

SAR 成像处理在高性能计算平台上的性能分析和优化¹

雷 斌 丁赤飏 吴一戎

(中国科学院电子学研究所 北京 100080)

摘 要: 高性能计算机系统是现代 SAR 地面处理系统中的重要组成部分。与其它高性能计算问题相比, SAR 成像处理有其特殊性。随着高性能计算服务器架构的不同, SAR 成像处理的性能存在很大的差别。该文分析了高性能计算服务器架构中的各个部分对 SAR 成像处理性能的影响, 并在此基础上指出可以通过一系列典型测试, 较为准确地预测在多节点环境下的 SAR 成像处理性能。通过对测试和计算结果的分析, 进一步指出对 SAR 成像处理的优化方向和策略。

关键词: SAR 成像处理, 高性能计算

中图分类号: TN929.5 **文献标识码:** A **文章编号:** 1009-5896(2004)08-1245-05

Performance Analysis and Optimization of SAR Imaging on HPC Platforms

Lei Bin Ding Chi-biao Wu Yi-rong

(Institute of Electronics, Chinese Academy of Science, Beijing 100080, China)

Abstract High Performance Computing (HPC) system is an important part of a modern SAR processing system. SAR imaging has its specific characteristics compared with other HPC problems. It performs with great difference on different HPC architectures. The effects on SAR imaging performance by every part of the HPC architecture are analyzed on this paper. Based on this analysis, some benchmarking could be made to more accurately predict the SAR imaging performance on a multi-nodes environment. From these benchmarks, the effective way to optimize the SAR imaging could be found.

Key words SAR imaging, High performance computing (HPC)

1 概述

近几年来, SAR 图像的应用得到了很大的发展。随着成像分辨率的提高, 所需处理的数据量和运算量大大增加, 使得 SAR 地面处理系统的规模不断扩大。在现有技术条件下, 提高系统性能的基本方法是增加更多的节点用于处理。当系统的节点数增加时, 由于在节点之间交换数据所带来的开销, 并行效率将下降, 这一特点在 SAR 成像处理中尤其严重。

从程序实现的角度来看, 目前应用于 SAR 成像处理的高性能计算机系统通常包括两类: 一类是采用 NUMA 或类似架构的全局共享内存系统, 本文中以 SGI Origin 2000 服务器为代表; 另一类是采用多个独立节点构成的集群系统, 本文中以联想深腾 1800 服务器为代表。

科学计算领域目前已经广泛采用 SPEC 等性能测试指标评估高性能计算平台的处理能力。由于针对性不同, 这些评测数据并不能反映相同的平台在完成 SAR 成像处理时的真实性能。

本文首先根据 SAR 成像处理的特点, 分析高性能计算服务器架构中影响 SAR 成像处理性能的环节, 其次, 从这些环节入手, 通过一些典型测试对高性能计算平台的处理能力进行预测, 最后, 根据测试的结果, 指出了对 SAR 成像处理进行进一步优化的方向和策略。

本文的一部分测试是基于 Intel MKL, SGI SCSL 和来自 AMD 的部分源代码的基础上完成的。

¹ 2003-04-07 收到, 2003-07-03 改回
863 计划信息获取与处理主题资助项目

2 SAR 成像处理特点分析

在 SAR 成像处理中常见的 Chirp scaling 算法的处理流程如图 1 所示^[1,2]，这一算法具有如下特点：

2.1 浮点运算集中

SAR 成像算法使用了大量的 FFT 运算和相位补偿运算，在相位补偿运算中包含了大量的三角函数运算。这些浮点运算占据了 SAR 成像处理中的很大一部分处理时间。

2.2 大数据量

SAR 成像一次处理的数据量非常庞大，并且与一般的音频和视频处理不同，SAR 成像处理不能采用数据流处理方式，只能采用数据块处理方式。由于现有的 32 位操作系统对应用程序可使用的地址空间存在 2 Gbyte 的限制，有时需要人为地将数据分配到多个节点进行处理。

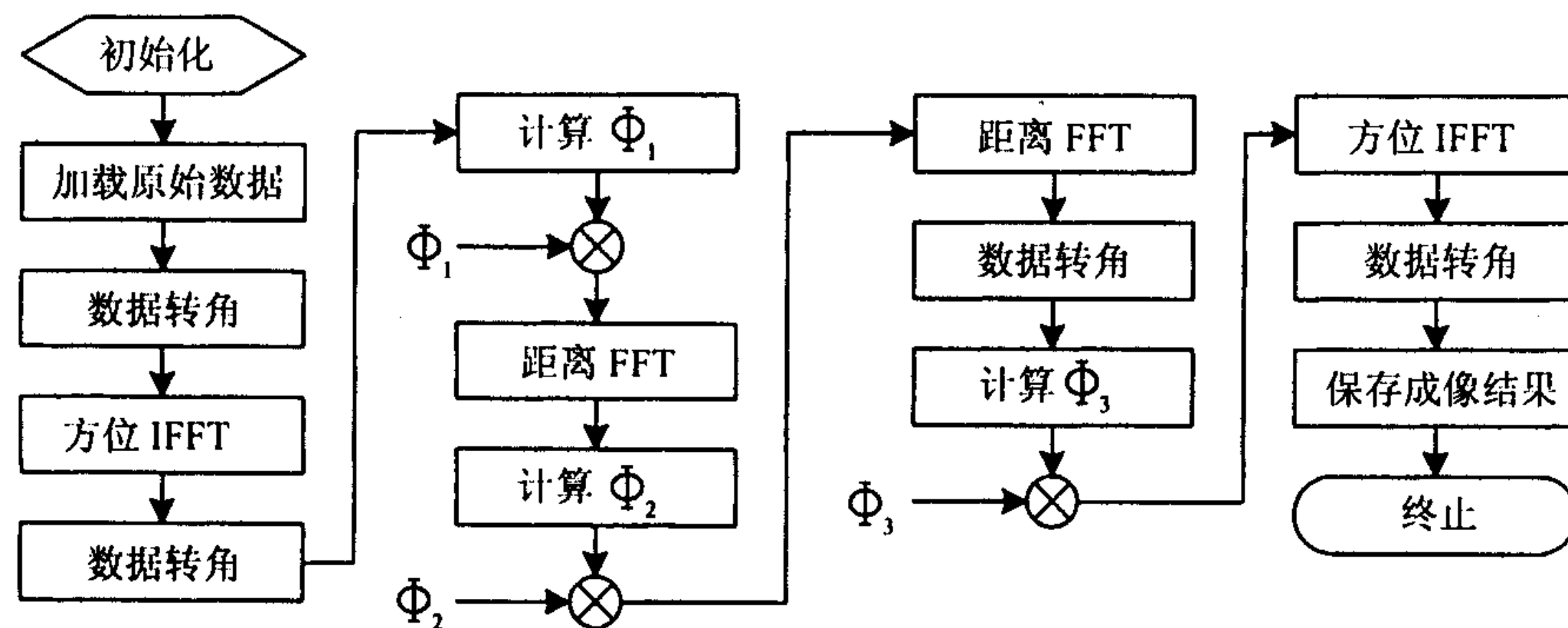


图 1 Chirp scaling 算法流程

2.3 在并行时需要在节点间交换大量的数据

SAR 成像处理在方位处理和距离处理之间需要对数据进行转角运算 (transpose)，在多个节点并行执行时，需要在节点间交换大量的数据。

3 高性能计算服务器架构的影响

只有结合 SAR 成像处理的特性，分析现代高性能计算机系统的各个部分对 SAR 成像处理的影响，才能最大限度地挖掘现有的高性能计算系统的处理资源。

高性能计算服务器架构中的处理器子系统、内存子系统和节点连接方式对 SAR 成像处理有较大的影响。

充分利用现代处理器提供的 SIMD 扩展指令，可以明显地提升 SAR 成像处理程序的浮点运算性能。对于 Intel 公司的处理器，这些扩展指令集包括 MMX, SSE, SSE2 等^[3,4]。MIPS 处理器采用的软件流水线技术对于充分利用处理器中的多个执行单元也有重要的作用^[5]。此外，现代处理器的高速缓存 (cache) 特性可以大大提高数据的存取速度，从而提升处理性能。

在不同的高性能计算机系统上，节点之间的数据交换速度存在着很大的差异，这种差异直接影响了 SAR 成像处理的并行效率。

在采用 NUMA 架构的高性能计算机系统上开发并行程序非常简单，但这些计算机系统所使用的分级内存体系^[5]使得在编写 SAR 成像处理程序时必须仔细地设计数据的存取方式。不恰当的数据存取方式会降低程序的并行效率。

4 典型性能测试

SAR 成像处理中包含以下几类典型运算：

4.1 FFT

对分裂基 FFT 算法的分析表明，对于 $N = 2^M$ 点的 FFT，平均每点所需的浮点运算次数为 $4M - 6$ ^[6]。实际测试表明，当 N 增大时，处理器的高速缓存命中率降低，单位时间内平均能执行的浮点运算次数也随之降低，FFT 运算的效率进一步地下降。图 2 给出的测试结果说明

了这一现象. 表 1 对比了几种常见处理器的 FFT 性能, 测试结果为对一幅 16384×16384 的雷达数据进行一维 FFT 所需的时间.

4.2 相位补偿运算

在 SAR 成像处理中经常需要执行如下的相位补偿运算: $B(k, l) = A(k, l) \exp\{j\varphi(k, l)\}^{[2]}$. 这一运算中最主要的运算量集中在 $\sin\varphi$ 和 $\cos\varphi$ 的计算上. 因此, 测试某一特定的计算平台的 SIN 和 COS 运算性能可以评估 SAR 成像处理过程中相位补偿运算的性能.

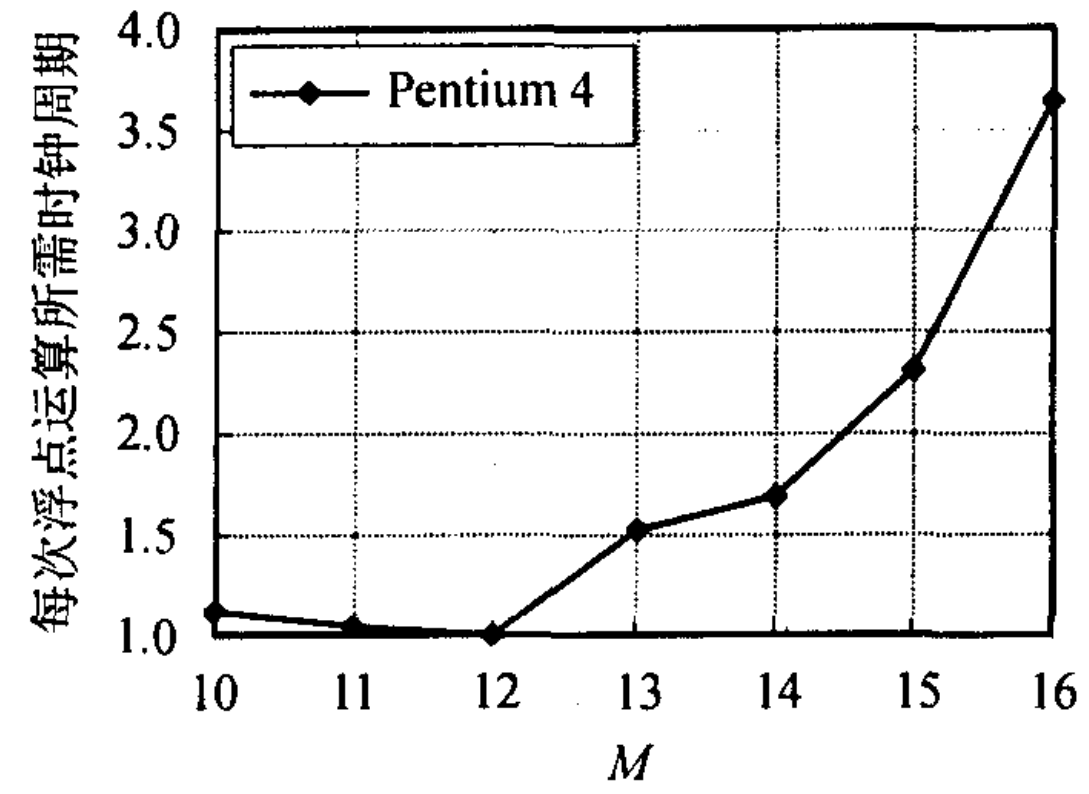


图 2 不同处理器的 FFT 性能

表 1 不同处理器的 FFT 性能和相位补偿运算性能

处理器	FFT (s)	相位补偿运算相乘 (s)
Intel Pentium 4 1.7 GHz	27.2	23.9
Intel Xeon 2.4 GHz	15.1	13.7
MIPS R12000 350 MHz	56.0	68.8

表 1 对比了几种常见处理器的相位补偿运算性能, 测试结果为对一幅 16384×16384 的雷达数据进行相位补偿所需的时间.

4.3 数据转角和节点间数据交换

SAR 成像处理算法在方位处理和距离处理之间需要进行一次数据转角运算. 数据转角运算要求针对处理器的高速缓存进行良好优化.

表 2 给出了采用不同的优化方式时数据转角运算的测试结果. 结果表明, 针对处理器指令集及高速缓存优化后的性能是优化前的性能的 4.28 倍.

表 2 不同优化方式时的数据转角运算性能

优化方式	数据转角运算性能 (MByte/s)
未优化	50.0
针对高速缓存优化	158.5
针对高速缓存及处理器指令集优化	214.0

在具有多个节点的高性能并行处理平台上, 数据转角运算意味着节点之间的数据交换. 这种数据交换是降低并行效率的关键因素.

设每次成像的图像大小为 $N \times N$, 算法在 M 个节点并行执行, 则在每次数据交换的过程中每个节点需要完成转角运算的数据量为 N^2/M , 需要与其它节点交换的数据量为 $2(M-1)N^2/M^2$ (包括从该节点传递到其它节点和从其它节点传递到该节点的数据量), 当 M 足够大时趋近于 $2N^2/M$.

以采用千兆以太网连接的深腾 1800 服务器为例, 使用 MPI 实现的两个节点间的数据交换速度为 $B=95$ MByte/s, 按照上述公式计算的 8 个节点时的等效数据交换速度为 434 MByte/s, 实际使用 MPI 测试得到的数据交换速度为 419 MByte/s, 与预测的结果基本接近.

以上测试项目中, 除了节点间数据交换速度的测试需要使用两个节点以外, 其余测试均可在单节点系统上完成.

5 多节点系统并行效率测试

根据典型运算的测试结果, 可以计算得出相应的高性能计算机系统配置多个节点时的成像处理速度和并行效率. 为了简单起见, 在下面的分析中均不考虑加载原始数据和保存成像结果所需要的时间.

对于标准的成像处理流程, 当配置 M 个节点时, 处理每幅图像所需要的时间为

$$T_M = \frac{1}{M} \left[4T_{\text{fft}} + T_\varphi + 3T_{\varphi \times} + 4T_{\text{tr}} + \frac{8(M-1)}{M} T_{\text{net}} \right] \quad (1)$$

其中 T_{fft} 为执行 N 次 N 点 FFT 或 IFFT 需要的时间, T_φ 为计算所有的相位参考函数需要的时间, $T_{\varphi \times}$ 为执行 $N \times N$ 次相位参考函数相乘需要的时间, T_{tr} 为执行 $N \times N$ 点的转角需要的时间, T_{net} 为在两个节点间交换 $N \times N$ 点数据需要的时间.

定义 M 个节点对应的并行效率为 $\eta_M = T_1 / (MT_M)$, 则

$$\eta_M = T_1 / [T_1 + 8(M-1)T_{\text{net}}/M] \quad (2)$$

当 M 足够大时,

$$\eta_M = T_1 / (T_1 + 8T_{\text{net}}) \quad (3)$$

以上公式成立的条件是当高性能计算机系统的节点增加时, 相应的系统带宽也成比例地增加. 当节点数在一定范围内时, 目前的高性能计算机系统能近似满足这一条件. 式 (3) 说明, 增加节点时, 并行效率将趋近于某一极限值, 而不会无限制地持续降低. 计算结果给出, 增加节点时, SGI Origin 2000 服务器的并行效率将趋近 84%, 而深腾 1800 服务器将趋近 36%.

图 3、图 4 和表 3 给出了两个典型系统的预测结果. 表 4 给出了预测结果与实际测试结果的对比, 两者基本一致.

由式 (2) 还可以得出, 如果要求 $\eta_M \geq \eta$, 则要求两个节点间的数据交换速度:

$$B \geq \frac{8(M-1)}{(1-\eta)MT_1} N^2 S \quad (4)$$

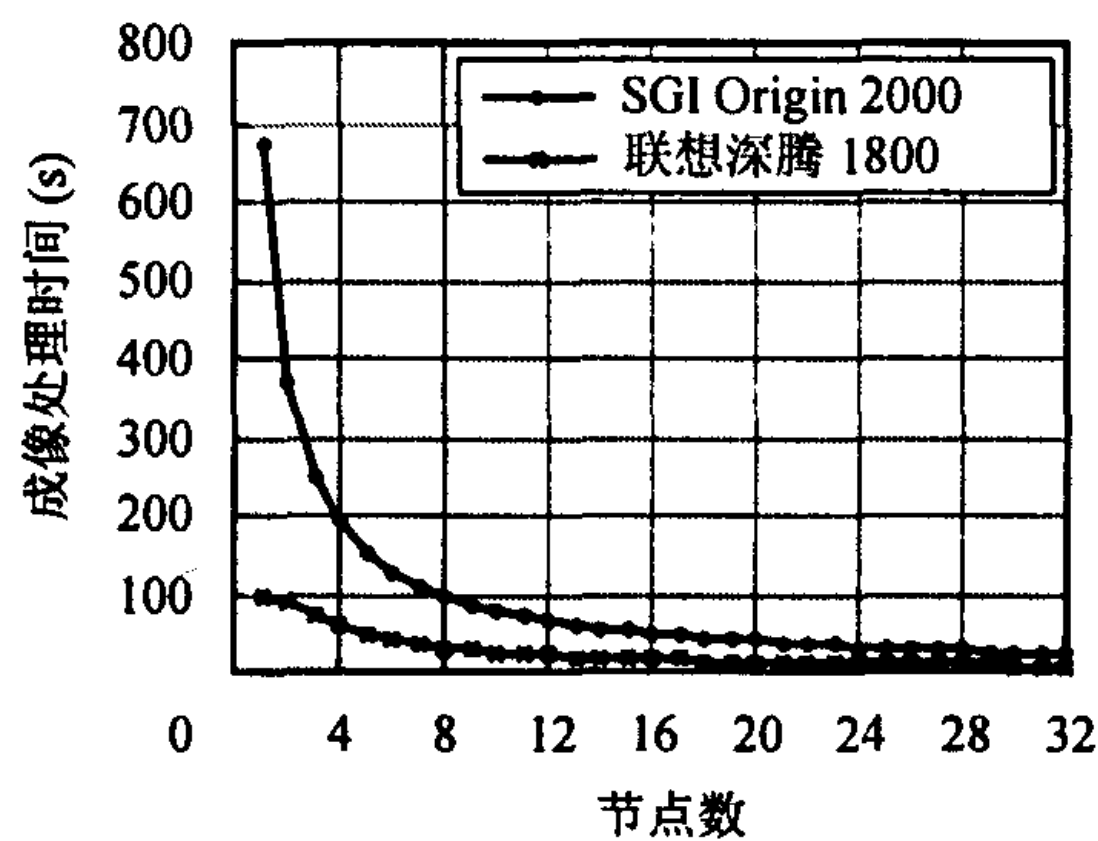


图 3 配置多节点时的成像处理时间

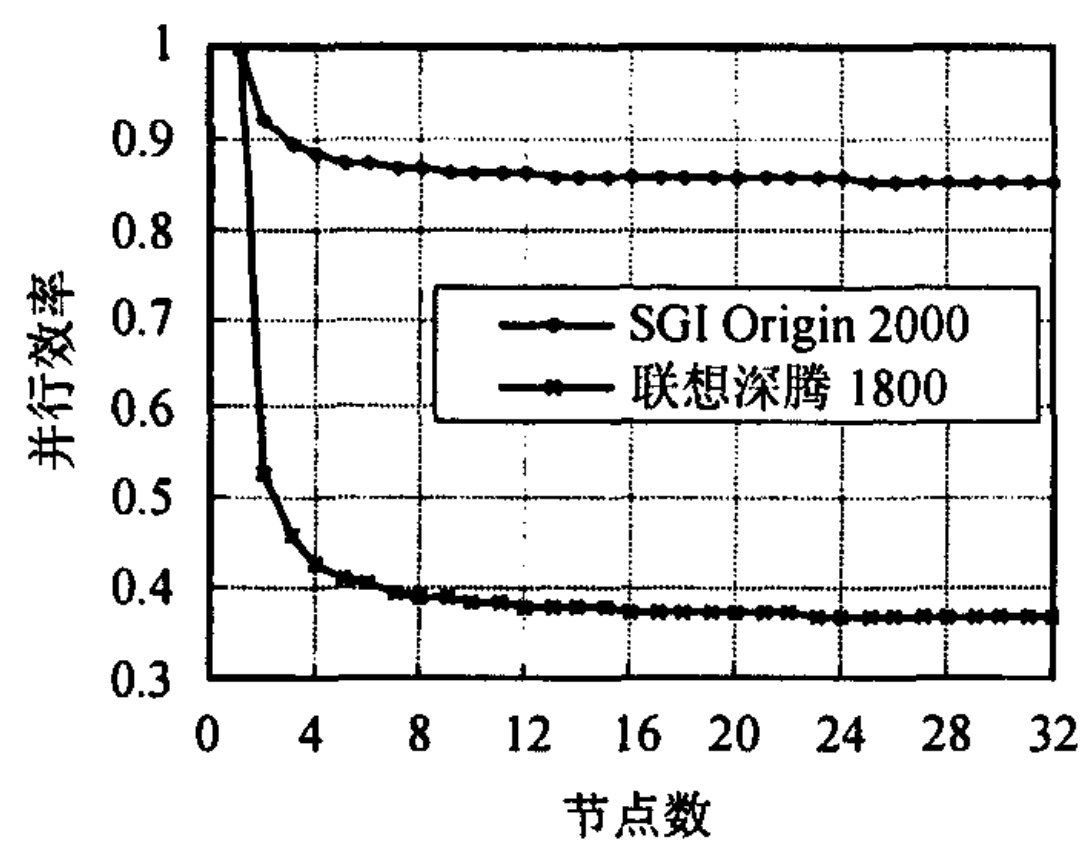


图 4 配置多节点时的并行效率

表 3 配置多节点时的成像处理时间和并行效率

节点数	1	8	16	32
SGI Origin 2100	674 s	97.4 s (86.5%)	49.2 s (85.7%)	24.7 s (85.3%)
联想深腾 1800	99.4 s	31.7 s (39.2%)	16.5 s (37.6%)	8.5 s (36.8%)

表 4 计算预测结果与实际测试结果的对比

系统	节点数	计算预测值 (s)	实际测试值 (s)
SGI Origin 2000	8	97.4	105.8
联想深腾 1800	8	31.7	33.9

其中 S 为每个像素所使用的数据类型的字节数, 如果成像过程中使用单精度浮点数保存中间结果, 则 $S=8$, B 的单位为 MByte/s. 由式 (4) 可知, 对于给定的并行效率要求, 所需的节点间数据交换速度, 亦即高性能计算系统的带宽, 与单个节点的成像处理速度有关.

6 优化方向和策略

根据得到的测试和计算结果可以计算对某一个环节进行的优化所能够带来的性能提升, 从而可以进一步确定优化的方向和策略. 下面从几个可能的方面加以简单讨论.

6.1 针对处理器进行优化

对于 FFT 的实现, 测试结果表明, 在 Pentium 3 处理器上, 使用 SSE 的实现与使用 X87 的实现相比, 可以获得 35.6% 的性能提升.

对于 SIN 和 COS 的计算, 测试结果表明, 在 Pentium 3 和 Pentium 4 处理器上, 采用 FSINCOS 指令与采用 FSIN 和 FCOS 指令相比, 分别可以获得 37% 和 71% 的性能提升^[7]; 采用 SSE 和 SSE2 还可以进一步获得 4.36 倍和 9.46 倍的性能提升 (计算精度稍有降低)^[7]. 在 SGI 的服务器上, 使用软件流水线技术进行优化, 可以获得接近 1 倍的性能提升.

6.2 针对高速缓存进行优化

测试结果表明, 在 Intel 平台上, 同时针对高速缓存和处理器进行优化, 对于数据转角运算, 可以获得 4.28 倍的性能提升, 对于内存复制操作, 可以获得 3.78 倍的性能提升.

6.3 采用速度更高的节点连接方式

采用速度更高的节点连接方式, 可以达到更高的并行效率, 充分利用单个处理节点的处理能力. 分析表明, 对于联想深腾 1800, 如果节点连接方式使用传输速度更快的 Myrinet, 32 个节点时的并行效率可以提高 58% 左右.

7 结论

SAR 成像处理的性能主要受高性能计算机系统架构中的处理器子系统、内存子系统和节点互连方式的影响, 通过对单节点时的 FFT 性能、SIN 和 COS 性能、内存访问速度, 以及两个节点时的节点间数据交换速度进行测试, 可以较为准确地计算同一高性能计算机系统架构在配置不同的节点数时能达到的成像处理速度和对应的并行效率, 在相关的工程实施中作为系统选型和确定配置方案的重要决策参考. 本文针对两种典型的平台给出了相应的测试和计算结果.

根据对测试结果的分析, 可以进一步优化 SAR 成像处理性能的方向和策略. 针对现代处理器的 SIMD 指令和针对高速缓存进行优化, 可以大幅度提高单个节点的处理能力; 采用速度更快的节点互连方式, 可以有效的改善系统的并行效率.

参 考 文 献

- [1] Raney R K, Runge H, Bamler R, Cumming I G, Wong F H. Precision SAR processing using chirp scaling. *IEEE Trans. on Geoscience and Remote Sensing*. 1994, 32(4): 786-799.
- [2] Ding Chibiao, et al.. Large beamwidth spaceborne SAR processing using chirp scaling. IGARSS'99, Hamburg, Germany. 1999, Vol.1: 527-529.
- [3] Intel. Intel Pentium 4 and Intel Xeon Processor Optimization Reference Manual. 2002.
- [4] Intel. IA-32 Intel Architecture Software Developer's Manual. 2002.
- [5] SGI. Performance Tuning Optimization for Origin 2000 and Onyx2. 1997.
- [6] Duhamel P, Holtmann H. Split-radix FFT algorithm. *Electronics Letters*, 1984, 20(1): 14-16.
- [7] Intel. Approximate Math Library for Intel Streaming SIMD Extensions. Oct. 2000.

雷 斌: 男, 1978 年生, 硕士生, 专业方向为信号与信息处理.

丁赤飏: 男, 1969 年生, 研究员, 研究方向为信号与信息处理和系统仿真.

吴一戎: 男, 1963 年生, 研究员, 研究方向为雷达信号处理、计算机并行信号处理等.