

# 曙光 5000 高性能计算机 Barrier 网络的设计

曹 政<sup>1),2),3)</sup> 王达伟<sup>1),2),3)</sup> 刘新春<sup>1),2)</sup> 孙凝晖<sup>1),2)</sup>

<sup>1)</sup>(中国科学院计算技术研究所 北京 100190)

<sup>2)</sup>(中国科学院计算机系统结构重点实验室 北京 100190)

<sup>3)</sup>(中国科学院研究生院 北京 100039)

**摘 要** 为优化 Barrier 操作的性能,提高大规模并行计算应用在曙光 5000 系统中的执行效率,文中提出了一种基于硬件的 Barrier 加速设计.该设计是采用树形 Barrier 算法,通过增强曙光 5000 互联网络交换芯片的功能,实现低延迟、可扩展、高可靠和可管理的 Barrier 网络.该网络支持并发 16 个 Barrier 操作,可在 Fat-Tree 拓扑环境下实现较低的 Barrier 操作延迟.相比已有实现,是更适合 Fat-Tree 拓扑的设计方案.理想情况下,1024 个节点的同步操作在 1.7 $\mu$ s 内完成.根据 Barrier 操作归约和分发过程的特点,分别采用请求应答和超时催促两种机制,为 Barrier 操作的可靠性提供保障.以该设计实现的 Barrier 网络原型系统已通过 FPGA 验证.

**关键词** 高性能计算机;多级互联网络;胖树;Barrier;同步;归约;分发;可靠

**中图法分类号** TP303

## Design of Barrier Network of Dawning 5000 High Performance Computer

CAO Zheng<sup>1),2),3)</sup> WANG Da-Wei<sup>1),2),3)</sup> LIU Xin-Chun<sup>1),2)</sup> SUN Ning-Hui<sup>1),2)</sup>

<sup>1)</sup>(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

<sup>2)</sup>(Key Laboratory of Computer System and Architecture, Chinese Academy of Sciences, Beijing 100190)

<sup>3)</sup>(Graduate University of Chinese Academy of Sciences, Beijing 100039)

**Abstract** To lower barrier operation's latency and improve large-scale parallel applications' efficiency in Dawning 5000 system, this paper proposes a hardware-based accelerating solution to barrier. The design, which implements tree-based barrier by enhancing Dawning 5000 switch chip, has features of low latency, high scalability, high reliability, and high serviceability. Dawning 5000 barrier network supports 16 concurrent barrier operations. Compared with related works in Fat-tree topology, it is a more proper solution. In ideal situation, the barrier operation of 1024 nodes can be finished within 1.7 microseconds. Based on characteristics of barrier reducing and distributing, two different mechanisms are used to guarantee reliability. The prototype system of proposed design has been verified on FPGA platform.

**Keywords** high performance computer; MIN; fat-tree; Barrier; synchronization; combine; distribute; reliability

收稿日期:2007-11-28;最终修改稿收到日期:2008-06-04.本课题得到国家“八六三”高技术研究发展计划项目“曙光 5000 高效能计算机”(2006AA01A102)资助.曹 政,男,1982 年生,博士研究生,主要研究方向为分布式计算、计算机体系结构、高性能互联网络等. E-mail: cz@ncic.ac.cn.王达伟,男,1980 年生,博士研究生,主要研究方向为分布式计算、计算机体系结构、高性能互联网络等.刘新春,男,1968 年生,博士,副研究员,主要研究方向为可重构计算、计算机体系结构、高性能互联网络等.孙凝晖,男,1968 年生,博士,研究员,博士生导师,主要研究领域为并行体系结构、分布式操作系统、高性能计算等.

## 1 引 言

Barrier 操作是一种全局同步操作,被广泛应用于并行计算领域.如 BSP 编程模型中,每完成一个超步的计算和通信,就调用一次 Barrier 同步.Barrier 操作时,系统处于阻塞等待状态,其性能直接影响到并行程序的性能.

对于全局同步频繁的应用,Barrier 操作甚至成为性能的瓶颈.Scott 对一个气象学程序分析后发现,128 个处理器规模下,Barrier 操作每  $200\mu\text{s}$  执行一次,若每次 Barrier 操作增加  $15\mu\text{s}$ ,则程序执行时间增加  $7\%$ <sup>[1]</sup>.因此,减少 Barrier 操作执行时间,可以达到优化程序性能的目的.

曙光 5000 高效能计算机是中国科学院计算技术研究所国家智能计算机中心开发的下一代超级计算机,采用超并行处理体系结构.面向大规模并行计算应用,旨在解决千万亿次带来的挑战.为提高大规模并行计算应用在曙光 5000 系统中的执行效率,需要对 Barrier 操作的性能进行优化.

针对 Barrier 操作优化的工作很多,基本围绕中央计数器、锦标赛<sup>[2]</sup>、Pairwise<sup>[3]</sup>和树形<sup>[4]</sup>4 种算法展开.

软件实现大多采用中央计数器算法,易于实现是该算法的最大优势.但由于处理机对计数器内存区读写操作频繁,产生热点访问,因而性能较差.

锦标赛和 Pairwise 算法属于分步骤的 Barrier 算法,提高了访存的并发度,一定程度上减少了热点访问.软件实现方式下,Pairwise 可以获得最好的性能<sup>[5]</sup>.这两类算法具有一个特点,那就是每个步骤中,节点的通信对象固定.基于这个特点,俄亥俄州立大学<sup>[6-7]</sup>对 Barrier 操作进行了优化.它在网卡中设置 RDMA 描述符队列,预先存储每个步骤通信所需的 RDMA 描述符.进行 Barrier 操作时,网卡自动从队列中取出描述符进行通信,队列为空时,标识 Barrier 操作完成.整个过程对主机透明,缩减了 RDMA 启动开销,加速了 Pairwise 算法.

软件实现 Barrier 操作,具有很好的灵活性,但由于存在较大的软件开销,无法得到很好的性能.为进一步优化 Barrier 操作的性能,许多高性能计算机系统采用硬件实现方案.NEC 公司的 Earth Simulator 用硬件实现了全局 Barrier 计数器(Global Barrier Counter),可在  $2\sim 3\mu\text{s}$  内完成 64 个节点的同步操作.Cray T3D<sup>[8]</sup>实现了度为 4 的树形 Barrier

专用网络,全局完成 Barrier 操作只需  $2\mu\text{s}$ .IBM BlueGene<sup>[9]</sup>实现的专用 Barrier 网络采用二项式树结构,全局 65536 个节点完成 Barrier 操作只需  $1.4\mu\text{s}$ .

专用网络可以获得较高的 Barrier 操作性能,却存在成本高、扩展性和灵活性差的缺点.实际系统中,属于不同应用的 Barrier 操作会同时存在,这些 Barrier 操作的处理机集合也可能存在交集.专用网多为硬连线电路,只能通过划分物理边界来实现多个 Barrier 操作并发执行.这使得 Barrier 操作的规模受限于确定的物理边界,无法自由配置.一个物理分区中只支持一个 Barrier 操作,导致同一个处理机上的多个 Barrier 操作串行执行<sup>①</sup>,降低了多任务在系统中的执行效能.

Cray T3E<sup>[11]</sup>和 Quardics<sup>[10]</sup>提供了基于数据网络的 Barrier 实现方案.Cray T3E 网络是直接网络,使用 3-d Torus 拓扑结构,支持并发 32 个 Barrier 操作,可以在  $2\mu\text{s}$  内完成 56 个节点的同步,相比于纯软件实现有 7 倍性能提升,但其实现方法只适于 3-d Torus 拓扑.Quardics 网络的典型拓扑是 Fat-Tree<sup>[11]</sup>,其 Barrier 网络可以在  $6\mu\text{s}$  内完成 64 个节点的同步.共享网络方式下,多个 Barrier 操作可并发执行,且能够获得较低的延迟和较好的可扩展性,是性价比较高的方案.因此,曙光 5000 Barrier 网络采用共享网络的方式,基于曙光 5000 高性能互联网络实现对 Barrier 操作的支持.

曙光 5000 高性能互联网络是一种多级网络(MIN),采用 Fat-tree 拓扑结构.现有的方案<sup>[1,6-7,10]</sup>选择网卡作为 Barrier 树的根节点,网卡负责 Barrier 协议实现.而 Fat-tree 拓扑下,网卡位于树形结构的最底层,基于网卡的方案必然导致通信路径过长,无法实现 Fat-tree 拓扑下最优的 Barrier 操作性能.使用网卡实现 Barrier 协议,适于实现端对端的可靠性协议,相比点对点的可靠性协议,大大增加了 Barrier 出错恢复的时间.

我们充分利用 Fat-tree 拓扑结构的特点,发挥共享网络实现方案的优势,从延迟、扩展性、可靠性和可管理性 4 个方面对曙光 5000 Barrier 网络进行设计,以获得最优的 Barrier 操作性能.Fat-tree 拓扑被广泛地应用在多处理机系统的数据互联网络<sup>[12-15]</sup>中,因此,本文的工作具有广泛的实际意义.

① Thorson G et al. Serialized race-free virtual barrier network. US Patent 6085303, 1997

本文第 2 节对网络设计过程中遇到的关键问题和解决方案进行介绍;第 3 节全面阐述曙光 5000 Barrier 网络部件的微体系结构,并着重介绍 Barrier 高可靠通信协议;第 4 节是对 Barrier 网络的性能评价;第 5 节给出全文的总结以及对未来工作的展望。

## 2 曙光 5000 Barrier 网络设计

曙光 5000 互联网络采用 Fat-tree 拓扑结构,具有扩展性好、确定路由无死锁的特点,处理机节点位于最底层叶子上,中间节点为交换机。曙光 5000 高性能网络是面向 1024 个处理机节点的大规模互联网络,目标在于实现节点间高带宽低延迟通信,为提高并行应用性能服务。曙光 5000 互联网络中的每个网卡都支持四路并发互联,与交换机一起构成多层互联网络结构。交换机的核心部件是交换芯片。曙光 5000 交换芯片为 16 端口全双工(单端口带宽 5Gb/s),交叉开关设计,支持多虚通道,使用基于绝对信用的流量控制机制,虚切入(VCT)交换方式。同时,曙光 5000 互联网络采用带外管理的方式,通过另一套监控网对各网络部件进行配置和监视。

充分利用上述曙光 5000 互联网络的特点,针对之前 Barrier 网络设计的不足,我们从低延迟、可扩展、高可靠和可管理配置 4 个方面对曙光 5000 Barrier 网络进行了设计,下面分节介绍。

### 2.1 低延迟与可扩展

基于曙光 5000 互联网络对 Barrier 网络进行设计,首先要选择一种适合网络拓扑的实现方案。Fat-tree 拓扑具有天然的树形结构,因此本文选择树形 Barrier 算法。

树形算法中,Barrier 操作分为两个过程,首先是 Barrier 到达通知,其次是 Barrier 完成通知。Barrier 到达通知是,处理机通知系统,本处理机到达 Barrier 同步点的过程,该过程具有自叶子到根逐级归约的特征,也被称为归约过程;Barrier 完成通知是,系统通知处理机,系统已到达 Barrier 同步点的过程,该过程具有自根到叶子逐级分发的特征,也被称为分发过程。归约和分发路径构成一次 Barrier 操作的通信路径。为减少 Barrier 通信时间,首先要缩减 Barrier 通信路径的长度。

Fat-tree 拓扑结构下,处理机节点均位于树形结构的最底层,记节点间的通信长度为  $L(\text{leaf} \rightarrow \text{leaf})$ ,节点与最高层交换机的通信长度为  $L(\text{leaf} \rightarrow \text{root})$ ,则

$$L(\text{leaf} \rightarrow \text{leaf}) = 2 \times L(\text{leaf} \rightarrow \text{root}).$$

若选择处理机节点作为 Barrier 树的根,则 Barrier 操作的通信路径长度为

$$L = 2 \times L(\text{leaf} \rightarrow \text{leaf}) = 4 \times L(\text{leaf} \rightarrow \text{root}).$$

若选择最高层交换机作为 Barrier 树的根,则 Barrier 操作的通信长度为

$$L = 2 \times L(\text{leaf} \rightarrow \text{root}).$$

因此选择最高层交换机作为 Barrier 树的根可将通信路径缩短一半,减小 Barrier 传输延迟,该方案下,交换机负责对 Barrier 数据包的归约和分发,Barrier 操作的过程如图 1 所示。

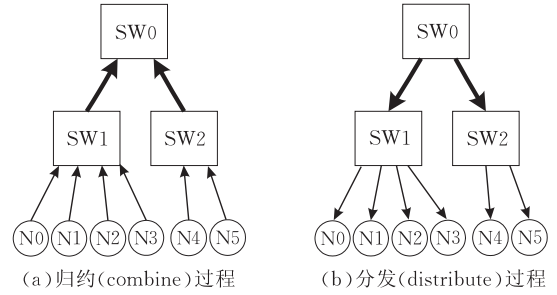


图 1 Barrier 执行过程示意图

利用 LogP 模型<sup>[16]</sup>,使用曙光 5000 互联网络的性能参数,对中央计数、锦标赛、Pair-wise、Quardics 的纯硬件 Barrier 树<sup>[11]</sup>以及本文的树形算法进行分析。

设定处理机与处理机间通信延迟为  $L(P)$ ,软件及网卡发送开销为  $O_s$ ,通信库及网卡接收开销为  $O_r$ ,系统的节点数为  $p$ 。两次 Barrier 操作之间有串行性,因此不考虑连续两次操作的间隔  $g$ ,只分析一次 Barrier 操作性能。为简化分析,令  $O_s = O_r$ ,应用于 Fat-tree 拓扑结构,假设多处理机同时到达 Barrier 同步点,几种算法的 Barrier 操作时间为

中央计数:

$$T = O_s + 2 \times (L(P) + P \times O_s).$$

锦标赛:

$$T = 2 \times \lceil \log_2 P \rceil \times (L(P) + 2 \times O_s).$$

Pair-wise:

$$T = \lceil \log_2 P \rceil \times (L(P) + 2 \times O_s).$$

Quardics 树形 Barrier(硬件):

$$T = 6 \times O_s + 3 \times L(P).$$

本文树形 Barrier:

$$T = 2 \times O_s + L(P),$$

其中

$$L(P) = (2 \times \lceil \log_8 (P/2) \rceil - 1) \times (d + l) + l,$$

$d$  为交换机 Barrier 模块的处理延迟, $l$  为两级交换

机间的传输延迟. 参数取值分别为  $d = 58\text{ns}$ ,  $l = 40\text{ns}$ ,  $O_s = O_r = 600\text{ns}$ .

如图 2 所示, 树形 Barrier 算法适合曙光 5000 Barrier 网络, 本文提出的以交换机为根的树形结构可以获得最佳的性能. 同时, 本文的设计具有较好的扩展性, 在 1024 节点的规模下, 完成一次同步操作只需  $1.7\mu\text{s}$ .

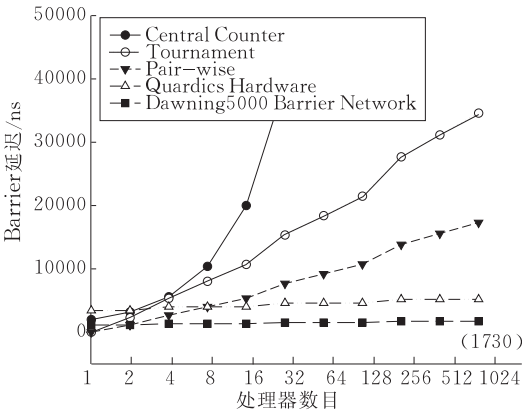


图 2 基于 LoGP 模型分析的 Barrier 性能

2.2 可靠性

网络实现 Barrier 操作, 除保障其低延迟外, 也要保障可靠性, 本节讨论的可靠性只涉及链路传输错误, 与软硬件故障及操作失误无关. 曙光 5000 高性能服务器系统的 1024 个处理机节点, 通过 320 个交换机进行互联, 共有 3072 条通信链路, 6144 个 Serdes(串并收发器), 以 BER 等于  $10^{-13}$  计算, 每天会出现一次全局 Barrier 错误. 一次 Barrier 数据包

出错, 轻则影响 Barrier 操作性能, 重则导致程序运行崩溃, 本文的树形结构为设计一种高效的点对点可靠性机制提供了可能.

Barrier 操作面临两类可靠性问题: (1) Barrier 包丢失或者出错; (2) 多次 Barrier 操作混杂. 首先通信链路和热噪声导致第一类问题, 对于这类错误, 检错和纠错的时机非常关键, 直接影响到操作的性能. 其次, 处理机完成 Barrier 操作具有异步性, 由于单播数据传输的影响, 异步性特征在共享网络方式下更为明显. 较早完成分发过程的处理机会提前进入下一次 Barrier 操作, 多次操作的混杂是第二类问题. 为保证可靠性而重传的 Barrier 数据包激化了这类问题的出现, 对多次 Barrier 操作的有效区分是解决问题的关键.

2.2.1 包丢失/出错处理

解决包丢失/出错, 现有的方法多为请求应答机制, 即发送方在一定时间内没有收到应答包, 则主动重传数据. 本文同时考虑另一种超时催促机制, 即接收方在一定时间内没有收到等待的数据, 则发送催促包请求发送方重传.

应用于 Barrier 操作, 对两类机制的工作流程进行分析, 记操作开始时间为  $t_0$ , 超时计数阈值为  $D$ , 相邻交换芯片间的通信延迟为  $T$ . 操作过程如图 3 所示, 对归约过程, 请求应答机制可以更及时地对错误进行恢复, 对分发过程, 两类机制完成出错恢复的时间相同. 因此归约过程适于采用请求应答的可靠性机制.

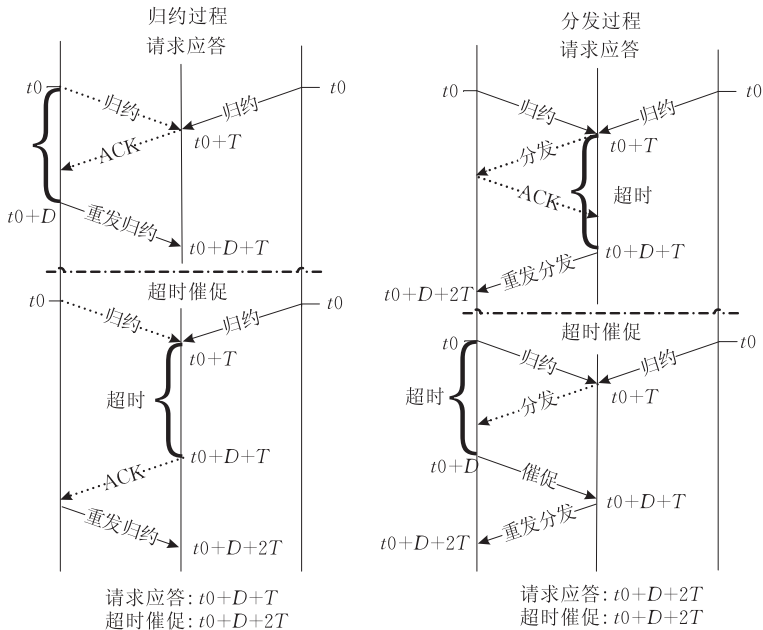


图 3 Barrier 包出错/丢失处理

对于分发过程可靠性机制的选择,还需考虑通信量和实现复杂度两方面的影响。采用请求应答,交换芯片每进行一次分发操作,就要搜集来自所有叶子端口的分发 ACK 包,处理过程类似于 Barrier 归约操作,需设置独立的处理状态完成。采用超时催促,每进行一次分发操作,只需处理个别叶子端口的分发催促包,无需设置处理状态。因此超时催促机制可以减少用于可靠机制的 Barrier 包数目,并简化 Barrier 状态机的设计,是适于分发过程的可靠性机制。

根据上述分析,设定 Barrier 操作的可靠性机制如下:交换芯片在发出归约包之后,启动进行归约重传和分发催促超时计数。若在重传计时超时之前,仍未接收到归约应答包,则启动归约包的重传;反之,停止重传计数。若在催促重传计时超时之前,仍未接收到分发包,则启动分发催促包的发送。

### 2.2.2 多次 Barrier 操作混杂

对于多次 Barrier 操作混杂,有如下定理成立。

**定理 1.** 在同一棵 Barrier 树中,最多同时存在两次 Barrier 操作。

证明。记一棵 Barrier 树中的网络部件集合为  $A = \{p_0, p_1, p_2, \dots, p_n\}$ , 序号为  $s$  的 Barrier 操作存在标记为  $B(s)$ ,  $B(s)$  的归约操作完成标记为  $C(s)$ , 分发操作完成标记为  $D(s)$ , 完成归约操作的部件集合为  $E(s)$ , 完成分发操作的部件集合为  $F(s)$ 。

设同一时刻,有序号分别为  $\{0, 1, 2, \dots, i, i+1, \dots, n\}$  个 Barrier 操作存在于一棵 Barrier 树中。启动顺序按序号大小升序排列。首先,根据 Barrier 操作特点获取下列推导。

对于一次 Barrier 操作,分发过程必须在归约过程完成之后:

$$\forall p \in A: C(s_i) \rightarrow \exists p \in A: D(s_i) \quad (1)$$

由于 Barrier 操作的串行性,对于一个网络部件来说,必须完成上一次的 Barrier 分发才能进行本次的 Barrier 归约:

$$\exists p \in F(s_{i-1}): D(s_{i-1}) \rightarrow \exists p \in F(s_{i-1}): C(s_i) \quad (2)$$

若  $B(s_{i-1}), B(s_i)$  同时存在,则由式(1)、(2)可知,系统状态为

$$(\exists p \in F(s_{i-1}): D(s_{i-1})) \wedge (\exists p \in F(s_{i-1}): C(s_i)) \quad (3)$$

由于  $F(s_{i-1}) \subset A$ , 故式(3)可表示为

$$(\exists p \in A: D(s_{i-1})) \wedge \neg(\forall p \in A: C(s_i)) \quad (4)$$

由式(1)、(4)可知,

$$F(s_i) = \emptyset \quad (5)$$

由式(5)、(2)、(1)可知,

$$E(s_{i+1}) = \emptyset, F(s_{i+1}) = \emptyset \quad (6)$$

由式(6)可知,序号为  $s_{i+1}$  的 Barrier 操作不存在,即最多存在连续两次的 Barrier 操作,且一个处于分发过程,一个处于归约过程。

因此采用单比特即可实现对多次 Barrier 操作的区分,在 Barrier 包和 Barrier 模块中均设定序号位,根据序号的匹配情况进行相应操作。

应用了本节可靠性方案的操作协议将在后面的章节进行详细说明。

### 2.3 可管理性

曙光 5000 互连网络采用带外管理,即上层软件通过另一套监控网络对曙光 5000 互连网络进行管理和监控。同样地,利用该监控网,可以实现上层软件对 Barrier 操作的管理和配置。

首先,与 Sangman Moh<sup>[17]</sup>的工作类似,在网卡和交换芯片中设置 Barrier 配置寄存器。上层软件通过监控网络对配置寄存器访问,写入相应的 Barrier 树形结构信息,灵活建立 Barrier 树,使 Barrier 树的构成不受物理边界限制。

其次,在网卡和交换芯片中设置 Barrier 操作寄存器,实时记录 Barrier 操作的状态。上层软件通过读取该寄存器,即可判断是否出现如软件崩溃或硬件故障导致的 Barrier 操作瘫痪。若确认 Barrier 操作已经瘫痪,上层软件可通过配置 Barrier 操作寄存器,对 Barrier 操作状态进行修改,使 Barrier 操作从故障中恢复。

最后,在网卡和交换芯片中设置 Barrier 性能寄存器。为实现可靠的 Barrier 操作,本文采用了基于超时重传的可靠性机制。若超时阈值过大,则影响 Barrier 出错恢复的时机,过小则会造成重传包或催促包的泛滥,影响正常的数据传输。Barrier 操作过程中,归约重传阈值应与链路状态呈正比,分发催促阈值应与交换机的层次成反比,与链路状态呈正比。上层软件通过配置 Barrier 性能寄存器,即可实现对两个阈值的修改,实现对 Barrier 操作性能的动态干预。

## 3 曙光 5000 Barrier 网络实现

上述的关键技术为保证 Barrier 操作的性能提供了基础,Barrier 网络的实现则直接影响操作的性能。网卡和交换芯片构成了曙光 5000 Barrier 网络。

网卡只负责 Barrier 操作的启动和结束,Barrier 的归约和分发操作均由交换机完成,因此网卡的实现不在此进行描述.

本节将详细描述曙光 5000 Barrier 网络为保证低延迟高可靠 Barrier 操作所做的具体实现.首先介绍交换芯片在微体系层次对 Barrier 操作的支持,然

后对 Barrier 模块和 Barrier 操作协议进行阐述.

### 3.1 交换芯片微体系结构

曙光 5000 交换芯片通过集成 Barrier 模块实现 Barrier 相关协议的处理,带外管理模块负责对 Barrier 模块进行配置和监控,实现上层软件对 Barrier 操作的管理,如图 4 所示.

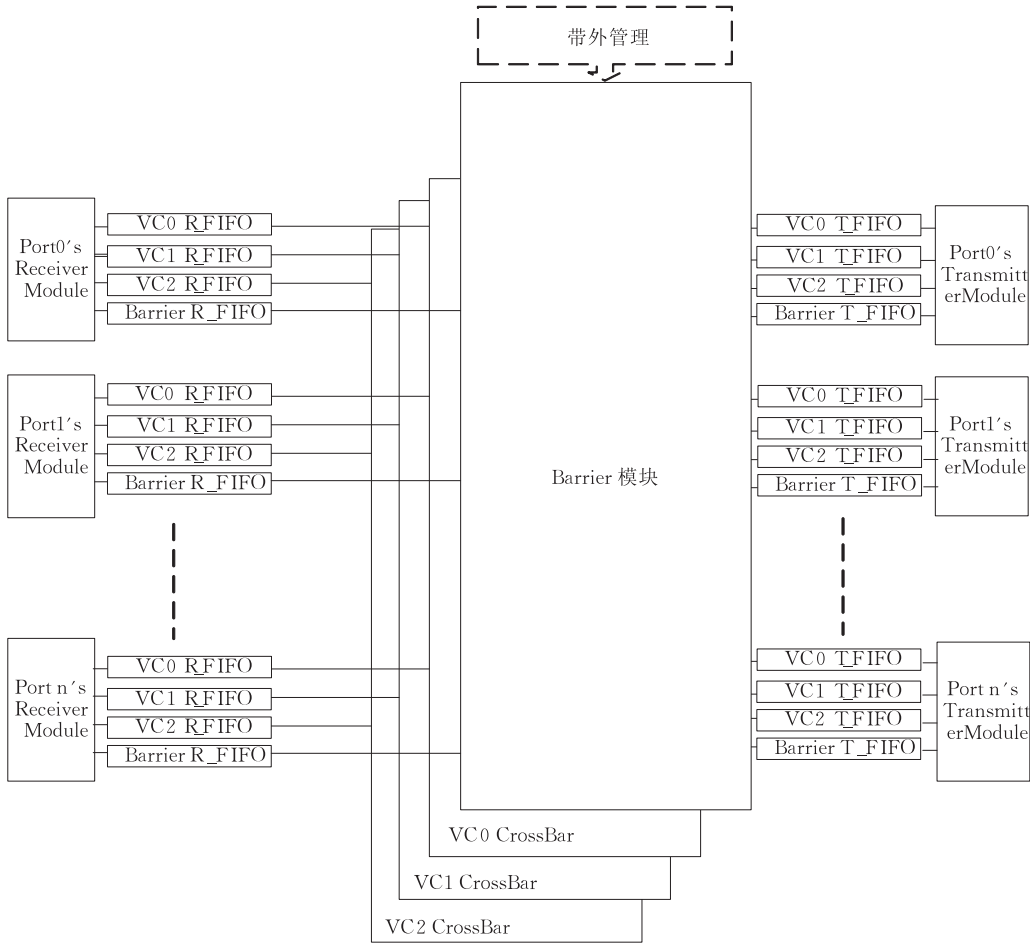


图 4 支持 Barrier 操作的交换芯片结构图

为了减少队头阻塞(HOL)对 Barrier 数据传输的影响,设置一个专用虚通道,负责所有与 Barrier 有关的数据传输.Barrier 虚通道采用与数据虚通道相同的流量控制机制和交换方式.多个虚通道在输出端口会产生对物理链路的竞争,为保证 Barrier 数据的快速传送,为 Barrier 虚通道设置最高优先级进行调度.

### 3.2 Barrier 模块

根据第 2 节中的设计,Barrier 模块负责 Barrier 包的归约和分发,并实现 Barrier 点对点可靠性机制.这些功能及操作流程均定义于本文设计的 Barrier 操作协议中,Barrier 模块是该协议的具体实现.下面将对 Barrier 协议和实现进行介绍,并对本文的 Barrier 操作协议进行可靠性证明.

#### 3.2.1 Barrier 操作协议及实现

基于之前的设计,定义 Barrier 操作相关包格式如图 5 所示,其中归约包和分发包用于 Barrier 正常操作,归约应答包和分发催促包用于 Barrier 可靠机制.

VC	Type	Seq	ID	CRC
----	------	-----	----	-----

VC: Barrier虚通道号  
Type: Barrier包类型,包括归约包、分发包、归约应答包和分发催促包。  
Seq: Barrier操作序号,区分连续两次操作。  
ID: Barrier操作组号。  
CRC: CRC校验域

图 5 Barrier 包格式

在 Barrier 模块中设置 Barrier 容错状态寄存器.如图 6 所示,分发状态位用来记录当前 Barrier



的分发状态。该状态位在 Barrier 操作完成分发时置位,下一次 Barrier 操作完成归约时复位。Barrier 序号位用来记录当前 Barrier 操作的序号,每完成一次 Barrier 操作,序号位取反。

Barrier分发状态位	Distribute Done	Barrier序号位	Sequence Number
1位	<div style="border: 1px solid black; padding: 2px; display: inline-block;">1</div>	1位	<div style="border: 1px solid black; padding: 2px; display: inline-block;">0</div>

图 6 Barrier 容错状态寄存器

基于上述设定,定义 Barrier 包中序号位为  $s$ , Barrier 模块序号位为  $s'$ ,Barrier 分发状态位为  $d$ ,约定如下规则。

**规则 1.** CRC 出错的包,不做任何处理;

**规则 2.** 当且仅当  $s=s'$ ,处理 Barrier 分发包和归约包;

**规则 3.** 当且仅当  $d=1$ ,且  $s \neq s'$ ,处理 Barrier 分发催促包,发送使用  $s$  填充的 Barrier 分发包;

**规则 4.** 当且仅当在空闲或归约阶段,处理 Barrier 归约包;当且仅当归约完成后,处理 Barrier 分发包;

**规则 5.** 当且仅当归约完成后,且  $s=s'$ ,处理 Barrier 归约应答包;

**规则 6.** 当且仅当归约完成后,启动超时计数,发送使用  $s'$  填充的分发催促和归约重传包;

**规则 7.** 对所有的 Barrier 归约包,发送使用  $s$  填充的归约应答包作为响应。

配合上述 7 项规则,Barrier 模块实现的状态机如图 7 所示,仅含有空闲、接收归约包和等待分发包 3 个状态,且状态转移条件简单,具有较低的实现复杂度。

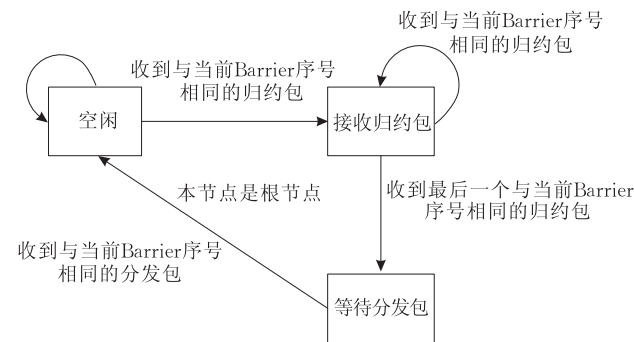


图 7 容错 Barrier 操作状态机

### 3.2.2 Barrier 操作协议可靠性证明

**定理 2.** 若 Barrier 树型结构中,任意一级的 Barrier 操作可靠,则整个 Barrier 操作可靠。

定理 2 的证明比较简单,这里不再赘述。

**定理 3.** 记序号为  $s_0$  的归约包丢失/出错为  $lc(s_0)$ ,序号为  $s_0$  的分发包丢失/出错为  $ld(s_0)$ ,收

到多余的序号为  $s_0$  的归约包为  $rc(s_0)$ ,收到多余的序号为  $s_0$  的分发包为  $rd(s_0)$ ,收到序号为  $s_1$  的归约包为  $rc(s_1)$ ,收到序号为  $s_1$  的分发包为  $rd(s_1)$ ,一级 Barrier 操作失败为  $fb(s_0)$ ,若

$$lc(s_0) \vee ld(s_0) \vee rc(s_0) \vee rd(s_0) \vee rc(s_1) \vee rd(s_1) \rightarrow fb(s_0),$$

即上述任一事件发生导致 Barrier 操作失败,则该级 Barrier 操作不可靠;反之,若上述任一事件发生都不会导致 Barrier 操作失败,则该级 Barrier 操作可靠。

证明. 对于一级 Barrier 部件,Barrier 操作受 Barrier 包驱动,产生不可靠的因素包括 Barrier 包的丢失和冗余。根据定理 1,通过枚举,获得  $\{lc(s_0), ld(s_0), rc(s_0), rd(s_0), rc(s_1), rd(s_1)\}$  错误事件全集。记每类错误造成 Barrier 操作失败的概率为  $P_i$ ,则根据离散事件概率,整个 Barrier 操作的失败概率为  $P=1-\prod_{i=1}^8 (1-P_i)$ 。

若各类错误均不会导致 Barrier 操作失败,即  $P_i=0$ ,则上式可知,  $P=0$ ,即 Barrier 操作失败概率为 0,Barrier 操作可靠,定理 3 可证。证毕。

根据定理 1,任意时刻,  $\{s_0, s_1\}$  构成 Barrier 操作的序号全集,Barrier 操作的可靠性证明只涉及序号  $\{s_0, s_1\}$ 。同时根据定理 2,证明相邻两级 Barrier 操作可靠,即可证明本文 Barrier 操作协议可靠。

证明。

(1)  $lc(s_0) \rightarrow fb(s_0)$  不成立。

归约过程采用基于请求应答的可靠性机制,当归约包真正丢失后,根据规则 6,发送方将会在一定时间后重新发送序号为  $s_0$  的归约包,因此,归约包的真正丢失不会对 Barrier 操作正确性造成影响。

(2)  $ld(s_0) \rightarrow fb(s_0)$  不成立。

分发过程采用基于超时催促的可靠性机制,当分发包真正丢失后,根据规则 6,接收方将会在一定时间后发出序号为  $s_0$  的催促包。根据规则 3,发送方作为对催促包的响应,重发序号为  $s_0$  的分发包,因此,分发包的真正丢失不会对 Barrier 操作正确性造成影响。

(3)  $rc(s_0) \rightarrow fb(s_0)$  不成立。

归约包未真正丢失时,由于重传时机问题,造成冗余重传归约包出现。根据规则 4,冗余归约包只会在归约阶段处理。归约状态下,重传归约包重复未丢失归约包的动,不对 Barrier 操作产生影响。

(4)  $rd(s_0) \rightarrow fb(s_0)$  不成立。

分发包未真正丢失时,由于重传时机问题,造成冗余重传分发包出现.序号为  $s_0$  的冗余分发包只会出现在空闲或归约阶段,这时序号为  $s_0$  的 Barrier 操作已完成,同时根据规则 4,该分发包不被处理,不对后续 Barrier 操作产生影响.

(5)  $rc(s_1) \rightarrow fb(s_0)$  不成立.

若序号为  $s_0$  的 Barrier 操作处于空闲或归约阶段,根据规则 2,不处理序号为  $s_1$  的归约包,Barrier 操作正确性不受影响;若序号为  $s_0$  的 Barrier 操作已完成归约,根据规则 4,不处理任何归约包,Barrier 操作正确性不受影响.

(6)  $rd(s_1) \rightarrow fb(s_0)$  不成立.

若序号为  $s_0$  的 Barrier 操作处于空闲或归约阶段,根据规则 4,不处理任何分发包,Barrier 操作正确性不受影响;若序号为  $s_0$  的 Barrier 操作已完成归约,根据规则 2,序号为  $s_1$  的分发包不被处理,Barrier 操作正确性不受影响.

结合上述证明,根据定理 3,可知单级的 Barrier 操作可靠,进而根据定理 2,整个 Barrier 操作可靠.

4 性能评价

根据前面的分析,曙光 5000 Barrier 网络是适合于 Fat-Tree 拓扑环境的实现方案,在理想条件下,可以获得较低的 Barrier 操作延迟.但理想情况下的延迟并不能反映实际的 Barrier 性能.影响 Barrier 延迟的因素包括软件开销、链路传输时间和单级 Barrier 处理时间,其中与本文具体实现相关的是单级 Barrier 处理时间.由于输出竞争的存在,单播数据传输对单级 Barrier 处理时间产生影响.本节采用时钟周期精确模拟的方法,对单播通信与单级 Barrier 操作间的影响进行评估.

本节测试中,单播数据包目标端口符合均匀分布,使用 16 个 Barrier 组,每组均有 15 个叶子端口节点,1 个父端口.本节的 Barrier 操作均同时涉及 16 个组,获取的 Barrier 延迟为 16 个组延迟的加合平均

$$T_{avg} = \sum_{i=0}^{15} \overline{T(b_i)} / 16,$$

其中  $\overline{T(b_i)} = \sum_{j=0}^n (T(c_j) + T(d_j)) / n$ ,  $\overline{T(b_i)}$  为第  $i$  组 Barrier 操作的平均延迟,  $T(c_j)$  为交换芯片完成一次 Barrier 归约操作时间,  $T(d_j)$  为交换芯片完成一次 Barrier 分发操作时间,  $n$  为 Barrier 操作的

次数.

在单播通信负载为 100% 的情况下,通过控制连续两次 Barrier 操作的时间间隔,调整 Barrier 执行频率,对 Barrier 行为进行测试.如图 8 所示,随着 Barrier 执行频率的增大,Barrier 的延迟逐渐下降,最终收敛于稳态.

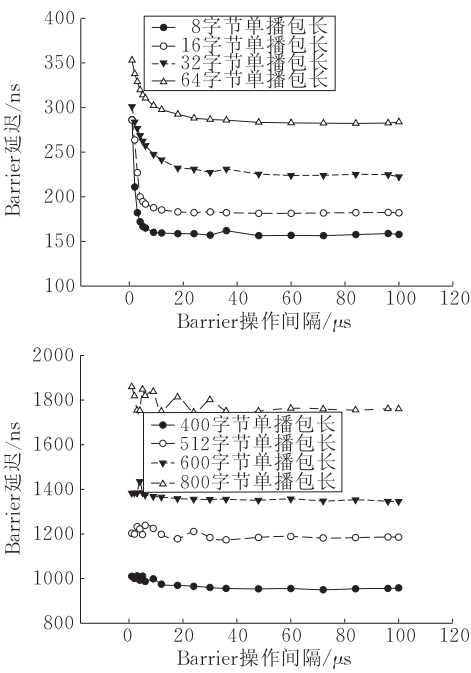


图 8 不同 Barrier 操作间隔下的 Barrier 延迟

在实际应用中,Barrier 操作的间隔在毫秒级别,因而收敛值可以反映实际 Barrier 操作的单级处理延迟.该收敛值随着单播包包长的不同而变化,见图 9 所示,在不同的单播负载率环境下,Barrier 操作延迟随单播包长的增长而线性增加.无单播通信时,存在多组 Barrier 操作竞争的情况下,单级 Barrier 操作完成归约和分发仅需 128ns(工作频率为 312.5MHz).此外,Barrier 操作在交换芯片中的延迟可根据单播通信特征进行估算.

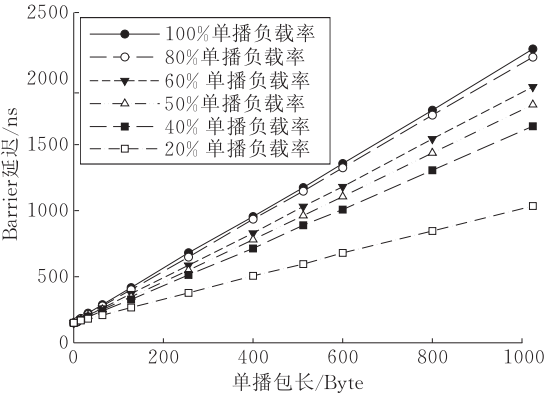


图 9 不同单播包长环境下的 Barrier 稳定延迟



由于 Barrier 虚通道拥有最高的传输优先级,因此 Barrier 也会对单播操作造成影响.在 Barrier 操作延迟达到稳定后,对不同单播包长环境下的单播带宽进行统计,得到单播负载率为 100% 时的单播带宽折损率曲线.如图 10 所示,平均单播带宽折损率为 1.14%,随着 Barrier 操作间隔的增大,以及单播注入率的下降,单播带宽折损率会继续下降,Barrier 操作对单播数据传输影响极小.

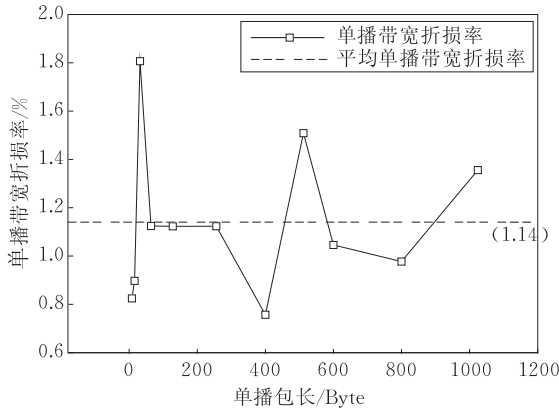


图 10 不同单播包长下的单播带宽折损率

## 5 结 论

使用共享网络实现对 Barrier 操作的加速,不仅可以获得软件实现的灵活性,还可以获得较低的 Barrier 操作延迟,相比于软件和专用网络实现,是性价比最高的方案.但现有的共享实现并没有获得在 Fat-tree 拓扑下的最优性能.

本文充分利用曙光 5000 高性能互联网络的特点,设计实现了曙光 5000 Barrier 网络.该网络支持并发的 16 个 Barrier 操作,且可由软件进行灵活的配置管理.模型分析表明,本文的 Barrier 网络是适合于 Fat-tree 拓扑的 Barrier 加速方案,可以获得较低的 Barrier 操作延迟和较好的扩展性.理论上,1024 节点规模下,一次全局 Barrier 操作只需 1.7 $\mu$ s.此外,本文对共享网络环境下,单播通信与 Barrier 操作之间的影响进行评测.模拟的结果表明,在曙光 5000 高性能网络中,Barrier 操作对单播数据的传输影响极小,单级 Barrier 操作的延迟与单播数据包长成线性关系.上层软件可根据单播通信特征,对 Barrier 操作开销进行估算.

受限于测试环境,本文未对大规模环境下的 Barrier 系统行为及 Barrier 对并行应用的影响进行评测.这部分工作将在未来的曙光 5000 高性能计

算机系统中开展.此外,曙光 5000 系统模拟器的 Barrier 网络模拟部分正在开发阶段,对 Barrier 系统行为的评测也会在模拟器中进行.

本文的设计在曙光 5000 网络验证平台上通过了 FPGA 验证.其中交换芯片正在进行 ASIC 实现,采用 0.13 $\mu$ m CMOS 工艺标准单元,系统频率在 312.5MHz.物理设计图如图 11 所示,其中 Barrier 模块面积为 625887 $\mu$ m<sup>2</sup>,仅占芯片面积的 2.1%.

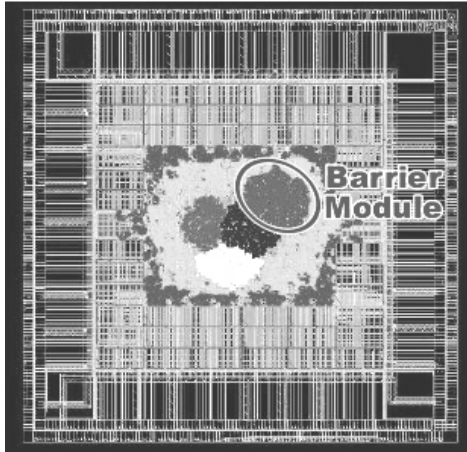


图 11 曙光 5000 交换芯片版图

## 参 考 文 献

- [1] Scott S L. Synchronization and communication in the T3E multiprocessor//Proceedings of the 7th International Conference on Architectural Support for Programming. Cambridge, MA, 1996: 26-36
- [2] Hensgen D, Finkel R, Manber U. Two algorithms for barrier synchronization. International Journal of Parallel Programming, 1988, 17(1): 1-17
- [3] Brooks D E. The butterfly barrier. International Journal of Parallel Programming, 1986, 15(4): 295-307
- [4] Scott M L et al. Fast contention-free combining tree barriers for shared memory multiprocessors. International Journal of Parallel Programming, 1994, 22(4): 449-481
- [5] Torsten H et al. A practical approach to the rating of barrier algorithms using the LogP model and Open-MPI//Proceedings of the International Conference on Parallel Processing Workshops. Oslo, Norway, 2005: 562-569
- [6] Buntinas B, Panda D K et al. Performance benefits of NIC-based barrier on myrinet/GM//Proceedings of the 15th International Parallel & Distributed Processing Symposium. San Francisco, 2001: 166-173
- [7] Gupta R et al. Efficient barrier using remote memory operations on VIA-based clusters//Proceedings of the IEEE International Conference on Cluster Computing. Chicago, 2002: 83-90

- [8] Adams D. Cray T3D system architecture overview. Cray Research Inc; Technical Report HR-040433, 1994
- [9] The BlueGene/L team. An overview of the BlueGene/L supercomputer//Proceedings of the International Conference for High Performance Networking and Computing (SC'02). Maryland, 2002; 1-22
- [10] Petrini F et al. Hardware- and software-based collective communication on the quadrics network//Proceedings of the IEEE International Symposium on Network Computing and Applications. Cambridge, MA, 2001; 24-35
- [11] Leiserson C E. Fat-Trees: Universal networks for hardware-efficient supercomputing. IEEE Transactions on Computers, 1985, 34(10): 892-901
- [12] Leiserson C E et al. The network architecture of the connection machine CM-5//Proceedings of the Symposium on Parallel Algorithms and Architectures. San Diego, 1992; 542-552
- [13] Zhou Jia-Zheng, Lin Xuan-Yi, Wu Chun-Hsien, Chung Yeh-Ching. Multicast in Fat-Tree-Based InfiniBand networks//Proceedings of the 4th IEEE International Symposium on Network Computing and Applications. Cambridge, MA, 2005; 239-242
- [14] Dunigan T H, Vetter J S. Performance evaluation of the SGI Altix 3700//Proceedings of the International Conference on Parallel Processing. Oslo, Norway, 2005; 231-240
- [15] Beecroft J, Addison D, Petrini F, McLaren M et al. Qs-NetII: Defining high-performance network design. IEEE Micro, 2005, 25(4): 34-47
- [16] Culler D et al. LogP: Towards a realistic model of parallel computation//Proceedings of the Principles Practice of Parallel Programming. San Diego, 1993, 28(7): 1-12
- [17] Moh S et al. A fast tree-based barrier synchronization on switch-based irregular networks//Proceedings of the 7th International Conference on High Performance Computing. Bangalore, India, 2000; 273-282



**CAO Zheng**, born in 1982, Ph. D. candidate. His main research interests include distributed computing, high performance computer architecture, and high performance interconnection networks.

**WANG Da-Wei**, born in 1980, Ph. D. candidate. His main research interests include architecture of high performance computer architecture, high performance interconnec-

tion networks.

**LIU Xin-Chun**, born in 1968, Ph. D. , associate professor. His main research interests include reconfigurable computing, high performance computer architecture and high performance interconnection networks.

**SUN Ning-Hui**, born in 1968, Ph. D. , professor, Ph. D. supervisor. His main research interests include architecture of parallel computer, distributed OS, and high performance computing.

## Background

This paper is supported by the National High Technology Research and Development Program (863 Program) of China project "Dawning 5000 High Productivity Computer (Project No. 2006AA01A102)". Dawning 5000 System uses HPP (Hyper Parallel Processing) architecture, which is proposed to meet challenges of petaflops computing. HPP architecture implements PGAS (Physical Globally Address Space) and supports both Message Passing and Share Memory programming models. Barrier operation, which is a kind of global synchronization operations, is widely used in both programming models. Barrier operation has global blocking semantic, as a result, it affects system performance directly. The larger system scale is, the more time is spent in barrier

operation. Optimizing barrier operations can not only lower execution time, but also benefits scalability of applications. In this case, Dawning 5000 System uses a hardware-based barrier network, which is proposed in this paper, to accelerate barrier operations. IBM, Cray and NEC implemented dedicated barrier networks in their supercomputers and had gained good performance. However, take cost-effective into account, embedded barrier network is suitable to Dawning 5000 System. Compared with other embedded barrier networks, Dawning 5000 Barrier Network can gain much better performance for fat-tree topology. In ideal situation, the barrier operation of 1024 nodes can be finished within 1.7 microseconds.