

# 与体系结构匹配的多级可重构任务划分方法

郝水侠<sup>1,2</sup> 曾国荪<sup>2</sup>

(江苏师范大学数学与统计学院 徐州 221116)<sup>1</sup> (同济大学计算机科学与技术系 上海 201804)<sup>2</sup>

**摘要** 异构系统是高性能计算发展的主要模式,云计算是异构计算的典型实例。其优势在于异构处理器能各尽其能,但在实际应用中异构系统的性能往往不能充分发挥,因为处理器特征与应用程序特征不匹配,造成系统效率低下。因此借助重构思想,提出与体系结构结合的多级可重构任务划分方法。定义了多级可重构的概念,分析了异构匹配的原理,给出异构特征分析过程,提出了基于异构特征匹配的多级可重构任务划分方法。最后通过仿真实验说明,与体系结构匹配的划分方法适合当前的异构系统。

**关键词** 计算任务,体系结构,多级可重构,任务划分

**中图法分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2014.08.013

## Multi\_reconfigurable Task Partitioning Method on Architecture Matching

HAO Shui-xia<sup>1,2</sup> ZENG Guo-sun<sup>2</sup>

(School of Mathematics and Statistics, Jiangsu Normal University, Xuzhou 221116, China)<sup>1</sup>

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)<sup>2</sup>

**Abstract** Heterogeneous systems have been the main model of high-performance computing currently, such as cloud platform. Its advantage is that heterogeneous system is feasible. Yet its performance cannot be fully utilized due to not matching between tasks executed and its architecture. So using reconfigurable ideas, the paper provided multi\_reconfigurable partitioning method combining architecture. Firstly the paper defined the multi-level reconfiguration concept. Secondly the paper analysed heterogeneous matching theory and a process of heterogeneous feature, further proposed heterogeneous multi-level task partitioning method. And finally the paper verified the effectiveness of the method.

**Keywords** Computing task, Architecture, Multilevel reconfiguration, Task partitioning

## 1 引言

Freund<sup>[1]</sup>1989年提出了异构优化理论,开创了异构计算研究的先河,异构计算已成为高性能计算的典型模式。在计算系统中,异构性无处不在,如计算机硬件平台异构、通信网络异构、计算机能力异构、应用程序和服务异构等等。异构计算是应用任务存在异构和计算机系统存在异构的必然产物<sup>[2]</sup>。异构计算只有当异构部件适合应用任务时才能发挥其效能,实际中往往存在分配的异构系统中的应用任务与体系结构不匹配,从而造成异构部件的浪费。可重构计算为异构计算中应用任务和体系结构异构特征不匹配提供了实现匹配的可能,它是高性能计算研究的另一个热点。可重构计算是根据应用的不同对结构固定的硬件计算平台进行配置从而完成相应计算任务的新型计算,是结合了应用编程能力的系统<sup>[3]</sup>。它不仅能够在硬件器件制造后针对计算任务进行定制,而且能够为计算任务提供大量的可定制的执行空间<sup>[4]</sup>。当应用任务不符合体系结构异构特征或者二者异构特征不匹

配时,可借助可重构计算<sup>[5]</sup>的能力,根据应用需求对结构及数据通路进行重构设计,或者可以根据体系结构特征或者应用任务异构特征来进行定制,使应用任务和体系结构异构特征匹配,进而实现应用任务在体系结构上执行的高效能。

使异构可重构系统中计算任务和体系结构实现高效能的关键是实现计算任务和体系结构匹配,这包括两个层面,一是计算任务类型和处理器处理性能一致;另外一个则是计算任务的结构特征和体系结构的结构特征一致。当计算任务个数远远大于处理器个数时,首先对计算任务划分和合并,通过有目标的组合,使计算任务和体系结构最大可能地匹配。并行计算任务划分往往转换成图的划分问题,常用的图划分方法有等分划分方法<sup>[6]</sup>、不等分划分方法<sup>[7]</sup>和多级超图划分方法<sup>[8]</sup>等。可重构计算的任务划分方法有:以通信代价为最小的时序多路划分方法<sup>[9]</sup>、根据硬件面积及约束条件来调整划分粒度的柔性粒度软硬件划分方法<sup>[12]</sup>。与传统同构环境划分相比,在异构系统中,任务划分必须与体系结构相结合才有实际意义。在目前划分文献中,考虑体系结构特征的较少,但是并

到稿日期:2013-05-20 返修日期:2013-07-20 本文受 863 项目(2007AA01Z425,2009AA012201),973 课题(2007CB316502),国家自然科学基金项目(90718015),NSFC-微软亚洲研究院联合资助项目(60970155),教育部博士点基金项目(20090072110035),上海市优秀学科带头人计划项目(10XD1404400),高效能服务器和存储技术国家重点实验室开放基金项目(2009HSSA06),校重点科研项目(13XLA11),院重点专业类经费教研项目(STYJYA1401)资助。

郝水侠(1973—),女,博士生,副教授,主要研究领域为异构计算、可重构计算,E-mail: sxhaotj@gmail.com;曾国荪(1964—),男,博士,教授,主要研究领域为异构计算、信息安全。

行任务分配的一些方法值得借鉴。如 Ma、Chen 和 Chung 研究了处理器同构网络异构系统下任务的最优分配算法,即通过遍历所有可能解的状态空间树来得到问题的最优解<sup>[14]</sup>。Snear 和 Ripoll 研究了处理器同构网络异构的并行系统中任务分配的启发式算法,算法分两个阶段,第一阶段是通过聚类 and 任务重构划分对原任务图进行收缩,第二阶段利用收缩后的任务图将任务映射到目标系统。结果表明该算法能很好平衡计算复杂性和映射质量<sup>[15]</sup>。在异构系统中的任务划分方法中,Fukushi 考虑了器件特征,但是没有考虑系统结构特征<sup>[13]</sup>。Verbauwhede 等指出应用和体系结构结合的可重构技术是下一代可重构计算发展的趋势<sup>[10]</sup>。我们也曾提出计算任务和体系结构结合的思想,但对于机器特征和结构特征描述还不够深入<sup>[11]</sup>。因此本文在研究异构特征的基础上,提出一种基于多级可重构的异构特征匹配的任务划分方法。

## 2 可重构概念

重构过程指事物调整结构的过程,目前重构已经应用在计算技术领域中,可重构计算为实现任务和系统的异构特征提供了匹配的可能性,重构性为充分发挥系统异构特征提供了空间。下面给出可重构计算、异构可重构的定义,并提出多级重构的概念。

**定义 1(可重构计算<sup>[18]</sup>)** 指在软件的控制下,利用系统中的可重用、可编程资源和器件,根据应用需求重新构造一个新的计算平台,以达到接近专用硬件设计的高性能。

可重构计算的本质是利用可编程器件多次重新配置逻辑单元的功能和互连的特性,使系统兼具灵活性、高可靠、低功耗、低成本、易于升级等多种优良特性。具有可重构计算特征的系统称为可重构计算系统。

**定义 2(异构可重构计算<sup>[18]</sup>)** 是指在并行计算系统中,为了实现高性能计算,挖掘应用程序中异构特征,通过整合通用处理器、可重构部件和加速部件的新型高性能计算机系统实现应用执行的高效能。

重构过程不仅可以在处理器内部重构,也可以在计算节点内重构等等,因此我们提出多级可重构概念。

**定义 3(多级重构)** 指计算系统实现处理机重构、计算节点内重构、计算节点间重构三级重构过程。这 3 个过程具体定义为:(1)处理器本身重构,也称为门级重构、电路级重构,主要是指以 FPGA 为主要的可重构器件,根据不同的任务需求进行不同的配置从而改变 FPGA 的内部结构以实现不同的功能;(2)计算节点内可重构,即每个计算节点使用的处理器类型根据应用算法的功能来选择,处理器的数量根据应用算法的规模确定,处理器之间的协作关系(并行或流水)根据应用算法的流程确定;(3)计算节点间可重构,例如根据应用问题的功能、规模和结构,调整确定计算节点的形态、计算组件类型和数量以及计算节点的连接关系等。

在可重构计算系统中,硬件的结构配置信息可以像软件程序一样被动态调用或修改。通过重构计算,我们一方面可以提高计算机的性能,使之能适应不同的要求;另一方面可以节省软硬件的开发费用,尽量使用已有的资源来构造新的系统,减少不必要的浪费。

## 3 异构匹配原理及异构特征分析

可重构过程实际上是开发系统的重构并行性。它是在并

行性概念中引入可重构因素,利用体系结构的可重构特征,通过重建、重组、重配置、自适应、自定制来调整体系结构适应异构计算任务的需求。异构计算将问题分割成具有不同异构特征的子任务,然后将它们送往能与其有最佳匹配的处理部件上进行计算,从而能以最短的时间来完成对问题的求解。异构计算的任务执行建立在系统重构的基础之上。

### 3.1 异构匹配原理

异构计算的目的是使用系统中最合适机器执行的串行代码部分,这样就可消除串行瓶颈,并获得超线性加速比,驱散了 Amdahl 定理给并行计算笼罩的阴影。用一个经典的例子来说明异构匹配优化原理,以便实现异构计算对高效能计算的支撑作用。如图 1 所示,假设有一个计算任务,可以用 CPU 串行程序实现,也可用 GPU 设计实现,或者用异构可重构系统设计实现,在此选用 4 种常见类型 vector、MIMD、FPGA 和 GPU。假设其在串行机上运行需 100 个时间单位,而在 GPU 机器上执行时,程序适合 GPU 执行的部分代码就可加速运行,程序的其它部分也会相对提高执行速度,执行时间缩短为 60 个时间单位。再进一步,如果相应的并行程序部分都在相匹配的异构计算部件上运行,执行时间缩短为 4 个单位时间,既大大缩短了程序执行的时间,也可相应地提高加速比,从而提高异构系统的效能。

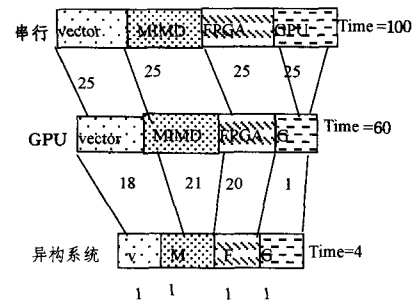


图 1 异构匹配实例

另外在算法设计中,常有利用结构特征提高算法性能的情况,例如矩阵-向量乘法,设计成串行矩阵-向量乘法,其运算时间为  $O(n^2)$ ;设计成带状划分的矩阵-向量乘法,当处理器个数和划分带数相同时,在超立方体和网孔结构上的运行时间为  $O(n)$ ;设计成棋盘划分的矩阵-向量乘法,在超立方体上的时间是  $O(n)$ ,而在网孔上的时间是  $O(\log n)$ 。根据前面的分析,在网孔上用同样多的处理器,棋盘划分的矩阵-向量程序比带状划分时要快。在并行算法调度过程中也有这样的经验,当并行算法按系统结构进行设计时,算法的时间复杂度会大大减小,机器的效率会大大提高。

因此,通过体系结构重构,可以实现异构优化匹配:(1)彰显个性,各尽所能,优势互补;(2)克服应用程序中的串行瓶颈;(3)整个系统的效能最优。

### 3.2 异构特征分析

在异构可重构计算中,异构特征指机器的性能特征及体系结构互联特征。异构特征分析指根据机器异构特征分析计算任务代码异构特征的过程。

首先是代码的异构特征分析。应用程序的异构特征可分为密集型特征、结构特征、粒度特征、能耗特征、性能特征等。这里主要分析代码的密集型特征,根据计算代码的特点,可以将其划分为适合 CPU 的、适合 FPGA 的、适合 GPU 的等程序类型,如图 2 所示。划分时以计算特征为主要衡量标准。计算指进行算术运算操作、关系运算操作、指令和逻辑运算操

作。计算特征是指应用程序中消耗 CPU 资源来完成算术运算的操作指令、关系运算的操作指令和逻辑运算的操作指令的特点。CPU 类型指应用程序中不可避免地存在一些控制相对复杂、计算相对简单以及无法用硬件综合的计算任务。FPGA 类型指执行应用程序中控制相对简单、数据访问比较规则、计算量相对较大的那些“计算密集型”任务。GPU 类型主要处理图形任务。

```

Program Example
S1 Read i,j;/*适合 CPU*/
s2:for i,j<10*5;
s3:a=i+j;
s4:a=i*j;
endfor;
} /*适合 FPGA*/
S6 ...../*适合 GPU*/
S7 ...../*适合 CELL*/
S8 Write a,b;/*适合 CPU*/
S9 End
    
```

图 2 代码的异构特征

其次是计算任务与体系结构图的结构特征分析。图的结构特征主要有平均度、直径、平均距离和等分带宽。平均度指节点的度之和除以节点的个数(有向图指平均入度和出度之和)。图的结构特征记为  $S = s(f_1, f_2, f_3, f_4)$ , 4 个参数分别代表平均度、直径、平均距离和等分带宽。直径指任何两点之间通信距离最大的值, 或者可以说是任意两点之间最短路径的最大值。平均距离指所有节点对之间距离的平均值。等分带宽指网络中存在通道的最小距离。如果去掉这些通道, 则网络分成两个相等的非连接的节点集合。

### 3.3 具有异构特征的计算任务与体系结构形式化定义

为了实现异构计算任务与体系结构匹配, 先给出计算任务和体系结构的形式化定义。

**定义 4(计算任务)** 定义为一个有向无环图 DAG, 用五元组表示, 记为  $CT = (V, E, H, W, C)$ , 其中:  $V = \{v_1, v_2, \dots, v_n\}$  是子任务节点集合, 节点  $v_i \in V$  代表计算任务中一个子任务。  $E = \{e_1, e_2, \dots, e_m\}$  是子任务节点间有向边集合,  $E = V \times V, e_k = v_i v_j \in E$  表示子任务  $i$  和子任务  $j$  之间存在依赖关系, 子任务  $v_j$  必须等待任务  $v_i$  完成后才能开始执行。  $H = \{h_1, h_2, \dots, h_n\}$  表示子任务的异构代码特征集合,  $h_i \in \{1, 2, 3, 4\}$ , 1 代表适合通用 CPU 的代码, 2 代表适合 FPGA 的代码, 3 代表适合 GPU 的代码, 4 代表适合 DSP 的代码。  $W = \{\omega_1, \omega_2, \dots, \omega_n\}$  是计算任务的计算量,  $\omega_i$  表示子任务  $v_i$  的计算量。  $C = (c_{ij})_{n \times n}$  是有向边的通信量矩阵,  $c_{ij}$  表示子任务  $v_i$  和子任务  $v_j$  的数据链路通信量, 如图 3 所示。

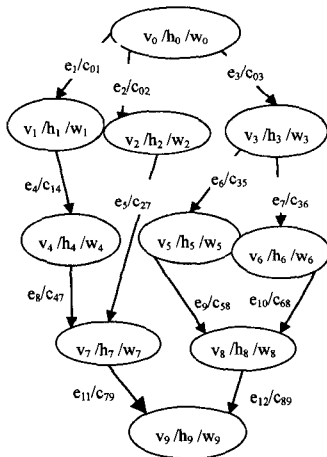


图 3 计算任务

体系结构指包括了一组部件及部件之间的关系, 具体来说指整合了通用处理器、可重构部件和各种加速部件并通过一定的拓扑结构将其连接起来的计算机系统。这里的体系结构的每一个处理单元由一个处理器和本地存储单元组成。处理单元不共享内存, 通信仅靠消息传递。具体定义如下:

**定义 5(体系结构)** 定义为一个带权无向图, 用五元组表示, 记为  $AG = (P, L, H, O, Q)$ , 其中:  $P = \{p_1, p_2, \dots, p_n\}$  是处理器的集合, 节点  $p_i \in P$  代表处理器集中的一个处理器。  $L = \{l_1, l_2, \dots, l_m\}$  是处理器通信的集合,  $L = P \times P, l_k = \overline{p_i p_j} \in L$  表示处理器  $p_i$  和处理器  $p_j$  之间有直接的通信关系;  $H = \{h_1, h_2, \dots, h_n\}$  表示处理器类型的特征集合,  $h_i \in \{1, 2, 3, 4\}$ , 1 代表通用 CPU 器件, 2 代表 FPGA 器件, 3 代表 GPU 器件, 4 代表 DSP 器件;  $O = \{o_1, o_2, \dots, o_n\}$  是处理器单位计算量的计算代价,  $o_i$  为处理器  $p_i$  单位计算量的处理代价。  $Q = (q_{ij})_{n \times n}$  为通信量矩阵, 表示处理器之间单位时间内传输的通信开销, 如图 4 所示。

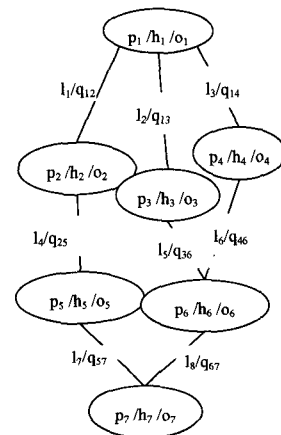


图 4 体系结构

## 4 多级可重构任务划分方法

多级可重构任务划分指在分析计算任务和体系结构特征的基础上, 通过分析计算任务和体系结构异构特征, 逐级划分, 逐级重构, 最终实现并行任务和体系结构最佳匹配。划分时不仅仅考虑应用问题的功能、规模和结构, 也考虑了计算节点的形态、计算组件类型和数量以及计算节点的连接关系等, 在划分时通过适当重构体系结构来实现划分后子任务量和系统负载均衡, 从而达到效能最佳。

### 4.1 多级重构过程

异构特征客观存在于计算任务和体系结构中, 通过挖掘计算任务和体系结构特征, 借助多级可重构实现计算任务和体系结构的最佳匹配, 达到系统计算和通信的负载均衡。根据多级重构的定义, 体系结构可以进行三级重构。一级重构是代码级重构, 根据程序代码的异构特征, 对器件部件内部进行重构, 即通过增加和删除器件类型, 使代码特征和器件类型完全匹配。这个过程定义为  $Recf1()$ 。二级重构是异构特征类型和大小级重构, 通过统计计算程序代码的大小多少, 重新配置器件类型及其个数, 使硬件的类型、个数与 HR-DAG 的节点的类型和计算量完全匹配。这个过程定义为  $Recf2()$ 。三级重构是结构互联级重构, 根据 CT 图, 重新连接体系结构互联方式, 使体系结构和计算任务的结构特征一致, 这个过程

定义为  $Recf3()$ 。

通过多级重构使得计算任务和体系结构的异构特征相同,这是划分时的三级标准。在划分时,除了这三级标准外,还需要考虑计算任务和体系结构的节点权值和边权值成比例,达到计算负载和通信负载均衡,从而实现计算任务和体系结构完全匹配。

#### 4.2 多级可重构任务划分算法(MRA)

多级重构的任务划分方法,是在多级重构匹配基础上实现自适应的任务划分。其思想是通过逐步划分和重构的迭代过程,实现计算任务的划分,使得划分后在体系结构上的执行时间最短。这个阶段我们假定计算任务和体系结构的异构类型一致,则计算任务直接用  $CT$  表示。

划分时以体系结构的节点个数划分为块的目标数,体系结构处理器计算能力和通信能力作为对应一个任务划分对象的目标值,这样可以保证计算负载和通信负载均衡,从而实现异构匹配。由于体系结构和计算任务图量纲不一致,因此按照等比例的方法保证静态异构负载均衡。当子任务的计算量及通信量分别与对应体系结构处理能力和通信能力成比例时,就认为它们两者匹配。根据体系结构的处理器能力和通信能力来划分计算任务,并与体系结构异构特征比较,若一致,则划分结束,否则对其体系结构进行重构,重新进行划分,直至划分后两者异构特征一致。

多级重构划分算法(Multi-Level Reconfigure Partitioning algorithms)

输入:计算任务  $CT=(V,E,H,W,C)$ ,体系结构  $G_0=(P,L,H,O,Q)$

输出:划分后计算任务  $CT'=(V',E',H',W',C')$

Multi-Level Reconfigure Partitioning algorithms ()

{ $k \leftarrow 1, Map' = \phi$ ; //  $k$  为节点块编号,  $Map'$  为中间过程

while  $k \leq m \wedge (\exists v \in V \text{ is undealed})$

// $m$  为体系结构节点个数,即  $CT$  划分的目标数

{ $v \leftarrow \text{get\_undealed\_vertex}()$ ; //随机选择未处理节点

$v_k' \leftarrow v, w_k' \leftarrow w(v), c_k' \leftarrow \text{edge\_cuts}(v), h_k' \leftarrow h(v)$ ;

// $\text{edge\_cuts}(v)$  函数返回  $v$  的边权和

set  $v$  to be dealt;

$w\_obj = \alpha \times \alpha_k, c\_obj = \beta \times \text{edge\_cuts}(p_k), h\_obj = h(p_k)$ ;

/\*  $\alpha, \beta$  是分别代表计算和通信边的比例系数,3 个参数分别对应节点特征值 \*/

while  $w_k' < w\_obj \times (1 - \epsilon_a) \wedge c_k' < c\_obj \times (1 - \epsilon_b)$

// $\epsilon_a, \epsilon_b$  允许的误差范围

{ $Q \leftarrow \text{adj}(v_k') \wedge h_k' = h(\text{adj}(v_k'))$ ;

//在  $G$  中求  $v_k'$  异构特征值相同邻接节点集

$u \leftarrow \text{arg}_u(\min(\frac{w(v_k') + w(u) + \text{edge\_cuts}(v_k' + \{u\})}{w\_obj} + \frac{\text{edge\_cuts}(v_k' + \{u\})}{d\_obj})) \wedge u \in Q$ ;

//选择影响最小节点

$v_k' \leftarrow v_k' + \{u\}, w_k' \leftarrow w_k' + w(u), c_k' \leftarrow \text{edge\_cuts}(v_k' + \{u\})$ ;

set  $u$  to be dealt;

$Map' \leftarrow Map' + \{(u, k)\}$ ;

endwhile

$Map' \leftarrow Map' + \{(v, k)\}$ ;

$k \leftarrow k + 1$ ;

endwhile

$CT' \leftarrow \text{reconfig}(Map', CT)$ ;

/\* 利用  $Map'$  和  $CT$ , 划分组合成符合要求的计算任务  $CT'$  \*/

通过上述算法,可以得到与体系结构匹配的划分图,其各个部分的权值与体系结构的结构特征匹配。

任务划分过程实际上是合并任务图节点及边的过程,划分算法是广度优先搜索算法的一种变形,在搜索时考虑了节点权值和边权值与总目标权值的比例,能保证划分  $|P|$  块。其复杂度也是  $O(|E|)$ 。因此算法的复杂度与边数存在线性关系。

#### 5 实验与性能分析

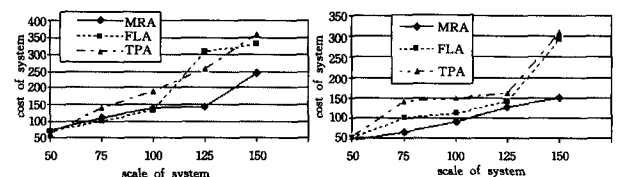
划分算法的性能主要通过实现的代价来测试,因此实验主要测试在不同的划分方法下系统所实现的代价。将其与柔性粒度划分方法(FLA)<sup>[12]</sup>、时序划分方法(TPA)<sup>[9]</sup> 进行比较。这些算法都能生成各种粒度结构,然后在模拟退火过程中随机试探这些粒度结构。实验所使用的划分算法都是基于软硬件协同设计的算法,资源调度算法均使用 ASAP 调度算法。实验结果的不同,主要是由于循环次数不同引起的。实验中的例子由普林斯顿大学的 EE Dept. CAD 实验室开发的开放源码工具 Task Graphs For Free (TGFF)生成。这个工具现在的最新版本是 TGFF 3.0.1,进一步情况可以参考文献[19]。实验以各组 20 个样本的平均性能作为结果。

实验的环境采用处理器 Intel 酷睿 i-375,内存为 1G,开发环境是 VC。先分析执行相同的循环次数时,不同算法所得到的不同的系统优化代价。表 1 列出了执行相同的次数时不同的算法给出的系统最好结果。表中给出了算法对 5 个不同规模系统的实现代价,在模拟退火执行的搜索次数增多的时候,所得到的系统代价也更加优化。

表 1 算法对不同系统的实现代价

系统	Tgff1	Tgff2	Tgff3	Tgff4	Tgff5	
节点数	50	75	100	125	150	
循环次数	6.41E+3					
MRA	79	113	146	161	282	
最小代价	FLA	83	143	195	262	300
TPA	70	103	132	318	334	
循环次数	6.53E+4					
MRA	53	73	98	126	168	
最小代价	FLA	64	87	106	179	234
TPA	50	69	126	150	256	

另外从图 5(a)、图 5(b)可以看出,在开始,MRA 普遍的优化效果要比其他两种算法差。而 TPA 在循环次数较少时有较好表现。随着循环次数的增加,MRA 相对于其他算法的优势逐渐凸现出来。



(a) 循环次数较少时的运行结果

(b) 循环次数较多时的运行结果

图 5

**结束语** 本文分析了计算任务及体系结构的异构特征及异构匹配原理,提出了异构特征匹配和多级可重构的思想,并在此基础上提出了基于异构特征匹配的多级可重构任务划分方法,从而为异构可重构计算及高效能计算提供了理论方法,也为高性能计算提供了新的并行编译思想,这种思想为异构可重构计算研究开拓了思路。另外也可将这种思想应用到目

(下转第 74 页)

## 参考文献

- [1] Rabin M O. Efficient dispersal of information for security, load balancing, and fault tolerance[J]. *Journal of the ACM*, 1989, 36(2):335-348
- [2] Hsu D F, Lyuu Y D. A graph-theoretical study of transmission delay and fault tolerance [J]. *International Journal of Mini and Microcomputers*, 1994, 16(1):35-42
- [3] Kleinberg J. Approximation algorithms for disjoint paths problems[D]. MIT, Cambridge, MA, May 1996
- [4] Li C, McCormick T, Simich-Levi D. The complexity of finding two disjoint paths with min-max objective function [J]. *Discrete Applied Mathematics*, 1989, 26(1):105-115
- [5] Xu D, Chen Y, Xiong Y, et al. On the complexity of and algorithms for finding the shortest path with a disjoint counterpart [J]. *IEEE/ACM Transactions on Networking*, 2006, 14(1):147-158
- [6] 张品, 章坚武, 李乐民, 等. QoS 约束下的链路分离路径问题研究 [J]. *通信学报*, 2006, 27(6):36-42
- [7] 熊轲, 裘正定, 郭宇春, 等. 多约束最短链路分离路径精确算法 [J]. *软件学报*, 2010, 21(7):1744-1757
- [8] 方效林, 石胜飞, 李建中. 无线传感器网络一种不相交路径路由算法[J]. *计算机研究与发展*, 2009, 46(12):2053-2061
- [9] Voigt T, Dunkels A, Braun T. On-Demand construction of non-interfering multiple paths in wireless sensor networks[C]//*Proceedings of the 2nd Workshop on Sensor Networks at Informatik 2005*. Bonn, Germany, 2005:150-154
- [10] Kawarabayashi K, Kobayashi Y. The induced disjoint paths problem[C]//*Proceedings of the 13th Integer Programming and Combinatorial Optimization (LNCS5035)*. Bertinoro, Italy, Springer, 2008:47-61
- [11] Zhang Ke-jia, Gao Hong, Li Jian-zhong. Finding multiple induced disjoint paths in general graphs[J]. *Information Processing Letters*, 2011, 111:1022-1026
- [12] Teo J Y, Ha Y, Tham C K. Interference-Minimized Multipath Routing with Congestion Control in Wireless Sensor Network for High-Rate Streaming[J]. *IEEE Transactions on Mobile Computing*, 2008, 7:1124-1137
- [13] Marathe M V, Breu H, Hunt H B, et al. Simple heuristics for unit disk graphs[J]. *Networks*, 1995, 25(2):59-68
- [14] 黄文奇, 金人超. 求解 SAT 问题的拟物拟人算法—Solar [J]. *中国科学(E 辑)*, 1997, 27(2):179-186
- [15] Huang Wen-qi, He Kun. A pure quasi-human algorithm for solving the cuboid packing problem[J]. *Science in China Series F: Information Sciences*, 2009, 52(1):52-58
- [16] 黄文奇, 许如初. 近世计算理论导引—NP 难度问题的背景、前景以及其求解算法研究[M]. 北京: 科学出版社, 2004:47-70

(上接第 66 页)

前高效能计算相应的问题求解中。但是在计算任务和体系结构通过重构实现异构特征匹配所造成的重构代价方面分析尚有不足, 如何保证以最小重构代价实现划分后异构特征的完全匹配是我们今后研究的重点。

## 参考文献

- [1] Freund R F. Optimal selection theory for superconcurrency [C]//*Proceedings of Conference on Supercomputing*. ACM: New York, 1989:699-703
- [2] Khokhar A, Prasamma V K, Shaaban M E. Heterogeneous computing: challenges and opportunities [J]. *Computer*, 1993, 26(6):18-27
- [3] Compton K, Hauch S. Reconfigurable computing: a survey of systems and software[J]. *ACM Computing Surveys*, 2002, 34(2):171-210
- [4] DeHon A, Wawrzyniek J. Reconfigurable computing: what, why, and implications for design automation[C]//*Proc of 36th Annual ACM/IEEE Design Automation Conference*. Louisiana, USA: ACM Press, 1999:610-615
- [5] 姜晶菲, 唐玉华, 刘福东, 等. 可重构异构系统结构研究[J]. *计算机技术与发展*, 2009, 19(1):25-28
- [6] Bruce H, Tamara G. Graph partitioning models for parallel computing[J]. *Parallel Computing*, 2000, 26:1519-1534
- [7] 沈轶伟, 曾国荪. 异构计算中一种图的非均衡划分算法[J]. *计算机科学*, 2006, 33(6):260-263
- [8] Selvakkumaran N, George K. Multiobjective hypergraph partitioning algorithms for cut and maximum subdomain-degree minimization[J]. *IEEE Transactions on Computer Aided Design of Integrated Circuits and System*, 2006, 25(3):504-517
- [9] Jiang Y, Wang J. Temporal partitioning data flow graphs for dynamically reconfigurable computing[J]. *IEEE Transactions on very Large Scale Integration System*, 2007, 15(12):1351-1361
- [10] Verbauwhede I, Schaumont P. The happy marriage of architecture and application in next-generation reconfigurable systems [C]//*Proceedings of the 1<sup>st</sup> conference on computing frontiers 2004*. Ischia, Italy, 2004:363-376
- [11] 郝水侠, 曾国荪, 谭一鸣. 计算任务与体系结构匹配的异构计算可扩展性分析[J]. *电子学报*, 2010, 38(11):2585-2589
- [12] Henkel J. An approach to automated hardware/software partitioning using a flexible granularity that is driven by high-level estimation techniques [J]. *IEEE Transactions on very Large Scale Integration System*, 2001, 9(2):273-289
- [13] Fukushi M, Horiguchi S. A self-reconfigurable hardware architecture for mesh arrays using single/double vertical track switches[J]. *IEEE Transactions on Instrumentation and Measurement*, 2004, 53(2):357-367
- [14] Neeb C, Wehn N. Designing efficient irregular networks for heterogeneous systems-on-chip[J]. *Journal of Systems architecture*, 2008, 54(2):384-396
- [15] Kapre N, Dehon A. SPICE<sup>2</sup>: spatial processors interconnected for concurrent execution for accelerating the SPICE circuit simulator using an FPGA[J]. *IEEE Transactions on Computer-aided Design of Integrated circuits and Systems*, 2012, 31(1):9-22
- [16] Senar M A, Ripoll A, et al. Clustering and reassignment based mapping strategy for message passing architectures[J]. *Journal of Systems Architecture*, 2003, 48(8-10):267-283
- [17] Juhasz Z, Turns S J. A new heuristic for the process-processor mapping problems [C]//*Proceedings of the Third Austrian-Hungarian Workshop on Distributed and parallel Systems*. 2000:91-94
- [18] 郝水侠, 曾国荪, 谭一鸣. 一种基于 DAG 图的异构可重构任务划分方法[J]. *同济大学学报*, 2011, 39(11):1693-1697
- [19] <http://helsinki.ee.princeton.edu/~vallerio/tgff/>