

# 基于FPGA可扩展的Mapreduce 架构设计与实现

李绍松<sup>1</sup>, 尹 栋<sup>2</sup>, 慕德俊<sup>2</sup>, 戴冠中<sup>2</sup>

(1. 中国人民解放军驻844厂军事代表室, 陕西 西安 710021;

2. 西北工业大学 自动化学院, 陕西 西安 710072)

**摘要:**在基于机群的Mapreduce架构模型基础上,提出了一种基于CPU和FPGA环境、可扩展的Mapreduce架构。通过网络连接和驱动模块,实现了计算机软件与可编程硬件之间的通信,其中,CPU主机主要完成于文件系统的通行,将复杂耗时的运算过程转移到FPGA平台中处理,并引入内部流水线处理过程,大幅度加速了系统运算过程。同时,该架构可将更多的任务扩展到多个FPGA平台,弥补了器件内部存储资源的局限性,提高了系统的性能。此外,软硬件之间的命令、状态等信息交互为管理在FPGA中扩展任务提供了有效途径。实验证明,此架构在大幅提高运算速度的同时,提供了较好的底层设备可扩展性和管理的灵活性。

**关键词:**Mapreduce架构;FPGA;可扩展;协处理系统;流水线操作

**中图分类号:**TP393.02

**文献标识码:**A

**文章编号:**1673-629X(2012)02-0103-04

## Scalable Mapreduce Framework Design and Realization Based on FPGA

LI Shao-song<sup>1</sup>, YIN Dong<sup>2</sup>, MU De-jun<sup>2</sup>, DAI Guan-zhong<sup>2</sup>

(1. Chinese Military Representative Manufactory, Xi'an 710021, China;

2. Automation School, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** It presents a scalable Mapreduce framework on FPGA to accelerate commodity hardware. In this design, commodity hardware (CH) runs the main framework to communicate with file system in networks, while FPGA based platform is linked with CH to run Mapreduce tasks. Due to resource limited in one chip, more tasks can be extended to more FPGA platforms to achieve high-speed performance. A virtual device driver is designed in software to manage tasks running in special hardware. According to internal pipeline design and scalability, the design is proved that it allows better performance than commodity hardware, and also provides advantages in scalability and flexibility.

**Key words:** Mapreduce framework; FPGA; scalable; coprocessor system; pipeline

## 0 引言

Mapreduce是实现海量数据处理的一种并行编程架构<sup>[1]</sup>。最初,该架构应用于基于并行数据处理过程以及分布式文件系统的机群系统中,对海量数据的运算处理。为了实现高速处理过程,流水线操作过程被引入到该并行处理架构<sup>[2]</sup>。同时,多核处理器结构的并行处理和高速缓存的特性也加速了数据处理,提高

了运算任务的并行度<sup>[3]</sup>。然而,在该应用架构中,<key,value>对中间值的获取和汇聚成为了此应用的瓶颈。对此,Yandong Mao等人提出了一种基于工作任务特性(例如key的数目及重复的次数)进行数据结构优化的架构模型<sup>[4]</sup>,在一定程度上提高了映射和汇聚任务的整体处理速率。然而,由于Mapreduce任务程序运行在操作系统的JAVA虚拟机中<sup>[5]</sup>,实现复杂数据计算的时间开销较大,因此,可编程专用处理器(如FPGA、GPU<sup>[6]</sup>)成为了实现高速数据处理系统的一条有效途径<sup>[7]</sup>。一种基于GPU的体系架构——Mars与之前的设计相比,具有更大的运算能力和内存带宽<sup>[8]</sup>,但该架构的编程灵活性和扩展性较差。同时,基于可编程器件FPGA的可重配性设计的FPMR系统通过建立常用数据通道,极大地加快了芯片内部运算过程、提

收稿日期:2011-07-14;修回日期:2011-10-21

基金项目:西北工业大学研究生创业种子基金(Z2011049)

作者简介:李绍松(1963-),男,高级工程师,主要研究方向计算机网络、控制理论;慕德俊,教授,博士生导师,主要研究方向为信息安全、网络控制;戴冠中,教授,博士生导师,主要研究方向为网络控制。

高了数据处理并行化<sup>[9]</sup>。但由于硬件资源的局限性,海量数据计算很难在同一个 FPGA 平台快速完成,而该设计未考虑硬件平台的可扩展性以及下层计算资源的管理,因此该设计难以广泛应用。此外,一种基于 FPGA 和 GPU 的 Mapreduce 架构<sup>[10]</sup>,通过并行处理方法提高运算性能,利用可编程线程对 FPGA 和 GPU 的通信接口进行控制。然而,该研究仅提供了计算机端的上层系统设计,并未提出编程器件的内部结构的设计,也未实现硬件设备之间的通信过程以及对任务处理过程的监控。

文中提出了一种基于 FPGA 和 CPU 处理环境可扩展的 Mapreduce 架构的设计和实现过程,在现有模型的基础上,将运算任务扩展到多个 FPGA 平台中高速、并行完成,并提供通信和编程接口,而计算机主要实现数据传输。因此,该架构可应用于基于机群的 Mapreduce 体系,具有较好的扩展性和灵活性,同时,在保持原有的网络结构及计算机硬件环境的前提下,提高了系统的数据处理速度。

## 1 架构设计

Mapreduce 架构由控制主机 (Master)、工作节点 (Worker Nodes) 构成,其数据处理过程可分为两个阶段:映射 (Map) 阶段和化简 (Reduce) 阶段<sup>[11]</sup>。映射阶段中,工作节点从文件系统中读取分割好的数据块,转换成初始 <key, value> 对,然后通过运算产生中间值,并存储在本地介质中;化简阶段中,工作节点远程获取映射阶段产生的结果,进行归纳合并,产生最终的结果。

在文中设计中,每个工作节点由一台计算机和多个 FPGA 平台组成。其中,可扩展的多 FPGA 平台主要完成阶段计算任务;计算机主要实现工作节点与外界通信以及与内部 FPGA 环境的数据传输,并且可提供额外的计算资源,增强节点的运算能力,其具体设计如图 1 所示。计算机主机拥有两个网络接口,一个与分布式文件系统通信,另一个通过以太网及交换机与多个 FPGA 平台连接。架构中每一个工作节点运行的映射/化简任务有两类:一类是运行在 FPGA 平台中以及在计算中额外计算资源中运行的处理任务;另一类则是实现计算机与 FPGA 处理器通信接口功能的虚拟任务。任一个运行在可编程器件中的运算任务,在 CPU Worker 中就有一个虚拟任务接口与之对应。此外,为实现对可编程器件配置和管理,在计算机操作系统层构建了 FPGA Worker 驱动模块,主要实现可扩展器件的初始化、重配置以及数据传输。在架构任务初

始化的过程中,根据控制主机的命令实例化工作任务:建立虚拟任务接口,通过驱动模块初始化 FPGA 器件,实例化具体的运算过程。在工作阶段中,CPU Worker 从分布式文件系统中获取数据,然后分配给各任务进行处理,之后运算结果经由网络接口传输至文件系统中存储。

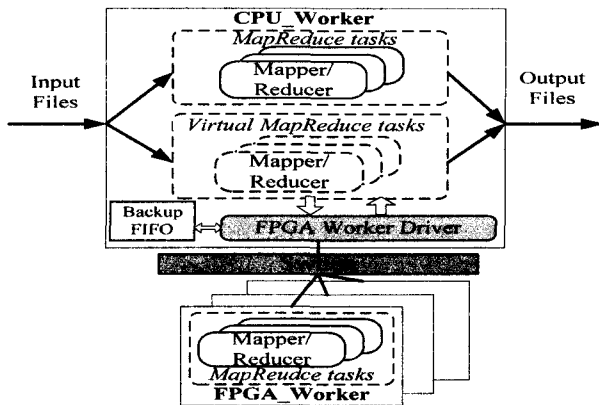


图 1 Worker Node 的结构

文中提出的 Mapreduce 架构主要针对现有设计的灵活性和扩展性进行改进。在优化扩展性方面,工作节点根据任务的复杂性,将运算分配到一个或可扩展到多个 FPGA 平台中完成。在提高灵活性方面,FPGA 驱动模块提供了基本运算过程的实例库,可根据实际任务中数据处理过程及时、有效地配置可编程器件。

## 2 架构实现

### 2.1 基于 Linux OS 的 CPU Worker 实现

文中采用现广泛应用的、基于分布式文件系统的 Hadoop Mapreduce 设计搭建整体架构。CPU Worker 运行在 Linux 操作系统层的 JAVA 虚拟机 (JVM) 环境中,主要实现三个功能:与架构中文件系统通信;配置和管理在可编程器件中实例化任务,实现软硬件之间数据传输;提供额外计算资源。

为构建虚拟任务,实现与可编程器件的通信,本设计对在软件环境中的任务进行改进,采用 Socket 传输方法,以 UDP 数据包方式传输数据。图 2 描述了在映

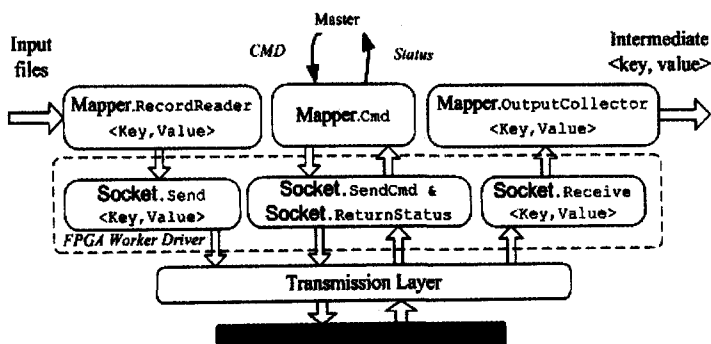


图 2 Map 过程中虚拟任务的实现

射阶段中虚拟任务的具体实现,化简阶段中虚拟任务实现方式与之类似。在 CPU 与 FPGA Worker 之间传输的主要有三种信息:数据、命令和状态。数据传输:CPU Worker 中的虚拟任务调用 Read.Record 从文件系统中获取原始数据,并且转换为初始化<key,value>对,通过 Socket.Send 将数据以 UDP 数据包方式,传输给运行在 FPGA 中的实例化任务;在完成运算后,虚拟任务通过 Socket.Receive 获得运算结果,并由 Output.Collector 初步化简传输至文件系统中存储。在数据包传输中,采用阻塞式 Socket 方式,即虚拟任务在发送完数据给 FPGA Worker 之后将等待,直至接收到返回计算结果才进行下一轮的传输。命令传达:在两种 Worker 之间传输的 UDP 包的数据域中,我们划分出 7 个字节的区域定义任务 id、控制命令、数据类型、数据长度、包的位置、<key,value>对的数据结构以及包长度。

如图 3 所示,任务 id 标识该包发送的目的任务标号;控制命令区域定义了控制主机发送给实例化任务的命令;数据类型指明了 key 和 value 的数据类型,包括 32 位整型、32 位单精度浮点型和 64 为双精度浮点型;数据长度表明<key,value>对中长度;数据位置反映该数据包是否是最后一个包。状态返回:运行在 FPGA Worker 中任务在接收到命令之后,通过查询当前状态、配置命令寄存器等方式执行指令,同时以数据包的方式通过 Socket.Return 返回状态信息。

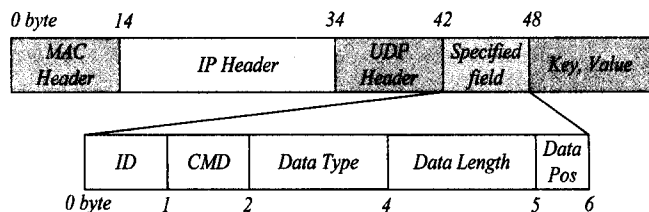


图 3 UDP 包的构造

## 2.2 基于 NetFPGA 的 FPGA Worker 实现

文中设计的可扩展架构可以应用于任何具有网络处理功能 FPGA 平台中。为缩短底层硬件开发周期,采用了斯坦福大学研发的 NetFPGA 实验板<sup>[12]</sup>作为 FPGA 平台,将实例化任务扩展到 Xilinx VirtexII Pro 芯片中完成。

FPGA Worker 具体实现如图 4 所示,主要由四部分构成:数据包接收和解析、包头校验、数据处理过程以及包封装和发送。当从网络接口接收到数据包时,将数据包拆分成包头和数据域,并行地完成包头处理和运算任务。包头处理过程:在接收到从数据包中分离的包头之后,HeaderVerification 模块首先校验包头

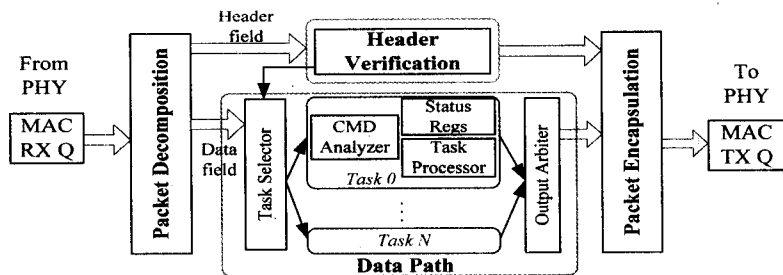


图 4 FPGA Worker 的实现

信息。若包头发现错误,则向数据处理过程报错,丢弃该包的所有数据,并向主机报错,要求重传;若包信息正确,则开始进行任务处理。数据处理过程:在任务处理过程在数据通路(Data Path)中完成。首先,Task Selector 根据任务 id 信息将数据传输到相应的实例化任务中;通过 CMD Analyzer 分析数据域中定义的控制命令信息,然后在 Task Processor 中执行相关处理过程。在完成任务之后,将结果、任务状态、更新后的 IP 包头信息,先组成 UDP 包,后按照以太网数据包格式封装并发送。由于采用片内流水线操作,FPGA Worker 实现了高速数据处理过程,远超过 NetFPGA 网络物理带宽(1Gbps),因此,该架构在数据处理过程中,连接计算机和 FPGA 平台之间的网络带宽以及传输机制成为了主要瓶颈。

## 3 实验及讨论

Mapreduce 编程架构广泛的应用于海量数据运算过程,如 Pagerank、机器学习以及地理图像处理等<sup>[13]</sup>。为测试所实现架构的性能,我们选取数据处理的常用模块矩阵乘过程。实验环境为:计算机硬件环境为 32 位双核 2.66GHz CPU、4GB 内存以及 500GB 硬盘,FPGA 工作频率为 125Hz,网络连接带宽均是 1Gbps;运算的数据类型为 32 位单精度浮点数。利用 Hadoop 分布式文件系统构建文件传输和存储环境<sup>[14]</sup>。

实验中,映射和化简过程在同一任务中完成,每一个任务进行矩阵中一行与一列的乘法运算,利用 FPGA 片内 RAM 缓存被乘矩阵列向量。我们首先在一个 FPGA 平台上运行一个任务,然后扩展任务数量,最后将多任务扩展到多 FPGA 平台。

实验结果如图 5 所示,当运算任务在同一芯片中完成时,该架构处理速度明显快于在计算机中实现的数据处理过程。然而,当运算数据量很小的时候,数据传输(socket 方式)所需的时间远大于 JVM 完成运算的时间开销,所以,在矩阵块较小的情况下,CPU 主机的运算速率较快。在一个 FPGA 平台中,随着任务数目的增加,架构的整体处理性能也随之提高。当任务

的数量从 1 个扩展至 8 个的过程中,系统整体运算速率近乎于线性地提升。

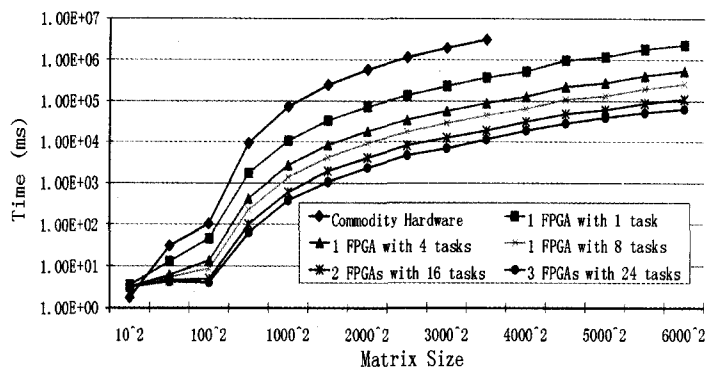


图 5 矩阵乘实验结果

由于存储资源的局限性,FPGA 器件能缓存的数据总量有限,因此,将更多的任务分配到多个 FPGA 平台中完成,最终由计算机进行结果汇聚生成最终结果。在此过程中,当运算任务扩展到两个 FPGA 平台运行时,架构的整体性能近似于成倍的提高。当 FPGA 平台数增至三台时,尽管数据处理性能仍在提高,但由于此时网络连接带宽(1Gpbs)成为了该架构的瓶颈,整体性能不再成线性关系增长。

## 4 结束语

文中设计并实现了基于 FPGA 和 CPU 的 Mapreduce 架构,通过并行处理和内部流水操作,大幅提高数据处理性能,并且在软件层提供了对可编程器件的配置和管理,保证了实际应用的扩展性和灵活性。该架构的提出具有较好的实际应用价值,在今后的工作中,将着重对构建 FPGA 配置的运算实例库,设计架构容错机制等方面继续深入研究。

### 参考文献:

- [1] Dean J, Ghemawat S. Mapreduce: simplified data processing on large clusters[C]//OSDI'04: Sixth Symposium on Operating System Design and Implementation. [s. l.]: [s. n.], 2004.
- [2] Condie T, Conway N, Alvaro P, et al. Mapreduce online[R/OL]. 2009-10. <http://www.eecs.berkeley.edu/Pubs/>
- [3] Chu C T, Kim S K, Lin Y A, et al. Mapreduce for machine learning on multicore[C]//Advances in Neural Information Processing Systems. Cambridge, MA: MIT Press, 2007: 281-288.
- [4] Mao Y, Morris R, Kaashoek M F. Optimizing mapreduce for multicore architectures[R]. Massachusetts: Computer Science and Artificial Intelligence Lab (CSAIL), Massachusetts Institute of Technology, 2010.
- [5] 张华伟, 魏庆. Java 运行原理与 Java 虚拟机[J]. 光盘技术, 2009(10): 40-42.
- [6] 张琴, 李芳. 可编程 GPU 技术的应用研究[J]. 泰山学院学报, 2008, 30(6): 20-23.
- [7] 刘吉, 杨德伟, 李立京, 等. 基于 FPGA 的高速数据处理系统设计[J]. 电子设计工程, 2011, 19(12): 158-161.
- [8] He B, Fang W, Luo Q, et al. Mars: A mapreduce framework on graphics processors[C]//PACT'08: 17th international conference on parallel architectures and compilation techniques. [s. l.]: [s. n.], 2008: 260-269.
- [9] Shan Yi, Wang Bo, Yan Jing, et al. Fpmr: Mapreduce framework on fpga[C]//FPGA'10: Proceedings of the 18th annual ACM/SIGDA international symposium on field programmable gate arrays. [s. l.]: [s. n.], 2010.
- [10] Yeung J, Tsang C, Tsoi K, et al. Map-reduce as a programming model for custom computing machines[C]//FCCM'08: 16th IEEE Symposium on Field-Programmable Custom Computing Machines. [s. l.]: [s. n.], 2008: 149-159.
- [11] 郑启龙, 房明, 汪胜, 等. 基于 MapReduce 模型的并行科学计算[J]. 微电子学与计算机, 2009, 26(8): 13-17.
- [12] Naous J, Gibb G, Bolouki S, et al. NetFPGA: reusable router architecture for experimental research[C]//Proceedings of PRESTO'08. [s. l.]: [s. n.], 2008: 10-18.
- [13] 李成华, 张新访, 金海, 等. MapReduce: 新型的分布式并行计算编程模型[J]. 计算机工程与科学, 2011, 33(3): 129-135.
- [14] 栾亚建, 黄翀民, 龚高晨, 等. Hadoop 平台的性能优化研究[J]. 计算机工程, 2010, 36(14): 262-266.
- [6] 张坤, 曹鸣. 一种基于小波变换的心电去噪算法[J]. 现代生物医学进展, 2009, 9(19): 3744-3746.
- [7] Dohono D L, Johnstone I M. Ideal spatial adaptation by wavelet shrinkage[J]. Biometrika, 1994, 81(3): 425-455.
- [8] Mallat S G. A theory for multiresolution signal decomposition: the wavelet representation[J]. IEEE Transaction on PAMI, 1989, 11(7): 673-693.
- [9] 李鸿强, 苗长云, 张龙宇, 等. 心电医疗监护物联网关键技术研究[J]. 计算机应用研究, 2010, 27(12): 4600-4603.
- [10] 谢燕江, 杨智, 范正平, 等. 应用小波变换去除膈肌心电图信号中的心电干扰[J]. 电子学报, 2010, 38(2): 366-369.
- [11] 王发牛, 程志友, 梁栋, 等. 一种冗余小波变换的心电信号噪声消除方法[J]. 计算机技术与发展, 2006, 16(11): 199-203.
- [12] 季虎, 孙即祥, 毛玲. 基于小波变换与形态学运算的 ECG 自适应滤波算法[J]. 信号处理, 2006, 22(3): 333-337.

(上接第 102 页)