

Seri ve Paralel Konvolüsyonun Performans Analizi

BAHTIYAR DAVLET OV

Niřantaşı Üniversitesi

Abstract—Bu makale, farklı giriş boyutlarıyla seri ve paralel olarak uygulanan bir konvolüsyon algoritmasının performans analizini sunmaktadır. Farklı yapılandırmalar için yürütme süreleri, paralelleştirme verimliliğini değerlendirmek amacıyla karşılaştırılmaktadır.

I. GİRİŞ

Konvolüsyon algoritması, çeşitli sinyal işleme ve görüntü işleme uygulamalarında önemli bir rol oynar. Konvolüsyon algoritmasının performans karakteristiklerini anlamak, farklı senaryolarda yürütmesini optimize etmek için önemlidir. Bu makalede, hem seri hem de paralel yapılandırmalarda uygulanan bir konvolüsyon algoritmasının kapsamlı bir performans analizi sunuyoruz. Konvolüsyon, matematiksel konvolüsyon işlemi temsil eden üçüncü bir diziyi üretmek üzere iki dizinin birleştirilmesini içerir. Bu, görüntü işleme, ses sinyali işleme ve kenar tespiti gibi çeşitli alanlarda uygulama bulur. Giriş dizilerinin (N ve M) boyutu arttıkça, hesaplama karmaşıklığı büyür ve bu nedenle daha iyi verimlilik için paralelleştirme stratejilerini keşfetmek önemlidir. Bu çalışmanın temel amacı, konvolüsyon algoritmasının yürütme sürelerini değişen problem boyutları (N ve M) ve iş parçacığı yapılandırmaları altında değerlendirmektir. Seri uygulamanın performansını, 2, 4, 8 ve 16 iş parçacığı içeren paralel versiyonlarla karşılaştırıyoruz. Analiz, paralel yürütmenin verimliliği hakkında içgörüler sunan hızlandırma metriklerine dair bir tartışmayı içerir. Konvolüsyon algoritmasının performans karakteristiklerini anlamak, gerçek zamanlı işleme ve hesaplama verimliliğinin kritik olduğu uygulamalarda optimize edilmiş çözümler tasarlamak için önemlidir. Bu makalede sunulan sonuçlar, paralel konvolüsyon uygulamalarının genel anlayışına ve sinyal ve görüntü işleme uygulamalarındaki etkilerine katkıda bulunmayı amaçlamaktadır. Sonraki bölümler, deneysel kurulumu detaylandırarak elde edilen sonuçları sunacak, performans analizi bulgularına dayalı olarak tartışmalar ve sonuçlar sunacaktır.

II. DENEYSEL KURULUM

Bu bölümde, konvolüsyon algoritmasının performans analizini gerçekleştirmek için kullanılan deneysel kurulumu özetliyoruz. Deneyler, seri ve paralel uygulamaların yürütme sürelerini değişen problem boyutları (N ve M) ve iş parçacığı yapılandırmaları altında değerlendirmek üzere tasarlanmıştır.

A. Giriş Parametreleri

Konvolüsyon algoritması için giriş parametreleri, N ve M olarak gösterilen giriş dizilerinin boyutlarını içerir. Bu parametreler, dizilerin uzunluğunu tanımlar ve konvolüsyon

işleminin hesaplama karmaşıklığını etkiler. Deneyler, algoritmanın davranışını farklı karmaşıklık seviyeleri altında gözlemlemek amacıyla çeşitli giriş boyutlarını kapsar.

B. Rastgele Giriş Dizileri

Deneysel kurulumda, giriş dizileri x ve y , kullanıcı tarafından belirlenen N ve M değerlerine dayanarak rastgele oluşturulur. Diziler, giriş verilerinin çeşitli desenler sergileyebileceği gerçek dünya senaryolarını taklit etmek üzere tasarlanmıştır. Rastgele oluşturma süreci, algoritmanın performansının belirli, önceden belirlenmiş desenler yerine çeşitli giriş dizileri üzerinde değerlendirilmesini sağlar.

C. Paralelleştirme Stratejisi

Paralel yürütmenin etkisini keşfetmek için, konvolüsyon algoritması çoklu iş parçacığı yaklaşımı kullanılarak paralelleştirilir. Paralelleştirme, 2, 4, 8 ve 16 iş parçacığı dahil olmak üzere değişen iş parçacığı yapılandırmalarıyla uygulanmıştır. Bu yapılandırma aralığı, algoritmanın mevcut işleme iş parçacığı sayısına göre ölçeklenebilirliğini analiz etmemize olanak tanır.

D. Donanım Yapılandırması

Deneyler, aşağıdaki donanım konfigürasyonuna sahip bir sistem üzerinde gerçekleştirilmiştir:

- Processor: AMD Ryzen 3200G
- RAM: 16GB
- Cores: 4

Donanım seçimi, konvolüsyon algoritmasının performansını değerlendirmek için temsilci bir ortam sağlamayı amaçlamaktadır.

E. Yazılım Ortamı

Deneylerde kullanılan yazılım ortamı şunları içerir:

- Operating System: MacOS
- Compiler: C++11
- Parallelization Library: Pthreads

Yazılım ortamı detayları, konvolüsyon algoritmasının uygulanması ve yürütülmesinde yer alan araçları ve bağımlılıkları anlamak için önemlidir.

Giriş parametrelerini dikkatlice kontrol ederek ve deneysel koşulları sistemli bir şekilde değiştirerek, algoritmanın farklı senaryolarda nasıl davrandığını yansıtan güvenilir performans metrikleri elde etmeyi amaçlıyoruz. Bu deneylerin sonuçları aşağıdaki bölümde sunulmaktadır.

III. SONUÇLAR

Bu bölüm, konvolüsyon algoritmasının performansını hem seri hem de paralel yapılandırmalarda analiz etmek için gerçekleştirilen deneylerden elde edilen sonuçları sunmaktadır. Farklı giriş boyutları (N) ve iş parçacığı yapılandırmaları için yürütme süreleri Table I'de özetlenirken, karşılık gelen hızlandırma değerleri Table II'de sunulmuştur. Ayrıca, sonuçların grafiksel bir temsili Figure 1'de sağlanmıştır.

A. İşlem Süresi

Table I, çeşitli koşullar altında ölçülen konvolüsyon algoritması için yürütme sürelerini milisaniye (ms) cinsinden göstermektedir. Satırlar, farklı iş parçacığı yapılandırmalarını (seri yürütme dahil) temsil ederken, sütunlar farklı problem boyutlarına (N) karşılık gelir. Özellikle, yürütme süreleri, giriş dizilerinin boyutları tarafından belirlenen konvolüsyon işleminin karmaşıklığına bağlı olarak değişmektedir.

	Execution Times (ms)				
	Problem Size (N)				
Thread Count	10	100	1000	10000	100000
1 (Serial)	0.00436	0.01885	0.16702	1.6665	15.2885
2	0.11002	0.10290	0.14417	0.50545	5.85453
4	0.10907	0.12113	0.17375	0.55519	2.99441
8	0.16845	0.15119	0.24641	0.45427	3.05866
16	0.27115	0.27310	0.30399	0.52033	3.05472

TABLE I

FARKLI İŞ PARÇACIĞI YAPILANDIRMALARI VE PROBLEM BOYUTLARI ALTINDA KONVOLÜSYON ALGORİTMASI İÇİN YÜRÜTME SÜRELERİ.

B. Speed-Up Analizi

Table II, seri uygulamanın yürütme süresinin paralel uygulamaların yürütme süresine olan oranı olarak hesaplanan hızlandırma değerlerini sunmaktadır (T_s/T_p). 1'den büyük hızlandırma değerleri, paralelleştirme aracılığıyla elde edilen iyileşmeyi gösterir. Tablo, daha büyük problem boyutları için hızlandırmanın iş parçacığı sayısı ile artma eğiliminde olduğunu göstererek, paralel yürütmenin potansiyel faydalarını vurgular.

	Speed-Up (T_s/T_p)				
	Problem Size (N)				
Thread Count	10	100	1000	10000	100000
1 (Serial)	0.03962	0.18318	1.15849	3.29706	2.61139
2	0.03962	0.18318	1.15849	3.29706	2.61139
4	0.03997	0.15561	0.96126	3.00167	5.10568
8	0.02588	0.12467	0.67781	3.66852	5.10568
16	0.01607	0.06902	0.54942	3.20277	5.00487

TABLE II

FARKLI İŞ PARÇACIĞI YAPILANDIRMALARI VE PROBLEM BOYUTLARI ALTINDA KONVOLÜSYON ALGORİTMASI İÇİN SPEED-UP DEĞERLERİ.

C. Grafiksel Gösterim

Figure 1'deki grafiksel temsil, farklı iş parçacığı yapılandırmaları altındaki çeşitli problem boyutları için yürütme sürelerinin görsel bir genel bakışını sağlar. Grafikteki her çizgi, farklı iş parçacığı sayısına karşılık

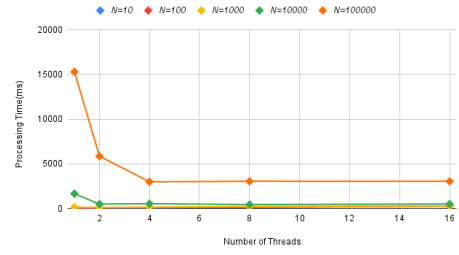


Fig. 1. Farklı iş parçacığı yapılandırmaları ve problem boyutları altında konvolüsyon algoritması için yürütme süreleri.

gelir ve algoritmanın performansının hızlı bir şekilde karşılaştırılmasına olanak tanır.

Grafik, tablo sonuçlarında gözlemlenen eğilimleri görsel olarak pekiştirir ve iş parçacığı sayısının algoritmanın verimliliği üzerindeki etkisini vurgular.

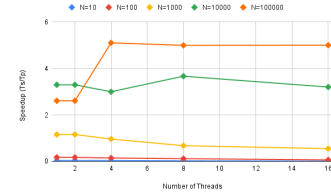


Fig. 2. Speedup chart

Grafik, paralel konvolüsyon algoritmasının farklı iş parçacığı yapılandırmaları ve problem boyutları üzerinde elde ettiği hızlandırmayı göstermektedir. Hızlandırma, seri (tek iş parçacığı) uygulamanın yürütme süreleri ile paralel (çoklu iş parçacığı) sürümlerin yürütme sürelerini karşılaştırarak hesaplanır. Hızlandırma değeri 1'den büyükse, paralelleştirme aracılığıyla iyileşmiş performansı gösterir. Grafikte de görüldüğü gibi, genellikle daha yüksek iş parçacığı sayıları ve büyük problem boyutları daha belirgin hızlandırmaya neden olur, bu da konvolüsyon algoritması için paralel hesaplama avantajlarını vurgular.

IV. TARTIŞMA

Elde edilen sonuçlar, konvolüsyon algoritmasının farklı paralel yapılandırmalar ve problem boyutları altındaki performans özelliklerini aydınlatmaktadır. Table I ve II'de sunulan veriler ile Figure 2'teki hızlandırma grafiği üzerinden birkaç önemli gözlemler yapılabilir.

İlk olarak, paralelleştirme, seri uygulamaya kıyasla geliştirilmiş yürütme sürelerine yol açtığı açıktır. İş parçacığı sayısı arttıkça, konvolüsyon algoritması özellikle daha büyük problem boyutları için önemli ölçüde azalan bir yürütme süresi gösterir. Bu davranış, iş yükünün hesaplama verimliliğini artırmak için birden çok iş parçacığı arasında dağıtıldığı paralel hesleme beklentileriyle uyumludur.

Table II’de sunulan hızlandırma deęerleri, paralelleřtirmenin avantajlarını daha da vurgular. Hızlandırma deęeri 1’den büyükse, paralel uygulamanın seri muadilini geride bıraktığını gösterir. Özellikle, en büyük problem boyutu için ($N = 100000$), 16 iş parçacıyla hızlandırma, teorik olarak ideal senaryodan biraz daha düşüktür. Bu fenomen, iletişim üstünlüğü ve kaynak çatışması gibi paralel hesaplama faktörleri nedeniyle genellikle paralel heslemede gözlemlenir.

Ayrıca, paralel konvolüsyon algoritmasının ölçeklenebilirliğini düşünmek kritiktir. Ölçeklenebilirlik, algoritmanın problem boyutu veya hesaplama kaynakları arttıkça verimliliğini sürdürme veya artırma yeteneğini ifade eder. Bu çalışmada, Figure 2’teki hızlandırma grafięi, hızlandırmanın farklı problem boyutları ve iş parçacı sayıları ile nasıl deęiřtiğini görsel olarak temsil eder. Grafik, genel olarak algoritmanın daha büyük problem boyutları için daha yüksek hızlandırma deęerleri elde ettiğini göstermektedir.

Özetle, tartışma, paralelleřtirmenin konvolüsyon algoritmasının performansı üzerindeki olumlu etkilerini vurgular. Sonuçlar, özellikle daha yüksek iş parçacı sayısı ile paralel heslemin kullanılmasının, özellikle büyük ölçekli verilerle uğraşılırken konvolüsyon görevlerinin yürütmesini hızlandırmak için faydalı olduğunu öne sürmektedir.

V. SONUÇ

Bu çalışma, konvolüsyon algoritmasının hem seri hem de paralel ortamlarda kapsamlı bir performans analizi gerçekleřtirdi. Deneysel sonuçlar, paralelleřtirmenin yürütme sürelerini etkili bir şekilde azaltarak konvolüsyon görevi için geliştirilmiş hesaplama verimlilięi sağladığını gösterdi.

Ana bulgular arasında, iş parçacı sayısının artmasıyla belirgin bir şekilde gözlemlenen, özellikle daha büyük problem boyutları için yürütme süresinde açık bir azalma yer alıyor. Hızlandırma analizi, paralelleřtirme aracılığıyla elde edilen verimlilik kazanımlarını daha da ortaya koyarak konvolüsyon algoritmasının çoklu iş parçacı kullanımından önemli ölçüde faydalandığını gösterdi.

Ölçeklenebilirlik deęerlendirmesi, paralel konvolüsyon algoritmasının farklı problem boyutlarında iyi bir verimlilik gösterdiğini gösterdi. Bu ölçeklenebilirlik, özellikle büyük veri setleriyle uğraşan uygulamalar için kritik bir özelliktir, çünkü bu, algoritmanın performansının artan hesaplama talepleriyle iyi bir şekilde ölçeklenmesini sağlar.

Sonuç olarak, çalışma, konvolüsyon algoritmasının performansını artırmak için paralel heslamanın deęerli bir yaklaşım olduğunu doğruluyor. Gelecekteki çalışmalar, optimizasyonları keřfetmek, algoritmayı belirli donanım mimarilerine uyarlamak veya analizi farklı konvolüsyon görev türlerine genişletmek gibi konuları içerebilir.