



MPI Nedir?

MPI başta dağıtık bellekli sistemler olmak üzere paralel algoritma yazmaya yarayan bir kütüphane standardı'dır. Bir programlama dili değildir. MPI kütüphanelerinin C, C++ ve Fortran dili aileleri için sürümleri bulunmaktadır. MPI, "mesaj geçme" mantığına dayalıdır. Değişik işlemciler, birbirleriyle iletiler sayesinde haberleşirler. Sistemimizde Intel MPI Library 2.0 ve mvapich-0.9.8 /mvapich2-0.9.5 yüklüdür.

MVAPICH/MVAPICH2 Nedir?

MVAPICH , MPI1'in infiniband üzerinden MPICH1 tabanlı ,yüksek performanslı bir uygulamasıdır. MVAPICH2 ise MPICH2 tabanlı ,yüksek performanslı MPI2 uygulamasıdır. Verbs Level Interface (VAPI) üzerinden yüksek performans ve ölçeklenebilir desteği bu paketlerle sağlanır. Bu paketler uDAPL, OpenIB/Gen2 ve TCP/IP standardı gibi iletim arayüzleri tarafından desteklenir.

MPI Nasıl Derlenir?

Öncelikle bazı çevre değişkenlerinin ayarlanması gerekiyor. Bash için olan ayarlar için **.bashrc** yada **.profile** dosyamıza aşağıdaki satırları ekleyelim:

```
# Intel Compiler
source /SFS/progs/intel/cce/9.1.038/bin/iccvars.sh
source /SFS/progs/intel/fce/9.1.032/bin/ifortvars.sh
source /SFS/progs/intel/ict/3.0/mpi/3.0/bin64/mpivars.sh)
```

tcsh için **.tcshrc** dosyamıza aşağıdaki satırları ekleyelim ;

```
source /SFS/progs/intel/cce/9.1.038/bin/iccvars.csh
source /SFS/progs/intel/fce/9.1.032/bin/ifortvars.csh
source /SFS/progs/intel/ict/3.0/mpi/3.0/bin64/mpivars.csh
```



Böylece intel derleyicilerini kullanabilir duruma geldik. Artık programımızı derleyelim:

```
$ mpicc -o mpihello.x mpihello.c
$ mpifort -o mpihello.x mpihello.f
```

MPI Programlarının Teslimi

Programımız da hazır. Sisteme işi verebiliriz:

```
$ bsub -q hpccq -n 6 -a intelmpi -o cikti%J.txt mpirun.lsf
/SFS/users/mercan/mpihello/mpihello.x
Job <9430> is submitted to queue <hpccq>.
```

burada "-n 6" programın kaç cpu kullanacağıdır. Fazladan yazdığımız "-a intelmpi" ile bsub'a hangi mpi kütüphanesini kullanacağımızı söylüyoruz. Ayrıca programı mpirun.lsf ile çalıştırdığımızı dikkat edin. Gelin bir de bjobs çıktısına bakalım:

```
$
          bjobs
JOBID   USER   STAT  QUEUE   FROM_HOST   EXEC_HOST   JOB_NAME
SUBMIT_TIME
9430    mercan  RUN   hpccq   cn02        2*cn20      *llo.HP64  Oct 16 12:57
                                           2*cn21
                                           2*cn22
```

burada görüldüğü gibi işimiz 3 farklı makinede 2'şer süreç (process) olarak çalışmış. Toplamda istediğimiz gibi 6 cpu ediyor. LSF her makinede kaç cpu olduğunu bildiği için, 2 cpu'lu makinelere, her makineye 2 süreç gelecek şekilde işi boş makinelere dağıtmış. LSF işimizi en hızlı



çalıştıracağını düşündüğü makinelere otomatik olarak gönderir. Ancak istersek bu aşamada ek şartlar sunarak işimizi belli özelliklere sahip makinelere gönderebiliriz.

MPI+OpenMP (Mixed) Programlarının Derlenmesi ve Teslimi

Mixed programlama düğüm içinde ortak paylaşımlı hafızanın dolayısıyla OpenMP'nin özelliklerini, düğümler arasında da MPI'ın özelliklerini kullanarak yapılan programlama çeşitidir.

MPI+OpenMP kodu aşağıdaki şekilde derlenebilir.:

```
[admin.gurel@lnode2 ~]$ mpiicc -o mixed -openmp mixedHello.c
mixedHello.c(32) : (col. 1) remark: OpenMP DEFINED REGION WAS PARALLELIZED.
```

Kodları çalıştırmak için,

```
[admin.gurel@lnode2 ~]$ bsub -a intelmpi -q usersq -o out.%J -e
err.%J mpirun.lsf ./mixed
```

Job <6243> is submitted to queue <usersq>.

```
[admin.gurel@lnode2 ~]$ bjobs
```

| JOBID | USER | STAT | QUEUE | FROM_HOST | EXEC_HOST | JOB_NAME |
|-------|------|------|-------|-----------|-----------|----------|
|-------|------|------|-------|-----------|-----------|----------|

| | | | | | | |
|------|---------|------|--------|----------|------------|--------------|
| 6243 | admin.g | PEND | usersq | anadolu2 | *f ./mixed | Mar 15 14:45 |
|------|---------|------|--------|----------|------------|--------------|

```
[admin.gurel@lnode2 ~]$ bjobs
```

| JOBID | USER | STAT | QUEUE | FROM_HOST | EXEC_HOST | JOB_NAME |
|-------|------|------|-------|-----------|-----------|----------|
|-------|------|------|-------|-----------|-----------|----------|

| | | | | | | | |
|------|---------|-----|--------|----------|------|------------|--------------|
| 6243 | admin.g | RUN | usersq | anadolu2 | d023 | *f ./mixed | Mar 15 14:45 |
|------|---------|-----|--------|----------|------|------------|--------------|

The output (if any) follows:

```
INFO: 10.128.1.23: Number of CPUs: 4
10.128.1.23: Hello world from process 0 of 1 thread number = 0
10.128.1.23: Hello world from process 0 of 1 thread number = 1
```



```
10.128.1.23: Hello world from process 0 of 1 thread number = 2
10.128.1.23: Hello world from process 0 of 1 thread number = 3
Job /SFS/LSF_HPC/6.2/linux2.6-glibc2.3-x86_64/bin/intelmpi_wrapper ./mixed
```

| TID | HOST_NAME | COMMAND_LINE | STATUS | TERMINATION_TIME |
|-------|-----------|--------------|--------|---------------------|
| 00000 | d023 | ./mixed | Done | 03/15/2007 14:45:26 |

Not: mixed kod intelmpi 3.0 ile çalıştırırken hata alınabilir onun için aşağıdaki satırı source edip
source /SFS/progs/intel/ict/2.0/mpi/bin64/mpivars.sh intelmpi2.0 ile çalıştırılmalıdır.

ÖNEMLİ NOT: Infiniband networkunu kullanacak uygulamalarınızda Intel MPI yada MVAPICH programlarını kullanınız. Diğer sürümler sistemde mevcut değildir ve kullanılmamalıdır. Aksi takdirde önemli performans düşüşleri yaşanacaktır.