基于 ibdump 的 InfiniBand 网络拥塞控制观测方法研究

曹光权 张子文 孙志刚 陈洪义 胥庆杰
(国防科学技术大学计算机学院 长沙 410073)

摘 要 在 InfiniBand(IB)体系网络中,拥塞控制(Congestion Control,CC)能够确保高性能和资源利用率,避免拥塞传播对于无辜流的性能损害。首先分析 IB 网络采用的 ECN(Explicit Congestion Notification)拥塞控制机制,然后提出一种集中控制的多点流量发生器 CTBG(Central Traffic Behavior Generator),它提供了对流量的统计能力。为了进一步剖析 IB 网络拥塞控制的细粒度行为,提出了基于 ibdump 和 wireshark 的观测方法。实验表明,提出的测量机制能够细粒度、低开销地观测 IB 网络的拥塞控制行为,其对拥塞控制机制的研究具有重要的指导意义。

关键词 InfiniBand,拥塞控制,观测,ECN,ibdump
中图法分类号 TP393 **文献标识码** A

Research on Observation Mechanism for Congestion Control in InfiniBand Network Based on ibdump

CAO Guang-quan ZHANG Zi-wen SUN Zhi-gang CHEN Hong-yi XU Qing-jie
(College of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract Congestion control in InfiniBand network can guarantee high performance and resource utilization. It avoids performance damage of congestion spreading to victim flow. This paper first introduced the InfiniBand congestion control mechanism of ECN (Explicit Congestion Notification). Then, a new tool named CTBG (Central Traffic Behavior Generator) was proposed to generate congestion traffic from multiple nodes and analyze the result in a central controller mode. In order to further investigate the congestion control behavior of InfiniBand, we proposed the new observation method based on ibdump and wireshark. The experiment indicates the observation method in this paper can observe the InfiniBand congestion control behavior with fine granularity and low cost. This method has important significance for the research of the InfiniBand congestion control mechanism.

Keywords InfiniBand, Congestion control, Observation, ECN, ibdump

1 前言

IB 是诞生于 1999 年的一种高带宽低延时互连交换体系结构。IB 与以太网相比具有更强的异构互联能力,在延时、带宽、虚拟化和管理配置等方面都有显著的提高。在 2011 年底最新的 top500 高性能计算机排名中,全球最快的 500 台计算机中已经有 42.6%使用 IB 互连[1],这表明 IB 已经成为最主流的高性能计算机互连技术。

在 IB 网络拥塞控制方面,Jose 等人最早对其 ECN 机制进行研究,提出异步的数据包标记行为能够改进公平性,放宽的速率增加条件能够改进静态和动态流量场景下的吞吐量[2]。Pfister 等人通过模拟验证 IB 拥塞控制如何在胖树结构下解决热点链路的拥塞问题[3]。Gran 等人对硬件实现的 IB 拥塞控制进行分析实验,结果表明通过调整参数能够有效地解决拥塞、提高 benchmark 性能,并通过解决停车位问题可改进公平性[4]。他们在文献[5]中又研究了在不丢包网络中拥塞控制、交换机仲裁和公平性之间的关系。

如何观测真实网络中的拥塞是研究 IB 网络拥塞控制的基础,这方面的研究目前相对较少,在文献[3,4]中也没有详细描述。本文对真实 IB 网络环境下如何观测拥塞控制行为进行了深入研究。主要创新包括:(1)为产生网络拥塞,设计实现了一个集中控制的多点流量发生器 CTBG,该软件可以同步或异步产生网络流量,进行网络性能评测和统计;(2)提出基于 ibdump 的观测 IB 网络拥塞行为的方法;(3)在真实环境中观测 IB 网络拥塞行为,对观测到的拥塞行为进行深入分析。上述研究对于进一步分析 IB 网络拥塞控制各项参数的影响和拥塞控制的 ECN 处理算法具有重要意义。

第 2 节描述 IB 网络拥塞控制原理;第 3 节描述基于 ibdump 的实验方法;第 4 节是实验结果分析;最后给出结论和下一步的工作。

2 IB 网络拥塞控制原理

2.1 IB 网络拥塞控制架构

IB 网络中的拥塞控制架构如图 1 所示。从宏观的角度

到稿日期:2012-06-09 返修日期:2012-09-20 本文受国家"863"重点基础研究发展规划(2011AA01A10)资助。
曹光权(1981—),男,硕士生,主要研究方向为高性能互连,E-mail:59904734@qq.com;张子文(1984—),男,博士生,主要研究方向为高性能互连;孙志刚(1973—),男,博士,研究员,主要研究方向为新型互联网体系结构、高性能路由器、流媒体传输机制。

来看，拥塞控制操作可以分为三步：第一步，当交换机检测到拥塞发生，交换机会将 IB 数据包的 BTH(Base Transport Header)中 FECN(Forward Explicit Congestion Notification)域置为 1(①)，进行 FECN 标识。第二步，当数据包到达目标 HCA(Host Channel Adapter)(②)，目标 HCA 会向源 HCA 返回带 BECN(Backward Explicit Congestion Notification)标识的 CNP(Congestion Notification Packet)(③)。第三步，当源 HCA 收到 CNP(④)，它会逐步减少发送速率，从而减少拥塞(⑤)。随着时间的推移，源 HCA 会逐渐恢复发送速率。

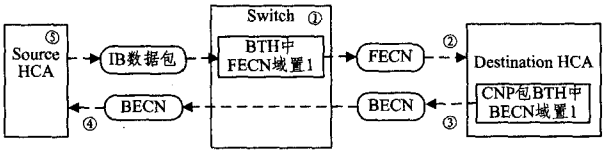

图 1　IB 网络中的拥塞控制架构

IB 网络的拥塞控制需要 IB 交换机和 HCA 都支持才能选择开启，IB 交换机和 HCA 维持着有关拥塞控制的一系列参数，这些参数决定着交换机什么时候检测到拥塞、以什么样的比率进行 FECN 标记、HCA 降低发送速率的速度和时长，如果这些参数进行了合理的设置，网络就会很好地解决拥塞问题，避免头堵塞(head-of-line blocking)，从而更充分利用网络资源。

IB 交换机中涉及拥塞控制的参数主要有 Threshold、Marking_Rate、Packet_Size(见表 1)，Threshold 参数决定 FECN 标记的强制性，其范围是 0 到 15，0 代表经过某个端口的数据包不进行拥塞标记，1 表示很高的 Threshold，会造成端口 Virtual Lanes(VL)进入拥塞状态太迟缓，很可能造成拥塞扩散，15 表示很低的 Threshold，相比设置为 1 的情况只有较小可能性造成拥塞扩散，但是代价是交换机没有真正拥塞时端口 VL 就可能已经处于拥塞状态[4]。2－14 是居于 1 和 15 之间递减的均匀分布；Marking_Rate 参数决定 FECN 标记的比率；Packet_Size 决定通过交换机某个端口的数据包大小。

表 1　拥塞控制的主要参数[6]

设备	参数	默认值[取值范围]
IB 交换机	Threshold	0xf [0-0xf]
	Marking_Rate	0xa [0-0xffff]
	Packet_Size	0x200 [0-0x3fc0]
HCA	CCTI_Increase	1
	CCTI_Limit	0
	CCTI_Min	0
	CCTI_Timer	0

在 HCA 中维护着一个 CCT(Congestion Control Table)，表项中的值表示逐步降低的发送速率，CCTI(Congestion Control Table Index)是 CCT 的指针，相关的参数主要有 CCTI_Increase、CCTI_Limit、CCTI_Min、CCTI_Timer(见表 1)，CCTI_Increase 表示 CCTI 增加的数量，决定 HCA 降低发送速率的速度；CCTI_Limit 是 CCTI 的上限，CCTI 不能比这个值更大；CCTI_Min 是 CCTI 的下限，默认为 0，表示初始时不降低发送速率；CCTI_Timer 决定 CCTI 的恢复时间，每当经过设置值的时间后，CCTI 会减少 1，HCA 进行发送速率恢复。

2.2　交换机和 HCA 拥塞行为

交换机行为包括：(1)拥塞检测。当超过 Threshold 时，交换机端口能识别它已进入端口 VL 拥塞状态，可以为每个端口设置不同级别的 Threshold。将 Threshold 设置为 1 则意味着将消耗 VL 大多数的缓冲区，而将 Threshold 设置为 15 则必须要考虑到 VL 的信用量(credits)。(2)分辨 root 和 victim 拥塞状态。当拥塞发生时，交换机要能够分辨出具体的拥塞状态，当交换机端口的输出 VL 超过 Threshold 并且有信用量用于发送数据，则进入 root 拥塞状态；当交换机端口的输出 VL 超过 Threshold 但是由于缺少信用量而延迟发送数据，则进入 victim 拥塞状态[6]。(3)拥塞标识动作。当交换机端口进入 root 拥塞状态或 victim 拥塞状态，交换机根据具体的参数设置，将数据包中 BTH 中 FECN 域置为 1，进行 FECN 标识。

HCA 行为指当 HCA 接收到 CNP 时，CCTI 根据 CCTI_Increase 的值增加，HCA 会以 CCTI 指向的 CCT 值对应的速率来进行数据发送，CCTI 的初始值是 CCTI_Min，CCTI 的上限是 CCTI_Limit，在单位时间内 HCA 接收到的 CNP 越多，发送速率就降低得越快。发送速率根据 CCTI_Timer 的设置值进行周期性的恢复，CCTI_Timer 的范围在 1.024μs 到 67ms 之间，当时间周期到达时，每一条与此 CCTI_Timer 关联的数据流的 CCTI 减 1，当 CCTI 到达 0 或 CCTI_Min 时，所有的数据流的发送速率都恢复到初始状态[6]。

观测 IB 网络的拥塞控制行为的主要难点在于：(1)IB 网络带宽很大，如何在很短的时间内发送数量流来产生拥塞；(2)IB 网络接口速率很高，如何快速地在大量数据中观测拥塞。为此，我们设计了集中控制的多点流量发生器 CTBG 来产生突发数据流造成拥塞，利用 ibdump 在线捕获 IB 数据包，采用 CTBG、wireshark 工具离线分析的方法对拥塞控制行为进行分析。

3　基于 ibdump 的实验方法

3.1　实验环境

硬件包括 IBS108Q 交换机和 IB Adapters。IBS108Q 交换机是基于 IBS216Q 交换机的精简版本，按照“天河(TH)-1”超级计算机系统要求，提出了正交结构设计、高速信号传输设计等，端口数为 108，端口速度为 4xQDR(40Gbps)，在 TH-1 系统满负荷、长时间的测试中表现稳定，具有较高的可靠性[8]。IB Adapters 是基于 Mellanox 公司 MT26428 芯片自行研发生产的 HCA，端口速度为 4xQDR。IBS108Q 和 IB Adapters 均支持拥塞控制。计算机结点包括 5 台宝德(PowerLeader) PR1760T 服务器，每台服务器含两枚 Intel Xeon E5540 双核 CPU、16GB ECC RAM、两个 8x PCIe 2.0 插槽、400GB 硬盘，均运行 RedHat Linux 6.1 x86_64 操作系统，使用 Mellanox OFED(Open Fabrics Enterprise Distribution)1.5.3。

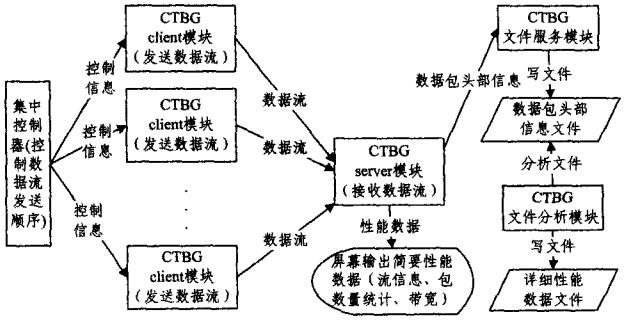

图 2　CTBG 总体设计

软件主要包括：

(1)CTBG 软件主要用于测试 IB 网络 IP over IB 性能，使用多线程技术，基于 C-S 模式，主要模块包括集中控制、client、server、文件服务、文件分析等模块。总体设计如图 2 所示，集中控制模块用于控制各个 client 模块的发送时序，可以同步或设定时间间隔延迟运行；client 模块用于尽最大能力向目的 HCA 发送数据流，发送时记录每个数据包的发送时间，当最后一个与首个数据包的时间差达到设定的发送时间，模块终止运行；server 模块用于接收 client 模块发送的数据流，它会监听默认端口，每当收到一个连接请求，就会开启一个子进程处理数据流，从而达到处理多 client 模块的目的，同时 server 与文件服务模块建立连接，将接受到的数据包的头部转发给文件服务模块处理；文件服务模块用于将 server 模块发送的所有数据存入指定文件；文件分析模块负责分析文件服务模块生成的文件，得到尽可能详细的流量统计信息。

(2)ibdump 是 Mellanox OFED 中的一个程序组件，专用于捕获 Mellanox 公司 ConnectX 系列 HCA 产品的数据包，生成的数据包可以通过 wireshark 工具进行图形化的数据分析。主要参数有-b、-mem[7]。当使用-b 选项时，如 ibdump -b 12 表示 ibdump 会同时不丢包地对 2^{12} 个 MTU 大小的包进行捕获，但是会消耗 2^{12} * MTU Bytes 的内存；当使用-mem 选项时，如 ibdump -mem 5GB 表示 ibdump 首先会直接占用 5GB 内存，再将捕获的数据包写入内存中，当写入的数据达到 5GB 大小时停止运行，最后将内存的数据写入硬盘指定文件。这种工作方式会减少丢包，但会额外消耗大量计算机资源。

(3)wireshark 是一种开源的网络数据包分析工具，可以尽可能详细地显示数据包的情况，最新的 1.7.1 版本能够支持 IB 数据包的分析。

3.2 实验方法

图 3 中 S1 和 S2 为 2 台交换机，H1 到 H5 为 5 台终端。为了便于产生拥塞，通过刷新交换机固件(firmware)的方式将网络速率限制为 SDR(10Gbps)，H1 和 H2 连接到交换机 S1，H3、H4、H5 连接到交换机 S2。按照实验拓扑，在 H1、H2、H3、H5 结点上部署 CTBG 软件的 client 模块，集中向 H4 发送数据流，后台运行 ibdump，用于捕获 CNP；在 H4 结点中部署 CTBG 软件的 server、集中控制、文件服务和文件分析模块，用于接收数据流，集中控制 client 模块发送数据流，生成数据包头部信息文件和分析文件，同时在 H4 结点上还后台运行 ibdump，捕获接收到的 IB 数据包。CTBG 程序运行完成后，使用文件分析命令对数据包头部信息文件进行分析，得到详细的流量统计信息，再使用 wireshark 程序对 ibdump 捕获的 IB 数据包进行分析。实验参数配置如表 2 所列。

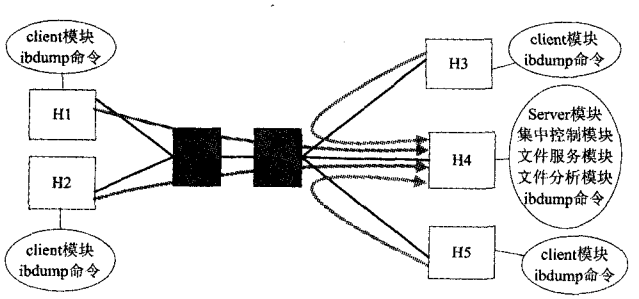

图 3 实验方法

表 2 实验参数配置[4]

设备	参数	配置值
IB 交换机	Threshold	0xf
	Marking_Rate	0xl
	Packet_Size	0x8
IB HCA	CCTI_Increase	1
	CCTI_Limit	127
	CCTI_Min	0
	CCTI_Timer	150μs

4 实验结果分析

4.1 关闭拥塞控制和开启拥塞控制的对照

图 4 描述的是在图 3 所示实验方法下关闭拥塞控制时 4 条数据流的吞吐量情况。我们逐条加入数据流，来观察对网络性能的影响。实验开始时只有 H1 在传输数据，它占据所有的带宽，H3 随后加入与 H1 平分带宽，在 H2 加入后，远端的 H1 和 H2 占用的带宽和与近端的 H3 占用的带宽是一致的，H5 最后加入后，H3 和 H5 各占用网络总带宽的 1/3，H1 和 H2 平均分配剩余的 1/3，也就是说近端的结点发送速率是远端结点的 2 倍，从网络应用的角度来看，显然对 H1 和 H2 是不公平的。

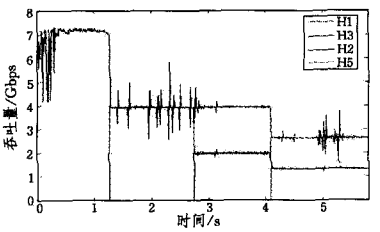

图 4 关闭拥塞控制

图 5 描述的是在图 3 所示实验方法下开启拥塞控制时 4 条数据流的吞吐量情况。当 H3 加入网络时，S2 交换机触发拥塞控制机制，交换链路出现拥塞，S2 交换机按表 2 所列的交换机参数配置进行 FECN 标识。H4 接收到带 FECN 标识的数据后，向 H1 和 H3 发送带 BECN 标识的 CNP，通知 H1 和 H3 降低发送速率来减少拥塞。H1 和 H3 按表 2 所列的 HCA 参数配置来进行降速，每收到一个 CNP，H1 和 H3 中的 CCTI_Increase 增加 1，直到 127 为止。初始时占用带宽大的 H1 接收到较多的 CNP，以较快的速度降低发送速率，而占用带宽小的 H3 接收到较少的 CNP，以较慢的速度降低发送速率，直到 H1 和 H3 速率一致。在这个降速的过程中，每经过 CCTI_Timer 设定 150μs，CCTI_Increase 减少 1，进行发送速率恢复，直到 0 为止。随着发送速率的降低和恢复，可以发现 H1 和 H3 的吞吐量数据出现了规律性的振荡，随后加入的 H2、H5 与之前 H1、H3 出现一致的情况。通过实验，我们验证了开启拥塞控制后，网络公平地对待每条数据流，实现了应用的公平性。

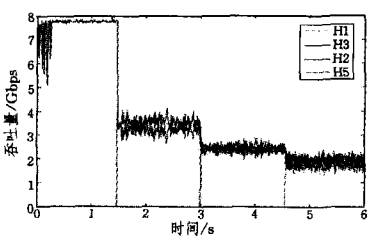

图 5 开启拥塞控制

4.2 FECN 和 BECN 的观测分析

对 FECN 和 BECN 的观测分析主要就是使用 wireshark 对 ibdump 捕捉的数据包进行分析，各个数据流的 FECN 标记情况是均匀分布的(见图 6)。由于 ibdump 程序只捕捉接收的数据包，不捕捉发送的数据包，在 H4 捕捉到的数据包可以观察到 FECN 对照，而 BECN 只能在各个发送端才能观察到。通过分析可知，当拥塞刚刚发生时，占据带宽大的数据流会在很短的时间(330μs 左右)接收到比占据带宽小的数据流相对多的 CNP，更快地降低发送速率，而占据带宽小的数据流收到相对少的 CNP，较慢地降低发送速率，当二者的发送速率接近一致时，FECN 和 BECN 都出现了稳定的分布，从而实现带宽的公平分配。实验中，H1 首先发送数据流，H2 在 1 秒后开始发送数据流，当拥塞发生时，在 330μs 左右的时间内，H1 收到了 25 个 CNP，而 H2 只收到了 10 个 CNP。

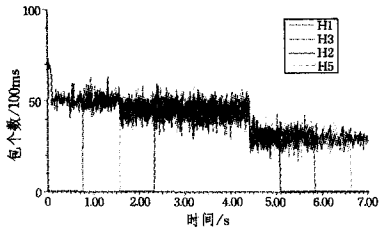

图 6 FECN 观测

4.3 ibdump 程序参数设置对 IB 网络性能的影响

表 3 no ibdump、ibdump -b、ibdump --mem 对照

参数		平均流量(Mb/s)
-b	2	6629.984256
	3	**6346.9363**
	4	6739.66387
	5	6573.76563
	6	6493.04474
	7	6692.145152
	8	**6806.667264**
	9	6721.32301
	10	6702.158848
	11	6686.116864
	12	6580.53632
--mem 5GB		**6083.919457**
no ibdump		**7327.796224**

表 3 是 no ibdump、ibdump -b、ibdump --mem 3 种情况下的流量对照，从中可以发现 ibdump 程序参数对 IB 网络性能的影响都很大。对于-b 参数，参数值为 3 时网络性能最差，平均流量只有 no ibdump 情况下的 86.61%；参数值为 8 时网络性能最好，但也只有 no ibdump 情况下的 92.89%。对于--mem 参数，因为实验中突发流量很大，5 秒左右的发送会产生约 4.4GB 的数据，实验中设定的参数为 5GB，ibdump 直接占用 5GB 的内存用于存储捕捉到的数据包。由于需要频繁地捕捉数据包，并储存到内存中，虽然减少了数据包丢失，但是系统的开销过大，对网络性能造成了很大的影响，平均流量只有 no ibdump 情况下的 83.04%，比 ibdump -b 3 的性能还要差。

结束语 实验分析表明，基于 ibdump 的 IB 网络拥塞控制机制和拥塞行为观测实验方法能很好地验证 IB 网络的拥塞控制机制，有效地解决拥塞问题，实现良好的公平性。同时对实验中 ibdump 各项参数对网络的性能影响也进行了分析，进行了 ibdump 参数设置的优化。

下步将继续研究分析 IB 网络拥塞控制各项参数的影响和拥塞控制机制的 ECN 处理算法。

参 考 文 献

[1] Top 500 supercomputer sites[OL]. http://top500.org/

[2] Santos J R, Turner Y, Janakiraman G J, et al. End-to-end congestion control for InfiniBand[C]// INFOCOM. 2003

[3] Pfister G, Gusat M, Craddock D, et al. Solving Hot Spot Contention Using InfiniBand Architecture Congestion Control[J]. Invited paper in High Performance Interconnects for Distributed Computing, July 2005

[4] Gran E G, Eimot M, Reinemo S-A, et al. First Experiences with Congestion Control in InfiniBand Hardware[C]// IPDPS' 10. 2010

[5] Gran E G, Zahavi E, Reinemo S-A, et al. On the Relation between Congestion Control, Switch Arbitration and Fairness[C]// CCGRID'11. 2011

[6] InfiniBand Architecture Specification[S]. Release 1.2.1, InfiniBand Association, 2007. http://www.InfiniBand.org, 2007

[7] Mellanox OFED for Linux User Manual Rev 1.5.3[M]. Mellanox Technologies. Ltd, 2011. http://www.mellanox.com,

[8] 吕高峰，苏金树，孙志刚，等. IBS216 交换机设计与实现[J]. 计算机研究与发展，2011，48：1-9

(上接第 34 页)

[5] Dimitrakopoulos G, Galanopoulos K, Mavrokefalidis C, et al. Low-Power Leading-Zero Counting and Anticipation Logic for High-Speed Floating Point Units[J]. IEEE Transactions on Very Large Scale Integration System, 2008, 16(7)

[6] Lee K T, Nowkda K J. 1GHz leading-zero anticipator using independent sign-bit determination logic[C]// Symposium on VLSI Circuit Digest of Technical Papers. 2000

[7] Zhang Ge, Hu Wei-wu, Qi ZI-chu. Parallel Error Detection for Leading Zero Anticipation[J]. J. Comput. Sci. &Technol, 2006, 21(6): 901-906

[8] Yao Tao, Gao De-yuan. A Novel Concurrent Error Detection Circuit for Leading Zero Anticipator[C]// 2nd International Conference on Computer Engineering and Technology. 2010

[9] Oklobdzija V. An Algorithmic and Novel Design of a Leading Zero Detector Circuit: Comparison with Logic Synthesis[J]. IEEE Transactions on VLSI System, 1993, 2(1): 124-128

[10] Hinds C N, Lutz D R. A Small and Fast Leading One Predictor Corrector Circuit[C]// Asilomar Conference on Signals, Systems and Computers. 2005: 1181-1185

[11] Oberman S F, Flynn M J. A variable Latency Pipelined Floating-point Adder[R]. CSL-TR-96-689. Stanford University, 1996

[12] 黄迟. 64 位高速浮点加法器的 VLSI 实现和结构研究[D]. 上海：复旦大学，2004

[13] 王颖. 高性能 CPU 中浮点加法器的设计与实现[D]. 上海：同济大学，2005