

基于ELPA的并行广义本征值求解器GenELPA

- 沈瑜 中国科学技术大学 超级计算中心 合肥 230026
- 孙广中 中国科学技术大学 计算机科学与技术学院 合肥 230026

摘要:

高效率的大规模并行求解广义本征值方程是在计算电子结构等高性能计算学科中的一个重要课题，也在未来E级计算的发展上有重要的意义。根据马普研究所近几年开发出的一种面向P级应用的本征值求解库ELPA，针对广义本征值问题，我们开发了一款开源并行广义本征值求解器GenELPA，实现了从广义本征值问题到一般本征值问题的转换方法，避免了潜在的问题；然后调用ELPA进行计算，具有良好的计算速度和并行效率；同时采用了类似ScaLAPACK的接口，更加易于使用。在第一性原理软件ABACUS中的实际应用表明具有理想的计算速度提升。

关键词：广义本征值，ELPA，电子结构，ABACUS

1. 前言

在很多需要数值计算的领域，如何高效求解大型矩阵的本征值和本征向量是常见的问题。例如在电子结构理论中，我们常常要把求解的Kohn-Sham方程转换为矩阵形式，然后求出本征值。^[1]很多时候系统需要通过多次自洽迭代达到稳态，而在每次迭代中都需要求解该方程，尤其在几百个原子以上的较大尺度下，该方程的求解成为最耗费时间的步骤，^[2]所以如何求解Kohn-Sham方程成为影响计算速度的关键因素之一。

一般情况下，Kohn-Sham方程是一个广义本征值方程，具有如下形式：

$$Ac=Bc\lambda \quad (1)$$

其中A和B为已知的 $N \times N$ 的矩阵，c是 $N \times N$ 的本征向量矩阵， λ 是以本征值为对角元的矩阵。很多情况下，A和B都是对称矩阵（如果为实数矩阵）或者厄米矩阵（如果为复数矩阵），同时B还是正定且非奇异矩阵。这种情况下，可以把广义本征值方程转化为一般本征值方程，然后进行求解。

本征值的求解是数值计算的一个基本问题，直接进行代数求解所有本征值和本征向量的计算复杂度为 $O(N^3)$ ，在需求解矩阵规模增长时，对计算能力的需要增长地非常快。但如果问题有一定特殊性，例如只需要一小部分本征值和本征向量、或者本征函数有一些约束条件等等，可以采用迭代等方法来降低计算复杂度^[3]。在某些情况下，甚至可以

绕过求解本征值的步骤从而使问题复杂度降低到 $O(N)$ ^[4]。另一方面，也可以采用大规模并行的方法来加速本征值的计算。尤其是近些年超级计算机的飞速发展，不仅顶级超级计算机的规模迅速扩大，而且千核乃至万核级服务器也日趋普及。目前，有很多工作是研究大规模并行本征值求解，例如ScaLAPACK^[5]、PLAPACK^{[6][7]}、eigen_s和eigen_sx^[8]、Elemental^[9]、EleMRRR^[10]等等。同时，基于众核的PLASMA和基于GPU的MAGMA也随着近年来异构计算的发展而开发出来^[11-14]。

2011年德国马克斯普朗克学会的科学家们为了解决在FHI-aims全电子结构软件中的针对上千个原子体系中需要求解的大规模对称和厄米矩阵的本征值问题，发展了面向P级应用的本征值求解库ELPA^[15]。ELPA具有相对传统ScaLAPACK更好的计算速度和并行效率，除了在FHI-aims^[16]里得了应用，也为很多专业软件采用，例如针对纳米尺度材料模拟软件OpenMX（Open source package for Material eXplorer）^[17]、量子化学和固体物理软件CP2K^[18]等等。

在本文中，我们将介绍一种自行开发的基于ELPA的广义本征值求解器GenELPA，该求解器以开源形式公开发布，它实现了转换广义本征值方程为一般本征值方程的方法，然后调用ELPA进行计算。为了方便使用，采用了类似ScaLAPACK的接口和自动选择最佳转化的方法，并提供了转化和求解分步调用的函数。我们将该求解器应用到由中国科学技

术大学量子信息重点实验室开发的第一性原理软件 ABACUS^[19]中, 获得了理想的效果。

2. 原理

求解广义本征值问题的可以先将广义本征值方程转化为一般本征值方程, 然后求解转化后的一般本征值问题。第一种转化方法是将B矩阵做Cholesky分解为上三角矩阵或者下三角矩阵, 这里我们以以上三角矩阵为例, 如下:

$$B=U^T U \quad (2)$$

其中U为上三角矩阵, U^T 是它的转置矩阵。

然后令

$$\tilde{A} = (U^{-1})^T A U^{-1} \quad (3)$$

$$\tilde{c} = U c \quad (4)$$

那么(1)就变为标准本征值问题:

$$\tilde{A} \tilde{c} = \tilde{c} \lambda \quad (5)$$

方程(5)可以通过ELPA来求解, 如果需要本征向量的话, 可以通过下式求得:

$$c = U^{-1} \tilde{c} \quad (6)$$

第二种方法是将B对角化后, 求得 $B^{-1/2}$ 。即先用ELPA求解关于B的本征值方程, 得到本征值和本征向量 $e = \{e_i, i=1, 2, \dots\}$ 和 $q = \{q_{ij}, i, j=1, 2, \dots\}$, 然后有:

$$B_{i,j}^{-1/2} = \sum_k q_{i,k} e_k^{-1/2} q_{k,j} \quad (7)$$

其中 $B^{-1/2}_{ij}$ 为矩阵 $B^{-1/2}$ 的矩阵元, q_{ik} 和 q_{kj} 为本征向量矩阵q的矩阵元, e_k 为第k个本征值。令:

$$\tilde{A} = B^{-1/2} A (B^{-1/2})^T \quad (8)$$

$$\tilde{c} = B^{1/2} c \quad (9)$$

则方程(1)可转变为一般本征值形式:

$$\tilde{A} \tilde{c} = \tilde{c} \lambda \quad (10)$$

同样, 上式可以通过ELPA来求解, 如果需要本征向量的话, 可以通过下式求得:

$$c = B^{-1/2} \tilde{c} \quad (11)$$

3. 功能与使用

GenELPA主要功能是将上述过程自动进行, 用户只需要输入矩阵A和B以及进程的相关信息, 就可以直接调用GenELPA进行计算。用户可自行选择转化方法以及是否计算本征向量。

在实际的电子结构计算中, 有时候在进行迭代计算时, B矩阵保持不变而A矩阵会随着迭代的进行而不断更新, 因此有必要将B矩阵的分解和求解最终结果分开, 以避免重复计算。在GenELPA中, 对这两步计算可以单独调用函数进行计算: 第一步将B矩阵做Cholesky分解为上三角矩阵U然后计算 U^{-1} , 或者将B矩阵对角化后计算 $B^{-1/2}$; 第二步再利用分解后得到的 U^{-1} 或者 $B^{-1/2}$ 和A矩阵计算本征值和本征向量(如果需要的话)。

在第一步中, 由于ELPA提供的Cholesky分解函数在某些较小的系统下可能会不稳定, 因此我们还提供了使用ScaLAPACK中的分解函数作为替代选择。因此加上对矩阵B进行对角化的方法, 总共有三种方法将矩阵B进行分解。在GenELPA中, 使用者可以指定使用某种方法, 也可以由程序自动选择。

GenELPA中的函数采用类似ScaLAPACK的命名规则: 第一个字母如果是p, 表明是个并行的函数; 第二个字母表示处理的数据类型: d为双精度实数, z为双精度复数。由于ELPA本身也有两个求解引擎“ELPA 1”和“ELPA 2”, 因此需要调用ELPA的函数也有两个不同的版本, 分别对应“ELPA 1”和“ELPA 2”, 其函数名分别用1和2结尾区别。

表1是目前GenELPA提供的函数。GenELPA的输入参数类似于ScaLAPACK, 此外还加入了在整个程序中只需要一次初始化的一些参数, 以避免重复计算。我们以pdSolveGenEigen1为例说明输入参数。

pdSolveGenEigen1的声明如下:

```
int pdSolveGenEigen1(int nev, int nFull, int narows, int nacols,
int *desc,
double *a, double *b, double *ev, double *q, double
*work,
int my_mpi_comm, int my_blacs_ctxt, int &method,
bool wantEigenVector, bool wantDebug);
```

表1 GenELPA函数列表

函数名	说明
pdSolveGenEigen1	使用ELPA 1直接求解广义本征值
pdDecomposeRightmatrix1	使用ELPA 1分解方程右侧矩阵
pdCheloskyDecomposeRightMatrix	对方程右侧矩阵进行Cholesky分解
pdDiagonalizeRightMatrix1	使用ELPA 1对方程右侧矩阵进行对角化
pdSolveEigen1	使用ELPA 1求解方程右侧矩阵被分解后的本征值问题
globalIndex	获得矩阵的局部坐标的全局坐标
localIndex	获得矩阵的全局坐标的局部坐标

表2 pdSolveGenEigen1参数列表

参数	意义
nev	需要获得的本征值个数, 按从小到大排序
nFull	输入矩阵的整体大小
narows	当前进程中的局部矩阵的行数
nacols	当前进程中的局部矩阵的列数
desc	当前进程中局部矩阵的描述数组, 遵循ScaLAPACK规则
a	广义本征值方程左侧矩阵
b	广义本征值方程右侧矩阵
ev	本征值数组, 注意不论nev为多少, 该数组大小至少为nFull
q	本征矢量数组, 注意不论nev为多少, 该数组大小至少为nFull*nFull
work	需要存放零时变量的工作数组, 大小至少为nFull*nFull
my_mpi_comm	使用进程组的MPI通信子
my_blacs_ctxt	MPI通信子对应的BLACS CONTEXT
method	求解方法: 0为自动选择; 1为使用ScaLAPACK对右侧矩阵进行Cholesky分解; 2为使用ELPA提供的Cholesky分解函数对右侧矩阵进行分解; 3为使用对角化方法对右侧矩阵进行分解。
wantEigenvector	是否需要获得本征向量
wantDebug	是否需要debug, 该参数将在调用ELPA的时候使用

其中, 各个参数意义详见表2。

使用GenELPA时, 有以下需要说明的地方:

1) 由于ELPA底层为FORTRAN语言编写, 因此二维数组 (a、b、q、work) 中的数据在内存中的存储需要按FORTRAN语言使用的列主序方式存储。至于进程矩阵, 则不限定采用行主序或者列主序。

2) 程序运行中, 数组a、b会被用作中间变量, 因此程序结束后, 变量a、b的值会被改变。如果希望保留输入数组, 请事先将输入数组复制一份。

3) 除非必要, 否则method方法最好选择0作为默认参数。当使用参数0时, 在程序结束后method会返回实际使用的方法序号。

4) 程序目前尚在开发中, 我们会尽快将采用ELPA 2进行计算和处理复数方程的部分补齐, 同时会进一步对性能调试优化。

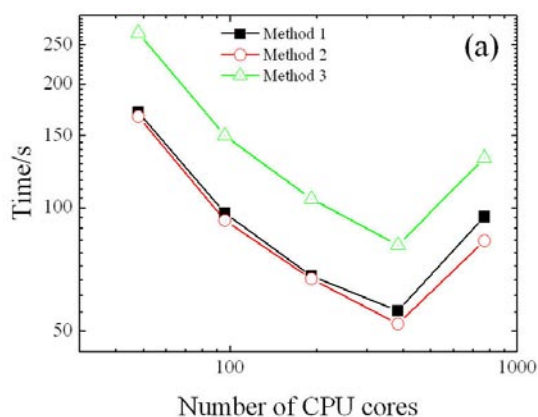
4. 性能测试

我们使用了20000 × 20000大小的矩阵进行了实际测试, 测量了不同方法所需要的矩阵分解时间、本征值计算时间和总计算时间。我们使用的测试平台是中国科学技术大学超级计算中心的TC4600集群^[20], 该集群的每个计算节点有24个CPU核心, 因此我们测试使用的CPU核心数为24的整数倍, 即48、96、192、384和768个。

图1和图2分别是计算全部 (20000) 和一半 (10000) 本征值的时间统计, 横轴为使用的CPU核心数, 纵轴为时间。其中子图 (a) 是三种方法所用整体时间的对比, 黑色方块线为方法1, 红色圆圈线为

方法2, 绿色三角线为方法3。可以看出在使用CPU核心数较少时方法1和方法2的速度几乎一样, 而使用CPU核心数较多时, 方法2要稍快一些。至于方法3, 所需时间一直要比其它两种方法长很多。子图 (b) 到 (d) 分别是方法1到方法3的分步骤时间统计, 其中黑色方块线为右侧矩阵分解时间, 红色圆圈线为计算本征值时间, 绿色三角线为总时间。由于方法3通过对角化右侧矩阵进行分解时需要计算右侧矩阵的本征值, 因此方法3的矩阵分解时间要接近计算本征值的时间。而方法1和方法2的矩阵分解使用Cholesky方法, 所需时间要远远小于计算本征值的时间。

测试结果表明, 首先, 使用Cholesky分解比对角化能够更快的将矩阵B分解; 其次, 在使用CPU核心数较少时, 采用ScaLAPACK和ELPA提供的Cholesky分解方法耗时几乎一样。



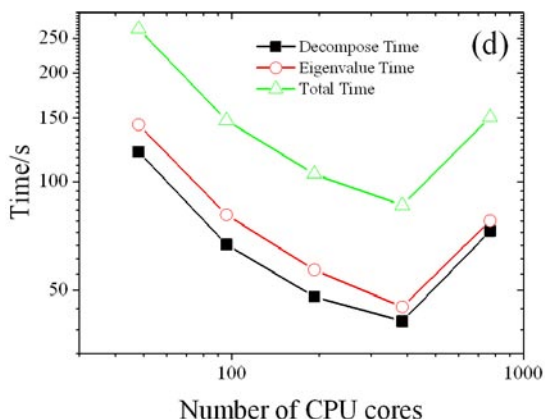
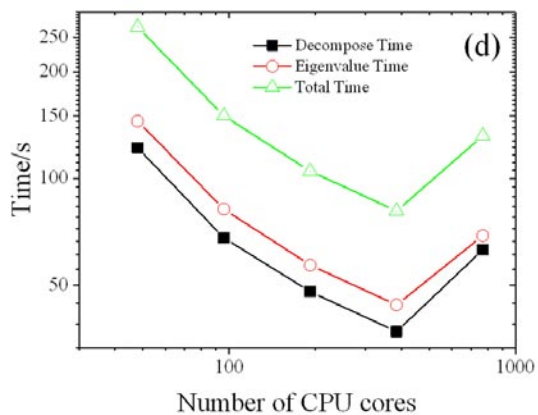
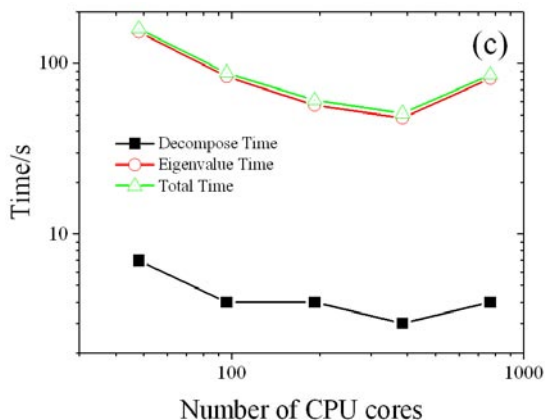
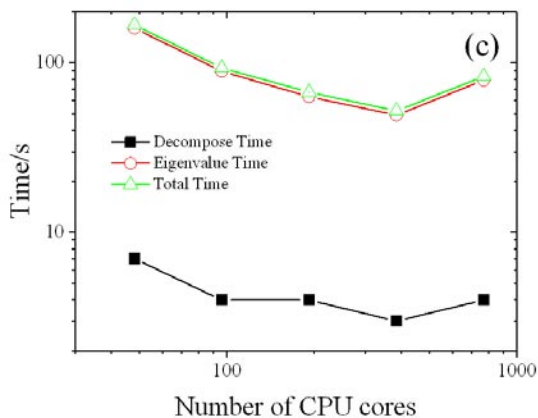
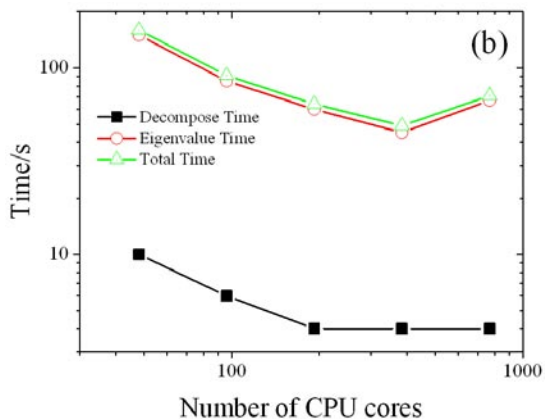
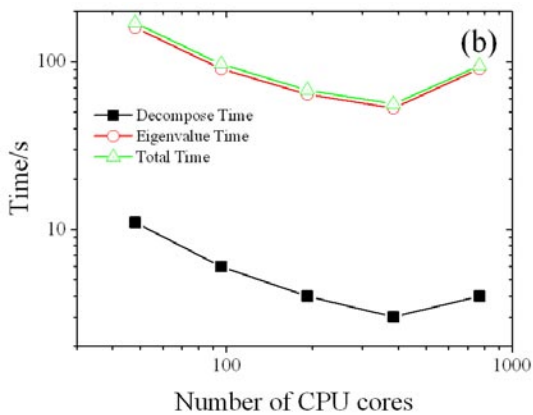


图1 计算所有本征值时的时间

图2 计算一半本征值时的性能对比

需要说明的是，该测试程序还没有得到充分优化，因此在大规模并行效率上还不如ELPA本身的效率，我们将在后续的工作中进一步对其优化。

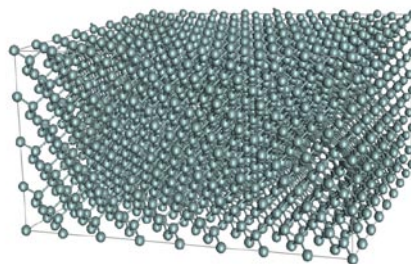
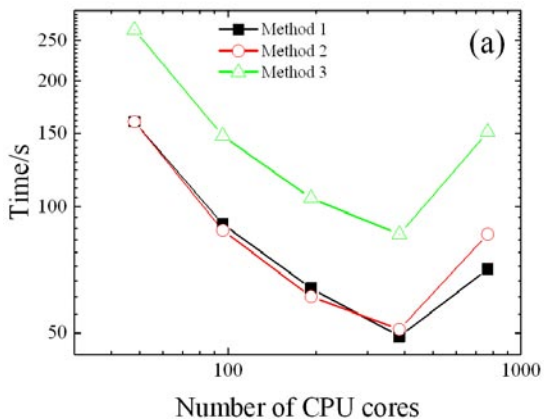


图3 包含2048个硅原子的测试算例

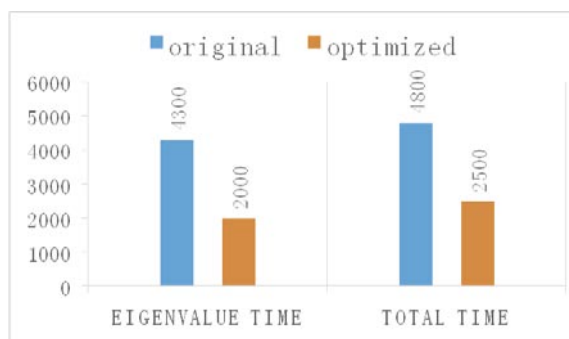


图4 计算时间对比

我们将此求解器应用到第一性原理软件 ABACUS，并使用一个由2048个硅原子（Si）组成的晶体中进行了初步测试（图3），该体系在每一个电子步的迭代中需要计算一次 26000×26000 大小的广义本征值方程，我们同样在TC4600系统中，采用96个CPU核心进行了测试。作为对比，ABACUS原始使用的求解器是由中科院网络信息中心开发的HPSEPS软件包^[21]。测试结果表明，采用GenELPA进行计算后，本征值求解部分计算时间由4300秒减少到2000秒，程

序整体运行时间由4800秒减少到2500秒，程序性能取得了显著地提高(图4)。

5. 总结

我们基于具有良好的计算速度和并行效率的本征值求解器ELPA编写了广义本征值求解器GenELPA，该求解器为用户提供了三种将广义本征值问题化为一般本征值问题的方法，简化了使用方式，避免了在求解某些问题时ELPA中可能存在的问题；实际测试表明该求解器还具有较好的计算速度和并行效率。针对电子结构计算等需要大规模并行求解广义本征值的问题，能够有效的简化难度，提高效率。

6. 致谢

本工作得到了中国科学技术大学中科院量子信息重点实验室的何力新教授、任新国教授和刘晓辉博士在应用集成和实例测试中的帮助。

参考文献：

- [1] Martin R M. Electronic structure: basic theory and practical methods[M]. Cambridge university press, 2004, P135.
- [2] Li P, Liu X, Chen M, et al. Large - scale ab initio simulations based on systematically improvable atomic basis[J]. Computational Materials Science, 2016, 112: 503 - 517.
- [3] Golub G H, Van der Vorst H A. Eigenvalue computation in the 20th century[J]. Journal of Computational and Applied Mathematics, 2000, 123(1): 35 - 65.
- [4] Goedecker S. Linear scaling electronic structure methods[J]. Reviews of Modern Physics, 1999, 71(4): 1085.
- [5] Blackford L S, Choi J, Cleary A, et al. ScaLAPACK users' guide[M]. Siam, 1997.
- [6] Alpatov P, Baker G, Edwards C, et al. PLAPACK: parallel linear algebra package design overview[C]//Proceedings of the 1997 ACM/IEEE conference on Supercomputing. ACM, 1997: 1 - 16.
- [7] Baker G, Gunnels J, Morrow G, et al. PLAPACK: High performance through high - level abstraction[C]//Parallel Processing, 1998. Proceedings. 1998 International Conference on. IEEE, 1998: 414 - 422.
- [8] Imamura T, Yamada S, Machida M. Development of a high performance eigensolver on the petascale next generation supercomputer system[C]//Proceedings of Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2010 (SNA+ MC2010). 2010.
- [9] Poulson J, Marker B, Van de Geijn R A, et al. Elemental: A new framework for distributed memory dense matrix computations[J]. ACM Transactions on Mathematical Software (TOMS), 2013, 39(2): 13.
- [10] Petschow M, Peise E, Bientinesi P. High - performance solvers for dense hermitian eigenproblems[J]. SIAM Journal on Scientific Computing, 2013, 35(1): C1 - C22.
- [11] Haidar A, Solcà R, Gates M, et al. Leading edge hybrid multi - GPU algorithms for generalized eigenproblems in electronic structure calculations[C]//International Supercomputing Conference. Springer Berlin Heidelberg, 2013: 67 - 80.
- [12] Agullo E, Demmel J, Dongarra J, et al. Numerical linear algebra on emerging architectures: The PLASMA and MAGMA projects[C]//Journal of Physics: Conference Series. IOP Publishing, 2009, 180(1): 012037.
- [13] Dongarra J J, Tomov S. Matrix Algebra for GPU and Multicore Architectures (MAGMA) for Large Petascale Systems[R]. University of Tennessee, Knoxville, 2014.
- [14] Buttari A, Dongarra J, Kurzak J, et al. The impact of multicore on math software[C]//International Workshop on Applied

Parallel Computing. Springer Berlin Heidelberg, 2006: 1 - 10.

[15] Marek, Andreas, et al. "The ELPA library: scalable parallel eigenvalue solutions for electronic structure theory and computational science." *Journal of Physics: Condensed Matter* 26.21 (2014): 213201.

[16] Blum V, Gehrke R, Hanke F, et al. Ab initio molecular simulations with numeric atom - centered orbitals[J]. *Computer Physics Communications*, 2009, 180(11): 2175 - 2196.

[17] Ozaki T. Variationally optimized atomic orbitals for large - scale electronic structures[J]. *Physical Review B*, 2003, 67(15): 155108.

[18] Hutter, Jürgen, et al. "CP2K: atomistic simulations of condensed matter systems." *Wiley Interdisciplinary Reviews: Computational Molecular Science* 4.1 (2014): 15 - 25.

[19] Xiao - Hui L, Mo - Han C, Peng - Fei L, et al. Introduction to first-principles simulation package ABACUS based on systematically improvable atomic orbitals[J]. *ACTA PHYSICA SINICA*, 2015, 64(18).

[20] 李会民 曙光TC4600百万亿次超级计算系统使用指南[OL], 2016, <http://sc.ustc.edu.cn/zlsc/tc4600/tc4600-userguide.pdf>

[21] 赵永华, HPSEPS软件包及其在千核上的并行计算性能[J], *科研信息化技术与应用*, 1.4(2010): 8

要闻集锦

IBM芯片技术重大突破 催生亿亿次超级计算机

近日据国外媒体报道，IBM的研究者在使用光脉冲来加速芯片间的数据传输方面取得了突破，该技术可以将超级计算机的性能提升一千多倍。IBM硅光子科学家Will Green称，这项叫做CMOS集成硅光子光学的技术在一块硅片上集成了光电模块，让电信号转化为光脉冲，使芯片能够以更高的速率通讯。

这项技术使超级计算机的计算能力大幅度提升，目前最快的超级计算机速度可达到2千万亿次或每秒2000万亿条指令，光子技术可以将速度提高到1亿亿次每秒，这有助于IBM实现在2020年制造出亿亿次超级计算机的设想，这对系统构建者来说是一项相当有趣的里

程碑。

Green还说，多光子模块可以集成在一块基片或主板上。新式的超级计算机已经开始使用光技术进行芯片间通讯，但更多的是在机架级和单一波长。IBM的技术可以使光通讯同时在多波长上实现。

该技术可以在标准芯片生产线上使用，无需任何特殊的工具，这样能节省成本。现有的演示采用的是一个130纳米CMOS节点，但IBM还在努力争取做到小于100纳米。该技术将替代现有的铜导线进行数据传递，光传输距离更远，更节能。

(魏嘉)