



Fluent Kullanımı

Fluent: Hesaplamalı Akışkanlar Mekaniği (CFD) alanında yaygın bir şekilde kullanılan ticari bir paket programdır. Kullanıldığı uygulama alanları olarak:

Uçak kanadı etrafında akış
Yanma
Serbest yüzey hidrodinamiği
Biyolojik akış
Yarıiletken üretimi
Temiz oda tasarımı
Atık su işleme
Motor tasarımı
Aeroakustik
Çok fazlı akışlar

şeklinde kısaca verilebilir.

Sistemimizde **64 bit Fluent 6.3.26** sürümü kullanılmaktadır. Toplam olarak Bilişim Enstitüsü'ne ait 10 adet tekil kullanıcı için 70 adet çok işlemci kullanım lisansı, İTÜ genel kullanımına açık 60 tekil kullanıcı için (60 adet çok işlemci kullanım) lisansı vardır.

Fluent kullanımı için yapılması gerekenler aşağıdaki şekilde adım adım açıklanmaktadır.

1. Fluent yazılımının düzgün bir şekilde çalışması için aşağıda belirtilen Linux çevre değişkenleri tanımlanmalıdır. Bu değişkenler **.tcshrc**, **.profile** veya **.bashrc** dosyalarının içine eklenmelidir.

c türevi shell (kabuk) için;

```
setenv LM_LICENSE_FILE 7241@160.75.2.91:${LM_LICENSE_FILE}
setenv FLUENT_ARCH lnamd64
setenv PATH /SFS/progs/Fluent.Inc/bin:${PATH}
```

*bash türevi shell (kabuk) için; *

```
export LM_LICENSE_FILE=7241@160.75.2.91:${LM_LICENSE_FILE}
export FLUENT_ARCH=lnamd64
export PATH=/SFS/progs/Fluent.Inc/bin:${PATH}
```



2.Çalıştırılacak Fluent işi için **girdi dosyaları (.cas, .dat, udf dosyaları vb.)** ve **journal file (.jou dosyası)** yaratılmalıdır. Bu dosyalar fluent in kullanıcı arayüzündende (GUI, Graphical User Interface) oluşturulabilir. Bunun için **cn02** veya **anadolu** (Linux Login Node) makinasına bağlanıp X-Window görüntüsünü kendi bilgisayarınıza yönlendirmeniz gerekmektedir. Bu yönlendirme için Windows kurulu bir makinada (Exceed vb.) bir programın kurulu olması gerekir. (Lütfen cn02 veya anadolu [login node] makinasında iş çalıştırmamaya özen gösteriniz!!! Bu makinalarda çalıştığı tespit edilen işler sistem yöneticileri tarafından durdurulacaktır.)

Örnek Fluent test işi ile ilgili dosyaları kopyalayıp sonraki adımlardaki değişiklikleri yapınız !!!

LSF İŞ TESLİM BETİÇİ: fluent.lsf

```
#!/usr/bin/ksh
#BSUB -a fluent          # burayi degistirmeyin !!!
#BSUB -J FLUENT         # iş ismi
#BSUB -o %J.out        # LSF cikti dosyasi ismi. %J sistemdeki iş
numarasını belirtir.
#BSUB -e %J.err        # LSF hata dosyasi ismi
#BSUB -n 12            # islemci sayisi, fluent -t parametresi ile ayni olmalı
#BSUB -q trccsq       # kuyruk ismi

# -----
# erase
rm -rf host.file

# get empty node names
np=`echo $LSB_HOSTS`

# generate new machine file
for i in $np
do
    echo $i >> host.file
done
# -----

fluent -lsf -pnmpi -cnf=./host.file 2ddp -i fluent.jou -t12
-p -g> fluent.out
```

ÖRNEK FLUENT JOURNAL DOSYASI: fluent.jou

```
file read-cas-data cavity
solve dual-time-iterate
10000
40
file write-case-data cavity_yeni
yes
yes
exit
```

3-Fluent journal ve veri dosyaları Fluent yazılımının arayüzünden veya bir text editörü kullanılarak oluşturulduktan sonra, fluent.lsf (LSF Betik dosyası) dosyasının oluşturulan Fluent dosyaları ile aynı dizine kopyalanıp üzerinde aşağıda belirtilecek değişikliklerin yapılması gereklidir.

3.1. #BSUB -J FLUENT satırı iş LSF üzerinde çalıştırılırken görüntülenecek tanımlama satırıdır. Bu satırdaki FLUENT tanımlayıcısı yerine işinizi tanımlayan herhangi bir kelime yazabilirsiniz.

3.2. #BSUB -n 12 satırı işinizi çalıştırmak istediğiniz işlemci sayısını belirtmektedir. Eğer Fluent' i çok işlemcili bir şekilde çalıştırmak istiyorsanız burada 12 olarak belirtilmiş rakamsal kısma kullanılmak istenen toplam işlemci sayısı yazılmalıdır. Seri olarak yani tek bir işlemcide çalıştırılacak işler için bu satırın tanımlanmasına gerek yoktur. (**DIKKAT >> Bu rakam script dosyasının en son satırında bulunan fluent komutundaki -t12 flag ı ile belirtilmiş işlemci sayısı ile aynı olmalıdır.Seri işler için -t flag ı tanımlanmaz.**)

3.3. #BSUB -q trccsq satırı işin çalıştırılacağı LSF kuyruğunu belirtmek için kullanılır. Her kullanıcı ancak kendi kullanıcılarının tanımlı olduğu kuyrukları kullanabilir.Bu kuyruklar öncelik sırasına göre tanımlanmış olup kullanıcının tanımlı olduğu kuyrukları öğrenmesi için aşağıdaki komutu çalıştırmalıdır.

```
bqueues -u kullanıcı_ismi
```

gelen listeden PRIO sütündeki en yüksek öncelikli kuyruğun seçilmesi sisteme verilen işin önceliğini artırır.

3.4. #BSUB -R "select[(ncpus == 4)]" satırı opsiyonel olarak kullanılabilir veya script dosyasından çıkarılabilir. Bu satır için 4 işlemciye sahip bir makinada çalıştırılmak istendiğini göstermektedir. Ancak bu durumda 2 işlemcili makinalar boş durumda olsa bile 4 işlemcili makinalarda iş var ise LSF gönderilen işi 4 işlemcili bir makinada iş bitene kadar



PENDING (askıya alma) durumunda bekletir ve RUN konumuna geçirmez. Bu satırın çok özel durumlar haricinde (performans testi vb.) kullanılması tavsiye edilmez.

3.5. Son olarak LSF betiğinin son satırında bulunan fluent komutunda "**2ddp**", "**-pnmpi**" (seri işler için kaldırılabilir) ve "**-i**" ile belirtilen journal dosyası ismi kısımları kullanıcı tarafından tanımlanmış işin gereklerine göre değiştirilebilir. NOT: fluent komutu HP Linux makinada /SFS/progs/Fluent.lnc/bin dizini altında bulunmaktadır.

3.6. **#BSUB -o %J.out, LSF_JOB_ID.out, #BSUB -e %J.err, LSF_JOB_ID.err** dosyalarını aynı dizin içinde oluşturur. Bu dosyalar içerisinde LSF ile ilgili bilgiler ve hata mesajları bulunur. "**fluent.out**" dosyasında ise fluent ekran çıktıları bulunur.

3.7. Yukarıda belirtilen satırların dışında hiçbir satırı değiştirmeyiniz !!!Son adımda aşağıdaki komot kullanılarak iş sisteme verilir.

```
bsub < fluent.lsf
```