Cluster Compute manuscript No.
(will be inserted by the editor)

Whole procedure heterogeneous multiprocessors low-power optimization at algorithm-level

Zhuowei Wang · Naixue Xiong · Hao Wang · Lianglun Cheng · Wuqing Zhao

Received: date / Accepted: date

Abstract Power consumption reduction is the primary problem for the design and implementation of heterogeneous parallel systems. As it is difficult to make progress in the low-power optimization in the hardware layer to meet the increasing need for power optimization, more attention has been paid to low-power optimization in the hardware layer. The relationship between the execution time and dynamic power consumption of programs divided between homogeneous and heterogeneous computing sections is analysed. In addition, the communication power consumption for data transmission and dynamic multi-task allocation are described. Afterwards, this study establishes a power model for the whole procedure of heterogeneous parallel systems. By using this model, a selection algorithm is designed for the optimal frequency of processors with optimal power consumption under time constraints, optimal descent-based time allocation algorithms in multiple computing sections, and profiling dynamic analysis-based integral linear programming at algorithm-level, separately. Finally, the validity of the power optimization algorithm is ascertained using typical applications.

Keywords Whole procedure · Heterogeneous parallel systems · Algorithm-level · Low-power optimization

Z.Wang · L.Cheng
School of Computer Science and Technology, Guangdong University of Technology, Guangzhou 510006, China
E-mail: wangzhuowei0710@163.com
L.Cheng
E-mail: llcheng@gdut.edu.cn

N. Xiong(✉)
Department of Computer Science, Georgia State University, USA; he is the corresponding author in this paper.
E-mail: nxiong@cs.gsu.edu

H.Wang
Department of ICT and Natural Sciences, Norwegian University of Science and Technology, Norway
E-mail: hawa@ntnu.no

W.Zhao
China Southern Power Grid, Guangzhou, China
E-mail: whuzhwq@163.com

1 Introduction

Heterogeneous parallel systems which integrate general-purpose processors with dedicated processors have become an important development for high-performance computing systems. Although this kind of system is characterised by a high peak velocity and a high peak efficiency, they still suffer from high power consumption. For existing supercomputer systems, TianHe-1A and K Computer systems have power consumptions of 4.04 MW and 12.66 MW, separately, while the newly developed TianHe-2A system even reaches a power consumption of up to 17.808 MW. Excessive power consumption gives rise to difficulty in the packaging, power supply, and heat dissipation of such systems. Therefore, reducing power consumption has become an important target for the optimization of heterogeneous parallel systems.

Existing studies on the low-power optimization of heterogeneous parallel systems mainly focus on underlying hardware and upper-level software [1][2]. Low-power optimization in hardware layers has been relatively well-researched and it is therefore difficult to achieve further developments therewith that may satisfy the increasing requirement for power optimization. As a consequence, the optimization of power consumption in the software layer has attracted widespread attention[3][4]. In the software layer, since different processors show dissimilar computation speeds and power

consumption overloads, low-power optimization for heterogeneous parallel systems is different from that for homogeneous systems. To make the most of the efficiency advantage of heterogeneous parallel systems, the following new problems need to be solved in the low-power optimization of these systems.

The modelling objects for the power consumption of heterogeneous parallel systems are complex. At present, there are few studies of the power models of heterogeneous parallel systems, most of which are constructed on the basis of some modification of the power models used for homogeneous systems. However, the processors in heterogeneous parallel systems have different architectures. Moreover, as most main processors are connected with accelerated processing units (APUs) through system buses, additional communication operations are bound to be introduced in the accelerated computation through scheduling APUs. Therefore, the modelling objects for the power consumption of heterogeneous parallel systems are more complex in comparison with those used for homogeneous systems. Traditional modelling objects concerning power consumption are mainly aimed at the dynamic power consumption generated by processors. The model merely considers the programs in a single computing section or maps them to the calculation resources of a specific type of processor(s). Nevertheless, practical scientific and engineering applications are generally composed of several serial and parallel computing sections. Under such conditions, users generally concern themselves with the relationship between power consumption and execution time of the whole application rather than a single computing section.

Energy saving effects are related to the algorithm selected for low-power optimization. With the gradual deepening of relevant investigations, low-power optimization has gradually changed from architecture-level and compiling-level to algorithm-level protocols[5][6]. The algorithm-level is a higher power optimization level compared with that of the compiling-, and architecture-levels. In terms of the low-power optimisation effect, the selection and optimisation of algorithms exert the most significant influence on the power consumption, which can reach up to 90% at maximum[7][8]. Low-power optimisation at algorithm-level can help designers to select an algorithm with low energy consumption and guide the optimisation design of software or hardware in later stages. Meanwhile, on the basis of grasping the application background and the execution characteristics of programs, more targeted and efficient optimising strategies can be formulated.

To solve the aforementioned problems, this research studies low-power optimisation at the algorithm-level for the whole procedure of heterogeneous parallel systems. Meanwhile, from the perspective of the executive characteristics of parallel programs, a power consumption model is established for the whole procedure of heterogeneous parallel systems. By using this model, efficient low-power optimisation methods are designed at algorithm-level. The rest of this study is organised as follows: Section 2 demonstrates the research status and progress of system-level power models and low-power optimisation for software; the contributions of this work are introduced in Section 3; Section 4 analyses, and establishes, the power model for the whole procedure of heterogeneous parallel systems (aiming at dynamic power consumption and communication power consumption); Sections 5 to 7 cover the design of the selection algorithm for the frequency of optimal processors, optimal descent-based time allocation algorithm in multiple computing sections, and profiling dynamic analysis-based integral linear programming; the experimental results are analysed in Section 8, and key conclusions are drawn in Section 9.

2 Related work

2.1 System-level power models

At present, the most commonly seen power models for heterogeneous parallel systems are simulation-based system-level power models. These models reveal the execution of units in an object system by simulating the execution of the system and then adding up the corresponding power consumptions. In comparison with command-level [9]and sampling-based power models [10], system-level power models show higher measuring accuracy and are generally built based on performance simulation. These models measure power consumption based on functional blocks, for example, summator, multiplier, controller, register file, cache, etc [11]. Currently, there are a large number of system-level power models discussed in academic circles [12][13]. These models are established on the basis of simple technological parameters and compute the power consumption of modules using analysis models or obtain empirical data through the use of a bottom-extraction test. The modelling objects are basically single processor units, or the whole processor, and the system power consumption considered is nothing to do with the execution of the applications and merely depends on the processors. However, in heterogeneous parallel systems, owing to the limitations of programming models or architectures, most parallel programs accomplish an application by executing different computing sections in succession using universal microprocessors and APUs. In addition, with the

constant improvement of parallel processing techniques and their support environment, more parallel programs are supposed to use parallel combined heterogeneous multi-processors to deal with single parallel computing sections [14][15]. In this way, the advantages of system parallel processing can be sufficiently exploited. Therefore, adopting existing system-level power models fails to result in the performance of more effective modelling for the overall power consumption [16][17]. Meanwhile, most main processors and APUs in heterogeneous parallel systems are connected using peripheral component interconnect (PCI) interfaces to transmit data with the single peak bandwidth of only 8 GB/s. In particular, the capacity of the video memory of accelerated processors represented by graphic processing units (GPUs) has failed to satisfy the requirements of modern scientific computations, thus further increasing the pressure on bandwidth for data communication [17]. For data-intensive applications, the data communication overhead among processors exerts a significant influence on the high power consumption of heterogeneous systems.

2.2 Low-power optimisation in the software layer

Low-power optimisation technologies in the software layer can be divided into three levels, namely the, architecture-, compiling- and algorithm-levels. Thereinto, dynamic voltage and frequency scaling (DVFS) and core processor unit shut-down are two key technologies used for architecture-level low-power optimisation [18][19][20][21]. Compiling-level low-power optimisation in the software layer mainly focuses on the low-power compiling optimisation of programs. Many scholars have investigated the influence of traditional compiling optimisation, such as, command adjustment, register allocation, cyclic transformation, and data conversion, on power consumption [22][23]. At present, studies of the algorithm-level low-power optimisation for heterogeneous parallel systems remain in their infancy. Korithikanti et al. [24] analysed the expandability of power consumption of parallel algorithms and found, through modelling, that different algorithms have different expandabilities with regards their power consumption. In addition, they pointed out that the number of processors with optimal power consumption under performance constraints needs to be determined at algorithm-level to minimise overall power consumption. Meanwhile, the potential for power optimisation at algorithm-level was revealed. Therefore, carrying out an investigation into low-power optimisation at algorithm-level in the software layer is of great practical significance.

A task scheduling algorithm, based on heterogeneous sensing, is an important means of low-power optimisation in heterogeneous parallel systems at algorithm-level. Several parallel sections in one program can be mapped in a multi-core processor or/and an APU. Since parallel sections show different program characteristics or execution units, the multiple parallel sections in one program always entail different power consumptions [25][26][27][28]. Meanwhile, most existing low-power optimisation methods aim at dynamic power consumption, while few studies have been carried out on the optimisation of communication power consumption. Multiple computing sections in one program are generally characterised by data dependence, while in several computing sections with data dependence, different methods of the division of tasks possibly lead to different communication overheads[29][30].

3 Contributions of this work

This study is conducted on the basis of established GPU power models at architecture-level. It not only considers the power consumption of processors in APUs, but also analyses the overall power consumption of heterogeneous parallel systems based on different executive modes of parallel programs on the systems. By adding the communication overheads arising from communication tasks in main processors and APUs, the power model for the whole procedure of heterogeneous parallel systems is built. On this basis, dynamic power consumption and communication power consumption are optimised at algorithm-level. The specific contributions of this work are as follows:

(1) This study proposes a power model for the whole procedure of heterogeneous parallel systems. In the system-level power modelling, considering the complexity of modelling objects for the power consumption of heterogeneous parallel systems, the power model is established from the perspective of the whole procedure so as to improve the accuracy of the power model. This is realised by analysing the relationship between the execution time and the dynamic power consumption of programs divided into multiple computing sections of its multi-processors. Besides, the common power consumption of data transmission and dynamic multi-task allocation is formally described.

(2) This research proposes algorithms for selecting the optimal frequency of processors and optimal descent-based time allocation algorithms in multiple computing sections. In dynamic low-power optimisation, aiming at the programs in heterogeneous parallel sections, the conditions for realising optimal power consumption are analysed using heterogeneous parallel processing under

the constraints of execution time. Based on this, the relation between power consumption in each computing section and the execution time is established. As a result, the power consumption problem is depicted as a problem for solving general multivariate extremes. In this way, the selection algorithm for the optimal frequency of processors and optimal descent-based time allocation algorithms in multiple computing sections are proposed.

(3) Profiling static analysis-based integral linear programming is proposed in this study. In the optimisation of communication power consumption, the optimisation problem is converted to an integral linear programming problem based on the method of division of tasks, operation level of processors, and static scheduling strategies of tasks. On this basis, this research develops profiling static analysis-based integral linear programming.

4 Energy consumption models for the whole procedure of heterogeneous parallel systems

The execution of parallel programs on heterogeneous parallel systems can be represented, in abstract terms, as shown in Fig.1. Thereinto, S, C, and P indicate the serial computing section, communication section, and parallel computing section, separately. The parallel computing sections independently finished using main processors or APUs are called homogeneous computing section programs; while those jointly realised through main processors and APUs are named heterogeneous computing section programs. The execution characteristics of parallel programs are defined using the notation summarised in Table 1.

Complementary metal-oxide-semiconductors (CMOSs) are basic computer devices, and their power demand consists of two parts: dynamic and static. Dynamic power is generated by CMOS state changes during use, while static power is mainly generated by leakage current when it is idle. In addition to the dynamic and static power, as the main processor and the acceleration components are mostly connected by system bus in heterogeneous system, accelerate execution will introduce additional communication operations to speed up the calculation process. Communication power has gradually become an important part of the heterogeneous system power. Therefore, the total power of heterogeneous systems (P) can be expressed as the sum of dynamic power (P_d), static power (P_s) and communication power (P_m).

$$P = P_d + P_s + P_m \tag{1}$$

$$E = P \times t \tag{2}$$

According to the law of physics, energy consumption is an integration of power into time, usually written as the product of average power and time. Although power and energy are two concepts, they are often used indiscriminately in different research fields: many researchers take energy as the actual optimization objective while designating their quest as low-power optimization. Hence, this research makes no strict distinction between power and energy, except the in those cases for which the optimization objective of power and energy are not in mutual agreement.

4.1 Dynamic energy consumption model

(1) Dynamic energy consumption modelling of homogeneous computing section

In homogeneous computing sections, if S_i is a serial section, the program is finished using a single r_i processor; while if S_i is a parallel section, the program is realised using all r_i processors. The relationship between the dynamic voltage and the frequency of processors can be approximately described by $f = KV^{\gamma-1}$,where K and γ are parameters related to the technology. It is recorded as $\alpha = \frac{\gamma+1}{\gamma-1}$ and therefore it can be regarded that the dynamic power consumption P_d is positively related to the α^{th} power of the frequency f, that is, $P_d = Kf^{\alpha}$. Let the execution time of the i^{th} computing section be denoted as t_i,while N_i and f_i show the number and operating frequency of r_i processors in the i^{th} computing section, separately. Then, the total power consumption in homogeneous computing sections can be expressed as:

$$E_d = \sum_{i=0}^{n-1} N_i K_i f_i^{\alpha} t_i \tag{3}$$

The study aims to minimise the total power consumption for the whole procedure in a given execution time T for the program model composed of multiple computing sections. Thereinto, the time constraint t_i in any computing section S_i is analysed as follows: if the i^{th} computing section S_i is a serial section, it is finished using only a processor under which condition the execution time satisfies the condition of $t_i \geq \frac{S_i}{v_i}$; while if the i^{th} computing section S_i is a parallel one, it is realised using all parallel r_i processors. In this case, the execution time satisfies $t_i \geq \frac{S_i}{N_i v_i}$.

Table 1 Execution characteristics of parallel programs

Notations	Definition
$S = \{s_0 \cdots s_{n-1}\}$	The program is divided into n sections based on the parallelism of computing sections where s_i denotes the task load in the i^{th} computing section.
$R = \{r_0 \cdots r_{m-1}\}$	The heterogeneous parallel system is composed of m kinds of processors.
N_j	The number of j^{th} $(0 \leq j \leq m-1)$ kind of processor r_j.
v_j	Speed at the highest frequency (task load finished by the processor per unit time).

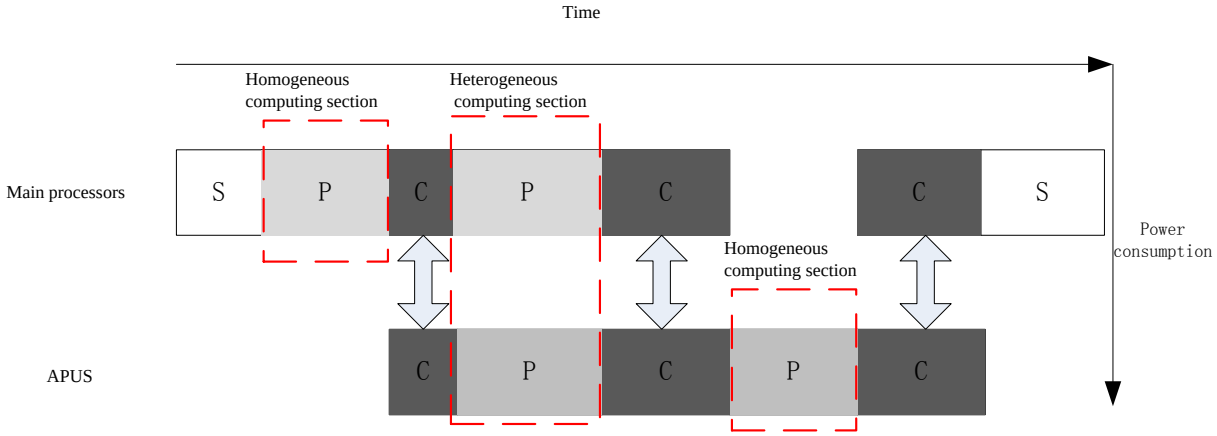


Fig. 1 Classifications of heterogeneous parallel programs

Therefore, the dynamic energy consumption modelling for the whole procedure of homogeneous computing sections can be formally described thus:

$$\begin{cases} minE_d = \sum_{i=0}^{n-1} N_i k_i f_i^{\alpha} t_i \\ s.t. \sum_{i=0}^{n-1} t_i \leq T \\ t_i \geq \frac{S_i}{v_i}, if\ S_i\ is\ serial\ section \\ t_i \geq \frac{S_i}{N_i v_i}, if\ S_i\ is\ parallel\ section \end{cases} \tag{4}$$

(2)Dynamic energy consumption modelling of heterogeneous computing sections

This research mainly studies CPU-GPU heterogeneous parallel systems. Therefore, the processors involved contain only CPUs and GPUs (suppose all CPUs are of the same type, as are the GPUs). The execution time of the i^{th} computing section is recorded as t_i, while N_C and N_G indicate the numbers of CPUs and GPUs used in this section, separately;k_C and k_G represent the relevant constants of CPUs and GPUs, respectively. In addition, f_C and f_G denote the operating frequencies of CPUs and GPUs in this section, separately; v_i^j shows the task load finished by the j^{th} kind of processor per unit time in the i^{th} computing section. Therefore, the total power consumption for programs in heterogeneous sections can be expressed as:

$$E_d = \sum_{i=0}^{n-1} (N_C k_C f_C^{\alpha} + N_G k_G f_G^{\alpha}) \cdot t_i \tag{5}$$

The low-power optimisation for the programs in heterogeneous computing sections can be studied by dividing it into two sub-problems in theory, that is, local-power optimisation in computing sections and overall power optimisation in the whole procedure. For the first sub-problem, it is crucial to establish the relationship between the optimal power consumption of processors and the execution time in computing sections. The second sub-problem is to distribute execution time in different computing sections on the basis of optimal power consumption. Therefore, the low-power optimisation for programs in heterogeneous computing sections can generally be concluded as a multivariate extreme value problem.The dynamic energy consumption modelling for the whole procedure of heterogeneous computing

sections can be formally described thus:

$$\begin{cases} minE_d = \sum_{i=0}^{n-1}(N_C k_C f_C^{\alpha} + N_G k_G f_G^{\alpha}) \cdot t_i \\ s.t. \sum_{i=0}^{n-1} t_i \le T \\ t_i \ge \frac{S_i}{\sum_{r_j \in R_i} v_i^j}, if\ S_i\ is\ serial\ section \\ t_i \ge \frac{S_i}{\sum_{r_j \in R_i} N_j v_i^j}, if\ S_i\ is\ parallel\ section \end{cases} \tag{6}$$

4.2 Communication energy consumption model

(1)Communication energy consumption modeling of homogeneous computing sections

In heterogeneous parallel systems, CPUs and GPUs are connected through a PCI-E bus which cannot be used to adjust the dynamic voltage and frequency. That is, the speed of execution of data communication and power consumption are fixed. Suppose that the PCI-E bus is a special functional unit and its power consumes $P_{m,0}$ and $P_{m,1}$ in operating, and idle, states, separately. Meanwhile, it is assumed that the communication cannot be interrupted, that is, multiple data communication operations need to be executed sequentially. Since the system bus is occupied separately by single communication operations, the communication overhead is positively related to data scale, whereas the data scale depends on the division strategy of two parallel tasks which show data dependency. In the programs of the homogeneous computing sections, the communication power consumption is mainly that communication overhead arising from the transmission of input data from CPUs to GPU storage space and restoring output data from GPUs to CPU storage space. The execution time of communication operations is recorded as t_C^G, showing the overhead of data communication between CPUs and GPUs. While $t_{m,0}$ denotes the time overhead of the PCI-E bus in idle state. Then, the communication power consumption of homogeneous programs is represented by:

$$E_m = P_{m,1} t_C^G + P_{m,0} t_{m,0} \tag{7}$$

(2) Communication energy modelling of heterogeneous computing secitons

in the programs of heterogeneous computing sections, communication power consumption mainly refers to the communication overheads arising from the division of multiple parallel tasks with data dependency in a single computing section. Since the practical efficiency of heterogeneous processors is directly related to the characteristics of the tasks, different division strategies are easily formulated among multiple tasks, thus introducing large communication overheads.$t_{x',z'}^{x,z}$ is recorded to represent the communication overhead of task x in the division mode z and that of the task x' in division mode z'. Then, the communication power consumption of heterogeneous programs can be expressed as:

$$E_m = \sum P_{m,1} t_{x',z'}^{x,z} + P_{m,0} t_{m,0} \tag{8}$$

4.3 Static energy consumption model

With the integrated circuit using nanometre technology, the shrinking transistor size makes the chip power density increase exponentially. The chip temperature also increases dramatically with power consumption, which affects the performance and reliability of system and reduces system lifetime. At the same time, the leakage current increases with chip temperature, so that the static energy consumption exceeds the dynamic energy consumption and has become a major source of chip energy consumption. With 65 nm technology, when the temperature is increased from 60 C to 80 C, the static energy consumption will increase by about 21%. Therefore in this study, we took the chip temperature into account in the power consumption model of the whole program in heterogeneous systems to increase the static energy consumption due to the leakage current. At first, the thermal analysis model of real-time systems was built based on the equivalent RC circuit approach to solve the working temperature of chips according to the thermal conductivity of the processor core. By analysing the relationship between the leakage current and the static energy consumption of the chip, we simulated HISPICE to fit curves and built the function of the leakage current in relation to chip temperature and voltage. Two working reference temperatures were introduced to build the quadratic function of the leakage current and the temperature, so the function of the static energy consumption and the chip temperature was obtained. The specific steps are as follows:

To study the thermal conductivity of the processor core, the equivalent RC circuit approach was employed to establish thermal analysis models in previous research [1]. The following formula is adopted to find the working temperature:

$$\frac{dT_{tem}}{dt} = \frac{P}{C_{th}} - \frac{T_{tem} - T_{amb}}{R_{th} C_{th}} = \alpha P - \beta (T_{tem} - T_{amb}) \tag{9}$$

Where T_{tem} and T_{amb} represent the chip temperature and ambient environment temperature, respectively, while

P denotes the chip energy consumption at time t;R_{th} and C_{th} refer to the equivalent thermal resistance and equivalent heat capacity, respectively. The systemic modes of the processor can be divided into the working state and dormant state. The processor only executes tasks in the working state, or it will return to its dormant state to reduce energy consumption and its temperature. The static energy consumption in the working state can be expressed by:

$$P_{static} = N_{gate} I_{leakage} V_{dd} \tag{10}$$

By using HSPICE to fit curves, the leakage current, as a function of the temperature and voltage, can be expressed as:

$$I_{leakage} = I(V_0, T_0)(AT^2 e^{\frac{\alpha V_{dd}+\beta}{T_{tem}}} + Be^{\gamma V_{dd}+\zeta}) \tag{11}$$

where A,B,α,β,γ,and ζ,indicate the empirical parameters as decided by the production process. When the working temperature T_{tem} changes within the normal range of 300 K to 380 K, $e^{\frac{1}{T_{tem}}}$ fluctuates slightly. After setting V_{dd},ference [2] further simplified the leakage current to the quadratic function of the temperature by introducing two reference temperatures TH and TL. The given execution time is T, so the static energy consumption related to the leakage current is calculated thus:

$$E_{static} = N_{gate}(\hat{A}T_{tem}^2 + \hat{B})V_{dd} \cdot T \tag{12}$$

where

$$\hat{A} = \frac{I_{leakage}(TH, V_{dd}) - I_{leakage}(TL, V_{dd})}{TH^2 - TL^2} \tag{13}$$

$$\hat{B} = I_{leakage}(TL, V_{dd}) - \hat{A} \times TL^2 \tag{14}$$

Therefore, the power model for the whole procedure, in homogeneous computing sections canbe formally described as follows:

$$\begin{cases} minE = \sum\limits_{i=0}^{N-1} N_i k_i f_i^{\alpha} t_i + P_{m,1} t_C^G + P_{m,0} t_{m,0} \\ +N_{gate}(\hat{A}T_{tem}^2 + \hat{B})V_{dd}T \\ \hat{A} = \frac{I_{leakage}(TH,V_{dd}) - I_{leakage}(TL,V_{dd})}{TH^2 - TL^2} \\ \hat{B} = I_{leakage}(TL, V_{dd}) - \hat{A} \times TL^2 \\ s.t. \sum\limits_{i=0}^{N-1} t_i + t_C^G + t_{m,0} \le T \\ t_i \ge \frac{S_i}{v_i}, if\ S_i\ is\ serial\ section \\ t_i \ge \frac{S_i}{N_i v_i}, if\ S_i\ is\ parallel\ section \end{cases} \tag{15}$$

For the i^{th} computing section, the execution time of tasks depends on the processors working for the longest time. Tasks are divided only in parallel computing sections rather than in serial computing sections. Therefore, $t_{v,z}$ is the execution time for task v in division mode z. Then, the power model for the whole procedure, in heterogeneous computing sections, after considering the communication,dynamic, and static energy consumption can be formally described as follows:

$$\begin{cases} minE = \sum\limits_{i=0}^{n-1} (N_C K_C (f_z^C)^{\alpha} + N_G K_G (f_z^G)^{\alpha}) \cdot t_{x,z} \\ +P_{m,1} t_{x',z'}^{x,z} + P_{m,0} t_{m,0} + N_{gate}(\hat{A}T_{tem}^2 + \hat{B})V_{dd}T \\ \hat{A} = \frac{I_{leakage}(TH,V_{dd}) - I_{leakage}(TL,V_{dd})}{TH^2 - TL^2} \\ \hat{B} = I_{leakage}(TL, V_{dd}) - \hat{A} \times TL^2 \\ s.t. \sum\limits_{i=0}^{n-1} t_{x,z} + t_{x',z'}^{x,z} + t_{m,0} \le T \\ t_{x,z} \ge \frac{S_i}{\sum\limits_{r_i \in R_i} v_i^j}, if\ S_i\ is\ serial\ section \\ t_{x,z} \ge \frac{S_i}{\sum\limits_{r_i \in R_i} N_j v_i^j}, if\ S_i\ is\ parallel\ section \end{cases} \tag{16}$$

5 The selection algorithm for the optimal frequency of processors

To optimise the dynamic power consumption in homogeneous computing sections, it is necessary to analyse the relationship to be satisfied among processors when the total power consumption of programs is optimal at first. Based on this, the relationship between the optimal frequency in computing sections and the time constraints can be obtained. Then, according to the power consumption in each computing section, the optimal operating frequency of processors in each computing section can be calculated under the given time constraints. In this way, the selection of the best algorithm for the optimal frequency of processors is acquired.

5.1 Balance theorems of power consumption in homogenous computing sections

Theorem 1: According to the total execution time T, when the power consumption of the system becomes optimal, the processors satisfy the following requirements: (1)If $T \ge \frac{\varsigma}{\pi^{\frac{1}{\alpha}}}$ the total power consumptions of processors in different computing sections are equal. That is, if $\forall S_i, S_j \in S$ is satisfied, $N_i p_i = N_j p_j$is always tenable. Under such conditions, the operating frequency of processors in the i^{th} computing section is $f_i = \frac{\omega_i \varsigma}{N_i V_i T}$. (2)If $\sum\limits_{i=0}^{n-1} \frac{S_i}{N_i V_i} \le T \le \frac{\varsigma}{\pi^{\frac{1}{\alpha}}}$, processors in some computing sections certainly work at the highest frequency, and those

in other computing sections show the same total power consumption. Proof: In heterogeneous parallel systems, the maximum operating frequencies of different processors are possibly different. Therefore, this study adopts the relative operating frequency as a substitute for the actual value. That is, the actual power consumption of processors is $p = Pf^{\alpha}$ where P denotes the maximum power consumption of processors in a computing section and f denotes the relative operating frequency of the processors (the ratio of the actual operating frequency of processors to their maximum frequency), separately. Li. K [30] proposed that in homogeneous multiprocessor systems, when the total energy consumption becomes optimal, all processors are bound to work at the same frequency. That is, they have the same power consumption. Therefore,p_i and P_i are used to denote the actual power consumption and the maximum power consumption of homogenous processors in the i^{th} computing section, respectively. It may be found that:

$$P_i = P_i f_i^{\alpha} = P_i(\frac{S_i}{N_i V_i t_i})^{\alpha} \tag{17}$$

Let $\beta_i = (\frac{P_i}{V_i^{\alpha}})^{\frac{1}{\alpha-1}}$, then the total dynamic power consumption for programs is expressed as:

$$E_d = \sum_{i=0}^{n-1} N_i K_i f_i^{\alpha} t_i = \sum_{i=0}^{n-1} N_i \frac{S_i^{\alpha}\beta_i^{\alpha-1}}{t_i^{\alpha} N_i^{\alpha}} t_i = \sum_{i=0}^{n-1} N_i \frac{S_i^{\alpha}\beta_i^{\alpha-1}}{t_i^{\alpha-1} N_i^{\alpha-1}} \tag{18}$$

The computing time of the computing sections is described as $F : \sum_{i=0}^{n-1} t_i - T = 0$. This study solves for the optimal energy consumption using the Lagrangian multiplier method. Let $\frac{\partial E_d}{\partial t_i} = h\frac{\partial F}{\partial t_i}$ where h is the Lagrange multiplier, then we obtain $\frac{S_i^{\alpha}\beta_i^{\alpha-1}}{t_i^{\alpha} N_i^{\alpha-1}} = \frac{h}{1-\alpha}$. The total energy consumption for all processors in the i^{th} computing section is represented by:

$$N_i p_i = \frac{S_i^{\alpha}\beta_i^{\alpha-1}}{t_i^{\alpha} N_i^{\alpha-1}} = \frac{h}{1-\alpha} \tag{19}$$

According to Formula (9), under the condition of optimal power consumption, $N_i p_i = N_j p_j$ is always tenable for all $\forall S_i, S_j \in S$. That is, the total dynamic power consumptions of processors in different computing sections are equal. Formula holds true under the condition that the operating frequency of all processors is smaller than their maximum frequency. By using the Lagrangian multiplier method, it is found that:

$$t_i = (\frac{1-\alpha}{\lambda})^{\frac{1}{\alpha}} \frac{S_i}{(\frac{N_i}{\beta_i})^{\frac{\alpha-1}{\alpha}}} \tag{20}$$

Let $\omega = (\frac{N_i}{\beta_i})^{\frac{\alpha-1}{\alpha}}$,then $t_i = (\frac{1-\alpha}{\lambda})^{\frac{1}{\alpha}} \frac{S_i}{\omega_i}$. Let $\varsigma = \sum_{i=0}^{n-1} \frac{S_i}{\omega_i}$, then $t_i = \frac{S_i}{\varsigma\omega_i} T$.Therefore, the operating frequency f_i of processors in the i^{th} computing section is:

$$f_i = \frac{S_i}{t_i N_i V_i} = \frac{\omega_i \varsigma}{N_i V_i T} \tag{21}$$

If the operating frequency of processors in any computing section requires to be not larger than 1, it needs to satisfy $T \geq \frac{\omega_i \varsigma}{N_i V_i} = \frac{\tau}{(N_i P_i)^{\frac{1}{\alpha}}}$. Suppose that $\pi = min\{P_i N_i | 0 \leq i \leq n-1\}$, then the execution time T needs to satisfy $T \geq \frac{\varsigma}{\pi^{\frac{1}{\alpha}}}$. The lower bound of the execution time T is $\sum_{i=0}^{n-1} \frac{s_i}{N_i V_i}$. When $\sum_{i=0}^{n-1} \frac{s_i}{N_i V_i} \prec T \leq \frac{\varsigma}{\pi^{\frac{1}{\alpha}}}$, the sets of computing sections with total power consumptions π are recorded as $S' = \{s_i | P_i N_i = \pi, s_i \in S\}$, and $\varsigma' = \varsigma - \sum_{s_i \in S'} \frac{s_i}{\omega_i}$ and $\pi' = min\{P_i N_i | s_i \in S - S'\}$. In the subset S' of computing sections, the relative operating frequency of all processors is greater than 1. Since dynamic energy consumption E_d s a convex function of t_i , it can be verified that processors in the system work at the highest frequency when their power consumption is optimal.

5.2 The selection of an algorithm for the optimal frequency of processors

With regard to the selection of the algorithm for the optimal frequency of processors, it is crucial to exclude the allocation results which violate the frequency constraint under the first constraint in the balance theorems of power consumption in homogenous computing sections. This exclusion is conducted according to the increasing order of the total power consumption of processors in each computing section. Meanwhile, the processors which work at the highest frequency are removed until the execution time of the rest of the computing section sets satisfies the time constraint $\pi \geq \frac{\varsigma}{\pi^{\frac{1}{\alpha}}}$. In this way, the optimal operating frequency can be calculated. Fig.2 shows the selection algorithm for the optimal frequency of processors.

In practical scenariosFrequent DVFS can result in extra hardware overhead, especially when the number of the processors increases. In the research, we reduced system energy consumption by scaling the operating frequency of the processors, and parallel task partitioning. Actual processors only can be operated at finite frequencies; however, parallel task partitioning, as a finer scaling method, can decrease system energy consumption by effectively applying the performance discrepancy of heterogeneous multi-core processors. The

authors found that, when the frequency of the processors is as low as the lowest operating frequency in the experiment, the power consumption cannot be reduced through the DVFS method when the relaxing factor increases further. In such conditions, the only choice is to transfer the tasks to high-efficiency processors through the scaling of task partitioning. Therefore, to avoid frequent application of DVFS, we should try to combine the frequency scaling and parallel task partitioning to reduce energy consumption. In addition, The fined-grained power management mechanism of Thread Motion (TM) can be used[]. To avoid the frequent scaling of the voltage and frequency of the processors, the TM method meets the computational demands of different applications through rapid TM among the cores of multiple processors with different computation speeds. The experimental results indicate that the method can improve the execution performance of processors when each core frequency is independently scaled through the scaling of two levels of voltage and frequency.

6 Optimal descent-based time allocation algorithm

For the dynamic low-power optimization of programs in heterogeneous computing sections, it is necessary to analyse the relationship to be satisfied among processors when the total power consumption of programs becomes optimal at first. Then, according to the execution time T, the dynamic low-power optimization of programs in heterogeneous computing sections can be regarded as a process with which to solve the time allocation problem of each computing section under the time constraint of the whole procedure. Differing from the programs in homogeneous computing sections, the optimal power consumption for the programs in heterogeneous computing sections is a piecewise function concerning the execution time. Therefore, it is difficult to obtain the optimal solution of the problem based on such an analysis. As a consequence, an optimal descent-based time allocation algorithm in multiple computing sections is proposed.

6.1 Balance theorems of power consumption in heterogeneous computing sections

Theorem 2: Suppose that the parallel section s s constituted by a heterogeneous multiprocessor set R in a parallel manner. According to the total execution time T,when the power consumption of the system reaches the optimal, the processors satisfy:

(1)If $t \geq \frac{s}{\rho}\psi^{\frac{1}{\alpha-1}}$, all processors show equal efficiency, that is, $\frac{V_j}{P_j} = \frac{V_k}{P_k}, \forall r_j, r_k \in R$. In this case, the operating frequency of the j^{th} type of processors is $f_j = \frac{s}{\rho t \beta_j V_j}$.

(2)If $\frac{s}{\sum_{r_j \in R} N_j V_j} \leq t \leq \frac{s}{\rho}\psi^{\frac{1}{\alpha-1}}$, there are certainly some processors working at the highest frequency and the processors in the rest processor sets have identical efficiencies.

Proof: In the programs of heterogeneous computing sections, the overall task loads accomplished by the j^{th} type of processors r_j are $s_j (0 < s_j < s)$. Thus it is known that the dynamic power consumption of these processors is:

$$p_j = P_j(f_j)^\alpha = P_j(\frac{s_j}{V_j N_j t})^\alpha \tag{22}$$

Let $\beta_j = (\frac{P_j}{V_j^\alpha})^{\frac{1}{\alpha-1}}$

$$p_j = \frac{\beta_j^{\alpha-1} S_j^\alpha}{N_j^\alpha t^\alpha} \tag{23}$$

The total power consumption of the programs is given by:

$$E_d = \sum_{r_j \in R} p_j N_j t = \sum_{r_j \in R} \frac{\beta_j^{\alpha-1} S_j^\alpha}{N_j^\alpha t^{\alpha-1}} = \frac{1}{t^{\alpha-1}} \sum_{r_j \in R} \frac{\beta_j^{\alpha-1} S_j^\alpha}{N_j^{\alpha-1}} \tag{24}$$

Considering that the total task constraint in computing sections is $F = \sum_{r_j \in R} S_j - S = 0$,it is known that the expression of power consumption and the constraints are functions of S_j. Therefore, the extreme values can be solved through the use of the Lagrangian multiplier method. Let $\frac{\partial E_d}{\partial S_i} = h\frac{\partial F}{\partial S_i}$ where h is a Lagrangian multiplier, $S_j^{\alpha-1} = \frac{h t_i^{\alpha-1} N_j^{\alpha-1}}{\alpha \beta_j^{\alpha-1}}$ is obtained. By substituting $S_j^{\alpha-1}$ into the expression for the power consumption, it may be seen that: $p_j = \frac{\beta_j^{\alpha-1} S_j^\alpha}{N_j^\alpha t^\alpha} = \frac{h S_j}{\alpha t N_j}$,

$$\frac{S_j}{N_j t p_j} = \frac{v_j}{p_j} = \frac{\alpha}{h} \tag{25}$$

Where, $v_j (v_j = \frac{S_j}{r_j t})$ is the actual operating speed of processors r_j in the parallel section s. Based on Formula (15), the efficiencies of all processors are equal when the total energy consumption reaches the optimal value:

$$\frac{v_j}{p_j} = \frac{v_k}{p_k}, \forall r_j, r_k \in R \tag{26}$$

The task load of $S_j = \frac{N_j \beta_j^{-1}}{\rho} S$ is allocated to the j^{th} type of processors which work at a frequency of $f_j =$

Algorithm 1 The selection algorithm for the optimal frequency of processors in homogeneous computing sections

It is known that there is a computing section set S under time constraint T and computing sections show a mapping relationship with processors as $F: s_i \to r_i, r_i = \langle P_i, V_i, N_i \rangle, r_i \in R$.

To solve for the operating frequency f_i of processors in each computing section:

1: $\pi = \min\{P_i N_i \,|\, s_i \in S\}$

2: $\tau = \sum_{s_i \in S} s_i / \omega_i$ where $\omega_i = (N_i / \beta_i)^{(\alpha-1)/\alpha}$ and $\beta_i = (P_i / V_i^{\alpha})^{1/\alpha-1}$.

3: *while* $T \geq \dfrac{\tau}{\pi^{1/\alpha}}$ *do*

4: The computing section subset satisfying $S_{sub} = \{s_i \,|\, P_i N_i = \pi\}$ is then selected.

5: For all computing sections: $s_i \in S_{sub}$, let $f_i = 1, T = T - \sum_{s_i \in S_{sub}} \dfrac{s_i}{V_i N_i}$.

6: The subset of computing sections is removed, that is, $S = S - S_{sub}$. Then, π and τ are recalculated.

7: *end while*

8: For $s_i \in S$, let $f_i = \dfrac{\omega_i \tau}{N_i V_i T}$.

Fig. 2 The selection algorithm for the optimal frequency of processors in homogeneous computing sections

$\frac{S_j}{v_j N_j t} = \frac{N_j \beta_j^{-1}}{\sum_{r_k \in R} N_k \beta_k^{-1}} \frac{S}{v_j N_j} = \frac{S}{\rho t \beta_j v_j} \leq 1$ where $\rho = \sum_{r_k \in R} N_k \beta_k^{-1}$. Let $\psi = max\{V_j/P_j | r_j \in R\}$, that is, the execution time of parallel sections must satisfy the following inequality:

$$t \geq \frac{s}{\rho \beta_j V_j} = \frac{s}{\rho}(\frac{V_j}{P_j})^{\frac{1}{\alpha-1}} = \frac{s}{\rho}\psi^{\frac{1}{\alpha-1}} \tag{27}$$

The lower bound to t is $\frac{s}{\sum_{r_j \in R} N_j V_j}$. When $\frac{s}{\sum_{r_j \in R} N_j V_j} \leq t \leq \frac{s}{\rho}\psi^{\frac{1}{\alpha-1}}$, there is at least one type of processors r_j whose operating frequency f_i is greater than 1. That is, the task load assigned to the j^{th} processor exceeds the maximum amount of computations that can be finished within the time constraint. Since the dynamic energy consumption E_d is a convex function of s_j , the total energy consumption reaches its optimum value when $s_j = V_j N_j t$. Afterwards, the remaining tasks $s - s_j$ are allocated to other processors. Under such conditions, if $t \geq \frac{s}{\rho}\psi^{\frac{1}{\alpha-1}}$, the efficiency of the processors has to be balanced; otherwise, the above process is repeated until $t \geq \frac{s}{\rho}\psi^{\frac{1}{\alpha-1}}$.

6.2 Optimal descent-based time allocation algorithm in multiple computing sections

Based on the balance theorems of power consumption in heterogeneous computing sections, the relationship between the optimal power consumption and the execution time in each computing section can be established. On this basis, the time allocated in each section for the whole procedure under time constraints can be solved. The descent in this algorithm means the power consumption reduced per unit time. The difference between the relaxed time constraint and the shortest execution time of programs is recorded as ΔT and this is called as the time to be allocated. This algorithm firstly divides

the time to be allocated into N sections. In each iteration of the algorithm, the time slice $\frac{\Delta T}{N}$ is distributed to the computing section with the largest power consumption reduction in current computing sections. The above iteration is repeated until no time can be allocated. Fig.3 illustrates the optimal descent-based time allocation algorithm in multiple computing sections.

7 Profiling static analysis-based integral linear programming

The low-power optimization for the whole procedure of heterogeneous parallel systems considering communication energy cost aims to minimise the total power consumption of the systems by determining the method of division of each task and the operating frequency of processors. This needs to satisfy the given performance constraints of programs simultaneously, and the total power consumption of the systems includes that for computing and communication. The low-power optimization based on integral linear programming (ILP) refers to the solution of the optimal solutions of problems using integral linear programming on the basis of obtaining all execution information about the required tasks. The information includes the method of division of tasks, run-levels of processors, and the static scheduling strategies of tasks. Therefore, the whole optimization process is mainly divided into two stages, namely, profiling-based static analysis, and the stage for solving optimal solutions through integral programming. In the static analysis stage, the task graphs of applications are established to describe the dependency of multiple computing tasks in the applications so as to guide the profiling process.

7.1 Profiling-based static analysis stage

(1)In the processor set P, the processor $p(p \in P)$ has n_p run-levels. The power consumption of task v at the k^{th} run-level of processor p is $p_{v,p,k}(0 \leq k \leq n_p - 1)$.

(2)Under mode of division z,task v is executed for $t_{v,z,p,k}$ by processor p at run-level k,For parallel computing sections, the execution time of tasks relies on the processor which works for the longest time. That is, the execution time of task v under mode of division z is $t_{v,z} = max\{t_{v,z,p,k}|p \in P, k \in [0, n_{p-1}]$.

(3)The execution time of the communication operation $e_v^{v'}$ is $t_{v',z'}^{v,z}$, which shows the data communication overhead of task v under mode of division way z, and that of task $v^{'}$ under mode of division way $v^{'}$.

7.2 ILP variables

For tasks $v \in V$, they are performed by processors $p \in P$ at run-level $k \in [0, n_p - 1]$ under the division strategy $z \in Z_v$, then: (1)The binary variable D is set to demonstrate the relationship between tasks and division strategies. If $D_{v,z} = 1$, task v is conducted under division strategy z.

(2)The binary variable M represents the mapping relationship between tasks and processors. When $M_{v,p} = 1$, task v is performed on processor p.

(3)The relationship between tasks and the run-levels of the processors is represented by the binary variable L. If $L_{v,p,k} = 1$,, task v is executed on processor p at run-level.

(4)The binary variable $U_{v,z,p,k}$ shows that task v is accomplished on processor p at run-level k in mode of division z, then $U_{v,z,p,k} = 1$. That is, it is supposed to meet $D_{v,z} = 1$ and $L_{v,p,k} = 1$ simultaneously.

7.3 ILP constraints

(1)System constraints: tasks can only be executed under one division strategy and at a certain run-level of a single processor.

(2)For the execution time and communication time of tasks, the relationship of the binary variable $U_{v,z,p,k}$ with $D_{v,z}$ and $L_{v,p,k}$ can be represented using the linearisation technique for non-linear constraints.

(3)Constraints concerning the assigning time and terminal time of tasks: for source nodes, their start time of execution is supposed to be longer than, or equal to, the assigning time; while for end nodes, the end time is expected to be no longer than the terminal time.

(4)Dependency constraints among tasks: data communication operations have to be performed after tasks are finished.

(5)As multiple computing tasks are assigned on one type of processor cannot be executed concurrently, it is necessary to sort these tasks. For any tasks with dependency distributed on the same kind of processors, their sequence is determined based on the dependency. For two tasks without dependency, they are executed merely under two conditions: task 2 is carried out after task 1 is finished, or task 1 is performed after the execution of task 2.

7.4 Optimal solutions based on integral programming

Considering variable constraints, minimising the total power consumption needed by heterogeneous parallel systems for accomplishing applications under given time

Algorithm 2: Optimal descent-based time allocation algorithm in multiple computing sections

It is known that the computing sections are initially conducted for $\{t_i\}$ under time constraint T, and there is a computing section set *s*. In addition, $\{e_i\}$ represents the function of the optimal power consumption of each computing section with the execution time. Besides, the iteration is performed N times.

To solve the time allocation t_i required for each computing section:

1: $\triangle T = T - \sum_{i=0}^{n-1} t_i$

2: Time slices are set as $\triangle t = \triangle T / N$.

3: *for* i=1:N do

4: The current computing section s_k with optimal descent is selected, that is,

$$\triangle e_k(t_k) = \max\{\triangle e_i(t_i), 0 \le i \le n-1\} \quad \text{where} \quad \triangle e_i(t_i) = e_i(t_i) - e_i(t_i + \triangle t).$$

5: Time slices are assigned. Let $t_k = t_k + \triangle t$.

6: *end* for

Fig. 3 Optimal descent-based time allocation algorithm in multiple computing sections

constraints is taken as the objective here. According to the power model established based on communication-aware multi-task division for the whole procedure system, the optimal solutions with regard to power consumption can be computed through integral linear programming.

8 Profiling static analysis-based integral linear programming

8.1 Test platform and test cases

The test platform in this study is a heterogeneous system composed of an Intel Core I7 920 Quad-Core CPU and two AMD 4870 GPUs. To examine the efficiency of the algorithms proposed on this system, the frequency of the storage of one GPU kernel is adjusted from 900 MHZ to 700 MHz so as to obtain two different GPU kernels with different performances. Thereinto, the kernel with the higher performance is recorded as GPU-H, while that with the lower performance is designated GPU-L. The specific parameters for this test platform are listed in Table 2.

Eight typical scientific applications are adopted (Table 3): the first six are selected from AMD APP SDK2.2 and realised using the OpenCL language. As the OpenCL language is applicable to both CPU and GPU platforms, the execution efficiency of the two platforms can thus be compared. At present, the applications in AMD APP SDK generally contain a kernel program. To test the effect of the optimization algorithms proposed in this study on multi-kernel programs, other test programs C SWIM and MGRID C are selected. The SWIM program is a solver for the equations of two-dimensional diving waves and is composed of three kernel computing processes and a parallel reduction process. Therefore, the three kernel computing processes can be performed through mapping them on GPUs, while the reduction process is finished using the CPU. The MGRID program is used to solve the three-dimensional Poisson equations using multiple grids and consists of four kernel computing processes. Therefore, the computing sections with the largest amount of computations can be accomplished by mapping them on GPUs, while those at smaller scales are completed by the CPU.

Table 2 Test platform parameters

Notations	Definition	
Processors	Intel Core I7 920 cpu	AMD 4870 GPU-H/GPU-L
Frequencies of processors(GHZ)	2.67,2.4,2.0,1.6	0.75,0.65,0.55
Storage frequency(GHZ)	1.33(DDR3)	0.9/0.7(GDDR5)
Cache	L1 I32 kB, D32 kB, L2 256 kB, L3 8 MB	-
Memory	8GB	1GB

Table 3 Test cases

Test program	Description	Problem size
Binomial option [BO]	Binomial option pricing model	262,144
Black-Scholes [BS]	Black-Scholes model for European options	1048,567
DCT	Discrete cosine transform	4096×4096
Matrix multiplication [MM]	Matrix multiplication	4096×4096
N-body[NB]	Particle simulation	40960
Monte Carlo Asian[MCA]	Monte Carlo analysis	40960
SWIM	Shallow water equation solver	2048×2048
MGRID[MG]	3-d Poisson equation solver	$256 \times 256 \times 256$

8.2 Experimental results and analysis

(1)Evaluation of lower-power optimization for programs in homogeneous computing sections

As MGRID and SWIM are composed of multiple kernel programs, they are mapped to the CPU and the GPU-H according to the characteristics of each kernel program to be executed sequentially. Taking MGRID for an example, it solves the Poisson equation using a multigrid method by executing the large-scale computation of fine grids on the GPUs while mapping the small-scale calculation onto the CPU for execution. As for SWIM, the sub-procedures including CALC1, CALC2, and CALC3, which have the highest power consumption are executed after being mapped onto the GPUs, while the reduction process with its more complex control structure is executed on the CPU.

Fig.4 show the variations of the optimal power consumption of MGIRD and SWIM programs with the time constraint, respectively. Table 4 and Table 5 demonstrate the operating frequencies of processors in all computing sections when the programs are under the constraint of different relaxation factors. As the processors have a minimum frequency, increasing the relaxation factor after the frequency of the processors is minimised cannot save any more power. For this reason, the figures only show the variation of power consumption with time constraint as far as the minimum frequency. In MGRID, the CPU accomplishes the computing process for small computational demand operations, so that less time and power are consumed; while the GPUs consume more power in executing large granular computations. As shown in Table 4, when the time constraint is applied, the operating frequency of kernel computations accomplished by the GPUs is reduced in priority. In SWIM, the CALC1, CALC2, and CALC3 computing processes realised by the GPUs are memory-intensive programs, and especially CALC3, which has the least dynamic power consumption compared with these other two programs. Therefore, the operating frequency for the computation of CALC3 is reduced last. As shown in Figures 4(a) and (b), when processors have fewer frequency series, the power consumption of processors can be optimised in a relatively small space. When the execution time is extended by 30% and 15%, the operating frequency of processors in each computing section reaches its minimum. Under such conditions, the energy cost arising from dynamic power consumption cannot be further reduced by the DVFS method.

(2) Assessment of low-power optimization for programs in heterogeneous computing sections The existing applications based on heterogeneous parallel systems are mainly composed of a single kernel program, as shown Table 3: except for MGRID and SWIM, the other six test programs are individually constituted by a kernel program. Therefore, these six kernel programs are combined to form a multi-kernel program to assess the low-power optimization of programs in heterogeneous computing sections.

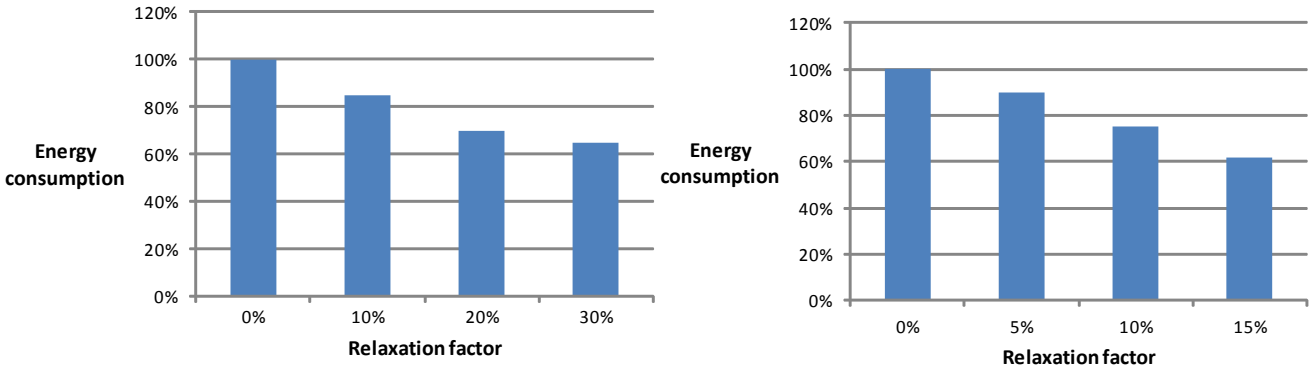
Before evaluating the optimization method, the programs are compared with regard to their performance and power consumption on heterogeneous processors. Fig.5 shows the comparison of the execution time and

Table 4 Variation of the run-level of processors with the time constraint (MGRID)

			GPU		CPU
Relaxation factor	RESID	RPRJ3	INTERP	PSINV	OTHER
0%	0	0	0	0	0
10%	1	1	1	1	0
20%	2	1	1	2	0
30%	2	2	2	2	3

Table 5 Variation of the run-level of processors with the time constraint (SWIM)

		GPU		CPU
Relaxation factor	CALC1	CALC2	CALC3	REDUC
0%	0	0	0	0
5%	1	1	0	0
10%	2	1	1	1
15%	2	2	2	3



(a)Variation of power consumption with time constraint (MGRID)

(b)Variation in power consumption with time constraint (SWIM)

Fig. 4 Variation of power consumption with time constraint (MGRIDSWIM)

power consumption of these programs which are completely mapped onto the GPU-H and the GPU-L: all of the results are the normalised values with those mapped onto the CPU used as benchmarks. Therefore, the larger the values, the higher the performance or power gain. As can be seen from the figure, the performance of all programs can be improved greatly on GPUs, particularly for computation-intensive applications such as BO, BS, DCT, and MM. Compared to the significant improvement in performance, power consumption is less obviously optimised on GPUs and the optimised power consumption is 57% of the original. This indicates that, considering the large difference between the performances of GPUs and CPUs, the performance is slightly improved while allocating a kernel program on the CPU or GPUs. Therefore, the GPU-H and GPU-L (obtained by adjusting the frequency of memory of GPUs) are regarded as two heterogeneous processers to allocate and map parallel computing tasks onto these two GPUs for parallel execution.

The shortest time T taken for the parallel execution of the two GPUs is applied as the benchmark to assess the variation of the optimal power consumpiton of programs in heterogeneous computing sections under the time constraint , where is the relaxation factor. Figure 5 shows the changes in the power consumption of each kernel program with changing relaxation factor. According to the parallel program division with optimal power consumption and the selection algorithm for the optimal frequency of processors, under a certain time constraint, the power consumption of the system can be reudced by adjusting the operating frequency of the processors or the method of division of parallel tasks. Therefore, although processors can only be operated at finite dispersed frequencies, the division of parallel tasks, as a fine-grid adjustment method, is able to decrease the power consumption by exploiting the performance differences between heterogeneous multi-processors. At the same time, the curve showing the relationship between the total power consump-

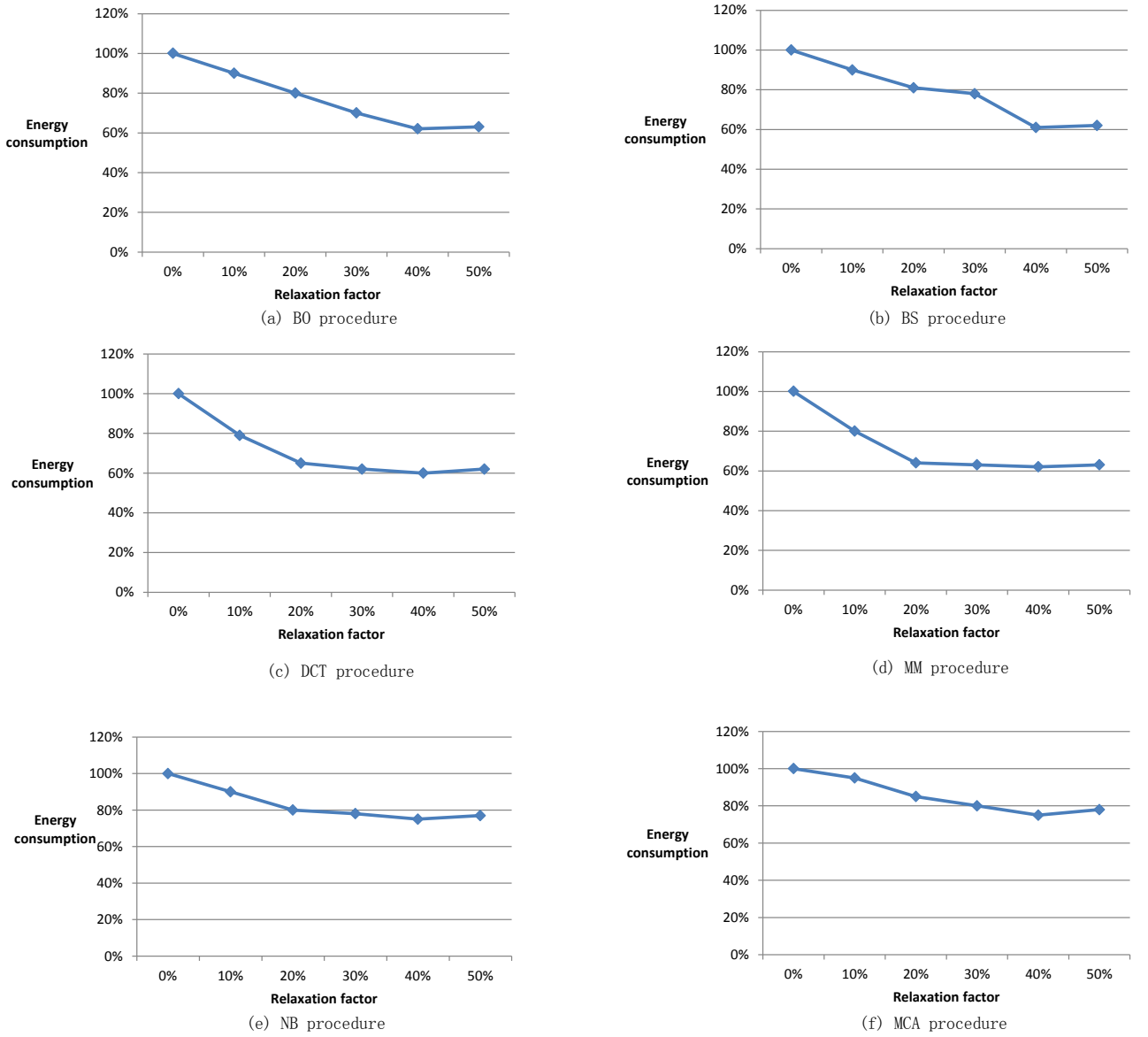


Fig. 5 Energy consumption variations of single kernel program with relaxation factor

tion and the execution time becomes smoother. The figure also shows that the system power consumption can be reduced by effectively applying the performance constraint of relaxation through adjusting the parallel processing of heterogeneous multi-processors. Meanwhile, those programs with different execution characteristics have dissimilar power consumptionCexecution time curves. Compared with computation-intensive programs (including: BO, BS, DCT, and MM), the memory-intensive programs (NB and MCA) have a power consumption which is less sensitive to the operating frequency of the core units, so their power consumption is less optimised, whereas the power consumption and execution time of MM are most sensitive to the operating frequency. When the relaxation factor reaches 25%, the power consumption is reduced by 30%. However, the processors then minimise their frequency so that the power consumption cannot be further decreased when using the DVFS method with an increasing relaxation factor, while, by allocating the tasks intelligently to processors with higher efficiencies, the optimization ratio of the power consumption can be gradually reduced.

After constructing the relationship between the power consumption and execution time of each kernel program, all independent kernel programs are integrated to a multi-kernel program to evaluate the low-power optimization of the optimal descent based time allocation algorithm in multiple computing sections. Fig.6 shows the variation of the total power consumption of the multi-kernel program with the applied time con-

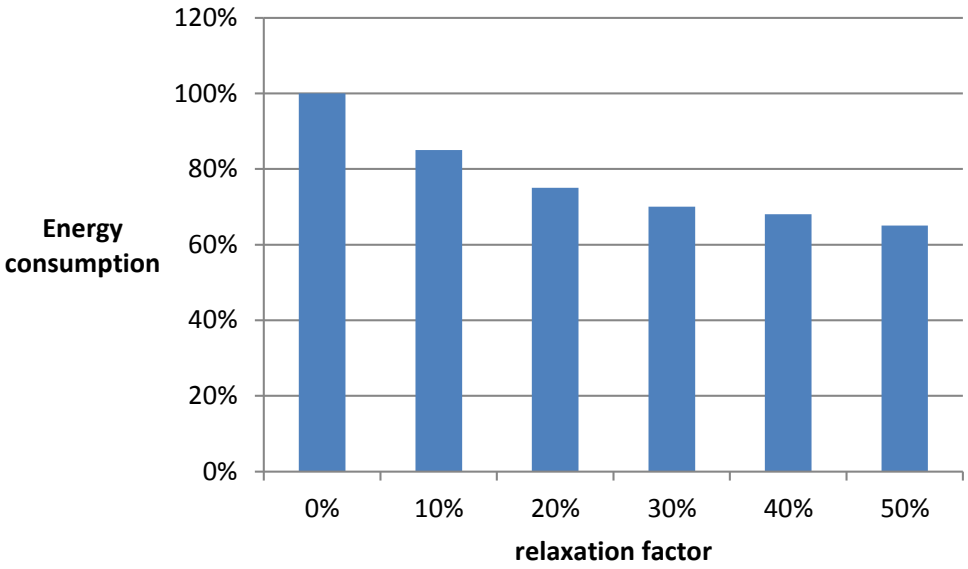


Fig. 6 Low-power optimization results of programs in multiple computing sections

straints. By using the algorithm, the total time constraint is divided into multiple time slices and gradient kernel programs are selected in priority order within the iteration to be allocated to the current time slice. In the test case, the number of time slices is set to match that of the kernels. As can be seen from the figure, low-power optimization in multiple computing sections can be realised by using the algorithm. With a relaxation factor of 30%,power consumption is reduced by 25%. Similar to Figure 5, as the relaxation factor increases beyond 30%, the power consumption cannot be further decreased using the relaxed time constraint due to the minimum processor frequency constraint.

(3) Influences of communication-aware multi-task division on the optimization of system power consumption

Sub-procedure RESID in program MGRID is tested: consisting of nested parallel loops, the sub-procedure is able to perform parallel execution at the data-level. With an iteration space of 256256256,program RESID is divided to sub-iteration spaces measuring 32256256 and mapped to the CPU and GPUs for parallel execution. The comparison of the execution time and power consumption with different modes of division is shown in Fig.7 where the abscissa represents the mode of division and the figures on the two sides of the oblique lines refer to the ratio of tasks allocated to the CPU to those allocated to the GPUs. Division into CPU-only (or GPU-only) mode means that all of the tasks are allocated to the CPU (or the GPUs). The figure makes a comparison under conditions with, and without, considering communication overheads. Not considering the communication overheads is to suppose that the input data of the GPUs are in the GPU storage space already, and the output data need not to be restored to CPU storage space. The experimental results indicate that optimal execution performance is obtained in division mode 25/75 without considering communication overheads. While, when communication overheads are taken into account, division mode 62.5/37.5 is found to bring about an optimal performance in which the execution time is twice that when applying division mode 25/75 without considering communication overheads. It can be seen that, as the performance can be significantly improved by using GPUs, without considering communication overheads, the power consumption reduces with the increasing ratio of tasks allocated to GPUs. While considering the communication overhead, communication inevitably consumes power which increasing with increasing ratio of tasks allocated to GPUs, thus offsetting the efficiency advantage bestowed by the GPUs. Therefore, in the communication perception-based task division and the adjustment of processor frequency, as two effective methods for the low-power optimization of heterogeneous systems, comprehensively determining the optimization spaces of the two methods can achieve an optimal optimization effect. On the basis of obtaining the execution information of all tasks, the low-power optimization method based on integral linear programming can find the optimal solution to problems while providing a benchmark for evaluating dynamic adaptive methods.

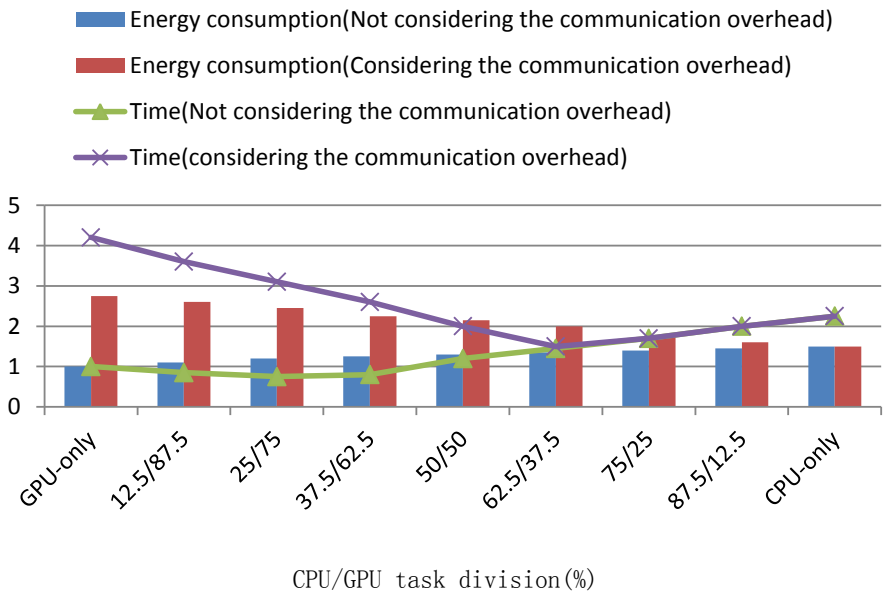


Fig. 7 Influences of energy consumption of communication and task division mode on the system power consumption

9 Conclusions

Low-power optimization of heterogeneous parallel systems is one of the key concepts necessary to make full use of the advantages of this kind of system. Aiming at the general procedure model composed of serial and parallel computing sections, the research considers the influences of communication overheads and task division on system power consumption. On this basis, the authors establish a power model for the whole procedure of heterogeneous parallel systems. By using this model, the selection algorithm for the optimal frequency of processors, optimal descent-based time allocation algorithm in multiple computing sections, and profiling dynamic analysis-based integral linear programming were designed at algorithm-level for programs in homogeneous and heterogeneous computing sections. The experimental results indicate that the proposed algorithms can reduce the power consumption of systems and therefore more efficiently exploit the performance advantage of heterogeneous systems.

Acknowledgement

This work was sponsored by National Natural Science Foundation of China (grant number 61300029,61672168).

References

1. Holmbacka S, Keller J, Eitschberger P, Lilius J. Accurate energy modelling for many-core static schedules. Euromicro International Conference on Parallel Distributed and Network-Based Processing, Turku, 2015:525-532.
2. Zhao Y, Li X, Ju L, Zong Z. Dependency-based energy-efficient scheduling for homogeneous muli-core clusters. IEEE International Conference on Trust, security and privacy in computing and communications. Melbourne, 2013:1299-1306.
3. Jayaseelan R. Application-specific thermal management of computer systems. Singapore: National University of Singapore, 2014.
4. Coskun AK. Efficient thermal management for multiprocessor systems.San Diego, California, USA: University of California, 2014.
5. Tabik S, Villegas A, Zapata EL, Romero, LF. Optimal tilt and orientation maps: a multi-algorithm approach for heterogeneous multicore-GPU system. Journal of Supercomputing, 2013, 66(1):135-147.
6. Qin X, Mishra P. TECS:Temperature-and Energy-constrained scheduling for multicore systems.IEEE Transactions on Parallel and Distributed Systems. 2015, 26(3):868-877.
7. Sreraman N, Govindarajan A. Vectorizing Compiler for Multimedia Extensions. International Journal of Parallel Programming, 2012; 28(4):363-400
8. Andreas K, Sylvain L. Compilation Techniques for Multimedia Processors.International journal of Parallel Programming, 2012;28(4)347-361.
9. Tiwari V, Malik S, Wolfe A, et al. Instruction level power analysis and optimization of software. In VLSI Design, Proceedings of Ninth International Conference .1996:326-328.
10. Chang F, Farkas K, Ranganathan P. Energy-driven statistical profiling: Detecting software hotspots. In Workshop on Power-Aware Computer Systems.2012
11. Landman P. High-level power estimation[C]. In Proceedings of the international symposium on Low power electronics and design. Piscataway, NJ, USA,1996:29-35.
12. Brooks D, Tiwari V, Martonosi M. Wattch: a framework for architectural-level power analysis and optimizations. In Proceedings of the 27th annual international symposium on Computer architecture. New York, USA, 2010:83-94.

13. Chen J, Dubois M, Stenstrom P. Integrating complete-system and user-level performance/power simulators: the SimWattch approach. In Performance Analysis of Systems and Software, IEEE International Symposium,2013:1-10.
14. Che S., Boyer M., Meng J., et al. Rodinia: A benchmark suite for heterogeneous computing. //Proceedings of 2013 IEEE International Symposium on Workload Characterization. 2013:44-54.
15. University of Illinois. Parboil Benchmark suite. http://impact.crhc.illinois.edu/parboil.php.
16. Luk C-K, Hong S, Kim H. Qilin: exploiting parallelism on heterogeneous multiprocessors with adaptive mapping. // Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture. New York, NY, USA, 2012:45-55.
17. Yang C, Wang F, Du Y, et al. Adaptive optimization for petascale heterogeneous CPU/GPU Computing. //Proceedings of IEEE International Conference on Cluster Computing. Los Alamitos, CA, USA, 2012:19-28.
18. Lorch J.R. Operating systems techniques for reducing processor energy consumption. [S.I.]: University of California, Berkeley, 2014.
19. Weiser M, Welch B, Demers A, Shenker S. Scheduling for reduced CPU energy. In Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation, USENIX Association : Monterey, California, 2014(353):13-23.
20. Govil K., Chan E., and Wasserman H. Comparing algorithms for dynamic speed-setting of a low-power CPU. //Proceedings of the 11st Annual International Conference on Mobile Computing and Networking, Berkeley, CA, 2013:13-25.
21. Lorch J.R., Smith A.J. Improving dynamic voltage scaling algorithms with PACE. // Joint International Conference on Measurement and Modeling of Computer Systems,. Cambridge, MA, 2014,29(1):50-61.
22. Seng J S, Tullsen D M. The effect of compiler optimizations on pentium 4 power consumption. Interaction between Compilers and Computer Architecture, Annual Workshop on 2013(0):51-56.
23. Zhu Y, Magklis G, Scott M L. The energy impact of aggressive loop fusion. //Proceedings of the 13 the International Conference on Parallel Architectures and Compilation Techniques. Washinton, DC, USA, 2014:153-164.
24. Kandemir M, Vijaykrishnan N, Irwin M. Influence of compiler optimizations on systems power. //Proceedings of the 37th Annual Design Automation Conference. New York, NY, USA, 2013:304-307.
25. Z. Zong, A. Manzanares, X. Ruan, and X. Qin, Ead and PEBE: Two energy-aware duplication scheduling algorithms for parallel tasks on homogeneous clusters. IEEE Trans. Comput, 2011(3):360-374.
26. P.A. La Fratta. P.M. Kogge. Energy-efficient multithreading for a hierarchical heterogeneous multicore through locality-cognizant thread generation. Parallel Distribution Computation, 2013,73(12):1551-1562.
27. Singh J, Betha S, Mangipudi B, Auluck N. Contention aware energy efficient scheduling on heterogeneous multiprocessors. IEEE Transactions on parallel and distributed systems. 2015,26(5):1251-1264.
28. Chiesi M, Vanzolini L, Mucci C. Power-aware job scheduling on heterogeneous multicore architectures. IEEE Transaction on parallel and distributed systems. 2015,26(3):868-876.
29. Hunold S, Rauber T, Suter F. Redistribution aware two-step scheduling for mixed Cparallel applications . IEEE International conference on cluster computing. 2013:50-58.
30. Dutot P-F, Takpe T, Suter F. Scheduling parallel task graphs on homogeneous multicluster platforms. IEEE Transactions on parallel and distributed systems. 2011,20:940-952.