



Bursa Teknik Üniversitesi Yüksek Başarımlı Hesaplama Laboratuvarı

KULLANIM REHBERİ

İÇİNDEKİLER

1. Proje Başvurusu	3
2. Kurulum ve Çalışma Ortamının Hazırlanması.....	3
a. MobaXTerm Kurulumu ve Çalışma Ortamının Hazırlanması	3
b. Bitvise SSH Kullanımı	5
3. İş Talebi Gönderme.....	8
a. Ek Yazılımların Yüklenmesi	8
b. Proje dosyalarının yüklenmesi.....	8
c. İş talebinin gönderilmesi	9
4. İş Talep Dosyası.....	9
5. Çalışma Örnekleri	11
a. Örnek Keras uygulaması	11
b. Örnek PyTorch uygulaması	12

1. Proje Başvurusu

HPC laboratuvarını kullanmak için başvuru yapacak arařtırmacıların başvuru formu aracılıđı ile başvuruda bulunması gerekmektedir. Akademik Hazırlık Programı, BTÜ bünyesinde hali hazırda çalışan ve/veya öğrenim gören tüm Öğretim Elemanları ile Lisans, Yüksek Lisans ve Doktora öğrencilerine, arařtırma projeleri başvurularını teşvik etmek amacı ile sağlanacak olan ücretsiz kaynak desteđini kapsamaktadır. Program kapsamında ücretsiz 2.000 çekirdek-saat, 20 GB depolama alanı özelliklerinde kullanım hakkı tanınmaktadır.

Proje başvuruları için <http://ceng.btu.edu.tr/hpclab/> adresinde bulunan akademik başvuru formu kullanılmalıdır. BTÜ-HPCLAB, VPN bağlantı ayar dosyası, Kullanıcı Adı, Şifre, SSH bağlantı IP adresi bilgilerinizi tarafınıza e-posta yoluyla iletir.

2. Kurulum ve Çalışma Ortamının Hazırlanması

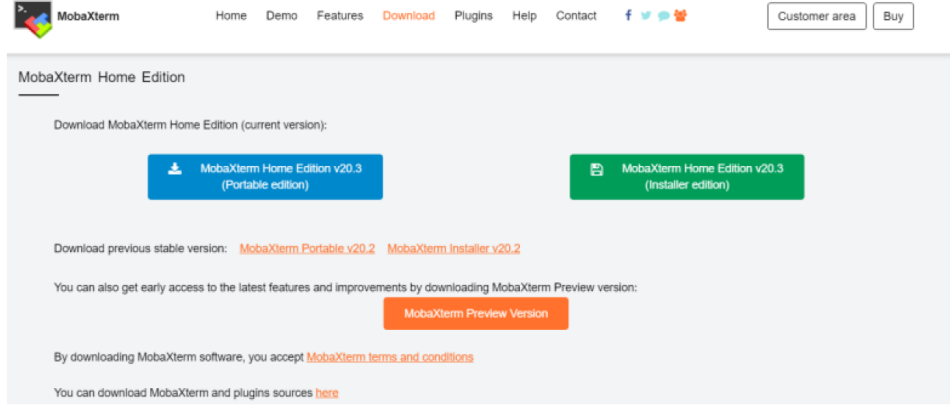
HPC hesap bilgileri kullanıcılara iletildikten sonra, kullanıcılar HPC kümesiyle bağlantı kurup, uygulama programlarını kullanabilirler. Bunun için HPC uzak bağlantısı ve çalışma ortamının hazırlanması ve çalışılacak uygulama programlarının yüklenmesi gerekmektedir. Windows ortamından HPC'ye bağlantı gerçekleřtirmek için ařađıdaki adımlar takip edilebilir.

1. Uzak sunucuya VPN üzerinden bağlantı gerçekleřtirmek için Open VPN yazılımı kullanılabilir. Bunun için bu uygulamayı řu adresten indiriniz ve kurunuz: <https://www.techspot.com/downloads/5182-openvpn.html>
2. Open VPN kurulumu gerçekleřtirildikten sonra, e-posta ile size iletilen vpn yapılandırma dosyasını uygulamaya yükleyerek HPC ađına giriř yapabilirsiniz.
3. HPC ađındaki sunucular Linux işletim sistemine sahiptir. İstemci cihazınızdan bu cihazlara bağlanmak için SSH arayüzüne ihtiyaç duyulmaktadır. Bu kapsamda çeřitli çözümler olmakla birlikte, BTÜ-HPCLAB MobaXterm ve (Bölüm 2.a) ve BitwiseSSH (Bölüm 2.b) yazılımlarının kullanımını önermektedir.

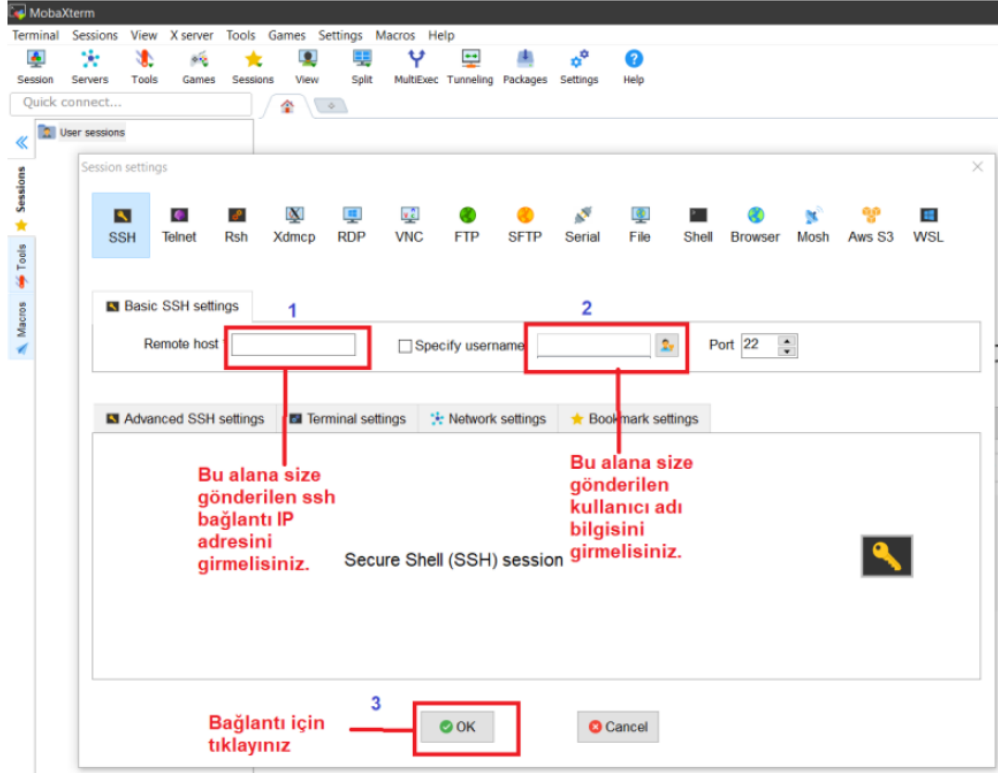
a. MobaXTerm Kurulumu ve Çalışma Ortamının Hazırlanması

1. MobaXterm yazılımı, řu adresten indirilebilmektedir (Şekil 1): <https://mobaxterm.mobatek.net/download.html>
2. Yazılımı kurulumu tamamlandıktan sonra IP, kullanıcı adı ve şifre bilgileri ile sisteme SSH bağlantısı oluřturmalısınız (Şekil 2).
3. MobaXterm SSH bağlantısı gerçekleřtirmek için MobaXterm yazılımının son üst menüsünde *Sessions* butonuna tıklanır.
4. Açılan pencerede *Remote host* ve *Specify username* bilgisi girilir ve *OK* butonuna tıklanarak bağlantı kurulur.
5. Açılan terminal ekranında HPC tarafından iletilen kullanıcı giriř bilgilerindeki şifre bilgisi girilmelidir. Bu şifre MobaXterm tarafından ilk bağlantıda sorulmaktadır. Daha sonradan tekrar aynı kullanıcı olarak bağlantı yapmak istediđinizde şifre girmeniz gerekmeyecektir.
6. Giriř yapıldıktan sonra terminalde *kullanıcıAdı@loginnode01* giriř düđümde olduđunuzu görürsünüz. Şekil 3'te MobaXterm ilk bağlantı ekranı görülmektedir. Burada sol tarafta size ait olan depolama alanını *home* dizini görebilirsiniz. Algoritmalar tarafından çalıştırılması istenen veriler ve kaynak kodları *upload* butonunu kullanılarak veya sürükleyip bırak yöntemiyle buraya eklenebilir.

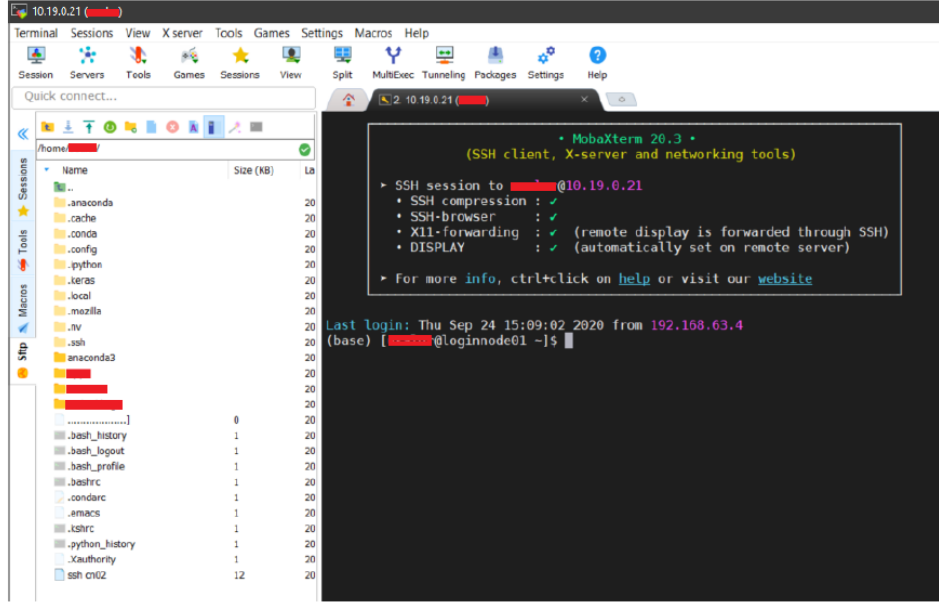
7. Gerekli dosyalar sistemde size ayrılmış olan çalışma dizinine aktarıldıktan sonra ihtiyaç duyulan ek yazılımların yüklenmesi gerekmektedir. (Bölüm 3.a)



Şekil 1: MobaXterm indirme sayfası



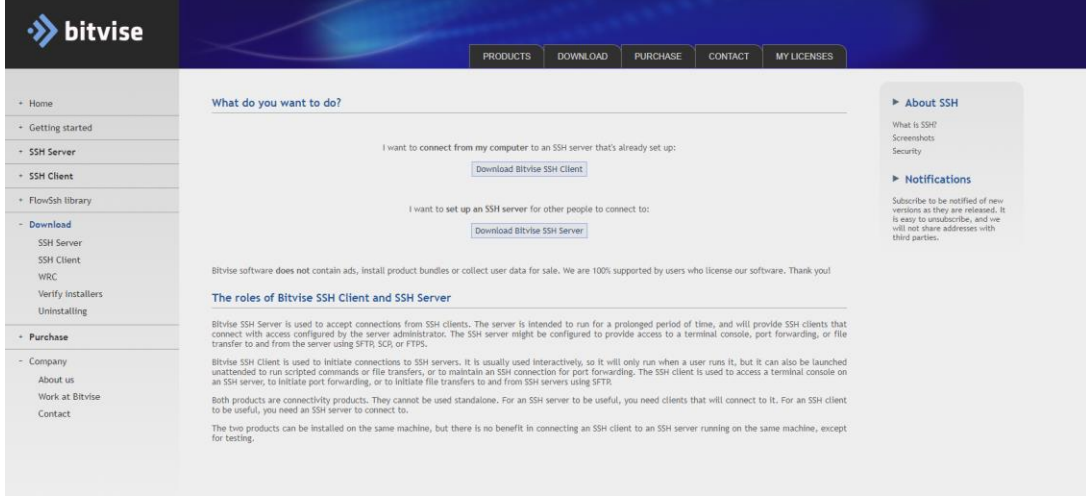
Şekil 2: MobaXterm kullanım arayüzü



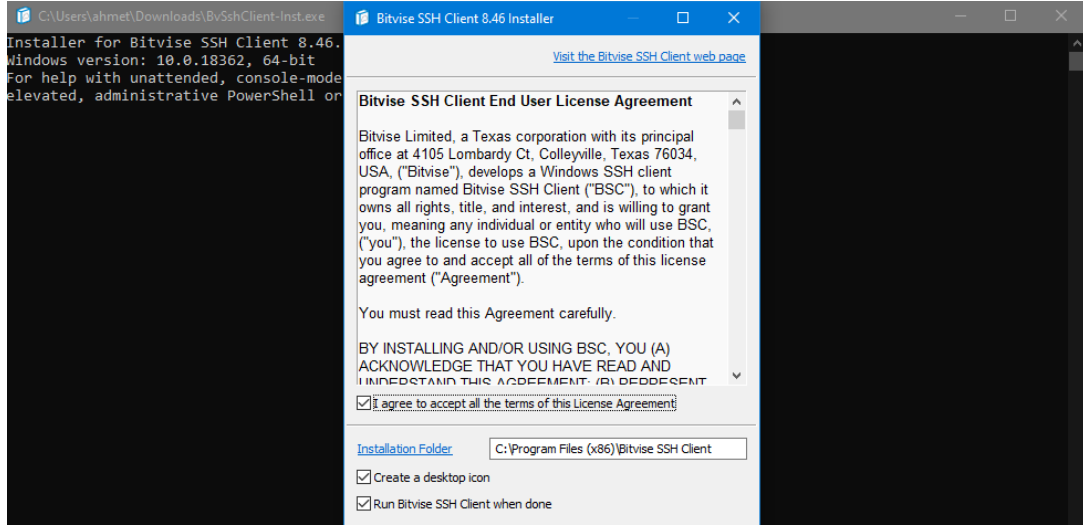
Şekil 3: MobaXterm SSH bağlantı ilk ekranı

b. Bitvise SSH Kullanımı

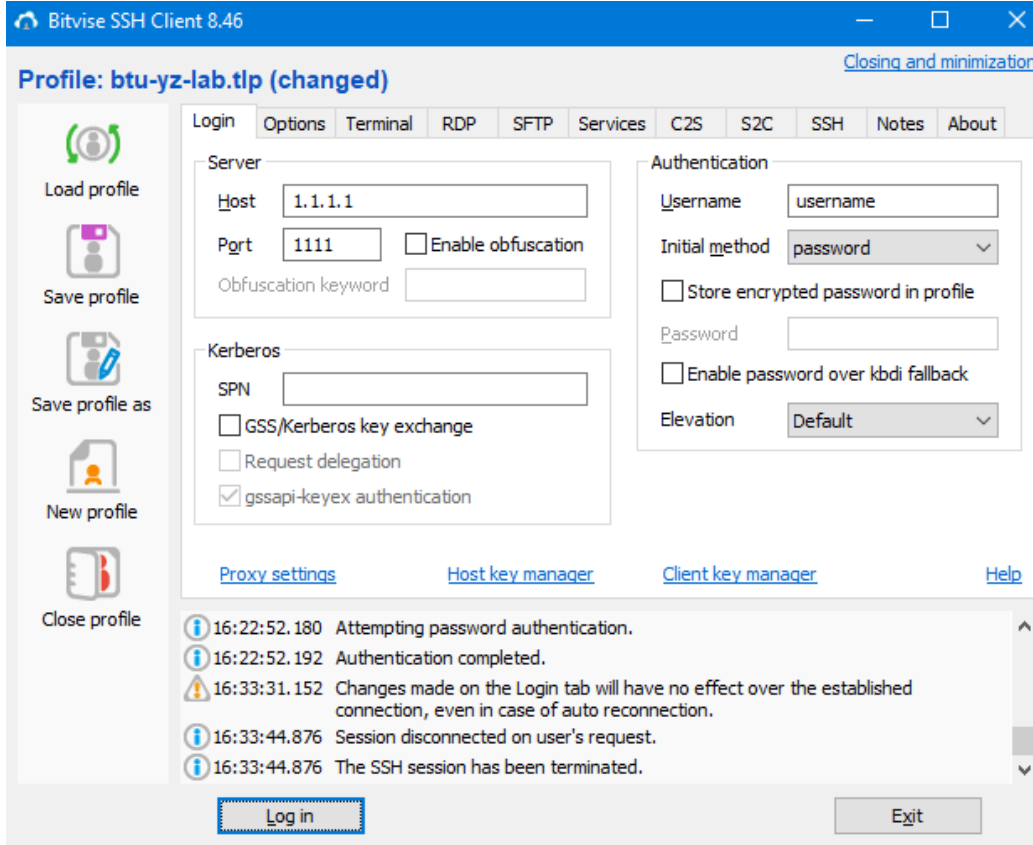
1. Şu adresten BitviseSSH İstemci (Client) kurulum dosyası indirilmelidir (Şekil 4). <https://www.bitvise.com/download-area>
2. İndirilen kurulum dosyası çalıştırılmalı ve kullanım şartları okunarak kabul edilmelidir (Şekil 5).
3. Uygulama kurulum sonrasında başlatıcı kullanılarak başlatılmalıdır (Masaüstünde kısayol oluşturulmuş olmalıdır, kontrol ediniz).
4. Giriş ekranı ile karşılaşacaksınız. Bu ekranda giriş için size iletilmiş olan IP adresi, port numarası, kullanıcı adı ve şifre bilgilerinizi ilgili alanlara girmelisiniz (Şekil 6).
5. Ardından “login” (Türkçe kullanıyor iseniz Giriş) butonuna tıklamalısınız. Bu aşamada soldaki menünün genişlediğini ve Terminal ile SFTP butonlarının eklendiğini göreceksiniz (Şekil 7).
6. Terminal butonu aracılığı ile komut satırına erişebilirsiniz, SFTP aracılığı ile Windows'ta alışılan Dosya Yöneticisine erişebilirsiniz ve dosyalarınızı kolayca sunucudaki alanınız ile kişisel bilgisayarınız arasında taşıyabilirsiniz. Burada SFTP, RDP gibi özellikler sunucuda bu özelliklerin sunuluyor olmasına bağlıdır. HPCLAB SFTP hizmeti vermek ile birlikte RDP hizmeti sunmamaktadır.
7. Giriş konfigürasyonunuzu unutmamak ve farklı sunucular arasında geçiş yaparken kolaylık sağlaması amacıyla kaydedebilirsiniz. Bu amaçla, Save profile butonunu kullanabilirsiniz (Şekil 7). Bu, belirttiğiniz dosya adresinde bir dosya oluşturacaktır. Bu dosyayı ve Load profile butonunu kullanarak daha sonraki işlemlerde profilinizi yükleyerek hızlıca kaldığınız yerden çalışmaya devam edebilirsiniz (Şekil 7).



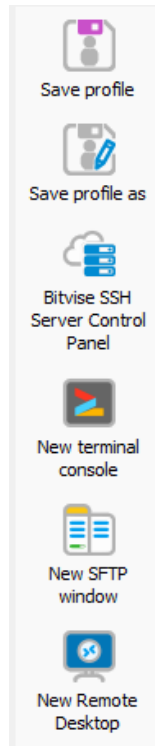
Şekil 4: BitviseSSH istemci kurulum indirme sayfası



Şekil 5: BitviseSSH kurulumu



Şekil 6: Bağlantı bilgilerinin BitviseSSH'e girilmesi



Şekil 7: Giriş sonrasında elde edilen kullanım menüsü

3. İş Talebi Gönderme

İş talepleri, slurm iş akış sistemine tabi olarak değerlendirilmektedir. Her platform için bir taslak iş ve iş talep dosyası oluşturulmuş ve /cta/share klasörü altında tüm kullanıcıların erişimine açık olarak paylaşılmıştır. Buradan yazılımınıza uygun iş talep dosyasını belirlemeli ve çalışma klasörünüze kopyalamalısınız. Kopyaladığınız iş talep dosyalarını kullanıcı klasörünüzde istediğiniz sayıda çoğaltabilir, farklı işler için kendinize gerekli sayıda öntanımlı başlatıcı konfigürasyonları tanımlayabilirsiniz. Kopyaladığınız dosya üzerinde ise sırası ile 3.a ve 3.b'de belirtilen adımları uygulamalısınız. İş talep taslak dosya isimleri yazılım platformlarına göre şu şekilde isimlendirilmiş ve konumlandırılmıştır:

- KERAS: keras_submit.sh
- TENSORFLOW: tensorflow_bench.sh, tensorflow_test.sh
- PYTORCH: slurm_pytorch.sh
- ANSYS: ansys_ksdyna_mpp_submit_v181.sh, ansys_mapdl_submit_v181.sh

a. Ek Yazılımların Yüklenmesi

Sistemde Tensorflow, Keras, PyTorch ile Ansys, FloEFD ve Converge yazılımları yüklü ve tüm kullanıcıların erişimine açık bulunmaktadır. Kullanıcının sadece ihtiyaç duyduğu ek yazılım tanımlamalarını yapması, çalışma için yeterli olmaktadır. Mevcut durumda hangi modüllerin erişilebilir olduğu şu komut yardımıyla öğrenilebilmektedir:

```
module avail
```

Bu modüllerden bir tanesi olan Anaconda ortamında bir paket ekleme gerçekleştirmek istenildiğinde, öncelikle Anaconda ortamının kişisel dizininizde aktifleştirilmesi gerekmektedir. Bunun için çalışma betiğinizdekine benzer bir şekilde (bknz. Şekil 8) şu komut çalıştırılmalıdır:

```
module load anaconda/3
```

Bu işlem sonrasında anaconda ile birlikte pip komutları da aktive olmaktadır. Böylece pip aracılığıyla istenilen yazılım yüklenebilir hale gelmektedir. Tensorflow, Keras ve PyTorch platformlarında, Python pip paket yönetim sisteminde bulunan tüm modüller bu şekilde eklenebilmektedir. Kurulum tamamlandığında başlatıcınızda module load anaconda/3 ile yüklediğiniz ortam, artık içeriğinde ek yazılımlarınızı da barındırıyor olacaktır. Eğer ek paket kullanmadıysanız, bu aşamayı atlayabilir ve 3.b aşamasına geçebilirsiniz.

```
#####
#source /etc/profile.d/lmod.sh
#####
# MODULES LOAD...
echo "Load Anaconda-3..."
module load anaconda/3
#####
```

Şekil 8: Ek yazılımların (modül) eklenmesi

b. Proje dosyalarının yüklenmesi

İş talep dosyasında kullanılacak modüller belirtildikten sonra proje dosyalarının çalışma dizininizde eksiksiz bulunduğunu doğrulayınız. Varsa eksikleri tamamlayınız ve tam veya görece yol ayarlarını

yapmayı unutmayınız. (veriseti ve kaynak kodları) sisteme yüklenmelidir. Yükleme alanı, size ayrılmış kullanıcı klasörü üzerinde dilediğiniz bir konumda bulunabilir. Bu konum proje ana klasörü olarak ifade edilmektedir.

c. İş talebinin gönderilmesi

SLURM üzerinden iş talebinde bulunulabilmektedir. Bunun için SSH bağlantısı açıldıktan sonra proje ana klasörüne gidilmeli ve sbatch komutu ile iş talebi gönderilmelidir. Örnek bir iş talebi şu komut yardımı ile gönderilebilmektedir:

```
sbatch keras_submit.sh
```

İş taleplerinin bir cihaza atanması sistemin yoğunluğuna göre değişkenlik göstermektedir. Bu nedenle iş göndermeden önce sistemin yoğunluğunu kontrol etmeniz gerekmektedir. Sistemdeki mevcut işlerin listesine şu komut yardımı ile ulaşılabilir:

```
squeue
```

İşiniz bir cihaza atandıktan sonra yine donanımsal gereksinim tespiti ve kaynak paylaşımı işinizin türüne göre 5-15 dakika sürebilmektedir. Bu, SLURM yöneticisinin tüm kullanıcılar için en optimize kaynak paylaşımını yapabilmesi için gereklidir. Eğer işinizin atama yapıldıktan sonra çok daha uzun sürelerde çalışmaya başladığını fark ederseniz, support@compacta.com.tr adresinden destek talebinde bulunabilirsiniz.

4. İş Talep Dosyası

İş talep dosyası, içeriğindeki parametreler vasıtasıyla yazılımınızın ihtiyaç duyduğu donanımsal ve yazılımsal kaynağı tanımlayabilmenize izin vermektedir (Şekil 9). Eğer yazılımınız daha fazla donanıma ihtiyaç duyuyorsa bu değerleri artırabilir, ihtiyaçtan fazla kullanıyorsa düşürebilirsiniz. İhtiyaç fazlası donanım kullanımı, iş talebinizin karşılanmasında gecikmeye, bazen hiç karşılanamamasına sebep olabileceğinden, kullanıcının uygun bir değer seçmesi gerekmektedir.

```

#!/bin/bash
#
# BTU HPC-LAB (c) 2021
# -= Resources =-
#
#SBATCH --job-name=jobname
#SBATCH --ntasks=1
#SBATCH --gres=gpu:1
#SBATCH --partition=short
#SBATCH --output=logs/t-%j.out
#SBATCH --mail-user=name.surname@btu.edu.tr

#####
#source /etc/profile.d/lmod.sh
#####

# MODULES LOAD...
echo "Load Anaconda-3..."
module load anaconda/3

#####

echo ""
echo "===== ENVIRONMENT VARIABLES ====="
env
echo "===== "
echo ""
echo ""

echo "Running Keras-Tensorflow command..."
echo "===== "
RET=$?

python /cta/users/akasif/kaGAN.py --num_windows 32 \
--seq_window_lengths 3 4 8 \

```

Şekil 9: Örnek iş talep dosyası

Şekil 9’da 1 numaralı dikdörtgen içerisinde işinize dair genel tanımlamalar yapılmakta ve kaynak talepleri belirtilmektedir.

--job-name: İşinizin adını tanımlarsınız.

--ntasks: Kaç düğüm kullanacağınızı belirtir.

--gres=gpu: Kaç GPU talebiniz olduğu belirtilir.

--partition: İşinizi ne kadar süre ile çalıştırmak istediğinizi belirtirsiniz. Sadece şu değerleri alabilir *short*, *longer*.

--output: İşinizin çıktı dökümlerinin saklandığı dosyadır. Burada çıktı dosyanızın hangi konumda ve hangi isimle oluşturulacağı belirtilmektedir.

--mail: E-posta iletişim adresi tanımlanabilmektedir. Olası sorunlarınızda destek taleplerinizin daha hızlı karşılanabilmesini sağlamaktadır.

Şekil 9’da 2 numaralı dikdörtgen içerisinde ortam değişkeni tanımlamaları yapılabilmektedir. Şekil 9’da 3 numaralı dikdörtgen içerisinde ise uygulamanızın başlatıcı dosyası (Şekil 9 için akasif kullanıcısının çalışma dizininde kaGAN.py dosyası) tam konum ve isim ile belirtilmekte, ayrıca varsa parametre tanımlamaları yapılmaktadır.

5. Çalışma Örnekleri

Bu bölümde, Keras, Tensorflow, PyTorch, Ansys ve Floent yazılımları ile örnek uygulamaların nasıl çalıştırılabileceğine dair tanıtımlar yer almaktadır.

a. Örnek Keras uygulaması

Bu bölümde, Keras platformu kullanılarak MNIST veriseti üzerinde sınıflandırma hedefleyen bir Derin Öğrenme çalışmasının nasıl uygulandığı açıklanmaktadır. Keras platformu BTU-HPC ortamında tanımlı ve çalışma herhangi bir ek paket kullanımı gerektirmemektedir. Bu nedenle tamamlanması gereken ilk görev kaynak kodu ve veriseti dosyalarının sisteme yüklenmesidir. Dosya yüklemelerinin yeni kullanıcılar için kullanıcı kök dizinine yapılması önerilmektedir. Böylece dosyaların yol tanımlamalarını yaparken isimlerini kullanmak yeterli olacaktır. Şekil 10'da bu örnekte kullanılan kaynak kodu görülmektedir. Bu kod kök dizinine yüklenmiştir.

```
import numpy as np
from tensorflow import keras
from tensorflow.keras import layers

# Model / data parameters
num_classes = 10
input_shape = (28, 28, 1)

# the data, split between train and test sets
(x_train, y_train), (x_test, y_test) = keras.datasets.mnist.load_data()

# Scale images to the [0, 1] range
x_train = x_train.astype("float32") / 255
x_test = x_test.astype("float32") / 255
# Make sure images have shape (28, 28, 1)
x_train = np.expand_dims(x_train, -1)
x_test = np.expand_dims(x_test, -1)
print("x_train shape:", x_train.shape)
print(x_train.shape[0], "train samples")
print(x_test.shape[0], "test samples")

# convert class vectors to binary class matrices
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)

model = keras.Sequential(
    [
        keras.Input(shape=input_shape),
        layers.Conv2D(32, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Conv2D(64, kernel_size=(3, 3), activation="relu"),
        layers.MaxPooling2D(pool_size=(2, 2)),
        layers.Flatten(),
        layers.Dropout(0.5),
        layers.Dense(num_classes, activation="softmax"),
    ]
)

model.summary()

batch_size = 128
epochs = 15

model.compile(loss="categorical_crossentropy", optimizer="adam", metrics=["accuracy"])

model.fit(x_train, y_train, batch_size=batch_size, epochs=epochs, validation_split=0.1)

score = model.evaluate(x_test, y_test, verbose=0)
print("Test loss:", score[0])
print("Test accuracy:", score[1])
```

Şekil 10: Keras'ta Mnist Sınıflandırma Kaynak Kodu

Kaynak kodu ve veriseti kök dizinine aktarıldıktan sonra iş talep dosyasının oluşturulması gerekmektedir. Keras için tanımlı iş talep dosyası /cta/share/job_scripts/ klasörü altında bulunan keras_submit.sh dosyasıdır. Bu dosyadan bir adet kullanıcı dizinine kopyalanmalıdır. Bu işlem için gerek SSH uygulamamızın dosya yönetici arayüzü gerekse Linux dosya kopyalama komutu (cp) kullanılabilir. Dosya kopyalandıktan sonra yine ister arayüzden bilgisayarınızdaki bir editör ile açılması sağlanarak, ister Nano ve Vim editörler kullanarak dosya üzerinde temel ayarların girilmesi sağlanır.

Kaynak kodu, veriseti ve iş talep dosyası hazırlandıktan sonra geriye aşağıdaki komutu çalıştırarak iş talebinde bulunmak kalmaktadır. İşin mevcut durumunu gözlemek için *squeue* komutu kullanılabilir, erken bitirmek için *scancel* komutu değerlendirilebilmektedir.

```
sbatch <iş_talep_dosyası_adi.sh> (sbatch keras_submit.sh)
```

Yazılımınız çalışma zamanı sona erdiğinde, iş talep dosyasında tanımladığınız dosya yolunda bir rapor oluşturmaktadır. Bu raporu inceleyerek tüm loglara erişim sağlanabilmektedir. Raporun öntanımlı adı şu şekildedir:

```
<iş_numarası>-deep.out (3524 numaralı iş için 3524-deep.out)
```

b. Örnek PyTorch uygulaması

Bu bölümde, PyTorch platformu kullanılarak bebek görüntüleri ile yapay yüz üretimi amacıyla “baby-faces” veriseti üzerinde yapay içerik üretimi hedefleyen bir Üretici Çekişmeli Ağ çalışmasının nasıl uygulandığı açıklanmaktadır. PyTorch platformu BTU-HPC ortamında tanımlı ve çalışma herhangi bir ek paket kullanımı gerektirmemektedir. Bu nedenle tamamlanması gereken ilk görev kaynak kodu ve veriseti dosyalarının sisteme yüklenmesidir. Dosya yüklemelerinin yeni kullanıcılar için kullanıcı kök dizinine yapılması önerilmektedir. Böylece dosyaların yol tanımlamalarını yaparken isimlerini kullanmak yeterli olacaktır. Şekil 10’da bu örnekte kullanılan kaynak kodu görülmektedir. Kod uzun olduğundan, fonksiyon gövdeleri çıkarılmıştır. Bu kod kök dizinine yüklenmiştir.

```
# Set random seed for reproducibility
manualSeed = 999
#manualSeed = random.randint(1, 10000) # use if you want new results
print("Random Seed: ", manualSeed)
random.seed(manualSeed)
torch.manual_seed(manualSeed)
# Root directory for dataset
dataroot = "baby-dataset"
# Number of workers for dataloader
workers = 2
# Batch size during training
batch_size = 128
# Spatial size of training images. All images will be resized to this
# size using a transformer.
image_size = 64
# Number of channels in the training images. For color images this is 3
nc = 3
# Size of z latent vector (i.e. size of generator input)
latentSize = 100
# Size of feature maps in generator
ngf = 64
# Size of feature maps in discriminator
ndf = 64
# Number of training epochs
num_epochs = 20
# Learning rate for optimizers
lr = 0.0002
# Beta1 hyperparam for Adam optimizers
beta1 = 0.5
# Number of GPUs available. Use 0 for CPU mode.
ngpu = 1

# We can use an image folder dataset the way we have it setup.
# Create the dataset
dataset = dset.ImageFolder(root=dataroot,
                           transform=transforms.Compose([
                               transforms.Resize(image_size),
                               transforms.CenterCrop(image_size),
                               transforms.ToTensor(),
                               transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5)),
                           ]))
# Create the dataloader
dataloader = torch.utils.data.DataLoader(dataset, batch_size=batch_size,
                                         shuffle=True, num_workers=workers)
# Decide which device we want to run on
device = torch.device("cuda:0" if (torch.cuda.is_available() and ngpu > 0) else "cpu")

# custom weights initialization called on netG and netD
def weights_init(m):
    # Generator Code
class Generator(nn.Module):
class Discriminator(nn.Module):

if name == 'main':
```

Şekil 10: PyTorch'ta Mnist Sınıflandırma Kaynak Kodu

Kaynak kodu ve veriseti kök dizinine aktarıldıktan sonra iş talep dosyasının oluşturulması gerekmektedir. Keras için tanımlı iş talep dosyası /cta/share/job_scripts/pytorch/ klasörü altında bulunan slurm_pytorch.sh dosyasıdır. Bu dosyadan bir adet kullanıcı dizinine kopyalanmalıdır. Bu işlem için gerek SSH uygulamamızın dosya yönetici arayüzü gerekse Linux dosya kopyalama komutu (cp) kullanılabilir. Dosya kopyalandıktan sonra yine ister arayüzden bilgisayarınızdaki bir editör ile açılması sağlanarak, ister Nano ve Vim editörler kullanarak dosya üzerinde temel ayarların girilmesi sağlanır.

Kaynak kodu, veriseti ve iş talep dosyası hazırlandıktan sonra geriye aşağıdaki komutu çalıştırarak iş talebinde bulunmak kalmaktadır. İşin mevcut durumunu gözlemlemek için *squeue* komutu kullanılabilir, erken bitirmek için *scancel* komutu değerlendirilebilmektedir.

```
sbatch <iş_talep_dosyası_adi.sh> (sbatch slurm_pytorch.sh)
```

Yazılımınız çalışma zamanı sona erdiğinde, iş talep dosyasında tanımladığınız dosya yolunda bir rapor oluşturmaktadır. Bu raporu inceleyerek tüm loglara erişim sağlanabilmektedir. Raporun öntanımlı adı şu şekildedir:

```
<iş_numarası>-deep.out (3412 numaralı iş için 3412-deep.out)
```