# HHSR:一种命令与数据分传片上网络原型

王 炜[1] 乔 林[2] 汤志忠[2] 李清宝[1]
(解放军信息工程大学信息工程学院计算机科学与技术系 郑州 450002)[1]
(清华大学计算机科学与技术系 北京 100084)[2]

**摘 要** 在前面工作的基础上,根据大规模、超大规模片上网络互连结构的性能特点,针对网络所传输信息的不同特性以及对传输的不同要求,提出了一种命令与数据分传的片上网络原型系统HHSR。该原型系统分别在两套具有不同拓扑结构的片上网络中传输命令和数据,选取速度较快且综合性能较好的单环分级互连网络用于命令包的传输,以满足其实时性的要求,选取速度稍慢但成本较低的六边形Mesh网格用于数据包的传输。实验结果表明,这种命令与数据分传的片上网络原型系统在牺牲一定的数据包传送时间和花费一定成本的基础上,保证和提高了命令与控制信息的传送速度,从而保证和提高了整个片上多处理器的性能。

**关键词** 片上多处理器,片上网络,拓扑,性能模拟
**中图法分类号** TP393.03 **文献标识码** A

## HHSR:A Prototype of Network-on-chip with Commands and Data Transferred Separately

WANG Wei[1] QIAO Lin[2] TANG Zhi-zhong[2] LI Qing-bao[1]
(Department of Computer Science and Technology, Institute of Information Engineering, PLA Information Engineering University, Zhengzhou 450002, China)[1]
(Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China)[2]

**Abstract** Based on the previous research, a kind of novel network-on-chip which transfers commands and data separately was proposed, with regard to the different requirements for the commands transmission and the data transmission in chip-multiprocessors and the characteristics of different topologies in the large and very-large scale. A prototype to this novel network-on-chip named HHSR was presented. HHSR transfers commands and data in two networks-on-chip with different topology in the same chip-multiprocessor, which transfers commands in single hierarchical ring, a kind of network-on-chip with higher speed and higher general performance, to meet the needs for real-time to the commands transmission, and data in 2-D hexagon mesh grid, a kind of network-on-chip with lower speed but lower cost. The prototype guarantees and improves the transmission speed of the commands so as to guarantee and improve the performance of the whole chip-multiprocessors with a little speed-down to the data transmission and a little more cost.

**Keywords** Chip multiprocessor, Network-on-chip, Topology, Performance simulation

## 1 引言

随着集成电路上晶体管资源的不断增加,多核处理器(Multicore Processors)或者称为片上多处理器(Chip Multiprocessor,CMP)中处理器核的数目必将不断增加,从目前的几个、十几个发展到几十个甚至更多。作为一个功能整体,需要将片上多处理器的各处理器核相互连接起来。随着片上系统和片上多处理器规模的增加,网络化方法被逐渐引入片内互连。为区别于片外网络,这种专门用于芯片内部互连的通信网络被称之为片上互连网络[1](On-Chip Interconnect Network,OCIN)或片上网络[2](Network-on-Chip,NoC)。

片上网络研究的根本问题是如何以更低的成本为片上多处理器提供更高的通信服务,使运行在片上多处理器上的应用程序获得最好的性能。片上网络的拓扑结构定义了网络内部结点的物理布局和互连方法,决定了结点度和网络链路数,从而决定了网络延迟、带宽、吞吐率和系统功耗、芯片面积和容错能力等,及影响路由策略和芯片的布局布线方法[3]。因此,选择和设计合适的片上网络拓扑结构,是片上网络研究中的关键技术之一[4]。

事实上,各种片上网络互连结构都有一定的适应范围。例如,有的互连结构在较小规模下性能良好,但是当网络规模增加时,性能下降较多;有些互连结构则在较小通信强度下性

到稿日期:2011-05-31 返修日期:2011-09-28 本文受国家自然科学基金项目(61073007),国家高技术研究发展计划(863)项目(2008AA01Z108),国家重点基础研究发展计划(973)项目(2007CB310900)资助。

王 炜(1975-),男,博士,讲师,主要研究领域为计算机系统结构、片上多处理器与片上网络,E-mail:wangwei05@gmail.com;乔 林(1972-),男,博士,副教授,主要研究领域为计算机系统结构、并行编译与优化技术;汤志忠(1946-),男,教授,博士生导师,主要研究领域为计算机系统结构、指令级并行编译技术;李清宝(1967-),男,教授,博士生导师,主要研究领域为计算机系统结构、网络信息安全。

能较好，但是当网络通信强度较大时性能较差。因此，在研究确定片上网络互连拓扑结构时必须考虑所需连接的对象以及实际进行传输的信息的情况。

本文根据大规模、超大规模片上网络互连结构的性能特点，针对网络所传输信息的不同特性以及对传输的不同要求，提出一种命令与数据分传的片上网络原型系统 HHSR。原型系统将所传输的命令和数据分别在两套具有不同拓扑结构的片上网络中进行传输，其中选取速度较快且综合性能较好的单环分级互连网络用于命令包的传输，满足其实时性的要求，使用速度稍慢但成本较低的六边形 Mesh 网格用于数据包的传输。实验结果表明，这种命令与数据分传的片上网络原型系统在牺牲一定的数据包传送时间、花费一定成本的基础上，保证和提高了命令与控制信息的传送速度，从而保证和提高了整个片上多处理器的性能。而且，随着片上多处理器内部通信局部性的增加和通信中命令包占全部通信比例的增加，其整体性能优势将更加明显。

## 2 相关工作

Mesh 网格[5]是最常用的片上网络互连结构之一，它将连接对象呈十字网格的形式连接到一起，结构简单、寻径方便，而且可扩展性好、功耗也较小，被广泛应用到片上多处理器中，例如 Trips[6]、Tile[7]、Teraflops[8]、Godson-3[9] 等内部均采用 Mesh 网格方式互连。

Mesh 网格中所有结点在某一个方向（水平或垂直）上实际是一个线性阵列，因此在较大规模网络连接中网络直径较大，传输延迟也较大[10,11]。Mesh 环网[12]（即 Torus）将 Mesh 网格的每一行和每一列分别环绕起来，从而降低网络直径，提高网络通信速度，但也增加了网络成本和功耗，并且给片上多处理器的布局布线带来一定困难[10,11]。

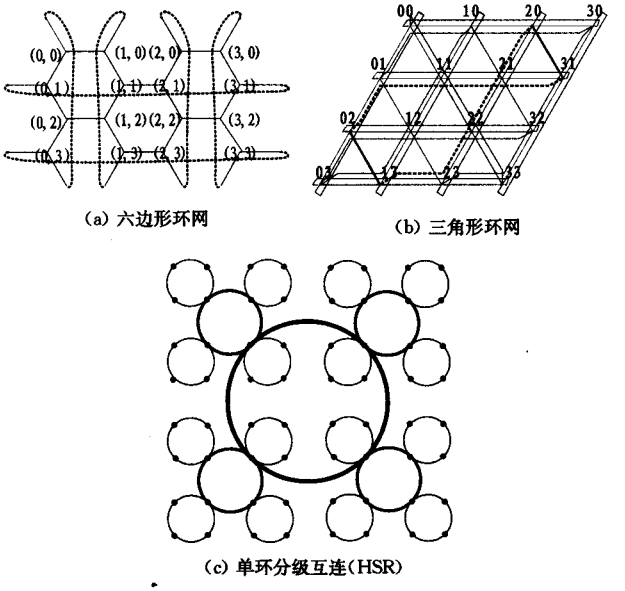
文献[13]对二维网格片上网络互连结构进行扩展，提出六边形网格片上网络互连结构和三角形片上网络互连结构，它们都和 Mesh 网格一样结构简单、规整，扩展性好。

六边形网格是 Mesh 网格的子图。如图 1(a)所示的是六边形环网，它是在 Mesh 环网的基础上减少了一半水平方向的链路，使得每一个节点的度由原来的 4 减少为 3，相应地其网络链路数也变为 Mesh 环网的 75%。若以网络链路数简单地表示网络成本[13-16]，则六边形环网的成本仅为 Mesh 环网的 75%。为描述片上网络的负载能力，文献[13]使用平均每一时刻网络中允许的新通信请求的最大值来表示网络负载，并定义网络通信传送完成率不低于 95%时网络所能承受的最大负载为网络的理想负载，它同时也是平均延迟随着网络负载增加而增加率最大时的网络负载；定义使网络通信传送完成率不低于 80%时网络所能承受的最大负载为网络的有效负载，它同时也是网络通信传送完成率随着网络负载增加而降低率最大时的网络负载。

实验结果表明，虽然六边形网格相对于 Mesh 网格而言等分带宽较小、平均通信延迟稍大、网络负载能力也较低，但是当网络通信强度相对固定并且不高于理想负载时，通信传输平均延迟与 Mesh 网格相当，而成本延迟积明显低于 Mesh 网格。因此在大规模、超大规模网络中，若不是特别追求高的网络通信速度和网络带宽，使用六边形网格互连方法将获得更好的综合性能[13,14]。

而三角形网格则是在 Mesh 网格的基础上增加某一方向的斜向连接链路构成的，如图 1(b)所示的是三角形环网。显然，三角形网格相对于 Mesh 网格而言具有等分带宽大、平均通信延迟低、网络负载能力高、通信传输可靠性高等特点。但是，这种结构成本较高，而且由于交叉开关的成本和复杂性均与所连接的链路数平方成正比[17,18]，故虽然其链路成本延迟积与 Mesh 网格相差不大，但是交叉开关成本延迟积却比 Mesh 网格大得多[13,14]。

二维网格及其扩展结构虽然具有很好的扩展性，但是当网络规模较大时，网络直径与平均寻径距离也较大，不利于高速通信传输。为提高通信传输速度，在大规模、超大规模片上网络中可以采用分级互连方法。事实上在很多情况下通信往往具有局部性：只有少数通信需要在相邻较远的结点间进行，更多的通信是在相邻较近的结点间进行[19,20]，这种通信局部性在片上多处理器上往往更加明显[21,22]。文献[15]结合片上网络通信的局部性特征，提出了如图 1(c)所示的基于卡诺图编码与寻径的单环分级（Single Hierarchical Ring，HSR）片上网络互连结构。




(a) 六边形环网　　(b) 三角形环网

(c) 单环分级互连(HSR)

图 1 新型片上网络互连结构

与 Mesh 等通过二维平面方式进行互连不同，HSR 通过级联方式将网络中的结点按区域逐层连结起来，不仅节省了链路，降低了网络直径和平均寻径距离，而且随着网络规模的增加，分级环互连方式下网络直径和平均寻径距离的减小越来越明显，逐渐抵消了级联链路网络冲突带来的性能损失，从而使得分级环方式相对于 Mesh 网格等互连结构的性能越来越好。

## 3 HHSR 命令与数据分传片上网络原型

片上多处理器内部结点间通信和传输的信息分为两类：一类为数据包，另一类是命令包。数据包往往比较大，需要分成若干数据片进行传输；而命令包往往较小，只需封装成一个数据片。数据包传输允许较长的时间延迟，对通信传输的实时性要求较低；而命令包通常完成应答、同步等功能，需要快速响应，对实时性要求较高。一般地，数据包信息的数量较少，并且很少对它进行广播或组播，命令包的数量则相对较多，而且经常出现广播或组播的情况（例如在 Cache 一致性协

议中)。

片上网络中所传递的不同信息对片上多处理器性能的影响也不同。一般而言,命令包对片上多处理器的性能影响更大,而数据包对性能的影响则相对要小得多。在适当采取一些措施,例如使用合理的编程模型、改变 Cache 的结构、容量、替换策略等,可以将片上网络结点间传递的数据包数量降到很低。而结点间传送的命令/控制信息虽然也能降低,但是其数量(例如 Cache 一致性协议中的消息)仍然很大。

由于数据包和命令包对通信传输速度的要求差别较大,若将它们混合在一个网络里进行传输,将使所有信息的平均传输速度相同。在较小规模网络中,有效负载下不同结构网络的平均延迟相差不大,网络成本也均可接受;但在大规模、超大规模片上网络中,由于有效负载下不同结构的平均延迟相差较大,同时不同结构成本差别也较大,若仍使用同一套网络,要么降低命令包的传输速度,影响系统的性能;要么片上网络成本过高,影响系统的综合性能。

如图 2 所示,各种结构网络中数据包与命令包在通信中所占的比例不同,因此网络的性能不同。随着通信中命令包比例的上升,网络传输性能线性提升。为了保证或提高网络通信中命令与控制信息的传输速度,使之不受到数量较少而且实时性要求不高的数据信息传输的影响,从而保证或提高系统的性能,一种可能的选择是使用两套网络,分别传输命令与控制信息和数据信息。

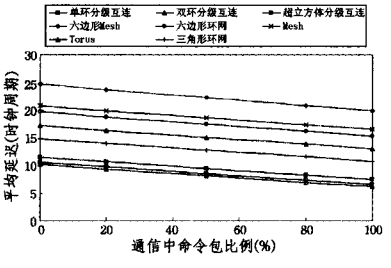



图 2 通信中命令包比例对网络性能的影响

通过命令与数据分传完成大规模、超大规模片上多处理器内部通信,既可使用完全相同的两套网络,也可选取两套不同的网络分别传输命令与控制信息和数据信息。无论哪种方案,为了不使命令与控制信息的传输速度受到影响,其中用于传输命令与控制信息的网络应该选择速度较快的结构。

文献[14]的分析结果表明,单环分级片上互连结构虽然其中少数级联结点的结构比较复杂、成本较高,但是整个网络传输速度较高、成本较低。在大规模、超大规模片上网络中,使用单环分级互连方式能获得较好的综合性能,因此可以选择使用单环分级互连网络传送命令与控制信息。

若使用两套单环分级互连结构分别传输命令信息和数据信息,整个系统命令与控制信息的传输速度仅有少量提升。虽然网络成本变成原来的两倍,且数据信息的传输速度也较快,但由于主要影响性能的命令与控制信息的传输速度提升较少,因此片上多处理器的综合性能得不到大的提升。

文献[13-16]的实验结果显示,在大规模、超大规模片上网络中,六边形 Mesh 网格虽然速度比较低、网络负载能力较低,但是其成本仅为 Mesh 网格的 75%左右。德克萨斯 A&M 大学的 Jin Yuho 等发现,虽然采用的 Mesh 网格片上网络中网络部分占用了 52%的芯片面积,但是 Mesh 网格 20%的链路其实从来都不使用[23]。由于数据包信息相比较而言对整个片上多处理器的性能影响较小,而且数据包传输的实时性要求一般较低,因此可以使用六边形 Mesh 网格来实现大规模片上多处理器中数据信息的传输。

这样,结合片上多处理器中不同信息传输的特点和不同信息对网络通信强度的影响,在大规模、超大规模片上网络中使用两套独立的通信传输链路:六边形 Mesh 网格用于传输数据包,单环分级互连结构用于传输命令包。即构成一种命令与数据分传的片上网络 HHSR(Hexa plus Hierarchical Single Ring)原型系统。

### 3.1 HHSR 成本

片上网络的成本主要由链路成本和路由结点成本构成。链路成本(含数据缓冲区)可以用链路数表示,而路由结点成本则可用路由结点内部交叉开关成本来表示[13-16]。交叉开关的成本与其硬件复杂性相关,而交叉开关的硬件复杂度为 $O(n^2,w)$,其中 $n$ 为输入/输出数量,$w$ 为带宽[17,18]。

HHSR 中的单环分级互连部分的级联链路数量仍然按照等差序列方式设置[15],图 3 将 HHSR 的成本与其它片上网络结构进行对比。

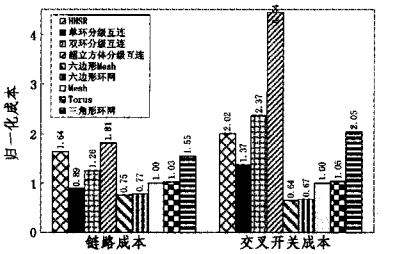



图 3 HHSR 网络成本

比较结果显示,HHSR 的成本较高,其中其链路成本比 Mesh 网格多了 64%,交叉开关成本则比 Mesh 网格多了一倍左右,但链路成本和交叉开关成本都与三角形环网相近。

### 3.2 HHSR 平均延迟与负载能力

为比较网络性能,同文献[13-16]一样,仍然在不同结构网络中模拟传输同一个基于全局均匀随机通信流量模型、通信强度为 1、命令包占 90%的通信请求序列,并分别记录不同结构中所有数据包、命令包以及全部通信的平均延迟。图 4 给出了 32×32 网络下不同结构通信的平均延迟情况。

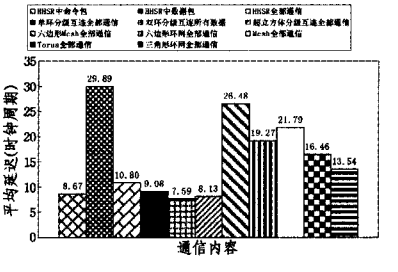



图 4 HHSR 通信传输平均延迟

命令与数据分传以后,同样通信流量序列在两套不同的网络中传输,降低了对应网络的通信强度。实验数据显示,HHSR 中命令包的传送速度比单环分级互连中命令包的传送速度提高了 4.47%,比 Mesh 网格提高了 60.2%,比三角形环网提高了 35.93%,因此 HHSR 达到了保持或提高命令与控制信息传送速度的目的。

虽然数据包使用六边形 Mesh 网格传送,平均延迟较大,一定程度增大了 HHSR 全部通信的平均延迟,但是 HHSR

全部通信的平均传输速度仍然较快，比 Mesh 网格提高了 50.44%，比三角形环网提高了 20.23%，仅比单环分级互连方式降低 18.95%。

上述实验数据基于全局均匀随机通信且命令包占全部通信的 90%的情况。但是，片上多处理器往往具有较强的通信局部性，而且局部通信概率越高，六边形 Mesh 网格相比其它结构的通信传输速度劣势越小[16]。因此在较高局部通信概率时，HHSR 全部通信的平均通信传输速度必然更快；另一方面，通信中命令包的比例越大，全部通信性能越由命令与控制信息的传送速度决定。而当命令包占全部通信的 90%时，由于数据包的长度为命令包长度的 5 倍[13-16]，相当于片上网络所传输的数据片中仅 64.3%是命令与控制信息，而其它 35.7%均为数据信息，因此实际上命令包占全部通信的 90%这个比例是比较小的。

由此可见，HHSR 命令与数据分传网络通过增加一定成本和牺牲数据包传输的性能，保证和提高了命令传送的速度，而且片上多处理器内部通信局部性越强，命令包在全部通信中的比例越高，其全部通信的平均速度越快。

实验数据显示，六边形 Mesh 网格在全部传送短包(数据包)时的理想负载为 1.5，而单环分级互连网络在全部传送长包(命令包)时的理想负载为 24，由于数据包长度为命令包长度的 5 倍，HHSR 在命令包占 90%的情况下的理想负载为 22.5。而同等情况下，单环分级互连的理想负载为 19.5，六边形 Mesh 网格则为 10.5。

考察 HHSR 工作在理想负载下的通信平均延迟。在命令包占 90%时，虽然 HHSR 的理想负载为 22.5，但是由于用于传送数据的六边形 Mesh 网格的理想负载只有 1.5，故只能使 HHSR 工作在通信强度为 15 的情况下。实验结果显示，此时 HHSR 中命令包的平均延迟为 7.65 跳，所有通信的平均延迟为 9.66 跳。与之对应，当单环分级互连工作在通信强度为 15 的情况下时，其通信平均延迟为 8.64 跳，HHSR 中命令的传送速度平均提高 11.5%。

图 5 将网络通信传输速度和成本综合考虑，并将 HHSR 的综合性能与其它网络进行对比。数据显示，理想状况下，拥有两套网络的 HHSR 的链路成本延迟积仅为 Mesh 网格的 81%，交叉开关成本延迟积也仅与 Mesh 网格相当，但是由于增加了一套链路，故其链路成本延迟积和交叉开关成本延迟积均高于单环分级互连方式。这表明与单环分级互连命令与数据混合传送方式相比，HHSR 通过花费一定成本，保证和提高了网络中命令与控制信息的传输速度。

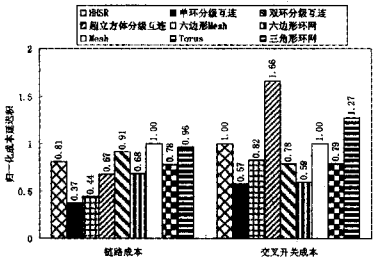



图 5 HHSR 综合性能

**结束语** 本文根据大规模、超大规模片上多处理器内部通信的特点和对不同信息通信的要求，结合不同结构大规模、超大规模片上网络的性能特点，提出了一种命令与数据分传的片上网络原型系统 HHSR。系统使用单环分级互连结构传送命令与控制信息，使用六边形 Mesh 网格传送数据信息，在牺牲一定的数据包传送时间、花费一定成本的基础上，保证和提高了命令与控制信息的传送速度，从而保证和提高了整个片上多处理器的性能。而且，随着片上多处理器内部通信局部性的增加和通信中命令包占全部通信比例的增加，HHSR 的整体性能优势越明显。

## 参 考 文 献


[1] Dally W J, Towles B. Route packets, not wires: on-chip interconnection networks[C]// Proceedings of the 38th Design Automation Conference (DAC 2001). New York, NY: ACM, 2001: 684-689

[2] Salminen E, Kulmala A, Hämäläinen T D. On network-on-chip comparison[C]// Proceedings of the Tenth Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD 2007). Los Alamitos, CA: IEEE Computer Society, 2007: 503-510

[3] Neeb C, Wehn N. Designing efficient irregular networks for heterogeneous Systems-on-Chip[C]// Proceedings of the Ninth Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD 2006). Los Alamitos, CA: IEEE Computer Society, 2006: 665-672

[4] 王炜，乔林，汤志忠. 片上网络互连拓扑综述[J]. 计算机科学，2011，38(10)：1-5

[5] Kumar S, Jantsch A, Soininen J, et al. A network on chip architecture and design methodology[C]// Proceedings of 2002 IEEE Computer Society Annual Symposium on VLSI (ISVLSI 2002). Los Alamitos, CA: IEEE Computer Society, 2006: 117-124

[6] Gratz P, Kim C, Sankaralingam K, et al. On-chip interconnection networks of the TRIPS chip[J]. IEEE Micro, 2007, 27(5): 41-50

[7] Wentzlaff D, Griffin P, Hoffmann H, et al. On-chip interconnection architecture of the tile processor[J]. IEEE Micro, 2007, 27(5): 15-31

[8] Vangal S R, Howard J, Ruhl G, et al. An 80-tile sub-100-W teraFLOPS processor in 65-nm CMOS[J]. IEEE Journal of Solid-State Circuits, 2008, 43(1): 29-41

[9] Hu Wei-wu, Wang Jian, Gao Xiang, et al. Godson-3: A scalable multicore RISC processor with X86 emulation[J]. IEEE Micro, 2009, 29(2): 17-29

[10] Neeb C, Wehn N. Designing efficient irregular networks for heterogeneous systems-on-chip[J]. Journal of Systems Architecture-Embedded Systems Design, 2008, 54(3/4): 384-396

[11] Elmiligi H, Morgan A A, El-Kharashi M W, et al. Power optimization for application-specific networks-on-chips: A topology-based approach[J]. Microprocessors and Microsystems-Embedded Hardware Design, 2009, 33(5/6): 343-355

[12] Dally W J, Towles B. Route packets, not wires: on-chip interconnection networks[C]// Proceedings of the 38th Design Automation Conference (DAC 2001). New York, NY: ACM, 2001: 684-689

[13] 王炜，乔林，杨广文，等. 扩展二维网格片上互连性能分析[J]. 清华大学学报：自然科学版，2010，50(1)：161-164

[14] 王炜，乔林，杨广文，等. 二维片上网络互连性能分析[J]. 计算机研究与发展，2009，46(10)：1601-1611

[15] 王炜，乔林，杨广文，等. 分级环片上网络互连[J]. 计算机学报，2010，33(2)：326-334

[16] 王炜,乔林,杨广文,等. 二维片上网络局部均匀随机通信性能分析[J]. 计算机研究与发展,2010,47(3):532-540

[17] Denneau M M. The Yorktown simulation engine [C]// Proceedings of the 19th Design Automation Conference. Piscataway, NJ:IEEE,1982:55-59

[18] Broomell G,Heath J. An integrated-circuit crossbar switch system design[C]//Proceedings of the 4th International Conference on Distributed Computing Systems(ICDCS 1984). Los Alamitos,CA:IEEE Computer Society,1984:278-287

[19] Faraj A, Yuan Xin. Communication characteristics in the NAS parallel benchmarks[C]// Proceedings of International Conference on Parallel and Distributed Computing Systems(PDCS 02). Cambridge:IASTED/ACTA Press,2002:724-729

[20] Vetter J F,Mueller F. Communication characteristics of large-scale scientific applications for contemporary cluster architectures[J]. Journal of Parallel and Distributed Computing,2003,63(9):853-865

[21] Tutsch D, Lüdtke D. Chip multiprocessor traffic models providing consistent multicast and spatial distributions[J]. SIMULATION,2008,84(2/3):61-74

[22] Pande P P,Grecu C,Ivanov A,et al. Design,Synthesis,and Test of Networks on Chips[J]. IEEE Design & Test of Computers,2005,22(5):404-413

[23] Jin Y,Kim E J,Yum K H. A domain-specific on-chip network design for large scale cache Systems[C]// Proceedings of the 13st International Conference on High-performance Computer Architecture(HPCA 2007). Los Alamitos,CA:IEEE Computer Society,2007:318-327

(上接第 298 页)

据传输所占的比例。从数据可以看出,在使用 Hybrid 方式之后很大程度上提高了计算所占的比例,同时也减少了因为 GPU 和 CPU 之间数据传输所占用的时间。

表 1 D3Q19-LBM MPI+CUDA 实现的计算与通信所占比例

| | 64 | 96 | 128 | 192 | 240 |
|---|---|---|---|---|---|
| comp(%) | 8.64% | 10.19% | 11.23% | 11.49% | 11.29% |
| mpi(%) | 2.53% | 1.85% | 1.48% | 1.13% | 1.01% |
| memcpy(%) | 88.83% | 87.95% | 87.29% | 87.38% | 87.70% |

表 2 D3Q19-LBM Hybrid 实现的计算与通信所占比例

| | 64 | 96 | 128 | 192 | 240 |
|---|---|---|---|---|---|
| comp(%) | 46.68% | 52.05% | 52.56% | 53.92% | 52.41% |
| mpi(%) | 3.37% | 3.21% | 2.08% | 1.74% | 1.47% |
| memcpy(%) | 38.59% | 36.42% | 36.14% | 34.78% | 34.32% |

MPI+CUDA 实现中绝大多数时间用在了 Device 和 GPU 直接的数据传输,性能上相比 MPI 的 D3Q19-LBM 实现提升了 2 倍,而 Hybrid 实现则提升了约 5 倍。这里,Hybrid 运行时,在单计算节点采用的是 2 个 MPI 进程,分别对应 2 个 GPU Device,每个进程设置的 OpenMP 线程数为 4,重复利用 4 核的计算性能。

参数 *fraction* 的数值也会影响到 Hybrid 的程序性能。从表 3 可以看出,对于碰撞过程来说,在 GPU 上的计算比例大更能获得高性能,但增加 GPU 上计算的同时,也会增加 GPU 和 CPU 之间的数据传输量和数据传输时间。移动过程由于涉及到边界上的通信过程,因此需要更多地利用到 CPU,以减少与 CPU 或 GPU 之间的数据传输,更好地提高性能。

表 3 fraction 对性能的影响

| | collision_time% | evolution_time% |
|---|---|---|
| fraction=0.25 | 38.72% | 35.16% |
| fraction=0.35 | 40.06% | 39.94% |
| fraction=0.5 | 38.08% | 45.56% |
| fraction=0.75 | 41.75% | 43.97% |

**结束语** 本文对 D3Q19-LBM 在异构平台上采用了多层次并行模式进行实现,考查了在 CPU+GPU 平台上应用的性能表现,并初步探索了 CPU 和 GPU 直接的负载均衡问题。从试验结果来看,Hybrid MPI+OpenMP+CUDA 能很大程度上提高应用性能,改善计算和通信时间比;在 GPU 和 CPU 之间可以通过计算量的配比参数来调节负载,并进一步优化性能。

异构平台上应用的实现及性能表现与应用的特点紧密相关。应用的计算及通信特征决定了其并行性,LBM 是 CFD 中通信模式比较简单的计算方法,CFD 在异构平台下的性能研究还有待进一步深入。文中对 D3Q19 的 MPI+CUDA 及 Hybrid 实现的性能还有待进一步的优化。

## 参 考 文 献

[1] nVIDIA. http://www.nvidia.com/object/cuda_home_new.html

[2] TOP 500 List. http://www.top500.org/lists/2011/06

[3] Stone J E, et al. Accelerating molecular modeling applications with graphics processors[J]. Journal of Computational Chemistry,2007,28(16):2618-2640

[4] Ogaa S,et al. GPU computing for 2-dimensional incompressible-flow simulation based on mulit-grid method[J]. Transactions of JSCES,Paper No. 20090021,2009

[5] HaraDa T,et al. Smoothed particle hydrodynamics on GPUs[C]// Proceeding of the Spring Conference on Computer Graphics. 2007:235-241

[6] Rossinelli D, et al. GPU accelerated simulation of bluff body flows using vortex particle methods[J]. Journal of Computational Physics,2010,229:3316-3333

[7] 郭照立,等. 格子 Boltzmann 方法的原理及应用[M]. 北京:科学出版社,2009

[8] Li W,et al. GPU-based Flow Simulation with Complex Boundaries[M]. GPU Gems2,Adison-Wesley,2005:747-764

[9] Fan Z,et al. GPU cluster for high performance computing[C]// SC'04:Proceedings of the 2004 ACM/IEEE Conference on Supercomputing. IEEE Computer Society,Washington,DC,USA,2004:47

[10] Qian Y,Succi S,Orszag S. Recent advances in lattice Boltzmann computing[J]. Ann. Rev. Comp. Phys. ,1995,3:195-242

[11] Armaly B,Durst F,Rereira J,et al. Experimental and theoretical investigation of backward facing step[J]. J. Fluid Mech. ,1983,127:473-496

[12] Wang Xian, et al. Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster [Z]. Parallel Comput,2011