A Method for Energy-efficient Optimization on Multi-Cores

Hua Jin, Aixin Wang and Yatao Zhu

***Abstract*—**Power-performance assignment is a popular method for energy-efficient optimization of multi-core processors nowadays, where power-performance models are commonly used to search for optimal configuration in two dimensions: core number and frequency. However, the state-of-the-art methods for searching optimal energy-efficient configurations between core number and frequency suffer from slow convergence speed, tremendous overhead, and poor scalability, which prevents them from practical applications.

In this paper, an efficient search method based on feasible direction method is proposed to quickly reduce search space in core number and frequency, as well as to quickly converge to the minimum point of energy consumption through the iterative process. Moreover, the power-performance model can be flexibly revised by measuring power and performance of each rational configuration. The experimental results show that, compared with Hill-climbing Heuristic which is one of the best existing search methods, our framework makes average elevations in the number of execution times, execution overhead, energy overhead by 38.6%, 43.9% and 46.7%, respectively. The enhancement will be 47.6%, 50.2% and 49.3%, when doubling the cores of a multi-core processor, and 44.7%, 49.1%, 53.2%, when doubling the frequency levels.

***Keywords*—**energy-efficient, optimization, multi-cores, power-performance, feasible direction method, Hill-climbing Heuristic.

I. INTRODUCTION

AS Dennard scaling law [1]-[5] is drawing to an end, the energy efficiency of multi-core processors cannot keep growing or stay unchanged when the die size expands. Therefore, power-performance assignment for energy-efficient optimization has become one of the most important technologies in the future development of multi-core processors. Previous studies on energy-efficient optimization can be classified into two categories. One is that the configuration of core number and frequency can be obtained by calculating the minimum energy consumption in performance-energy model. The results are highly dependent on the precision of the model, which are limited by the current theoretical research findings. The other is that the configuration of core number and frequency can be obtained by gradually searching for the minimum energy consumption through repeatedly amendment of the workloads. The results depend on both the precision of the performance-energy model and the measured results. Compared to the first one, the advantage of the second category is that the demand for precision of the performance-energy model is not critical, where the calculation based on the model may help to reduce the search space and predict the search direction, and the subsequent model computation process can be modified by the measured results. However, the current implementations have the following problems. Firstly, the convergence procedure is quite long because each search for the optimal energy-efficient configuration can only be processed in a single dimension of core number or frequency. Secondly, the search processes for the core number and frequency are ranked by grade results, which are quite consumptions both in energy and in performance overhead. Thirdly, the scalability of the present methods cannot manage situations where a large number of cores and a very high frequency are involved.

To solve the abovementioned problems, we propose an efficient search method for optimal energy-efficient configuration. The method is based on feasible direction method (FDM), which is a classical mathematics method for solving optimal problems of nonlinear program. The objective function and the feasible set can be transformed to a convex, so as to reduce search space in the two dimensions of core number and frequency. It can also converge to the minimum point of energy consumption through iterative processes with a rapid ratio. Compared to Hill-climbing Heuristic, which is one of the best search methods for energy-efficient optimization, our experimental results have demonstrated the following advantages of our approach. Firstly, each search execution can simultaneously reduce the search space for both core number and frequency, which results in a sharply reducing of the number of search execution required to find the optimal energy-efficient configuration. Compared to Hill-climbing Heuristic, the average number of search execution reduced by 38.6%. Secondly, the execution time and energy overhead can be dramatically reduced alongside with the drop of the number of search execution. Compared to Hill-climbing Heuristic, the execution time and energy overhead were reduced by 43.9% and 46.7% respectively. Thirdly, we proved by experiment that our method had a better performance when the number of cores and the frequency level of processor increased. The experimental results showed that, compared with Hill-climbing

This work was supported in part by the Post-Doctoral science foundation of China (No. 2017M620916).

Hua Jin is with the College of Information Science and Technology, Agricultural University of Hebei, Baoding 071001, Hebei, China .

Aixin Wang is with the College of Information Science and Technology, Agricultural University of Hebei, Baoding 071001, Hebei, China

Yatao Zhu is with the Institute of Computing Technology, Chinese Academy Sciences, Beijing 100080, China. (corresponding author; e-mail: zhuyatao@ict.ac.cn).

Heuristic in the number of search executions, execution time and energy overhead, our method made an average reduction of 47.6%, 50.2% and 49.3% respectively when doubling the number of cores; and 44.7%, 49.1%, 53.2% respectively when doubling the frequency levels.

The rest of this paper is organized as follows: section II gives a summary of the current search methods for energy-efficiency optimization and their deficiencies. Section III presents our modeling of power-performance, and section IV further describes detailed solution framework for energy-efficient optimization. The case studies and experimental evaluations are shown in section V. And finally, we have our conclusions and future work in section VI.

II. RELATED WORK

The study of space exploration of core number and frequency configurations for energy-efficiency optimization has been widely studied. To find better a solution for energy-efficiency optimization under constraints of performance, Li and Martinez[6] provided Hill-climbing Heuristic to search the optimal configuration of core number and frequency. For example, in a system with the frequency of L levels and N cores, the searching cost is L•logN.

Curtis-Maury[7] introduced thread concurrency into design space of power-performance[6]. Taking into account technological differences of inter-cores and sharing resources of threads, the design space of power-performance was expanded to L•2N. This work fitted the formula of events rate by sampling, which could obtain the frequency level of optimal energy-efficiency. Nevertheless, binary search method was used for searching in the dimension of core number, which would result in lower search efficiency.

Moreover, Wang [8] expanded the design space of power-performance to heterogeneous multi-cores platforms. When CPUs and GPUs shared the same chip, the optimal configuration could be found by searching the space of core number and frequency of both the CPUs and the GPUs respectively, which maximized the throughput under the constraint of overall energy consumption. However, the exhaustive method taken for searching the core number and frequency of CPUs and GPUs had high overheads in practical implementation, and the corresponding searching cost was L•N.

Lee[9] and Lee[10] further studied the three-dimensional design space composed of core size, core number and frequency. PCPG (Per-Core Power-Gate) was utilized to adjust the number of cores and frequency of each core. Effects of homogeneous core size[9], heterogeneous core size[10], core number and frequency level on performance were analyzed under constraints of power consumption. The core number and frequency level for optimum performance were obtained through search execution, where the system provided a range of available frequency levels for each level of core number. However, the frequency level, which satisfied the constraint of power consumption and optimal core number, was selected through exhaustive method. Thus, the maximum number of search execution was L•N, which would surely brought higher overheads.

Besides the execution overhead, workloads also play a crucial role in accuracy of the power-performance models. Bogdan and Marculescu[34] presented a workload characterization and showed its impact on multicore platform design. They proposed a mean field approach to analyze the traffic dynamics in multicore systems and showed how the non-stationary effects of the NoC workload could be effectively captured, which was of fundamental significance for re-thinking the very basis of multicore systems design. Qian et al.[34] explored the performance evaluation of NoC-based multicore systems, which covered from traffic analysis to NoC latency modeling.

In summary, the current search methods for energy-efficient optimization are exhaustive methods or branch methods. Although different prediction methods were utilized to search for frequency level, workload procedure has to be run more than once to approach power-performance target at each level of core number. The purpose of the replacement methods is to decrease the search space of core number or frequency level, and consequently reduce the search costs [31]-[34]. In this paper, we propose a search method based on a feasible direction method, which can quickly reduce search space in the two dimensions of core counts and frequency simultaneously. It is guaranteed that the method converges to the minimum point of energy consumption in the iterative process, and exhaustive method will be used to revise search results in small range near the optimal solutions. Thus, our method not only reduces the number of search executions, but also decreases time cost and energy overhead during the searching process for the optimal configuration of the core number and frequency.

III. POWER-PERFORMANCE MODEL

To search for minimum energy consumption of multi-cores, we built a mathematical model of power performance.
According to [11-20], energy consumption P of on-chip devices in processor is defined as:

$$P = ACV^2 f + VI_{leak,V_0,T_0} e^{\phi(V,T)} \tag{1}$$

Where dynamic power is ACV^2f, static power is $VI_{leak,V0,T0}e^{\Phi(V,T)}$, leakage current is $I_{leak,V0,T0}e^{\Phi(V,T)}$, V_0 and T_0 is the normalized maximum power supply voltage and temperature. Function $\Phi(V,T)$ is a linear combination of voltage V and temperature T, which can be expressed as $a_1+a_2V+a_3T+a_4VT$. a_1, a_2, a_3 and a_4 can be obtained by curve fitting. According to (1), energy consumption of n-core processor can be expressed below:

$$P(n, f) = n(ACV^2 f + VI_{leak,V_0,T_0} e^{\phi(V,T)}) \tag{2}$$

The relation of voltage and frequency in this paper is defined as[11]-[13]: $V = \beta_1+\beta_2 f$, where β_1 is the value of minimum voltage (V_{min}) and β_2 is defined as (V_{max} - V_{min})/ f_{max}, where V_{max} is the value of maximum voltage and f_{max} is the value of maximum frequency.

Let the serial execution time of a parallel program on a single core be T_s, and its execution time on an n-core processor be defined as:

$$T(n,\ f) = T_s\left[\frac{F}{f}(1-p+\frac{p}{n}+w_P)+w_M\right] \tag{3}$$

Where p denotes the ratio of parallel part of program in total workload, w_p is on-chip parallel overhead, w_M is off-chip memory overhead. F denotes the reference frequency, which is the maximum frequency of processor in this paper. Therefore, energy consumption of the program executed on the n-core processor can be defined as: $E(n,f) = P(n,f)T(n,f)$. Finally, energy-efficient optimization with the performance constraint of execution time Tt can be expressed as follows:

$$\begin{aligned} &Min\ E(n,\ f) \\ &s.t. T - T_t \le 0, 0 < f < f_{\max}, 1 \le n < N \end{aligned} \tag{4}$$

Where N is the maximum number of cores available in multi-core processors.

According to (1) and (3), execution time is a nonlinear constraint, and energy consumption and execution time both have first derivative. Thus the above expression (4) of energy-efficient optimization can be converted to a linear programming by feasible direction method[21]. As long as the choice of initial core number and frequency is feasible, the lowest energy consumption and its configuration of core number and frequency can be ultimately determined through finite number of iterative executions.

IV. ENERGY-EFFICIENT OPTIMIZATION BASED ON FEASIBLE DIRECTION METHOD

The key idea of our method is the application of feasible direction method to search for minimum energy consumption in multi-cores. In Figure 1, we take a parallel program with a parallel load ratio of p (p=0.99) as an example, which performs all energy consumption values at frequencies range [0, 4 GHz] on n cores, where n is an integer and $n \in [1, 16]$, and the execution time at the highest frequency of the single core is the performance constraint.

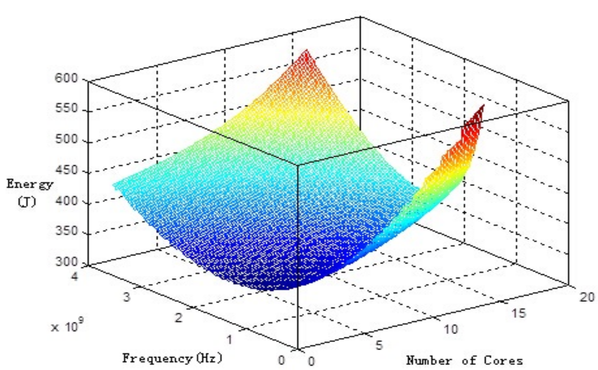


Fig. 1. Distribution of Energy-efficiency in Core-frequency

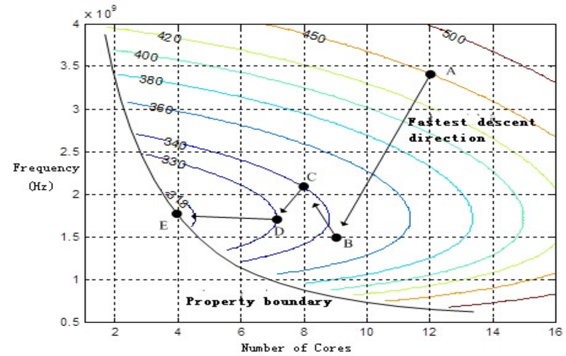


Fig. 2. Distribution of Contours of Energy-efficiency in Core-frequency

As shown in Figure 1, it can be seen that the energy consumption in the following three regions is high: high frequency, high number of cores, and high number of frequency-cores. Thus, any search method for the program will make the frequency-cores number moving to a low cores number and low frequency area. However, the lower the number of cores and frequencies, the more possibilities the energy consumption may not work. We can see that in the low cores number - low frequency region, only part of the cores number and frequency are shown in the figure. The reason is that the cores number-frequency configuration outside of this boundary cannot satisfy the performance constraints. In addition, we can see that the darker color represents the lower energy consumption, where the lowest energy consumption area is near the performance constraints. In order to facilitate the observation of energy consumption declining in the cores number-frequency space, we take contour analysis of the curve surface in Figure 1, as shown in Figure 2.

In the case of Figure 2, it is possible to perform the A-B-C-D-E in the direction from the initial point A through the feasible direction search. We ignore the energy consumption contours which are not marked in the Fig.2. The search process can be summarized as:

(1) From point A (n = 12, f = 3.4GHz) , starting along the fastest decline in the energy consumption-performance model to search for integer number of cores and frequency, find the performance of cores number from 8 to 12 in the direction of fastest decline within the bounds of the performance constraints, where point B (n = 9, f = 1.5 GHz) is the lowest point in the direction of energy consumption.

(2) From point B (n = 9, f = 1.5GHz), obtain its power consumption and performance, calculate the fastest decline in the direction starting from B, find the performance of cores number from 1 to 9 in the direction within the bounds of the performance constraints, where point C (n = 8, f = 2.1 GHz) is the lowest point in the direction of energy consumption.

(3) From point C (n = 8, f = 2.1GHz), obtain its power consumption and performance, calculate the fastest decline in the direction starting form C, find the performance of cores number from 6 to 8 in the direction within the bounds of the performance constraints, where point D (n = 7, f = 1.7 GHz) is the lowest point in the direction of energy consumption.

(4) From point D (n = 7, f = 1.7GHz), obtain its power consumption and performance, calculate the fastest decline in the direction starting from the D point, find the performance of cores number from 4 to 7 in the direction within the bounds of the performance constraints, where point E (n = 4, f = 1.8 GHz) is the lowest point in the direction of energy consumption.

Detailed steps of energy-efficiency optimization based on feasible direction method are presented as follows.

A. Feasible Descent Direction of Core Number and Frequency

Choose the feasible initial core number and frequency (n_k, f_k) to satisfy the performance constraint. Given the execution time $T_k(n_k, f_k)$ and power consumption $P_k(n_k, f_k)$, a feasible descent direction of core number and frequency can be calculated, which must meet the following conditions:

$$\begin{cases}\nabla E_k^T \mathbf{d}_k < 0 \\ \nabla (T-T_t)_k^T \mathbf{d}_k \le 0\end{cases} \Rightarrow \begin{cases}\left(T_k P_{n_k}{}' + P_k T_{n_k}{}'\right) d_{n_k} + \left(T_k P_{f_k}{}' + P_k T_{f_k}{}'\right) d_{f_k} < 0 \\ T_{n_k}{}' d_{n_k} + T_{f_k}{}' d_{f_k} \le 0\end{cases} \tag{5}$$

Where gradient components of power consumption and execution time can be calculated as follows:

$$\begin{cases} P_{f_k}{}' = \dfrac{\beta_2 P_k}{\beta_1 + \beta_2 f} + nAC(\beta_1 + \beta_2 f)(\beta_1 + 2\beta_2 f) \\ P_{n_k}{}' = \dfrac{P_k}{n} \\ T_{f_k}{}' = -\dfrac{T_k / T_s - w_M(n_k)}{f} \\ T_{n_k}{}' = T_s \left[\dfrac{1}{n}\left(\dfrac{2F}{f} w_P(n_k) - \dfrac{1}{2} w_M(n_k) \right) - \dfrac{1}{n(n-1)} \left(\dfrac{F}{f}(1 + w_P(n_k)) + w_M(n_k) - T_k \right) \right] \end{cases} \tag{6}$$

B. Selection of Feasible Direction

Construct the following linear programming problem to select feasible direction d_k , which satisfies the condition (5) by linear programming approach.

$$\begin{aligned} & Min\ z \\ & s.t. \begin{cases} \left(T_k P_{n_k}{}' + P_k T_{n_k}{}'\right) d_{n_k} + \left(T_k P_{f_k}{}' + P_k T_{f_k}{}'\right) d_{f_k} - z < 0 \\ T_{n_k}{}' d_{n_k} + T_{f_k}{}' d_{f_k} - z \le T_t - T_k \\ -1 \le d_{n_k} \le 1 \\ -1 \le d_{f_k} \le 1 \end{cases} \end{aligned} \tag{7}$$

Therefore, nonlinear programming problem (4) can be converted into a linear programming problem (7), where the Topkis-Veinott revising method [11] is used for the second constraint condition. The revision ensures that the optimal solution z_k of linear programming through gradually iteration will eventually converge to zero. If z_k of k-th iteration turns into zero, it implies that components of descent direction of (7) are minimal and (n_k, f_k) is very close to the optimal solution. Thus, given ε being the infinitesimal value more than zero, when $| z_k | \le \varepsilon$, the iteration should be stopped and (n_k,f_k) should be considered as an approximate optimal solution to (4).

C. Model Computation for Energy-efficient Optimization

If the optimal solution to (7) does not meet the formula $| z_k | \le \varepsilon$, the above optimal solution d_k is the feasible descent direction of the (k+1)-th iteration. The input of the (k+1)-th search in space of core number and frequency is shown below, where a is the feasible descent distance.

$$n_{k+1} = n_k + a d_{n_k} \qquad f_{k+1} = f_k + a d_{f_k} \tag{8}$$

(n_{k+1}, f_{k+1}) locates in the feasible direction, where the starting point is (n_k, f_k). The feasible descent distance a can be obtained by one-dimension search method.

Because core number and frequency of multi-core processors are discrete and the optimal configuration of them cannot be obtained by calculating the extremum of equations of objective function, searching for minimum energy consumption in the space of core number and frequency is inevitable. Besides, core number must be integer values and frequency must be at some voltage-frequency level. Supposing the largest number of cores in a multi-core processor is N and frequency between 0 and f_{max} is divided into L levels, the interval of frequency level is $\Delta L = f_{max}/L$. Because the step size of search might not be obtained to meet integer values of core number and frequency level simultaneously, core number and frequency are taken as main directions to find search step size a respectively. One is that the step size based on core number can satisfy the core number changing in terms of integer values, and the value of its lowest frequency level will be selected. The other is that the step size based on frequency can satisfy the frequency level changing interval, and the value of its lowest core number will be selected. Detailed process is presented as follows.

1) Take core number as the main search direction

Let step size of the i-th search in feasible direction d_k be: $a_{n,i} = i/|d_{nk}|$, where i = 1, 2, 3…m. Thus core number of the i-th search is $n_k + a_{n,i} \bullet d_{nk}$ and frequency is selected as $f_k + a_{n,i} \bullet d_{fk}$. The step size ensures the integer value of core number. In order to guarantee searching along the feasible direction, the selection of integer value of frequency level is ignored temporarily, although it must meet the constraints as follows.

$$\begin{cases} T_t - T(n_k + a_{n,i} d_{n_k},\ f_k + a_{n,i} d_{f_k}) \ge 0 \\ 1 \le n_k + a_{n,i} d_{n_k} \le N \\ 0 < f_k + a_{n,i} d_{f_k} \le f_{\max} \end{cases} \tag{9}$$

If step size of the i-th search meets condition (8), energy consumption $E(a_{n,i})$ can be calculated as $E(n_k+a_{n,i} \bullet d_{nk}, f_k+a_{n,i} \bullet d_{fk})$ and the (i+1)-th search will continue. Otherwise, the search will stop. Setting $m = i$ -1, the step size an of lowest energy consumption searched along feasible directions of core number is $a_n = \arg\min_{1 \le i \le m} E(a_{n,i})$.

However, the selection of frequency, $f_k + a_{n,i} \bullet d_{fk}$, is not always an integer value, which means that it is also necessary to revise the frequency. As shown in Figure 3, current feasible direction is AB and C is the point of the lowest energy consumption. The choice of frequency level with an integer value might be the nearest point D of lower frequency level or the nearest point E of higher frequency level. When core number is determined, decreasing or increasing frequency may obtain the lower energy consumption. Thus energy consumption and performance of D and E need to be calculated separately. The point, which satisfies the performance constraints and has relatively lower energy consumption, is selected as the approximate feasible point of feasible direction for the (k+1)-th search. Subsequently the feasible direction will be revised to AE or AD.

2) Take frequency as the main search direction

Let step size of the i-th search in feasible direction d_k be: $a_{f,i} = i\Delta L/|d_{fk}|$, where i = 1, 2, 3…q. Thus, the frequency of the i-th search is $f_k + a_{f,i} \bullet d_{fk}$ and the core number is selected as $n_k + a_{f,i} \bullet d_{nk}$. The step size ensures the integer value of frequency level. In order to guarantee searching along the feasible direction, the selection of integer value of core number is ignored temporarily, though it must satisfy the constraints as follows.

$$\begin{cases} T_t - T(n_k + a_{f,i} d_{n_k}, f_k + a_{f,i} d_{f_k}) \geq 0 \\ 1 \leq n_k + a_{f,i} d_{n_k} \leq N \\ 0 < f_k + a_{f,i} d_{f_k} \leq f_{\max} \end{cases} \tag{10}$$

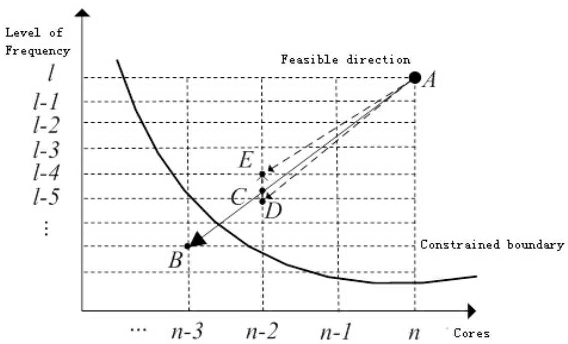


Fig. 3. Core number as Search Direction to Determine the Step Size

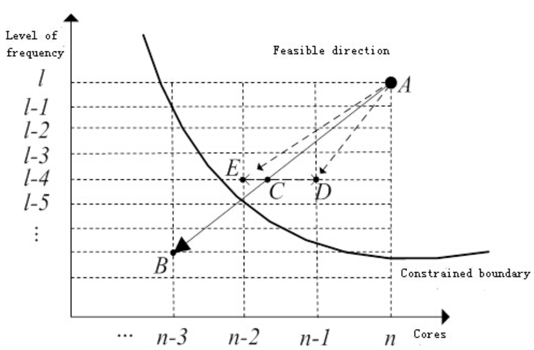


Fig. 4. Frequency as Search Direction to Determine the Step Size

If step size of the i-th search meets condition (9), energy consumption $E(a_{f,i})$ can be calculated as $E(n_k+a_{f,i} \bullet d_{nk}, f_k+a_{f,i} \bullet d_{fk})$ and the (i+1)-th search will continue. Otherwise, the search will stop. Setting q = i-1, the step size a_f of lowest energy consumption searched along feasible directions of frequency level is $a_f = \arg\min_{1 \leq i \leq q} E(a_{f,i})$. Nevertheless, the selection of core number, $n_k+a_{f,i} \bullet d_{nk}$, is not always an integer value, hence it is necessary to revise the core number. As shown in Figure 4, current feasible direction is AB and C is the point of the lowest energy consumption. The choice of core number might be the nearest point D of the larger core number or the nearest point E of the smaller core number. When frequency level is determined, decreasing core number may obtain lower energy consumption and increasing one may obtain higher energy consumption. Thus, energy consumption and performance of D and E need to be calculated separately. If point E meets the performance constraints, it will be selected as the approximate feasible point of feasible direction for the (k+1)-th searching. Otherwise, the point D will be selected. Subsequently the feasible direction will be revised to AE or AD.

Finally, the step size a can be selected as an or a_f, which provides the lower energy consumption in the above two kinds of search process.

$$a = \begin{cases} a_n, & E(a_n) < E(a_f) \\ a_f, & E(a_n) \geq E(a_f) \end{cases} \tag{11}$$

In the following experiments of this paper, the maximum number of cores is set to 16, which is far less than the number of available frequency levels. Therefore, in order to obtain fine-grained step size of searching along feasible directions, the core number will be selected as the main search direction.

D.Revision of Model Computation

If the configuration of core number and frequency level, which satisfies the performance constraint, cannot be obtained through searching along the feasible direction in the last iteration, the core number and frequency in the last execution is regarded as the optimal solution. However, due to the space of core number and frequency is discrete, some deviation between the optimal solution from model computation and the practical optimal solution may exist. Thus, it is necessary to revise the core number and frequency used in the last execution. Taking the search direction of core number as an example, the revising process is shown in Figure 5.

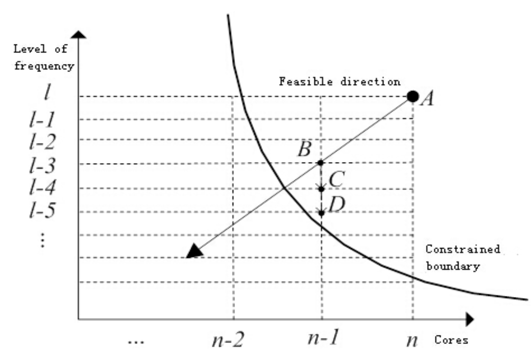


Fig. 5. Core Number as Search Direction to Revise the Search Process for Lowest Energy Consumption

Along the feasible direction of point B, which is obtained in the last execution, there is no integer value of core number which can meet the performance constraint. Thus, the process of searching for minimum energy consumption will be terminated. However, the energy consumption of point B is not the lowest one meeting the performance constraint. If increasing frequency level from point B leads to higher performance, energy consumption may increase or decrease. Because point B is very close to the minimum point of energy consumption, and an obtuse angle occurs between the increasing direction of frequency level and the feasible descent direction of point B, which means the increase of frequency level deviates from the descent direction of energy consumption, lower frequency level will be selected to search for energy consumption points which is lower than point B.

The selection of frequency level depends on the feasible direction of final execution point. If its direction of frequency level d_{fk} is greater than zero, it is necessary to revise the search results along the direction of frequency level increase to obtain the lower energy consumption point. If d_{fk} <=0, it is necessary to revise the search results along the direction of frequency level decreasing to obtain the lower energy consumption point. As shown in figure 3, the direction of frequency level of point B is negative (d_{fk} <0). Thus it is necessary to search for the core number and frequency along the direction of frequency level decrease until it cannot meet the performance constraint. Then the two revised points C and D are obtained. Finally the revised point with the lowest energy consumption can be regarded as the optimal energy-efficiency point.

V. EXPERIMENTS

A. Experimental Platform

The architecture of the multi-core processor for this experiment is constructed with Simics and GEMS[22]-[26] simulators. Simics provides virtual cores and operating system to execute test cases and GEMS simulates components such as memory, cache and NoC (Network on Chip). Orion2.0 is used to assess energy consumption. A homogeneous 16-core processor is built based on the simulation platform. Each core simulates a Sparc-III-plus processor. The NoC is a two-dimensional mesh, which consists of 16 routers connected together and each router connects to a core.

B. Evaluation benchmarks

In our experiments, evaluation benchmarks are selected from PARSEC2.1[27]-[30], which includes 12 programs from different areas, such as computer vision, video encoding, financial analytic, animation physics and image processing. In the experiments, the initialization of each test program is skipped and a little part of serial codes and total parallel codes of each program are retained. Medium packages are selected as the input to ensure that every test program has sufficient simulation time to sample data and calculate energy consumption and performance[15].

C. Evaluation of Performance and Scalability

a) Performance

Figure 6 shows the flow chart of EOFDM from the initialization to the first execution, the initial point is obtained by using the method in [24].

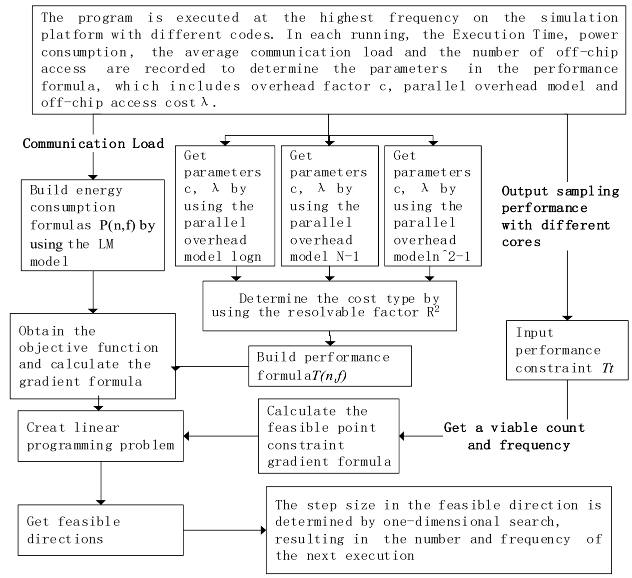


Fig. 6. Flow Chart of EOFDM

Firstly, the serial execution time of each test program on a single core with the highest frequency is recorded as performance criteria, and the target performance of each program on multi-core platform is set to half of the serial execution time. In order to obtain the performance parameters, the test programs are implemented on multi-core platform at the highest frequency with 2, 8, 16 cores respectively.

To explore a wider range of space of core number and frequency, the frequency, which ranges from 100MHz to 4GHz, is simulated in the experiments, and 40 levels of frequency are provided with step size of 100MHz. Due to the constraint of the target performance, some lower levels of frequency are not tested. As the comparison experiments, Climbing Heuristic [5] and Binary Hill-Climbing [5],[8] are also adopted to search for optimal configuration of core number and frequency. We take 8 cores and 4GHz as the starting point to search for the optimal configuration of core number and frequency in the experiments. In addition, sampling overhead and revising overhead are added to our method. Considering all of the above factors, the execution times of our method can be calculated as below:

Execution Times of Our Method = Sampling Times + Searching Times+ Revising Times.

Table 1 lists the iterative process of each program by using EOFDM. The colorless units in table 1 show the beginning of search at the initial feasible point in each program. The iterative process will not stop until available steps could not be found. The deep color units in the table 1 show the optimal number of core and frequency obtained by using a modified search process after the iteration stops. The number of core and frequency with underline is the approximate optimal solution obtained by using our method. Because the running time of each iteration point in table 1 is very large, the time and the energy consumption of the program at the highest frequency are normalized. In addition, the components of the feasible direction in the two directions of the core number and frequency are listed in table 1. In every column, the lowest energy consumption, number of cores and frequency level are labeled in each program through exhaustive method, which is considered the exact optimal solution in this experiment.

Analyses of the search process of the 6 programs can be obtained are:

1) Except Blackscholes, all programs can use EOFDM to find the minimum energy consumption of the core number and frequency;

2) The revised search increases a lot of execution time and energy consumption, and the search accuracy of EOFDM directly affects the number of modified search. That is to say, the larger the EOFDM search results deviate from the optimal solution, the greater the search execution overhead will be;

3) As the number of cores in every search gradually approaches the optimal solution, the frequency that appears in the optimal solution may hop up and down, because in the decrease direction of energy consumption, the frequency decreases quicker than the number of cores. The fastest decline in energy consumption mainly points to the direction of the decline of number of cores;

4) Each search process is accompanied by a reduction of energy consumption, but the performance may be reduced slightly under the premise of meeting the constraints. The overall trend is close to the optimal solution with the energy consumption close to the performance constraints.

Figure 7 shows the comparison of execution times of our method, Hill-Climbing Heuristic and Binary Hill-Climbing. Since the execution times of Binary Hill-Climbing are much more than that of the other two search methods, it is meaningless to take Binary Hill-Climbing as a target comparison method. We therefore remove the data of Binary Hill-Climbing from our experiment results.

The comparisons of our method and Hill-Climbing Heuristic about execution overhead and energy overhead are shown in figure 8 and figure 9. Compared to Hill-Climbing Heuristic in the execution times, execution overhead and energy overhead, our method makes an average reduction by 38.6%, 43.9% and 46.7% respectively. It can be seen that, when searching along feasible direction, we quickly reduce search space in the two dimensions of core number and frequency simultaneously. Our method has outstanding advantages in execution overhead and energy overhead, when compared to Hill-Climbing Heuristic.

b) Scalability

To validate the scalability of our method, the number of cores increases from 16 to 32, and the frequency decreases from 100MHZ to 50MHZ in the comparison experiments. Target performance of each test program implemented on multi-core platform is set to 2 times the performance of single core with highest frequency. We took 16-core, 4GHz as a starting point to search the optimal configuration of core number and frequency. Execution times, execution overhead and energy consumption are measured to compare the scalability of our method and Hill-Climbing Heuristic. Table 2 lists the iterative process of double core number and double frequency of each program by using EOFDM and Hill-climbing Heuristic respectively. The comparison experiments with different cores and frequency levels are respectively shown as figures 10, 11 and 12, where Nx2 represents doubling cores of a multi-core processor and Fx2 represents doubling frequency levels.

From figures 10, 11 and 12, compared to Hill-Climbing Heuristic in execution times, it can be seen that when using 16-core platform and frequency interval of 100MHz, execution overhead and energy consumption, our method makes an average reduction by 38.6%, 43.9% and 46.7% respectively; 47.6%, 50.2% and 49.3% when doubling cores of a multi-core processor, and 44.7%, 49.1%, 53.2% when doubling the frequency levels. Moreover, when doubling the cores of a multi-core processor and frequency levels, performance of our method just slightly downgrades, while energy consumption of our method doesn't significantly increase. Therefore, it can be concluded that our method has a certain degree of scalability.

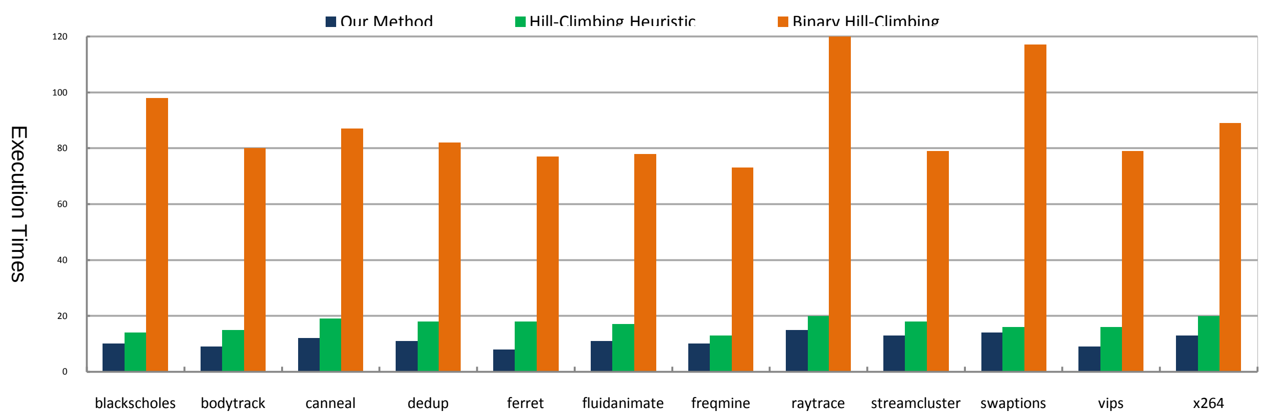


Fig. 7. Execution Times Comparison of Our Method, Hill-Climbing Heuristic and Binary Hill-Climbing

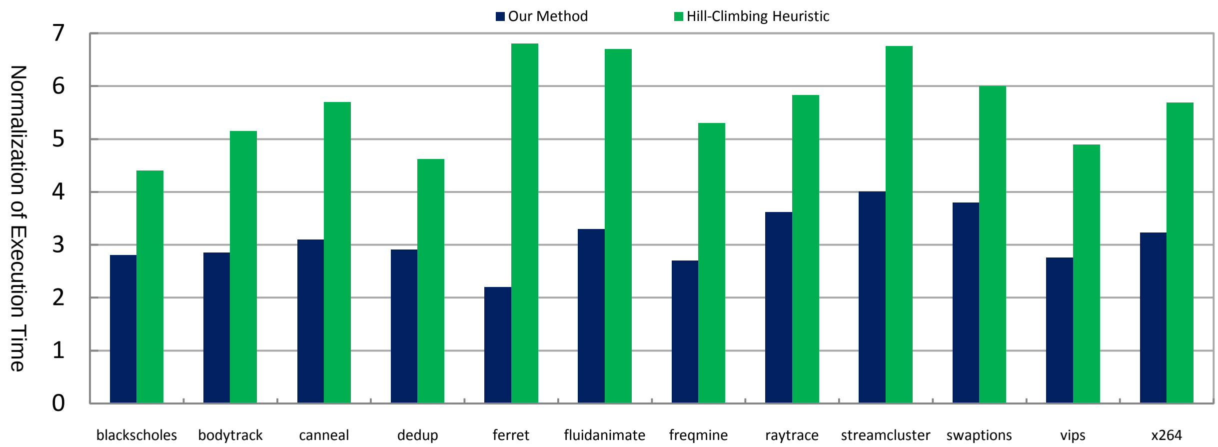


Fig. 8. Execution Overhead Comparison of Our Method and Hill-Climbing Heuristic

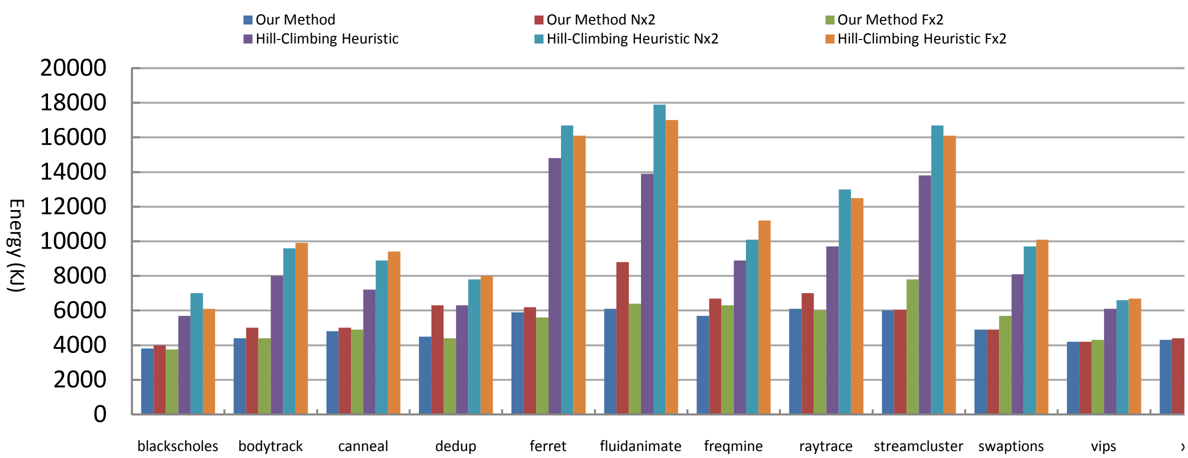
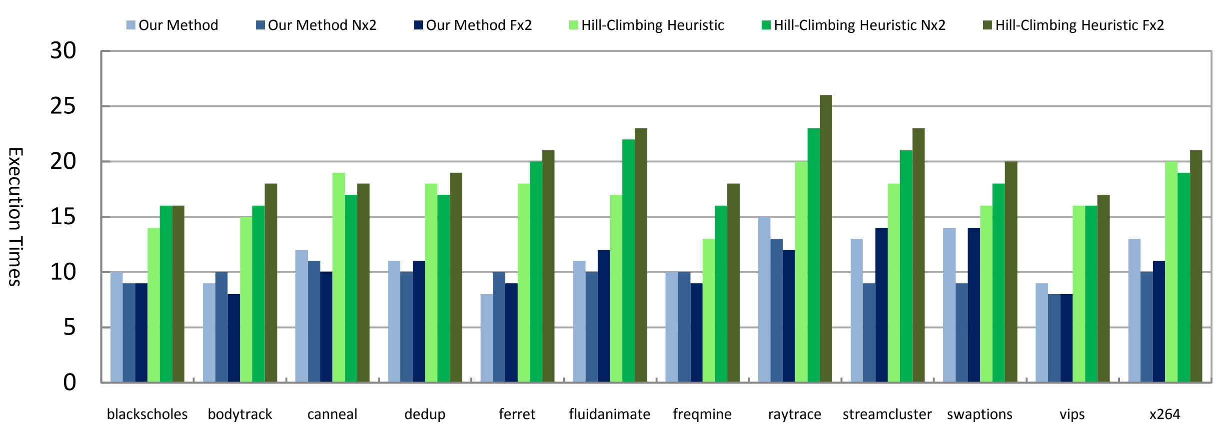


Fig. 9. Energy Overhead Comparison of Our Method and Hill-Climbing Heuristic



- **Fig. 10.** Execution Times Comparison of Our Method and Hill-Climbing Heuristic with Different Cores and Frequency Levels

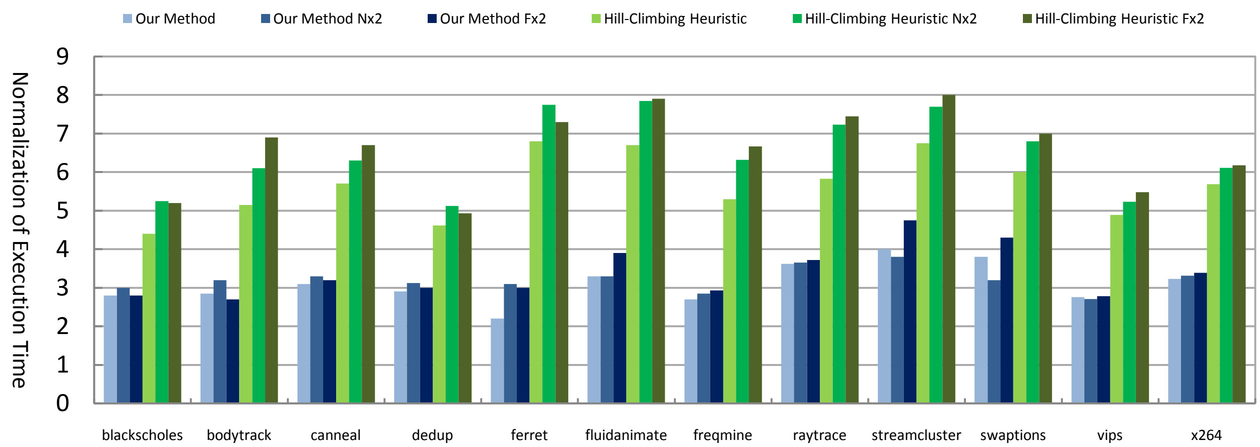


Fig. 11. Execution Overhead Comparison of Our Method and Hill-Climbing Heuristic with Different Cores and Frequency Levels

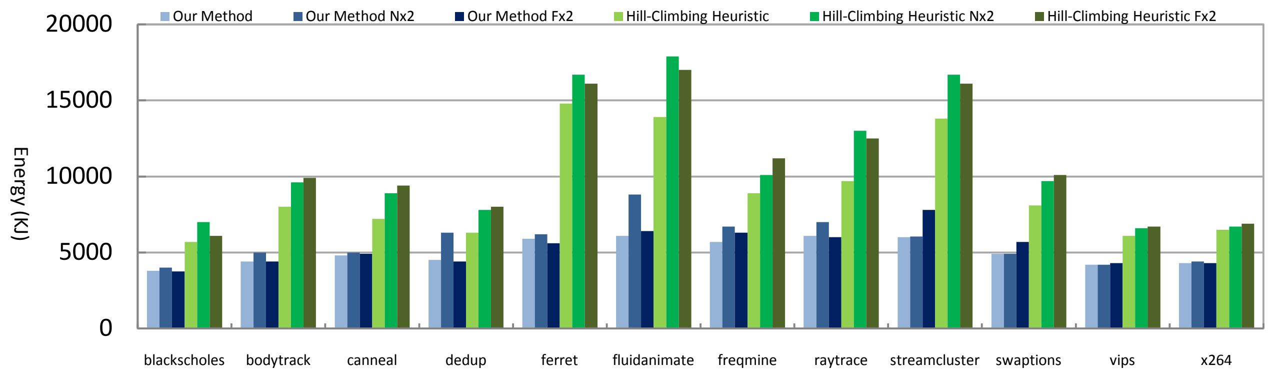


Fig. 12. Energy Overhead Comparison of Our Method and Hill-Climbing Heuristic with Different Cores and Frequency Levels

Table 1. The iterative process of the core number and frequency of each program by using EOFDM

Program	Execute Times	Cores Number	Frequency(GHz)	Energy Cost(kJ)	Execute Time (Normalization)	dn	df	a
Blackscholes Optimal Cores=6 Optimal Frequency=1.7 Lowest Energy=342	1	8	4	492	0.146	-1	-1	2
	2	6	2	345	0.373	-1	-0.94	N/A
	3	6	1.9	344	0.393			
	4	6	1.8	343	0.415			
	5	**6**	**1.7**	342	0.439			
	6	6	1.6	343	0.466			
Bodytrack Optimal Cores=4 Optima lFrequency=2.6 Lowest Energy =498	1	8	4	840	0.209	-1	-1	2
	2	6	2	516	0.493	-1	0.23	1
	3	5	2.2	501	0.481	-1	0.33	1
	4	**4**	**2.6**	498	0.476	-1	0.44	N/A
	5	4	2.5	481	0.514			
Ferret Optimal Cores=4 Optima lFrequency=2.6 Lowest Energy=607	1	8	4	1353	0.254	-1	-0.38	3
	2	5	2.8	701	0.397	-1	-0.21	1
	3	**4**	**2.6**	607	0.483	-1	0.9	N/A
	4	4	2.5	594	0.502			
Fluidanimate Optimal Cores=4 Optimal Frequency=2.7 Lowest Energy=508	1	8	4	912	0.227	-1	-0.9	1
	2	6	2.2.	557	0.472	-1	0.34	1
	3	5	2.5	545	0.441	-1	0.35	1
	4	4	2.8	537	0.437	-1	0.77	N/A
	5	**4**	**2.7**	508	0.486			
	6	4	2.6	499	0.505			
Streamcluster Optimal Cores=4 Optimal Frequency=2.5	1	8	4	1007	0.251	-1	-0.82	2
	2	6	2.4	606	0.351	-1	0.26	2
	3	4	2.9	493	0.424	-1	0.54	N/A

Lowest Energy= 460	4	4	2.8	484	0.439			
	5	4	2.7	476	0.455			
	6	4	2.6	467	0.473			
	7	**4**	**2.5**	460	0.492			
	8	4	2.4	452	0.512			
	1	8	4	546	0.136	-1	-1	2
	2	6	2	370	0.353	-1	0.036	1
Swaptions	3	5	2.1	369	0.398	-1	-0.354	N/A
Optimal Cores=5	4	5	2.0	365	0.418			
Optimal Frequency=1.7	5	5	1.9	362	0.440			
Lowest Energy=357	6	5	1.8	359	0.465			
	7	**5**	**1.7**	357	0.492			
	8	5	1.6	356	0.523			

Table 2. The iterative process of doubling core number and doubling frequency of each program

		32 cores				Frequency step 50MHz			
		EOFDM		Hill-climbing Heuristic		EOFDM		Hill-climbing Heuristic	
program	Execution times	cores	frequency	cores	频率（GHz）	cores	frequency	cores	frequency（GHz）
Blackscholes	1	16	4	16	(4,0.7,0.8)	8	4	8	(4,1.15,1.2)
Optimal Cores=5	2	14	2	8	(4,1.1,1.2)	6	2	4	(4,2.05,2.10,2.15)
Optimal	3	13	0.6	4	(4,2,2.1,2.2)	6	1.9	2	(4)
Frequency=1.8	4	8	2.2	2	(4)	6	1.8	6	(4,1.4,1.45,1.5)
Lowest	5	**5**	**1.8**	6	(4,1.4,1.5)	**6**	**1.7**	**5**	(4,1.7,1.75,**1.8**)
Energy=335	6	5	1.7	**5**	(4,1.7,**1.8**)	6	1.65		
Bodytrack	1	16	4	16	(4,1.2,1.3)	8	4	8	(4,1.6,1.65,1.7)
Optimal Cores=4	2	14	2	8	(4,1.6,1.7)	6	2	**4**	(4,2.45,2.5,2.55,**2.6**)
Optimal	3	12	1.5	**4**	(4,2.5,**2.6**)	5	2.25	2	(4)
Frequency=2.6	4	6	2	2	(4)	**4**	**2.6**	6	(4,1.9,1.95,2.0)
Lowest	5	5	2.2	6	(4,1.9,2.0)	4	2.55	5	(4,2.10,2.15,2.2)
Energy=498	6	**4**	**2.6**	5	(4,2.1,2.2)				
	7	4	2.5						
Ferret	1	16	4	16	(4,3.4,3.5)	8	4	8	(4,1.9,1.95,2.0,2.05)
Optimal Cores=4	2	12	2	8	(4,1.9,2.0,2.1)	5	2.85	**4**	(4,2.4,2.45,**2.55**)
Optimal	3	8	2.1	**4**	(4,2.4,2.5,**2.6**)	4	2.65	2	(4)
Frequency=2.55	4	4	2.8	2	(4)	4	2.6	6	(4,1.9,1.95,2,2.05,2.1)
Lowest	5	4	2.7	6	(4,1.9,2.0,2.1)	**4**	**2.55**	5	(4,2.1,2.15,2.2,2.25)
Energy=600	6	**4**	**2.6**	5	(4,2.1,2.2,2.3)	4	2.5		
	7	4	2.5						
Fluidanimate	1	16	4	16	(4,1.2,1.3,1.4,1.5)	8	4	8	(4,1.7,1.75,1.8,1.85)
Optimal Cores=4	2	12	1.7	8	(4,1.7,1.8,1.9)	6	2.2.	**4**	(4,2.5,2.55,2.6,**2.65**)
Optimal	3	7	2	**4**	(4,2.5,2.6,**2.7**)	5	2.55	2	(4)
Frequency=2.65	4	4	2.9.	2	(4)	4	2.85	6	(4,1.9,1.95,2,2.05,2.1)
Lowest	5	4	2.8	6	(4,1.9,2,2.1)	4	2.8	5	(4,2.1,2.15,2.2,2.25,2.3)
Energy=503	6	**4**	**2.7**	5	(4,2.1,2.2,2.3)	4	2.75		
	7	4	2.6			4	2.7		
	8					**4**	**2.65**		
	9					4	2.6		
Streamcluster	1	16	4	16	(4,3.4,3.5,3.6)	8	4	8	(4,1.9,1.95,2,2.05)
Optimal Cores=4	2	10	2.2	8	(4,1.9,2,2.1)	6	2.35	**4**	(4,2.3,2.35,2.4,2.45,**2.5**)
Optimal	3	7	2	**4**	(4,2.3,2.4,**2.5**)	4	2.85	2	(4)
Frequency=2.5	4	4	2.6	2	(4)	4	2.8	6	(4,1.9,1.95,2,2.05)
Lowest Energy=	5	**4**	**2.5**	6	(4,1.9,2,2.1)	4	2.75	5	(4,2,2.05,2.1,2.15,2.2)
460	6	4	2.4	5	(4,2,2.1,2.2)	4	2.7		
	7					4	2.65		
	8					4	2.6		
	9					4	2.55		
	10					**4**	**2.5**		
	11					4	2.45		
Swaptions	1	16	4	16	(4,0.6,0.7)	8	4	8	(4,0.9,0.95,1,1.05,1.1)
Optimal Cores=5	2	13	1	8	(4,0.9,1,1.1)	6	2	4	(4,2.0,2.05,2.1)
Optimal	3	8	2	4	(4,2,2.1)	5	2..05	2	(4)
Frequency=1.7	4	5	1.8	2	(4)	5	2.0	6	(4,1.3,1.35,1.4,1.45)
Lowest	5	**5**	**1.7**	6	(4,1.3,1.4,1.5)	5	1.95	**5**	(4,1.6,1.65,**1.7**)
Energy=357	6	5	1.6	**5**	(4,1.6,**1.7**)	5	1.9		
	7					5	1.85		
	8					5	1.8		
	9					5	1.75		
	10					**5**	**1.7**		
	11					5	1.65		

VI. CONCLUSIONS

Power-performance assignment for energy-efficient optimization has become one of the most important technologies in the future development of multi-core processors. However, the existing methods of searching for optimal energy-efficient configurations in the space of core number and frequency are characterized by slow convergence speed, great overhead and poor scalability. In this paper, a search method based on feasible direction method (FDM) for energy-efficient optimization has been proposed to sharply reduce search space in the two dimensions and to converge to the point of minimum energy consumption quickly in the iterative process. Simultaneously, the practical energy and performance of each feasible configuration can revise the model calculation. In the future, we will focus on the experiment and theoretical analysis about energy-efficient optimization of heterogeneous many-core processors.

When you submit your final version, after your paper has been accepted, prepare it in two-column format, including figures and tables.

REFERENCES

[1] R. H. Dennard, F. H. Gaensslen, V. L. Rideout, et al. "Design of ion-implanted MOSFET's with very small physical dimensions", *IEEE Journal of Solid-State Circuits*, vol.9, no.5, pp. 256-268, 2007.

[2] D. H. Woo, H. H. Lee. "Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era", *Computer*, vol.41, no.12, pp. 24-31,2008.

[3] Hadi Esmaeilzadeh, Emily Blem, Renee Amant, et al. "Dark Silicon and the End of Multicore Scaling", *IEEE Micro*, vol. 32, no. 3, pp. 122-134, 2012

[4] D. Mark Hill, R. Michael Marty, "Amdahl's Law in the Multicore Era", *Computer*, vol.41, no.7, pp. 33-38, 2008.

[5] Li Jian and F. Martínez Jose, "Dynamic power-performance adaptation of parallel computation on chip multiprocessors", in *the 12th International Symposium on High-Performance Computer Architecture*, New York, 2007, pp. 77-87.

[6] M. Curtis-Maury, A. Shah, F. Blagojevic, et al. "Prediction Models for Multi-dimensional Power-Performance Optimization on Many Cores", in *Proc of the 17th international conference on Parallel architectures and compilation techniques*, New York, 2008, pp. 250-259.

[7] Wang Hao, Sathish Vijay, Singh Ripudaman, et al. "Workload and Power Budget Partitioning for Single Chip Heterogeneous Processors", in *PACT'12. Minneapolis*, Landon, 2012, pp. 401-410.

[8] Lee Jungseob, Kim Nam Sung, "Analyzing Potential Throughput Improvement of Power- and Thermal-Constrained Multicore Processors by Exploiting DVFS and PCPG", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol.20, no.2, pp. 225-235, 2012.

[9] Lee Jeong-Gun, Jung Eungu, and Shin Wook, "An Asymptotic Performance/Energy Analysis and Optimization of Multi-core Architectures", in *International Conference on Microelectronics*, New York, 2008, pp. 85-90.

[10] Trevor Mudge, "Power: a first-class architectural design constraint", *Computer*, vol.34, no.4, pp. 52-58, 2007.

[11] Kim N S and Mudge T. Total Power-Optimal Pipelining and Parallel Processing under Process Variations in Nanometer Technology[C]. ICCAD' 05. NJ: IEEE, 2005:535-540.

[12] R. Ge, X. Feng and K. W.Cameron, "Modeling and Evaluating Energy-Performance Efficiency of Parallel Processing on Multicore Based Power Aware Systems", in *IPDPS 2009*, New York, 2009, pp. 1-8.

[13] Grunwald Dirk, B. Morrey Charles, Philip Levis, et al. "Policies for dynamic clock scheduling", in *Proc of the 4th conference on Symposium on Operating System Design & Implementation*, New Yor, 2000, pp. 6-19.

[14] A. K. Mishra, C. R. Das, M. Kandemir, et al. "CPM in CMPs: Coordinated Power Management in Chip-Multiprocessors", in *International Conference for High Performance Computing, Networking*, Paris, 2010, pp. 1–12.

[15] Vijay Anand Korthikanti, Gul Agha, "Energy-Performance Trade-off Analysis of Parallel Algorithms for Shared Memory Architectures", *Sustainable Computing: Informatics and Systems*, vol.1, no.3, 2011, pp.167–176.

[16] O. Azizi, A. Mahesr, B. C. Lee, et al. "Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis", *Acm Sigarch Computer Architecture News*, vol.38, no. 3, pp. 26-36, 2010.

[17] N. S. Kim, T. Austin, D. Blaauw, et al. "Leakage current: Moore's law meets static power", *Computer*, vol.36, no.12, pp. 68-75, 2003.

[18] R. Jejurikar, C. Pereira, R.Gupta, "Leakage Aware Dynamic Voltage Scaling for Real-Time Embedded Systems", in *the 41st Design Automation Conference*, New York, 2004pp. 275-280.

[19] K. Meng, R. Joseph, R. P. Dick, et al. "Multi-optimization power management for chip multiprocessors", in *International Conference on Parallel Architectures & Compilation Techniques*, New York, 2008, pp.177-186.

[20] Xie Zheng, Li Jianping, Chen Zhi, "Theory and Method for Nonlinear Optimization", Beijing: Higher education press, 2010.

[21] The Multifacet GEMS Development Team. Multifacet GEMS, (2012-01-04)[2014-11-06], http:.research.cs.wisc.edu/gems/

[22] S. Zikos, H. D. Karatza, "Performance and energy aware cluster-level scheduling of compute-intensive jobs with unknown service times", *Simulation Modelling Practice & Theory*, vol.19, no.1, pp. 239-250, 2011.

[23] Zhang Shuai, Liu Zhiyong, Dongrui Fan, et al. "Energy- performance modeling and optimization of parallel computing in on-chip networks", in *ISPA'13*, New York, 2013, pp. 879-886.

[24] Zhang Shuai, Song Fenglong, Liu Zhiyong, et al. "Energy-Performance Analysis and Optimization for Networks-on-Chip", *Chinese Journal of Computers*, vol.36, no.5, pp. 988-1003, 2013.

[25] Gabriel Loh, "The Cost of Uncore in Throughput-Oriented Many-Core Processors", in *Workshop on ALTA*, New York, 2008.

[26] T. Agerwala, S. Chatterjee, "Computer architecture: Challenges and opportunities for the next decade", *IEEE Micro*, vol.25, no.3, pp. 58-69, 2005.

[27] 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)*, New York, 2009.

[28] Ravindra Jejurikar, Cristiano Pereira. "Leakage Aware Dynamic Voltage Scaling for Real Time Embedded Systems", in *Proceedings of the 41st annual Design Automation Conference*, 2004.

[29] Z. Qian, P. Bogdan, C. Y. Tsui, et al. "Performance Evaluation of NoC-Based Multicore Systems: From Traffic Analysis to NoC Latency Modeling", *Acm Transactions on Design Automation of Electronic Systems*, vol.21, no.3, pp. 1-38, 2016.

[30] P. Bogdan, R. Marculescu , "Workload characterization and its impact on multicore platform design", in *International Conference on Hardware/software Codesign and System Synthesis*, New York. 2010, pp. 231-240.

[31] R. David, P. Bogdan, R. Marculescu, "Dynamic power management for multicores: Case study using the intel SCC", in *IEEE/IFIP*, *International Conference on Vlsi and System-On-Chip*, New York, 2012, pp.147-152.

[32] P. Bogdan, R. Marculescu, S. Jain, "Dynamic power management for multidomain system-on-chip platforms: An optimal control approach", *Acm Transactions on Design Automation of Electronic Systems*, vol.18, no.4, pp. 96-95, 2013.

[33] P. Bogdan, "Mathematical Modeling and Control of Multifractal Workloads for Data-Center-on-a-Chip Optimization", in *International Symposium on Networks-On-Chip*, New York, 2015, pp. 1-8.

[34] Paul Bogdan and Radu Marculescu, "Workload characterization and its impact on multicore platform design", in *Proceedings of the eighth*

IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis (CODES/ISSS '10), New York, 2010.

[35] Qian et al. "Performance Evaluation of NoC-Based Multicore Systems: From Traffic Analysis to NoC Latency Modeling", *ACM Trans*, vol.21, no.3, pp. 31-38, 2016.

Hua Jin was born on Sep. 23, 1980**.** She received the Master degree in computer science and technology from Agricultural University of Hebei. Currently, she is a researcher (lecturer) at Agricultural University of Hebei, China. Her major research interests include computer simulation, big data and multiprocessor. She has published many papers in related journals. In addition, he is a co-chair of International Conferences in recent years.

Aixin Wang was born on Dec. 26, 1978**.** He received the Master degree in computer science and technology from Agricultural University of Hebei. Currently, he is a researcher (lecturer) at Agricultural University of Hebei, China. His major research interests include information security and network. He has published many papers in related journals.

Yatao Zhu was born on Oct. 15, 1978**.** He received the PhD degree in computer science and technology from the Institute of Computing Technology, Chinese Academy Sciences. Currently, he is a researcher (post doctor and assistant research fellow) at the Institute of Computing Technology, Chinese Academy Sciences. His major research interests include multiprocessor design and artificial intelligence. He is invited as a reviewer by the editors of some international journals, such as *SUCOM*. He has published many papers in related journals.