

ASC20-协方差

对covariance的代码进行优化，目标平台是KNL计算卡。源代码有FORTRAN和C两个版本，任选其一进行优化，优化目标是Covariance.c或Covariance.f90中的协方差计算函数（即计时部分）。

可以在Makefile中选定变编译的版本，下图选中的是C版本。

```
14 Covariance.o: Covariance.f90
15 #   ${FC} ${FFLAGS} -c Covariance.f90
16   ${CC} ${CFLAGS} -c Covariance.c
```

运行：执行exe/benchmark.sh进行算例测试，脚本中共有三个算例，第三个算例要跑的时间相当长，可以进行一定程度的优化之后再取消注释运行。

```
7 echo "exp1:"
8 rm exp.ini
9 echo "4000 8000 10" >> exp.ini
10 ./cov
11 echo "exp2:"
12 rm exp.ini
13 echo "12000 10000 16" >> exp.ini
14 ./cov
15 #echo "exp3:"
16 #rm exp.ini
17 #echo "50000 50000 20" >> exp.ini
18 #./cov
```

执行完毕之后，会有计时（total time）和验证（verification），当verification为correct时说明算例结果正确。

完成要求：

1. 此题作为选拔题中的附加题，为**选做性质**，注意是**选做**，面试成绩主要以必做题为基准，当然如果此题完成质量较高会有加分。
2. （若学有余力）可以两种形式完成：
 1. 提交一份优化思路即可，其中包括对现有性能瓶颈的分析，以及给出你的解决方案。
 2. 也可直接修改代码，使用运行时间来说明你的优化效果。并在报告中说明你具体的优化方法，以及提交代码文件。

TIPS:

1. 默认使用intel编译器编译，所以运行之前需要通过命令 `source /opt/intel/parallel_studio_xe_2018.1.038/psxevars.sh` 来配置Intel编译器环境变量。
2. 需要使用KNL平台进行优化，具体操作请阅读KNL上机指南。
3. Linux基础请自行熟悉。

优化的小提示：

1. 向量化计算
2. 访存优化（需要对体系结构有一定了解）
3. 指令集
4. 矩阵的常用优化手法



COVARIANCE.zip

附件：