \\ &\times (T_{p0}^2 + (2T_{p0} - \frac{q(V_{GS}-V_{th})}{\eta k})(T_p - T_{p0})). \end{aligned} \tag{11}$$

Provided the actual temperature value T_p is close to the reference temperature point T_{p0}, the approximation accuracy of I_{lin} can be guaranteed. From previous research, it has been shown that due to the characteristic of today's semiconductor process, such local linear approximation of leakage current has high accuracy around the expansion point.

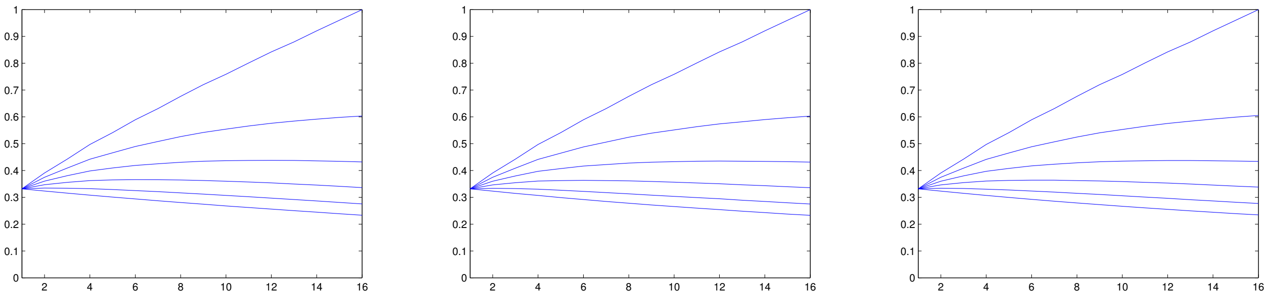
With the linearized relation of subthreshold current and temperature, the relation between static power and temperature in linear form can also be achieved as:

$$\begin{aligned} p_s &= V_{dd} I_{leak} \\ &= V_{dd} \times (I_{lin} + I_{gate}) \\ &= V_{dd} \times (I_{lin}(T_p) + I_{const}) \end{aligned} \tag{12}$$

The linearized static power equation in matrix form is

$$P_s = P_0 + A_s T \tag{13}$$

$$(G - B_c A_s)T(t) + C\frac{dT(t)}{dt} = B_c(P_d(t) + P_0) \tag{14}$$



(a) a (b) b (c) c

Fig. 1. PPW

D. Non-iteration based power estimation

As a property of Taylor expansion approximation, it is accurate only when the actual temperature T_p is close to the expansion point T_{p0}. To find a totally accurate outcome, $T_p = T_{p0}$ is necessary. However, such strategy requires T_{p0} to be updated for each iteration, which would lead to longer computing time than we expected.

We can notice that the temperature of cores can be classified into two parts, one is for active-state cores around $70\,°\mathrm{C}$, one is for idle-state cores around $40\,°\mathrm{C}$. Therefore, the approximation accuracy cannot be guranteed by implementing only one expansion point.

Therefore, in order to balance the accuracy and computing cost, we set two expansion point T_{p1} and T_{p2}, one for active cores and another one for idle state cores.

E. Greedy based acceleration of power estimation

In previous sections, to find the optimal active core distribution which leads to highest performance-per-watt, for a n-core system with different active core numbers, a combinational method with high complexity is implemented. However, it's especially impractical when the number of cores is too big. Therefore, a greedy based method which can find a sub-optimal active core distribution with much less time consumption is applied.

For a n-core system with n_a active cores, the basic idea of finding such sub-optimal solution is described as follows: we first find the optimal solution for only one active core. Next, we fix the first active core position determined by the first step, and find the optimal solution of two cores, with the second active core position determined. Please note that although we say optimal in the second step, such solution is only the optimal solution with the first active core fixed at the position determined by the first step, but not the true optimal solution for general two active cores. Similarly, in the $(i+1)$-th step, we look for the optimal solution for $i+1$ active cores with the positions of i active cores found in all previous steps remain fixed. By proceeding such strategy for n_a steps, the sub-optimal solution for n_a active cores can be achieved.

II. Experimental Results

In this section, we evaluate both accuracy and efficiency of the proposed performance-per-watt estimation technique.

A. Experiment setup

Through HSPICE simulation, the impact of temperature on device leakage can be characterized. With the collected data, we can obtain the parameters of model through curve fitting as shown in Fig.x. The ambient temperature is set to be $20\,°\mathrm{C}$.

For accuracy and speed comparison, we first perform the iteration based power estimation, which is accurate but very time-consuming, therefore we consider it as the golden accuracy baseline (golden for short). Then, two acceleration methods, the taylor expansion based method