众核多态实时计算模型设计及应用

罗眉[1,2], 周兴社[1], 张凯龙[1], 胡英英[1]

(1.西北工业大学 计算机学院, 陕西 西安 710129; 2.西安高科技研究所, 陕西 西安 710025)

摘 要:为了满足复杂嵌入式实时系统中应用任务对计算的需求,提高计算有效性,将多态实时计算的概念引入众核架构中,设计了一种众核多态实时计算模型。提出了计算态的概念,并设计了三种计算态以适用于不同类型任务。仿真测试结果表明,在较低通信衰减的情况下,此众核多态实时计算模型使应用任务在以不同计算态进行计算时均可获得较高的加速比,从而提高了系统对计算量大的应用任务的响应速度。

关 键 词:多态计算,众核,计算态,计算模型
中图分类号:TP399 **文献标志码**:A **文章编号**:1000-2758(2016)02-0338-05

复杂嵌入式实时系统具有多种应用任务,多应用任务对嵌入式高效能计算提出了更高需求。通用计算使系统响应时间无法满足大计算量任务的实时性需求。因此,应依据应用任务及其需求的具体变化,设计面向应用驱动的计算模型,使其能够实时、动态地选择合适的计算模式。

1 相关工作

目前,在飞机、智能机器人等复杂嵌入式应用领域,越来越多地采用了众核处理机。众核处理机适用于密集计算,国内外对众核处理机的研究主要集中在体系结构、资源调度、CPU+GPU 异构系统相关技术以及众核应用模型[1-4]的研究。

多态计算技术[5]是指在复杂嵌入环境下支持系统多级资源重构的技术,用来反映一种计算系统软、硬件结构和计算模式适应任务需求变化的思想。目前国内外对多态技术的研究热点主要集中于体系结构、资源配置和任务调度等方面[6-8]。近年来的研究结果表明,适应不断变化应用需求的一种先进模式是建立多态计算模式。文献[8]提出了一种用户满意度驱动的多态计算模型,并将多态计算与众核相结合,但仍是围绕通用移动计算而开展的研究。目前,面向复杂嵌入式实时系统应用,且将多态计算技术和众核相结合的研究相对较少。因此,本文结合众核处理机的特点,将多态计算的概念引入到众核架构中,针对复杂嵌入式实时系统应用,建立了众核多态实时计算模型,并从逻辑上实现了众核的 3 种计算态,为快速响应不同应用任务提供了有效支撑。

2 众核多态实时计算模型

在嵌入式实时系统中,主要有 3 类任务:一类任务是计算复杂,需要占用系统全部可用资源以保证其时间约束;一类任务可独立并行执行;还有一类任务可被分解成若干个具有前驱后继的子任务执行。因此根据嵌入式实时系统应用特点,本文构建了一种众核多态实时计算模型,描述了众核资源上的系统服务软件及任务集,分为众核多态任务模型和众核多态实时服务模型。

2.1 众核多态任务模型

2.1.1 相关概念

众核多态任务模型负责接收应用任务,并按其计算需求分为单任务、多任务分区、多任务流式 3 种计算模式。为有效描述众核多态任务模型,引入如

收稿日期:2015-09-08 **基金项目**:国家自然科学基金(61572403)与中国高校基础研究基金(3102015JSJ0002)资助
作者简介:罗眉(1978—),女,西北工业大学博士研究生,主要从事多态计算与分布式并行计算研究。

下相关定义。

定义 1(计算态)指众核处理单元的核资源在不同组织形式下所对应的并行计算模式。

定义 2(应用任务)应用任务是系统的宏观行为目标。系统应用任务集由 $\boldsymbol{\Gamma}=\{T_1,T_2,\cdots,T_n\}$ 表示,$\forall T_i \in \boldsymbol{\Gamma}$ 称为一个应用任务。

定义 3(多态计算任务) 众核计算平台管理和执行的具体任务指多态计算任务,它们能够运行于众核处理机,且有不同计算形态。系统计算任务集由 $\tilde{\boldsymbol{T}}=\{PT_1,PT_2,\cdots,PT_m\}$ 表示,$\forall PT_i \in \tilde{\boldsymbol{T}}$ 称为一个多态计算任务。公式(1)描述了多态计算任务相关属性。

$$\forall PT = < PT_{ID},PT_{pr},PT_{type},PT_{\tilde{I}},PT_{\tilde{O}} > \quad (1)$$

式中,PT_{ID} 为计算任务号;PT_{pr} 为计算任务的优先级,同一计算任务在不同应用任务集中的优先级由系统指派;PT_{type} 为任务需要的计算态,$PT_{type}=0$ 时表示众核以单任务计算态工作,$PT_{type}=1$ 时表示众核以多任务分区计算态工作,$PT_{type}=2$ 时表示众核以多任务流式计算态工作;$PT_{\tilde{I}}$ 为任务输入数据;$PT_{\tilde{O}}$ 为任务输出数据。

2.1.2　众核单任务计算态

单任务计算态适用于计算复杂、时间约束强的应用任务,将众核运算单元内部的所有核资源同时分配给单一任务。单任务计算满足:

$$O_s = K(I_s) \quad (2)$$

式中:I_s 表示单任务输入数据,且满足 $\tilde{I}=I_s$;O_s 为输出数据,且满足 $\tilde{O}=O_s$;K 为计算该任务需要的 Kernel 数。

2.1.3　众核分区计算态

分区计算态适用于多个相互独立并行执行的应用任务。由于任务间相互独立,对众核资源进行划分时,会形成不同的分区,使各任务在不同的分区内执行。

将多个独立计算任务输入数据集记为 $\boldsymbol{I}_p,\tilde{\boldsymbol{I}}=\boldsymbol{I}_p=\{I_{p_1},I_{p_2},\cdots,I_{p_n}\}$;输出数据集记为 $\boldsymbol{O}_p,\tilde{\boldsymbol{O}}=\boldsymbol{O}_p=\{O_{p_1},O_{p_2},\cdots,O_{p_n}\}$;分区任务计算需要的 Kernel 数记为 $\boldsymbol{K}_p,\boldsymbol{K}_p=\{K_{p_1},K_{p_2},\cdots,K_{p_n}\}$。其中,$n$ 表示应用任务数,K_{p_i} 表示计算任务 i 所需要的 Kernel 数。于是众核分区计算可表示为:

$$\begin{cases} O_{p_i}=K_{p_i}(I_{p_i}) \\ \forall i \neq j,\quad K_{p_i} \neq K_{p_j} \end{cases} \quad i,j \in [1,n] \quad (3)$$

式中,$K_{p_i} \neq K_{p_j}$ 表示计算任务 i 和计算任务 j 不存在数据依赖关系(即相互独立),且满足 $\sum_{i=1}^{n} K_{p_i} \leqslant R$,$R$ 为众核中包含的 Kernel 总数。

2.1.4　众核流式计算态

流式计算态适用于关联并行处理的应用任务,任务之间由一组具有前驱后继关系的子任务并行执行,依据计算需求及全局优化策略,将众核资源划分为多个分区,每个分区执行一个子任务,各子任务间通过共享存储机制实现数据的快速交换。

在该计算态下,计算任务可分解为多个计算步,将各计算步表示为 $k_1,k_2,\cdots,k_i,\cdots,k_m$,其中 m 代表组成计算任务的计算步(Kernel) 数($i \in [1,m]$);用 $\tilde{I}=I_{st}=\{I_{st1},I_{st2},\cdots I_{sti}\cdots,I_{stn}\}$ 代表输入数据流集合,$\tilde{O}=O_{st}=\{O_{st1},O_{st2},\cdots O_{sti}\cdots,O_{stn}\}$ 为输出数据流集合,其中 n 表示应用任务数,$i \in [1,n]$,则流式计算可表示为:

$$\begin{cases} O_{sti}=k_m \cdot k_{m-1} \cdot \cdots \cdot k_1(I_{sti}) \\ \forall a < b,\ k_{a,i} \rightarrow k_{b,i} \quad a,b \in [1,m],i \in [1,n] \end{cases} \quad (4)$$

其中“·”表示用多个计算步组成一个计算任务。流式计算由多个这样的计算任务组成,每个计算任务将输入数据流 I_{sti} 经由 m 个计算步,最终产生输出数据流 O_{sti}。式中“→”表示依赖关系,表示第 b 个计算步对第 i 个数据流的处理必须依赖于第 a 个计算步对第 i 个数据流的处理结果,即同一数据流必须依次执行各个具有前驱后继关系的计算步。

2.2　众核多态实时服务模型

众核多态实时服务模型负责屏蔽底层硬件资源的异构性,使得系统各组成部分之间可以无差别地进行数据通信。同时,为应用提供运行时系统服务,有可用资源时通过通信接口进行数据传输。

本文基于通信有限状态机模型(communicating finite state machines,CFSM)[9] 构建了一种多态实时服务模型,其形式化描述如下:

$N=2$,

$S_1=\{$CONNECTED_1,SEND_WAIT,DATA_WAIT,SENDING$\}$

$S_2=\{$CONNECTED_2,RECV_WAIT,B_RECV_WAIT,RECVING$\}$

$o_1=$CONNECTED_1

$o_2=$CONNECTED_2

$M_{12}=\{$DATA_SEND,DATA_COM,ERROR,NO_DATA$\}$

$$M_{21} = \{\text{DATA\_ REQ}, \text{B\_ DATA\_ REQ}\} \quad (5)$$

式中，N 为正整数，表示通信过程数目。由于众核多态实时计算模型实现 PowerPC 和众核之间通信，因此 $N=2$；$[S_i]_{i=1}^{N}$ 是 N 个状态集合，S_i 代表通信过程 i 的状态集合；o_i 代表通信过程 i 的初始状态，是 S_i 中的一个元素；$[M_{ij}]_{i,j=1}^{N}$ 是 N^2 个集合，M_{ij} 代表通信过程 i 向过程 j 发送的消息集合，其中对于所有的 i 而言，M_{ii} 为空。基于 CFSM 的多态实时服务模型通信过程如图 1 所示。

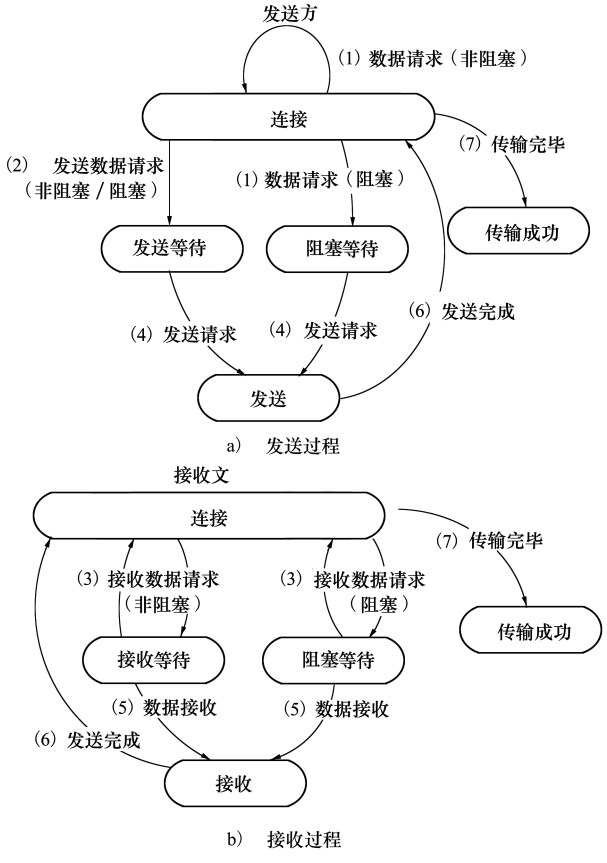

图 1 基于 CFSM 的多态实时服务模型通信过程

图中的阻塞发送方式是指发送方一直阻塞直到所要发送数据传输完成后返回，非阻塞发送是指发送方在给底层协议发送数据的指令后立即返回，真正的数据传输由底层协议完成。

3 仿真测试

仿真测试分为 2 个方面：①测试众核多态任务模型的有效性；②测试众核多态实时服务模型的通信效率。

3.1 实验设置

1）众核多态任务模型

众核多态任务模型包含 3 种计算态，分别用于不同类型的应用任务。以加速比为衡量依据，测试多态任务模型中 3 种计算态的有效性。

硬件环境为众核控制单元采用 2 块 Intel Sandy-Bridge E5-2609 的 4 核 CPU，每块 CPU 主频为 2.4 GHz，15M 缓存，众核计算单元采用基于 Kepler K20 架构的 GPU，时钟频率 0.71 GHz，CUDA 核心数 2 496。

（1）单任务计算态：测试用例如表 1 所示。当同一计算任务在计算规模增大时，分别记录运行在多核 CPU 和 GPU 上的计算时间，取加速比为多核 CPU 执行时间/GPU 执行时间。

表 1 单任务计算态测试用例

计算任务	属性				
	PT_{ID}	PT_{pr}	PT_{type}	PT_I	PT_O
PT_1	1	20	0	*Matrix A*,*B*	*A*+*B*
PT_2	2	15	0	*Matrix A*,*B*	*A*-*B*
PT_3	3	33	0	*Matrix A*,*B*	*A* * *B*

（2）多任务分区计算态：测试 2 个独立的应用任务，如表 2 所示，一个应用任务包含 6 个计算任务，另一个应用任务包含 32 个计算任务。为验证方便，计算任务的输入输出仍采用表 1 中的数据，当计算规模变化时，测试多个计算任务并发执行和分区执行时间，取加速比为多任务并发时间/多任务分区执行时间。

表 2 多任务分区计算态测试用例

计算任务属性	应用任务	
	T_1	T_2
<PT_{ID},PT_{pr},PT_{type}>	(PT_{11},PT_9,PT_6,PT_{10},PT_{17},PT_{15})	(PT_4,PT_5,…,PT_{28})
	(<11,7,1>,<9,9,1>,…,<15,18,1>)	(<4,3,1>,<5,6,1>…,<28,18,1>)

（3）多任务流式计算态：测试 2 个应用任务，一个由 3 个计算任务组成，另一个由 32 个计算任务组成，如表 3 所示。为验证方便，计算任务的输入输出仍采用表 1 中的数据，记录当计算规模变化时，多任务串行执行和多任务流式执行时间，加速比为多任

务串行执行时间/多任务流式执行时间。

表 3　多任务流式计算态测试用例

计算任务属性	应用任务	
	T_5	T_8
$<PT_{ID},PT_{pr},PT_{type}>$	$PT_2\rightarrow PT_7\rightarrow PT_5$	$PT_2\rightarrow PT_5\rightarrow PT_8,\cdots,PT_{13}\rightarrow PT_{33}\rightarrow PT_{11}\rightarrow PT_{28}$
	(<2,4,2>,<7,10,2>(<2,4,2>,<5,12,2>)	<2,4,2>,<5,14,2>,…,<28,30,2>)

(说明:1.应用任务 T_8 所包含的 32 个计算任务中,$PT_2\rightarrow PT_5\rightarrow PT_8$ 表示计算任务 8 依赖于计算任务 5,计算任务 5 又依赖于计算任务 2。2.不同流式任务,例如 $PT_2\rightarrow PT_5\rightarrow PT_8$ 和 $PT_{13}\rightarrow PT_{33}\rightarrow PT_{11}\rightarrow PT_{28}$ 可划分到不同分区。)

2) 众核多态实时服务模型

多态实时服务模型的作用是实现应用任务和硬件之间的通信交互,因此以通信接口的通信速率相比于理论峰值的衰减程度来测试多态实时服务性能。测试方法为使用 1/5 000 s 为时钟周期,每次传送 4MB 字节数据并接收到确认信息后记录 1 次所用的周期数,共传输 128MB 数据。通信速率理论峰值为 1.25 Gbit/s。多态实时服务运行在主控处理机上,采用 Curtiss-Wright 公司 VPX6-185 的 PowerPC。

3.2　实验结果及分析

根据 3.1 的设置,得到众核多态任务模型及多态实时服务模型测试结果,分别如图 1、图 2 所示。

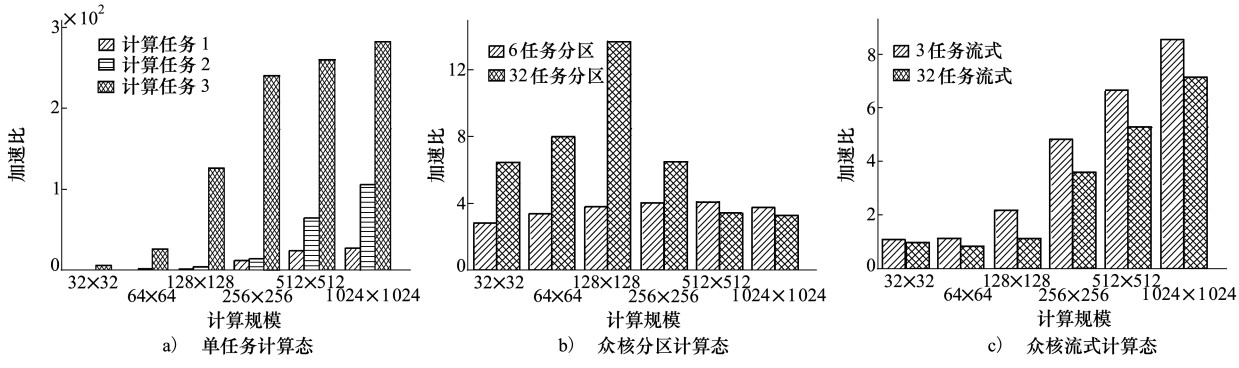

图 2　众核多态任务模型测试结果

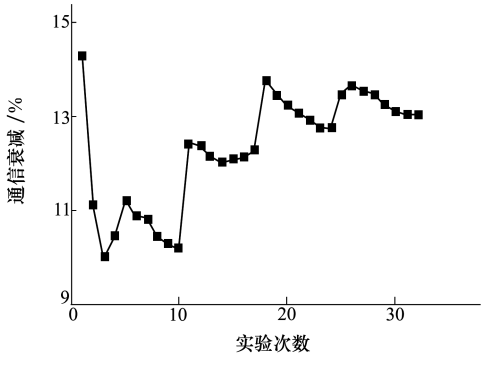

图 3　众核多态实时服务模型测试结果

图 1 显示了众核多态任务模型在以不同计算态工作时获得的加速比,实验结果表明:

1) 众核工作于单任务计算态时,从图 1a) 中看出,当计算规模较小时,由于启动 GPU 上的内核需占用一定时间,因此和多核 CPU 相比,获得的加速比较小,但随着计算规模增大,采用 GPU 计算单个任务可取得较高的加速比。

2) 众核工作于多任务分区计算态时,从图 1b) 中看出,当任务数增多,计算规模增大,由于任务组中的某些计算任务占用了较多的计算资源,虽然不能达到最大的任务级并行,但在满足系统负载的情况下,分区计算态可较好的适应多个独立任务需求,缩短计算时间,提高系统对多个任务的响应速度。

3) 众核多任务流式计算态适应于多个具有前驱后继关系的任务,从图 1c) 中可以看出,随着任务计算规模增大,任务之间需要传送数据和同步,采用流式计算,当后续计算任务数据到达时,前驱计算任务已经完成,该计算态不仅使 GPU 全局存储器充分利用,也获得了较高的加速比。

图 2 表明,多态实时服务相对于系统接口的通信速率有一定的影响,但平均衰减在 12% 左右,能够保持通信的高效性。

4　结　论

论文结合不同类型任务需求,提出了一种众核多态实时计算模型及 3 种计算态,并对所设计的模

型进行了测试。测试结果表明,众核多态实时计算模型能较好地适应复杂嵌入式实时系统的应用需求,实现快速计算,保证对不同类型任务的快速响应。

参考文献:

[1] Gibson D, Wood D A. Forwardflow: a Scalable Core for Power-Constrained CMPs[J]. ACM SIGARCH Computer Architecture News, 2010, 38(3): 14-25

[2] 曹仰杰, 钱德沛, 伍卫国, 等. 众核处理器系统核资源动态分组的自适应调度算法[J]. 软件学报,2012,23(2): 240-252
Cao YangJie, Qian Depei, Wu Weiguo, et al. Adaptive Scheduling Algorithm Based on Dynamic Core-Resource Partitions for Many-Core[J]. Journal of Software,2012,23(2):240-252 (in Chinese)

[3] Anderson Boettge Pinheiro, Francisco Heron de Carvalho Junior, Neemias Gabriel Pena Batista Arruda,et al. Fusion: Abstractions for Multicore/Manycore Heterogenous Parallel Programming Using GPUs[J]. Lecture Notes in Computer Science, 2014, 8771: 109-123

[4] Stephane Louise, Paul Dubrulle, Thierry Goubier. A Model of Computation for Real-Time Applications on Embedded Manycores [C]//2014 IEEE 8th International Symposium on Embedded Multicore/Manycore SoCs,2014:333-340

[5] Fernando J. Dynamically Reconfigurable Processing Engine for Polymorphic Embedded System[D]. Martinez Vallina, Chicago, Illinois, 2007

[6] Wu Yi, Zhou Xingshe, Wu Xiao, et al. An Embedded Real-Time Polymorphic Computing Platform Architecture[C]//2013 International Conference on Mechatronic Sciences, Electric Engineering and Computer,2013:2326-2330

[7] Arshdeep Bahga, Vijay K. Madisetti. A Dynamic Resource Management and Scheduling Environment for Embedded Multimedia and Communications Platforms[J]. IEEE Embedded Systems Letters, 2011,3(1):24-27

[8] Zhang Zhang, Swamy D. Ponpandi and Akhilesh Tyagi. An Evaluation of User Satisfaction Driven Scheduling in a Polymorphic Embedded System[C]//2014 IEEE 28th International Parallel & Distributed Processing Symposium Workshops,2014:263-268

[9] Brand D, Zafiropulo P. On Communicating Finite-State Machines[J]. Journal of the ACM, 1983, 30(2): 323-342

Design and Application of Polymorphic Real-time Computational Model Using Many-Core Architecture

Luo Mei[1,2], Zhou Xingshe[1], Zhang Kailong[1], Hu Yingying[1]

(1.Department of Computer Science and Engeenring, Northwestern Polytechnical University, Xi'an 710129, China
2.Xi'an Research Inst of Hi-Tech Hongqing Town, Xi'an 710025, China)

Abstract: In order to improve the computational efficiency and fit the needs of complex embedded real time systems, the concept of polymorphic real time computing is introduced in many-core parallel architectures. A polymorphic real time computational model is designed and the concept of computating mode is proposed.The paper design and test three types of computing mode for computational model. The results and their analysis show preliminarily that the proposed model in different computing modes can obtain higher speedup under lower attenuation of communication and the response time of tasks is reduced.

Keywords: computational efficiency, conceptual design, design of experiments, embedded software, intelligent systems, mathematical models, parallel architetures, real time systems, scalability; computing mode, many-core, polymorphic computing