软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*,2011,22(Suppl.(2)):208−216

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

# OpenBLAS:龙芯 3A CPU 的高性能 BLAS 库*

张先轶[1,3+], 王 茜[1,3], 张云泉[1,2]

[1](中国科学院 软件研究所 并行软件与计算科学实验室,北京 100190)

[2](中国科学院 计算机科学国家重点实验室,北京 100190)

[3](中国科学院 研究生院,北京 100190)

## OpenBLAS: A High Performance BLAS Library on Loongson 3A CPU

ZHANG Xian-Yi[1,3+], WANG Qian[1,3], ZHANG Yun-Quan[1,2]

[1](Laboratory of Parallel Software and Computational Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100190, China)

[2](State Key Laboratory of Computing Science, The Chinese Academy of Sciences, Beijing 100190, China)

[3](Graduate University, The Chinese Academy of Sciences, Beijing 100190, China)

+ Corresponding author: E-mail: xianyi@iscas.ac.cn

**Zhang XY, Wang Q, Zhang YQ. OpenBLAS: A high performance blas library on loongson 3A CPU**. ***Journal of Software*, 2011,22(Suppl.(2)):208−216.** http://www.jos.org.cn/1000-9825/11042.htm

**Abstract**: BLAS is a fundamental math library in scientific computing. Thus, each CPU vendor releases optimized BLAS library for its own CPU. Loongson CPU series are developed by the Institute of Computing Technology, Chinese Academy of Sciences. In 2010, it released Loongson 3 CPU series. This paper introduces the open source BLAS library OpenBLAS, which is forked on GotoBLAS 2-1.13 BSD version. BLAS Level 3 functions of OpenBLAS is optimized on Loongson 3A quad cores CPU. In sequential optimizations, blocking, hand coding assembly kernel, Loongson 3A special instructions and reordering instructions are utilized. The performance of BLAS Level 3 subroutines exceeded GotoBLAS and ATLAS by about 75% and 17%. Meanwhile, it exceeded GotoBLAS and ATLAS by about 103% and 36% in double precision functions. In parallel multi-threads optimization, this study used interleaved data buffer layout to avoid shared L2 Cache conflictions among multi-threads. OpenBLAS achieved 3.47 speedups on quad cores. In 4 threads, the performance of OpenBLAS BLAS Level3 functions exceeded GotoBLAS and ATLAS by about 69% and 34%, 89% and 55% in double precision functions.

**Key words**: BLAS; Loongson 3A; optimization; multicore; MIPS64

**摘 要**: BLAS 是科学计算中最基础的数学库之一,各 CPU 厂商都推出了针对各自 CPU 的优化 BLAS 库.龙芯 CPU 是中国科学院计算技术研究所自主研制的通用 CPU,目前已推出了龙芯 3 号系列.介绍了基于 GotoBLAS 2-1.13 BSD 版的开源项目 OpenBLAS,针对龙芯 3A CPU 的优化工作.在 BLAS 3 级函数的单线程优化上,运用了分块,手工核心汇编,使用龙芯 3A 扩展指令、汇编指令重排等技术.BLAS 3 级函数平均性能高于 GotoBLAS 和 ATLAS

* 基金项目: 国家自然科学基金(61133005);国家高技术研究发展项目(863)(2009AA01A129, 2009AA01A134);国家重大专项核高基项目(2009ZX01036-001-002)

收稿时间: 2011-07-15; 定稿时间: 2011-12-02

75%和 17%.其中,双精度函数高于 GotoBLAS 和 ATLAS 103%和 36%.在 BLAS 3 级函数并行化方面,采用数据缓冲区交错布局等技术,减少多线程对共享 L2 Cache 的争抢.OpenBLAS BLAS 3 级函数的 4 线程并行加速比达到 3.47.4 线程 BLAS 3 级函数平均性能高于 GotoBLAS 和 ATLAS 69%和 34%,其中,双精度函数高于 GotoBLAS 和 ATLAS 89%和 55%.

**关键词**: BLAS;龙芯 3A;性能优化;多核;MIPS64

BLAS(basic linear algebra subprograms)是一个基本线性代数核心子程序集,主要包括向量和矩阵的基本操作[1].BLAS 分为 3 级:BLAS 1 级实现向量和向量的操作,BLAS 2 级实现矩阵和向量的操作,BLAS 3 级实现矩阵和矩阵的操作.其中,BLAS 3 级的性能高低至关重要.比如,在高性能计算领域中广泛采用的 LINPACK 性能测试,严重依赖 BLAS 库 3 级的性能.因此,各个 CPU 厂商都推出了针对各自 CPU 优化的 BLAS 实现,比如 Intel 公司的 MKL[2],AMD 公司的 ACML[3],IBM 公司的 ESSL[4].同时,也有 BLAS 的开源实现,比如核心手工汇编优化的 GotoBLAS[5],应用自适应优化技术的 ATLAS 等[6].

龙芯 3A CPU 是第一款龙芯 3 号多核处理器,其基于可伸缩的龙芯 3 号系列多核互联架构设计,集成了 4 个 GS464 处理器核,4 个 2 级缓存模块[7].每个 GS464 处理器核支持 MIPS64 指令集,此外增加了 128bit 访存和 X86 虚拟机指令等;采用 4 发射超标量结构,包含两个定点,两个浮点和 1 个访存部件;具有 32 个逻辑寄存器,64 个物理寄存器.Cache 行大小为 32 字节,每个处理器核含有独立的 L1 指令 Cache 和 L1 数据 Cache,采用 4 路组相联随机替换策略,容量各为 64KB.L2 Cache 为 4 个处理器核共享 4MB,同样使用 4 路组相联随机替换策略.

OpenBLAS 是我们在 GotoBLAS 2-1.13 BSD 版基础上发起的 BLAS 库开源项目,项目主页见文献[8],当前主要工作是为龙芯 3A CPU 提供高性能的 BLAS 库实现.本文的主要贡献是:针对龙芯 3A CPU 的结构特点,通过选择合适的分块,手工汇编核心函数,应用龙芯的 128bit 访存指令和预取指令以及指令重排等,优化 GEMM(矩阵乘法)函数的单线程性能,从而实现全部 BLAS 3 级函数的单线程优化;优化多线程实现方式,特别是数据布局交错布局,减少龙芯 3A CPU 中多核对共享的 L2 Cache 的争抢,提高了多线程性能和加速比;在龙芯 3A CPU 上,通过与 GotoBLAS 和 ATLAS 库的性能对比,展示了 OpenBLAS 的优化效果和未来改进方向.

本文第 1 节主要介绍 OpenBLAS 针对龙芯 3A CPU 的 BLAS 3 级优化,包括单线程的 GEMM 优化和多线程并行化方面的优化技术.第 2 节评测 OpenBLAS 性能的单线程和多线程性能,与 GotoBLAS 和 ATLAS 进行对比和分析.第 3 节介绍在龙芯 CPU 上优化 BLAS 函数的相关工作.第 4 节总结和展望.

## 1 针对龙芯 3A CPU 的 BLAS 3 级优化

本节重点介绍 BLAS 3 级中,矩阵乘法 GEMM 函数的优化方法.BLAS 3 级中的其余函数,可将其计算核心转化为 GEMM 函数的调用,算法见文献[9].

### 1.1 GEMM优化

根据 Goto 等人的论文[10],矩阵乘法 GEMM,在进行合理的分块后,其计算核心如图 1 所示,最内层的循环为 GEBP,即块状(block)的子矩阵 A 乘以条状(panel)的子矩阵 B,最后累加至子矩阵 C.GEBP 的实现细节如图 2 所示,在 X86 CPU 等平台上,由于数据处于 L2 Cache,并不会使处理器的流水线停顿,所以选择合适的 $Mc$ 和 $Kc$(通常为 L2 Cache 大小的一半),使得子矩阵 A 存在 L2 Cache 中.而对于子矩阵 B 的宽度 $Nc$ 的选择上,需要满足子矩阵 B 的大小不能超过 TLB 所覆盖的大小.进一步的,使用 $Mr\times Nr$ 的寄存器分块技术,使用 $Mr$ 个寄存器保存当前的 A 元素,$Nr$ 个保存 B 元素,以及 $Mr\times Nr$ 个 C 元素.同时,还需 $Mr$ 和 $Nr$ 个寄存器,存储下一组的 A 和 B 元素.通过此种方法,Goto 等人在 X86 CPU 上得到了较高的 GEMM 性能.

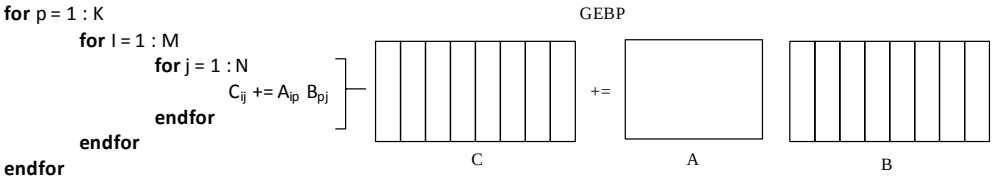



图 1 矩阵乘法核心 GEBP 示意图

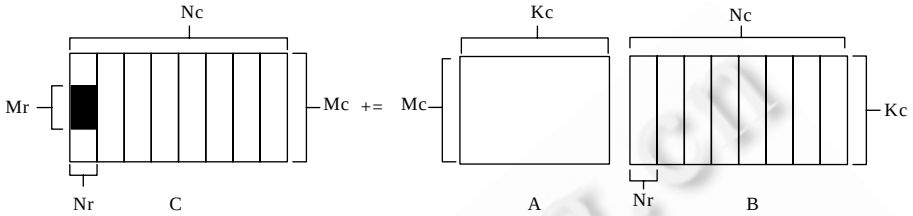



图 2 GEBP 实现细节图

然而,此 X86 CPU 上的优化算法和假设等并不能简单移植到龙芯 3A.首先,杨荣秋等人在文献[11]中已经分析龙芯 2F CPU,如果数据存在 L2 Cache 中,会造成流水线停顿从而影响性能.而龙芯 3A 的处理器核与龙芯 2F 的硬件参数特性类似,所以存在同样的性能瓶颈;其次,由于 L1 Cache 和 L2 Cache 都是使用了随机替换策略,较不容易预测替换行为,所以合适的分块大小尤为重要.如果分块过大,造成 A,B 和 C 的元素在 Cache 中冲突加剧,在随机替换策略下,必然会造成部分期望常驻 Cache 的 A 元素被挤出.

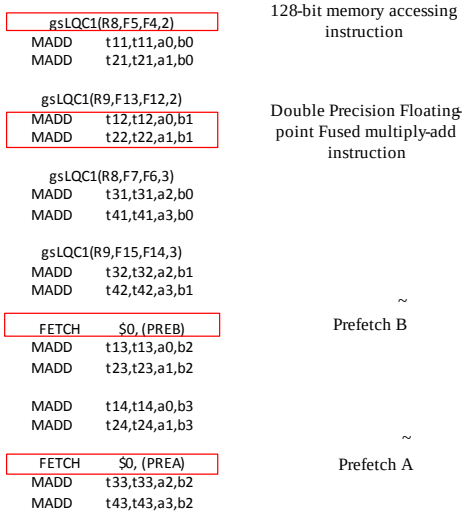



图 3 GEBP 核心汇编代码片断

针对以上在龙芯 3A CPU 上所遇到的问题,我们进行了如下针对性优化:

首先,减小子矩阵 A 的大小,保证其存放在 L1 Cache 中,而不是 L2 Cache 中,避免访问 L2 Cache 而造成的流水线停顿.比如,在双精度 DGEMM 下,经过我们实验,较优的参数是 *Mc* 为 32,*Kc* 为 116,*Nc* 为 1 000.其中,子矩阵 A 的大小为 29KB,占用 L1 Cache 的不到一半.

其次,由于龙芯 3A CPU 中每个处理器核具有 32 个逻辑寄存器,所以我们选取 4×4 的寄存器分块,即每次计算 4 个 A,4 个 B,16 个 C,读取下一组的 4 个 A,4 个 B,从而占满 32 个寄存器.

再次,使用龙芯 3A CPU 中特有的 128-bit 的访存指令 gsLQC1 等,即每次可读写 2 个 64-bit 寄存器.同时,配合龙芯 3A CPU 的预取指令,对子矩阵 A 和子矩阵 B 进行适当预取.

最后,指令重排.因为龙芯 3A 处理器核心每时钟周期可以发射 2 条浮点指令和 1 条访存指令,所以我们将浮点计算指令和访问指令进行交叉的重排.重排后的核心代码片断如图 3 所示,在图中,我们已将不同类型指令用线框标出,可以看到我们对 gsLQC1,预取指令和浮点乘加计算指令的排布.

### 1.2 多核并行化

在多核并行化上,初始使用了对列方向进行划分的方法,4 线程的并行化如图 4 所示,每线程计算对应的 C 部分.但应用此种初始的并行化方法,并不能得到良好的加速比.

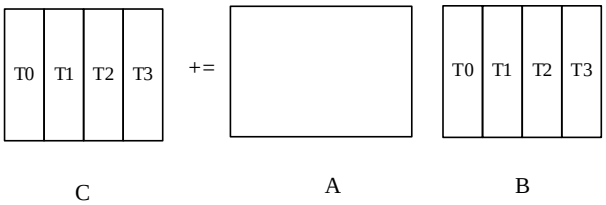



图 4 矩阵乘法在 4 线程下的划分示意图

我们采用测量弱可扩展性的方法,即保持每个处理器核计算相同的工作负载,比较单核与 4 核中各线程,DGEMM 分别在打包子矩阵 A,打包子矩阵 B 和 GEBP 核心的时间,如果两者的时间越接近,说明并行加速效果越好.结果见表 1,GEBP 核心和打包子矩阵 A 在 4 线程并行化后比单线程变慢 2 倍以上,打包子矩阵 B 变慢 4 倍.通过计算分块大小,打包后的子矩阵 B 所需占用的空间约为 906KB,在单线程下,可以顺利存在 4MB 的 L2 Cache 而不引起过多的冲突.但是在 4 线程下,所需的子矩阵 B 总空间约为 3.6MB,必然会导致各线程间对共享的 L2 Cache 的争抢,从而导致打包子矩阵 B 时间大幅增长,也造成了之后 GEBP 核心的明显变慢.

**表 1** DGEMM 中核心函数性能对比表

| | 打包子矩阵 A | 打包子矩阵 B | GEBP 核心 |
|---|---|---|---|
| 单核时间(s) | 2.782 | 0.586 | 14.122 |
| 4 核中各线程平均时间(s) | 7.439 | 2.586 | 34.271 |

因此,我们的主要优化思路是减少各线程在共享 L2 Cache 上的冲突.一方面,我们减小多线程下分块 B 的大小,从而减小对 L2 Cache 的消耗,比如对于 DGEMM,在 4 线程下将 *Nc* 值减小为原来的 1/4.另一方面,将各个线程的打包后子矩阵 A 与打包后子矩阵 B 数据缓存,进行交错的排列布局.如图 5 所示,为 4 线程情况下打包后的子矩阵 A 与 B 的内存布局图,此方法可尽量避免在 L2 Cache 中各个线程间的相互冲突和替换.

| T0 Buffer A | T1 Buffer A | T2 Buffer A | T3 Buffer A | T0 Buffer B | T1 Buffer B | T2 Buffer B | T3 Buffer B |
|---|---|---|---|---|---|---|---|

图 5 多线程子矩阵 A 与 B 交错布局示意图

## 2 性能测试与评价

### 2.1 实验平台与测试用例

龙芯 3A 测试平台的配置见表 2,为利用龙芯 3A CPU 上的两个内存控制器,配置 2 条 2GB 的内存.在编译器上,我们使用了最新的 GCC 4.6.1 版本,其已增加了针对龙芯 3A 处理器的优化补丁.在作为性能对比的 BLAS 库上,我们选取所能得到的 GotoBLAS 和 ATLAS 的最新版本,其中 ATLAS 使用了 GCC 编译器的 -march=loongson3a 参数,使用 GCC 针对龙芯 3A 的优化,以期在龙芯 3A CPU 上通过自动调优得到较好的性能.

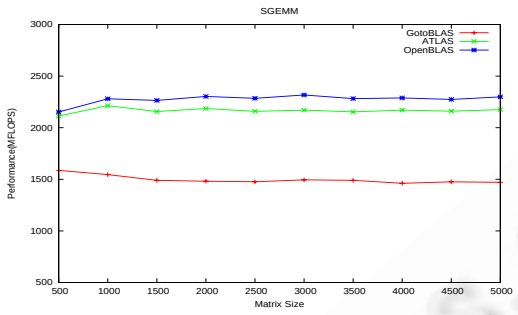
**表 2** 龙芯 3A 测试平台配置表

| | |
|---|---|
| CPU | 1 路龙芯 3A(4 核),主频 800MHz |
| 内存 | 2 条 DDR-3 2GB 内存,共 4GB |
| 操作系统 | Debian 6.0,内核 Linux-2.6.36 |
| 编译器 | GCC-4.6.1(支持–march=loongson3a) |
| GotoBLAS | GotoBLAS2-1.13 BSD 版,编译参数: –march=mips64 –mabi=64 –O2 |
| ATLAS | ATLAS-3.9.45 版,编译参数: –march=loongson3a –mabi=64 –O3 |
| OpenBLAS | 编译参数: –march=mips64 –mabi=64 –O2 |

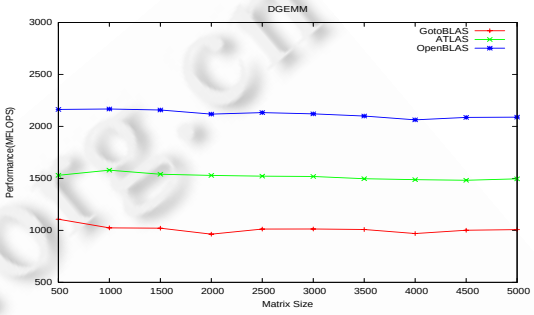
在测试用例上,我们选择了方阵进行测试,元素值采用随机生成的方法.同时,为评价在不同规模下的性能,我们选取从 500~5000,步长为 500 的共 10 种矩阵维数进行测试.

### 2.2 单线程性能对比

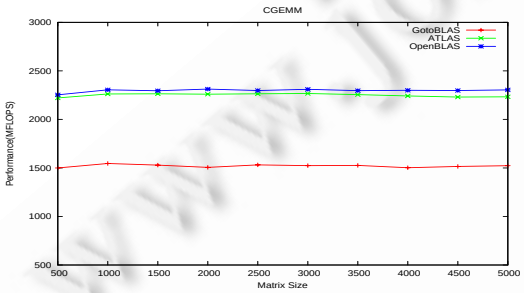
图 6 展示了 BLAS 3 级中的核心函数 GEMM 在 4 种精度下的性能对比,可以看到 OpenBLAS 在测试的所有情况下都取得了最优的性能,ATLAS 使用自适应优化技术后性能居中,而 GotoBLAS 因为没有针对龙芯 3A CPU 进行优化,所以性能最差.在双精度和双精度复数情况下,OpenBLAS 优势明显,平均性能超过 ATLAS 39.6%和 27.7%,相对于 GotoBLAS 平均性能提升分别为 109.5%和 107.3%.在单精度和单精度复数下,OpenBLAS 相对于 ATLAS 的优势并不明显,分别只高出 5.0%和 2.0%;相对于 GotoBLAS,性能分别提升 52.0%和 51.1%.因此,OpenBLAS 在单精度和单精度 GEMM 还具有明显的改进空间,我们推测是在 OpenBLAS 中没有使用单精度浮点向量(ps)指令,造成性能稍低.



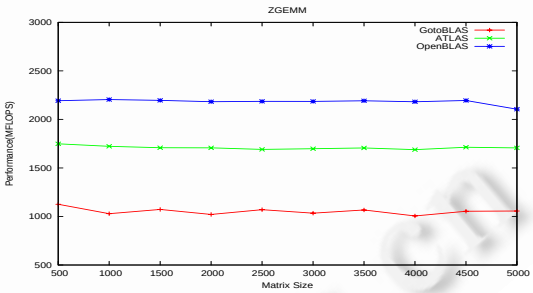

(a) 单精度 SP Real GEMM




(b) 双精度 DP Real GEMM




(c) 单精度复数 SP Complex GEMM




(d) 双精度复数 DP Complex GEMM

图 6 单线程 GEMM 性能对比图

图 7 是 BLAS 3 级所有函数单线程平均性能对比.与 GEMM 函数的结果类似,OpenBLAS 在所有函数中的平均性能都最优,ATLAS 居中,GotoBLAS 性能最差.OpenBLAS 相对于 GotoBLAS 和 ATLAS 的平均加速比分别为 1.75 和 1.17,各精度下的加速比见表 3.在双精度和双精度复数的函数上,取得了较优的性能,但是在单精度和单精度复数上,相对 ATLAS 优势并不明显,还存在一定优化空间.

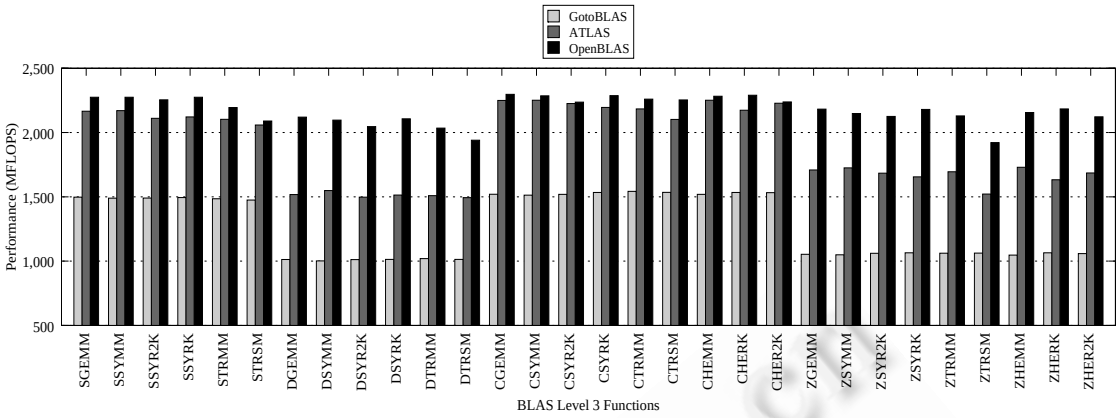



图 7 单线程 BLAS 3 级函数平均性能对比图

**表 3** OpenBLAS 单线程性能对比表

| | 单精度 BLAS 3 级函数 | 双精度 BLAS 3 级函数 | 单精度复数 BLAS 3 级函数 | 双精度复数 BLAS 3 级函数 |
|---|---|---|---|---|
| 相对于 GotoBLAS 加速比 | 1.50 | 2.03 | 1.49 | 2.01 |
| 相对于 ATLAS 加速比 | 1.05 | 1.36 | 1.03 | 1.27 |

### 2.3 多线程性能对比

图 8 为 4 线程的 BLAS 3 级中的核心函数 GEMM 在 4 种精度的的性能对比,图 9 为 4 线程的 BLAS 3 级所有函数的平均性能对比.与单线程的结果类似,OpenBLAS 在测试的所有情况下都取得了最优的性能,ATLAS 性能居中,GotoBLAS 性能最差.其中,OpenBLAS 的 DGEMM 在 4000 规模下,性能存在较大的下滑,我们推测与采用的多线程间的子矩阵 A 和 B 的交错布局算法有关,还需进一步的分析和改进.

**表 4** BLAS 库 4 线程并行化加速比表

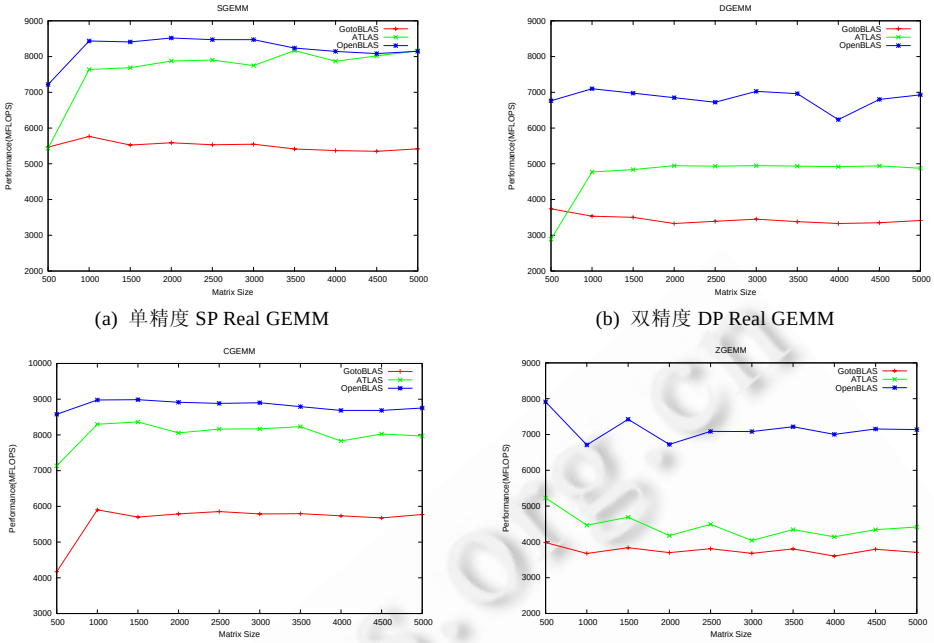
| | 单精度 BLAS 3 级函数 | 双精度 BLAS 3 级函数 | 单精度复数 BLAS 3 级函数 | 双精度复数 BLAS 3 级函数 |
|---|---|---|---|---|
| GotoBLAS | 3.64 | 3.35 | 3.77 | 3.54 |
| ATLAS | 3.23 | 2.74 | 3.45 | 2.82 |
| OpenBLAS | 3.62 | 3.12 | 3.84 | 3.31 |

GotoBLAS,ATLAS 和 OpenBLAS 在 4 线程下的并行化加速比见表 4,GotoBLAS 的并行化加速比较高,OpenBLAS 的加速比比 GotoBLAS 稍低,而 ATLAS 的加速比与其余两者差距较大.GotoBLAS 的加速比很高,我们认为是由于其单线程最慢,所以造成了并行效果好.而 ATLAS 的并行化效果最差,说明其在并行实现方式上存在一定的改进空间.OpenBLAS 的并行化效果较好,在 4 线程下 BLAS 3 级整体平均加速比为 3.47.

表 5 汇总了 4 线程的 OpenBLAS 在 BLAS 3 级函数在各精度下相对于 GotoBLAS 和 ATLAS 的平均加速比.整体上,相对于 GotoBLAS 和 ATLAS 的平均加速比分别为 1.69 和 1.34.由于 ATLAS 的并行化效果不理想,所以 OpenBLAS 相对 ATLAS 的加速比高于单线程的加速比.

**表 5** OpenBLAS 4 线程性能对比表

| | 单精度 BLAS 3 级函数 | 双精度 BLAS 3 级函数 | 单精度复数 BLAS 3 级函数 | 双精度复数 BLAS 3 级函数 |
|---|---|---|---|---|
| 相对于 GotoBLAS 加速比 | 1.49 | 1.89 | 1.52 | 1.88 |
| 相对于 ATLAS 加速比 | 1.19 | 1.55 | 1.15 | 1.50 |




(a) 单精度 SP Real GEMM (b) 双精度 DP Real GEMM

(c) 单精度复数 SP Complex GEMM (d) 双精度复数 DP Complex GEMM

图 8 4 线程 GEMM 性能对比图

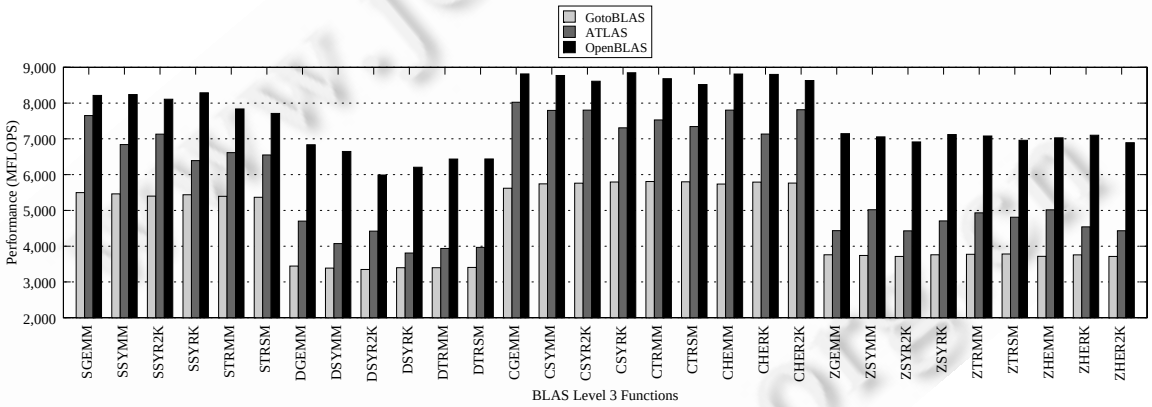



图 9 4 线程 BLAS 3 级函数平均性能对比图

## 3 相关工作

杨荣秋等人[11]探讨了在单线程双精度矩阵矩阵乘函数(DGEMM)在龙芯 2F CPU 上的优化工作,提出了地址交错的算法,改进了 L1 Cache 冲突的问题,并在龙芯 3A CPU 的模拟器上,进行了进一步的优化和改进.苏波等人[12]在龙芯 2F CPU 上对 ATLAS 进行改进,通过多种优化访存的技术,使其性能提升 50%.顾乃杰等人[13]介绍了在龙芯 2F 上针对 DGEMM 的优化以及 LINPACK 测试的结果.李毅等人[14]针对龙芯 3A CPU,在 ATLAS 的 BLAS 2 级的部分,进行了单线程和多线程的优化工作.据我们所知,目前尚未有针对龙芯 3A CPU 的所有 BLAS 3 级函数单线程和多线程优化的文献.

## 4 结论与展望

本文介绍了 OpenBLAS 在 BLAS 3 级函数,针对龙芯 3A CPU 的硬件特性,通过选择合适分块,手工汇编核

心函数,应用龙芯的 128-bit 访存指令和预取指令以及指令重排等技术,进行单线程优化.从而取得了较高的性能,OpenBLAS BLAS 3 级函数相对于 GotoBLAS 和 ATLAS 的平均加速比分别为 1.75 和 1.17,其中,双精度函数相对于 GotoBLAS 和 ATLAS 加速比达到了 2.03 和 1.36.在并行化上,我们通过优化并行实现方式和数据布局的方法,减少多线程间了龙芯 3A CPU 共享的 L2 Cache 的冲突,得到了较好的并行效果.OpenBLAS 4 线程的 BLAS 3 级的平均加速比为 3.47,相对于 GotoBLAS 和 ATLAS 的平均加速比分别为 1.69 和 1.34.其中,双精度 BLAS 3 级相对于 GotoBLAS 和 ATLAS 的加速比分别为 1.89 和 1.55.

OpenBLAS 在单精度和单精度复数的 BLAS 3 级函数上,相对于 ATLAS 的优势不明显,单线程下只高出 5%和 3%,说明性能还存在提升空间,是未来工作的主要方向之一.此外,还需继续针对龙芯 3A CPU 进行 BLAS 2 级和 1 级的优化工作.

**致谢** 感谢中国科学院计算技术研究所的孟小甫,北京龙芯中科技术有限公司的李文刚和孙志等人,在龙芯 3A 实验平台的调试与测试上提供的帮助.感谢杨荣秋在针对龙芯的矩阵乘法优化上对我们的指导和帮助.

**References**:

[1] Dongarra J. Basic linear algebra subprograms technical forum standard. International Journal of High Performance Applications and Supercomputing, 2002,16(1):1−111. [doi: 10.1177/109434200201600101]

[2] Intel MKL homepage. http://software.intel.com/en-us/articles/intel-mkl/

[3] AMD ACML homepage. http://developer.amd.com/libraries/acml/pages/default.aspx

[4] IBM ESSL homepage. http://www-03.ibm.com/systems/software/essl/

[5] GotoBLAS homepage. http://www.tacc.utexas.edu/tacc-projects/gotoblas2

[6] ATLAS homepage. http://math-atlas.sourceforge.net/

[7] Loongson 3A Processor User Manual Part 2. Institute of Computing Technology, the Chinese Academy of Sciences. 2011. http://www.loongson.cn/uploadfile/file/shouce/3A_user_manual_P2_GS464bit_V1.1.pdf (in Chinese)

[8] OpenBLAS homepage. http://xianyi.github.com/OpenBLAS/

[9] Goto K, Van De Geijn R. High-Performance implementation of the level-3 BLAS. ACM Trans. on Mathematical Software, 2008, 35(1):1−18. [doi: 10.1145/1377603.1377607]

[10] Goto K, Van De Geijn R. Anatomy of high-performance matrix multiplication. ACM Trans. on Mathematical Software, 2008,34(3): 1−25. [doi: 10.1145/1356052.1356053]

[11] Yang RQ. Optimization of GotoBLAS math library based on Loongson processor [MS. Thesis]. Beijing: Graduate University, the Chinese Academy of Sciences, 2009 (in Chinese with English abstract).

[12] Su B, Li K, Xu ZG, He SS. Optimization of memory access based on loongson2F. Computer Systems and Applications, 2010, 19(1):171−175 (in Chinese with English abstract).

[13] Gu NJ, Li K, Chen GL, Wu C. Optimization of BLAS based on Loongson 2F architecture. Journal of University of Science and Technology of China, 2008,38(7):854−859 (in Chinese with English abstract).

Li Y, He SS, Li K. Optimization of BLAS Level 2 Based on multi-core Loongson 3A. Computer Systems and Applications, 2011,20(1):163−167 (in Chinese with English abstract).

**附中文参考文献**:

[7] 龙芯 3A 处理器用户手册第 2 部分.中国科学院计算技术研究所.2011.http://www.loongson.cn/uploadfile/file/shouce/3A_user_manual_P2_GS464bit_V1.1.pdf

[11] 杨荣秋.基于龙芯处理器的 GotoBLAS 数学库优化[硕士学位论文].北京:中国科学院研究生院,2009.

[12] 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.

[13] 顾乃杰,李凯,陈国良,吴超.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):854−859.

[14] 李毅,何颂颂,李恺.多核龙芯 3A 上二级 BLAS 库的优化.计算机系统应用,2011,20(1):163−167.

**张先轶**(1983－),男,山东龙口人,博士生,助理研究员,主要研究领域为并行算法与软件,针对 GPU 等异构部件的性能优化.

**王茜**(1987－),女,硕士生,主要研究领域为并行算法与并行软件,自适应性能优化技术.

**张云泉**(1973－),男,博士,研究员,主要研究领域为并行算法与并行软件,海量信息处理,性能评测.