维普资讯 http://www.cqvip.com

异构分布式 DSP 系统中的针对简单路径 DFG 的有效资源分配策略

何俊梅　邹显春

（西南大学计算机与信息科学学院　重庆 400715）

摘　要　DSP 系统随着其功能单元种类的增多，其资源分配问题趋向于分布式系统中的任务调度问题，但 DSP 系统中的任务调度由于兼具传统 DAG 任务图任务间有关联和实时系统有时限要求的双重特性，使得已有分布式系统中的资源分配策略无法适用，而已有的关于 DSP 系统资源分配的策略则只适用于 FU 集中分布的情况。本文提出了一种分布式异构 DSP 系统中的资源分配策略，可以适用于 FU 分布式处理任务的情况，在满足了 DSP 系统关于时限的要求的前提下使系统整体资源耗费最小。

关键词　数字信号处理，资源分配，任务调度，功能单元，系统耗费

An Effective Resource Allocating Policy Used to Handle Simple Path DFG in Distributed DSP Systems

HE Jun-Mei　ZOU Xian-Chun

(Faculty of Computer and Information Science, Southwest University, Chongqing 400715)

Abstract　Alonging with the increasing number of the types of the functioning unit, the resource allocating problems in DSP systems is much more like the similar problems in distributed and parallel systems. Traditional methods used to solve this kind of problems are no longer suited for the DSP systems its combined characteristics both like event driving and data driving dada flows. The policies use in DSP before only suited for the centralized FUs. In this paper, we propose a new resource allocation policy to handle the situation of distributed allocated FUs which can finally get a configuration of the resources at the lowest cost within the time limitation.

Keywords　DSP, Resource allocation, Job scheduling, FU, System cost

1　引言

在异构系统中最优化资源分配策略的代价函数是提高分布式系统自治并发处理性能的一个重要途径。资源分配的最优化问题根据所针对的任务流类型的不同可分为：1)事件驱动，2)数据驱动两大类。事件驱动型的任务流其特点主要体现在任务流间没有数据依赖性[1,2]，针对这种类型的资源分配策略主要指实时系统中的任务流控制，由于任务间相对独立，因此可通过限制满足条件构建其代价函数，并在解空间上应用各种启发式搜索策略来实现任务流的均衡负载[3]。而数据驱动任务流则指任务间存在数据依赖关系，任务流可用 DAG 图准确描述[4]。针对 DAG 图的资源分配所构建的代价函数(cost function)包括最小化整体响应时间[5,6]，最大化系统吞吐量[4]等。应用于数据驱动任务流的资源分配策略主要有：(1)基于权重的分配策略[7,8]。这种方法将优先级赋给各个任务，并根据优先级进行资源分配。(2)基于聚类的分配策略[9]。其主要思想是将一类在关系上相互依赖紧密的任务定位到同一个处理器上，这样可以有效地减少通信开销。然而这些已有的分布式系统资源分配策略都不适用于实时数字信号处理系统 DSP。

DSP 中的应用是通过数字方式处理信号需要特殊的高速功能单元 FU，来实现额外的加法和乘法操作[10~12]。由于越来越多的可供选择的 FU 种类出现，同一个操作可以在不同类型的 FU 上以不同的耗费代价执行。因此，一个重要的问题产生了：如何产生一种调度可以将一个合适类型的 FU 分配给每个操作，使得在保证需求的前提下使得系统整体耗费最小。这个问题类似于传统分布式系统中的资源分配问题，但是 DSP 却具有其特殊的系统特性：

(1)任务执行时间与耗费分离，系统整体耗费为单个任务耗费和。

(2)任务调度针对不同类型的处理单元(FU)，FU 数量无限，可选择多个同类型的 FU，且每个任务都有一个独立的 FU 保证其执行。

(3)任务间有关联，且总体任务执行有时限。

DSP 系统的任务调度兼具了传统 DAG 任务图调度的特性(任务间有关联)和实时系统任务调度的特性(任务流执行有时限)。这使得已有的资源分配策略并不适用于 DSP，没有一种现成的异构分布式系统中的资源分配策略是满足这个要求的。

文[13]提出了一种集中式的 DSP 资源分配策略，可以处理 DSP 系统中当所有 FU 都在一个环境中时的情况，却无法解决当 FU 分布处理任务时的最优资源分配问题。

本文提出了一个异构分布式 DSP 系统中进行最优资源分配的策略，在满足任务流执行时限的前提下可以得到最优资源分配分布。这里最优资源分布指的是系统整体耗费最小。

2　DSP 系统环境构建

给定一个 DFG(Data Flow Graph)任务流图 $G=\langle U,E\rangle$，其中 $U=\{u_i \mid i\in[1,n]\}$，表示共有 n 个任务，即 $|U|=n$；$E=$

何俊梅　讲师，主要研究领域为数据库技术、软件工程。　邹显春　副教授，主要研究领域为计算机网络、软件工程。

维普资讯 http://www.cqvip.com

$\{e_{ij}\mid i,j\in[1,n],\exists u_i\rightarrow u_j\}$，表示任务间的依赖关系，若 u_j 依赖于 u_i 的执行结果，则存在边 e_{ij}，如图 1。DSP 系统环境可用一个 8 元组表示$\langle G,P,f,F,L,\tau,C\rangle$：

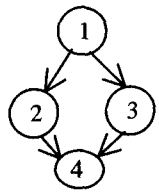

图 1　样例 DFG

(1)$P=\{P_i\mid i\in[1,M]\}$，$|P|=M$，，表示系统中共有 M 种 FU；

(2)$F=\{F_i\mid i\in[1,n]\}$，用于执行 n 个任务的 n 个 FU，有 $|F|=n$；

(3)$f(F_i)=P_j$，$i\in[1,n]$，$j\in[1,M]$，f 函数是任务 u_i 的执行 FU F_i 与 FU 种类 P_j 的关联映射函数；

(4)$\tau=\{\tau(u_i,P_j)\mid i\in[1,n],u_i\in U,j\in[1,M],P_j\in P\}$，任务 u_i 在 P_j 类 FU 上执行的时间，如图 2 所示；

(5)$L=\{L(f(F_i),f(F_j))\mid i,j\in[1,n],F_i,F_j\in F,\exists e_{ij}\in E\}$表示执行任务 u_i 和 u_j 之间的通信时间耗费，不同类型的 FU 之间的通信时间耗费不同；

(6)$C=\{c(u_i,P_j)\mid i\in[1,n],u_i\in U,j\in[1,M],P_j\in P\}$，任务 u_i 在 P_j 类 FU 上执行的耗费，如图 2 所示。

任务	P_1		P_2		P_3	
	τ	c	τ	c	τ	c
1	1	5	2	6	2	6
2	2	6	2	6	1	5
3	3	5	2	5	2	7
4	1	7	2	7	4	7

图 2　任务在不同种类 FU 上的执行情况

3　问题提出

我们最终的目标是要在执行单元 FU 是异构且承分布式分布的 DSP 环境中处理任务间具有关联的任务流图，且最终在保证任务流执行时限的情况下求得耗费最小的过程中获得对应的资源分配结果，我们构建问题的目标函数为 *Minimize* $\sum_{i\in[1,n]}c(u_i,F_i)$，即最小化系统的整体开销。且具有限制条件：

$$\sum_{i\in[1,n]}\tau(u_i,F_i)+\sum_{\forall i,j\in[1,n],i\neq j,\forall e_{ij}\in E}L(f(F_i),f(F_j))\leqslant T$$

其中 $\sum_{i\in[1,n]}\tau(u_i,F_i)$ 表示当任务串行执行时其执行时间总和(最坏情况)，$\sum_{\forall i,j\in[1,n],i\neq j,\forall e_{ij}\in E}L(f(F_i),f(F_j))$ 表示所有通信所占用的时间耗费。限制条件表示任务执行时间和通信耗费在最坏情况下都必须保证在任务流执行时延 T 以内，只有保证这个限制条件才有可行解。

4　一种改进的启发式搜索算法-PS 算法

PS 算法(Path-based Scheduling Algorithm)是对文[13]所提出的 Path-Assign 算法的一个改进算法，PS 算法充分考虑了分布式 FU 的通信耗费情况，最终可以在 DFG 图为简单路径的情况下求得系统整体耗费最小的目标函数，且保证其限制条件。PS 算法描述如下：

```
PS()
{ for i= 1 to n {
    allocate array Xi[1,2,…,T] to ui;/* ①
  for j= 1 to T {
      Xi[j]=
    { min_{k∈[1,M]}{x_{i-1}[j-τ(ui,Pk)-L(f(F_{i-1},Pk))]+c(ui,Pk)}
          若 j-τ(ui,Pk)-L(f(F_{i-1},Pk))≥T_min(i-1)        /*②
      无可行解,否则
        record (ui,Xi[j]-X_{i-1}[j]) /*③
    }
  }
}
```

①将 u_i 与数组 $X_i[1,2,\cdots,T]$相关，$X_i[j]$存储从 u_1 到 u_i 路径上的系统耗费的最小值。

②条件 $j-\tau(u_i,P_k)-L(f(F_{i-1},P_k))\geqslant T_{\min}(i-1)$表示在任务 u_i 执行的时刻其时限表现为若要使得 u_i 可以分配在类型为 P_k 的 FU 上，必须要使得从这个时刻回退剩余的时间必须要大于等于其前驱任务 u_{i-1}最少的执行时间。这样才可以保证整体任务执行的时间耗费可以保证在实现 T 以内。且此时 $X_i[j]$总是从 u_1 到 u_i 路径上的系统耗费的最小值。

③在求得 $X_i[j]$的过程中获得相应的最优任务调度分布$(u_i,F_i(j)-F_{i-1}(j)))$，即在时刻 j 时处理单元 u_i 及其上面对应的任务量 $c(u_i,F_i(j)-F_{i-1}(j))$。

可以证明 $X_i[j]$总是从 u_1 到 u_i 路径上的系统耗费的最小值。当 $i=1$ 时，$X_1[j]=\min_{k\in[1,M]}\{c(u_1,P_k)\}$，显而易见此时 $X_1[j]$为最小值，假设当 $i=n-1$ 时满足，则 $X_n[j]$也是满足的。

5　算法应用

当处理单元 u_i 分散的分布在系统中时，由于在 PS 算法中考虑了处理单元之间相互通信的通信耗费，因此 PS 算法可以准确的计算出任务分布，而 Path-Assign 算法忽略了处理器间通信的耗费，因此它只能适用于处理单元集中在一起的情况。PS 算法通过计算每个时刻处理单元的最优分布来最终实现目标函数的最优化，我们也证明了它。

结论　本文提出了一个异构分布式 DSP 系统中进行最优资源分配的策略，通过 PA 算法的启发式搜索，可以在 DFG 为简单路径的情况下得到最优资源分配分布，并满足任务流执行的时限。这里最优资源分布指的是系统整体耗费最小。我们为分布式 DSP 系统构建了一个 FU 分布式处理任务的算术模型，改进的 PA 算法考虑了 FU 间通信的耗费，使得 DSP 任务分配不再只局限于 FU 集中分布的情况。

参 考 文 献

1　Rotaru T, Nageli H- H. Dynamic load balancing by diffusion in heterogeneous systems. Journal of Parallel and Distributed Computing, 2004, 64(4): 481～497

2　Georgiadis L, Nikolaou C, et al. A fair workload allocation policy for heterogeneous systems. Journal of Parallel and Distributed Computing, 2004, 64(1): 507～519

3　Kafil M, Ahmad I. Optimal Task Assignment in heterogeneous Distributed Computing Systems. IEEE Concurrency on Complex Distributed Systems, 1998, 6(3): 42～51

4　Ali S, Kim J-K, et al. Greedy Heuristics for Resource Allocation in Dynamic Distributed Real-Time Heterogeneous Computing Systems. In: 2002 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'02), Las Vegas, NV: CSREA Press, 2002. 519～530

5　Ranaweera S, Agrawal D P. A Task Duplication Based Scheduling Algorithm for Heterogeneous Systems. In: Proceedings of the 16th International Parallel and Distributed Processing Symposium, Florida: IEEE Computer Society Press, 2002. 445～450

(下转第 292 页)

维普资讯 http://www.cqvip.com

入 TSMQ 队列时，路由服务器会自动地从 TSMQ 队列中将消息转发出去，直至目的地。通信实现流程如图 5 所示。

结束语 与用 SSL 协议开发的面向消息传输中间件不同的是，TSMQ 只是在普通 MOM 上增加模块实现功能的扩展，并不是在中间件与传输层之间增加一个新的安全套接层，即可以简化系统的复杂度，提高系统运行效率，也可以避免增加安全套接层后带来应用层数据安全路由的问题。因为增加安全套接层以后密钥的协商和数据通信的加密工作就放在安全套接层了，无论是客户端与路由服务器，还是路由服务器与路由服务器之间的连接，路由服务器都要参与会话密钥的协商，也就是说路由服务器的安全直接决定整个通信的安全性，这就带来了巨大的安全隐患，路由服务器一旦被攻击者控制，攻击者就可以轻易地取得消息明文。而在整个 TSMQ 通信过程中，路由服务器只知道消息的目的地而不知道传输的具体明文，这只有知道会话密钥的会话双方客户端才知道，从而消除了安全隐患。

参 考 文 献

1 Blahut R，Clancy T，Hua X，et al. Secure Middleware for Infrastructure Systems [R]. University of Illinois，Urbana-Champaign，January 2004. http://www.ifp.uiuc.edu/～kiyavash/papers/secureMW_04_TR.pdf

2 ndersen A，Blair G，Myrvang P H，Stabell-Kulo T. Security and middleware [C]. In：Object-Oriented Real-Time Dependable Systems，2003. Proceedings of the Eighth International Workshop on，Jan. 2003. 186 ～ 190

3 Stallings W. Cryptography and Network Security Principles and Practice (ThirdEdition)[M]. PrenticeHall，2003

4 Nash A，Duane W 著. 张玉清，等译. 公钥基础设施(PKI)：实现和管理电子安全[M]. 北京：清华大学出版社，2003

5 Tanenbaum A S 著. 潘爱民译. 计算机网络(第 4 版)[M]. 北京：清华大学出版社，2004. 8

6 李婉婷. 基于 J2EE 的安全中间件的研究与实现[D]. 解放军信息工程大学，2005. 6

(上接第 283 页)

6 Ravindran B，Li Peng，et al. Proactive resource allocation for asynchronous real-time distributed systems in the presence of processor failures. Journal of Parallel and Distributed Computing，2003，63(12)：1219～1242

7 Sih G C，Lee E A. A Compile Time Scheduling Heuristic for interconnection Constrained Heterogeneous Processors Architectures. IEEE Trans. Parallel and Distributed Systems，1993，4(2)：175～187

8 王永炎，王强，等. 基于优先级别表的实时调度算法及其实现. 软件学报，2004，15(3)：360～370

9 Bajaj R，Agrawal D P. Improving Scheduling of Tasks in a Heterogeneous Environment. IEEE Transaction on Parallel and Distributed Systems，2004，15(2)：107～118

10 Paulin P G，Knight J P. Force Directed Scheduling for the Behavioral Synthesis of ASICs. IEEE Trans. Comuter-Aided Design of Integrated Circuits and Systems，1989，8(6)：661～679

11 Wang C Y，Parhi K K. Resource Constrained Loop List Scheduler for DSP Algorithms. J VLSI Signal Processing，1995，11：75～96

12 Ito K，Lucke L. ILP-Based Cost Optimal DSP Synthesis with Module Selection and Data Format Conversion. IEEE Trans. VLSI systems，1998，6：582～594

13 Shao Z，Zhuge Q F，et al. Efficient Assignment and Scheduling for Heterogeneous DSP Systems. IEEE Trans. Parallel and Distributed Systems，2005，16(6)：516～525

(上接第 287 页)

法利用两种方法分别对二进制代码进行反汇编，引入了相互验证的过程。这种二次反汇编算法能够报告出反汇编结果中的错误，从而限制错误的传播。错误的代码可以根据应用的需求进一步进行处理。在 SPECint-95 和 SPECint-2000 上的实验证明，这种方法可以有效地对二进制文件进行反汇编。

参 考 文 献

1 GNU Project - Free Software Foundation，objdump，GNU Manuals Online. http://www.gnu.org/manual/binutils-2.10.1/html chapter/binutils4.html.

2 Larus J R，Ball T. Rewriting Executable Files to Measure Program Behavior. Software-Practice and Experience，1994，24(2)：197～2181994.

3 Larus J R，Schnarr E. EEL：Machine-Independent Executable Editing. In：Proc. SIGPLAN '95 Conference on Program-ming Language Design and Implementation，June 1995. 291～300

4 Cifuentes C，Van Emmerik M，Ung D，et al. Preliminary Experiences with the UQBT Binary Translation Framework. In：Proc. Workshop on Binary Translation，Oct. 1999

5 Theiling H. Extracting Safe and Precise Control Flow from Binaries. In：Proceedings of the 7th Conference on Real-Time Computing Systems and Applications，Dec. 2000

6 Cifuentes C，Gough K J. Decompilation of Binary Programs. Software-Practice and Experience，1995，25(9)

7 Cifuentes C，Van Emmerik M. Recovery of Jump Table Case Statements from Binary Code. In：Proceedings of the Inter national Workshop on Program Comprehension，May 1999

8 De Sutter B，De Bus B，De Bosschere K，et al. On the Static Analysis of Indirect Control Transfers in Binaries. In：Proc. International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)，2000

9 罗云彬. Windows 环境下 32 位汇编语言程序设计. 北京：电子工业出版社，2002