

Intel Parallel Studio ile Paralel Hesaplama

Muammer Akçay¹, Hüseyin Aşkın Erdem²

¹ Dumlupınar Üniversitesi, Bilgisayar Mühendisliği Bölümü, Kütahya

² Hava Harp Okulu, Havacılık ve Uzay Teknolojileri Enstitüsü, Bilgisayar Mühendisliği ABD., İstanbul

makcay26@gmail.com, herdem@hho.edu.tr

Özet: Bu çalışmada temel olarak paralel hesaplama konusu ele alınmıştır. Niçin paralel hesaplama kullanılması gerektiği, paralel hesaplamaların sağladığı avantajlar ve dezavantajlar açıklanmıştır. Paralel hesaplama eğitiminde kullanılacak programlardan Intel Parallel Studio ve bu programın kullanımına yer verilmiştir. Ayrıca paralel hesaplama kütüphaneleri ve uygulamaları konuları da incelenmiştir. Paralel Hesaplama eğitiminde Intel Parallel Studio kullanımı gerçekleştirilerek uygulamalar ile desteklenmiştir.

Anahtar Sözcükler: Paralel Hesaplama, Matris Çarpımı, Intel Parallel Studio.

Abstract: In this study, parallel computing subject basically examined. Why we should use parallel computing, advantages and disadvantages of parallel computing are explained. Intel Parallel Studio, a program that can be used in parallel computing education is placed in this article. Also in this work, parallel computing libraries and applications are examined. Utilization of “Intel Parallel Studio” in Parallel Computing is implemented in this work and supported by applications.

Keywords: Parallel Computing, Matrix Multiplication, Intel Parallel Studio.

1. Giriş

Gordon Moore'un (Intel'in [11] kurucularından) öne sürdüğü Moore Yasası'na göre, “mikroişlemcilerin içinde bulunan transistor sayılarının her iki yılda bir iki katına çıkmasıyla, işlemci hızlarının iki kat artacağı” varsayılmaktadır [1]. Ancak bu şekilde bir yaklaşım göstermektedir ki günümüz bilgisayarlarında, transistor sayıları fazlaştıkça, bilgisayarlarda hız artışının yanında yüksek güç tüketimi ve ısı artışı da olmaktadır. Bu gibi durumları engellemek için yeni nesil bilgisayarın neredeyse tamamına yakını çok çekirdekli (iki veya daha fazla) olarak üretilmektedir. Bu sayede hız artışı sağlanırken oluşan yüksek ısı artışı da önlenmeye çalışılmaktadır.

üzerindeki etkisidir. Çekirdek sayısı arttıkça işlemlerin veya çalıştırılan uygulamaların farklı çekirdeklere atanarak icra edilmesiyle bilgisayarın hızında, ısı artışı da önlenerek önemli derecede iyileşme sağlanabilmektedir. Bu iyileşme ise çekirdeklerin etkili bir şekilde kullanılmasıyla elde edilen “performans artışı” olarak ifade edilebilir. Yeni çekirdeklerin eklenmesi ile performans artışının sağlanabilmesi temel sorun haline gelmiştir. Programlar, eğer sadece uygun metotları ve komutları içeriyorlarsa diğer çekirdeklerin kullanılmasını sağlayarak performansta artış sağlayabilmektedir. Uygun metotları yazmak için ise kullanılacak en etkili yöntemlerin başında, işlemleri veya hesaplamaları “paralel” olarak kodlamak gelmektedir. “Paralel” olarak kodlanmış bir programda eğer paralel komutlar doğru şekilde ve doğru yerlerde

Çekirdek artışından sağlanan avantajların en önemlilerinden birisi bilgisayarın performansı

kullanılmışlarsa, program, işlemleri istenilen sayıda çekirdekte veya çekirdeklerin tümünde eş zamanlı olarak icra ederek (paralel hesaplama yaparak) sonucu, seri yapılan bir hesaplama göre daha hızlı bir şekilde elde edebilir.

Bir algoritmanın paralelleştirilmesinin temel amacı, çoklu işlemci çekirdeklerinin avantajlarından en etkili ve doğru şekilde yararlanabilmektir [2]. Çok büyük veriler kullanan ve çok uzun hesaplama zamanı gerektiren uygulamalarda, hesaplama süresinin kısaltılması için en etkili yöntemlerin birisi, işlemleri çoklu çekirdeklere paralel olarak paylaşmaktır [3]. Böylece hesaplama süresinin kısaltılması ve dolayısıyla performans artışı hedeflenmektedir.

En yalın tanımı ile paralel hesaplamada, icra edilecek olan görev (program, uygulama, komut kümesi...) belirli kıstaslara göre parçalar halinde gruplara bölünür ve her grup ayrı bir görev parçası olarak işleme alınır. Her grupta senkronize olarak elde edilen sonuçlar, ortak bir havuzda (bilgisayar veya bellek birimi...) toplanarak hesaplama sonuçlandırılır. Buradaki amaç, yapılan görevin bölünerek, paralel olarak parça parça hesaplatılmasıyla, seri yapılan bir hesaplama göre sonuca daha kısa bir sürede ulaşabilmektir.

Günümüzde hesaplamalar ve işlemler çok büyük ölçekli veriler üzerinden yürütüldüğü için paralel hesaplama ve programlama yaklaşımları bu veri hesaplamalarının gerçekleştirilmesinde son derece önemli bir rol oynamaktadır [18].

Paralel hesaplama ile çoklu çekirdek kullanımının sağladığı avantajlar arasında; hesaplama süresinin kısaltılmasıyla hız artışının sağlanması, çok büyük verilerin kullanıldığı problemlerin çözülebilmesi, eşzamanlı çalışma gerçekleştirilebilmesi, ısı artışının önlenmesi ve daha düşük maliyetli kullanım ortamları sağlanması sayılabilir [4].

Paralel hesaplamanın kullanıldığı yoğun veri içeren uygulamalar; hava tahminleri, okyanus modellemeleri, uçak tasarımları, simülasyonlar, büyük ticari veritabanları şeklinde sıralanabilir [4]. Bunların yanında dijital şehir uygulamaları için de paralel hesaplama yöntemi kullanılmaktadır [5].

Bu çalışmada, Intel Parallel Studio 2008 [12] programı (Programın 30 günlük deneme sürümü kullanılmıştır) kullanılarak bir "matris çarpım işlemi" örneği oluşturuldu. 1000x1000'lik iki tane matris oluşturularak, seri ve paralel olarak işlem sürelerini görebilmek amacıyla bu matrisler rastgele (*random*) sayı generatörüyle doldurulup ayrı ayrı dosyalara yazdırıldı ve sonrasında bu matrisler, yazdırıldıkları bu dosyalardan tekrar okutularak çarpıldı ve sonuç başka bir dosyaya yazdırıldı. Böylelikle "Intel Parallel Studio 2008" ortamında, seri ve paralel olarak çalıştırılan matris çarpım programının hesaplanma süreleri ölçülerek, seri ve paralel hesaplamının performansa etkisi gözlemlendi.

İkinci bölümde paralel hesaplama kavramının tanımlanması, kullanılabilecek kütüphane ve ortamlar ile hesaplamaların nasıl paralelleştirilebileceği açıklanmış, üçüncü bölümde paralel hesaplamının artı ve eksi yönleri değerlendirilmiştir. Dördüncü bölümde paralel hesaplama programlarından Intel Parallel Studio tanıtılmış, beşinci bölümde bu program ile gerçekleştirilen bir matris örneği incelenmiştir. Altıncı bölümde sonuçlar değerlendirilmiştir.

Bir sonraki bölümde hesaplamaların paralel yapıda nasıl gerçekleştirilebileceği genel olarak açıklanmıştır.

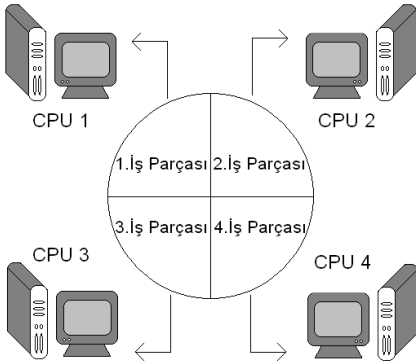
2. Hesaplamanın Parallelleştirilmesi

Hesaplanması istenen işlem veya problem, şekil 1'de gösterildiği gibi öncelikle hesaplama yapacak birim (bilgisayar veya CPU) sayısı kadar iş parçalarına ayrılır. Bu küçük iş parçaları farklı birimlere gönderilir.

Yalnız, burada önemli olan nokta, bu iş parçalarının işlemlerini, gönderildiği birimlerde eş süre kısıtı içerisinde tamamlaması gerekliliğidir. Süre sonunda bulunan hesaplama sonuçları merkezdeki (veya ortak) bir birime ya da bilgisayara alınırlar. Böylece, iş parçalarının paralel olarak hesaplanması ve paralellik kullanılarak daha kısa sürede işlem sonucuna ulaşılması sağlanmış olacaktır.

Paralel hesaplama kullanımı ile temelde;

1. Hesaplama sonuçlarının, hesaplama süresinin kısaltılarak daha kısa sürede ve daha hızlı bir şekilde elde edilmesinin sağlanması,
2. Bilgisayarın, işlemci ve bellek gibi parçalarının en verimli şekilde kullanılarak, performansın artırılması, bu şekilde de bilgisayarın hızında artış sağlanması, gibi durumların gerçekleştirilmesi hedeflenmektedir [7].



Şekil 1. Dört eş parçaya ayrılmış bir işin ayrı bilgisayarlarda (işlemci) eş zamanlı olarak çalıştırılması [8].

İşlemlerin paralel olarak hesaplanabilmesi amacıyla çeşitli kütüphanelerden yararlanılabilir. Bunların başında MPI (Message Passing Interface) Kütüphanesi [14] kullanılabilir [1] [3]. MPI [14] dışında Cilk [10], OpenMP [16] ve Co-Array Fortran [9] gibi kütüphanelerden de faydalanılabilir [7]. Bunların yanında LAM [14] ve MPICH [15] [1] uygulamaları da kullanılabilir [3] [7] [6].

Bir sonraki bölümde paralel hesaplamadaki avantaj ve dezavantajlardan bahsedilmiştir.

3. Paralel Hesaplamanın Artı ve Eksileri

Paralel hesaplamının öncelikli hedefi işlemlerin kolay ve hızlı bir şekilde tamamlanmasıdır. Bu da temel işin, farklı iş bölümlerine ayrılması ve sonuçların aynı sürede hesaplanmasıyla gerçekleştirilmektedir. Böylece, tüm işlemlerin tek bir makinede hesaplanması yerine birden fazla makinede iş bölümü uygulandığı için işlemler daha kısa sürede tamamlanır ve zaman avantajı sağlanır [6] [8].

Paralel hesaplama hız ve süre konusunda oldukça avantajlı olduğu halde, işlemin küçük iş parçalarına bölünmesi ile bir takım sıkıntılar oluşabilir. Bu sıkıntıların başında, iş parçalarının bir veya birkaçının doğru şekilde sonuçlandırılmadığı veya doğru sonucun verilen sürede hesaplanmadığı durumlar örnek olarak verilebilir. Böyle bir durumda, diğer tüm hesaplamalar tam olarak sonuçlandırılabilir bile hesaplaması tamamlanmayan iş parçası veya parçaları yüzünden tüm işlem yarım kalacak ve böylece ya sonuç doğru olarak elde edilemeyecek ya da sonuç hiç elde edilemeyecektir [7].

Bir sonraki bölümde Intel Parallel Studio 2008 [12] ve Intel Parallel Studio 2008'in [12] kullanıldığı bir matris çarpım hesaplama uygulaması ele alınmıştır.

4. Intel Parallel Studio ile Paralel Hesaplama

Paralel hesaplama uygulamalarında temel amaç, yapılması planlanan iş veya görevin, küçük görev parçalarına ayrılıp, farklı bilgisayar veya işlemcilere paylaştırılarak ve görev parçalarını alan her bir birimin mutlaka aynı süre içerisinde işlemi bitirmesi koşuluyla, hesaplamaların yapılması ve elde edilen sonuçların tek bir yerde toplanmasıdır.

Bu sayede işlemlerin veya problemlerin çözümü daha hızlı bir şekilde sonuçlandırılır.

Hesaplamaları paralel olarak gerçeklemek için kullanılabilir ortamlar arasında, çok çekirdekli-multicore Intel işlemcilerde [2] [3] [11] (paralellik sağlayabilmek için) “Intel Parallel Studio 2008” [12], “Microsoft Visual Studio 2008” [17] ortamıyla bir arada kullanılarak paralelleme işleminin gerçekleştirilmesini sağlar.

Paralel hesaplama işlemleri için Intel Parallel Studio 2008’de [12] üç temel aşama kullanılabilir [12].

Bu aşamalar;

1. Analiz & Tasarım (*Analysis & Design*)
 2. Kodlama & Hata Bulma (*Coding & Debugging*)
 3. Doğrulama & Ayarlama (*Verification & Tuning*)
- olarak sıralanabilir.

Intel Parallel Studio 2008’de [12] bu üç temel aşamayı gerçekleştirmek için Intel Parallel Studio [12] ortamında bulunan “Intel Parallel Composer” [12], “Intel Parallel Inspector” [12] ve “Intel Parallel Amplifier” [12] uygulamaları kullanılabilir [12].

4.1 Composer

Paralel hesaplama uygulamalarında paralelliği sağlayan “kodların” yazıldığı, derlendiği ve çalıştırıldığı uygulamadır. Paralel kodlar Composer’daki derleyici ve kütüphaneler ile tasarlanır ve yazılır. Intel C++ Compiler ile uygulamaların oluşturulmasına olanak verir [12].

4.2 Inspector

Paralel kodların “bellek (*memory*)” ve “thread hatalarının” ve eksikliklerinin tespit edildiği uygulamadır. Bu şekilde daha güvenilir kodlar yazılması sağlanır. Kodlamalarda oluşabilecek hatalar önceden tespit edilerek, kodlar daha güvenli hale

getirilir. Ayrıca, kodlardaki *datarace* veya *deadlock* gibi olasılıklar da belirlenebilir. Bellek hataları açısından bakıldığında, kodların oluşturabileceği bellek yetersizlikleri gibi olasılıklar da açığa çıkarılabilir. Inspector ile yapılabilecek analizlerin zaman bazındaki kontrol dereceleri, “*analiz yapılandırma*” kısmından arttırılıp azaltılarak ayarlanabilir [12].

4.3 Amplifier

Kodlardaki işlem sürelerinin, yazılan kodların “performans sonuçlarının” gözlemlenebildiği ve performanslarının incelenebildiği uygulamadır. Öngörülme durumların yol açabileceği performans kayıplarının tespit edilmesinde de bu uygulamadan yararlanılabilmektedir. Yazılan kodlara ait uygulamaların CPU’da en çok zaman harcadığı kısımların tespit edilmesi sağlanır. Seri veya paralel kodlarla yazılan programda oluşabilecek darboğazların (*bottleneck*) belirlenmesi ile hız artışının sağlanması amaçlanır [12].

4.4 Advisor

Intel Parallel Studio 2011’de [12] bulunan bir uygulamadır. Bu uygulama ile temel olarak, paralel uygulamaların daha hızlı ve hatalardan arındırılarak daha güvenli bir şekilde tasarlanması amaçlanır. Paralel Advisor’da bulunan “*Advisor workflow*” özelliği ile uygulamalara “adım adım paralellik” eklenmesi sağlanır. Seri kodlara farklı yollardan paralellik eklenmesini sağlayan bir yöntemdir [12].

Intel Parallel Studio’nun 2013 versiyonuna, 2008 ve 2011 versiyonlarının bazı bileşenlerinin geliştirilmesi ile yeni eklentiler yapılmıştır. Bunların başında, OpenMP 4.0’ın [16] belirli temel özelliklerinin desteklenmesi, Intel Composer [12] uygulaması ile çalıştırılabilen Intel Math Kernel Library’nin (Intel MKL) [13] geliştirilmesi ve Intel Advisor’ın kullanımının

yenilenmiş eğitim ve yardım penceresi ile daha kolaylaştırılması sayılabilir [13].

Sonraki bölümde Intel Parallel Studio 2008 [12] programının kullanıldığı bir uygulamaya yer verilmiştir.

5. Paralel Uygulama

Visual Studio 2008 [17] programıyla bütünleşik olarak Intel Parallel Studio 2008 [12] programı (Programın 30 günlük deneme sürümü kullanılmıştır.) çalıştırılarak, bir “matris çarpım işlemi” örneği oluşturuldu. Kodladığımız bu programda, işlem sürelerini görebilmek amacıyla, 1000x1000’lik iki tane matris oluşturularak, bu matrisler rastgele (*random*) sayı generatörüyle doldurulup ayrı ayrı dosyalara yazdırıldı. Sonra bu matrisler, yazdırıldıkları bu dosyalardan tekrar okutularak çarpıldı ve sonuç başka bir dosyaya yazdırıldı.

Uygulamamızdaki temel amaç, bu matrislerin okunup çarpılma işlemleri ve tüm diğer işlemler için “CPU’nun kullanımını” ve bu işlemlerde “ne kadar süre harcadığımızı” hesaplayıp, bu sürelerde hangi fonksiyonların ya da metodların CPU’yu en fazla meşgul ettiğini görmektir. Böylece, programın icrasında en fazla süre harcanan kısım(lar) tespit edilerek bu noktada paralellik sağlanabilir. Programımızda paralellik sağlamak için Intel Parallel Studio 2008 [12] ortamı kullanılmıştır.

Programa paralel komutlar eklenmeden önce nereye paralellik uygulanacağına karar verebilmek için programdaki metod ve fonksiyonların icra sürelerine bakıldı. Bunun için iki yöntem kullanıldı:

i. Birinci yöntemde, öncelikle hangi kısımda en çok zaman harcadığını görebilmek için yazdığımız şekilde hiçbir değişiklik yapılmadan, matris çarpım programı normal şekilde, seri olarak (paralel kodlar eklenmeden) Microsoft Visual Studio 2008 [17] ortamında derlendi.

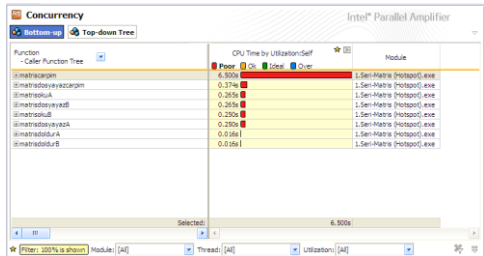
Programa eklediğimiz süre ölçüm kodları ile kullanılan fonksiyonların işlem süreleri gözlemlendi. Elde edilen sonuçlara göre en çok zamanın, matris çarpımı işleminde harcadığı görüldü. Yaklaşık olarak 6844 ms.

```
ca:\Documents and Settings\Administrator\Desktop\Matris Intel Parallel Stud
A Matrisi doldurma islemi icin gecen sure = 16ms.
B Matrisi doldurma islemi icin gecen sure = 15ms.
A Matrisi dosyaya yazma islemi icin gecen sure = 188ms.
B Matrisi dosyaya yazma islemi icin gecen sure = 187ms.
Carpma islemi icin gecen sure = 6844ms.
Carpma islemini dosyaya yazma icin gecen sure = 281ms.
```

Şekil 2. Matris hesaplama programı seri olarak çalıştırıldığında, işlem sürelerini gösteren ekran görüntüsü.

Şekil 2’deki ekran görüntüsünde de görüldüğü gibi programda en çok zaman “çarpma işleminde” harcanmıştır.

ii. İkinci yöntem olarak ise kodlarımızda bulunan metod veya fonksiyonlarda harcanan süreler, programımıza süre ölçme kodları eklemeksizin Intel Parallel Studio’nun [12] “Intel Parallel Amplifier” [12] uygulamasıyla erişebiliriz. Bunun için program kodlarımızı Intel Parallel Amplifier [12] ile derlersek şekil 3’te olduğu gibi CPU’yu en yoğun kullanarak en çok meşgul eden kod parçasının, çarpım işleminin gerçekleştiği fonksiyon olan “matris çarpım fonksiyonu” olduğu görülebilir. Yaklaşık olarak 6500 ms.



Şekil 3. Intel Parallel Amplifier ile hesaplanan icra süreleri.

Birinci ve ikinci yöntemlerden elde ettiğimiz sonuçlardan görüldüğü üzere, kodlarımızda CPU’yu en çok kullanan kısım, matris çarpımının gerçekleştirildiği kod satırlarıdır. Kodladığımız programımızdaki işlemlerin

daha hızlı gerçekleşmesi için bu fonksiyonun olduğu kısma paralel kodları ekleyerek paralellik uygulamalıyız. Bu nedenle paralelliği sağlayacak komutlar, matris çarpımını sağlayan döngülerden oluşan kod satırlarına eklendi.

Eklenen paralel komutlar ile paralellik sağlanarak, programdaki matris çarpım fonksiyonunun icra süresinin azaltılması amaçlanmıştır. Paralellik uygulanacak fonksiyon ise seri programımıza eklemiş olduğumuz süre ölçüm komutlarıyla ve Intel Parallel Amplifier [12] ile tespit edildi. Programımıza, paralelliği sağlayacak komutlar eklendikten sonra, derleme işlemi için “Intel Parallel Composer” [12] uygulaması kullanıldı. Matris çarpım fonksiyonumuzun icra süresi, paralelliği sağlayan komut eklendikten sonra 2266 ms.’ye düşmüş ve yaklaşık olarak fonksiyonun icra süresinde 4500 ms.’lik bir azalma gözlemlenmiştir. Şekil 4’te verilen ekran görüntüsünde matris çarpım fonksiyonunun icra süresinin paralellik sağlandıktan sonra azaldığı görülmektedir.

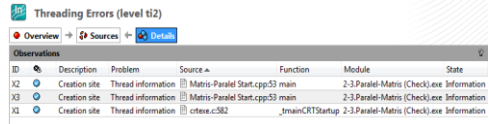
```
C:\Documents and Settings\Administrator\Desktop\Matris Intel Parallel Studio
A Matrisi doldurma islemi icin gecen sure = 16ms.
B Matrisi doldurma islemi icin gecen sure = 15ms.
A Matrisi dosyaya yazma islemi icin gecen sure = 172ms.
B Matrisi dosyaya yazma islemi icin gecen sure = 172ms.
Carpma islemi icin gecen sure = 2266ms.
Carpma islemini dosyaya yazma icin gecen sure = 281ms.
```

Şekil 4. Matris hesaplama programı paralel olarak çalıştırıldığında, işlem sürelerini gösteren ekran görüntüsü.

Bu sayede, Intel Parallel Studio [12] ortamı kullanılarak, “Intel işlemcili” [11] çift çekirdekli (2-core) bir bilgisayarda matris çarpım fonksiyonunun icrası iki çekirdeğe paylaştırılarak, paralellik sağlanmıştır.

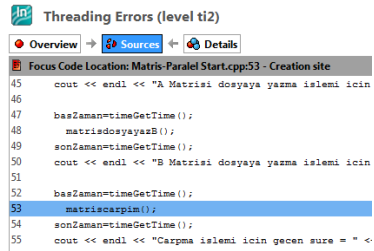
Yazılan programa ait thread hataları veya bellek (*memory*) eksiklikleri ise “Intel Parallel Inspector” [12] uygulamasıyla gözlemlenebilir. Uygulama çalıştırıldığında karşımıza gelen ekrandan, *Analysis Type* [12] kısmından seçilebilecek *Memory Errors*

[12] veya *Threading Errors* [12] adımlarından, kodda oluşabilecek hata veya eksikliklerin tespiti ve analizi şekil 5 ve şekil 6’daki gibi yapılabilir [12].



Şekil 5. Intel Parallel Inspector uygulaması, thread hataları analiz sonucu.

Intel Parallel Inspector [12] uygulamasının çalıştırılmasıyla elde ettiğimiz sonuçlardan, *main()* fonksiyonundan çağırılan matris çarpım fonksiyonunda oluşabilecek olası bir thread hatası belirtilmiştir.

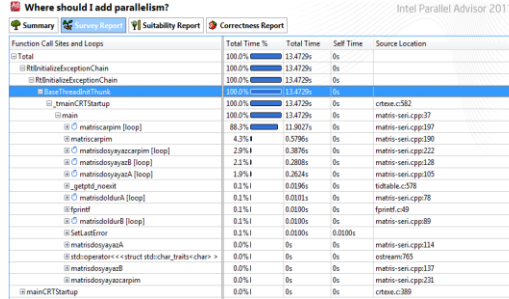


Şekil 6. Intel Parallel Inspector, thread hataları analiz sonuçları detaylı gösterimi.

Intel Parallel Studio 2008’deki Composer, Inspector ve Amplifier uygulamalarının dışında Intel Parallel Studio 2011’e [12] eklenen “Intel Parallel Advisor” [12] uygulamasıyla ise seri kodlarda, nereye, nasıl ve hangi şekillerde paralellik uygulanması gerektiği değerlendirilerek, paralel hesaplama gerçekleşmesi sağlanabilir. Advisor [12] ile yazılan programa paralellik eklenmesi adım adım gerçekleştirilerek, kodların hangi kısmında paralellik kullanılabileceği konusunda programcıya yardımcı olunur.

Advisor ile matris hesaplama programımızın paralel kodlar eklenmemiş olan seri kısmı çalıştırıldı. Öncelikle, Advisor uygulamasının *Survey Analysis* [12] kısmından programın icra süreleri incelendi. Bu sonuçlara göre en çok zaman harcanan fonksiyonun şekil 7’de görüldüğü gibi *main()* fonksiyonundan

çağrılan matris çarpım fonksiyonu olduğu görüldü.



Şekil 7. Intel Parallel Advisor, Survey Analysis uygulaması sonuçlarının gösterimi.

En çok zaman harcanan kısım olarak matris çarpım fonksiyonu belirlendikten sonra, matris hesaplama programına paralellik uygulamak için en uygun kod parçasının, bu fonksiyondaki çarpımı sağlayan satırlar olacağı varsayıldı. Bu satırların paralellğe uygun olup olmadığını denetlemek için *Suitability Analysis* [12] ile çarpım işleminin yapıldığı komutlarda, paralellik sağlandığı zaman performansın nasıl etkileneceği incelendi. Son aşama olarak, *Correctness Analysis* [12] ile bu eklediğimiz kodlarda oluşabilecek veri paylaşım hataları gibi hataların tespiti yapıldı.

6. Sonuçlar

Bu çalışmada paralel hesaplama konusu ve Microsoft Visual Studio 2008 ile birlikte kullanılan Intel Parallel Studio 2008 uygulaması ele alındı. Paralel hesaplamada temel amaç, küçük iş parçalarına ayrılan problem çözümlerinin aynı zaman aralıklarında hesaplanması kaydıyla, bulunan sonuçların tek bir birimde (merkez bilgisayar...) birleştirilerek, işlemlerin daha kısa sürede, daha hızlı ve etkin bir şekilde yapılabilmesinin sağlanmasıdır. Paralel hesaplama işlemleri, MPI gibi çeşitli kütüphanelerle veya Intel Parallel Studio gibi programlama ortamı uygulamaları ile gerçekleştirilebilir.

Dumlupınar Üniversitesi Bilgisayar Mühendisliği Bölümü öğrencilerine, Paralel Hesaplama dersinde, Intel Parallel Studio 2008 programı kullanılarak bu uygulamalar yapıldı. Bu makalede verilen aşamalar ve Intel Parallel Studio uygulamaları kullanarak, öğrencilerin paralel kodlar geliştirmeleri, ekran görüntüleri olarak bu konuda bilgi edinmeleri ve paralel hesaplama gerçekleştirebilmeleri sağlandı.

Kaynaklar

[1] Petersen W. P., Arbenz P., "Introduction to Parallel Computing", Oxford University Press, United States, 2004, pp 1-5, 156-161.

[2] Alexander M, Gardner W., "Process Algebra For Parallel And Distributed Processing", Chapman & Hall/CRC Press, United States of America, 2009, pp 15-18.

[3] Wittwer T., "An Introduction to Parallel Programming", 1st Edition, VSSD, The Netherlands, 2006, pp 3-13.

[4] Grama A., Gupta A., Karypis G., Kumar V., "Introduction to Parallel Computing", 2nd Ed., Addison Wesley, 2003, pp 1-11.

[5] Application of Parallel Computing in Digital City, Zhu Dingju, High Performance Computing and Communications, HPCC '08. 10th IEEE International Conference, 2008.

[6] "MPICH2 Kullanarak Windows, Linux ve "Windows veya Linux" İşletim Sistemleri Yüklü Bilgisayarlardan Oluşan Küme Kurulumu ve Bu Kümede C++ Dilinde Yazılmış Paralel Programları Yazıp Çalıştırma Rehberi", Sururi E., 2006: http://www.emresururi.com/mpich2/mpich2_rehberi_v1_2.pdf

[7] Akçay M., Erdem H.A., "Paralel Hesaplama ve MATLAB Uygulamaları", Akademik Bilişim Muğla Üniversitesi, 2010.

[8] MPI Programlamaya giriş ve motivasyon (Kısım 1)", Akıncı Ö., 2009:
http://www.uybhm.itu.edu.tr/documents/basarin09sunum/01_Giris_ve_Motivasyon_akinci_v3.pdf

[9] Co-Array Fortran Resmi Sitesi:
<http://www.co-array.org/>

[10] "Multithreaded Programming In Cilk", Frigo M., 2011:
http://www.ll.mit.edu/HPEC/agendas/proc07/Day3/25_Frigo_Pres.pdf

[11] Intel İşlemci Resmi Sitesi:
<http://www.intel.com/products/processor/index.htm>

[12] Intel Parallel Studio (IPS) Resmi Sitesi:
<http://software.intel.com/en-us/intel-parallel-studio-home/>

[13] Intel Parallel Studio XE Suites-What's New:
<http://software.intel.com/en-us/intel-parallel-studio-xe/#pid-3388-862>,

[14] LAM/MPI Parallel Computing Resmi Sitesi: <http://www.lam-mpi.org/>

[15] MPICH Resmi Sitesi:
<http://www.mcs.anl.gov/research/projects/mpi/mpich1/>

[16] OpenMP Resmi Sitesi:
<http://openmp.org/wp/>

[17] Microsoft Visual Studio Resmi Sitesi:
<http://www.microsoft.com/visualstudio/tur>

[18] Programming on Parallel Machines, Matloff N., University of California:
<http://heather.cs.ucdavis.edu/~matloff/158/PLN/ParProcBook.pdf>