集群网络评测模型的新探索*

唐 渊[1,2+], 孙家昶[1,2], 张云泉[1,2], 张林波[3]

[1](中国科学院 软件研究所 并行计算实验室,北京 100080)

[2](中国科学院 软件研究所 计算机科学国家重点实验室,北京 100080)

[3](中国科学院 数学与系统科学研究院 科学与工程计算国家重点实验室,北京 100080)

New Consideration on the Evaluation Model of Cluster Area Network

TANG Yuan[1,2+], SUN Jia-Chang[1,2], ZHANG Yun-Quan[1,2], ZHANG Lin-Bo[3]

[1](Laboratory of Parallel Computing, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[2](State Key Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China)

[3](Academy of Mathematics and System Science, The Chinese Academy of Science, Beijing 100080, China)

+ Corresponding author: Phn: +86-10-62542214, E-mail: tang@mail.rdcps.ac.cn, http://www.rdcps.ac.cn/~tang

Received 2003-06-26; Accepted 2004-03-17

Tang Y, Sun JC, Zhang YQ, Zhang LB. New consideration on the evaluation model of cluster area network. *Journal of Software*, 2005,16(6):1131–1139. DOI: 10.1360/jos161131

Abstract: Traditional Cluster Area Network(cLAN)'s evaluation model takes only latency, bandwidth, routing, congestion, network topology and some related aspects into consideration. Are these factors ENOUGH to describe the real applications' communication behavior or predict its performance on cLAN? In the large quantity of NAS Parallel Benchmarks' tests(version 2.4) on a modern supercomputer——DeepComp 1800, which is of LINUX Cluster architecture, it is found that the real performance of cLAN could be greatly affected by a special communication pattern(LU pattern). Further investigation reveals that the cLAN's capacity of dealing with LU mode is independent of the known performance factors such as latency, bandwidth and so on. So it is necessary to take some new considerations on cLAN's evaluation model and add one new factor to reflect the abnormal phenomenon. The new model also provides some challenges in parallel algorithm design and application performance improvement on the LINUX Cluster.

Key words: cLAN (cLuster area network)'s evaluation model; NPB (NAS parallel benchmarks); LINUX cluster; communication performance evaluation; communication pattern

* Supported by the National Natural Science Foundation of China under Grant No.60303020 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2004AA104020 (国家高技术研究发展计划(863)); the National Grand Fundamental Research 973 Program of China under Grant No.G1999032805 (国家重点基础研究发展规划(973))

作者简介: 唐渊(1976－),男,上海人,博士,主要研究领域为高性能计算,高性能计算机的评测,分析与优化技术,Beowulf LINUX机群系统上的通信协议的设计、分析与实现,大规模并行数值软件的通信优化技术;孙家昶(1942－),男,研究员,博士生导师,主要研究领域为大规模科学与工程计算的方法与软件;张云泉(1973－),男,博士,副研究员,主要研究领域为大型并行数值软件,并行程序设计和性能评价,并行计算模型;张林波(1962－),男,博士,研究员,博士生导师,主要研究领域为高性能计算.

摘 要: 传统集群网络(cluster area network,简称 cLAN)的评测模型主要考虑了延迟、带宽、路由、拥塞、网络拓扑结构等因素.但这些因素是否足以描述实际应用程序在集群上的通信行为,或者对其在集群系统上的性能给出一个很好的预测呢?当对 NAS Parallel Benchmark(2.4 版本)在集群系统深腾 1800(DeepComp 1800)上进行大量测试时发现,集群网络的通信性能可以被一种特殊的通信模式(LU 模式)所严重影响.更深入的研究表明,这个影响 LU 模式的因素是独立于前面所述的如延迟、带宽、路由、拥塞、网络拓扑结构等因素的.因此有必要对集群网络的评测模型重新进行审视,并增加一个新的性能评测因子以反映这个新发现的现象.从研究结果来看,这个重新审视也将对集群系统上的并行算法设计以及实际大规模科学计算的应用程序性能的优化提供一些新的思路.

关键词: 集群网络(cLAN)评测模型; NPB;LINUX 集群系统;通信性能评测;通信模式

中图法分类号: TP393 **文献标识码**: A

Linux 集群系统作为一种新兴的超级计算的解决方案具有极高的性能价格比.相对而言,其网络部分则是最大的瓶颈.目前,有百兆 Ethernet,Myrinet[1],Dolphin's SCI,Giganet,Gigabit Ethernet[2]等可供选择的解决方案.就我们所知,传统的集群网络(cLAN)评测模型主要考虑了如延迟[3]、带宽[4]、拥塞[5]、路由[5,6]、网络拓扑[5,6]结构等因素.但这些因素是否足以描述集群系统上实际应用程序的通信行为,或者对其通信性能进行比较精确的预测呢?这是本文中想继续深入研究的一个问题.

NAS Parallel Benchmarks(NPB)是一组公认的用于评测大规模并行机/超级计算机的标准测试程序[7].2002 年 11 月 19 日,NAS 发布了 NPB2.4 beta release,引入了 Class D 和并行 I/O(parallel I/O)[8].基于这个最新版的 NPB2.4 以及我国的集群系统——深腾 1800(由中国科学院数学与系统科学研究院与联想集团共同研制,全球超级计算机 TOP500 排名第 43 位——http://www.top500.org/list/2002/11),我们做了大量测试,并发现集群网络的通信性能在一种特殊的通信模式——LU 模式(参见第 3.1 节的具体定义)下会受到极大的影响.由于深腾 1800 同时配备了 Myrinet2000 和普通的百兆 Ethernet 卡,我们对 LU 模式在这两种不同网络上的运行结果进行了比较,结果发现在 LU 模式下,Myrinet2000 的通信性能要比普通 Ethernet 差很多,甚至可以通过一定的参数调整一直差下去(同样的计算结点,同样的 MPI 源码,仅仅是底层的通信介质不同).而且,在另一台集群系统——曙光 4000L(由中国科学院计算技术研究所与曙光集团共同研制,也同时配备 Myrinet2000 与普通百兆 Ethernet 卡)上的测试结果,以及在与 Myricom 公司的交流中也证实了我们在深腾 1800 上得到的结论.

无疑,与百兆 Ethernet 相比,几乎所有已知的网络通信性能因子,如延迟、带宽、对拥塞的处理、路由、网络拓扑结构等,Myrinet 都要好得多,而且我们在深腾 1800 和曙光 4000L 上的实测结果也证实了这一点.所以,LU 模式这种反常行为的存在使我们猜测,可能还存在一些其他的网络通信性能因子,在这些因子上,Myrinet2000 是不如普通百兆 Ethernet 的.

本文的贡献主要在于:

(1) 发现集群网络的通信性能可以被一种特殊的通信模式——LU 模式极大地影响,以至于在某些情况下造成了 Myrinet2000 比普通百兆 Ethernet 要差.这也可以引起对不同集群网络(cLAN)环境下并行程序性能优化的另一角度的思考.

(2) 通过这一特殊的 LU 模式发现了一个新的集群网络通信性能评测因子,而这一新的性能评测因子是独立于已知的,如延迟、带宽、拥塞、路由等因子的.也就是说,延迟、带宽等好的网络,如 Myrinet2000 等,在这一新的性能因子上不一定也好.

(3) 试图建立一个新的集群网络通信评测模型,并将新发现的这一评测因子加入该模型,以对经典的基于 Log*P* 模型的集群网络通信评测模型作出一个必要和适当的补充.

1 测试环境

1.1 深腾1800

深腾 1800 共 256 个计算结点,2 个 I/O 结点,4 个登录结点,一个主控结点,一个副控结点.每个计算结点由双

Intel Xeon 2GHz 的处理器和 1GB 内存组成,其峰值为 4Gflops(如果使用 SSE2,可以达到 8Gflops).整个系统的浮点运算峰值为 1Tflops(如果使用 SSE2,可以达到 2Tflops).该集群系统同时配备了 Myrinet2000 和普通百兆 Ethernet(Intel pro100).

深腾 1800 的缺省编译器为:Intel® Fortran Compiler for 32-bit applications,Version 6.0,GNU Fortran 0.5.26 20000731(Red Hat Linux 7.1 2.96-98),GNU cc/gcc/g++ version 2.96.其 Myrinet2000 的 MPI 版本为 MPICH-GM 1.2.1..7b,GM 版本为 1.5.2.1;普通百兆 Ethernet 的 MPI 版本为 MPICH-1.2.4-p4mpd.

深腾 1800 的批处理作业调度系统为 OpenPBS 版本 2.3pl16.编译器缺省优化选项为:-O3.NPB 测试程序所用的随机数发生器为“randi8”.

1.2 曙光4000L

曙光 4000L 的每个计算结点为 Intel Pentium4 2.4GHz,操作系统内核为 Linux 2.4.18.网络为 Myrinet2000(LANai9),GM 版本为 1.6.3.

2 深腾 1800 上的 Log*P* 模型参数测试[9]

从图 1 和图 2 可以看出,在我们的测试平台——深腾 1800 上,Myrinet2000 的 Log*P* 模型的各个实测参数确实要比百兆 Ethernet 好得多.

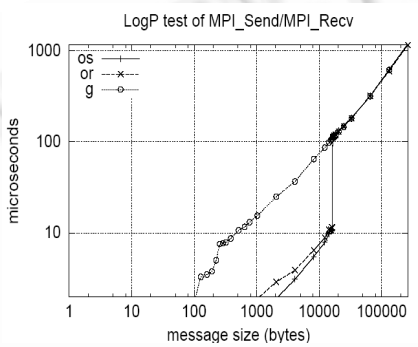

Fig.1 Log*P* parameters of Myrinet 2000

图 1 Myrinet2000 的 Log*P* 模型参数

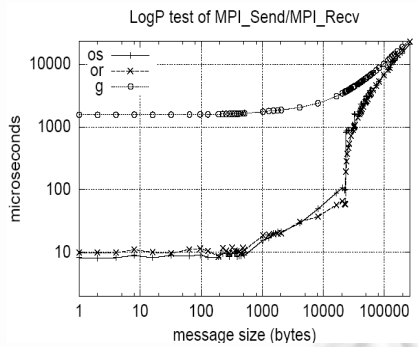

Fig.2 Log*P* parameters of Ethernet

图 2 Ethernet 的 Log*P* 模型参数

3 深腾 1800 上的 LU 程序测试

我们编译了 NPB2.4 所有程序的“CLASS C”和“CLASS D”(除了‘IS’程序没有 CLASS D 以外),并将它们分别运行在 2,4,8,…,最多到 256 个计算结点上,底层的通信介质(communication media)分别采用 Myrinet2000 和百兆 Ethernet 以进行结果之间的比较.所有的程序,除了 LU,在 Myrinet 上的运行都要比在百兆 Ethernet 上的快,这也是符合我们对其 Log*P* 模型参数的测试的.只有 LU,无论编译成几个进程运行(2,4,8,…,128/256),也无论采用多大的问题规模(CLASS=A/B/C/D)和采用什么优化选项(-O/-O2/-O3),LU 程序在 Ethernet 上的运行速度总是快于相应的 Myrinet 版本.图 3 是 NPB 的 LU 程序在测试平台——深腾 1800 上的运行结果(CLASS C/D).

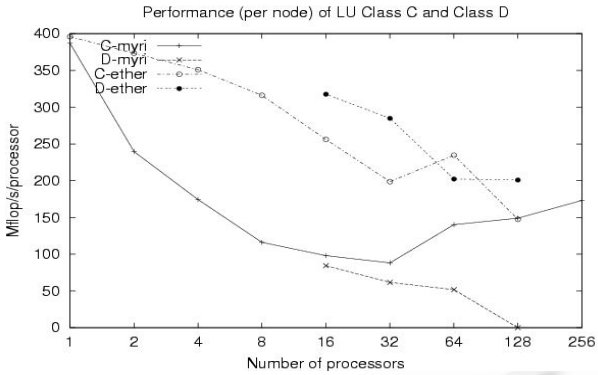

Fig.3 NPB2.4's LU on DeepComp 1800

图 3 深腾 1800 上的 LU 测试

图 3 中另一个有趣的现象是:在 Ethernet 上,最高的'Mflop/s/process'分数属于 CLASS D 的曲线,而在 Myrinet 上,CLASS C 曲线的'Mflop/s/process'较高.

3.1 LU模式

很明显,从图 3 可以看出,所有的 LU 程序在 Ethernet 上的表现要好于 Myrinet,而且还可以看到,这个现象独立于问题的规模、所编译的进程数以及编译器优化选项.更进一步的研究表明,它和 Myrinet 上的接收模式,即 MPICH-GM 的'recv mode'是'polling' 'blocking'还是'hybrid'也是无关的.

我们认为,"对网络拥塞的处理"也不是造成这一现象的原因.因为,当我们将 LU 程序只编译为 CLASS A 和仅仅两个进程的时候,其在 Ethernet 上的运行仍然快于 Myrinet.而事实上,在这种编译条件下,程序的通信行为仅仅是点对点的传递一个不超过 4Kbytes 的数据包而已.

由此,我们猜想,可能可以从 LU 程序中提取出一小段非常简单的程序代码,其中,由若干关键参数控制了 Myrinet 比 Ethernet 慢的程度.

图 4 就是我们从 NPB 的 LU 程序抽取出来的 LU 模式(LU 模式,即从 NAS Parallel Benchmarks 版本 2.4 中的 LU 程序中抽象出来的如图 4 所示的一段影响网络通信性能的关键代码),其中定义了 4 个重要的参数:

*loop*1:每个进程的外循环次数(在我们的基准程序中设为 2000).

*loop*2:每个进程的内循环次数(在我们的基准程序中设为 60).

msgSize:所传递的消息大小,在整个程序执行过程中保持不变(在我们的基准程序中设为 4000 Bytes).

compLoop:定义为一个局部函数 tm_comp_us()中所执行的浮点运算次数,该参数决定了每个内循环中除了 MPI 语句外的浮点运算次数,也即大致的 MPI 语句之间的时间延迟(在我们的基准程序中设为 10^8).

另外,我们需要注意的是,在图 4 的 LU 模式中给出的 MPI 通信模式为标准的阻塞式通信('MPI_Send'/'MPI_Recv').

3.2 寻找原因

为了寻找产生 LU 模式这种奇怪现象的原因,我们对图 4 中得到的 LU 模式进行了深入跟踪,并得到了如下一些统计结果,见表 1.其中的'send time','recv time','comp time'指的是程序中所有'发送'、'接收'以及'tm_comp_us()'操作时间的总和.*.base 指的是当 4 个参数的值正如图 4 中所定义时的程序执行结果;*.discard 指的是当所有的'tm_comp_us()'操作被抛弃,即图 4 的每个内循环中除了 MPI 操作以外,没有多余的浮点操作时的程序执行结果.

```
#define loop1 2000
#define loop2 60
#define msgSize 4000
#define compLoop 10^8
```

Process 0	Process 1
Startwtime = MPI_Wtime();	*Startwtime* = MPI_Wtime();
For (*j* = 1; *j* <= *loop*1; *j*++) {	For (*j* = 1; *j* <= *loop*1; *j*++) {
For (*i* = 1; *i* <= loop2; *i*++) {	For (*i* = 1; *i* <= *loop*2; *i*++) {
Tm_comp_us(*compLoop*);	MPI_Recv(*msgSize* * MPI_CHAR)
MPI_Send(*msgSize* * MPI_CHAR);	Tm_comp_us(*compLoop*);
}	}
For (*i* = 1; *i* <= loop2; *i*++) {	For (*i* = 1; *i* <= loop2; *i*++) {
MPI_Recv(*msgSize* * MPI_CHAR);	Tm_comp_us(*compLoop*);
Tm_comp_us(*compLoop*);	MPI_Send(*msgSize* * MPI_CHAR);
}	}
}	}
Endwtime = MPI_Wtime();	*Endwtime* = MPI_Wtime();

Fig.4 LU pattern

图 4 LU 模式

Table 1 Statistics comparison of LU pattern

表 1 LU 模式的统计比较

Process	Total time (s)	Send time(s)	Recv time(s)	Comp time(s)
Proc 0.ether.base	483.39	2.87	15.66	462.36
Proc 1.ether.base	483.39	2.94	14.24	463.73
Proc 0.myri.base	721.46	0.40	254.61	463.62
Proc1.myri.base	721.52	0.42	247.03	470.46
Proc 0.myri.discard	4.60	0.45	2.48	-
Proc 1.myri.discard	4.60	0.45	2.10	-

从表 1 中我们可以看到,尽管 Myrinet 版本的总运行时间要长,但其‘send time’仍比 Ethernet 的要短(注,由于在 Myrinet 和 Ethernet 上运行的 MPI 程序源代码完全相同,所以它们也应该执行了相同数目的‘MPI_Send’/‘MPI_Recv’).造成 Myrinet 比 Ethernet 慢的主要操作在‘接收操作’,即‘MPI_Recv’上.

如果我们将图 4 中的‘tm_comp_us()’操作全部去掉,即得到表 1 中‘*.discard’行所示的结果.而这两行的结果显示了当‘tm_comp_us()’操作全部去掉后,Myrinet 的‘recv time’又变得非常好了,符合从图 1 和图 2 中所测得的 Log*P* 参数对比得到的结果.而‘tm_comp_us()’代表的是 MPI 进程之间进行消息传递时所能感受到的额外的延迟,或者说对应的‘MPI_Send’与‘MPI_Recv’之间的不匹配的程度.所以,显而易见,在我们的测试平台——深腾 1800 上,如果没有消息传递时所插入的额外延迟——‘tm_comp_us()’的话,Myrinet 本身的发送与接收并没有什么问题.但是,当插入同样长度的延迟——具有同样 compLoop 参数的‘tm_comp_us()’操作后,Myrinet 的‘MPI_Recv’所受到的影响却远远大于 Ethernet.

现在,根据已经得到的部分结果,我们可以至少排除以下 3 种原因[10]的影响:

(1) 由所发送消息长度的不同决定的所使用的协议不同:表 1 中,*.base 和*.discard 两种版本所传递的消息长度保持 4KBytes 不变.所以,这两个版本在各自的 Myrinet(MPICH-GM & GM)或 Ethernet(MPICH & TCP/IP)上所使用的消息传递协议应该也保持不变,根据《MPI Specification》[11]的语义,对标准的阻塞式通信,加入和不加入 MPI 操作之间额外的延迟‘tm_comp_us()’,不应该对程序的通信行为有任何影响.而且,所传递的消息长度很小,才 4KBytes,无论是在 Myrinet(MPICH-GM & GM)还是 Ethernet(MPICH & TCP/IP)都处于使用短消息直接发送协议的范围中,还远远达不到使用,如 Rendezvous Protocol 或一些同步发送协议的消息长度.

(2) 竞争/加锁:在图 4 中的‘tm_comp_us()’操作中涉及的数据与其他 MPI 操作数据无关,没有任何数据间交互或者依赖现象存在.

(3) Memory 效应:在图 4 中的所有发送/接收的消息都重复使用同样的字节数组——‘*sendMsg*’/‘*recvMsg*’,其情形与简单的 ping-pong 测试完全相同.

在排除了以上这 3 个可能的原因之后,还有一个可能的原因:“接收操作←→同步延迟”——不同循环发送的消息之间可能互相影响,甚至造成拥塞[10].从图 4 中我们也确实可以看到一些发送与接收操作之间的不匹配,

其中增加了‘tm_comp_us(*compLoop*)’的计算延迟.那么 LU 模式的存在是否说明 Myrinet 对消息传递间加入的这个计算延迟,也就是发送与接收操作之间的不匹配更敏感呢?

如果‘*msgSize*’,‘*loop*1’,‘*loop*2’固定为‘4KBytes’,‘2000’,‘60’,那么增加‘*compLoop*’,即图 5 中所示的‘tm_comp_us’ unit,也就是增加了对应的‘发送’与‘接收’操作之间不匹配的程度/延迟,就会造成‘接收时间’(recv time)不同程度的增加.而且,从图 5 的曲线来看,对应于相同的‘*compLoop*’增加量,Myrinet 的‘接收时间’(recv time)的增加明显要比 Ethernet 快.具体表现为图 5 中代表 Myrinet ‘recv time’的实心圈曲线的斜率远远高于代表 Ethernet ‘recv time’的小叉曲线.另外,我们也可以看到,无论 Myrinet 还是 Ethernet,随着‘*compLoop*’的增加,主要是‘接收时间’(recv time)相应增加,‘发送时间’(send time)基本不变.也就是说,无论‘*compLoop*’如何增加,可以观察到如下公式成立:

$$\text{recv\_time.myri} - \text{recv\_time.ether} = \text{total\_time.myri} - \text{total\_time.ether} \quad (1)$$

现在,我们也可以解释图 3 中另一个奇怪的现象:即为什么在 Ethernet 上,LU 的 Class D 的‘Mflop/s/process’成绩最高,而在 Myrinet 上却是 Class C 最高?这显然是因为 Class D 的问题规模更大,则对应于图 4 中‘*compLoop*’的计算延迟也更大.而 Myrinet 对这个‘*compLoop*’的计算延迟更为敏感,使得其 Class D 的‘接收时间’(recv time)要大于 Class C 时的情况,并使得其总时间也相应延长.

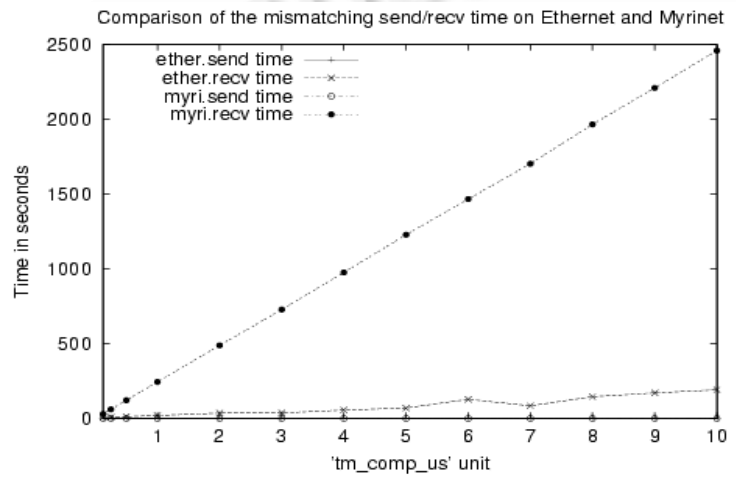

Fig.5 Comparison of the sensitivity to the mismatch timing of send/recv operations between Myrinet and Ethernet

图 5 Myrinet 和 Ethernet 对消息发送/接收间不匹配的敏感度的比较

除了‘*compLoop*’以外,实验表明,另外 3 个参数: ‘*msgSize*’,‘*loop*1’,‘*loop*2’对 LU 模式也有不同的影响,见表 2.表 2 中,‘*compLoop*’参数一直固定为 10^8(次浮点运算).第 1、第 2 行数据表明,当其余 3 个参数固定,增大外循环次数‘*loop*1’的值,将会使 Myrinet 与 Ethernet 的 LU 模式运行总时间之差呈线性增长.如将第 3 行数据和第 1 行比较,尽管总的消息传递数目没变,但是通过了不同的内、外循环次数的重组,即 2000×60=4000×30,在消息长度相同的情况下,Myrinet 的运行总时间却明显不同.若将第 3 行和第 4 行进行比较,只有所传递的消息长度‘*msgSize*’不同,其余 3 个参数保持不变,更长的消息长度使得 Myrinet 比 Ethernet 慢得更多了.

Table 2 The impacts of ‘*msgSize*’, ‘*loop*1’, and ‘*loop*2’ parameters on the total running time of LU pattern

表 2 ‘*msgSize*’, ‘*loop*1’, ‘*loop*2’参数对 LU 模式运行总时间的影响

MsgSize (KB)	*Loop*1	*Loop*2	Time on Myrinet	Time on Ethernet
4	2000	60	698.61	494.12
4	4000	60	1396.19	985.61
4	4000	30	489.97	508.31
10	4000	30	694.93	509.16

4 新的评测模型

由前面的实验数据与分析可以看到,集群网络上点对点通信的性能受对应的发送/接收操作之间的计算延迟(即 LU 模式中的‘*compLoop*’参数)的影响确实很大.但如果一个实际的应用程序,如 NPB 中的 LU 存在这样的计算延迟,那么底层的通信介质必须有很好的能力来处理它.然而正如我们在文中所见,一个具有良好延迟、带宽、路由、拥塞处理、网络拓扑结构等的网络,如 Myrinet2000,并不意味着该网络也有良好的处理 LU 模式的能力.所以,这就提醒我们有必要将这个因素也加入到经典的、基于 Log*P* 模型的集群网络评测模型中,对其作出一个必要的完善与补充.

从图 5、表 2 以及其他的一些实验数据,我们可以简单地认为一个集群网络处理 LU 模式的能力可由如下公式来刻画:

$$SEMI(m)=\frac{(tc_i)/(tc_1)}{(tr_i(m)-tr_0(m))/(tr_1(m)-tr_0(m))}.$$

$$tc_1=f(tr_0(\mathrm{m}),ts_0(m)).$$

SEMI(SEnsitivity to MIsmatch timing)是我们定义的集群网络处理 LU 模式的能力因子.从直观上看,它就是图 5‘recv time’曲线每一点的斜率.

m 代表了图 4 中的‘*msgSize*’参数.

$tr_0(m)./ts_0(m)$代表了通信进程间对应的发送/接收操作匹配得很好,即图 4 中‘*compLoop*’参数为 0 的情况下,整个程序的接收时间与发送时间之比(‘recv time’/‘send time’).

$tc_1(m)$代表了当一个单位的‘tm_comp_us()’(在图 4 的基准程序中即定义为 10^8 次浮点运算时间)存在时,进程的 MPI 发送与接收操作所看到的计算延迟(表 1 中的‘comp time’).

$tc_i(m)$代表了 i 个单位的‘tm_comp_us()’存在时,进程的 MPI 发送与接收操作所看到的计算延迟(表 1 中的‘comp time’).

$tr_1(m)$代表了当‘*msgSize*’为 m,并存在一个单位的(tm_comp_us())计算延迟时,整个程序的接收时间(recv time).

$tr_i(m)$代表了当‘*msgSize*’为 m,并存在 i 个单位的(tm_comp_us())计算延迟时,整个程序的接收时间(recv time).

f 函数表明每一个单位的计算延迟(tm_comp_us()),即‘*compLoop*’,究竟该取多大,依赖于 tr_0 和 ts_0 参数,但其具体的对应关系(函数)还不是很清楚.

5 LU 模式在 Myrinet2000 环境下可能的解决方案及一些建议

正如我们前面所看到的,LU 模式对原本性能很好的 Myrinet2000 的通信性能有着极大的影响,也就是说,Myrinet2000 对 LU 模式的处理能力因子——SEMI 不够好.在得到其原因之后,它的解决方案也就很显然了:

1) 改变算法,避免 LU 模式的出现.这个解决方案又可以有两种具体的措施:

- 完全重新设计算法,重新改写程序源代码.毫无疑问,在大多数情况下,这样做工程浩大.
- 由于 Myrinet 的 SEMI 因子不够好,主要体现在当进程间对应的 MPI 发送与接收操作不匹配的情况下,接收操作的时间加长.那么就可以通过《MPI Specification》[11]中定义的同步模式——‘MPI_Ssend’/‘MPI_Recv’对,强制使其匹配,以减少不匹配情况下巨大的接收延迟.其具体的改进结果见表 3.

2) 对于一个大规模的集群系统来说,构建系统时首要考虑的因素应该是将要在该系统上计算什么样的题,然后根据所计算题目的性质,选择合适的网络,而不应该盲目地追求低延迟和高带宽.比方说,如果知道所主要运算题的性质类似于本文中的 LU 模式类型,那从表 3(第 2 列和第 6 列数据之对比)中我们可以看到,普通的百兆以太网无疑是一个性能价格比十分高的选择.

Table 3 Comparison of gtandard mode with gynchronous mode on Myrinet and Ethernet

表 3 标准通信模式和同步通信模式在 Myrinet 和 Ethernet 上的比较

NPB v2.4 LU Class=C					
Standard mode('MPI_Send'/'MPI_Recv')			Sync mode('MPI_Ssend'/'MPI_Recv')		
# of proc	Ethernet(s)	Myrinet	# of proc	Ethernet(s)	Myrinet
1	5200.88	5261.64	1	5152.28	5223.09
2	2730.95	4255.01	2	2724.50	2596.91
4	1451.96	2922.85	4	1581.26	1397.41
8	806.34	2194.38	8	968.90	721.64
16	497.63	1292.61	16	566.92	382.89

从表 3 中我们可以看到,若将 NPB 中的 LU 程序的通信模式从标准模式('MPI_Send'/'MPI_Recv')改为同步模式('MPI_Ssend'/'MPI_Recv'),其在 Myrinet 上的通信性能可以得到大幅度的提高,就 LU 程序本身几乎表现出了一个接近于线性的加速比.而原本性能就很好的 Ethernet,则由于额外的同步开销,性能稍有所下降,但仍保持了一个很好的加速比.表 3 也表明了集群网络通信性能之改善依赖于底层具体的通信介质.

6 相关工作

Log*P*[3]和 Log*GP*[4]模型是衡量集群网络性能最常用的两种通用模型.但文献[12,13]的工作都同时指出,针对现代集群网络的一些新的特性,若通信和通信之间可以有重叠,消息传递流水线机制等[13]都使得原来的Log*P*模型简单的消息传递串行的假设不再成立,因此必须针对这些新的特性使用不同的测试程序和测试机制.

在文献[13]中,针对 Cray T3E,IBM RS/6000 SP,Quadrics,Myrinet 2000,Gigabit Ethernet 等在不同的通信协议层采取了不同的测试机制;而文献[12]的工作更是针对 Quadrics,Infiniband,Myrinet 等现代集群网络的 OS Bypass,用户级通信等新特性设计了一些新的微型测试程序(Microbenchmark),如 1 个 Master、多个 Slave 的通信方式的测试,如单向、双向带宽的测试,发送、接收缓存可重用性测试,应用程序不平衡的通信方式测试(unbalanced communication patterns);文献[14,15]等工作则论述了在集群系统中系统进程的噪音(noise)对大规模科学计算的应用程序造成的影响、量化及消除等.

7 结论以及未来的工作

从以上的实验数据和分析可以看到,LU 模式确实可以极大地影响集群网络的通信性能.以前关于这方面的研究就我们所知,只局限于如何消除消息传递间的计算延迟/不匹配[10],而没有将其提高到集群网络通信性能的一个评测因子,成为评测模型的一个重要组成部分.但在本文中我们发现,一个网络即使具有低延迟、高带宽、很好的路由与网络拓扑结构等也不意味着它一定有很好的处理 LU 模式问题的能力,如我们所测试的 Myrinet2000 和普通百兆以太网的对比.而对一些实际存在的、具有 LU 模式的应用程序,如 NPB 中的 LU 程序,完全重新设计算法和重写代码显然工程浩大,这就需要底层通信平台对此类问题有一个很好的处理能力.

清楚了原因,解决方案也就可以得到了.当我们将 NPB 的 LU 程序中的标准通信模式改换为同步通信模式后,Myrinet 低延迟、高带宽的优势重又显现出来.表 3 的实验数据同时也表明了该解决方案是通信平台相关的.

尽管我们指出了将处理 LU 模式问题的能力引入集群网络的评测模型中的重要性,并且提出了我们的模型(SEMI 因子),但也应该看到图 4 中定义的所有 4 个参数都会对 LU 模式的解决产生或多或少的影响(表 2).我们的模型中所定义的 SEMI 因子仅仅是刻画出了其最主要的影响因素.因此,在以后的研究中,我们还应该对这个模型进一步地细化.

除了 LU 模式以外,是否还有其他一些特殊的通信模式会对集群网络的通信性能产生独立的影响?LU 模式对其他一些集群网络,如 Dolphin SCI,Gigabit Ethernet,InfiniBand 等是否也会产生如 Myrinet2000 般的重要影响?即在这些同样低延迟、高带宽的商业性集群网络上,处理 LU 模式的能力是否也与其低延迟、高带宽的外表不符合?这些方面都是需要我们继续深入研究的.

References:

[1] Boden NJ, Cohen D, Felderman RE, Kulawik AE, Seitz CL, Seizovic JN, Su WK. Myrinet: A gigabit-per-second local area network. IEEE Micro, 1995,15(1):29−36.

[2] Hsieh J, Leng T, Mashayekhi V, Rooholamini R. Architectural and performance evaluation of GigaNet and Myrinet interconnects on clusters of small-scale SMP servers. In: Donnelley J, ed. Proc. of the Super Computing 2000. Washington: IEEE Computer Society, 2000. 18−26.

[3] Culler D, Karp R, Patterson D. Log*P*: Towards a realistic model of parallel computation. In: Chen M, Halstead R, eds. Proc. of the 4th ACM SIGPLAN Symp. on Principles and Practice of Parallel Programming. New York: ACM Press. 1993. 1−12.

[4] Alexandrov A, Ionescu MF, Schauser KE, Scheiman C. LogGP: Incorporating long messages into the Log*P* model—one step closer towards a realistic model for parallel computation. Journal of Parallel and Distributed Computing, 1997,44(1):71−79.

[6] Moritz CA, Frank MI. LoGPC: Modeling network contention in message-passing programs. IEEE Trans. on Parallel and Distributed Systems, 2001,12(4):404−415.

[7] Ino F, Fujimoto N, Hagihara K. LogGPS: A parallel computational model for synchronization analysis. ACM SIGPLAN Notices, 2001,36(7):133−142.

[8] Saphir W, Woo A, Yarrow M. The NAS parallel benchmark 2.1 results. Report NAS-96-010, 1996.

[9] Van der Wijngaart RF. NAS parallel benchmarks version 2.4. NAS Technical Report NAS-02-007, 2002.

[10] Kielmann T, Bal HE, Verstoep K. Fast measurement of Log*P* parameters for message passing platforms. In: Rolim JDP, ed. Proc. of the IPDPS 2000 Workshops on Parallel and Distributed Processing. London: Springer-Verlag. 2000. 1176−1183.

[11] Hempel R. Basic message passing benchmarks, methodology and pitfalls. SPEC Workshop. 1999. http://www.ccrl-nece/technopark.gmd.de

[12] Snir M, Otto S, Huss-Lederman S, Walker D, Dongarra J. MPI: The Complete Reference. Cambridge: The MIT Press, 1996.

[13] Liu JX. Chandrasekaran B, Yu WK, Wu JS, Buntinas D, Kini S, Panda DK, Wyckoff P. Microbenchmark performance comparison of high-speed cluster interconnects. IEEE Micro, 2004. 42−51.

[14] Bell C, Bonachea D, Cote Y, Duell J, Hargrove P, Husbands P, Iancu C. Welcome M, Yelick K. An evaluation of current high-performance networks. In: Proc. of the Int'l Parallel and Distributed Processing Symp. (IPDPS 2003). IEEE CS Press, 2003.

[15] Kumar S, et al. Opportunities and challenges of modern Communication Architectures: Case study with QsNet. In: Panda DK, Duato J, Stunkel C, eds. Proc. of the 18th Int'l Parallel and Distributed Processing Symposium (IPDPS'04). Santa Fe: IEEE Computer Society. 2004. 182a.

[16] Petrini F, Kerbyson DJ, Pakin S. The case of the missing supercomputer performance: Achieving optimal performance on the 8192 processors of ASCI Q. In: McGraw JR, ed. Proc. of the ACM/IEEE SC2003 Conf. Phoenix: IEEE Computer Society, 2003. 55.