T.C. SANAYİ VE TEKNOLOJİ BAKANLIĞI

#MİLLİ TEKNOLOJİ HAMLESİ

TÜBİTAK

TÜBİTAK ULAKBİM Türk Ulusal Bilim e-Altyapısı (TRUBA)

Fadime GÖNÜLERİ, Sevil SARIKURT MALCIOĞLU

arfq@truba.gov.tr

TÜBİTAK ULAKBİM
AĞ TEKNOLOJİLERİ BİRİMİ
TRUBA EKİBİ

31 AĞUSTOS 2026

ÖNEMLİ HATIRLATMA

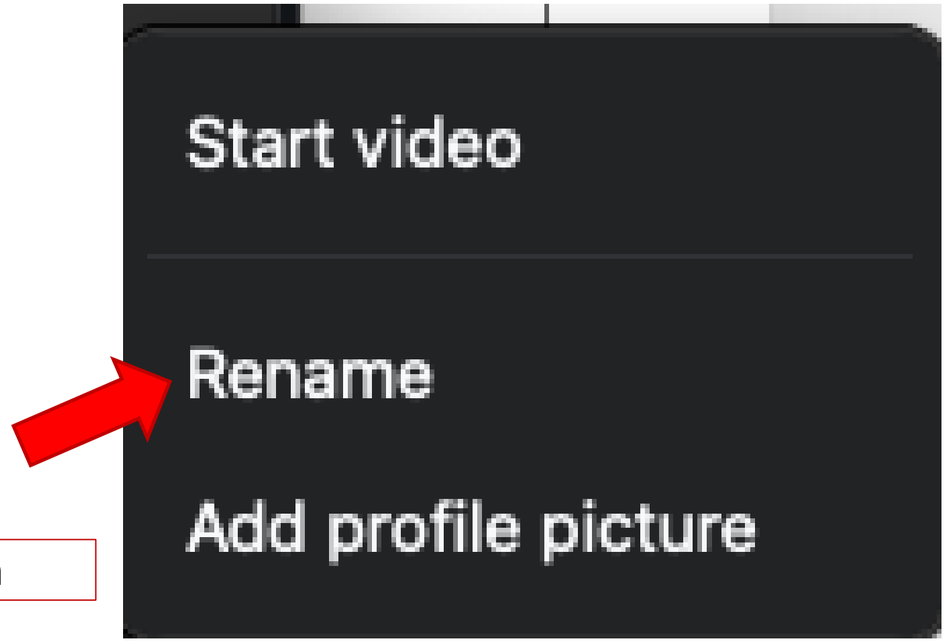
- **Bugünkü TRUBA bilgilendirme oturumuna her takımdan en az 2 (iki) üyenin katılımı zorunludur. Eğitime hiçbir üyesi katılmayan takımlar, TRUBA sistemi üzerinde yürütülecek ikinci ön eleme (uygulamalı değerlendirme) aşamasına dahil edilmeyecektir. Eğitime katılım şartını sağlayan takımlara TRUBA kullanıcı hesabı bilgileri iletilecektir.**
- **Zoom'da takma isim kullanılmamalı. Zoom oturumunda Gerçek Ad-Soyad bilgisi ile Takım Adı Bilgisi girilmeli.**
- **Bilgilendirme kılavuzu:**

https://indico.truba.gov.tr/e/SSBquantumalg

Zoom

TRUBA HandsOn Egitimi-Kuantum Algorit...

TRUBA_On_Eleme_Kurallar_ve_Bilgilendir...



Ad Soyad-QubitTeam

İÇİNDEKİLER

https://dareplanet.tech/en/insights/high-performance-computing-review/

Yüksek Başarımlı Hesaplama Kaynağı

- Performansı çok yüksek bilgisayar sistemleri.
- Bir masa/dizüstü bilgisayardan çok daha yüksek hesaplama gücüne ulaşmak için birden çok bilgisayarın bir araya getirildiği sistemler.
- Böylece bilim, mühendislik ve iş hayatındaki büyük problemler çözülebilir.

*Yüksek Başarımlı Hesaplama (YBH), paralel işleme tekniklerinin kullanılmasıyla, normal bilgisayarlarla çözülemeyen karmaşık ve büyük ölçekli simülasyon ve yapay zeka problemlerinin **süper bilgisayar altyapılarında çözülmesidir**.*

- TRUBA ARF Hesaplama Kümesi: 35 bin dizüstü bilgisayara eşit işlem kapasitesine sahip

Yüksek Başarımlı Hesaplama Kullanım Alanları

TÜBİTAK ULAKBİM

Bilimsel Kullanım

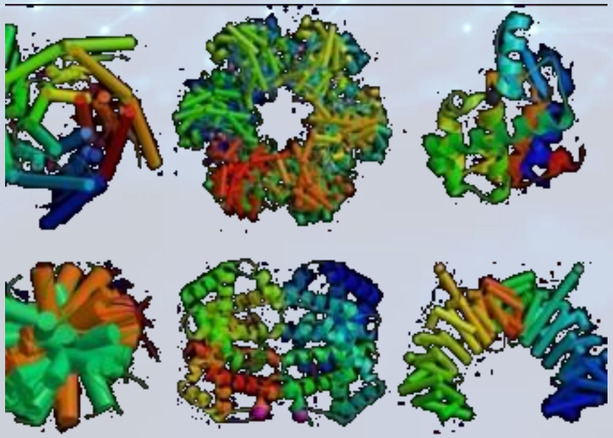
- İklim modelleme
- Hesaplamalı kimya
- Biyoinformatik, ilaç dizaynı
- Hesaplamalı akışkanlar dinamiği
- Hesaplamalı malzeme bilimleri
- Astrofiziksel modellemeler

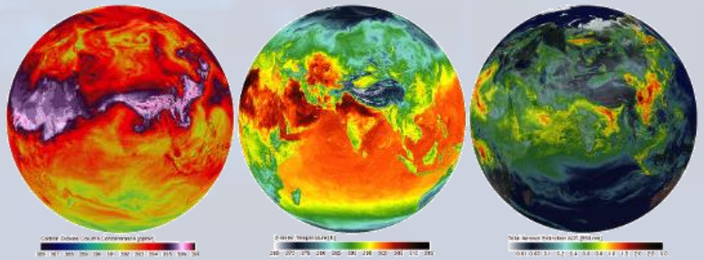
Endüstriyel Kullanım

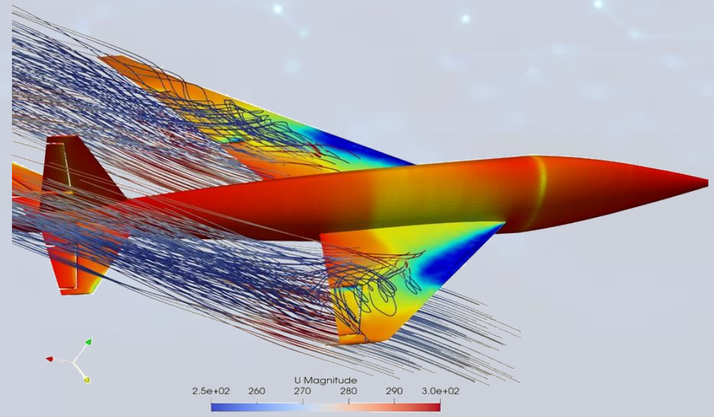
- Ürün tasarımı ve üretimde zaman kazanma
- Yeni malzemelerin tasarımının hızlanması
- Finans ve ekonomik modelleme
- Web servisleri, tarama motorları

Yapay Zeka Uygulamaları

- Yüksek hızda veri analizi
- Büyük veri ile veri madenciliği
- Yapay sinir ağlarının eğitilmesi
- Görüntü işleme
- Büyük dil modelleri (ChatGPT, LLAMA, BART, LLaMA)







Türk Ulusal Bilim e-Altyapısı (TRUBA)

TÜBİTAK ULAKBİM Veri Merkezi – ARF

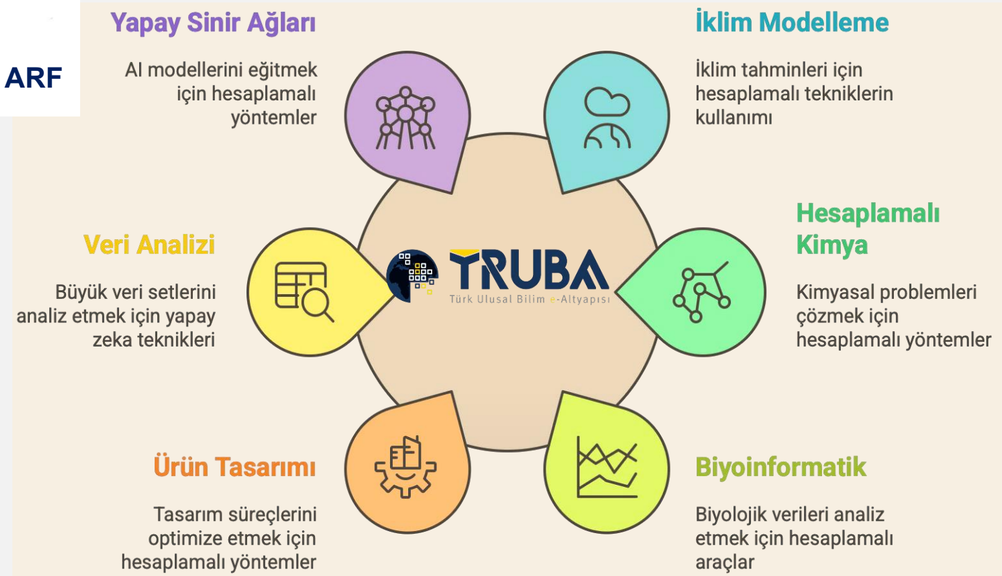
HEDEFLERİMİZ

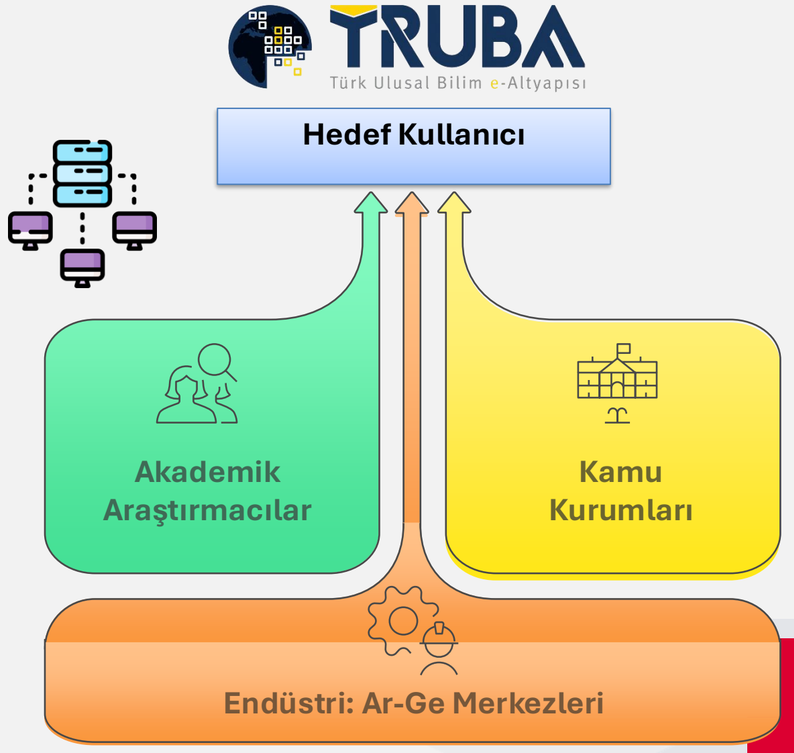
- Araştırmacılarımızın dünya ile eşit şartlarda; rekabet etmelerine olanak sağlayacak araştırma altyapısı ve destek servislerini sunmak,
- Özel/kamu/üniversite araştırmacılarına, mevcut altyapı ve bilgi birikimi kullanılarak, en ileri teknoloji imkanları ulusal çapta sunmak.

KAYNAKLARIMIZ

- 80.000 CPU
- 504 GPU kartı
  - 144-(P100, V100), 72-A100, 96-H100, 192-H200
- ~22PB yüksek performanslı depolama

https://docs.truba.gov.tr/





Top500 & Green500

- HPL (LINPACK) testi, sistemleri dünyanın en hızlı 500 (Top500) ve en enerji-verimli 500 (Green500) süper bilgisayarı listelerine yerleştirir.
- ARF sistemi orfoz kümesinden, ARF-ACC sistemi kolyoz-cuda (NVIDIA H100/H200) kümesinden oluşmaktadır.

Dönem	ARF (Top500)	ARF-ACC (Top500/Green500)
Haziran 2026	447	165/54
Kasım 2025	422	145/45
Haziran 2025	395	266/37
Kasım 2024	354	227/26
Haziran 2024	313	


TOP 500 CERTIFICATE
The List.
ARF - ThinkSystem SD650 v3, Xeon Platinum 8480+ 56C 2GHz, Infiniband NDR200
TRUBA, Turkey
is ranked
No. 313
among the World's TOP500 Supercomputers
with 3.00 PFlop/s Linpack Performance
in the 63rd TOP500 List published at the ISC24
Conference on June 01, 2024.
Congratulations from the TOP500 Editors
Erich Strohmaier NERSC/Berkeley Lab
Jack Dongarra University of Tennessee
Horst Simon NERSC/Berkeley Lab
Martin Meuer Prometeus


TOP 500 CERTIFICATE
The List.
ARF ACC - ThinkSystem SD650-N V3, Xeon Gold 6548Y+ 32C 2.5GHz, NVIDIA H100/H200, Infiniband NDR200, Rocky Linux 9.4
TRUBA, Turkey
is ranked
No. 145
among the World's TOP500 Supercomputers
with 12.09 PFlop/s Linpack Performance
in the 66th TOP500 List published at the SC2025
Conference on November 18, 2025.
Congratulations from the TOP500 Editors
Erich Strohmaier NERSC/Berkeley Lab
Jack Dongarra University of Tennessee
Horst Simon NERSC/Berkeley Lab
Martin Meuer Prometeus


The GREEN 500 CERTIFICATE
ARF ACC - ThinkSystem SD650-N V3, Xeon Gold 6548Y+ 32C 2.5GHz, NVIDIA H100 80GB HBM3, Infiniband NDR200, Rocky Linux 9.4
TRUBA, Turkey
is ranked
No. 26
among the World's TOP500 Supercomputers
with 52.547 GFlops/watts Performance
in the Green500 List published at the SC24
Conference on November 19, 2024.
Congratulations from the Green500 Editors
Wu-chun Feng Virginia Tech
Kirk Cameron Virginia Tech

https://top500.org/lists/top500/
https://top500.org/lists/green500/2026/06/

TRUBA Sistemine Bağlantı – ARF ACC Kümesi

TRUBA sistemindeki hesaplama kümelerine bağlantı iki aşamada gerçekleştirilir:

- **TRUBA OpenVPN bağlantısının kurulması**
  - **OpenVPN istemcisi çalıştırılır.**
  - **TRUBA VPN profili kullanılarak bağlantı kurulur.**
- **SSH Bağlantısı**
  - **Kullanıcı arayüz sunucusuna bağlantı**

    ssh -l <<kullaniciadi>> 176.16.6.16

Yarışma kapsamında tanımlanan kullanıcı isimleri **<<hackathongXXuY>>** formatındadır.

→ XX kısmı takım numarasını,
→ Y kullanıcı numarasını belirtmektedir.
TRUBA kullanıcı adı ve şifre bilgileri katılımcılara ayrıca bireysel olarak iletilecektir.

https://docs.truba.gov.tr/2-temel_bilgiler/openvpn_installation/openvpn_info.html
https://docs.truba.gov.tr/1-kaynaklar/arf_acc/arf_acc_baglanti.html

TRUBA Dosya Sistemi (Kullanıcı Dizinleri)

Alan	Dosya Yolu (path)	Kullanım Amacı	Kullanım Kısıtlamaları ve Kritik Uyarılar
Home (Kalıcı depolama alanı)	/arf/home/$USER	Ev dizini, küçük boyutlu program kurulum dosyaları Kalıcı olarak saklanması gereken küçük çıktı dosyaları	Büyük veri setleri saklanmamalıdır Yoğun I/O gerektiren hesaplamalar çalıştırılmamalıdır Uzun süreli veri arşivleme yapılmamalıdır
Scratch (Geçici ve yüksek hızlı depolama alanı)	/arf/scratch/$USER	Aktif simülasyonların çalıştırılması, Çıktı dosyaları için geçici depolama, Büyük veri setleri üzerinde işlem yapılması Paralel I/O gerektiren uygulamalar	Önemli veriler/çıktı dosyaları mutlaka yerel bilgisayara kopyalanmalıdır Uzun süreli veri saklanamaz Belirli süre üzerinde işlem yapılmayan dosyalar sistem tarafından periyodik olarak silinir

- Simülasyonlar için "/arf/scratch/username" dizininin kullanılması gerekmektedir.
- “Home” kotası dolunca SSH ile bağlantı başarısız olabilir.
- Conda/pip ile doğrudan /arf dizini altına kurulum önerilmiyor → binlerce küçük dosya=inode limiti

Kota kontrolü:
du –sh /arf/home/$USER

Dosya sayısı:
find /arf/home/$USER –type f | wc –l

Yerel bilgisayardan TRUBA’ya, TRUBA’dan yerel bilgisayara dosya transferi:
https://docs.truba.gov.tr/2-temel_bilgiler/dosya_transferi/index.html

ARF ACC Hesaplama Kümesi – Kuyruk Bilgileri

ARF ACC Erişim Politikası → Sadece yetkili projeler

Kuyruk	Sunucu Sayısı	GPU Modeli	Sunucu Başına GPU	GPU Başına Çekirdek Sayısı (Minimum CPU)	Maksimum İş Çalıştırma Süresi
palamut-cuda	9	NVIDIA A100 80 GB	8	16	3 gün
kolyoz-cuda (H100)	24	NVIDIA H100 80 GB	4	16	3 gün
kolyoz-cuda (H200)	48	NVIDIA H200 141 GB	4	16	3 gün

- İş gönderimi için **en az 16 çekirdek ve en az 1 GPU** talep edilmelidir.
- kolyoz-cuda kuyruğuna iş gönderirken SLURM betik dosyasında GPU tipini **-C** parametresiyle H100 veya H200 olarak belirtmelisiniz.
- Kuyruk ve donanım detaylarının güncel durumunu görmek için cuda-ui arayüzünde aşağıdaki komutu kullanabilirsiniz:

  scontrol show partition=<kuyruk_adi>

Temel Erişim ve İş Akışı

SLURM Kaynak Yöneticisi ve Betik Dosyası

- SLURM (Simple Linux Utility for Resource Management): açık kaynak kodlu iş yükü ve kuyruk yönetim sistemi
- Çalıştırılacak uygulamanın ihtiyaç duyduğu kaynaklar bir betik dosyası (genellikle `.slurm` veya `.sh` uzantılı) içerisinde `#SBATCH` yönergeleri ile tanımlanır.

SLURM betik dosyasında belirtilmesi gereken temel kaynak parametreleri

Kuyruk Adı	İlgili simülasyonun hangi kuyrukta yürütüleceği	#SBATCH --partition
SLURM Hesabı	Sisteme iş göndermekle yetkili hesap adı	#SBATCH --account
Düğüm Sayısı	İş yükünün kaç farklı sunucuya dağıtılacağı	#SBATCH --nodes
İşlemci Çekirdeği (CPU)	Süreç veya iş parçacığı (thread) başına ayrılacak çekirdek miktarı	#SBATCH --cpus-per-task
GPU Sayısı	Kullanılacak grafik işlemci sayısı ve türü	#SBATCH --gres=gpu
Zaman Limiti (Walltime)	İşin tamamlanması için talep edilen azami çalışma süresi	#SBATCH --time

Örnek SLURM Betik Dosyası

```
#!/bin/bash
#SBATCH --account=hackathongXX      # cekirdek-saat kotasi olan takim hesabi adi
#SBATCH --reservation= ssbquant     # yarisma kapsaminda rezerve edilen sunucuları kullanmak için gerekli
#SBATCH --job-name=kuantum_sim      # Isin adi
#SBATCH --partition= kolyoz-cuda    # Isin calisacagi kuyruk bilgisi
#SBATCH -C H200                      # Bu satir H200 GPU'sunu belirtiyor. → Yarışma kapsamında özel rezervasyon tanımı olduğu için ihtiyaç yok
#SBATCH --nodes=1                           # Kullanilacak dugum (node) sayisi
#SBATCH --cpus-per-task=16                  # Gorev başına tahsis edilecek CPU çekirdek sayısı
#SBATCH --gres=gpu:1                        # Talep edilen GPU kaynagi: 1 adet GPU
#SBATCH --time=00:30:00                     # Azami çalışma süresi (days-hours:minutes:seconds)
#SBATCH --output=cikti_%j.out               # Standart çıktı dosyası (%j = Job ID)
#SBATCH --error=hata_%j.err                 # Hata çıktı dosyası (%j = Job ID)

# 1. Ortam Degiskenleri ve Modul Yukleme
echo "Is Baslatildi: $(date)"
echo "Calisilan Dugum: $(hostname)"

# Gerekli modulleri yukleyin veya conda ortamınızı aktif edin
# module load ...
# source activate kuantum_env

# 2. Uygulamanın Calistirilmasi
python kuantum_simulasyonu.py

echo "Is Tamamlandi: $(date)"
exit
```

Belirtilen zaman (`--time`) sınırı aşıldığında SLURM işi otomatik olarak sonlandırır

https://docs.truba.gov.tr/2-temel_bilgiler/slurm-betik-ozellik.html
https://docs.truba.gov.tr/2-temel_bilgiler/gpu.html

Temel SLURM Komutları

Komut	Görevi ve Açıklaması
sinfo –p kuyruk_adi	İlgili kuyrukta yer alan sunucular hakkında genel bilgi verir (idle, allocated, mix v.b)
sbatch run.slurm	İş betiğini SLURM kuyruk sistemine gönderir ve bir Job ID üretir.
squeue -u $USER	İşi sisteme gönderen kullanıcıya ait kuyrukta bekleyen (PD/Pending) ve çalışan (R/Running) işleri listeler.
sacct -j JOB_ID	Tamamlanan, zaman aşımına uğrayan veya hata alan işin durumunu ve geçmiş kaynak kullanımını raporlar.
sstat -j JOB_ID	Çalışan ilgili işin CPU ve bellek kullanımını gösterir
scancel JOB_ID	Belirtilen JOBID'ye sahip kuyruktaki veya çalışmakta olan işi iptal eder/durdurur.
cat cikti.out	İş tamamlandığında üretilen çıktıyı görüntüler.
srun --jobid=<JOB_ID> --overlap --pty bash	İşin çalıştığı sunucuya (iş aktif olarak çalışırken) bağlantı sağlar. Yeni bir terminal üzerinden bağlantı sağlamanız önerilir. İlgili terminalde “nvidia-smi” komutu ile çalışan işin GPU’lar üzerindeki yükünü gözlemleyebilirsiniz.

Temel SLURM Komutları

SLURM ile ilgili detaylı bilgiler için TRUBA dokümantasyon sayfasındaki bilgiler incelenebilir:

→ **SLURM Komutları:**

https://docs.truba.gov.tr/2-temel_bilgiler/slurm_komutlari_ve_dosyalari.html

→ **SLURM Betik Özellikleri:**

https://docs.truba.gov.tr/2-temel_bilgiler/slurm-betik-ozellik.html

İş Gönderiminde (Job Submission) Dikkat Edilmesi Gereken Hususlar

TÜBİTAK ULAKBİM

TRUBA'da kaynakların verimli kullanılması, işlerinizin kuyrukta bekleme sürelerinin optimize edilmesi ve hesaplamalarınızın kesintiye uğramadan tamamlanması için iş betiklerinizi (.slurm) hazırlarken aşağıdaki parametrelere dikkat edilmelidir:

Çekirdek & Bellek Orantısı	**TRUBA'da işleriniz için SLURM betik dosyanızda belirttiğiniz çekirdek sayısıyla orantılı olarak bellek ayrılmaktadır. Toplamda ihtiyaç duyduğunuz bellek miktarına göre farklı işlemci ve görev sayıları ile ilgili bellek miktarına erişebilirsiniz**	**Talep edilen bellek miktarı uygulamanın anlık gereksiniminin altında kalırsa işiniz sistem tarafından Out Of Memory (OOM) hatasıyla sonlandırılır. Yüksek RAM gereksinimi için CPU artırın.**
Kuyruk (Partition) ve Süre	**H200 GPU'lar için kolyoz-cuda kuyruğunu seçin; «--time» değerini işin zaman aşımına uğramayacağı güvenli bir sınırla belirleyin.**	**Süre aşıldığında işiniz SLURM tarafından *Time Limit Exceeded* uyarısıyla durdurulur.**
Yerleşim Geometrisi	**Paylaşımlı bellek ve tek sunucu GPU işleri için --nodes=1 kullanın; iş parçacığı ihtiyacını --cpus-per-task ile karşılayın.**	
Hızlandırıcı (GPU) & Konteyner	**--gres=gpu:X ile GPU'ları besleyecek yeterli CPU çekirdeği ve sistem belleği ayrıldığını, ayrıca Apptainer üzerinden çalıştırılıyorsa --nv bayrağının aktif edildiğini doğrulayınız.**	

TRUBA Türk Ulusal Bilim e-Altyapısı

İş Gönderiminde (Job Submission) Dikkat Edilmesi Gereken Hususlar

Hızlı Kontrol Listesi

```
#SBATCH --partition=kolyoz-cuda      # İş yüküne uygun doğru partition seçildi mi?
#SBATCH --nodes=1                    # Yerleşim geometrisi tek düğüm olarak ayarlandı mı?
#SBATCH --cpus-per-task=16           # CPU/Thread sayısı ihtiyacı tam karşılıyor mu?
#SBATCH --gres=gpu:1                 # GPU kaynağı doğru talep edildi mi?
#SBATCH --time=01:00:00              # Süre limiti güvenli bir pay ile belirlendi mi?
#SBATCH --reservation=ssbquant       # Rezervayon tanımı yapıldı mı?
#SBATCH --account=hackathongXX # SLURM'a iş göndermekle yetkili TRUBA proje kodu yazıldı mı?

SIF_PATH=/arf/sw/containers/quantumcomp/cuquantum-25.11-cuda13.0.1.sif

# Konteyner GPU bayrağı (--nv) ile mi başlatılıyor?
apptainer exec --nv $SIF_PATH python kod.py
```

Konteyner Teknolojisi Nedir?

Uygulamayı; çalışma zamanı (runtime), kütüphaneler, bağımlılıklar ve ortam değişkenleriyle birlikte tek bir paket haline getirir.

Kırılgan Yazılım Ortamı

Bağımlılık çakışmaları ve tutarsızlık

Tutarlı Yazılım Ortamı

Taşınabilir, güvenilir ve tutarlı

Uygulamayı bağımlılıklarla birlikte paketler

Yazılım ortamını sistemden izole eder

Ekipler arası iş birliğini artırır

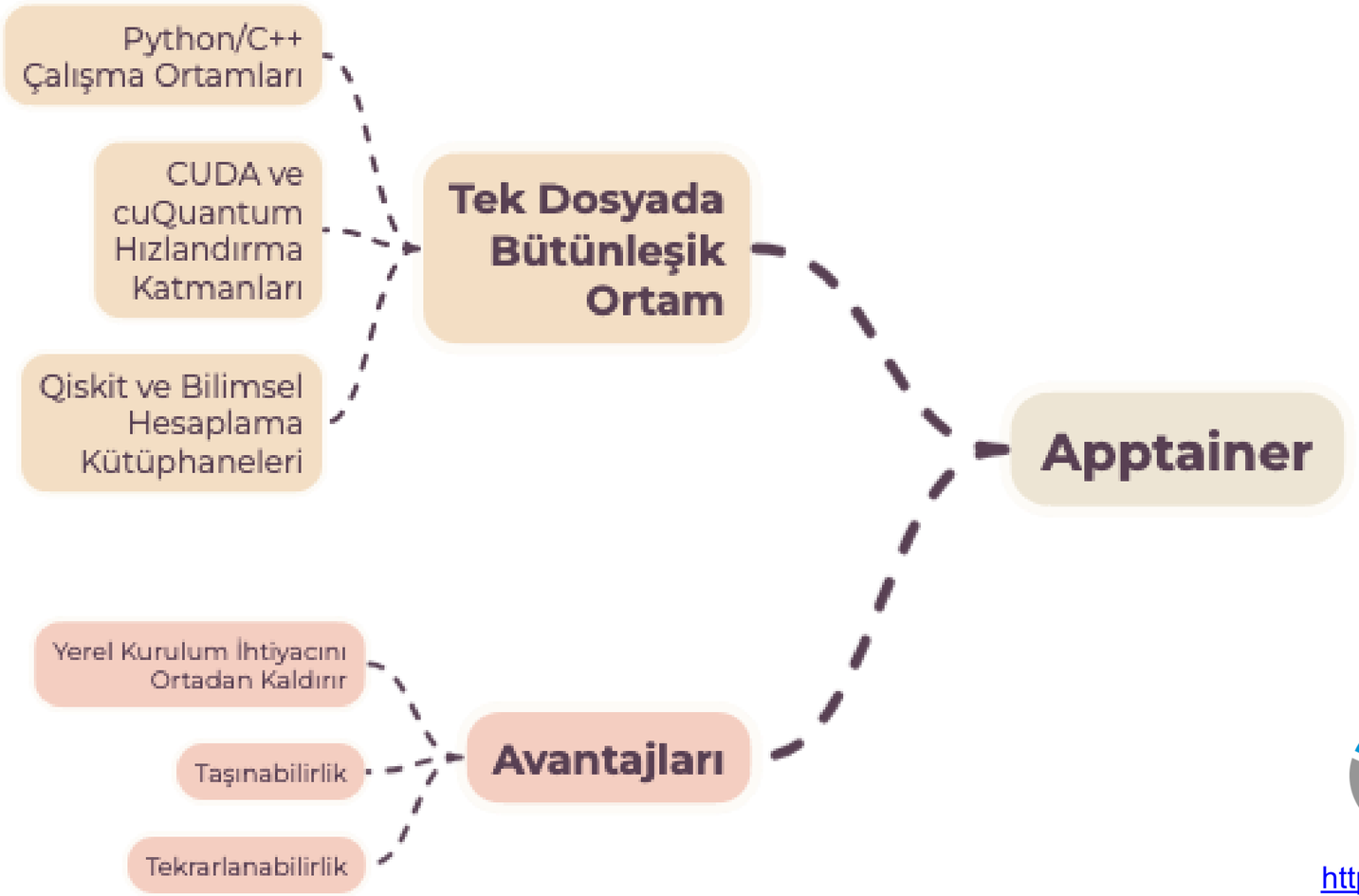
Konteynerler HPC İş Akışında Neyi Çözer?

İş Akışı Aşaması	Konteynerin Çözümü / Katkısı
Geliştirme	Tüm kütüphane ve araçları tek bir ortamda paketleme
Paylaşım	Ekip üyelerine aynı çalışma ortamını sunma
Toplu İş (Batch)	SLURM işlerini her düğümde standart ve tutarlı ortamda koşturma/çalıştırma
Hata Ayıklama	Sorunlu çalışma ortamını yerelde veya kümede kolayca yeniden oluşturma
Tekrarlanabilirlik	Aynı .sif imajı ile aynı koşulları yakalama

- ***Çözdüğü Problem*** → **Yazılım ortamı uyumsuzlukları, derleme karmaşıklığı ve bağımlılık çatışmaları.**
- ***Çözmediği / Kapsamadığı Durumlar*** → **Hatalı SLURM betikleri, verimsiz I/O ve veri organizasyonu, HPC kaynak/kota politikaları.**

Apptainer Nedir?

HPC İçin Optimize Konteyner Mimarisi: Kök yetkisi (root/sudo) gerektirmeyen, çok kullanıcılı süper bilgisayar güvenlik modeline tam uyumlu platformdur.



https://apptainer.org/docs/user/latest/

**Singularity→Apptainer: 2021 yılında Linux Foundation çatısı altına geçerek Apptainer adını almıştır. Eski literatür ve dokümanlardaki "Singularity" terimi ile Apptainer aynı teknolojiyi temsil eder.

Apptainer Temel Çalıştırma Komutları

Kullanım Amacı	Örnek Komut	Açıklama
Standart Yürütme (CPU)	apptainer exec imaj.sif python kod.py	Python kodunu konteyner ortamında CPU üzerinde çalıştırır.
GPU Hızlandırmalı Yürütme	apptainer exec --nv imaj.sif python kod.py	NVIDIA GPU ve CUDA sürücülerini konteynere bağlar.

Yarışma Kapsamında Kullanacağımız Konteyner İmajı:

SIF_PATH=/arf/sw/containers/quantumcomp/cuquantum-25.11-cuda13.0.1.sif

Yarışma Ortamı İçin Örnek Komut:

apptainer exec --nv $SIF_PATH python kuantum_kod.py

TRUBA Konteyner Kullanım Dokumantasyonu
https://docs.truba.gov.tr/2-temel_bilgiler/konteyner_kullanimi.html

Apptainer’e Ek Python Paketleri Kurulumu

TRUBA’da kullanılan Apptainer, temel çalışma ortamını sağlar. Ancak bazı projelerde konteyner içinde hazır bulunmayan ek Python paketlerine ihtiyaç olabilir. Bu durumda konteyner içeriğini değiştirmek yerine, ek paketler kullanıcı çalışma dizini altında pydeps adlı (klasör adı isteğe bağlı olarak değişkenlik gösterebilir) bir klasöre kurulabilir.

pydeps, proje için gerekli ek Python kütüphanelerinin tutulduğu klasördür. SLURM dosyasında bu klasör PYTHONPATH ile tanıtıldığında Python, çalıştırma sırasında bu paketleri görebilir.

TRUBA’ da Örnek Kullanımı

- **Önce proje klasörüne geçilir:**

```
cd /arf/scratch/$USER/proje_adi

#pydeps klasörü oluşturulur:
mkdir -p pydeps
```

- **Gerekli paketler bu klasöre kurulur:**

```
apptainer exec -B "$PWD:$PWD" /arf/sw/containers/quantumcomp/cuquantum-25.11-cuda13.0.1.sif python -
m pip install --target "$PWD/pydeps" qiskit-optimization==0.6.1
```

- **SLURM dosyasında Python çalıştırılmadan önce şu satır eklenir:**

```
export PYTHONPATH="$PROJECT_DIR/pydeps:$PYTHONPATH"
```

Apptainer İmajını «Definition File» ile Özelleştirme

- Hazır sunulan .sif imajında bulunmayan özel Python paketlerini mevcut çalışma ortamına entegre etmek için mevcut .sif dosyasını temel (base image) alan bir Definition File (.def) hazırlayarak üzerine yalnızca gerekli ek kütüphaneleri kurulabilir.
- Avantajları:
  - Sıfırdan CUDA, cuQuantum veya Qiskit derleme zahmetini ortadan kaldırır.
  - Temel imajın kararlılığını korurken dakikalar içinde güncel ve özelleştirilmiş yeni bir .sif üretir.
- Definition File (custom_env.def) Mimarisi
  - Mevcut bir .sif imajını genişletmek için kullanılan temel yapı:

```
Bootstrap: localimage
From:/arf/sw/containers/quantumcomp/cuquantum-25.11-cuda13.0.1.sif

# İhtiyaç duyulan ek Python kütüphanelerinin kurulumu
%files
        requirements.txt /requirements.txt

%post
        pip install --no-cache-dir –r /requirements.txt

%test
        python –c "import numpy; print('numpy OK:', numpy._version_)"

%runscript
        python "$@"
```

→ Bootstrap: localimage & From: Kaynak olarak yerel diskteki mevcut .sif dosyasını gösterir.

→ %post: Yeni paketlerin imajın içine kalıcı olarak yüklendiği derleme adımıdır.

→ %test build sırasında otomatik çalışır – başarısız ise build durur.

Yeni İmajın Derlenmesi (apptainer build)

custom_env.def

```
Bootstrap: localimage
From:/arf/sw/containers/quantumcomp/cuquantum-25.11-cuda13.0.1.sif

# İhtiyaç duyulan ek Python kütüphanelerinin kurulumu
%files
        requirements.txt /requirements.txt

%post
        pip install --no-cache-dir –r /requirements.txt

%test
        python –c "import numpy; print('numpy OK:', numpy._version_)"

%runscript
        python "$@"
```

TRUBA Konteyner Kullanım Dokumantasyonu
https://docs.truba.gov.tr/2-temel_bilgiler/konteyner_kullanimi.html

TRUBA Apptainer Eğitimi
https://github.com/TRUBA-HPC/Intro2ContainerUsageHPC

- **«custom_env.def» dosyasında ilgili düzenlemeler yapıldıktan sonra yetkili derleme ortamında:**

  apptainer build customized_quantum.sif custom_env.def

  komutu çalıştırılır.

- **Çalıştıracağınız kod için artık yeni oluşturduğunuz *.sif dosyasını (customized_quantum.sif) kullanmanız gerekmektedir. SLURM betik dosyanızda «customized_quantum.sif» dosyanızın yer aldığı dizini belirtmeniz gerekmektedir. Örneğin:**

  SIF_Custom=/arf/home/hackathongXX/customized_quantum.sif

  apptainer exec --nv $SIF_ Custom python kuantum_kod.py

--nv Bayrağı

- «--nv» bayrağı, Apptainer/Singularity gibi konteyner araçlarında, konteyner içinde NVIDIA GPU desteğini etkinleştirmek için kullanılır.
- Ne Yapar?
  - Ana sistemdeki (işin çalıştığı sunucudaki) NVIDIA CUDA sürücülerini ve aygıt düğümlerini (/dev/nvidia*) dinamik olarak konteyner içine eşler.
  - GPU hızlandırmalı kütüphanelerin (cuQuantum, CUDA-Q, PyTorch vb.) fiziksel GPU'yu doğrudan görmesini ve kullanmasını sağlar.
- Ne Yapmaz?
  - SLURM üzerinden GPU tahsis etmez (GPU betik dosyasında #SBATCH --gres=gpu:X ile istenmelidir).
  - Fiziksel düğümde bulunmayan sanal bir GPU kaynağı oluşturmaz.

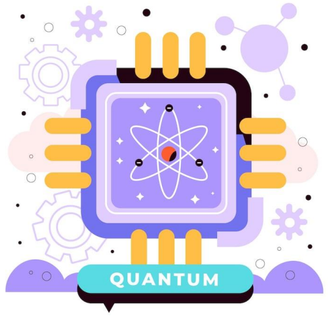
https://apptainer.org/docs/user/main/gpu.html#nvidia-gpus-cuda-standard

Kuantum Devre Simülasyonu ve GPU Hızlandırması

- Kuantum devrelerinin klasik donanımlar üzerinde simüle edilmesi, kubit sayısı arttıkça üstel olarak büyüyen durum uzayı (2^N) nedeniyle devasa bellek ve işlem gücü gerektirir.
- Qiskit Aer ve NVIDIA cuQuantum SDK, durum vektörü ve tensör ağı işlemlerini NVIDIA GPU mimarisi üzerinde paralelleştirerek hesaplama sürelerini önemli ölçüde kısaltır.
- NGC tarafından sunulan ve TRUBA'da modifiye edilen Apptainer imajı; Qiskit Aer, CUDA ve cuQuantum hızlandırma katmanlarını önceden entegre edilmiş olarak barındırır ve doğrudan NVIDIA H200 GPU gücünden yararlanmanızı sağlar.

Kübit Sayısı	Hafıza Gereksinimi	Sunucu Sayısı	Sistemdeki Hafıza (ARF-ACC H200)
32	64 GB	1	564 GB
33	128 GB	1	564 GB
34	256 GB	1	564 GB
35	512 GB	2	960 GB
36	1024 GB	2	1.1 TB
37	2048 GB	4	~2.2 TB
38	4 TB	8	~4.5 TB
39	8 TB	16	~9 TB
40	16 TB	32	~18 TB

"Memory" = precision x 2^n



Qiskit Aer:https://github.com/Qiskit/qiskit-aer
NVIDIA cuQuantum SDK: https://developer.nvidia.com/cuquantum-sdk
CUDA Quantum Örnekleri: https://nvidia.github.io/cuda-quantum/0.7.0/using/examples/cuquantum.html

Qiskit Aer ile GPU Simülasyonu Tanımlama

- **Qiskit Aer Mimarisi: Kuantum devrelerinin geleneksel bilgisayarlar üzerinde simülasyonu için optimize edilmiş farklı yöntemler ve arka uçlar (backends) sunar.**
- **GPU Hızlandırmalı Simülatör Tanımı:**

```
from qiskit_aer import AerSimulator
# GPU üzerinde durum vektörü (statevector) tabanlı simülatör başlatma
simulator = AerSimulator(method="statevector", device="GPU")
```

→ **method="statevector"**

Kuantum durumunun 2^N boyutlu durum vektörü yaklaşımıyla tam doğrulukla simüle edileceğini belirtir.

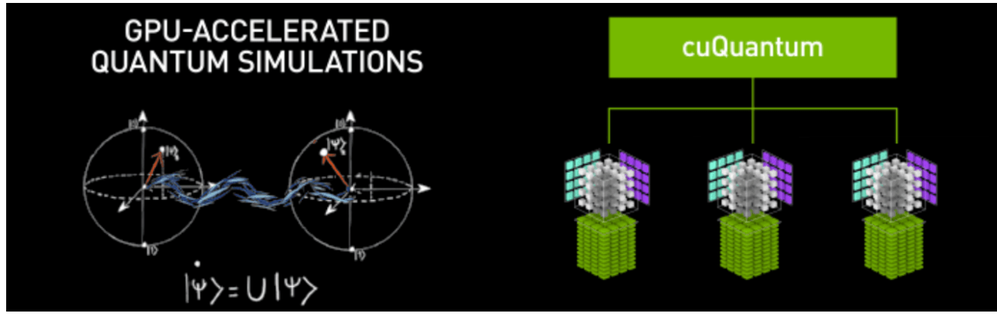
→ **device="GPU"**

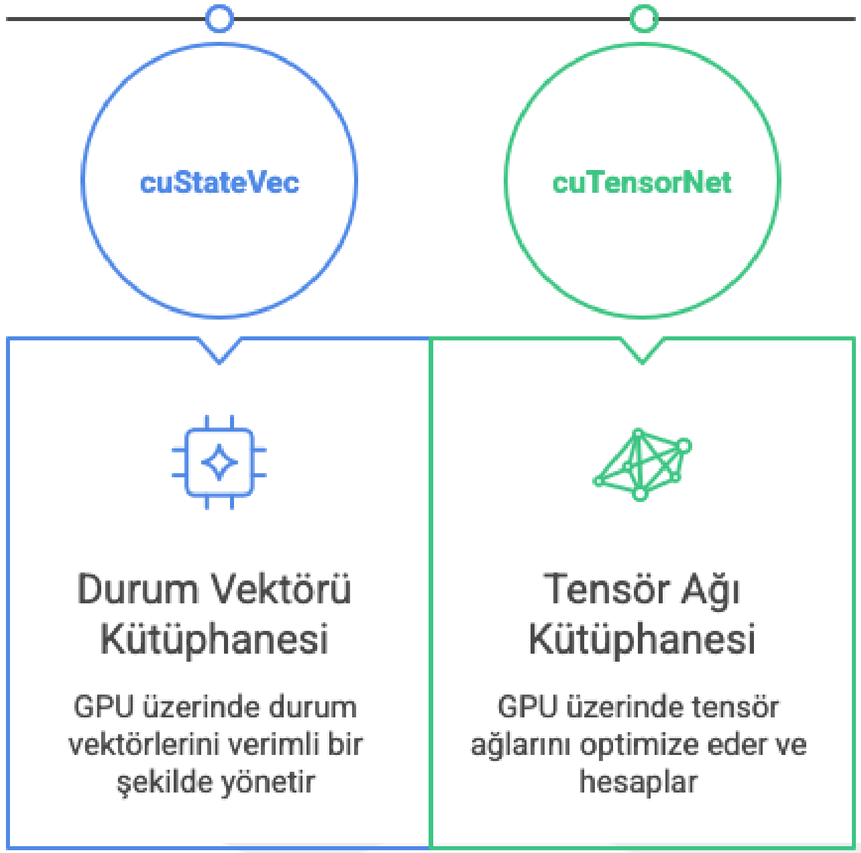
Lineer cebir ve kapı matris operasyonlarının NVIDIA GPU üzerinde koşturulacağını ifade eder.

Qiskit Aer:
https://github.com/Qiskit/qiskit-aer
https://qiskit.github.io/qiskit-aer/stubs/qiskit_aer.AerSimulator.html

NVIDIA cuQuantum Kütüphane Bileşenleri

- cuQuantum SDK, GPU üzerinde çalışan iki temel hızlandırma kütüphanesinden oluşur:
- **cuStateVec (Durum Vektörü Kütüphanesi):**
  - 2^N boyutlu durum vektörlerini doğrudan GPU küresel belleğinde (HBM3e) işler.
  - Tek ve çok kübitli kuantum kapı operasyonlarını yüksek bellek bant genişliğiyle uygular.
  - Multi-GPU Desteği: Durum vektörünü dilimleyerek (state partitioning) birden fazla H200 GPU arasında dağıtır.
- **cuTensorNet (Tensör Ağı Kütüphanesi):**
  - Kuantum devrelerini tensör ağları (tensor networks) biçiminde modeller.
  - Düşük dolaşıklığa sahip veya derin devrelerde durum vektörüne alternatif ölçekleme sunar.
  - İşlem sırasını optimize ederek bellek tüketimini ve hesaplama süresini minimuma indirir (Contraction Optimization).





NVIDIA cuQuantum SDK: https://developer.nvidia.com/cuquantum-sdk
CUDA Quantum Örnekleri: https://nvidia.github.io/cuda-quantum/0.7.0/using/examples/cuquantum.html

TRUBA’da Uygulama Çalıştırma

SSH Bağlantısı

TRUBA'ya SSH ile bağlan

Kod Hazırlığı

Python ve SLURM iş dosyaları hazırlanır

İş Gönderme

İş sbatch ile kuyruğa gönderilir

Düğüm Ayırma

SLURM uygun hesaplama düğümünü ayırır

Konteyner Başlatma

Apptainer konteyneri başlatılır

GPU Çalıştırma

Python kodu GPU üzerinde çalışır

Uygulama Çalıştırma

Uygulama başarıyla çalışıyor

Önemli Bilgiler: İletişim Kuralları ve Final Raporu

Destek talebi oluştururken mesajınızda aşağıdaki bilgilerin yer alması zorunludur:

- Takım Adı
- TRUBA Proje Kodu (hackathongXX)
- TRUBA Kullanıcı Adı (hackathongXXuY)
- İlgili İş Numarası (SLURM Job ID) ve Hatanın Alındığı Çalışma Dizini
- SLURM betik dosyasının adı
- Alınan Hata Mesajının Detayı (Log çıktısı)

Sınav süresince yaşanabilecek teknik sorunlara yönelik destek talep etmek/destek vermek üzere SLACK kanalı kullanılacaktır.

Sorularınızı SLACK kanalı üzerinden iletebilirsiniz.

slack

3 Eylül 2026 Perşembe günü gerçekleşecek olan yarışma kapsamında yarışmacılar final raporlarını INDICO üzerinden açılacak kayıt formları ile **belirtilecek süre içerisinde** yükleyeceklerdir.

https://indico.truba.gov.tr/e/SSBquantumalg

Önemli Bilgiler

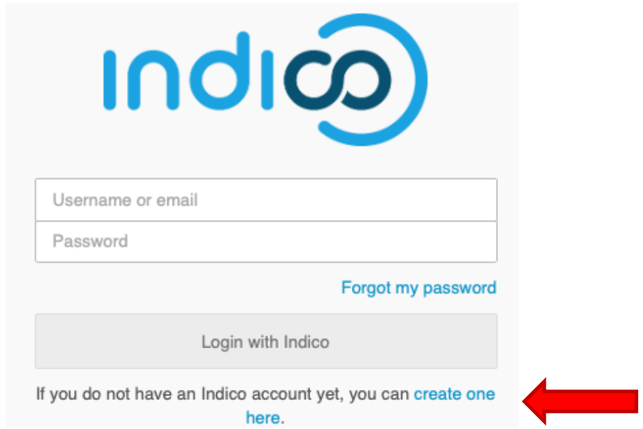
- Yarışma günü katılımcılardan talep edilecek final raporları Indico sayfası üzerinden sisteme yüklenecektir.
  - Final raporunun sisteme yüklenmesinden Takım Lideri sorumludur.
  - Indico'da final raporunun yüklenmesi için oluşturulan sayfa yarışma günü belirlenen zaman aralığında otomatik olarak açılacaktır ve yarışma bitiminde otomatik olarak kapanacaktır.
  - **Final raporunu sisteme yükleyebilmek için sorumlu kişinin (Takım Lideri'nin) Indico sayfasında kullanıcı hesabı açması gerekmektedir.**
  - Indico kullanıcı hesabı/profili https://indico.truba.gov.tr sayfası üzerinden yarışma gününden önce mutlaka yapılmalıdır.

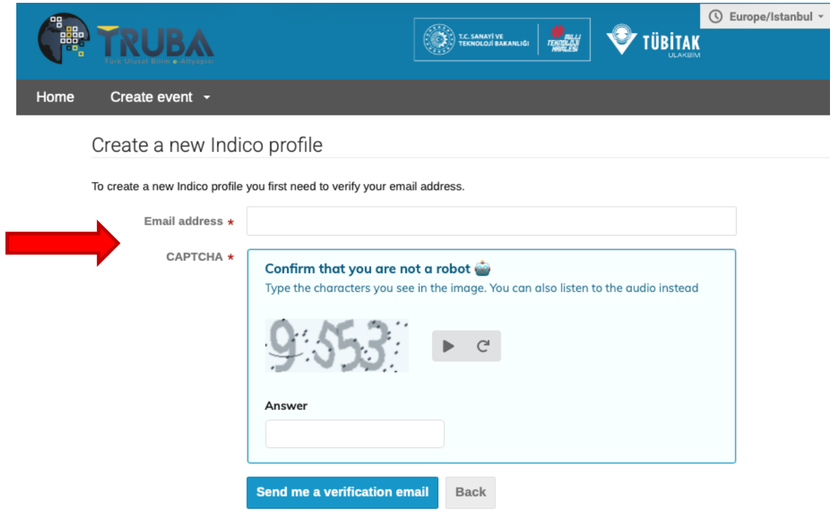
Genel bilgiler, eğitime ait sunum dosyaları/dokumanlar vb. bu yarışma için açılmış olan Indico sayfası üzerinden de paylaşıma açık olacaktır:

https://indico.truba.gov.tr/e/SSBquantumalg

Önemli Bilgiler: TRUBA Indico Sayfası Profil Oluşturma

https://indico.truba.gov.tr





- Final raporunu sisteme yükleyebilmek için Takım Lideri'nin Indico sayfasında kullanıcı hesabı açması gerekmektedir.

Önemli Bilgiler

- **TRUBA kullanıcı adı ve şifreleri kişiye özeldir.**
  - İlgili kullanıcı bilgilerinin bir başkası ile paylaşılması TRUBA kullanım politikasına aykırıdır.
  - Kullanıcı adı ve şifre bilgisi ile tek bir OpenVPN bağlantısı sağlanabilir. Farklı bir cihazdan ikinci bir OpenVPN bağlantısına izin verilmemektedir.
- Sisteme iş göndermekle yetkili hesap takımlara atanmış olan «hackathongXX» grup hesaplarıdır.
- «hackathongXXuY» bireysel kullanıcı hesaplarıdır.

T.C. SANAYİ VE TEKNOLOJİ BAKANLIĞI

#MİLLİ TEKNOLOJİ HAMLESİ

TÜBİTAK ULAKBİM

TEŞEKKÜRLER

TÜBİTAK | TÜRKIYE BILIMSEL VE TEKNOLOJIK ARAŞTIRMA KURUMU

Kullanıcı Destek

trubadestek@ulakbim.gov.tr

TRUBA Dokümantasyon

http://docs.truba.gov.tr

TRUBA Sosyal Medya

https://www.linkedin.com/company/truba