文章编号:1001 - 2486(2005)05 - 0039 - 05

一种基于 MPSoC 的网络处理器模型及其应用*

张晓明,孙志刚,张民选
(国防科技大学 计算机学院,湖南 长沙 410073)

摘 要:从 MPSoC 系统设计角度出发提出了网络处理器的参数化分析模型,称为 NePlat。该模型采用数据流进程网络(DPN,Dataflow Process Network)描述网络应用,构造参数化异构硬件资源,并将应用模型映射到体系结构资源上评价网络处理器性能。

关键词:MPSoC;网络处理器;NePlat
中图分类号:TP393 文献标识码:A

An Analytical Modeling Methodology for MPSoC Based Network Processors

ZHANG Xiao-ming, SUN Zhi-gang, ZHANG Min-xuan
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A parameterized analytical model of network processors is presented in terms of MPSoC design methodology, called NePlat. In this model, network applications are represented by Dataflow Process Network (DPN). Heterogeneous network processor resources are described by system parameters, and then the application model is mapped into the heterogeneous resources to analyze the performances of network processors.

Key words: MPSoC; network processors; NePlat

近年来互联网的规模不断扩大,新的网络服务不断涌现。为满足复杂多样的网络协议的处理能力和灵活性需求,网络处理器已经广泛应用于多种网络设备中。另一方面,随着深亚微米工艺的迅速发展,高性能、低功耗多处理器片上系统(MPSoC,MultiProcessor System-on-Chip)已广泛用于各种应用领域,例如信号处理系统、流媒体处理、网络协议处理等。与通用处理器相比,基于 MPSoC 的网络处理器能更好地满足系统并行性、可扩展性和灵活性需求。因此,网络处理器广泛采用 MPSoC 体系结构实现,其基本结构包含一组处理单元(PE,Processing Element)、协处理器(CoP,Co-Processor)、硬件逻辑块(HLB,Hardware Logic Block)、存储资源(Mem,Memory)、网络接口(NI, Network Interface)和互联网络(CN,Communication Network)。

基于 MPSoC 的网络处理器在单芯片上集成了大量异构资源,其设计开发过程变得更加困难。在芯片的前期设计中,设计者需要根据应用领域需求描述和评价异构资源配置体系结构,选择优化的网络处理器系统设计方案,为后期开发和部署提供设计指南。许多研究机构和企业对异构 MPSoC 网络处理器的系统设计做了广泛的研究[1~4]。本文开发了网络处理器开发平台 NePlat(Network Processor Platform)。

1 NePlat 平台结构

基于 MPSoC 的网络处理器设计平台 NePlat 采用系统级模型分析方法实现,如图 1 所示。NePlat 选择 RFC1812 中的 IPv4 转发、支持 QoS 的 IPv4 转发和 IPsec VPN 三种典型网络应用场景,而后构建每个应用场景的数据流进程网络[5](DPN,Dataflow Process Network)模型,用于支持任务到异构资源的分配。同

* 收稿日期:2005 - 05 - 20
基金项目:国家 863 高技术研究发展计划基金项目(2003AA115130);国家自然科学基金项目(90104001);国家重点基础研究发展计划项目(2003CB314802)
作者简介:张晓明(1976—),男,博士生。

时，NePlat 收集 MPSoC 结构中的各个功能部件构成异构资源库，包括 PE、CN、CoP、HLB、Mem 和 NI。根据设计空间提供的体系结构配置信息，体系结构构造器从异构资源库中构造相应的网络处理器体系结构实例。设计平台的核心是空间搜索过程，即综合评价应用程序在各种体系结构上实现性能选择出的优化的系统设计。根据系统设计目标的约束规范（例如系统吞吐率、报文延时、实现复杂度等），空间搜索过程将网络应用的 DPN 图映射到体系结构实例上，构成候选方案，并通过系统性能分析器评价候选方案，如果不满足目标约束，则调整应用映射过程和体系结构配置参数，再次评价系统，最终输出优选方案。

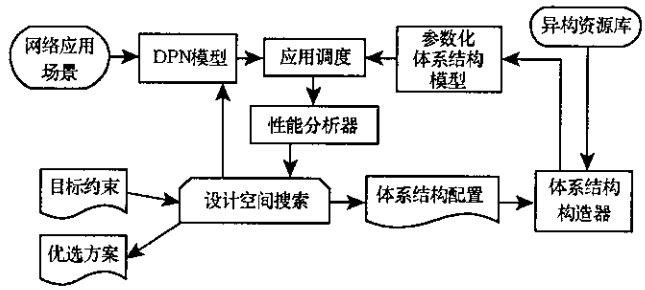

图 1 网络处理器的分析模型

Fig.1 Analytical model of network processors

2 NePlat 平台的分析模型

2.1 应用程序

针对网络处理器的 benchmark 很难在同一标准下评价不同体系结构。尽管如此，目前仍出现了许多网络处理器的 benchmark，包括 CommBench[6]，NetBench[7] 以及 NPBench[8]。为了标准化 NePlat 分析模型，从上述标准 benchmark 程序包中选择三种应用场景：(1)以报文头处理为主的 RFC1812 – IPv4 转发，来源于 NetBench 中的 Route 例程和 CommBench 中的 Radix 例程。(2)以流分类和队列管理为主的 IPv4 Diffserv 转发，来源于 NPBench 中的 TQG 程序包。(3)以报文净荷加密处理为主的 IPsec VPN，来源于 NetBench 中的 DH 例程。这三种应用场景基本能够代表网络处理器不同的负载特性，分别记为 a_{fwd}、a_{diff} 和 a_{sec}。设 $A = \{a_{fwd}, a_{diff}, a_{sec}\}$ 为本文所讨论的网络处理器应用空间。

在 NePlat 平台的应用模型中，设应用 $a \in A$ 对应的 DPN 模型为 (P_a, C_a)，其中 $V_a = \{a_{fwd}, a_{diff}, a_{sec}\}$ 为 DPN 中的 n_a 个进程结点，$C_a = \{C_{ij} \mid i, j \in \{1, 2, \cdots, n_a\}\}$ 表示进程间的 FIFO 缓冲通道。表 1 列出了 DPN 模型中进程的统计参数，每次测试取 *packetnumber* 个报文进行统计，用 $compl_i = \frac{\sum IC_{p_i}}{packetnumber}$ 表示 DPN 网络中结点 p_i 的计算能力，并用 $comm_{ij} = \frac{\sum (cap_{p_i,p_j} \cdot fcomm_{p_i,p_j})}{packetnumber}$ 表示 p_i 到 p_j 的弧 c_{ij} 的权值（即通信量）。另外，DPN 模型还引入了应用程序的访存特性统计参数。

表 1 DPN 模型中进程的统计参数描述

Tab.1 Statistic parameters of processes in DPN model

参数	描述*
IC_p	进程 p 所需执行的指令总数
$miss_p$	进程 p 运行时的 Cache 失效率
$fmem_p$	进程 p 访问 mem 的频度
$fcomm_{p_i,p_j}$	进程 p_i 和 p_j 间数据通信频度
cap_{p_i,p_j}	进程 p_i 和 p_j 间数据通信量

* 所有情况均表示处理单个报文时的统计特征

2.2 体系结构

NePlat 的体系结构空间记为 $Arch = (R, conf)$，其中，$R = \{PE, Cop, HLB, mem, CN\}$，$Conf = (\pi, \zeta)$，由异构资源配置 π 和体系结构拓扑 ζ 构成。网络处理器中的 Cop 和 HLB 都由硬件逻辑实现，区别在于前者可执行 PE 的协处理器指令，而后者只能实现特定逻辑功能。存储器分为三种类型：片上存储（记为 oc – RAM）、片外 SRAM 和片外 DRAM，并记 $mem \in \{oc - RAM, SRAM, DRAM\}$。为了避免体系结构配置种类的组合爆炸问题并且考虑网络处理的实际实现，限定了资源配置范围，如表 2 所示。

表 2 NP 硬件体系结构的配置参数和配置约束

Tab.2 Configurable parameters and constraints in NP hardware architecture

参数	取值约束	描 述
f_H	100MHz ~ 800MHz	PE 时钟频率
n_{pe}	1 ~ 64	PE 数目
t	1 ~ 32	每个 PE 线程数目
s_{ic}, s_{dc}	1KB ~ 1024KB	指令/数据 cache 容量
f_L	100MHz ~ 400MHz	HLB,CN,NI 及 Mem 时钟频率
n_{Cop}	1 ~ 8	Cop 数目
w_{mem}	16bit ~ 128bit	*mem* 位宽
c_{mem}	1KB ~ 256MB	*mem* 容量
r_{ni}	100Mbps ~ 2500Mbps	NI 接口的速率
n_{ni}	1 ~ 16	NI 接口数目
w_{cn}	16bit ~ 64bit	CN 接口位宽
cBlock	16B ~ 2KB	CN 单位数据块尺寸
ζ_{cn}	点到点,共享总线	CN 拓扑模式
τ_{DRAM}	50ns ~ 100ns	片外 DRAM 访存时间
τ_{SRAM}	2ns ~ 10ns	片外 SRAM 访存时间
τ_{oc-RAM}	$1/f_L$	片上 RAM 访存时间
$s(r)$	与硬件综合相关	资源 $r \in R$ 的硬件代价(逻辑门数)

同时,引入资源配置结构的约束条件 RC1 ~ RC4:

RC1:(时钟域)系统硬件仅包含两个时钟域 f_H 和 f_L。所有 *PE* 及其相连的 Cop 采用 f_H,其他部件采用 f_L。

RC2:(PE 组织)PE 采用 RISC 指令集,单发射流水线结构,每个 PE 内包含 t 个硬件线程。每个 PE 内 L1 数据/指令 Cache,所有 Cache 采用直接映射模式并通过共享总线连接相连到数据存储器 mem-data。

RC3:(存储器分配)应用程序相关的 storage 到硬件存储器 mem 的分配方法约定为:Cache 采用 oc - RAM,mem-data 和 mem-buf 存储器采用 DRAM 实现,mem-flow 和 mem-table 存储器使用 SRAM 实现。

RC4:(CN 模式)通信网络 CN 可采用共享总线或点到点连接两种模式之一。

当进程映射到异构资源上时,我们建立各个部件的时间分析模型。

2.2.1 PE 执行开销

当 PE 使用单线程执行进程 p 时,总访存开销为 $memstall = fmem_p \cdot (miss_p \cdot \tau_{DRAM,pe} + \tau_{SRAM,pe})$,进程 p 访问 Cop 的开销为 $copstall = p_{Cop,p} \cdot \tau_{Cop,p}$。PE 的单线程流水线 CPI(Cycles Per Instruction)表示为 $CPI_P(1) = 1 + memstall + copstall$。当 PE 中有 t 个硬件线程同时执行 a 时,通过有效线程调度机制可隐藏部分流水线停顿开销。根据文献[9]的分析结果,此时 $CPI_P(t) = 1 + \frac{memstall + copstall}{t}$。$a$ 的 CPU 时间(以秒为单位)为 $\tau_P^{PE} = CI_P \cdot CPI_P(t)/f_H$。

2.2.2 CN 通信延时

在通信网络上的两个部件中,部件之间都采用一致的通信接口与其他部件通信,通信接口带宽 $BW_{cn} = w_{ch} \cdot f_L$,其中 w_{cn} 为通信接口位宽。单位通信数据块 *cBlock* 传输时间为 $\tau_{trans} = cBlock/BW_{cn}$(忽略传输线延时)。不失一般性,设一个互联网络 CN 上包含 $Q(Q \in N^+)$ 个部件,任意部件 $i(1 \leqslant i \leqslant Q)$ 的通信请求到达某个部件 $j(1 \leqslant j \leqslant Q)$ 的请求事件服从参数为 λ_{ij} 的泊松分布,j 对 i 的服务时间 T 是固定值

$\mu = 1/\tau_{trans}$。不妨记 i 到 j 的通信时间记为 τ_{ij}^{CN}。若 $i = j$(同一部件),则 i,i 之间不采用 CN 连接,此时 $\tau_{ij}^{CN} = 0$。若互联网络通信方式为点到点模式,则 $\tau_{ij}^{CN} = \tau_{trans}$。若互联网络为共享总线,则 τ_{ij}^{CN} 表示为 $M/D/1$ 排队系统的逗留时间 $\tau_{ij}^{CN} = \left[1 + \frac{\rho}{2(1-\rho)}\right]\tau_{trans}$,其中 $\rho = \sum_{1 \leqslant i \leqslant Q, i \neq j} \lambda_{ij} \cdot \tau_{trans} < 1$。

2.2.3 进程 p 映射到 Cop 和 HLB 的执行开销

当进程实现 p 为 Cop 和 HLB 时,硬件逻辑的执行开销分别表示为 τ_p^{Cop} 和 τ_p^{HLB},这两个参数通过硬件行为级模拟的方法获取。具体而言,NePlat 使用 VHDL 描述协处理器和 HLB,在 modelsim 环境中进行功能模拟可获得它们执行报文处理功能时的时钟周期参数。

当进程 p 采用 Cop 实现,部件 $r \in U$ 与 Cop 通过 CN 通信时,部件 r 访问 Cop 的总时间为 $\tau_{r,Cop} = \tau_{r,Cop}^{CN} + \tau_p^{Cop}$。同理,当进程 p 采用 HLB 实现,部件 $r \in U$ 与 HLB 通过 CN 通信时,则 r 访问 HLB 的总时间为 $\tau_{r,HLB} = \tau_{r,HLB}^{CN} + \tau_p^{HLB}$。

2.2.4 存储器访问开销

部件 $r \in U$ 与存储器 mem 通过 CN 通信,则 r 访问 mem 的总时间 $\tau_{r,mem} = \tau_{r,mem}^{CN} + \tau_{mem}$。

2.3 异构资源映射和性能评价

我们定义 $X_{p,r}$ 和 $X_{c,r}$ 分别为 $p \in P_a$ 和 $c \in C_a$ 分配给资源 $r \in R$ 的决策变量。若 p 映射到 r,则 $X_{p,r} = 1$,否则 $X_{p,r} = 0$ 若 c 映射到 r,则 $X_{c,r} = 1$,否则 $X_{c,r} = 0$。我们定义映射约束:

AC1 对于 $p \in P_a$,满足 $X_{p,r} = 1, r \in U = \{PE, Cop, HLB\}$ 且 $\sum_{r \in R} X_{p,r} = 1$;

AC2 对于 $c \in C_a$,满足 $X_{c,r} = 1, r \in V = \{mem, CN\}$,且 $\sum_{r \in R} X_{c,r} = 1$;

AC3 若存在 $r \in R, p_1, p_2 \in P_a$,使得 $X_{p_1,r} = X_{p_2,r} = 1$,则 $X_{c_{ij},r} = 1$ 且 $c_{ij} = mem$;

AC4 若存在 $r_1, r_2 \in R, p_i, p_j \in P_a$ 且 $r_1 \neq r_2, p_i \neq p_j$,使得 $X_{p_j,r_2} = X_{p_j,r_2} = 1$,则 $X_{c_{12},r} = 1$ 且 $c_{12} = CN$。

AC1 表明每个进程只能分配给 PE、Cop、HLB 之一。AC2 表明每个通道只能分配给 Mem 和 CN 之一。AC3 表明如果两个进程分配给同一个资源,则它们间的通道只能分配给存储器 mem。AC4 表明如果两个进程分配给不同的资源,则它们间的通道只能分配给互联网络。根据 AC_i $(i = 1 \sim 4)$,映射优化问题分解为如下两个步骤:(1)将进程分成 $K = |U|$ 组;(2)将每个分组映射到 U 资源之一,将通道映射到 V 之一。

步骤 1 采用负载均衡的进程合并算法。该算法每次合并计算通信比 $\Delta_{ij} = \frac{compl_i + compl_j}{comm_{ij}}$ 最小的两个相邻结点 i,j,最终得到新的数据流进程网络 DPN',并且满足:(1) $v(DPN') \leqslant |R|$,其中 $v(G)$:$G \to N^+$ 表示图 G 中的进程数目,$|R|$ 表示 Arch 中的资源总数;(2)如果存在 DPN'' 满足条件(1),则 $v(DPN'') \leqslant v(DPN')$。经过步骤 1,$DPN'$ 的每个结点对应原始 DPN 中的一个进程分组 I_i $(i = 1, \cdots, K)$,$K \leqslant |R|$,即分组 I_i 即为 DPN' 的一个进程。

在步骤 2 中,$X_{ij,r}$ 表示将 I_i 映射到资源 r,$X_{cij,CN}$ 表示 I_i 到 I_j 的通道 C_{ij} 映射到 CN 上。针对特定硬件体系结构 $Arch_i = (R_i, conf_i)$,在确定了应用到资源的某种映射方法之后,NePlat 进一步引入资源拓扑结构 ζ 的约束条件:

RC_5:(Cop 拓扑)如果多个 PE 共享一个 Cop,则 Cop 通过共享总线与这些 PE 相连,否则采用点到点连接。

RC_6:(PE 拓扑)对于所有映射到 PE 上的分组集合 $\{I_i \mid X_{I_j,PE_i} = 1\}$,只有当 I_i 的通道唯一连接到另一个分组 I_j 时,PE_i 和 PE_j 之间的 CN 才使用点到点模式,其他情况下它们之间采用共享总线连接。

RC_7:(存储器拓扑)mem 为共享资源,所有资源 $r \in U$ 都通过共享总线连接到 mem 上。

RC_8:(HLB 拓扑)如果映射到 HLB 的分组 I_i 与其他分组 I_j 之间存在通道 c_{ij},则 $c_{ij} = mem$,且 HLB 通过共享总线连接到 mem 上。

至此,在资源配置约束 $RC_1 \sim RC_4$ 和拓扑约束 $RC_5 \sim RC_8$ 下,NePlat 平台能够完全确定网络处理器的

系统执行特性。DPN′结点 I_i 映射到资源 $r \in U$ 时的执行时间设为 $T_{I_j,r} = ET_{I_i,r} + CT_{I_j}$,其中执行时间为 $ET_{I_j,r} = \tau_{I_j}^{r} \cdot X_{I_j,r}$,通信开销为 $CT_{I_j} = \sum_{j \in k, j \neq 1} \sum_{r \in U} \tau_{ij}^{CN} X_{I_j,r} \cdot X_{c_{ij},CN}$。在某种映射方式 ψ 下的系统执行性能近似采用 DPN′分组的最长执行时间,表示为 $T(arch_i, \psi) = \max_{i=1}^{K} xET_{I_j,r}$。设应用 a 在 $Arch_i$ 下的映射优化目标值可表示为 $\phi_{archij}^{a} = \min_{\psi} T(arch_i, \psi)$。NePlat 采用随机映射方法近似求解优化的映射方案。其基本思想是:重复地随机选择一个满足约束 $AC_1 \sim AC_4$ 的映射,并且在约束 $RC_1 \sim RC_8$ 下采用分析方法评价系统,获得 $T(arch_i, \psi)$,经过多次循环从中选出 $T(arch_i, \psi)$最小的映射方案。这种方法较好地逼近最优映射方法。

2.4 设计空间搜索

根据表 2,NePlat 将体系结构空间记为 $Q = \{Arch_i \mid i$ 表示第 i 种资源组合情况$\}$,应用空间为 $a \in A$。假设同类型资源的实现代价相同,则 $Arch_i$ 的硬件实现代价 $S_i = \sum_{r \in R} n_r \cdot s(r)$。体系结构空间的搜索目标是寻找优化的体系结构,使得 $g(A, Q) = \min_{arch_j \in Q} \left(\sum_{a \in A} \omega_a \phi_{arch_i}^{a}, S_i \right)$,其中 $g(A, Q)$为二元目标函数,ω_a 为应用 a 的权值,满足 $0 \leqslant \omega_a \leqslant 1$ 且 $\sum_{a \in A} \omega_a = 1$。

3 总结和进一步工作

基于 MPSoC 结构的网络处理器设计需要考虑应用领域需求和异构体系结构特征。我们将网络应用测试例程描述为 DPN 模型并提取应用程序的统计特性,从而有助于将应用程序映射到网络处理器系统结构上。采用参数化分析模型评价不同应用下的网络处理器性能,能够指导网络处理器系统设计优化。初步给出了网络处理器设计空间开发分析模型 NePlat,下一步工作包括针对应用实例获取性能结果,开发优化的映射方法和设计空间搜索算法,用于实现特定应用需求和优化的网络处理器体系结构之间的紧密耦合及性能优化。另外,网络处理器中 IP 分组处理的顺序问题有待进一步研究。

参 考 文 献:

[1] Grunewald M, Niemann J-C, et al. A Framework for Design Space Exploration of Resource Efficient Network Processing on Multiprocessor SoCs [A]. In: Proc of the the 3rd Workshop on Network Processors & Applications[C], Madrid, Spain: Morgan Kaufmann Publishers, 2004.

[2] Ramaswamy R, Weng N, Wolf T. Application Analysis and Resource Mapping for Heterogeneous Network Processor Architectures[A]. In: Proc. of Network Processor Workshop[C], Madrid, Spain, 2004, 103-119.

[3] Weng N, Wolf T. Pipelining vs. Multiprocessors-choosing the Right Network Processor System Topology[A]. In: Proc. of ANCHOR 2004[C], Munich, Germany, 2004.

[4] Franklin M A, Wolf T. A Network Processor Performance and Design Model with Benchmark Parameterization[A]. In: Network Processor Workshop [C], Cambridge, MA, 2002, 63-74.

[5] Lee E A, Parks T M. Dataflow Process Networks[J]. Proceedings of the IEEE, May 1995.

[6] Wolf T, Franklin M. CommBench—A Telecommunication Benchmark for Network Processors[A]. In: Proceedings of IEEE International Symposium on Performance Analysis of Systems and Software[C], Austin, TX, 2000.

[7] Memik G, Mangione-Smith W H. The NetBench Web Site. http://istanbul.icsl.ucla.edu/NetBench.

[8] Lee B K, John L K. NpBench: A Benchmark Suite for Control plane and Data Plane Applications for Network Processors[A]. In: Proceedings of 21st International Conference on Computer Design[C], San Jose, CA, 2003.

[9] Agarwal A. Performance Tradeoffs in Multithreaded Processors[J]. IEEE Transactions on Parallel and Distributed Systems, 1992, 3(5): 525-539.