

多核龙芯 3A 上二级 BLAS 库的优化^①

李 毅, 何颂颂, 李 恺

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)

摘要: 针对龙芯 3A 体系结构以及二级 BLAS 库函数的特点, 在指令级、存储级和线程级抽取并行方案, 总结了一些合适的优化方法, 并对其进行了定量的分析。实验表明, 这些优化可以将二级 BLAS 函数单线程的性能提升 20% 以上, 多线程下也可以得到 2.5 倍左右的加速比, 这对今后多核龙芯上的系统软件优化工作有着一定的帮助。

关键词: 龙芯 3A; BLAS; 优化; Gemv; Ger; 访存; 多线程

Optimization of BLAS Level 2 Based on Multi-Core Loongson 3A

LI Yi, HE Song-Song, LI Kai

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: According to characteristics of Loongson 3A architecture and BLAS level 2, this article derives the parallel solutions from instruction level, storage level and thread level. We summarize some suitable optimization methods and make a quantitative analysis. Experiment shows that the single-threading performance of BLAS level 2 is increased by 20%, and the multi-threading speedup reaches to 2.5. All of these will give some help to the optimization of system software on multi-core Loongson 3A.

Keywords: Loongson 3A; BLAS; optimization; Gemv; Ger; memory access; multi-threading

1 引言

1.1 多核龙芯 3A 介绍

高性能计算(High-Performance Computing, 简称 HPC)作为一种尖端计算机技术, 广泛用于科学计算、气象预报、地质勘探、金融统计、电子商务、高分子材料研究等领域, 现有的 HPC 核心处理部件-----通用处理器基本都是国外制造, 龙芯作为中国科学院计算机技术研究所研制的新型通用处理器, 具有高性能、低成本、低功耗、高集成度的特点, 可以满足科学研究和企业应用中大部分高性能计算的需求, 它的出现填补国内计算机行业在 HPC 核心技术方面的空白。

最新一代的多核龙芯 3A 主要面向服务器和高性能机的应用, 它采用了 65nm 的制作工艺, 在单芯片内集成了 4 个主频接近 GHz 的 GS464 处理核, 四核共享 4MB 的二级 Cache, 兼容 MIPS64 指令集, 支持 128

位访存部件, 此外和龙芯 2F 一样具备四发射超标量结构, 支持动态调度、寄存器重命名、转移预测等乱序执行技术^[1], 可以最大程度上提升系统的指令级并行和数据级并行。

1.2 数学库 BLAS

BLAS 库是传统的线性数学函数库标准, 具备较好的稳定性以及较高的性能, 能够为关于矩阵的应用提供较优的基础运算支持。Intel 和 AMD 都有分别针对它们处理器的商用 BLAS 库-----MKL 和 ACML, 而基于 MIPS 架构的龙芯没有专用的 BLAS 库, 所以需要大量的移植工作才能保证高性能数学库在龙芯上的正常运行, 由于 ATLAS 移植性较好, 本文所提到的 BLAS 指的都是 ATLAS[2], 它的性能比原始的 Fortran 版 BLAS 高出 3 到 6 倍, 是现今最流行的开源线性数学库之一。

^① 基金项目: 国家高技术研究发展计划(863)(2008AA010902); 自然科学基金(60833004)

收稿时间: 2010-04-29; 收到修改稿时间: 2010-05-27

1.3 本文组织结构

BLAS 总共分为三级，一级完成向量-向量运算，二级完成矩阵-向量运算，三级完成矩阵-矩阵运算。本文主要围绕龙芯 3A 体系结构下二级 BLAS 的优化展开叙述，其中第一章是引言，介绍相应的背景知识，第二章介绍二级 BLAS 库的特点，第三章介绍优化方法及其分析，第四章是综合实验结果，最后是总结与展望。

2 BLAS level 2的特点

二级 BLAS 的访存量和运算量都是 $O(n^2)$ 级，因此性能从很大程度上依赖于数据在存储系统之间的迁移，在多线程结构下任务分配之后可能会调用一级库函数完成单线程的运算。它的特点就是访存连续性强、密集性高，所以优化就是以访存优化为主其他优化为辅，最理想的目标是实现计算指令完全被访存操作掩盖，当然对于具体函数还是要考虑函数本身的特点使用合适的优化方案。

BLAS 库中的矩阵在函数中都会被转化成列主序的存储形式，同列间不同行元素紧凑存储。列间距 LDA 是大于等于行数的参数，在函数实现中通常作为相邻列首地址间隔的参数在程序中使用，每个普通矩阵 A 包含 4 个参数：行数 M、列数 N、列间距 LDA 以及数组的首地址&A，一个 M 行 N 列的普通矩阵如图 1 所示：

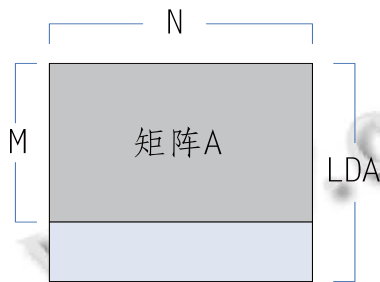


图 1 普通矩阵示意图

二级 BLAS 函数一般是两层循环结构，最典型的的就是普通矩阵向量乘 Gemv 和向量交叉乘 Ger，它们各有四种形式的函数 d、s、z、c，分别代表双精度实数、单精度实数、双精度复数和单精度复数，完成功能如(1)和(2)所示：

$$y = \alpha Ax + \beta y, A - m \times n \quad (1)$$

$$A \leftarrow \alpha xy^T + A, A - m \times n \quad (2)$$

二级 BLAS 函数没有数据分块，具备高循环次数、小循环体的特点，这种结构的特点是编译器通常能够很好地优化该循环，但是手工优化能够提供额外的收益^[3]，具体的优化技术在第三章中将有详细介绍。

3 优化细节分析

随着处理器技术的日渐成熟，与之对应的体系结构也越来越复杂，而现今算法层面的优化已经越来越难，软件优化就必须与硬件体系结构相配合才能得到理想的效果，本节从多种层次上抽取并行方案来分析二级 BLAS 库的优化方法。

3.1 指令优化

使用硬件指令的特性来提高程序性能是一种常见的优化技术，本质上它是利用硬件指令来代替软件逻辑，由于硬件速度要快得多，所以可以有效提升程序的性能。在基于龙芯 3A 的 BLAS 库的优化过程中，我们用到了乘加指令^[4]、128 位访存指令以及非阻塞预取指令。

乘加指令 MADD 是完成浮点操作数相乘后累加的功能，龙芯 3A 上的 MADD 是由 FPU 部件负责的全流水指令，二级 BLAS 存在大量的乘后累加操作，尽可能使用 MADD 指令能将机器指令从两条减少到一条，同时减少了运算时的 RAW 相关，提高了流水线的装填效率。图 2 是 MADD 指令对双精度实数 Gemv 性能的影响，由图可见，即使当 Cache 失效情况下 MADD 指令也有 40%左右的提升。

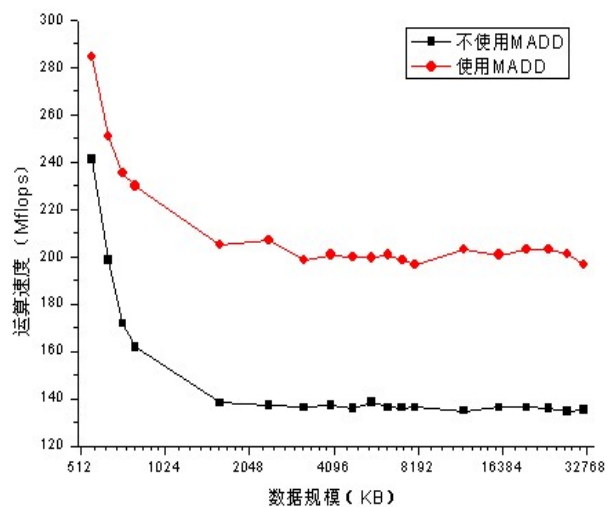


图 2 龙芯乘加指令性能图

128 位访存指令可以利用龙芯 128 位访存部件提高数据的存取效率^[1]。该指令类似于 Intel 的 SSE 系列指令集中 movapd 指令，可以在 5 个周期内完成 2 个双精度数在浮点寄存器和内存间的迁移，这与 64 位的 ldc1 指令有相同的指令延迟。举例来说，实数 Gemv 的访存量是 $MN \times (K+1)/K + M$ ，M、N 是行和列的规模，K 是外层循环展开度，当 M、N 较大且 K 不小于 2 时访存量会大于计算量，在顺序执行的情况下，运算指令会等待访存指令的发射。如果此时利用 128 位访存指令，访存量会减少到，规模较大时访存指令个数减少了将近一半。

多核龙芯 3A 上实现了硬件的非阻塞预取指令^[1]，它的循环间隔只有 1 个周期，而利用 likely 跳转指令的软件预取则需要 10~20 个周期^[5]，此时可以将预取指令直接与其他访存和浮点指令混合，然后乱序提交到指令队列当中，最大限度减少了流水线断流造成的上百个周期的性能损失。

3.2 访存优化

所谓访存指的就是数据在不同存储介质之间的迁移，这些存储介质之间包括外存和内存之间、内存和 Cache 之间、Cache 和寄存器之间。程序员在设计的过程中要考虑到不同介质间数据迁移所引入的代价。访存优化指的就是提升高层存储器中数据的利用率，从而降低数据迁移频率，对于访存和计算相当的二级 BLAS 库来说访存的性能尤为重要。

循环展开是一种常见的优化手段，它以增加代码尺寸为代价，降低分支转移次数、提高指令调度空间、减少数据相关度、增强寄存器可重用性^[3]。与循环展开密切相关的概念是循环展开因子，它指的就是循环展开的粒度，gcc 默认展开因子是 2 或者 4，该因子的选择比较保守，不能发挥龙芯 32 个通用浮点寄存器的性能。我们在手动循环展开的时候会根据函数本身的特点选择合适的展开因子，一般来说访存数据越多、运算越复杂，展开因子的选择不宜越大。

龙芯 Cache 使用的是随机替换策略，当运算规模超过 Cache 容量时被替换出去的数据很有可能是下次循环需要使用的数据，从而造成访存失效。举例来说，对于行列为 N 的方阵 Gemv，不展开时向量 x 需要重复访问 N 次，此时整个函数的访存量就是 $2N^2 + N$ ；此时如果进行外层循环展开会减少 x 向量的访存次数，

整个访存量就会变成 $(1+1/K)N^2 + N$ ，由于计算量达到了 $2N^2$ ，所以只要使得 $(1+1/K)N^2 + N < 2N^2$ 即 $K1 > N/(N-1)$ 时，可以为运算访存的重叠创造必要条件。由于 N 一般比较大，所以理论上外层展开度 K1 只要大于等于 2 即可，而编译器本身就具备这种优化能力，但是实际情况是龙芯本身的计算能力就远强于供数能力，所以手动循环展开减少访存指令数量是一种非常有效的方法^[6]，此时外层循环展开度的选择一般就比默认的 2 或 4 更好，图 3 是双精度 Ger 使用 8×8 和 4×4 循环展开后的性能比较，可图可见，合适的循环展开因子可以有效提升 BLAS level 2 的性能。

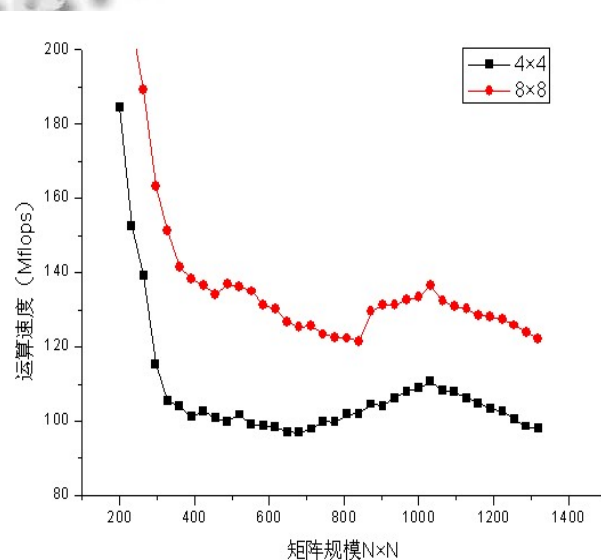


图 3 双精度 Ger 循环展开性能比较图

对于部分二级库函数，还可以使用循环融合的方法进行优化。循环融合就是把具有相同控制的相邻循环合并成同一个循环，它可以减少循环开销，在并行化循环之前增加循环内的工作粒度，另外融合的同时可以合并引用相同数据的循环，从而改进数据的局部性^[3]。部分二级库函数存在多次乘法、加法或乘加的运算，此时数据相关的输入或输出操作数被剥离到不同的循环结构当中，如果运算规模非常大的话就会存在已经被替换出 Cache 的数据会被再次提交使用的情况，从而造成不必要的内存到 Cache 的数据迁移，这样的代价完全可以通过循环融合来避免。比如二级函数 Syr2 完成 $y \leftarrow \alpha xyT + \alpha yxT + A$ 的操作，原库中使用了两次循环的 Ger。此时合并两个循环减少向量 x、y

和矩阵 A 的 load 次数,使得两次乘加在一个循环内完成,运算访存比的上限就会由 $2K/(1+K)$ 增加到 $4K/(2+K)$, K 为外层循环展开度。该比值的增加意味着性能提升空间的上升,图 4 显示的是双精度循环融合后性能上限的比较,基于乘加指令的循环融合可以将性能上限提升将近 40%多, Gemv 或 Ger 中循环展开减少的仅仅是一个输入向量的访存,而在 Syr2 中循环融合可以同时减少两个输入向量和一个输入矩阵的访存。

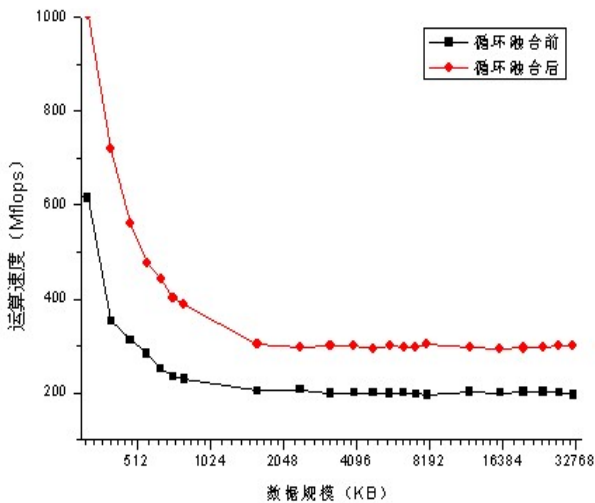


图 4 syr2 循环融合前后性能上限比较图

3.3 线程优化

线程优化指的就是在多核处理器对多个线程执行任务的合理划分。任务划分的基本原则就是均衡负载、降低线程间通讯、减少共享变量。和前面的优化相比,线程本身的调度都是交给操作系统完成,和体系结构的关系不是那么密切。当然,并非在多核处理器上多线程就能取得明显的加速效果,一些问题本身就是比较复杂难以并行化的,线程数量越多反而会增加程序本身调度的开销。能够利用多核多线程的应用主要是计算密集型应用和 I/O 密集型应用^[7], BLAS 库就是典型的浮点计算密集型应用,对于运算量为 $O(n^2)$ 的二级 BLAS 来说,多线程的优化可以更大程度上利用系统的数据通路。OpenMP 是在共享存储体系结构上的一个编程模型,具备简单、可移植性好和可扩展等优点^[8],它也是我们线程优化时的工具。

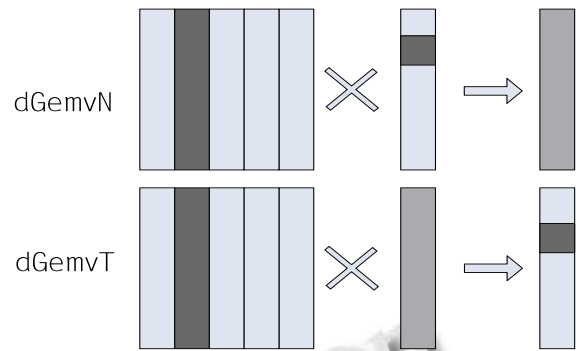


图 4 GemvN 和 GemvT 多线程任务分配比较图

非转置的 GemvN 和转置的 GemvT 都使用 8×4 的循环展开,由图 4 中容易看出, dGemvN 每个线程完成矩阵 A 一列 8 个元素和向量 x 中一个元素分别相乘的过程,最后将 y 中 8 个元素都各更新一次,它的线程内地址非连续,相邻线程数据首地址间隔 LDA; dGemvT 每个线程完成矩阵 A 一列元素和向量 x 中的一整列做乘积后累加,最后把结果加到 y 向量的对应元素上,它的线程内地址连续,线程间间隔跳变。实测表明在双路四核的处理器上,8 线程的 dGemvT 的性能要比 dGemvN 高出 20% 以上,所以保证线程内数据的局部性可以有效利用处理器资源以及多核计算能力。

4 实验结果

实验平台选用的是 KD-60,它是我国首台采用国内自主设计、具有自主知识产权的龙芯多核通用处理器的万亿次高性能计算机系统,总共集成了 80 个龙芯 3A 四核通用处理器,主频为 825MHz,峰值计算速度为 1TFlops,具有高计算密度、低功耗、低成本、低占地等特点^[9]。操作系统是 Linux 2.6.26,编译器是支持 n32 的 gcc 4.3.2,ATLAS 选用的是最新版的 3.8.3,测试的函数就是第二章提及的双精度实数 Gemv 和 Ger,矩阵和向量的参数都是列主序。

图 5 和图 6 所示的是利用第三章的优化方法得到的数据,测试选用的都是 Cache 失效情况下的矩阵规模,单线程平均能得到 20% 左右的提升。在使用 MADD 指令和 128 访存指令的情况下,运算量和访存量都缩小了一半,循环展开后运算访存比接近 2: 3,四核 4 线程运算加速比为 4 而数据拷贝的加速比接近 2,假设程序全流水运行,此时理论上加速比为 $(2 \times 4 + 3 \times$

2)/(2+3)=2.8, 由图 5 和图 6 可见 4 线程加速比大概在 2 倍多, 表 1 所列的是 Gemv 和 Ger 在 Cache 是否失效情况下的加速比。

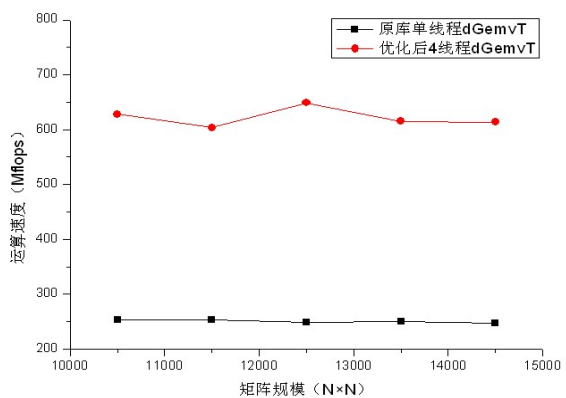


图 5 双精度实数 GemvT 性能比较图

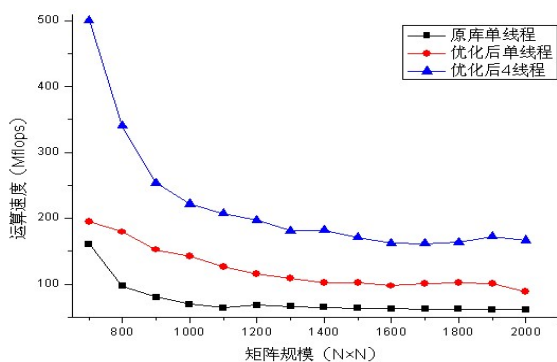


图 6 双精度实数 Ger 性能比较图

表 1 二级 BLAS 加速比对照表

函数名	运算规模	4 线程平均加速比
双精度 Gemv	二级 Cache 内	3.1
	Cache 失效	2.3
双精度 Ger	二级 Cache 内	3.6
	Cache 失效	2.6

以上的实验结果说明了在多核龙芯 3A 上, 我们对二级 BLAS 库的优化得到了较好的加速性能。

5 总结与展望

本文介绍了二级 BLAS 库函数特点, 同时详细说明了它们在龙芯 3A 上的一些优化方法。其中核心介绍了二级库中使用最频繁、被调用次数最多的二级函数 Gemv 和 Ger, 单线程性能提升达到 20% 左右, 多线程情况下也能有 2 倍以上的加速比, 达到了预期的效果。

龙芯 3A 作为第一款多核龙芯处理器, 配套的系统软件移植和优化工作尚处在起始阶段, 本文为我们以后的后续工作打下了基石, 随着项目的逐步深入, 定量的综合优化技术方案尚待于我们进一步的探索。

参考文献

- 1 中国科学院计算技术研究所. 龙芯 3A 处理器用户手册 0.1 版. 2009.
- 2 Whaley RC, Petitet A, Dongarra JJ. Automated Empirical Optimization of Software and the ATLAS Project. September 19, 2000.
- 3 Gerber R, Smith KB. 王涛, 单久龙, 孙广中译. 软件优化技术. 北京: 电子工业出版社, 2007.
- 4 Price C. MIPS IV Instruction Set. Rev 3.2. September 1995.30-66.
- 5 顾乃杰, 李凯, 陈国良, 等. 基于龙芯 2F 体系结构的 BLAS 库优化. 中国科学技术大学学报, 2008, 38(7): 855-858.
- 6 苏波, 李凯, 徐志广, 何颂颂. 龙芯 2F 上的访存优化. 计算机系统应用, 2010, 19(1): 171-175.
- 7 Butenhof DR. Programming with POSIX Threads. Addison-Wesley, June, 1998.
- 8 陈国良, 安虹, 陈峻, 郑启龙, 单久龙. 并行算法实践. 北京: 高等教育出版社, 2004.
- 9 KD-60. [2010-4-7]. <http://210.45.114.167/>