I.J. Education and Management Engineering **2011, 3, 31-38**
Published Online September 2011 in MECS (http://www.mecs-press.net)
DOI: 10.5815/ijeme.2011.03.05

Available online at http://www.mecs-press.net/ijeme

Performance Analysis for Heterogeneous & Reconfigurable Computing Based on Scheduling

Yiming Tan, Guosun Zeng, Shuixia Hao

Department of Computer Science and Technology, Tongji University, Shanghai, China

Abstract

Right now, heterogeneous & reconfigurable computing is a research hot in the area of high performance computing. Due to the heterogeneity of application tasks and reconfigurability of system architecture, performance analysis for heterogeneous & reconfigurable computing becomes rather difficult. Unfortunately, the existing techniques and methods are no longer suitable for use. This paper presents a performance analysis method based on task scheduling. It builds on system architecture model and task model of heterogeneous & reconfigurable computing. By making use of heterogeneity matching matrix and reconfigurability coupling matrix we achieve optimal selection and matching between computational tasks and processing units. Through task scheduling algorithm, the completion time of application task run on heterogeneous & reconfigurable computing system can be calculated. Finally, we carry out case study.

Index Terms: component; heterogeneous & reconfigurable computing, performance analysis, heterogeneity matching matrix, reconfigurability coupling matrix

1. Introduction

Driven by demand of scientific research and engineering computing, the exploring of high performance computing has never stopped. In the 35th Top500 supercomputer list which was announced on May 31, 2010, the first is "Jaguar" which was developed by Cray Inc. This supercomputer posted a 1.75 petaflop/s performance speed running the Linpack benchmark and the theoretical peak capability reached 2.3 petaflop/s. Homogeneous computing, which was the main computing paradigm of high-performance computing traditionally, had already turned into heterogeneous computing (HC) [1]. For example, the supercomputer "Roadrunner"[2] was the representative of heterogeneous computing system. Obviously, HC has become one of the trends of high performance computing. On the other hand, reconfigurable computing (RC)[3] has been introduced into high performance computing, because it can offers users with high performance and flexibility for a wide range of computationally demanding problems. Thus, heterogeneous & reconfigurable computing

Supported by the 863 program under grant of 2007AA01Z425 and 2009AA012201, 973 program under grant of 2007CB316502, the National Natural Science Foundation of China under grant of 90718015，the joint of NSFC and Microsoft Asia Research under grant of 60970155, Shanghai Outstanding academic leaders program under grant of 10XD1404400, the State Key Laboratory of High-end Server & Storage Technology under grant of 2009HSSA06.

(HRC)[4], which employs various hardware accelerators and reconfigurable units, will be the new trend of high performance computing. HRC has all the advantages of HC and RC, such as, efficiency, flexibility, cost-effective, low power and stability etc. As a new computing paradigm, performance analysis for heterogeneous & reconfigurable Computing becomes very difficult, because of the heterogeneity of tasks and reconfigurability of architecture. However, performance analysis is the key step of selecting proper type and quantity processing unit, reconfiguring a appropriate architecture and predicting the performance bottleneck. Thus, this paper presents a performance analysis method for HRC

2. The basic concept of Heterogeneous & Reconfigurable Computing

Heterogeneous & reconfigurable computing (HRC) is developed on the basis of heterogeneous computing (HC) and reconfigurable computing (RC), that is, HRC is a combination of HC and RC. The definitions of HRC, HC and RC are the following.

Definition 1: Heterogeneous Computing (HC) is a computing environment, which employs various performance and functionality processing units (such as PC, vector, SIMD, MIMD, FPGA, GPU, DSP and CELL etc.) and these processing units are connected by high-speed interconnection. This computing environment makes good use of the heterogeneity of tasks and processing units to execute an application collaboratively. The goal is to minimize the execution time. Figuratively speaking, HC seems like a symphony orchestra uses various musical instruments to play a wonderful music.

Definition 2: Reconfigurable Computing (RC) is a variable computing platform, which can rewrite the reconfigurable units at a proper timing according to the characteristics of application tasks under the control of software. The RC system can reach or approach the performance of specific processing unit. The reuse of processing unit makes RC system high performance, flexibility, stability and low power.

Definition 3: Heterogeneous & Reconfigurable Computing (HRC) is a hybrid computing paradigm, which combines all these characteristics of HC and RC.

HRC makes use of these advantages of HC and RC to achieve high performance computing. The main idea of HRC is to select the CPU, FPGA, GPU, Cell, DSP and other types of processing unit flexibly according to the characteristic of tasks and rewrite the reconfigurable units to create a new computing platform. The new computing platform can be created through rewriting these reconfigurable logics and rebuilding interconnection.

2.1 Heterogeneous & Reconfigurable Computing System Model

Heterogeneous & reconfigurable computing systems (HRCS) is a new high performance computing system, which may contain general-purpose processor (CPU), application-specific integrated circuit (ASIC) and reconfigurable unit (FPGA). In order to carry out performance analysis, it needs modeling HRCS system.

Definition 4: Heterogeneous & Reconfigurable Computing Systems (HRCS) can be defined as a two-tuples HRCS = (V_P, E_P). VP = {p_1, p_2, … , p_M } represents a set of processing units, p_i represents the i-th computing units. EP = {e_1, e_2, … , e_L} represents a set of interconnection structure.

2.2 Heterogeneous & Reconfigurable DAG Model

In order to carry out performance analysis, besides the HRCS system model, it also need establish task model. Here, we use heterogeneous & reconfigurable DAG (HR-DAG) to depict the application.

Definition 5: Heterogeneous & reconfigurable DAG (HR-DAG) can be defined as a six-tuples HR-DAG = (V_T, E_T, W, D, H, R). V_T represents a set of tasks. E_T represents a set of edges and it also depicts a partially ordered set, that is, $E_T \in V_T \times V_T$. W represents a set of computation workload of task. D represents a set of communication workload between two tasks. H represents a set of heterogeneous characteristics of task. R represents a set of reconfigurable characteristics between two tasks.

The heterogeneous characteristic $h_i \in H$ is a description of the task type, computing needs or execution model of a task. For example, according to the parallelism, various tasks can be divided into SIMD, MIMD, vector, superscalar, pipeline, dataflow and so on[1].

The reconfigurable characteristic $r_i \in R$ is a description of the requirement about communication mode(such as point to point, point to many, many to many) or topology(such as linear Array, Mesh-Connected, Tree-Connected and Cross-Bar) between tasks. These reconfigurable characteristics are intelligent decision-making of reconfigurable logics.

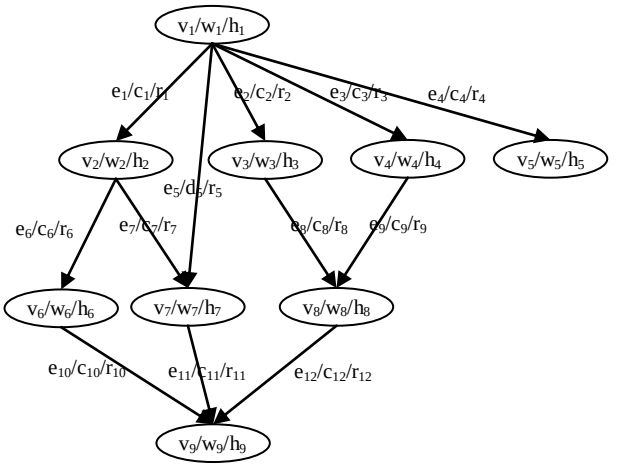

Fig 1. HR-DAG

Every program can be described by HR-DAG. Fig. 1 is a HR-DAG, these parameters on nodes are task number/ computation workload/heterogeneous characteristic and these parameters on edges are partially ordered/communication workload/reconfigurable characteristic.

3. Heterogeneous & Reconfigurable Computing Performance Measurement Matrix

3.1 Heterogeneity Matching Matrix

In the heterogeneous & reconfigurable computing (HRC), whether it can achieve high performance, not only depend on execution speed of processing unit and the quality of parallel algorithms, but also depend on the matching degree between task and processing unit. The latter is particularly important sometimes. For example, control codes can not perform effectively on FPGA and vector tasks may perform poor on non-vector machines. In order to describe the optimal matching between tasks and processing units, we propose the parameter about matching degree [6].

Definition 6: Heterogeneity Matching Matrix (Ma) depicts the matching degree between various tasks and different processing units. Here, $Ma=(v_{ix})N\times M$, v_{ix} represents the execution speed of task t_i running on processing unit p_x, $1 \le i \le N$, $1 \le x \le M$.

$$M_a = \begin{array}{c} \\ t_1 \\ t_2 \\ \vdots \\ t_N \end{array} \begin{array}{c} \begin{array}{cccc} p_1 & p_2 & \cdots & p_M \end{array} \\ \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1M} \\ v_{21} & v_{22} & & \\ \vdots & & \vdots & \\ v_{N1} & & & v_{NM} \end{pmatrix} \end{array}$$

Fig 2. Heterogeneity Matching Matrix (Ma)

Fig. 2 is heterogeneity matching matrix (Ma). By means of physical measurements, the execution speed v_{ix} is determined.

According to heterogeneity matching matrix (Ma), it can achieve optimal matching between tasks and processing units. For example, when task t_i need to be mapped or scheduled, the processing unit p_x which has the maximum execution speed v_{ix} will be the target component. Next, the actual execution time of ti can be calculated according to the (1).

$$T_{comp}(t_i) = \frac{w_i}{v_{ix}} . \qquad (1)$$

Here, Let w_i denote the computation workload of task t_i, $1 \le i \le N$, $1 \le x \le M$. The optimal matching between tasks and processing units reflects the efficiency of heterogeneous & reconfigurable computing (HRC).

3.2 Reconfigurability Coupling Matrix

Interconnection is another important factor which affects the performance of heterogeneous & reconfigurable computing (HRC) performance. Different interconnections have different communication capability and execution mode, which can be denoted by coupling degree. Generally speaking, the greater coupling degree the smaller communication overhead [3].

In heterogeneous & reconfigurable computing system (HRCS), the interconnection may be reconfigured according to the task characteristic. It can make the interconnect resources used effectively and reduce the communication overhead.

After interconnection reconfiguration, communication ability will change. In order to describe the impact of interconnection on the communication capability, we propose the parameter about coupling degree.

Definition 6: Reconfigurability Coupling Matrix Co depicts the coupling degree between communication characteristic and interconnection. Here, $C_o = (c_{i,j})_{K \times L}$, $1 \le i \le K$, $1 \le j \le L$. $c_{i,j}$ denotes the coupling degree between communication characteristic and interconnection, $0 \le c_{i,j} \le 1$.

$$C_o = \begin{array}{c} \\ r_1 \\ r_2 \\ \vdots \\ r_K \end{array} \begin{array}{c} \begin{array}{cccc} t_1 & t_2 & \cdots & t_L \end{array} \\ \begin{pmatrix} c_{11} & c_{12} & \cdots & c_{1L} \\ c_{21} & c_{22} & \cdots & c_{2L} \\ \vdots & \vdots & \vdots & \vdots \\ c_{K1} & c_{K2} & \cdots & c_{KL} \end{pmatrix} \end{array}$$

Fig 3. Reconfigurability Coupling Matrix Co

Fig. 3 is reconfigurability coupling matrix Co. The coupling degree $c_{i,j}$ can be determined by measuring the performance of task execution on different interconnection.

Let d_{ab} denote the communication workload between task t_a and t_b, B the bandwidth, the actual communication time between task t_a and t_b can be calculated according to the (2).

$$T_{comm}(t_a, t_b)=\frac{d_{ab}}{B \cdot c_{i,j}} . \quad (2)$$

Here, $1 \le a, b \le N$. The reconfiguration of interconnection reflects the flexibility and adaptability of heterogeneous & reconfigurable computing (HRC).

4. Performance Analysis Method Based on Scheduling

Completion time is the key metric to evaluate the performance of heterogeneous & reconfigurable computing system (HRCS). However, because of the heterogeneity of tasks and reconfigurability of architecture, it is difficult to create a mathematical formula about completion time. So, we try to calculate the completion time by a scheduling algorithm.

4.1 Parameters Assumption

Performance analysis for heterogeneous & reconfigurable tasks running on the HRCS is the most common and difficult condition. The main idea of this performance analysis method is to match the application tasks and processing units optimally based on heterogeneity matching matrix Ma and calcualte the executiong time of task according to (1). Next, the interconnection will be reconfigured according to reconfigurability coupling matrix Co and the communication time between tasks can be calcualted according to (2). Last, the total completion time can be calculated. Table 1 lists the parameters and definitions used in the scheduling algorithm.

Table 1 Parameters and Definitions

parameter	definition
M	number of processing units
N	number of tasks
Ma	heterogeneity matching matrix
Co	reconfigurability coupling matrix
wa, $1 \le a \le N$	computation workload of task a
dab, $1 \le a, b \le N$	communication workload between task a and task b
B	bandwidth
parents(a), $1 \le a \le N$	list of direct predecessors for each task a

4.2 MaCo-based Heterogeneous & Reconfigurable Performance Analysis(MaCo-HRPA)

Algorithm: MaCo-HRPA
Input: All inputs listed in table 1
Output: Total_ completion _time
{
L1: $t_s(a)=0$， $t_e(a)=0$
L2: $\rho \leftarrow \{1, 2, \ldots, M\}$
L3: $\varepsilon \leftarrow \{a|\ \text{indegree}(a)=0, 1 \le a \le N\}$

L4: Do until ε empty {
L5: (1) For each executable task a∈ε
L6: x=scheduling (a, ρ, Ma)
L7: if x > 0
L8: ε ← ε - { a }
L9: ρ ←ρ - {x}
L10: v_{ax} = select_v(a, x, Ma)
L11: $T_{comp}(a)=\frac{w_a}{v_{ax}}$
L12: $t_e(a) \leftarrow t_s(a) + T_{comp}(a)$
L13: ρ ←ρ + {x}
L14: (2) For each immediate successor b of task a
L15: r_i ←HR-DAG (a , b)
L16: t_j ←select_c(r_i, Co)
L17: reconfiguring (t_j)
L18: $T_{comm}(a,b)=\frac{d_{ab}}{B \cdot c_{i,j}}$
L19: $t_s(b) \leftarrow t_e(a) + T_{comm}(a)$
L20: indegree(b) ← indegree(b) –1
L21: if indegree(b)=0
L22: ε ← ε + { b}
L23:
}
L24: Total_ completion _time ← max{ $t_e(b)$, 1≤ b ≤ N }

5. Case Study

In order to validate the accuracy of the performance analysis method, we create an experiment platform. This platform has one dual-core 2.4 GHz AMD Opteron CPU, two NVIDIA Quadro FX 5600 GPU cards (each has 1.5 GB of on-board memory), two Nallatech H101-PCIX FPGA accelerator card (with 16 MB of SRAM and 512 MB of SDRAM), and InfiniBand and Ethernet PCIe network cards. We design a set of microbenchmarks, it consists of matrix multiply (MM), DES, N-body (PP algorithm), SVM, 2-DFFT. Fig. 4 compares the predicted completion time with the measured completion time for these microbenchmarks.

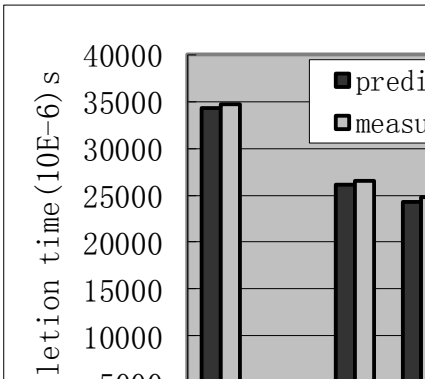

Fig 4. Comparision between predicted value and measured value

According to the results, the geometric mean of the error in the execution time prediction for microbenchmarks is 4.1%. Because of this performance analysis method neglect the tasks' start-up time, reconfiguration time and other system overhead, the measured times are all bigger than the predicted times. We find that DES achieves the optimal performance on FPGA, even the execution time is 4 orders of magnitude smaller than GPU. It shows that the optimal matching is important to performance.

6. Raletd work

Traditionally, performance analysis techniques for high performance computing can be classified into three categories: analytical modeling, measurement and simulation [7-9]. Adve, etc [10]developed a conceptually simple modeling technique called deterministic task graph analysis that provides detailed performance prediction for shared-memory programs with arbitrary task graphs, a wide variety of task scheduling policies, and significant communication and resource contention. This model assumed deterministic task execution times (while retaining the use of stochastic models for communication and resource contention). Smith, etc [11] developed an analytic modeling methodology for characterizing the performance of applications running on high performance reconfigurable computing (HPRC) resources. This methodology included the impact of RC hardware, communication, application load imbalance, background load, and heterogeneous resources. The model differed from earlier pure software models in that it accounts for the hardware execution time and related hardware overhead. Javadi, etc [12] presented a model based on queuing networks and probabilistic analysis to predict the average message latency of super-cluster systems with network heterogeneity. The analysis was based on a parametric family of fat-trees, the m-port n-tree, and a deterministic routing algorithm was proposed.

7. Conclusion

This paper presented the definitions about heterogeneous computing (HC), reconfigurable computing (RC) and heterogeneous & reconfigurable computing (HRC) and discussed the relationship among them. We proposed the heterogeneous & reconfigurable computing system (HRCS) model and heterogeneous & reconfigurable DAG (HR-DAG) model. HR-DAG extended the traditional DAG model, it was able to reflect the heterogeneity of tasks and reconfigurability of architecture. Heterogeneity matching matrix Ma was proposed to depict the matching degree between various tasks and different processing units, reconfigurability coupling matrix Co depicted the coupling degree when these processing units execute different tasks. We designed a heterogeneous & reconfigurable performance analysis method based on scheduling algorithm--MaCo-HRPA, which took into account the heterogeneity and reconfigurability simultaneously and more applicable than these previous techniques. In the future, we plan to research the generation of reconfigurability coupling matrix Co, the interaction between communication characteristic and interconnection and the quantitative study about coupling degree.

References

[1] A.A. Khokhar, V.K. Prasanna, M.E. Shaaban. Heterogeneous computing: Challenges and opportunities. IEEE Computer, 1993, 26(6):18–27.
[2] K.J.Barker, K.Davis, A.Hoisie. Entering the petaflop era: the architecture and performance of Roadrunner. Proceedings of the ACM/IEEE conference on Supercomputing, 2008, pp15-21.
[3] K.Compton , S.Hauck. Reconfigurable Computing: A Survey of Systems and Software. ACM Computing Surveys, 2002, 34(2):171-210.
[4] F. Thoma, M.Kuhnle, P. Bonnot. MORPHEUS: Heterogeneous Reconfigurable Computing. Proceedings of 17th International Conference on Field Programmable Logic and Applications (FPL07), 2007, pp409–414.

[5] Zeng Guosun, Lu Xinda. Automatically Extracting The SIMD/MIMD Heterogeneity Hiding A Program [J].Journal of Computer Research & Development, 2000, 37(11):1397-1403(in Chinese)
[6] V.E.Bazterraa, M.Cumaa, M.B.Ferraroa. A general Framework to Understand Parallel Performance in Heterogeneous, Clusters: Analysis of A New adaptive Parallel Genetic Algorithm. Journal of Parallel and Distributed Computing, 2005, 65(1): 48-57.
[7] Yuan Lulai, Zeng Guosun. Dynamic Level Scheduling Based on Trust Model in Grid Computing [J]. Chinese Journal of Computers, 2006, 29(7): 1217-1224(in Chinese)
[8] A.Pombortsis, E.Papaefstathiou, A.Veglis. PASE: A performance analysis simulation environment. Simulation Practice and Theory, 1994, 2(1):43-59.
[9] J.Dongarra, A.Malony, S.MooreS. Performance Instrumentation and Measurement for Terascale Systems. Proceedings of the ICCS Conference (LNCS 2660), 2003, pp53–62.
[10] V.S.Adve, M.K.Vernon. Parallel program performance prediction using deterministic task graph analysis. ACM Transactions on Computer Systems, 2004, 22 (1): 94-136.
[11] M.C.Smith , G.D.Peterson. Parallel application performance on shared high performance reconfigurable computing resources. Performance Evaluation, 2005, 60(1-4):107-125.
[12] B.Javadi, M.K.Akbari, J.H.Abawajy. A performance model for analysis of heterogeneous multi-cluster systems. Journal of Parallel Computing, 2006, 32 (11-12):831-851.