

“神威·国实杯”第七届国产CPU并行应用挑战赛初赛试题

PCG 求解线性方程组 $Ax=b$

程序介绍

PCG(preconditioned conjugate gradient, 预处理共轭梯度算法)是一种利用多次迭代对方程组进行求解的方法。相比于使用直接法求解方程组,其对于存储空间的要求不高且扩展性良好,在涉及方程组求解的科学计算应用中具有一定的优势。大规模方程组求解在高性能计算机上进行计算时,使用直接法往往会因为程序的崩溃而导致之前的计算成为无效计算。再次进行计算时,之前已经计算过的部分要进行重新计算,浪费了大量的时间与计算资源,使用PCG算法能够有效地解决这一问题。

PCG涉及到的应用领域广泛,这些应用涵盖了水工、土建、桥梁、机械、电机、冶金、造船、飞机、导弹、宇航、核能、地震、物探、气象、渗流、水声、力学、物理学等几乎所有的科学研究和工程技术领域,广泛的应用于求解热传导、电磁场、流体力学等连续性问题。在商用软件中也得到了广泛的应用,如ANSYS、EPANET、MODFLOW、MATLAB、FLUENT等。

算法流程

本赛题需要进行并行优化的部分为 PCG 迭代法求解，详细流程如下所示：

下所示：

$$x_0 := 0$$

$$r_0 := b - Ax_0$$

$$z_0 := M^{-1}r_0$$

$$p_0 := z_0$$

$$k := 0$$

repeat

$$a_k := \frac{r_k^T z_k}{p_k^T A p_k}$$

$$x_{k+1} := x_k + a_k p_k$$

$$r_{k+1} := r_k - a_k A p_k$$

if r_{k+1} 足够小（本赛题中设置为 $1e-10$ ），循环终止

$$z_{k+1} := M^{-1}r_{k+1}$$

$$\beta_k := \frac{z_{k+1}^T r_{k+1}}{z_k^T r_k}$$

$$p_{k+1} := z_{k+1} + \beta_k p_k$$

$$k := k + 1$$

end repeat

其中， x_{k+1} 就是得到的解。

接口介绍

本题中需要优化的函数如下图所示：

```
/* PCG
double pcg_solve(const LduMatrix &ldu_matrix, double * source, double *psi, int maxIter, double tolerance, double normfactor);
void ldu_to_csr(const LduMatrix &ldu_matrix, CsrMatrix &csr_matrix);
void csr_spmv(const CsrMatrix &csr_matrix, double *x, double *b);
void csr_precondition_spmv(const CsrMatrix &csr_matrix, double *vec, double *val, double *result);
void v_dot_product(const int nCells, const double *vec1, const double *vec2, double *result);
void v_sub_dot_product(const int nCells, const double *sub, const double *subed, const double *vec, double *result);
void pcg_init_precondition_csr (const CsrMatrix &csr_matrix, Precondition &pre);
void pcg_precondition_csr(const CsrMatrix &csr_matrix, const Precondition &pre, double *rAPtr, double *wAPtr);
double pcg_gsumMag(double *r, int size);
double pcg_gsumProd(double *z, double *r, int size);
void free_pcg(PCG &pcg);
void free_csr_matrix(CsrMatrix &csr_matrix);
void free_precondition(Precondition &pre);
```

- (1)pcg_solve 接口是 PCG 求解的入口，**该接口的参数不可修改。**
- (2)ldu_to_csr 为矩阵数据转换接口，可以将 LDU 格式的数据转换为 CSR 格式。
- (3)csr_spmv 为实现 PCG 求解步骤中 SPMV 算法的接口。
- (4)csr_precondition_spmv, v_dot_product, v_sub_dot_product, pcg_init_precondition_csr 和 pcg_precondition_csr 为 PCG 算法中预条件子部分的实现接口，具体算法流程请参考提供的赛题代码。
- (5)pcg_gsumMag 方法计算归一化残差范数，pcg_gumProd 方法更新搜索方向，计算步长 alpha 的分母。具体请参考提供的赛题代码。
- (6) free_pcg, free_csr_matrix, free_precondition 为算法中的空间释放接口。

程序说明

1. 本题程序是 PCG 求解方法，参赛者参照主核代码，**以计算效率为考核目标，只考察从核并行优化，不涉及 MPI 并行**，请严格按照算法流程优化计算。

例：论文《基于相对边缘化 BA 的 ORBSLAM3 室内定位算法研究》

中，通过在 PCG 算法流程中设定阈值，选择超过阈值的状态量进行更新，减少冗余计算，获取更高求解矩阵效率，是对 PCG 算法本身的改进，改变了算法流程，**属于未按照赛题算法流程优化计算！**

2. 本题数据读取和方程初始化部分已完成。
3. **除 pcg.cpp, slave.c, run.sh 和 makefile 文件以外，其余代码都不允许被修改。**
4. 预处理算法的类型不是本赛题的考察目的，**因此接口介绍（4）中的预处理流程不允许修改。**

例：预处理方法有很多种，如 Jacobi 预处理、不完全 Cholesky 预处理等，换用其他预处理方法改变了预处理流程，**属于未按照赛题预处理流程优化计算！**

5. 本赛题共针对三种不同规模的矩阵进行求解，矩阵规模分别为 44181、58908 和 88362。
6. 本赛题提供一个简单的从核化示例供选手参考。
7. 结果说明

每个时间步求解结束后调用 check_result 函数进行结果校验，如果结果有误，错误的结果会在终端输出并退出运行。

check_result 注意事项：

- 1) **求解迭代次数完全相同。**
- 2) **残差值绝对误差不大于 $1e-11$ 。**

源代码获取：

参赛者登录分配的 CPC 账号，赛题代码路径为：`/home/export/online1/share/cpc2023`，当前路径权限为只读，参赛者自行将代码拷贝到自己的路径即可。

编译：

程序路径下：`make`

```
[cpc2301@sw_hpc_138 test]$ make
sw9g++ -mhost -mieee -mftz -fpermissive -I. -c main.cpp -o main.o
sw9g++ -mhost -mieee -mftz -fpermissive -I. -c pcg.cpp -o pcg.o
sw9gcc -mslave -msimd -mieee -c slave.c -o slave.o
sw9g++ -mhybrid -o pcg_solve main.o pcg.o slave.o -L. -lpcg_solve
```

运行：

1、程序路径下：`sh run.sh`

```
[2023- 5-25] [15:52:38] [INFO] Mesh >> read mesh!
[2023- 5-25] [15:52:39] [INFO] Step >> Main solve start!
[2023- 5-25] [15:52:39] [INFO] Step >> mesh : 1, run time: 0.00
[2023- 5-25] [15:52:40] [INFO] PCG: init residual = 1.816396e+12, final residual = 8.472088e-11, iterations: 141
[2023- 5-25] [15:52:40] [INFO] Step >> run time: 0.00, time cost: 0.7714s
[2023- 5-25] [15:52:40] [INFO] Check >> check result correct!
```

run.sh 作业提交命令的参数中，主核数(-n 1)不允许更改。

2、每个时间步计算完成后，输出本次 `pcg_solve` 迭代次数、运行时间和验证结果。

```
[2023- 5-25] [15:52:40] [INFO] Step >> mesh : 1, run time: 0.01
[2023- 5-25] [15:52:41] [INFO] PCG: init residual = 1.453117e+07, final residual = 7.221247e-11, iterations: 112
[2023- 5-25] [15:52:41] [INFO] Step >> run time: 0.01, time cost: 0.6164s
[2023- 5-25] [15:52:41] [INFO] Check >> check result correct!
```

程序会输出各规模的求解时间，程序整体运行完成后，输出 PCG 算法的总求解计算时间。本赛题采用总求解计算时间（`total time cost`）作为最终的评价标准。

```
[2023- 5-25] [16: 1:55] [INFO] Step >> mesh : 3, run time: 1.99
[2023- 5-25] [16: 1:57] [INFO] PCG: init residual = 1.222063e-05, final residual = 9.292327e-11, iterations: 151
[2023- 5-25] [16: 1:57] [INFO] Step >> run time: 1.99, time cost: 1.7751s
[2023- 5-25] [16: 1:57] [INFO] Check >> check result correct!

[2023- 5-25] [16: 1:57] [INFO] Time: mesh : 3, pcg solve time: 331.0798s
[2023- 5-25] [16: 1:57] [INFO] Time: total time cost: 551.6729s
```

评分规则：

- 1、试题满分为 100 分。
- 2、参赛选手必须保证计算结果的正确性。程序将运行 3 次，统计 total time cost 总求解计算优化时间的最优值。
- 3、优化时间最佳的程序得满分 100 分。
- 4、其余程序得分：以满分程序的优化时间为基准算出其他参赛队伍得分，计算公式如下：

$$\text{得分} = \sqrt{\frac{\text{满分程序求解时间}}{\text{该程序求解时间}}} \times 100$$

- 5、参赛选手若修改源码中已明确不能修改部分或做其他不正当的改动将导致成绩无效。

硬件环境

作业运行占用资源的的核组数为 1（不可修改）。

-----本次比赛唯一解释权归 CPC 组委会所有-----