

GRID Üzerinde Gerçeklenen Matris İşlemlerine Bir Portal

Üzerinden Erişim

Orhan Dağdeviren¹, İlker Korkmaz²

¹ İzmir Yüksek Teknoloji Enstitüsü, Bilgisayar Mühendisliği Bölümü

² İzmir Ekonomi Üniversitesi, Bilgisayar Mühendisliği Bölümü

orhandagdeviren@iyte.edu.tr, ilker.korkmaz@ieu.edu.tr

Özet: GRIDler, birçok farklı kullanıcının heterojen kaynaklarından oluşur ve onlar için bu kaynaklara uzaktan erişim sağlar [1-4]. GRID kaynaklarına yetkili kullanıcıların İnternet üzerinden erişebilmesi için portal tasarımları yapılmaktadır [5-10]. Bu çalışmada, yazarlar tarafından tasarlanan bir portal aracılığıyla GRID kaynaklarına yetkili erişim ve örnek bir hizmet olarak sunulan matris işlemlerinin gerçekleştirilmesi aktarılmaktadır. İlgili GRID ortamının etkinliğini göstermek için farklı boyutlarda matrisler üzerinde performans testleri de yapılmıştır.

Abstract: GRIDs consist of heterogeneous computational resources of various users, and provide them with remote access to these resources [1-4]. Portal designs are made for authorized user to access GRID resources via Internet [5-10]. In this study, accessing to GRID resources through a portal designed by the authors and the implementation of the matrix operations as an example service is presented. In order to show the efficiency of the related GRID environment, performance experiments on various sized matrix data are tested.

Anahtar Kelimeler: GRID, Portal, Matris İşlemleri, Performans Testi.

1. Giriş

Günümüzde, yüksek miktarda verinin kullanıldığı alanlarda ve/veya yüksek hızlarda işlemci gücünün gereksinim olduğu alanlarda performans kriteri önemli bir vurgudur. Dolayısıyla, bu tarz yüksek kaynaklar gerektiren hesaplamalar için sadece bir girdi/çıkış ortamı olarak düşünmeden coğrafi olarak dağıtık bir ağ ortamı oluşturulup tüm kaynakların etkin kullanılması, bu performans kriterini önemli ölçüde arttıracak bir olgudur. GRID hesaplama kavramı, bu niyetle ortaya çıkmış bir teknolojidir [1-4]. GRID hesaplama ortamına erişim güvenli ve kesintisiz bir biçimde gerçekleştiği sürece, kullanıcılar bu ortamı, donanım altyapısından bağımsız olacak bir şekilde kendi kişisel izinleri doğrultusunda kullanabilmektedir.

Portal, kullanıcının GRID üzerinde etkin çalışmasını sağlayan bir arayüzdür [5-10]. GRID

portallarının kullanımı, iki ana zorlukla başa çıktıkları için, çok önemli görülebilir [7]. Bunlardan ilki, GRID ortamının, takvimleme, güvenlik, servis kalitesi, veri transferi, ağ protokolü gibi kullanıcıya zor gelebilen karmaşık yapıları içermesidir. İkincisi de, kullanıcının bilimsel çalışmasında yararlandığı programın kullanım zorluğudur. Bu iki zorluk düşünüldüğünde, kullanıcının konsol gibi kısıtlı bir arayüz üzerinden çalışması sıkıntılı olmaktadır. Portal gibi kullanıcıya yol gösteren bir arayüz, bu durumda faydalı olacaktır.

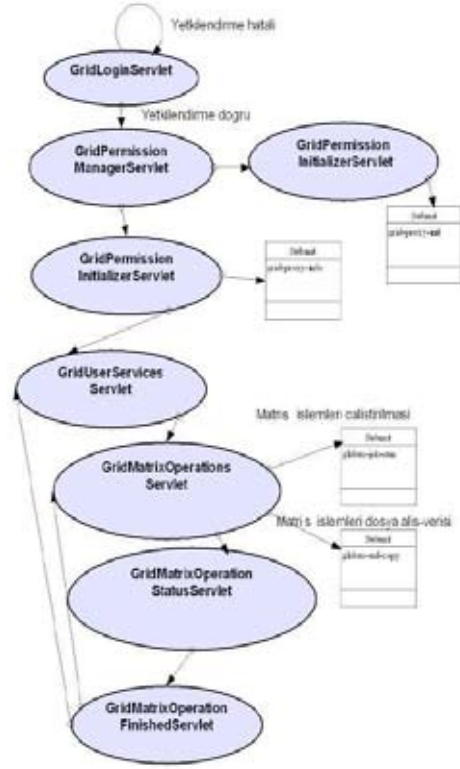
Bu çalışma, EGE Üniversitesi Uluslararası Bilgisayar Enstitüsü (UBE) Linux laboratuvarındaki, Globus bazlı GRID ortamında, örnek uygulama olarak seçilen matris işlemlerinin gerçekleştirilmesini ve bu servislerin bir web portalı aracılığıyla İnternet üzerinden erişilerek izinli bir şekilde kullanılmasını kapsamaktadır. Tasarlanan portal sayesinde, UBE Linux labo-

ratuvarındaki GRID ortamına, istenirse gelecekte İnternet üzerinden iş gönderilebilir. Çalışmada, GRID ortamındaki iki ayrı makinenin dağıtık ve paralel kullanımı ile matris toplama, çıkarma ve çarpma işlemleri gerçekleştirilmiş ve veri setinin değişken boyutları için tek makinedeki seri işlemler ile GRID ortamındaki paralel işlemler karşılaştırılmıştır.

2. Web Portalı

Bir web portalı genel anlamda, World Wide Web (WWW) üzerinde bir site olup siteye erişen ziyaretçiye kişisel servisler sunar. Böylece her ziyaretçi kendine has bir servise erişebilmek için ilgili bağlantıya yönlendirilir. Dağıtık donanım altyapısına sahip çok kullanıcıli sistemlerin web portalı desteğiyle kullanılması, kolay kullanılan amaca yönelik bir arayüz sunulabildiği için, günümüzde yaygın hale gelmiştir. Bu çalışmada kullanılan portal, WWW üzerindeki açık bir alanda olmayan UBE laboratuvarında gerçekleştirilmiştir, fakat istenirse WWW erişimi mümkün kılınabilir.

Bu çalışmadaki sisteme portal desteği katabilmek amacıyla, Apache Tomcat (v_6.x) sunucusu kullanılmıştır. Tomcat, temelde bir web barındırıcısı (*web container*) olup web programlamada uygulamaları yaratmakta kullanılan, birbirleriyle etkileşebilen mantıksal bileşenlerin tutulduğu alanı sunmaktadır. Tomcat, bir web sunucusu ile entegre olarak JAVA kodu çalıştırmaya yarayan ve SUN tarafından belirlenen “servlet” ve “Java Server Pages” (JSP) spesifikasyonlarını gerçeklemektedir. Genel olarak bir *servlet*, bir web sunucusunda JAVA platformu kullanarak dinamik içerikli kod geliştirmeye yarayan ve kendisine gelen istekleri ilgili uygulamaya veya içeriğe yönlendirerek cevaplayan bir nesnedir. Bu çalışmada kullanıcının GRID ortamına erişebilmesi ve bu ortamda kendisine sunulan hizmetlere kendisine has dinamik parametreler ile erişebilmesi için *servlet* yapıları kullanılmıştır. Kullanılan *servlet* yapılarının ağ akış şeması Şekil 1’de verilmiştir.



Şekil 1. Ağ akış şeması.

Tasarlanan portalda, kullanıcının ilk erişebileceği arayüz, sisteme giriş sayfası olup burada kullanıcı kendisine ait parolayı sunarak kimliğini doğrulamak durumundadır. Bu sayfada yanlış bir parola girilmesi durumunda ya da bu sayfayı yan geçit (*by-pass*) yaparak diğer servis sayfalarına geçilmek istendiği durumda, giriş sayfası aşılamamaktadır. *Servlet* yapılarının kendi içinde sunduğu oturum (*session*) bilgileri ile denetlenen sayfalar arası yönlendirme, sadece ilgili *servlet* aracılığıyla barındırılan JAVA kodunun çalıştırılabilmesi veya ilgili sayfaya göndermenin *servlet* içindeki dinamik bilgilerle kontrollü bir şekilde yapılabilmesi ile mümkündür.

Bunların yanı sıra, giriş ekranında kullanılan parola denetleme mekanizması, seçilen parolanın MD5 kullanılarak özü alınmış halde sak-

lanması ve denetlenecek olan parolanın da aynı işlemde geçerek karşılaştırılması yöntemine dayanmaktadır. Bu aşamada, MD5 yöntemi ile öz alma işlemi kullanmak yerine farklı öz alma veya şifreleme yöntemleri de kullanılabilir. Projede yazılan kodlarda bu değişim imkanlarını gerçekleştirmek, program geliştiricileri için sadece ilgili modülleri düzenlemek anlamındadır. Proje kodu, tamamen JAVA ortamında yazılmış olup bu modüllere ait olan sınıflardaki değişimler kolay gerçekleştirilebileceğinden dolayı, esnek bir yapıdadır.

Ayrıca, projenin portal kısmını oluşturan kodlar ve GRID üzerinde çalıştırılacak olan kaynak dosya kodları, tümüyle Eclipse açık yazılım çatısı ortamında JAVA dili ile geliştirilmiştir.

3. GRID Orta Katmanı

Uygulamada, portala erişildikten sonra, sistemde *servlet* yapılarının GRID orta katmanında çalıştırdıkları ilgili GRID komutları Şekil 1'deki ağ akış şemasında belirtilmiştir.

Portala erişilip GRID sistemine giriş izni olan kullanıcıların parolalarının kontrolü sonucu, *servlet* aracılığıyla, kullanıcı, GRID hizmetlerine yönelik ilk sayfaya yönlendirilir. Bu sayfada, kullanıcının GRID üzerinde çalışma yapmayı başlatabilmesi için öncelikli iş olarak, zaman sınırlı bir kredi bileti alması zorunludur. GRID ortamındaki sertifika otoritesi görevini üstlenen makinede, Globus bazlı GRID ortamında desteklenen “grid-proxy-init” komutunu çalıştıran kullanıcı bu hakkı almaktadır. Uygulamadaki sertifika izninin zaman sınırı 12 saattir. Kullanıcı, 12 saat sonunda çalışmaya devam edebilmek için yine ilgili komutu kullanarak kendine bilet almak durumundadır.

Kullanıcıların giriş ekranından yönlendirildiği *servlet* sayfası, bilet hakkı almak üzere ilgili komutu çalıştıracak arayüzü, bu hakla ilgili olarak sertifika bilgilerini ve kalan zamanı sunan “grid-proxy-info” komutunu çalıştıracak ara-

yüzü ve hakkı olan kullanıcıları operasyonlara yönlendiren arayüzü sunmaktadır. Böylece, kullanıcılar herhangi bir ek komut kullanmadan, dinamik sayfalardaki ilgili açıklamalar ile kolayca işlemlerine yönlendirilebilmektedir.

Dinamik sayfalar aracılığıyla yönlendirilerek operasyon sayfasına ulaşan kullanıcılar, kendilerine sunulan servisleri nihayet kullanabilir. Burada tekrar vurgulamakta yarar görülen şey şudur ki, kullanıcılar asla bu sayfalara direkt erişemezler. *Servlet* yapıları bu erişimleri ilk sayfadaki giriş izninden sonra yönlendirme mekanizması ile denetlemektedir. Bu projede, GRID ortamında gerçekleştirilen bir uygulama hizmeti olarak, başlıca matris işlemlerinin gerçekleşmesi sunulmuştur. Bu doğrultuda, kullanıcılar operasyon sayfasında, matris operasyonlarını seçeceklerdir. Eğer, daha sonradan sisteme GRID ortamında gerçekleştirilebilen başka bir uygulama, hizmet olarak katılacaksa, bu sayfanın barındırdığı koda ilgili eklemeler esnek bir şekilde kolayca dahil edilebilir. Çünkü, bu kısım artık, GRID üzerinde çalıştırılması başarılı olarak gerçekleştirilen her uygulamayı bağımsız bir şekilde içerebilecektir.

Sunulan matris operasyonları sayfasına erişen kullanıcılar, burada kendi belirttikleri matris dosyalarını dinamik parametreler aracılığıyla göstererek, seçtikleri matris işleminin GRID ortamında yapılması isteğinde bulunabilir. Matris dosyaları, kullanıcıların okunma izni olan dosyalara, ilk satırlara matrisin satır ve sütun boyutlarını ve diğer satırlara da kendi hazırladıkları matris verilerini kaydetmeleri ile oluşturulmaktadır. Bu matris dosyaları, kullanıcının seçtiği GRID üzerinde çalışan 2 makinede ve yine kullanıcının seçtiği yük dağılımları ile dağıtılabilir ve ilgili seçilen operasyon paralel bir şekilde GRID üzerinde yürütülebilir. Sunulan matris işlemleri toplama, çıkarma ve çarpmadır. Bu işlemler, programın beklediği giriş ve çıkış dosyalarının biçim kurallarına uyularak ilgili JAVA kodlarının hazırlanması koşulu ile arttırılabilir. Ayrıca, sistemde kullanılacak

olan makine isimleri, makine sayıları ve makineler arası yük dağılımları farklı hale getirilebilir. Fakat, bu parametrelerdeki dinamik yapı, kodun güncel hali ile yeterince esnek değildir. Kodda, GRID ortamındaki donanımlara bağlı parametrelerin değişimi, yeniden düzenleme ile daha esnek hale getirilebilir.

Seçilen matris işlemi ve verilen matris dosyaları sonucu, veriler, ilgili makinede belli oranlarda paylaştırılarak 4 yeni matris dosyası oluşturulur. Bu dosyalardan 2 tanesi, belirtilen yük dağılımına uygun bir şekilde GRID üzerinde belirtilen diğer makineye gönderilir, diğer 2 tanesi de verilen makinede çalıştırılır. Her iki makinede de bu matris operasyonlarının tümünü çalıştırabilecek olan aynı hedef dosyalar bulunmaktadır. GRID üzerinde çalıştırılmak istenen kodlar bu ilk makineden diğer tümüne, sistem açılmadan önce gönderilmiştir. Makinelerin çalışma performanslarının birbirinden bağımsız gelişmesi için, ilk makine, paylaştığı dosyaları izlek (*thread*) aracılığıyla gönderir ve kendi dosyaları ile sonucunu hesaplayıp, diğer izlek cevaplarını yine dosya olarak alıp bu sonuç dosyalarını birleştirir. Böylece, ilk makineye verilen girişler ile oluşturulan sonuç, arayüzde belirtilen sonuç dosyası içeriğine kaydedilir. Bu işlemler esnasında, *servlet* bekleme konumundadır ve işlem bitince çalışma zamanı bilgisini de vererek, işlemin bittiğini belirten sayfaya yönlendirme yapar. Bu sayfadan sonra, kullanıcı yeni bir işlem yapmak isterse veya GRID sertifikası hakkında bilgilerini almak isterse ilgili sayfalara yönlendirilebilir. Eğer kullanıcı çıkış yapmak isterse, yine *servlet* kontrol mekanizması ile oturum sonlandırılır ve artık kullanıcının yeni bir giriş yapması için ilk giriş sayfasında tekrar parola kontrolü yapması gereklidir. Böylece, sistemden güvenli çıkış gerçekleştirilmiş olmaktadır.

Matris işlemlerinin yapılması esnasında, verilerin paralel olarak işlenmesi amacıyla, matrisler, toplama ve çıkarma için satırlara göre parçalanmaktadır. Örnek olarak 10x10 boyutunda

iki matris toplanacaksa, verilen yük dağılımının eşit olduğu durum için dört adet 5x5 matris oluşturulacaktır. İlk 2 matris, ana matrislerin ilk 5 satırlık verisini, diğer 2 matris de ana matrislerin kalan 5 satırlık bilgisini tutmaktadır. Matris çarpma işlemi içinse, ilk matris dosyası verilen yük dağılımı doğrultusunda parçalanırken ikinci dosya tümüyle kullanılır. Bu durumda yine ilk matris satır bazlı parçalanmıştır ama diğer matris aynen kullanılacaktır. İlk iki dosya izlek aracılığıyla, “gsiftp” servisi kullanılarak ilgili diğer makineye gönderilir ve seçilen operasyona göre orada seçilen ikili dosyada yer alan “globus-job-run” komutu ile çalıştırılıp oluşturulan cevap yine “gsiftp” komutu ile uzaktaki makineden geri gönderilir. Bu esnada, çalışılan makine kendi verilerini paralel olarak hesaplamakla ve diğerinden gelen sonuç ile kendi sonucunu tek bir sonuç dosyası altında birleştirmekle görevlidir.

Dosyaların, başka makinelere iletimi ve geri alımı esnasında trafik nedeniyle gider (*overhead*) artabilir, fakat büyük boyutlarla yapılan işlemler için bu giderin, paralellikten gelen hız ile kazanılan avantaj sayesinde katlanılabilir hale geleceği beklenmektedir. Çeşitli boyutlarda matrislerle yapılan işlemlerde ölçülen zaman sonuçlarının bu beklenti ile tutarlı olduğu gözlenmiştir.

3. Deneysel Sonuçları

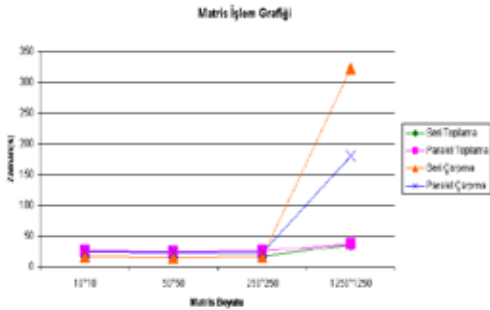
Bu çalışmada, ilgili *servlet* aracılığıyla erişilen matris operasyonları sayfasında, kullanıcı, GRID üzerinde kodun dağıtılmasını istediği makineleri belirtebilmektedir. Bu kısımda, eğer sadece bir makine gösterilirse, işlemler tek bir makinede, hiçbir dosya parçalaması yapılmadan ve hiçbir veri gönderme işi gerçekleşmeden, sadece ilgili operasyonun hedef dosyası üzerinden “globus-job-run” ile seri bir şekilde gerçekleştirilecektir. Bunun yanı sıra, uygulama için iki makine seçildiğinde, verileri parçalama, gönderme, geri alma ve birleştirme işlemleri ek zaman maliyeti getirecektir. İki

ayrı makinenin paralel çalışma avantajı, matrislerin boyutları çok büyük olduğunda daha belirgin bir şekilde ortaya çıkacaktır.

GRID ortamı kullanılarak, seri ve paralel olarak örnek uygulamalar yapılmıştır. Giriş matrisleri, JAVA kütüphanesi desteği ile rastgele tamsayı değerleri olarak üretilmiş ve her iki ayrı uygulama için üretilen aynı verilerin çalışma zamanları kıyaslama olarak ele alınmıştır. 10x10, 50x50, 250x250, 1250x1250 boyutlarında giriş matrisleri içeren dosyalar oluşturulmuştur. Daha büyük boyutlar da denenebilir, fakat çalışma zamanı çok uzamaktadır. Amaç olarak, bir işlem için, GRID ortamında iki makinede paralel çalışan kodun, GRID üzerinde tek bir makinede seri çalışan koddan daha hızlı çalışabildiği ilk an yakalanmaya çalışılmıştır. Bu andan sonra, ilgili bulunan boyutlardan daha büyük boyutlar ile işlem yapıldığında, paralel kodun seri koddan daha hızlı çalışacağı düşünülmektedir. Tablo 1’de çalışma süreleri eşit yük dağılımına (%50) göre özetlenmiş, Şekil 2’de ise bu süreler grafik şeklinde gösterilmiştir.

İşlem	10x10		50x50		250x250		1250x1250	
	s	p	s	p	s	p	s	p
Toplama	16	25	16	25	18	25	36	38
Çarpma	17	24	16	23	17	23	322	181

Tablo 1. Matrislerin ilgili boyutları için işlemlerin seri ve paralel çalışma zamanları (sn).



Şekil 2. Matris işlemleri grafiği.

Alınan sonuçlar, aşağıdaki Tablo 1’de saniye bazında belirtilmiştir. Tablo 1’deki “s” seri olarak tek bir makinede çalışma durumunda geçen zamanı, “p” ise paralel olarak iki makinede çalışma durumunda geçen zamanı belirtmektedir.

Tablo 1’de ve Şekil 2’de görüldüğü üzere, 10x10, 50x50 ve 250x250 veri boyutları için paralel ve seri gerçekleştirilen çarpma ve toplama işlemleri yaklaşık olarak aynı zamanlarda bitmiştir. Bunlara karşın, 1250x1250 boyutlarındaki matrisler için paralel çarpma işlemi, seri çarpma işlemine göre daha kısa sürmüş olup bu durumda iki makinenin paralel işlem gücü, tek makinedeki seri işlem çözümüne göre ön plana çıkmıştır.

4. Sonuçlar

Bu çalışma, GRID ortamında gerçekleştirilen örnek matris uygulamalarının, bir portal arayüzü sayesinde, etkin ve kolay bir şekilde kullanılabilmesi amacıyla hazırlanmıştır. UBE Linux laboratuvarında, Globus GRID ortamına bağlı olan “linux11.ube.ege.edu.tr” ve “linux12.ube.ege.edu.tr” makinelerinde uygulamalar denenmiştir. Matris işlemlerinden, toplama, çıkarma ve çarpma uygulamaları, örnek servisler olarak kullanıcıların hizmetine sunulmuş ve rastgele veri içeren matris dosyaları hazırlanarak çeşitli örnek denemeler başarıyla gerçekleştirilmiştir.

Gerçeklenen testlerin çalışma zamanları dikkate alınarak, GRID ortamında, belli veri boyutunu aşan matris çarpım işlemlerinde GRID ortamındaki kaynakların paralel kullanımının avantajı görülebilmektedir. Fakat, düşük boyutlu matris işlemlerinde, ortamın maliyet ve gider dezavantajı sebebiyle, GRID ortamında çok kaynaklı çalışma zamanının, seri bir işleme göre daha fazla olduğu gözlenmiştir. Matris verilerinin 1250x1250 boyutlarından daha fazla olduğu durumda, GRID ortamında paralel kaynak kullanımının, zaman açısından avantajı görülmeye başlanmıştır.

Kaynaklar

- [1] Foster, I., Kesselman, C., Tuecke, S. ‘The Anatomy of the Grid: Enabling Scalable Virtual Organizations’, *Int. Journal of High Performance Computing Applications*, vol.15, no.3, 2001, pp 200-222.
- [2] Foster, I., ‘What is the Grid? A Three Point Checklist’, *Grid Today*, vol.16, 2002.
- [3] Foster, I., ‘The Grid: Computing Without Bounds’, *Scientific American Magazine*, 2003.
- [4] Foster, I., Kesselman, C., Tuecke S., ‘The Anatomy of the Grid: Enabling Scalable Virtual Organizations’, *Int. Journal of Supercomputer. App.*, vol.15, 2001.
- [5] Phumisuth, Y., Achalakul, T., ‘Grid Portal Design and Usability Evaluation’, *ISCIT*, 2006, pp 186-193.
- [6] Cai, Y., Cao, J., Li, M., Chen, L., ‘Portlet-based Portal Design for Grid Systems’, *GCCW*, 2006, pp 571-575.
- [7] Feng, Z., Yang, S., Long, S., Chen, D., Guo, L., ‘Research on Integrating Service in Grid Portal’, *LNCS*, vol.3032, 2004, pp 821-824.
- [8] Tourino, J., Martin, M.J., Tarrío, J., Arenaz, M., ‘A Grid Portal for an Undergraduate Parallel Programming Course’, *IEEE Trans. on Education*, vol.48, 2005.
- [9] Novotny, J., Russell, M., Wehrens, O., ‘GridSphere: A Portal Framework for Building Collaborations’, *1st Int. Workshop on Middleware for Grid Computing*, 2003.
- [10] Wang, X.D., Yang, X., Allan, R., ‘Top Ten Questions to Design a Successful Grid Portal’, *2nd Int. Conf. on Semantics, Knowledge, and Grid*, 2006.