GPU Programlama ve Paralel Hesaplama

Yapay Zekâ ve Veri Dersleri Rehberi · Kısım VI: Sistem ve Uygulama

CPU ve GPU | İş parçacığı modeli | Bellek hiyerarşisi | Paralel desenler | Başarım ölçümü | Komşu dersler

Sercan KÜLCÜ · Giresun Üniversitesi · Bilgisayar Mühendisliği

© Sercan Külcü. Tüm hakları saklıdır. · Sunumun etkileşimli HTML sürümü:
https://sercankulcu.github.io/files/ai_courses/sunumlar/Bolum_37_GPU_ve_Paralel_Hesaplama.html

GİRİŞ

Ders Hangi Soruyu Sorar?

Binlerce çekirdek aynı hesabı nasıl paylaşır?

GPU programlama, aynı işlemi binlerce veri öğesine aynı anda uygulayan **veri paralel** hesaplamayı inceler. Ders CUDA gibi bir model üzerinden iş parçacıklarını düzenlemeyi, bellek hiyerarşisini verimli kullanmayı ve hızlanmayı ölçmeyi öğretir; derin öğrenmeyi pratikte mümkün kılan da bu donanımdır.

Bir bakışta

YAKLAŞIM	GİRDİ	ÇIKTI
veri paralel hesaplama	paralelleştirilebilir algoritma	hızlandırılmış program

BAŞARI ÖLÇÜTÜ	MATEMATİK YÜKÜ	VERİ İHTİYACI
hızlanma, bellek bant genişliği	düşük-orta	gerekmez

Kısa Tarihçe

1966
Flynn bilgisayarları SISD, SIMD, MISD ve MIMD olarak sınıflandırır.

1967
Amdahl, sıralı kalan kısmın paralel hızlanmayı sınırladığını gösterir (Amdahl yasası).

1999
NVIDIA GeForce 256, "GPU" adıyla piyasaya çıkar.

2007
NVIDIA CUDA'yı yayımlar: GPU, C benzeri bir dille genel amaçlı programlanabilir olur.

2012
AlexNet iki GPU üzerinde eğitilip ImageNet'i kazanır; GPU derin öğrenmenin motoru olur.

2017
Volta mimarisi tensör çekirdeklerini getirir: matris çarpımına özel donanım birimleri.

DERS

Temel Kavramlar

SIMT ve warp
32 iş parçacığı aynı komutu birlikte yürütür.

İş parçacığı düzeni
Izgara (grid), blok ve iş parçacığı hiyerarşisi.

Bellek hiyerarşisi
Yazmaç, paylaşılan bellek, global bellek.

Birleşik erişim
Komşu iş parçacıkları komşu adreslere erişmeli.

Eşzamanlama
Bariyer, atomik işlemler ve yarış durumu.

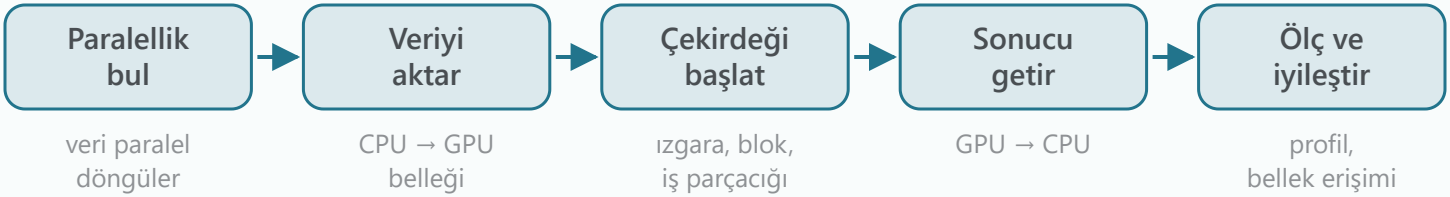
Amdahl yasası
Sıralı kalan kısım hızlanmanın üst sınırını koyar.

Doluluk
Çekirdekleri meşgul tutacak kadar iş parçacığı.

Paralel desenler
Map, indirgeme, önek toplamı, stencil.

DERS

Tipik İş Akışı



Örnek: İki milyon elemanlı iki vektörü toplamak için her elemana bir iş parçacığı atanır: diziler GPU belleğine kopyalanır, 256 iş parçacıklı bloklarla çekirdek başlatılır ve sonuç geri alınır. Ölçüm çoğu zaman sürenin hesapta değil, veri kopyalamada geçtiğini gösterir.

DERS

Derste Neler İşlenir?

1. Paralel mimariler: Flynn sınıflaması, CPU ve GPU
2. CUDA modeli: ızgara, blok ve iş parçacığı
3. Bellek hiyerarşisi ve CPU-GPU veri aktarımı
4. Birleşik erişim ve paylaşılan bellek
5. Warp yürütmesi, dallanma ayrışması, doluluk
6. Eşzamanlama, atomik işlemler ve yarış durumu
7. Paralel desenler: indirgeme ve önek toplamı
8. Stencil, histogram ve seyrek matris işlemleri
9. Döşemeli matris çarpımı ve tensör çekirdekleri
10. Akışlar, asenkron kopyalama ve çoklu GPU
11. Profil çıkarma: Nsight ve roofline modeli
12. CPU'da OpenMP, kümelerde MPI

Kazandırdığı beceriler

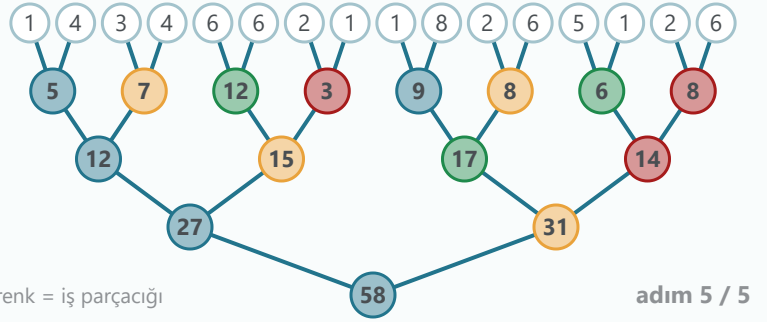
CUDA C çekirdeği yazmak | Nsight ile darboğaz bulmak | Python'da Numba ve CuPy | OpenMP ile CPU paralelliği | hızlanmayı doğru ölçmek

Tipik bir dönemin akışıdır; sıra ve kapsam üniversiteden üniversiteye değişir.

DENE

Binlerce Sayıyı Birlikte Topla

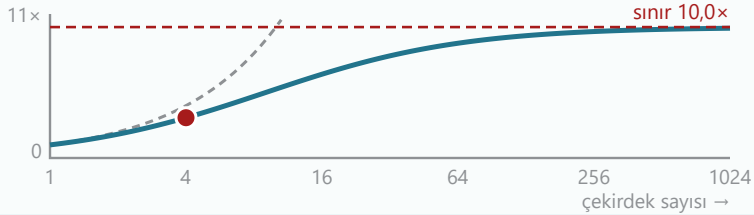
Bir ağın kaybını hesaplamak için milyonlarca sayı toplanır; GPU bunu ağaç biçiminde, çiftleri aynı anda toplayarak yapar. İş parçacığı sayısını değiştirin, adım adım izleyin; Amdahl yasası sınırı gösterir.



renk = iş parçacığı

adım 5 / 5

Amdahl: hızlanma (kesik çizgi: ideal)



İş parçacığı sayısı p = **4**

Programın seri kısmı = **%10**

Adım adım oynat

16 sayı, **4** iş parçacığı: **5 adım** (tek iş parçacığıyla 15). Hızlanma 3,00×.
Amdahl: seri kısım %10 iken 4 çekirdek **3,08×**; sonsuz çekirdekle bile en çok **10,0×**.

Paralel programlama dersi, işi **bağımsız parçalara** bölmeyi öğretir: n sayının toplamı log n adımda biter. Ama seri kalan küçük bir kısım bile hızlanmanın tavanını belirler.

KONUM

Ön Koşullar

Bu dersten önce bilinmesi yararlı olan dersler ve neden gerektikleri.

Ders	Neden gerekir
C ile Programlama	işaretçiler, diziler ve bellek ayırma: CUDA C'nin temeli
İşletim Sistemleri	iş parçacıkları, eşzamanlama ve yarış durumu
Mikroişlemciler	yazmaç, önbellek ve bellek hiyerarşisi

KONUM

Bu Ders Sana Göre mi?

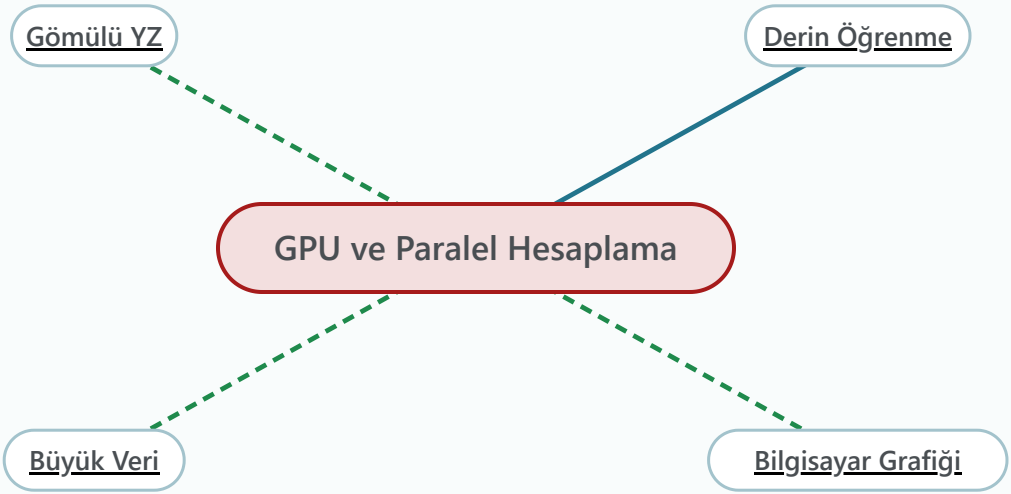
Kendine sor; "hayır" dediklerin için sağdaki sütun nereden başlayacağını gösterir.

Kendine sor	Hayırsa önce
C'de işaretçi ve malloc ile dinamik bir diziyi yönetebiliyor musun?	C ile Programlama dersi
Aynı değişkene yazan iki iş parçacığında yarışı açıklayabiliyor musun?	İşletim sistemlerindeki iş parçacığı ve eşzamanlama
Önbellek ile ana bellek arasındaki hız farkını açıklayabiliyor musun?	Mikroişlemciler ya da bilgisayar mimarisi dersi
İki matrisin çarpımını iç içe döngülerle yazabiliyor musun?	Doğrusal Cebir

Çoğuna "evet" diyebiliyorsan derse hazırsın; bir ya da iki "hayır" dönem içinde kapanabilir.

KONUM

Dersin Komşuları



ÖRTÜŞME

Örtüştüğü Dersler

İlişki	Ders	Ortak konu
temel alır	Derin Öğrenme bu dersi temel alır	büyük ağların paralel eğitimi, tensör işlemleri
yöntem paylaşır	Bilgisayar Grafiği ile yöntem paylaşır	grafik boru hattı, gölgelendiriciler
yöntem paylaşır	Büyük Veri Sistemleri ile yöntem paylaşır	paralel ve dağıtık hesaplama
yöntem paylaşır	Gömülü Yapay Zekâ ile yöntem paylaşır	donanım hızlandırıcılar

AYRIŞMA

Ayrıştığı Noktalar (1/2)

Ölçüt	GPU ve Paralel Hesaplama	Derin Öğrenme	Büyük Veri
Temel soru	Binlerce çekirdek aynı hesabı nasıl paylaşır?	Çok katmanlı ağlar ham veriden kullanışlı temsilleri nasıl kendisi öğrenir?	Tek bir makineye sığmayan veri nasıl saklanır ve işlenir?
Yaklaşım	veri paralel hesaplama	çok katmanlı temsil öğrenme	dağıtık depolama ve hesaplama
Girdi	paralelleştirilebilir algoritma	ham veri: görüntü, metin, ses	çok büyük ve hızlı akan veri
Çıktı	hızlandırılmış program	öğrenilmiş temsil ve tahmin	işlenmiş veri, sorgu sonucu
Başarı ölçütü	hızlanma, bellek bant genişliği	doğruluk, kayıp, ölçeklenme	ölçeklenme, gecikme, maliyet
Matematik yükü	düşük-orta	yüksek: türev, matris, olasılık	düşük
Veri ihtiyacı	gerekmez	çok; GPU gerekir	çok fazla

- **Derin Öğrenme:** Derin öğrenme GPU'yu hazır kütüphanelerle kullanır; bu ders o kütüphanelerin altındaki çekirdekleri ve bellek erişimini yazar.
- **Büyük Veri:** Büyük veri sistemleri işi çok sayıda makineye dağıtır; bu ders tek bir cihazdaki binlerce çekirdeği verimli kullanır.

AYRIŞMA

Ayrıştığı Noktalar (2/2)

Ölçüt	GPU ve Paralel Hesaplama	Bilgisayar Grafiği	Gömülü YZ
Temel soru	Binlerce çekirdek aynı hesabı nasıl paylaşır?	Bir sahnenin tanımından gerçekçi bir görüntü nasıl üretilir?	Yapay zekâ modelleri küçük, düşük güçlü cihazlarda nasıl çalışır?
Yaklaşım	veri paralel hesaplama	geometri ve ışık benzetimi	kaynağı kısıtlı çıkarım
Girdi	paralelleştirilebilir algoritma	sahne: model, ışık, kamera	eğitilmiş model, cihaz kısıtları
Çıktı	hızlandırılmış program	görüntü ya da animasyon	küçük ve hızlı model
Başarı ölçütü	hızlanma, bellek bant genişliği	gerçekçilik, kare hızı	gecikme, enerji, bellek
Matematik yükü	düşük-orta	yüksek: doğrusal cebir, geometri	orta
Veri ihtiyacı	gerekmez	gerekmez	eğitim verisi zaten var

- **Bilgisayar Grafiği:** Bilgisayar grafiği GPU'yu gölgelendiricilerle görüntü çizmek için programlar; bu ders onu genel amaçlı hesap için kullanır.
- **Gömülü YZ:** Gömülü YZ az güçle yetinmeyi öğretir; bu ders en büyük hızlandırıcıdan en yüksek başarımı almayı öğretir.

UYGULAMA

Uygulama Alanları

Derin öğrenme
Matris çarpımı ve evrişim tensör çekirdeklerinde.

Bilimsel benzetim
Akışkanlar, iklim ve moleküler dinamik.

Görüntü ve video
Gerçek zamanlı süzgeçler, kodlama ve çözme.

Finans
Monte Carlo ile risk ve fiyatlama hesabı.

Biyoenformatik
Dizi hizalama ve protein yapısı hesapları.

Veri analitiği
GPU üzerinde tablo ve veri tabanı işlemleri.

UYGULAMA

Riskler ve Etik

Şişirilmiş hızlanma

Veri aktarımını dışarıda bırakmak ya da zayıf bir CPU sürümüyle kıyaslamak, hızlanmayı olduğundan büyük gösterir.

Sessiz yarış hataları

Eşzamanlaması eksik bir çekirdek çoğu çalıştırmada doğru, bazen yanlış sonuç verir; hata zor yakalanır.

Kayan nokta farkları

Paralel toplamada işlem sırası değişir; sonuçlar CPU'dakinden ve çalıştırmadan çalıştırmaya biraz farklı çıkar.

Enerji ve maliyet

Büyük modelleri eğiten GPU kümeleri çok elektrik tüketir; verimsiz kod bu bedeli boşuna büyütür.

Ayrıntılı tartışma: Açıklanabilir Yapay Zekâ ve Yapay Zekâ Etiği.

UYGULAMA

Araçlar ve Örnek Veriler

Araçlar ve kütüphaneler

CUDA | OpenCL | OpenMP | MPI | Numba | CuPy | NVIDIA Nsight

Klasik örnek problemler

Vektör toplama | Matris çarpımı | Paralel indirgeme | Önek toplamı (tarama) | Histogram | N-cisim benzetimi

Ölçüt **hızlanmadır**: aynı problemin sıralı CPU sürümü ile paralel GPU sürümü aynı girdide karşılaştırılır.

UYGULAMA

Sık Karıştırılanlar

Yaygın düşünce	Doğrusu
GPU her programı hızlandırır.	Yalnız çok sayıda bağımsız ve benzer işlem içeren kısımlar hızlanır; dallanan ve sıralı kısımlar hızlanmaz.
Çekirdek sayısı iki katına çıkınca süre yarıya iner.	Amdahl yasası: sıralı kalan kısım, elde edilebilecek hızlanmanın üst sınırını belirler.
En pahalı iş hesaptır.	Çoğu GPU programında darboğaz bellek bant genişliği ve CPU ile GPU arasındaki veri aktarımıdır.
Paralel programlama yalnız derin öğrenme içindir.	Benzetim, grafik ve bilimsel hesap GPU'yu derin öğrenmeden önce de kullanıyordu.

SONRASI

Bu Dersten Sonra

Bu dersi temel alan ya da genişleten dersler

Derin Öğrenme

İlgili meslekler

GPU yazılım mühendisi

Yüksek başarımlı hesaplama (HPC) uzmanı

MÖ altyapı mühendisi

Oyun ve grafik motoru geliştiricisi

Bilimsel hesaplama araştırmacısı

GPU programlama, derin öğrenmenin ve grafiğin altındaki motoru açar. Buradan dağıtık eğitim, derleyiciler ve hızlandırıcı tasarımı gibi sistem konularına ya da büyük veri sistemleri ve gömülü YZ gibi rehberdeki komşu derslere geçilebilir.

SONRASI

Kaynaklar

- Kirk, Hwu ve El Hajj, *Programming Massively Parallel Processors*, 4. baskı, Morgan Kaufmann, 2022.
- Sanders ve Kandrot, *CUDA by Example*, Addison-Wesley, 2010.
- Hennessy ve Patterson, *Computer Architecture: A Quantitative Approach*, 6. baskı, Morgan Kaufmann, 2017.
- Amdahl, "Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities", *AFIPS Conference Proceedings* 30, 1967.

Son

Bölüm 37: GPU Programlama ve Paralel Hesaplama

← Önceki bölüm · Alanın haritası · Sonraki bölüm →