Paralel Programlama Ortamları

Elis Soylu
ESKİŞEHİR OSMANGAZİ ÜNİVERSİTESİ
esoylu@ogu.edu.tr

AB Konferansı Şubat' 2015

Paralel Programlama Ortamları

- Paralel Programlama Tanımı
- İş Parçacığı Tabanlı Yöntemler
  - Java Thread ile Paralel Programlama
  - CUDA ile paralel programlama
  - PThread ile Paralel Programlama
  - OpenMP ile Paralel Programlama
  - Haskell ile Paralel Programlama
- Dağıtık Yöntemler
  - MPI ile Paralel programlama
- Karakteristik Özellikler
- Performans ve Analizler

Paralel Programlama

Paralel programlama uygulandığı biçimler bakımından aşağıdaki farklılıkları içerir.

- Paralelliğin hangi kısımda yapıldığı
- Paralel program kısımlarının tanımlanma şekli
- İşlemcilerin haberleşmesi
- Paralel kısımların eşzamanlı çalışma prensipleri [1]

Bu farklılıklar ile paralel programlama işleyişi iki ana grupta toplanabilir.

- İş parçacığı Tabanlı Yöntemler
- Dağıtık Yöntemler [2]

İş parçacığı Tabanlı Yöntemler

AB Konferansı Şubat’ 2015

Java Thread ile Paralel Programlama

- **Thread** : Bir programın birbirinden bağımsız işlem parçalarını yürütmekle görevli en küçük parçasıdır.

- Birçok programlama dilinde **thread** yapıları kütüphaneler yardımıyla oluşturulur. Bu kütüphanelerden kullanılışı açısından **Java Thread** uygulama ara yüzü en basit kullanıma sahip olanıdır.

- **Thread**'ler arası senkronizasyon ve haberleşme işlerinin dengeli olabilmesi için **mutex, lock, barriers** ve **deadlock** gibi metotlara ihtiyaç duyulur [3].

CUDA ile paralel programlama

- Cuda, Nvidia firması tarafından üretilen grafik işlemci birimi kullanılarak paralel programlama yapılan mimari çeşididir.

- CUDA işlemciyi kullanarak mimari yardımıyla **kernel** adı verilen programın serileştirilmesiyle paralel yapıya ulaşılır [4].

- **Thread** denilen yapılar uygun sayıda bir araya gelerek **warp** denilen yapıları, warp'lar birleşerek **blok**'ları, blok'lar birleşerek **grid** denen yapıları meydana getirirler [5].

PThread ile Paralel Programlama

- Birçok çok çekirdekli işlemci **thread**'ler için bir POSIX standardı olarak bilinen **POSIX THREAD** (PThread) yapısını kullanır. PThread yapısı, C programlama dilinin tipleri, fonksiyonları ve sabitlerini içeren bir kütüphanedir.

- '**pthread**' ön eki kullanılmak üzere yaklaşık 100 civarında PThread prosedürü bulunmaktadır. Bu prosedürler 4 ana başlık altında toplanabilir:

  - Thread yönetimi (thread oluşturma, birleştirme,…)
  - Mutex yapıları
  - Koşul değişkenleri
  - Senkronizasyon

- **pthread_create, pthread exit, pthread cancel, pthread attr init, pthread attr destroy** en çok kullanılan pthread fonksiyonlarıdır [6].

OpenMP ile Paralel Programlama

- OpenMP, shared memory mimarisini C, C++, Fortran dillerinde destekleyen bir ara yüzdür.
- OpenMP kütüphanesi taşınabilir bir özellik sunması açısından bilgisayarcılar için paralelleştirmede önemli bir yöntemdir.
- Ompgetthreadnum() fonksiyonu ile her thread kendine özgü işi yapar.
- Hem görev hem de veri paralelliği açısından OpenMP önemlidir [7].

Haskell ile Paralel Programlama

- Haskell, GHC isimli derleyiciye sahip bir fonksiyonel programlama dilidir. Çok çekirdekli mimariler için geniş kütüphane seçeneğiyle oldukça etkilidir [8].
- Haskell dili, paralelliği **SMP** (shared-memory multi-processed) üzerinde gerçekleştirilir.
- **Thread** denilen bağımsız yapılarla paralel olarak işlemciler üzerinde istenilen işlemler kolaylıkla gerçekleştirilir.
- **Eden** ve **GpH**, Haskell dilinin kullanıcıya görev paralelleştirmesini ve verileri ifade etmesini sağlayan önemli türevlerinden biridir [9].

Dağıtık Yöntemler

AB Konferansı Şubat’ 2015

MPI ile Paralel programlama

- Temel işleyişi, ayrık adres uzaylarına sahip işlemciler arasında mesaj alıp - verme yöntemi şeklindedir.

- İşlemciler arasında haberleşmeyi sağlamak için **MPI_COMM_WORLD** komutuyla bir ortam oluşturulur. Bununla birlikte,

- MPI_Init
- MPI_Finalize
- MPI_Comm_Size
- MPI_Comm_Rank
- MPI_Send
- MPI_Receive

- en sık kullanılan MPI komutlarıdır [10].

Karakteristik Özellikler

- **Java** dilinin önemli bir kısmını aşırı yüklenmiş metot yapısı oluşturur. Java dili genel olarak nesnelerle program altyapısını oluşturur. En önemli özelliği platformdan bağımsız olmasıdır [11].

- **Haskell** dili, dilin tasarımı gereği metot ile tip tanımını birbirinden ayırır. Tip sınıfı ile objenin kullanacağı kurallar belirlenerek bu tanımlama yapılır. Bir fonksiyonel programlama dili olarak Haskell, polimorfik olması ve yan-etkisiz çalışması ile diğer dillerden ayrılır [12].

- **MPI**, yapısı gereği hata kontrolünü yapması ve okunabilirliğinin kolay olması ile açıkça ifadesi mümkündür. Görev paralelleştirme metodunu kullanır.

Karakteristik Özellikler

- **CUDA**, mimariye uygun şekilde tekrar düzeltilebilir yapısı vardır. Class-metot'lar yardımıyla yeni fonksiyonlar üretebilir ve kolayca kullanılabilir [13].

- **PThread**, çok işlemcili bir bilgisayarda birçok thread yapısını aynı anda çalıştırır. Thread'ler birbirini engellemeyecek düzende iş yapar.

- **OpenMp** ise, taşınabilir ve ölçeklendirilebilir olması açısından paralelliğe uygundur.

Bu karakteristik özellikler, matris çarpımı algoritması ile 1000x1000 tipli bir matriste performans incelemesini gösteren tablo aşağıdadır.

Programlama Dilleri	Donanım	Süre	Hızlanma
OpenMP	Intel Core 2 Quad	3.802s	3.320x
Java Thread	Intel Xeon Dual Core	8.73s	7x
CUDA	GeForce GTX 280 SC	0.25s	2.56x
MPI	Intel Core 2 Quad	0.111s	5.70x
PThread	8-process Sun Ent. 2 GB M.	0.056s	0.9371x
Haskell	Intel Core i-7 6 GB M	7.2s	3.8x

Sonuçlar ve Analizler

Bu tabloda matris çarpımı algoritması, blok parçalama yöntemiyle oluşturulmuştur. Bu algoritmanın karmaşıklığı N, işlemci sayısı olmak üzere seri olarak O(N), paralel olarak O(log N) dır.

- Elde edilen performanslardan **OpenMP** platformu ile 4 kere çalıştırılıp ortalama süre hesaplanmıştır [14].
- **Java** için, Java Hotspot JIT in Version 1.6.0_13 ortamında test edilmiştir [15].
- **CUDA** platformu ise belirtilen GeForce işlemcisinde test edilmiştir [16].
- **MPI** ile elde edilen sonuçlar matristeki blokların parçalanması yöntemiyle elde edilmiştir [17].
- **PThread** için diğerlerinden farklı bir makinede yüksek bir performans elde edilmiştir [18].
- **Haskell** platformunda GpH modülü yardımıyla elde edilen her thread yapısına bir blok karşılık gelecek şekilde sonuç elde edilmiştir [19].

Sonuç ve Öneriler

- MPI, Java Thread, CUDA, Haskell, PThread, OpenMP platformlarının paralellik ile gösterdiği karakteristiksel özellikler temel özellikleriyle birlikte incelenmiştir.
- MPI ve PThread platformları paralellik açısından daha gelişmiş bir ortama sahiptir.
- Matris çarpımında blok parçalama yöntemiyle incelenen platformlarda derlenerek performanslarındaki farklılıklar gözlenmektedir.
- Algoritmaya uygunluğu açısından etkin seçilen bir platform nanosaniye derecesinde de olsa daha hızlı sonuç verebilir.

Sorular?

Kaynaklar

- [1] Erarslan, G., "Paralel Programlama ve MPI", http://seminer.linux.org.tr/wp-content/uploads/ParalelProgramlamaveMPI.pdf [Son Erişim Tarihi:13 Aralık 2014].
- [2] Ergün, U., Sayar, A., "Fonksiyonel Programlama Dilleri ile Paralel Programlama", Niğde Üniversitesi Mühendislik Bilimleri Dergisi, Cilt 3, Sayı 2, 1-17, 2014.
- [3] Meyer, B., Pedroni, M., "Concurrent Programming with Java Threads", Software Architecture, 2010, http://se.inf.ethz.ch/old/teaching/2010-S/0050/slides/13_softarch_self_study_threads.pdf [Son Erişim Tarihi: 10 Aralık 2014].
- [4] Akcay, M., Sen, B., Orak, I., M., Celik, A., "Paralel Hesaplama ve CUDA", 6th International Advanced Technologies Symposium 2011 (IATS'11), June 2011.
- [5] Akaydın, B., Tunçel M., “Introduction to CUDA”, ITU Cuda education, 2013.
- [6] Prasad J., “Shared Memory Programming with pthreads”, Inter-University Centre for Astronomy & Astrophysics Pune, India 2012.
- [7] en.wikipedia.org/wiki/OpenMP [Son Erişim Tarihi: 18 Aralık 2014].
- [8] https://www.haskell.org/haskellwiki/Haskell_for_multicores [Son Erişim Tarihi: 10 Aralık 2014].
- [9] Coutts, D., Löh, A., “Modern Programming languages”, Copublished by the IEEE CS and the AIP, 12, 1521-9615, 2012.

- [10] https://computing.llnl.gov/tutorials/mpi [Son Erişim Tarihi: 11 Aralık 2014].
- [11] Çay T., İşcan F., "Harita Mühendisliğinde Kullanılan Programlama Dilleri ve Yazılımları", Akademik Bilişim'02, 2002.
- [12] Sabel, D., Schmidt-Schauß, M., "Conservative Concurrency Haskell", 27th Annual ACM/IEEE Symposium on Logic in Computer Science, 2012.
- [13] Bhardwaj D., "Parallel Computing", Indian Institute of Technology, Delhi –110 016 India, 2002.
- [14] Chowdhury, R., "Parallel Computing with OpenMP to solve matrix Multiplication", UCONN BIOGRID REU Summer, 2010.
- [15] Häuser, J. , "A Test Suite for High-Performance Parallel Java", Center for
- Advanced Computing Research, 2000.
- [16] Dotzler, G., "JCudaMP: OpenMP/Java on CUDA", Programming Systems Group, 2010.
- [17] Quinn, M., "Parallel Programming in C with MPI and OpenMP", International Edition, 2003.
- [18] http://www.cs.cmu.edu/~scandal/papers/sc98/ [Son Erişim Tarihi: 18 Aralık 2014].
- [19] Loidl, H-W., "Comparing Parallel Functional Languages : Programming and Performance", Journal Higher-Order and Symbolic Computation, Kluwer Academic Publishers, Volume 16 Issue 3, 203-251, 2003.

ELİS SOYLU
esoylu@ogu.edu.tr

MUAMMER AKÇAY
muammer.akcay@dpu.edu.tr