
 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页


SWOpenFoam 使用说明

Version 0.3

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页

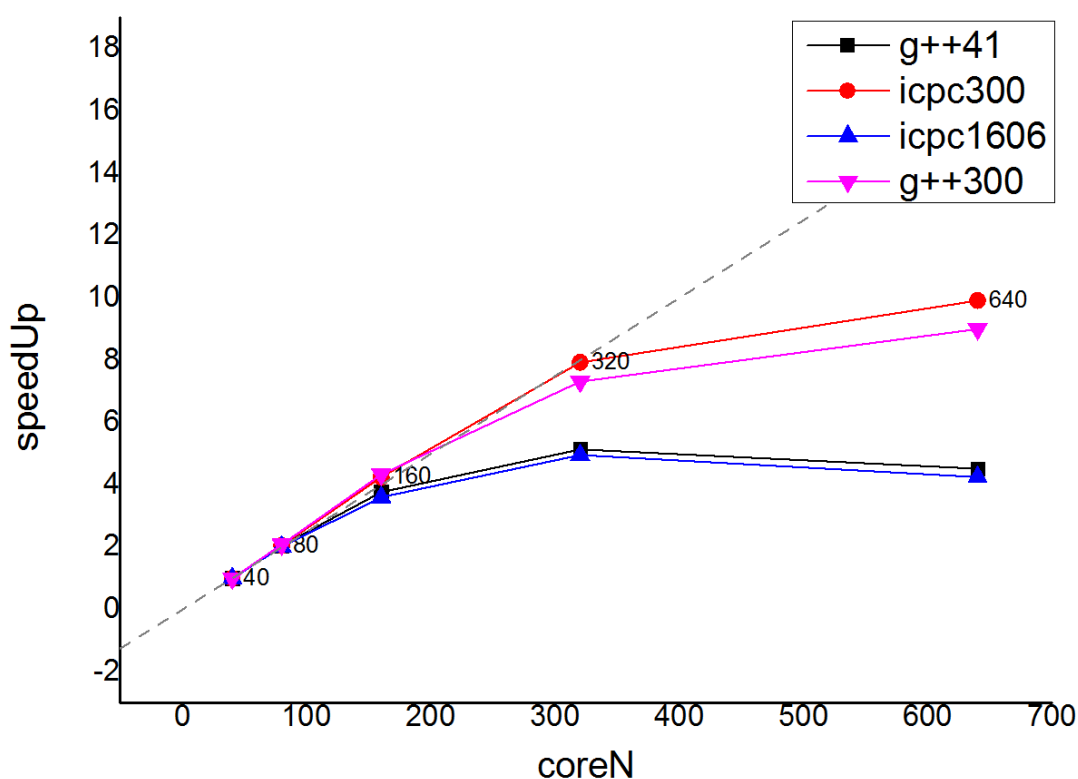
目录

一	概述	3
二	使用方法	4
	1. SWOpenFOAM 的使用方法	4
	2. 几点注意	5
三	OpenFOAM 用户扩展程序编译	6
	1. 商用 x86 平台上扩展程序编译	6
	2. 国产神威平台扩展程序编译	6
四	结语	8

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页

一 概述


目前在神威国产平台和商用 x86 平台上分别部署了 OpenFOAM-2.1.1、OpenFOAM-3.0.0 和 OpenFOAM-v1706 版。OpenFOAM 尚未利用从核，只能在主核上利用 MPI 并行。根据我们的测试（如下图），OpenFOAM-3.0.0 及以上版本中，OpenFOAM-3.0.0 具与其他版本相当的单核性能，但具有最好的 MPI 扩展性。对于使用高于 3.0.0 版本 OpenFOAM 的用户，在功能不受到较大限制的情况下，强烈建议转向 3.0.0 版本以获得更好的性能。



322 万非结构网格 pisoFoam 求解扩展性对比，图例代表所使用的编译器和 OpenFOAM 版本

针对的主从混合并行优化正在展开，很快会有具利用从核加速的 OpenFOAM 版本提供给用户使用。

本说明的第二章节介绍了直接使用 SWOpenFOAM 的相关方法，直接使用 OpenFOAM 的客户可以只阅读该章节；第三章节主要介绍了 OpenFOAM 用户扩展程序在太湖之光上的编译方法。

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页

二 使用方法

1. SWOpenFOAM 的使用方法

在太湖之光上运行 OpenFOAM 算例之前，可以利用一键宏配置相关编译和运行环境，针对不同平台、不同版本的 OpenFOAM，配置命令分别为：


平台	国产平台	x86 平台
OpenFOAM-3.0.0	of300sw	of300x86
OpenFOAM-2.1.1	of211sw	of211x86
OpenFOAM-v1706		ofv1706x86

运行配置命令之后使用 `which` 应该可以找到相关可执行程序。切换到算例所在目录后，请使用 `bsub` 提交相应的队列进行测试和计算。OpenFOAM 自带的 `Allrun` 脚本在集群环境下不能够使用，建议手动提交，或者编写脚本进行批量任务。国产平台 OpenFOAM 使用示例如下图：

```
[swsjtu@psn004 motorBike]$ of211sw
[swsjtu@psn004 motorBike]$ which simpleFoam
/home/export/online1/SHARE/OpenFOAM_SW/OpenFOAM-2.1.1/platforms/linux64swgccDP0pt/bin/simpleFoam
[swsjtu@psn004 motorBike]$ ls
0.org Allclean Allrun constant system
[swsjtu@psn004 motorBike]$ cp -r 0.org/ 0
[swsjtu@psn004 motorBike]$ bsub -I -b -q q_sw_share -share_size 7000 -host_stack 1024 -n 1 blockMesh
Job <6525822> has been submitted to queue <q_sw_share>
some node is sleeping, waiting for dispatch ...
dispatching ...
/*-----*/
|=====|
| \      / | F ield           | OpenFOAM: The Open Source CFD Toolbox
|  \    /  | O peration      | Version: 2.1.1
|   \  /   | A nd            | Web: www.OpenFOAM.org
|    \/    | M anipulation   |
|-----|
/*-----*/
Build : 2.1.1-baacd438ab95
Exec : /home/export/online1/SHARE/OpenFOAM_SW/OpenFOAM-2.1.1/platforms/linux64swgccDP0pt/bin/blockMesh
Date : Mar 16 2017
Time : 09:04:14
Host : "vn002901"
PID : 1212
~~~~~, tmp is (nil)
Case : /home/export/online1/systest/swsjtu/OpenFOAM-2.1.1/tutorials/incompressible/simpleFoam/motorBike
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Allowing user-supplied system call operations

// ***** //
Create time
```

x86 版本的 OpenFOAM 可以在登录节点直接运行，如下图：

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页


```
[swsjtu@psn004 motorBike]$ of300x86
[swsjtu@psn004 motorBike]$ which simpleFoam
/home/export/online1/SHARE/OpenFOAM_x86/OpenFOAM-3.0.0/platforms/linux64IccDPInt640pt/bin/simpleFoam
[swsjtu@psn004 motorBike]$ blockMesh
/*-----*/
|=====|
| \ \ \ / | F i e l d | OpenFOAM: The Open Source CFD Toolbox
| \ \ \ / | O p e r a t i o n | Version: 3.0.0
| \ \ \ / | A n d | Web: www.OpenFOAM.org
| \ \ \ / | M a n i p u l a t i o n |
|=====|
/*-----*/
Build : 3.0.0-6abec57f5449
Exec : blockMesh
Date : Mar 16 2017
Time : 09:37:20
Host : "psn004"
PID : 30206
Case : /home/export/online1/systest/swsjtu/OpenFOAM-2.1.1/tutorials/incompressible/simpleFoam/motorBike
nProcs : 1
sigFpe : Enabling floating point exception trapping (FOAM_SIGFPE).
fileModificationChecking : Monitoring run-time modified files using timeStampMaster
allowSystemOperations : Allowing user-supplied system call operations

// ***** //
Create time
```

但本地运行一般限于单进程，小算例测试，用于检查编译是否正常。一般情况下请使用 `bsub` 命令提交到 x86 队列，例如“`bsub -I -q q_x86_share -n 4 simpleFoam`”，否则会造成登陆节点负载过大，影响其它用户。

2. 几点注意

- 国产神威平台上的 OpenFOAM 不支持 `code stream` 模式，也不支持 `functionObject`，因此建议后处理步骤在求解结束后再单独进行。
- 由于 OpenFOAM 较为庞大，包含的可执行程序 and 库文件众多，我们未经一一测试。如果发现某个国产神威平台上的前后处理程序不能正常运行，可以切换商用 x86 版本进行后处理。如果 solver 发生错误，请先检查算例文件是否正确配置，然后与我们支持人员联系。
- OpenFOAM 各版本的字典输入文件可能格式上有差异，如果发生读入错误，请对应加以修改。
- 当需要使用自编脚本完成批处理时，脚本中的作业提交命令（`bsub ...`）务必使用 `-I` 选项，使作业阻塞，以确保脚本命令按先后顺序执行。如果想后台运行可以使用“`nohup 你的脚本 &`”方式，但是要记住进程号以便出错时终止脚本。
- 对于需要较长参数的可执行程序，如 `transformPoints -scale "(0.001 0.001 0.001)"`，直接用 `bsub` 提交可能出现参数传递错误。最好将该命令包裹在一个脚本中，然后用 `bsub` 命令提交该脚本。

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页

三 OpenFOAM 用户扩展程序编译

1. 商用 x86 平台上扩展程序编译

原理上来讲, x86 是通用平台, 可以编译任何版本的 OpenFOAM 及其扩展程序。但是限于系统和编译器版本的原因, 这里仅仅针对 OpenFOAM-3.0.0 版本介绍扩展程序编译方法。

首先可以利用已经编译好的 OpenFOAM-3.0.0 库。运行 of300x86 配置好编译环境, 配置后编译器为 icpc, 选项为 OpenFOAM 默认编译选项。

使用 OpenFOAM 自有编译脚本进行编译: 仿照 OpenFOAM 文件组织方式, 首先建立一个目录包含生成某个对象 (库文件或者可执行程序) 所需要的源文件, 并在该目录下建立 Make 子目录。在 Make 子目录下创建两个配置文件 files 和 options, 分别指定源文件和创建目标、引用目录和链接库, 示例如下图

```

1 EXE_INC = \
2 -I$(LIB_SRC)/TurbulenceModels/turbulenceModels/lnInclude \
3 -I$(LIB_SRC)/TurbulenceModels/incompressible/lnInclude \
4 -I$(LIB_SRC)/transportModels \
5 -I$(LIB_SRC)/transportModels/incompressible/singlePhaseTransportModel \
6 -I$(LIB_SRC)/finiteVolume/lnInclude \
7 -I$(LIB_SRC)/meshTools/lnInclude \
8 -I$(LIB_SRC)/fvOptions/lnInclude \
9 -I$(LIB_SRC)/sampling/lnInclude
10
11 EXE_LIBS = \
12 -lturbulenceModels \
13 -lincompressibleTurbulenceModels \
14 -lincompressibleTransportModels \
15 -lfiniteVolume \
16 -lmeshTools \
17 -lfvOptions \
18 -lsampling
19
20 files
21
22 1 pisoFoam.C
23 2
24 3 EXE = $(FOAM_APPBIN)/pisoFoam

```


右侧 options 里面 EXE_INC 定义了 include 目录, 一般是相应模块下 lnInclude 目录, 该目录将对应模块下所有源文件建立了符号链接。EXE_LIBS 定义了需要链接的库, 右侧 files 文件中按行列出了需要编译的源文件, 以及编译输出对象 EXE。这些为编译可执行程序需要配置的变量, 编译库文件方法类似, 只是库文件变量前缀为 LIB。另外, 注意将最终输出对象的路径修改为本地路径, 否则会导致错误, 因为 FOAM_APPBIN 等默认路径一般用户没有写权限。完成文件组织后, 在包含 Make 目录和所有源文件的目录下运行 wmake [type] 进行编译, 编译可执行程序 type 为空, 编译动态库 type 为 libso, 编译动态库 type 为 lib。

如果用户要使用 OpenFOAM-2.1.1 作为基本库, 操作与 OpenFOAM-3.0.0 类似。用户也可以自行编译不同版本 OpenFOAM 及扩展程序, 但高版本中往往利用了 C++11 特性, 目前平台上编译器 (ICC-4.0.3 和 GCC-4.4.7) 对 C++11 支持并不好。

2. 国产神威平台扩展程序编译


国产神威平台上的编译与 x86 平台最大的差异在于所有库文件只能编译成静态库。在原本的 x86 体系中, 动态库的依赖关系是分层次的, 也就是说每次生成动态库时只需要链接与之直接有依赖的库, 链接生成可执行程序亦然。单由于 OpenFOAM 使用输入参数来决定 C++ 子类类型, 按照原始配置并使用静态链接, 实际运行时会因为缺少子类定义而导致出错。因此, 在神威平台上编译 OpenFOAM 必须建立依赖库文件补全机制, 并将静态库中所有内容都链入可执行程序中。

编译国产神威版本的 OpenFOAM 与 x86 基本有相似之处。以 OpenFOAM-3.0.0 为例, 先运行 of300sw 进行编译环境的配置, 默认会使用 swg++453 进行编译, 使用

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页

swld453 进行链接。swg++453 和 swld453 并不是默认编译器，目前只用于编译 OpenFOAM。然后，参照 x86 文件组织方法组织好文件，但注意 options 文件要按依赖顺序尽量写全所有依赖的静态库（OpenFOAM 的库和用户扩展的库）。在正式编译之前先把/home/export/online1/SHARE 下的 include、lib、和 bin 拷贝到自己的 \${HOME} 目录，里面存有编译依赖的头文件、库和编译器，然后将 \${HOME}/bin 加入 PATH 环境变量。之后，使用 wmake 进行扩展模块的编译。

使用 wmake 编译过程中 findOption.sh 脚本会自动追踪依赖，查找和搜集所有依赖的 OpenFOAM 库文件，并创建新的 options，将原始的 EXE_LIBS 补全为 EXE_LIBS_NEW。虽然用户不用关心这一过程，但是 findOptions.sh 尚不完善，有可能在 EXE_LIBS_NEW 中出现重复的库，导致链接出错。请用户注意检查。需要注意的是，虽然静态链接只在生成可执行程序时进行，而生成静态库不需要链接，但是静态库的 options 将为 findOption.sh 提供依赖路径，因此也需要尽量按依赖顺序写全其中的 LIB_LIBS 变量。

 国家超级计算无锡中心 National Supercomputer Center in Wuxi	文件编号	Xxx
	版本/更改	A/0
帮助文档（外部共享）	文件总页	页

四 结语

您如果在使用过程中发现该说明没有提及的错误，请联系我们的工作人员。我们一直在对 OpenFOAM 进行持续的优化，力争为用户提供一个高效稳定的版本。

联系人	邮箱
任虎	renhu@mail.nscwx.cn
朱一西	zhuyx@mail.nscwx.cn