

系统噪音影响的量化分析

武林平 魏 勇 徐小文 刘 旭

(北京应用物理与计算数学研究所 北京 100094)

(wlp@iapcm.ac.cn)

Impact of System Noise by Quantitative Analysis

Wu Linping, Wei Yong, Xu Xiaowei, and Liu Xu

(Institute of Applied Physics and Computational Mathematics, Beijing 100094)

Abstract More attention should be paid on system noise for large-scale parallel application, although the system noise has little impact on one process. One quantitative analysis method for system noise's impact named FWQ-MPI is presented. Four quantitative indicators are given: the proportion of the amount of noise, the proportion of noise impact, the actual/ideal ratio of communication to calculation time. Three iterative methods are selected as the research objects and the micro benchmarks run on a MPP machine with 512 Double six-core nodes. The test results show the impact mechanism of system noise on parallel program performance, and also show several characterizations of system noise as follows. 1) The proportion of the amount of noise is relatively small, accounting for the entire computing time 2% to 6%; 2) But the system noise has much impact(when the parallel scale is 1024, 2048 and 4096, the proportion of noise impact is about 30% to 70%); 3) The impact of system noise will be increased when the parallel scale increases, will be decreased when the amount of calculation increases; 4) The impact of system noise is mainly reflected by the "actual ratio of communication to calculation time" is far less than the "ideal ratio of communication to calculation time".

Key words system noise; impact of system noise; quantitative analysis; load balance; numerical simulation; parallel program

摘 要 尽管系统噪音对单个进程的影响有限,但对于大规模并行程序性能的影响不容忽视.提出一种基于并行程序计算-通信特征的噪音影响定量评估方法 FWQ-MPI,并给出噪音影响的 4 个量化指标.选取求解稀疏线性代数方程组的 3 种迭代方法作为研究对象,抽取迭代方法的计算、同步通信特征形成微测试程序;在实际系统上的测试数据明确了系统噪音对并行程序性能的影响机理,并得到系统噪音对并行程序性能影响的若干规律:1)BSP 并行程序运行过程中,系统噪音量比例不大,约为 2%~6%;2)但系统噪音对 BSP 并行程序的性能有着较大的影响(当并行规模在 1024,2048,4096 时,噪音影响比例约为 30%~70%);3)其影响随着并行程序规模的扩大而增加,随着 2 次同步通信间隔内计算量的增加而降低;4)系统噪音的影响主要体现在 BSP 并行程序的“实际计算通信时间比”要远小于“理想计算通信时间比”.

关键词 系统噪音;系统噪音影响;量化分析;负载平衡;数值模拟;并行程序

中图法分类号 TP316.4

并行程序在运行过程中,位于计算结点上的进程经常由于 CPU 要处理来自硬件、固件、操作系统、系统管理软件等部件的干扰而被迫中断,这些由于系统部件所引起的计算进程中断被称之为系统噪音^[1].

相关研究表明,系统噪音对单个进程的影响较小,特别是以 Linux 作为计算结点操作系统的多核多处理器集群系统,每个处理器核的系统噪音比例仅约为 1%^[1-3].

而图 1 所示的“块同步并行(bulk synchronous parallel, BSP)”模式,为了满足负载平衡需求,并行程序设计人员需尽可能均匀地分割任务,从而保证并行程序的每个进程具有相同的计算性能、通信性能、访存性能,从而使得所有进程均在相同的时刻到达同步点、降低进程之间的相互等待(在科学计算中应用广泛、用于求解大规模稀疏线性代数方程组的迭代方法是这类应用的典型例子,以开源软件 HPCCG^[4]为例,在每个迭代步内的 2 次全局归约 Allreduce 通信之间,所有进程具有相同的负载,包括一次向量校正计算 axpy、一次开平方 sqrt、与相邻进程点对点通信、一次稀疏矩阵向量乘 SpMV、一次向量内积 dot). 但系统噪音的扰动将破坏这种负载平衡,使得各计算进程不能在同一时刻到达同步点,造成并行程序的实际运行过程如图 2 所示,从而引起进程间的相互等待现象,且随着并行规模的扩大这种相互等待将加剧.

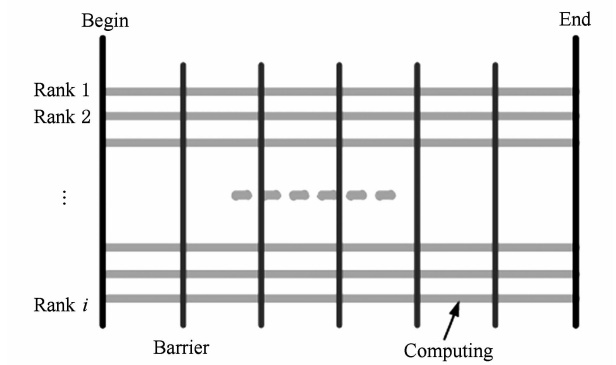


Fig. 1 BSP parallel model.

图 1 BSP 并行模型

针对系统噪音的量化研究主要包括如下 2 方面:

1) 噪音量的量化分析. 通常采用标准负载来刻画系统噪音量,逐步形成 3 种系统噪音量的分析方法,分别为:固定工作量(fixed work quantum, FWQ)、固定时间量(fixed time quantum, FTQ)、噪音序列(detour series, DTS)^[1].

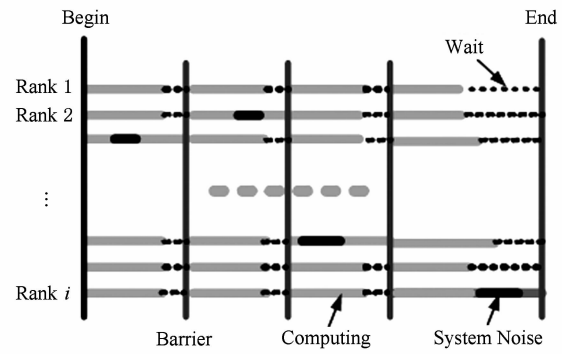


Fig. 2 The real state of BSP parallel program.

图 2 BSP 并行程序的实际运行过程

2) 噪音影响的量化分析. 噪音对并行程序性能的实际影响与噪音特征、噪音量、并行程序计算特征及通信模式等因素有关. 在文献[5]中,通过仿真的方法量化系统噪音的影响:根据指定系统噪音特征(间隔、噪音持续时间、周期性等特征)生成噪音事件序列,基于并行程序通信性能模型 LogGOP 建立并行程序执行的离散事件仿真过程. 该仿真方法能够评估任意并行规模下的噪音影响而不受实际系统规模的限制,但需事先建立精确的系统噪音特征参数、高性能计算机通信系统性能参数、应用程序计算及通信特征参数,否则将严重影响其仿真精度. 本文设计一种基于实测的系统噪音影响量化评估方法 FWQ-MPI:根据图 1 的 BSP 并行模型,提取并行程序的计算、MPI 同步通信特征形成微测试程序,通过实际运行微测试程序直观地获得噪音影响量化结果. 本文主要贡献包括:1) 提出一种基于实际测量的噪音影响量化评估方法 FWQ-MPI;2) 给出噪音影响的 4 个量化指标:噪音量比例、噪音影响比例、实际计算通信时间比、理想计算通信时间比;3) 明确了系统噪音对并行程序性能的影响机理及若干影响规律.

1 量化分析方法

若干变量说明:

n 为并行规模; m 为循环次数; P_{FWQ} 为“固定工作量”,一组固定计算量的负载.

1.1 FWQ-MPI 原理

根据并行程序的 BSP 模型,并行程序的执行模型如下:

```
for(){
    Do_work();
    MPI_SYNC();
}
```

其中, Do_work 代表每个进程需完成的计算任务, MPI_SYNC 表示每一步循环结束前的同步操作。

为了描述噪音影响, 需要对上述 BSP 并行模型添加高精度计时器, 采用硬件时钟来记时, 如 X86 的 $rdtsc$ 指令^[6] (如果 CPU 主频为 3 GHz, 则 $rdtsc$ 计时器的分辨率为 0.3 ns, 从而保证记时精度)。

FWQ-MPI 的测量流程如下:

```
for() {
    T1 = rdtsc(); /* 获取当前时刻 */
    Do_work(P_FWQ);
    T2 = rdtsc();
    MPI_SYNC();
    T3 = rdtsc();
}
```

其中, 根据实际应用计算特征构造 Do_work 函数的输入参数 P_{FWQ} . 针对第 i 进程的第 j 个循环步计算得到:

1) P_{FWQ} 计算时间

$$W_{i,j} = T2_{i,j} - T1_{i,j}. \quad (1)$$

2) MPI 同步时间

$$R_{i,j} = T3_{i,j} - T2_{i,j}. \quad (2)$$

当不存在系统噪音扰动的理想情况下, 同一循环步中所有进程的计算时间 $W_{i,j}$ 及同步时间 $R_{i,j}$ 应该相等, 并行程序将按照图 1 的执行轨迹运行; 而实际上由于存在系统噪音的扰动, $W_{i,j}$ 呈现一定波动而非恒定值, 这种波动使得所有进程不在同一时刻到达同步点, 继而影响到 $R_{i,j}$ 的波动. 为了量化系统噪音的影响, 给出若干量化指标。

1.2 量化指标

根据 1.1 节所述 FWQ-MPI 原理, 对于第 j 次循环步, 计算下列参数:

1) 该迭代步中所有进程计算时间的最小值

$$W_{\min,j} = \min(W_{0,j}, W_{1,j}, W_{2,j}, \dots, W_{n-1,j}). \quad (3)$$

2) 该循环步中所有进程计算时间的最大值

$$W_{\max,j} = \max(W_{0,j}, W_{1,j}, W_{2,j}, \dots, W_{n-1,j}). \quad (4)$$

下面给出若干描述系统噪音影响的量化指标:

1) 噪音量比例. 把 $W_{\min,j}$ 近似为第 j 次循环步中没有噪音干扰时负载 P_{FWQ} 的计算时间, 则第 i 个进程的噪音量可计算为 $(W_{i,j} - W_{\min,j})$. 并行程序在整个运行过程中的噪音量比例可计算为

$$\sum_{j=1}^m \left(\sum_{i=0}^{n-1} (W_{i,j} - W_{\min,j}) \right) / \sum_{j=1}^m \left(\sum_{i=0}^{n-1} W_{i,j} \right). \quad (5)$$

2) 噪音影响比例. 针对第 j 次循环步, MPI 同步操作的开始时刻取决于最慢进程的计算时间

$W_{\max,j}$, 也即针对每个进程 i , 在完成自身 P_{FWQ} 计算负载后需要等待的时间可计算为 $(W_{\max,j} - W_{i,j})$, 这样的等待时间就是系统噪音所产生的影响. 并行程序在整个运行过程中的噪音影响比例可计算为

$$\sum_{j=1}^m \left(\sum_{i=0}^{n-1} (W_{\max,j} - W_{i,j}) \right) / \sum_{j=1}^m \left(\sum_{i=0}^{n-1} W_{i,j} \right). \quad (6)$$

3) 实际计算通信时间比. 计算通信时间比对于并行程序的可扩展性有很大影响, 并行程序性能分析中常用的“计算通信比”一般使用 0 号进程的计算

时间与通信时间比来表示: $\sum_{j=1}^m W_{0,j} / \sum_{j=1}^m R_{0,j}$. 而本

定义中的计算时间、通信时间分别为所有进程的计算时间之和和通信时间之和, 实际计算通信时间比为

$$\sum_{j=1}^m \left(\sum_{i=0}^{n-1} W_{i,j} \right) / \sum_{j=1}^m \left(\sum_{i=0}^{n-1} R_{i,j} \right). \quad (7)$$

4) 理想计算通信时间比. 所谓理想情况是指并行程序在消除噪音干扰后的计算时间与通信时间之比. 事实上, 系统噪音增加了并行程序的 MPI 同步时间, 而实际无噪音干扰的 MPI 同步开销很小, 因此, “理想计算通信时间比”刻画了消除噪音干扰后的计算通信比, 是系统噪音分析的重要参考. 具体计算过程包括 2 步: 1) 使用 OSU Micro Benchmarks^[7] 测量无计算负载情况下的 MPI 同步开销 $R_{i,j}$; 2) 计算“理想计算通信时间比”:

$$\sum_{j=1}^m \left(\sum_{i=0}^{n-1} W_{i,j} \right) / \sum_{j=1}^m \left(\sum_{i=0}^{n-1} (R_{i,j} \mid W_{i,j} = 0) \right). \quad (8)$$

2 微测试程序的设计

根据 FWQ-MPI 测量原理及相关噪音影响的量化指标, 需要结合并行程序的实际计算、同步特征设计相关微测试程序, 包括计算负载 P_{FWQ} , MPI 同步 2 方面。

2.1 负载特征

在很多数值模拟应用中, 求解稀疏线性代数方程组是核心计算模块, 在很多实际应用中占据很大的时间比例^[8]. 本文选取求解稀疏线性代数方程组的迭代方法作为研究对象, 抽取迭代方法的计算特征、同步特征形成微测试程序。

文献[8]对迭代方法有详细分析: 迭代方法包含稀疏矩阵-向量乘积 (SpMV)、向量内积 (dot)、向量校正 (axpy) 3 类计算及 MPI 全局规约同步负载, 具体如下:

求解稀疏线性代数方程组 $Ax = b$, 其中稀疏矩

阵 $\mathbf{A}=(a_{i,j})_{l\times l}$, 定义向量 $\mathbf{x}=(x_1,x_2,\cdots,x_l)^\top$, $\mathbf{y}=(y_1,y_2,\cdots,y_l)^\top$, $\mathbf{z}=(z_1,z_2,\cdots,z_l)^\top$ (l 为矩阵和向量的规模) 和常数 a , 则 3 种计算、MPI 全局规约同步负载为:

1) 稀疏矩阵-向量乘积.

$$\mathbf{y}=\mathbf{Ax}, y_i=\sum_{j=1}^l a_{i,j}x_j, i=1,2,\cdots,l. \quad (9)$$

2) 向量内积.

$$d=(\mathbf{x},\mathbf{y}), d=\sum_{i=1}^l x_iy_i. \quad (10)$$

3) 向量校正.

$$\mathbf{z}=\mathbf{ax}+\mathbf{y}, z_i=ax_i+y_i, i=1,2,\cdots,l. \quad (11)$$

4) 每次向量内积中包含一次 MPI 全局规约同步操作.

实际应用中的迭代方法是上述 4 种负载的不同组合, 本文的微测试程序包含了表 1^[8] 中的 3 种典型迭代方法: Jacobi 方法、CG (共轭梯度方法)、BiCGSTAB (双正交共轭梯度稳定化方法).

Table 1 The Workload of Three Iterative Methods in Microbenchmark

表 1 微测试程序中 3 种方法单次迭代的基本负载次数

Iterative Method	SpMV	dot	MPI_Allreduce	axpy
Jacobi	1	1	1	1
CG	1	3	3	3
BiCGSTAB	2	6	6	7

2.2 微测试程序

根据 2.1 节所述的负载特征, 采用 FWQ-MPI 测量原理及量化指标的计算方法, 实现微测试程序.

1) 输入参数

并行规模 n 、迭代次数 m 、向量长度与矩阵规模 l 、迭代方法 d (Jacobi, CG, BiCGSTAB).

2) 输出结果

噪音量比例 R_N 、噪音影响比例 R_I 、实际计算通信比 R_CT_A 、理想计算通信比 R_CT_I .

3) 测量代码

① 针对 Jacobi 迭代方法 ($d==\text{Jacobi}$).

$count=0$;

$\text{double } T1[m], T2[m], T3[m];$

$T0=\text{rdtsc}()$;

$\text{While}(count<m)\{$

$T1[count]=\text{rdtsc}()$;

$\text{Do_SpMV}()$;

$\text{Do_dot}()$;

$\text{Do_axpy}()$;

$T2[count]=\text{rdtsc}()$;

$\text{MPI_Allreduce}()$;

$T3[count]=\text{rdtsc}()$;

$\}$

参考式(1)~(4), 针对第 j 次迭代步的第 i 个进程, 计算 $W_{i,j}=T2[j]-T1[j]$, $R_{i,j}=T3[j]-T2[j]$; 针对第 j 次迭代步的所有进程, 计算 $W_{\min,j}$ 和 $W_{\max,j}$. 再据式(5)~(8)分别计算得到噪音量比例、噪音影响比例、实际计算通信比、理想计算通信比.

② 针对 CG 迭代方法 ($d==\text{CG}$).

$count=0$;

$\text{double } T1[m], T2[m], T3[m], T4[m],$

$T5[m], T6[m], T7[m];$

$T0=\text{rdtsc}()$;

$\text{While}(count<m)\{$

$T1[count]=\text{rdtsc}()$;

$\text{Do_SpMV}()$;

$\text{Do_dot}()$;

$\text{Do_axpy}()$;

$T2[count]=\text{rdtsc}()$;

$\text{MPI_Allreduce}()$;

$T3[count]=\text{rdtsc}()$;

$\text{Do_axpy}()$;

$\text{Do_dot}()$;

$T4[count]=\text{rdtsc}()$;

$\text{MPI_Allreduce}()$;

$T5[count]=\text{rdtsc}()$;

$\text{Do_axpy}()$;

$\text{Do_dot}()$;

$T6[count]=\text{rdtsc}()$;

$\text{MPI_Allreduce}()$;

$T7[count]=\text{rdtsc}()$;

$\}$

针对第 j 次迭代步的第 i 个进程, 计算:

$W1_{i,j}=T2[j]-T1[j]$,

$R1_{i,j}=T3[j]-T2[j]$,

$W2_{i,j}=T4[j]-T3[j]$,

$R2_{i,j}=T5[j]-T4[j]$,

$W3_{i,j}=T6[j]-T5[j]$,

$R3_{i,j}=T7[j]-T6[j]$.

针对第 j 次迭代步 (3 次计算负载) 的所有进程, 计算 $W1_{\min,j}$ 及 $W1_{\max,j}$, $W2_{\min,j}$ 及 $W2_{\max,j}$, $W3_{\min,j}$

及 $W3_{\max,j}$;再根据式(5)~(8)分别计算得到噪音量比例、噪音影响比例、实际计算通信比、理想计算通信比.

③ 针对 BiCGSTAB 迭代方法($d==\text{BiCGSTAB}$). BiCGSTAB 与 CG 的测量代码类似,区别在于:每次迭代步中 BiCGSTAB 的 $Do_SpMV()$ 计算负载为 2 次, $Do_dot()$ 及 $MPI_Allreduce()$ 为 6 次, $Do_axpy()$ 为 7 次,限于篇幅这里不做详细介绍.

3 测量结果

3.1 输入参数及测量平台

根据实际应用需求,微测试程序的输入参数:

- 1) 并行规模为 1024,2048,4096 共 3 种;
- 2) 向量长度(矩阵规模)为 8192,16384,32768,65536 共 4 种,向量长度代表计算粒度,向量越长则表明 2 次 MPI 同步间隔内的计算量越大(后文所使用的“计算粒度”均表示该含义);
- 3) 迭代次数为 30 000 次;
- 4) $MPI_Allreduce()$ 数据为一个双精度浮点数(double,8 B).

所有测量在一台包含 512 个双路 6 核计算结点的并行机上进行(共 6 144 个处理器核);计算结点操作系统使用 Linux;处理器为 Intel xeon 系列;内存控制器集成在处理器内部,配置为 NUMA 结构,单结点共享 48 GB 内存;微测试程序在每个处理器上分派 4 个进程(每计算结点分派 8 个进程).

3.2 噪音影响的机理

为了显示系统噪音对 BSP 并行程序性能影响的机理,使用微测试程序的 Jacobi 代码测量,分别得到迭代过程中每个进程的计算时间和 MPI 同步时间.图 3 给出了其中某次迭代过程中每个计算进程的计算时间与 $MPI_Allreduce$ 时间分布:采用 6 个计算结点,每结点 8 个进程;横轴为进程编号,纵轴为时间.从图 3 可以看出,尽管在该迭代步中每个进程具有相同的负载,但进程 6,12,30,36 比其他进程的计算时间增加约 20%.上述分析可见,在系统噪音的影响下某些进程的计算时间过长,从而增加了 $MPI_Allreduce$ 过程的时间开销.

而随着系统规模、数值模拟应用并行度的攀升,系统噪音的发生概率升高,由原来的小概率事件逐渐演化为确定性事件,成为超大规模高性能计算机系统的常态,这些常态化的干扰因素对数值模拟应用程序的计算、通信均带来干扰.例如,在点对点通

信情况下,局部进程的计算、数据通信性能下降,必然影响到与之有通信关系的进程;特别是对含有聚合通信的数值模拟程序,由于聚合通信的全局同步特征,局部进程的计算、通信性能的降低必然波及到全局,从而导致数值模拟应用程序性能的下降,并且这种影响随着并行规模的增加而增大.例如文献[9]中通过降低系统噪音 ASCII Q 中的干扰,8192 并行规模的数序 SAGE 性能提升 2 倍;文献[10]中,基于噪音干扰注入的方法评估系统噪音对 CTH, SAGE,POP 这 3 个数值模拟应用的影响,万核规模时其性能下降达 20%~60%.

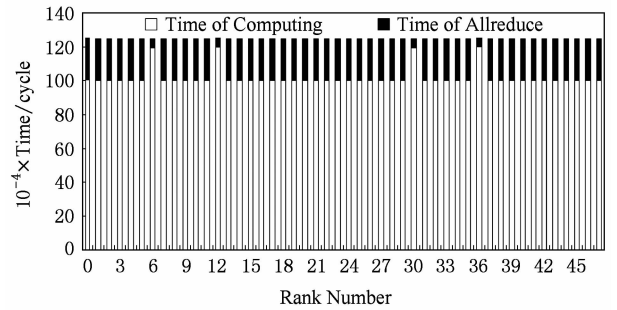


Fig. 3 Time of computing and allreduce in one iterative step.

图 3 单次迭代中的计算与同步时间分布图

3.3 数据分析

使用 3.1 节的输入参数进行测量,通过分析测量数据,获得系统噪音对并程序性能影响的若干规律.具体情况如下:

1) BSP 并行程序运行过程中,系统噪音量比例不大,约为 2%~6%.表 2,3 分别给出了 Jacobi,

Table 2 The Proportion of the Amount of Noise for Jacobi
表 2 Jacobi 迭代方法的噪音量比例

Parallel Scale	Length of Vector			
	8 192	16 384	32 768	65 536
1 024	3.97	3.76	3.42	4.49
2 048	3.71	3.95	3.37	4.54
4 096	3.13	3.85	3.53	5.44

Table 3 The Proportion of the Amount of Noise for BiCGSTAB
表 3 BiCGSTAB 迭代方法的噪音量比例

Parallel Scale	Length of Vector			
	8 192	16 384	32 768	65 536
1 024	3.54	3.9	2.99	5.13
2 048	3.61	3.99	3.12	5.35
4 096	2.97	3.86	3.2	5.91

BiCGSTAB 两种迭代方法运行过程中的噪音量比例,系统噪音量比例并没随着并行规模的增加而上升,随着计算粒度的增加系统噪音量比例有所上升但上升幅度不大,整体噪音量保持平稳.

2) 尽管噪音量比例较小,但对 BSP 并行程序有较大的性能影响(噪音影响比例较大),当并行规模在 1024,2048,4096 时,噪音影响比例约为 30%~70%. 表 4,5 分别给出了 Jacobi,BiCGSTAB 两种迭代方法运行过程中的噪音影响比例,表 4,5 中数据显示,随着并行规模的扩大,噪音影响增加;而随着计算粒度的增加,噪音影响降低. 因此,系统噪音对于大规模、细计算粒度 BSP 并行程序的性能影响较大,值得关注.

Table 4 The Proportion of Noise Impact for Jacobi

表 4 Jacobi 迭代方法的噪音影响比例

Parallel Scale	Length of Vector			
	8 192	16 384	32 768	65 536
1 024	37.99	34.71	34.13	25.66
2 048	54.59	33.88	38.44	25.6
4 096	65.31	58.56	53.93	43.83

Table 5 The Proportion of Noise Impact for BiCGSTAB

表 5 BiCGSTAB 迭代方法的噪音影响比例

Parallel Scale	Length of Vector			
	8 192	16 384	32 768	65 536
1 024	32.44	33.93	32.16	24.48
2 048	54.3	33.64	34.69	28.4
4 096	87.39	65.75	58.8	49.28

3) 在 3.2 节给出了系统噪音对并行程序性能的影响机理,即系统噪音的干扰增加了每次 MPI 同步的时间开销. 因此,系统噪音的影响主要体现在 BSP 并行程序的“实际计算通信时间比”要远小于

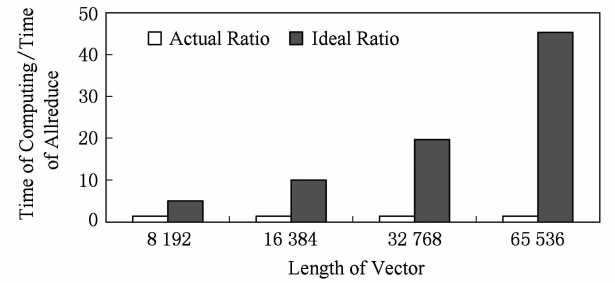


Fig. 4 Time of computing/time of allreduce for 1024 parallel scale CG iterative method.

图 4 CG 迭代方法在 1024 并行规模时的计算通信比

“理想计算通信时间比”. 图 4,5 的测量数据显示: “理想计算通信时间比”是“实际计算通信时间比”的 5~30 倍.

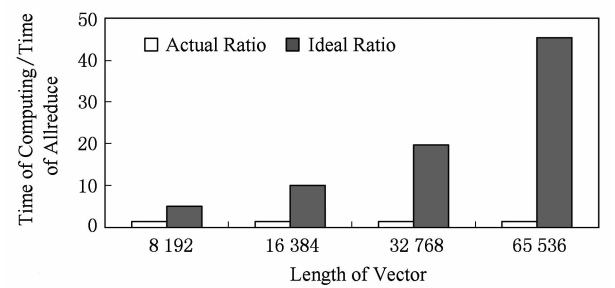


Fig. 5 Time of computing/time of allreduce for 4096 parallel scale CG iterative method.

图 5 CG 迭代方法在 4096 并行规模时的计算通信比

4 结论及将来的工作

随着并行规模的扩大,MPI 同步通信开销已成为制约 BSP 并行程序可扩展性的重要瓶颈. 本文探讨了系统噪音扰动对 BSP 并行程序性能的影响机制,给出一种基于实测的系统噪音影响的量化方法 FWQ-MPI. 根据该方法,针对在数值模拟程序中有广泛应用、用于求解大规模稀疏线性方程组的 3 种常用迭代方法(Jacobi,CG,BiCGSTAB),通过提取它们的计算、同步通信特征形成噪音影响的微测量程序. 在实际系统上的测量数据表明:

- 1) 对于 BSP 并行程序,虽然系统噪音对单个进程的影响很小^[5](一般约为 0.1%~1%),但系统噪音增加了每次 MPI 同步通信的时间开销.
- 2) BSP 并行程序运行过程中,系统噪音量比例不大,约占整个计算时间的 2%~6%.
- 3) 尽管噪音量比例较小,但对 BSP 并行程序的性能有较大影响(噪音影响比例较大),当并行规模在 1024,2048,4096 时,噪音影响比例约为 30%~70%.
- 4) 系统噪音的影响随着并行程序规模的扩大而增加,随着 MPI 同步间隔内计算量的增加而降低. 因此,系统噪音对于大规模、细计算粒度 BSP 并行程序的性能影响较大.
- 5) 系统噪音的影响主要体现在 BSP 并行程序的“实际计算通信时间比”要远小于“理想计算通信时间比”. 实测数据显示,“理想计算通信时间比”是“实际计算通信时间比”的 5~30 倍.

本文的 FWQ-MPI 分析方法可用于高性能计

算机的系统评测过程,也可用于 BSP 并程序的性能问题定位. 另外,系统噪音不仅仅影响大规模数值模拟并行应用程序,对于大规模数据处理、网络搜索、电子商务等领域的应用也有很大影响,造成服务系统的延迟波动. 文献[11]中,Google 的 Jeff Dean, Luiz Barroso 系统地阐述了大型服务的容延技术.

针对系统噪音的研究可分为 2 个范畴:1)研究噪音的量化分析方法,包括噪音量的测量及噪音影响的评估;2)根据量化分析结果,研究系统噪音影响的抑制策略. 因此,未来的研究将立足于下列 2 个问题:

1) 寻找噪音源. 根据噪音出现的位置、时间特征,把噪音分成 2 大类:①固定噪音. 该类噪音具有较固定的时间分布特征或位置分布特征,例如某处理器核上具有较大噪音,可通过计算该处理器核上的实测计算通信比来定位,通过开发专用的测试程序来探寻原因. ②随机噪音. 该类噪音的出现时间、位置呈现随机或者细微的周期性,需要通过数据分析掌握该类噪音的分布特征,通过系统采样工具掌握其源头.

2) 噪音影响的抑制策略. 为了降低噪音对并行程序的性能影响,通过噪音源的定位,分别探求针对固定类噪音、随机类噪音的应对手段.

参 考 文 献

[1] Wu Linping, Wei Yong, Liu Xu. The quantitative measurement of system noise in multiprocessors multicore clustered systems [C] //Proc of the 2012 Naitonal Annual Conf on High Performance Computing. Beijing: China Computer Federation, 2012: 1-5 [2013-12-01]. <http://www.sciinfo.cn/ShowDetail.aspx?d=1019&id=HYW201311260000009109&m> (in Chinese)
(武林平, 魏勇, 刘旭. 多核集群中系统噪音的测量[C] // 2012 全国高性能计算学术年会. 北京: 中国计算机学会, 2012: 1-5)

[2] Gioiosa R, Petrini F, Davis K, et al. Analysis of system overhead on parallel computers [C] //Proc of the 4th IEEE Int Symp on Signal Processing and Information Technology. Piscataway, NJ: IEEE, 2004: 387-390

[3] Beckman P, Iskra K, Yoshii K, et al. Benchmarking the effects of operating system interference on extreme-scale parallel machines [J]. Cluster Computing, 2008, 11(1): 3-16

[4] Herowx M A. HPCCG: A simple conjugate gradient benchmark code for a 3D chimney domain on an arbitrary number of processors [CP/OL]. [2014-03-13]. <http://www.mantevo.org/downloads/HPCCG-1.0.html>

[5] Hoefler T, Schneider T, Lumsdaine A. Characterizing the influence of system noise on large-scale applications by simulation [C] //Proc of the 2010 ACM/IEEE Int Conf for High Performance Computing, Networking, Storage and Analysis. Piscataway, NJ: IEEE, 2010: 1-11

[6] Intel. Intel 64 and IA-32 Architectures Software Developer's Manual, B: Instruction Set Reference, N-Z [M]. Santa Clara, California: Intel Corporation, 2010: 251-252

[7] Dhabaleswar K. Osu micro-benchmarks [CP/OL]. [2014-03-13]. <http://mvapich.cse.ohio-state.edu/benchmarks/>

[8] Xu Xiaowen, Mo Zeyao, Wu Linping. Analysis of communication-to-computation based on asymptotic size for iterative methods [J]. Chinese Journal of Computers, 2013, 36(4): 782-789 (in Chinese)
(徐小文, 莫则尧, 武林平. 迭代方法中基于渐近规模的通信与计算比分析[J]. 计算机学报, 2013, 36(4): 782-789)

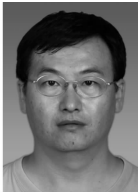
[9] Petrini F, Kerbyson D K, Pakin S. The case of the missing supercomputer performance: Achieving optimal performance on the 8192 processors of ASCI Q [C] //Proc of the 2003 ACM/IEEE Conf on Supercomputing. Piscataway, NJ: IEEE, 2003: 55-55

[10] Ferreira K B, Bridges P, Brightwell R. Characterizing application sensitivity to OS interference using kernel level noise injection [C] //Proc of the 2008 ACM/IEEE Conf on Supercomputing. Piscataway, NJ: IEEE, 2008: 19-29

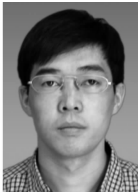
[11] Dean J, Barroso L A. The tail at scale [J]. Communications of the ACM, 2013, 56(2): 74-80



Wu Linping, born in 1977. PhD and associate professor. His current research interests include high efficiency system management and fault tolerance for high performance computing systems.



Wei Yong, born in 1965. Technician. His current research interests include system management for high performance computing systems.



Xu Xiaowen, born in 1978. PhD and associate professor. His current research interests include parallel computing.



Liu Xu, born in 1981. PhD and associate professor. His current research interests include parallel computing.