

## اجرای برنامه OpenFoam-2.3.0 با استفاده از برنامه مدیر کار

قبلاً درباره ضرورت استفاده از برنامه مدیر کار و نحوه ارسال یک کار (اجرای یک برنامه) به آن توضیح داده شد. در این راهنما نحوه تعامل برنامه مدیر کار را با برنامه OpenFoam نسخه ۲,۳,۰ فرا خواهیم گرفت. در ادامه فرض می‌شود که خواننده با مفاهیم اصلی برنامه مدیریت کار (فرمت اسکریپت و ارسال آن) آشنایی دارد.<sup>۱</sup>

مثال: اجرای یک کد نمونه از برنامه OpenFoam که برای اجرا به چهار هسته برای پردازش موازی نیاز دارد.

قسمت اول: در این قسمت مراحل پیش پردازش<sup>۲</sup> برای اجرای برنامه قبل از ارسال کار واقعی انجام می‌شود.

مرحله (۱) برای اجرای دستورات OpenFoam، لازم است مسیر فایل‌های اجرایی و ماجول‌های مربوطه برای کاربر بارگذاری شوند. به همین خاطر در ترمینال لینوکس، دستور زیر را وارد کنید.<sup>۳</sup>

```
mahmood@cluster:~$ of230
```

مرحله (۲) یک پوشه برای کپی کردن فایل‌های مثال بسازید. توجه کنید که طبق استاندارد OpenFoam، هر کاربر یک پوشه به نام OpenFOAM دارد که در آن یک پوشه دیگر به نام کاربر به همراه نسخه OpenFoam باید وجود داشته باشد. به همین خاطر، دستور زیر را وارد کنید.

```
mahmood@cluster:~$ mkdir -p ~/OpenFOAM/mahmood-2.3.0
```

مرحله (۳) یک مثال نمونه از سری کدهای آموزشی OpenFoam را به پوشه ساخته شده در مرحله قبل منتقل کنید. در این مثال، می‌خواهیم کد propeller را اجرا کنیم. بنابراین دستور زیر را وارد کنید تا مثال propeller از مبدا به پوشه مورد نظر منتقل شود.

```
mahmood@cluster:~$ cp -r $FOAM_TUTORIALS/incompressible/pimpleDyMFoam/propeller/  
~/OpenFOAM/mahmood-2.3.0/
```

<sup>۱</sup> برای یادآوری، به راهنمای استفاده از برنامه MATLAB به آدرس <http://scuhpcc.blog.ir/1395/01/06> مراجعه کنید.

<sup>۲</sup> Preprocessing

<sup>۳</sup> قسمت‌هایی که با رنگ سبز پر نشان داده شده‌اند، اعلان ترمینال هستند و نیازی به تایپ آن نیست. قسمت اول نام کاربری است (در اینجا mahmood) و قسمت بعدی نام سیستمی است که از کاربر دستورات را می‌پذیرد (در اینجا همان سرور که به نام cluster شناخته می‌شود). قسمت بعدی نشان دهنده فولدر فعال است به این معنی که کاربر در کدام مسیر قرار دارد (در اینجا علامت

~ به معنی مسیر خانه (/home/mahmood)

مرحله (۴) به پوشه مورد نظر که حاوی مثال propeller است وارد شوید و دستور Allrun.pre را اجرا کنید. این دستور حدود ۱۰ دقیقه زمان می برد و خروجی به شکل زیر خواهد داشت:

```
mahmood@cluster:propeller$ ./Allrun.pre
Running blockMesh on /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
Running surfaceFeatureExtract on /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
Running snappyHexMesh on /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
Running renumberMesh on /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
Running topoSet on /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
Running createPatch on /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
```

مرحله (۵) دستور decomposePar را وارد کنید تا مسئله به چهار قسمت تقسیم شود. در پایان اجرا مشاهده خواهید کرد که چهار زیر مسئله درست شده است و کلمه End. نشان دهنده اجرای موفق دستور است.

```
mahmood@cluster:propeller$ decomposePar
.
.
.
.

Processor 0: field transfer
Processor 1: field transfer
Processor 2: field transfer
Processor 3: field transfer

End.
```

قسمت دوم: در این قسمت، یک اسکریپت نوشته می شود تا دستور اصلی اجرای برنامه به صورت موازی اجرا شود.

مرحله (۱) یک اسکریپت برای ارسال به برنامه مدیر کار بنویسید و دستور اجرای mpirun را درون آن قرار دهید. به عنوان مثال، اسکریپت زیر را به نام submit.tor ذخیره کنید.

```
#PBS -q default
#PBS -l nodes=1:ppn=4
#PBS -N prop-test
#PBS -o log.txt
#PBS -j oe

cd $PBS_O_WORKDIR

of230
mpirun -np 4 pimpleDyMFoam -parallel | tee log.txt
```

در اسکریپت فوق، دستور #PBS -l nodes=1:ppn=4 به مدیر کار اعلام می کند که از چهار هسته (ppn) یک گره پردازشی (nodes) نیاز است زیرا برنامه قبلاً به چهار زیر مسئله تقسیم شده بود و اکنون نیاز است که چهار زیر مسئله به موازات یک دیگر اجرا شوند. دستور #PBS -o log.txt به مدیر کار اعلام می کند که خروجی برنامه (آن چیزی که در صفحه نمایش نوشته می شود را در فایل log.txt ذخیره کند.

مرحله (۲) از دستور qsub برای ارسال اسکریپت استفاده کنید.

```
mahmood@cluster:propeller$ qsub submit.tor
```

قسمت سوم: در این قسمت باید از روند اجرای برنامه مطلع شد.

مرحله (۱) با اجرای دستور qstat می‌توانید از زمان اجرای برنامه آگاه شوید. همچنین خروجی برنامه، همان طور که در اسکرینیت مرحله نوشته شد، در فایل log.txt نوشته می‌شود که با دستور cat می‌توان محتوای آن را مشاهده کرد.

```

mahmood@cluster:propeller$ qstat
Job id          Name          User          Time Use S Queue
-----
67.cluster      prop-test     mahmood       00:10:44 R default

mahmood@cluster:propeller$ cat log.txt
/*-----*\
|=====|
|  \ \ / /  F i e l d          | OpenFOAM: The Open Source CFD Toolbox |
|  \ \ / /  O p e r a t i o n  | Version: 2.3.0                       |
|  \ \ / /  A n d                | Web: www.OpenFOAM.org                 |
|  \ \ / /  M a n i p u l a t i o n |                                         |
|-----*\
Build : 2.3.0-f5222ca19ce6
Exec : pimpleDyMFOam -parallel
Date : Mar 28 2016
Time : 17:05:45
Host : "compute-0-0.local"
PID : 20960
Case : /home/mahmood/OpenFOAM/mahmood-2.3.0/propeller
nProcs : 4
Slaves :
3
(
"compute-0-0.local.20961"
"compute-0-0.local.20962"
"compute-0-0.local.20963"
)

Pstream initialized with:
floatTransfer : 0
nProcsSimpleSum : 0
.
.
.
.
.
.

```

نکته‌ای که باید به آن توجه کرد این است که هر یک از مراحل اجرای برنامه را می‌توان به صورت اسکرینیت نوشت و به مدیر کار ارسال کرد. به عنوان مثال، زمانی که مسئله بزرگ است و زمان زیادی برای مرحله پیش پردازش آن (برای تقسیم به مسائل کوچکتر) نیاز است، می‌توان دستور مربوطه را در یک اسکرینیت به صورت جداگانه نوشته و آن را به مدیر کار ارسال کرد. در نتیجه، در صورت قطع ارتباط کاربر با سرور، برنامه بدون مشکل اجر خواهد شد.