

**EURO**

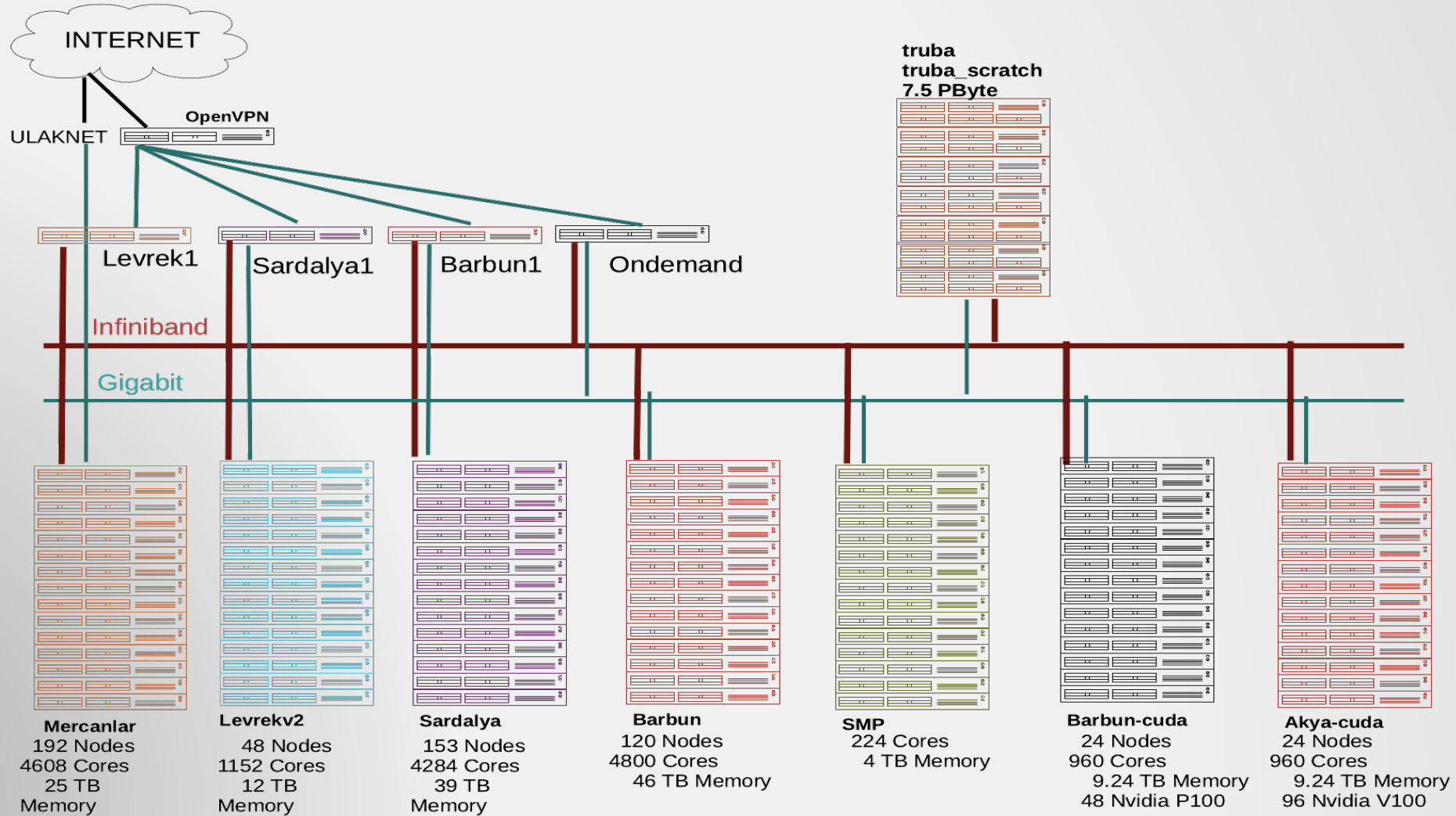
Sefa Arslan, TRUBA, 09.01.2021

# Kimlerin Kullanımına Açık?



- Türkiye'deki tüm yerli ve yabancı doktora ve üzeri tüm akademisyenler
- ARDEB ve BAB kapsamındaki bursiyerler
- TÜBİTAK enstitüleri, kamu – özel projeler

# Sunucu Aileleri ve Kuyruklar



# CPU Kaynakları

Sunucu Ailesi	Yıl	#Node	Mimari	İşlemci	Performans	Bellek
Mercan	2011	192	12 çekirdek x 2 CPU	Opteron 6176 2.30GHz	220.8	128 GB
Levrek	2013	128	8 çekirdek x 2 CPU	Xeon E5-2690 2.90GHz	371.2	256 GB
Levrekv2	2014	48	12 çekirdek x 2 CPU	Xeon E5-2680 v3 2.50GHz	960	256 GB
Orkinos	2016	1	14 çekirdek x 16 CPU	Xeon E7-4850 v3 2.20GHz	7.9Tflops	4TB
Sardalya	2017	128	14 çekirdek x 2 CPU	Xeon E5-2690 v3 2.60GHz	1164Gflops	256 GB
Barbun	2018	120	20 çekirdek x 2 CPU	Xeon Scalable 6148 2.40GHz	2048Gflops	384 GB

# GPU Kaynakları

Sunucu Ailesi	Yıl	#Node	Mimari	İşlemci	Performans	Bellek	#GPU
Barbun-cuda	2018	24	20 çekirdek x 2 CPU +2 x Nvidia P100 GPU	Xeon Scalable 6148 2.40GHz	2048Gflops +9400Gflops	384 GB + 2x16 GB HBM2	2
Akya-cuda	2018	24	20 çekirdek x 2 CPU +4 x Nvidia V100 GPU	Xeon Scalable 6148 2.40GHz	2048Gflops + 4x7800Gflops	384 GB + 4x16 GB HBM	4

# Hesaplama Kuyrukları

Kuyruk	#Node	Sunucular	Zaman Limiti	MemPerCPU	Açıklama
Single	5	Levrekv2	15 Gün	9500MB	1 çekirdekli işler
Short	24	Levrekv2,Sardalya,Barbun	4 Saat	8000MB	
Mid1	271	Levrekv2,Sardalya,Barbun	4 Gün	8000MB	
Mid2	271	Levrekv2,Sardalya,Barbun	8 Gün	8000MB	
Long	271	Levrekv2,Sardalya,Barbun	8 gün	8000MB	
Smp	1	Orkinos	15 Gün	17000MB	
Mercan	128	Mercan	15 Gün	4200MB	
Levrekv2	20	Levrekv2	15 Gün	9500MB	
Sardalya	116	Sardalya	15 Gün	8000MB	
Barbun	120	Barbun	15 Gün	8500MB	
Barbun-cuda	24	Barbun-cuda	15 Gün	8500MB	Yalnızca CUDA
Akya-cuda	24	Akya-cuda	15 Gün	8500MB	Yalnızca CUDA
Debug	441	Tüm sunuculr (orkinos hariç)	15 Dakika	8000MB	Test işleri
Interactive	4	Levrekv2	15 Gün	8000MB	İnteraktif işler

# Depolama Kaynakları

Depolama	Boyut	Amaç	Tür
/truba	7.5 Pbyte	Ev ve uygulama dizinleri	Lustre
/truba_scratch		İş çalıştırma	Lustre
/tmp	450GB-1.5Tbyte	İş çalıştırma	Yerel

# Kotalar

CPU, CPU-Hour, Disk Quota



- Her bir araştırmacı için
  - 128 çekirdek (aynı anda en fazla)
  - 1 TB disk kotası (Home+Scratch, grup tabanlı kota)
  - 920 bin saat + 300 bin saat arttırım
- Projeler için
  - Her projenin ihtiyacına göre belirlenir.



- **Normal Kullanıcılar**
  - Ülkedeki tüm yerli ve yabancı araştırmacılar
  - Ücretsiz kullanım
- **Sözleşmeli Hesaplar**
  - Üniversitelerden araştırma grupları
  - ARDEB yada BAB projeleri
  - Ücretli ve öncelikli kullanım
  - Proje ihtiyacına göre göre belirlenen kaynak miktarı
- **Kamu kurumları veya üniversiteler**

# Ön Gereksinimler

Temel Linux Bilgisi



- CentOS-7.3 (RedHat Enterprise 7 ) İşletim sistemi
- Linux temel yapısı
- Dosya, izin yapısı
- Dosya ve izinlerde kullanıcı hakları
- Temel komutlar ve uygulamalar
- Cp, mv, mkdir, cd , find, vi, nano, wget vs..
- SSH, scp, sftp, X-Forwarding

# Ön Gereksinimler

Uygulama Bilgisi



- Neyi hesaplar, nasıl hesaplar
- Nasıl çalışır (seri, MP, MPI)
- Diğer uygulama ve kütüphanelere bağımlılık
- Çalışır hale getirilmesi (kurulum/derleme)
- Parametreler, input output yapısı,
- İhtiyaç duyulan kaynak miktarı (çekirdek, bellek, süre)

# Ön Gereksinimler

Altyapı Bilgisi



- Altyapının çalışma biçimi
- Mevcut kuyruk ve donanımların özellikleri
- Nasıl bağlantı yapılacağı
- Kullanım kuralları vs..

# Kullanıcı Arayüzleri ve Bağlantı

Doğrudan Bağlantı



Levrek1 – 193.140.99.24

\$ ssh -l kullanıcı\_adi levrek1.ulakbim.gov.tr

# Kullanıcı Arayüzleri ve Bağlantı

OpenVPN Bağlantısı



- Aynı anda aynı kullanıcı adı ile tek bir OpenVPN bağlantısı
- TRUBA kullanıcı adı ve parolası ile bağlantı
- SSH, SCP , X

Levrek1:

```
$ ssh -l kullanıcı_adi 172.16.7.1
```

Sardalya1:

```
$ ssh -l kullanıcı_adi 172.16.8.1
```

Barbun1 :

```
$ ssh -l kullanıcı_adi 172.16.10.1
```

# Kullanıcı Arayüzleri ve Bağlantı

Web bağlantısı OpenOnDemand



- <https://ondemand.truba.gov.tr> <https://172.16.0.200>
  - SSH terminali (levrek1'e)
  - Dosya gezgini
  - Dosya editörü
  - Dosya upload/download
  - Linux desktop (XFCE4)
  - İnteraktif uygulamalar (sadece interactive kuyruğu)
  - Görsel uygulamalar
  - Jupyter
  - Web üzerinden iş gönderme ve takip
- Ayrıntılı tanıtım ve kullanım bilgisi [wiki'de](#)

# Kullanıcı Arayüzleri ve Bağlantı

Kullanıcı Arayüzlerinin Kullanımı



- Dosya kopyalama/taşıma
- Dosya düzenleme
- Kuyruklara iş gönderme
- Gönderilecek işlerin kısa süreli testi
- Uygulama/kütüphane derleme
- Dosya Aktarımı

Kullanıcı arayüzlerinin amacı dışında yada yanlış kullanılmasından dolayı, sunuculara bağlı kullanıcılar rutin işlerini yaparken sıkıntı yaşayabilirler. Bu nedenle ilgili kullanıcının hesabı askıya alınabilir.



# Kuyrukta İş Çalıştırma

Klasik Yöntem: Sbatch



- Betik dosyaları (örnek dosyalar: /truba/sw/scripts)
- Komut satırından başlatılan uygulamalar.
- Etkileşimsiz uygulamalar
- Uygulama parametreleri: komut satırı yada input dosyaları
- Komut satırı yada dosya içi SLURM parametreleri
- --workdir tanımı, bağlı yollar

```
sbatch -n 40 -N 1 -gres=gpu:2 -p barbun-cuda test_job.slurm
```

# Kuyrukta İş Çalıştırma

İnteraktif işler



- Kümeler yoğun biçimde kullanıldığı için, bu şekilde kuyruğa gönderilen işler yine de kuyrukta bekleyeceklerdir.
- İşin ne zaman başlayacağı belli olmadığı yada uzun süreler ekran başında beklenmesi gerekebileceği için interaktif iş çalıştırma TRUBA ve benzeri kümeler için yeterince uygun değildir.
- **İnteraktif işlerin çalışmasını kolaylaştırmak için interactive kuyruğuna eklemeler ve düzenlemeler yapılması planlanmaktadır.**

# Kuyrukta İş Çalıştırma

İnteraktif işler



- ***Srun***

- Herhangi bir komutu hesaplama sunucusu üzerinde çalıştırmak için kullanılır
- İşler interaktif olacak çalıştırılabilir.

```
srun -n 40 -N 1 -gres=gpu:2 -p barbun-cuda hello_world_app
```

- ***Salloc***

- Hesaplama sunucusu üzerinde komut satırı almak için kullanılır.
- Komut satırı alındıktan sonra komutlar srun ile çalıştırılır.

- ***Sattach***

- Sbatch ile kuyruğa gönderilmiş ve çalıştırılmış bir işin, çalışmakta olan iş adımına (betik içinde srun ile çalıştırılmış adım) terminal bağlantısı yapmak için kullanılır.

- ***OpenOnDemand***

- Interactive kuyruğuna görsel ve interaktif uygulama göndermek için kullanılır

- Bellek:
  - Kuyruğa göre değişen çekirdek başına düşen bellek miktarı
  - Çekirdek sayısı ile orantılı bellek miktarı (Otomatik)
- Zaman
  - İş süresi kuyruk süresini geçemez
  - Daha uzun işler sıraya alınmaz.
  - Zamanında bitmeyen iş otomatik olarak sonlandırılır.
  - **Kısa kuyrukların iş önceliği daha yüksektir:**  
Ör: mid1 kuyruğu üzerinden barbunlara (feature) gönderilen işlerin önceliği doğrudan barbun kuyruğuna gönderilen işlerinkinden daha yüksektir.

# SLURM Dosya Yapısı

## Kaynak Tanımları

```
#!/bin/bash
#SBATCH -p mercan
#SBATCH -A sefa
#SBATCH -J gaussian09_test
#SBATCH -N 1 # hesabin dagitilacagi node adedi
#SBATCH -n 4 # is icin toplamda kullanılacak cekirdek adeti
#SBATCH --time=2:00:00

export g09root=$HOME
export GAUSS_SCRDIR=/tmp/$SLURM_JOB_ID
mkdir -p $GAUSS_SCRDIR
. $g09root/g09/bsd/g09.profile

echo "SLURM_NODELIST $SLURM_NODELIST"
$g09root/g09/g09 < gaussian_egitim.com

rm -rf $GAUSS_SCRDIR
exit
```

# SLURM Dosya Yapısı

## Kaynak Tanımları



- **-n, --ntasks** : çekirdek sayısı değil görev sayısıdır.. Her bir görev için birden fazla çekirdek kullanılabilir(yada tersi). Betik içerisinde **srun** komutu verildiğinde komut -n sayıda çalıştırılır. Çekirdek sayısı **-c** ile özel olarak belirtilmezse **-n** çekirdek sayısı olarak kullanılabilir.
- **-N, --node**: çekirdeklerin toplamda kaç farklı node tarafından sağlanacağı Talep edilen çekirdeklerin nodelar tarafından eşit sayıda sağlanması için -N ile birlikte **-tasks-per-node** kullanılmalıdır.
- **-c, --cpus**: her bir görev için kaç çekirdek kullanılacağı. Varsayılan olarak 1
- İş için talep edilen toplam çekirdek sayısı  $n \times c$  olacaktır.

# SLURM Dosya Yapısı

## Kaynak Tanımları



- **-A, --account** :işin hangi kuyruk hesabı üzerinden çalıştırılacağıdır. Örneğin kişisel hesap yada proje hesabı. Bu parametre kullanılmazsa, işi gönderen Linux kullanıcısının kuyruk hesabına gönderilir.
- **-t, --time**
- **-C , --constraint**: Aynı kuyruk içerisinde birden fazla sunucu türü bulunabilir. İşin hangi sunucu türüne gideceğini tanımlamak için kullanılır. Ör:  
mid1 kuyruğundaki barbun sunucuları
- **-d, --dependency**:İşler arasında bağımlılık yaratmak için kullanılır. Örneğin **59 numaralı iş başarılı bir şekilde tamalandıktan sonra bu işi çalıştır**
- **--array**
- **--no-requeue**
- **--reservation**

# SLURM Dosya Yapısı

## Array Kullanımı



```
#SBATCH -J gaussian09_test
#SBATCH -N 1 # hesabin dagitilacagi node adedi
#SBATCH -n 4 # is icin toplamda kullanılacak cekirdek adeti
#SBATCH --time=2:00:00
#SBATCH --array=1-100:1

export g09root=$HOME
export GAUSS_SCRDIR=/tmp/$SLURM_JOB_ID
mkdir -p $GAUSS_SCRDIR
. $g09root/g09/bsd/g09.profile

echo "SLURM_NODELIST $SLURM_NODELIST"
$g09root/g09/g09 < gaussian_input-$SLURM_ARRAY_TASK_ID.com
```

- Input dosyaları gaussian\_input-1.com, gaussian\_input-2.com vs
- Tek seferde 100 iş.



# Uygulama Derleme ve Kurma

Sistem Dizinleri ve Merkezi Dizin Farkları



- Sistem Dizinleri
  - /usr /bin /lib /lib64 /opt vs
  - Yereldir. Her sunucunun kendine özeldir.
  - Sistem dizinlerine kurulan uygulamalar sadece kurulum yapılan sunucuda kullanılabilir.
  - Aynı uygulamanın tüm sunucularda çalıştırılabilmesi için her sunucu üzerinde aynı kurulumun yapılması gereklidir.
  - Sistem dizinlerine kurulum yapmaya sadece sistem yöneticisi yetkilidir.
  - Yönetisel ve teknik sebeplerden dolayı kurulum sonrasında yeni herhangi bir uygulamanın kurulması tercih edilmemektedir.

# Uygulama Derleme ve Kurma

Sistem Dizinleri ve Merkezi Dizin Farkları



- Merkezi Dizin

- /truba , /truba\_scratch ve tüm altdizinleri
- Her suncucu tarafından erişilebilir
- Tek bir kurulum yeterlidir.
- Kullanıcılar kendi ev dizinlerine /truba/home/\$USER herhangi bir uygulamayı kurabilir
- /truba/sw ve alt dizinlerindeki kurulumlar sistem yöneticisi tarafından yapılır. Kullanıcılar herhangi bir yükleme yapamazlar ama mevcutları kullanabilirler.

# Uygulama Derleme ve Kurma

Su Apt Yum Pip ve Diğer Araçlar



- Yönetici izni gerektirir. Kullanıcıların kullanmasına izin verilmez.
- Sistem dizinlerine yükleme yaparlar. Yerel kurulum.
- Yapılacak kurulumları kullanıcılar hesaplama sunucularında kullanamazlar.
- Pip --user parametresi ile kullanıcı ev dizinine yükleme yapmak mümkündür.

# Uygulama Derleme ve Kurma

Ev dizinlerine Uygulama kurma --prefix



- Kurulum dökümanları genelde sistem yöneticisine yönelik yazılır.
- Varsayılan anlatımda, kurulum sistem dizinlerine yapılır
- Her uygulama ve her yapılandırma ve derleme adımları farklı olabilir.
- Uygulamayı derleme öncesinde yapılandırırken  
**`./configure --prefix=/truba/home/sefa/apps/`**
- Prefix ve pekçok farklı opsiyon için  
**`./configure --help`**
- Sırası ile `./configure – make – make install`

# Uygulama Derleme ve Kurma

\$PATH \$LD\_LIBRARY\_PATH Çevre Değişkenleri:



- Sistem dizinlerinde olmayan uygulama ve kütüphaneleri sistem bulamaz.
- Bulması için yardımcı olmak lazım. Ör:/truba/home/sefa/apps/gromacs 'a kurulmuş bir uygulamanın düzgün çalışabilmesi için

```
export PATH=/truba/home/sefa/apps/gromacs/bin:$PATH
```

```
export LD_LIBRARY_PATH=/truba/home/sefa/apps/gromacs/bin:$LD_LIBRARY_PATH
```

- PATH : çalıştırılabilir dosyalar için
- LD\_LIBRARY\_PATH : kütüphane dosyaları için
- Betik dosyasının içine, .bash\_profile, .profile yada .bashrc içine yazılabilir. .bash\_profile en uygunu
- Bu tanımların tüm uygulamalar için yapılması gerekir. TRUBA tarafından kurulan uygulamalar için **module** mekanizması kullanır. Export/set/unset kolaylaştırır.

# Uygulama Derleme ve Kurma

İşlemci Mimarilerine göre Derleyici Parametreleri



- Intel
  - E5v1 - AVX
  - E5v3 - AVX2
  - E5v4 - AVX2
  - GOLD - AVX512
- AMD
  - Opreron: AVX
- Derleyicilerin işlemci mimarisi parametreleri
  - Intel derleyiciler: icc ifort icpc mpiicc mpiifort vs (icc –help)
    - xHost, , –x<code> , –ax<code>, –mtune, –march, –mcpu**
  - GCC derleyiciler
    - march, –mtune, –mcpu**

- **Modül Dosyaları**

PATH, LD\_LIBRARY\_PATH ve diğer önemli çevre değişkenlerinin export/set/unset edilmesini otomatikleştir.

Örnek modül dosyası içeriği: centos7.3/comp/gcc/9.2

```
module-whatis    " loads the 7 GCC compilers and libraries "

prepend-path     --delim=:      PATH          /truba/sw/centos7.3/comp/gcc/9.2/bin
prepend-path     --delim=:      LD_LIBRARY_PATH /truba/sw/centos7.3/comp/gcc/9.2/lib
prepend-path     --delim=:      LD_LIBRARY_PATH /truba/sw/centos7.3/comp/gcc/9.2/lib64
setenv           INCLUDEDIR    /truba/sw/centos7.3/comp/gcc/9.2/include
setenv           INCLUDEDIR    /truba/sw/centos7.3/comp/gcc/9.2/include
setenv           CC            /truba/sw/centos7.3/comp/gcc/9.2/bin/gcc
setenv           CXX           /truba/sw/centos7.3/comp/gcc/9.2/bin/g++
setenv           FC            /truba/sw/centos7.3/comp/gcc/9.2/bin/gfortran
setenv           F77           /truba/sw/centos7.3/comp/gcc/9.2/bin/gfortran
```

# Modüller

Temel Komutlar



- `module avail`
- `module load centos7.3/comp/gcc/9.2`
- `module unload centos7.3/comp/gcc/9.2`
- `module list`
- `module purge`



# Modüller

Adlandırma



**centos7.3/app/gromacs/2019-impi-mkl-PS2019-GOLD**

- **centos7.3**: Hangi işletim sistemi için hazırlandığı
- **app**: modülün türü ( app, comp, lib)
- **gromacs** : uygulamanın adı
- **2019**: uygulamanın versiyonu
- **Impi**: kullanılan mpi kütüphanesi
- **Mkl**: kullanılan matematik kütühanesi
- **PS2019**: kullanılan derleyici versiyonu

# İşlerin Kuyrukta Beklemesi

- Sbatch, srun yada slalloc ile kuyruğa gönderilen işler, iş için yeterli kaynağın oluşması ve sıralarının gelmesi için için kuyrukta bir süre bekleyebilir.
- Bekleme süresini etkileyen faktörler :
  - Hesap önceliği
  - İşin gönderildiği kuyruğun önceliği
  - Talep edilen kaynak miktarı
  - Kuyrukların yoğunluğu

# İşlerin Kuyrukta Beklemesi



- **Resource:**

İşin sırası gelmiştir, yeterli kaynak oluştuğunda iş başlatılacaktır

- **Priority**

İşin sırası henüz gelmemiştir. Boşalacak kaynaklar, daha önceki işler için kullanılacaktır

- **AssocGrpCPULimit**

Hesap için izin verilen tüm çekirdek miktarı aktif olarak kullanılmaktadır. Boşta kaynak olsa bile iş çalışmayacaktır.

- **AssocGrpCPUMin**

Hesap için tanımlanan tüm hesaplama zamanı kullanılmıştır. Hesaplama zamanı kotası arttırılana kadar hesabın işleri başlatılmayacaktır.

# Tmp Ve Scratch Kullanımı

/tmp



- Yüksek IO gerektiren okuma yazma işleri
- Geçici dosyaların yazılacağı alan
- Kullanıcı arayüzü üzerinden erişim imkanı yoktur.
- İş bittikten sonra erişmek mümkün değildir.
- /tmp'nin kullanılacağı programın parametrelerinde tanımlanmalı

**Gaussian için: export GAUSS\_SCRDIR=/tmp/\$SLURM\_JOB\_ID**

**Quantum Espresso için outdir=/tmp/\$SLURM\_JOB\_ID ve disk io='none'**

- Her uygulamada farklı olabilir

Uygulamada tanımlamak mümkün değilse betik dosyası içerisine yazılacak mkdir, cp , mv, rm gibi temel Linux komutları kullanılarak input dosyaları iş başlamadan önce /tmp'ye taşınabilir, iş bittikten sonra gerekli olan veri home dizine taşınabilir.

# Tmp ve Scratch Kullanımı

/truba\_scratch/\$USER



- Merkezi dizin üzerindeki yoğunluğu azaltır.
- I/O beklemelerini en aza indirir
- İşin performansını arttır.

- **Scratch:**

- Tüm işlerin çalıştırılması için
- Yüksek performanslı
- Kotasız
- Geçici, 1 aydan eski dosyalar otomatik olarak silinir

TRUBA'daki /truba/home ve /truba\_scratch dizinleri aynı donanımlar üzerindedir. İlerki aşamalar için scratch'i ayırma planları bulunmaktadır. Sorunsuz bir geçiş için scatch dizinini kullanma alışkanlığı şimdiden edinilmelidir.

- **TRUBA**

- <http://wiki.truba.gov.tr>
- <https://www.truba.gov.tr>
- <https://portal.truba.gov.tr>
- <https://ondemand.truba.gov.tr>

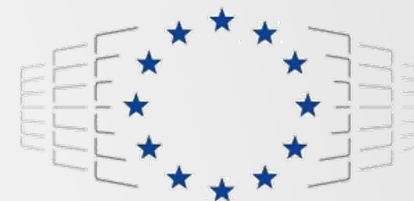
- **EuroCC**

- <https://eurocc.truba.gov.tr>
- <https://indico.truba.gov.tr>

- **Faydalı linkler**

- [https://www.youtube.com/channel/UCcApq\\_5VZaUzNsRyekWqwQA](https://www.youtube.com/channel/UCcApq_5VZaUzNsRyekWqwQA)
- <https://gcc.gnu.org/onlinedocs/gcc/Option-Summary.html>
- <https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl/link-line-advisor.html>
- <https://slurm.schedmd.com/tutorials.html>
- <http://openondemand.org>

# Teşekkürler!



**EuroHPC**  
Joint Undertaking

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 951732. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Germany, Bulgaria, Austria, Croatia, Cyprus, Czech Republic, Denmark, Estonia, Finland, Greece, Hungary, Ireland, Italy, Lithuania, Latvia, Poland, Portugal, Romania, Slovenia, Spain, Sweden, United Kingdom, France, Netherlands, Belgium, Luxembourg, Slovakia, Norway, Switzerland, Turkey, Republic of North Macedonia, Iceland, Montenegro