

## LAB 2

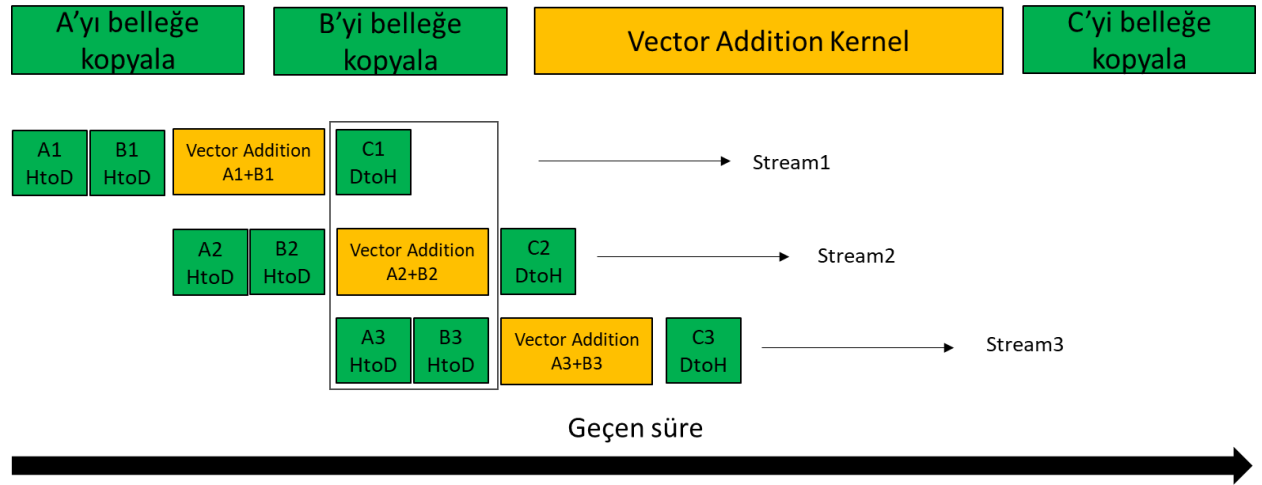
### Vector Addition with Streams

#### AMAÇ

Bu labda Lab1'de çözülmüş olan "vector addition" örneğinin CUDA Stream'ler kullanılarak hızlandırılması amaçlanmaktadır. Katılımcılar verilen kodun GPU ile alakalı olan kısımlarını CUDA programlama dili ile yazacaklardır.

#### PROBLEM

Verilen iki farklı vektörün aynı indeksteki elemanlarının toplandığı ve sonucun üçüncü vektöre kaydedildiği bu problemde veri seti üç eşit parçaya ayrılıyor ve her bir parça birbirinden bağımsız şekilde "vector addition" işlemi yapıyor. Amacımız tek GPU üzerinde bu üç parçanın işlemlerini CUDA Stream kullanarak örtüştürmek ve hızlanma elde etmektir. Örnek görsel aşağıda verilmektedir:



#### YAPILACAKLAR

CUDA kodu yazılarak gerçekleştirilecek işlemler aşağıda verilmektedir:

- GPU ana belleğinde yer ayrılacak
- Blok büyüklüğü ve grid büyüklüğü belirlenecek
- CUDA Stream'ler yaratılacak
- Parçaların aşağıdaki işlemleri CUDA Stream'ler kullanılarak overlap edilecektir
  - CPU belleğinden GPU belleğine veri transferi
  - CUDA Kernel çalıştırma
  - GPU belleğinden sonuçların CPU belleğine transferi
- GPU belleği temizlenecek
- CUDA Kernel içeriği yazılacak

**Lab2\_Soru.cu** dosyası kullanılacak. Bu dosyadaki **//@** ile başlayan satırlara ilgili kodlar yazılacak. **vim Lab2\_Soru.cu** komutu ile dosyayı açıp **i** tuşuna basınca dosyada değişiklikler yapabilirsiniz. **Lab2\_Soru.cu** dosyasındaki değişiklikleri ilk kez kaydederken önce **Esc** tuşuna sonra **:wq!** tuşlarına basmalısınız. Daha sonra kaydederken önce **Esc tuşuna** sonra **:wq** tuşlarına basabilirsiniz.

### **KODUN ÇALIŞTIRILMASI**

Size ait açılan **/truba/home/egitimXX/** dizininde verilen **Lab2.slurm** betik dosyasını kullanarak **sbatch Lab2.slurm** komutu ile kodunuzu çalıştırabilirsiniz. Oluşacak **Lab2.out** dosyasındaki programın çıktısını **vim Lab2.out** komutu ile görebilirsiniz.