

Yaşar Üniversitesi
Bilgisayar Mühendisliği Bölümü
COMP 4920 Mezuniyet Tasarım Projesi II, Bahar 2020
Bitirme Projesi Özeti

Proje Kodu ve Adı:	DJAVAC: Automated Load Distribution Engine for Java
Proje Takımı:	HALİL BÜLENT ORHON halilbulentorhon@gmail.com YAREN GÜNDÜZ yarennn.gunduzzz@gmail.com FULYA ERTAY fulya.ertay@gmail.com BERKE SAYAN bersay97@hotmail.com
Proje Danışmanları:	Assistant Professor Hüseyin Hışıl
Proje Çıktıları:	<ol style="list-style-type: none">1. Final Report2. Requirements Specifications Document3. Design Specifications Document4. Product Manual5. Product-Hardware Subsystem, if any6. Product-Software Subsystem7. Any other deliverables
Proje Web Adresi:	https://djavacteam.wixsite.com/djavac

Proje Özeti

1.Giriş

Geliştirdiğimiz DJAVAC projemiz, yazılım alanıyla ilgili belirli bir sorunu çözer. Java programcısının, kod yazarken yaşadığı sıkıntıları çözmeyi amaçlamaktayız. Örneğin, Java programcı aynı metodları birçok kez, editör programda yazarak, zaman kaybetmektedir. Bu nedenle hedefimiz, programcının yazdığı kodları ağ üzerinden alarak, sunucu makinaya göndermektedir. Bu şekilde, sunucu, istemcinin kodlarını çalıştırır ve sonucu tekrar istemciye, arayüz programı aracılığıyla iletir. Java dili kodlayan programcı kendi kodlarını yazar ve bu kodun 'dağıtılmış' anahtar kelime olarak pragması varsa, programcının kodu, sunucu yürütmesine uygun hale getirmek için ayrıştırılır, dağıtılır ve oluşturulur. Sunucu yürütme tamamlandığında, kodun çıktıları veya sonuçları her zamanki gibi istemciye döndürülür. Bu işlemi yaparken istemci kodu başka bir java dosyası olarak kopyalanır ve koddaki diğer değişiklikler burada tamamlanır. Java RMI yardımıyla, RMI arabirimini kullanarak, istemci tarafı kodlarındaki RMI uygulama dosyalarındaki değişiklikleri iletmek için daha iyi bir bağlantı sağlar.

2.Gereksinimler

DJAVAC projesinin çalışabilmesi için bazı işlevsel gereksinimlere ihtiyaç duymaktadır. Örneğin, herhangi bir Java editörü, Java ortamı (JDK, JRE, JVM) ve Windows 10 veya Ubuntu işletim sistemlerine ihtiyaç duymaktadır. Ayrıca, sistem iki kullanıcıya ihtiyaç duymaktadır. Bunlar istemci ve kullanıcı. İstemci ve sunucunun çalışabilmesi için, her biri ilgili kullanıcı arayüz programını çalıştırmalıdır. DJAVAC takım üyeleri, projenin gereken kodlarını, Eclipse Java editör program kullanarak yazmıştır. Ayrıca, proje çalışabilmek için her iki kullanıcının da belli bir ağa bağlı olması gereklidir. Projemiz, istemciden kodları alıp, sunucu tarafında çalıştırabilmesi için, RMI API ye ihtiyaç duymaktadır. Bunlara ek olarak, DJAVAC projesi işlevsel olmayan gereksinimlere de ihtiyacı vardır. Örneğin, performans gereksiniminde, DJAVAC sistemi ölçeklendirilebilir olmalıdır. İstemci veya sunucu sayısı artarsa, sistem hem istemci hem de sunucular için işlemleri hızlı bir şekilde yürütmelidir. Bu ölçeklenebilirlik dağıtılmış bir şekilde karşılanmalıdır. DJAVAC sistemi, daha erişilebilir olması için istemci ve sunucu arayüzlerine sahip olacaktır. Sistem yalnızca Java editör programını, Java

programlama dilini ve RMI mimarisini kullanmalıdır. Diğer bir işlevsel olmayan gereksinim ise, güvenlik gereksinimidir. DJAVAC sisteminde, hem istemci hem de sunucu güvenlik duvarlarını durdurmalıdır. İstemci dağıtılmış kodunu sunucuya gönderdiğinde veya sunucu istemcinin işlevinin sonucunu istemciye döndürdüğünde bu önemlidir. Ayrıca, DJAVAC sisteminin özgünlük mekanizmasına ihtiyacı vardır. İstemci ve sunucu birbirleriyle iletişim kurduğunda, istemcinin veya kullanıcının kimliğini doğrulamak için IP adreslerini ve bağlantı noktası numaralarını girerler. Ayrıca, DJAVAC sistemi TCP / IP sistemini kullanarak iletişim sağlar.

3.Tasarım

Projemizin tasarım kapsamında iki arayüz dosyamız vardır. Sunucu, “Sunucu arayüz programı” ile etkileşime girer ve İstemci “İstemci arayüz programı” ile etkileşime girer. Sunucu ilk olarak, Central Server, Server ve RMI Server düğmelerine bastıktan sonra, istemcinin kodunu kabul etmeye hazır olacaktır. Ardından, İstemci Dağıt ve Kod Oluşturucu düğmesine ve Yürüt düğmesine basar, istemci kodlarını, sunucuya gönderir ve sunucu kod sonucunu istemciye döndürür. Tüm bu işlemler RMI API kullanılarak tasarlanmıştır. Ayrıca, projemizde yer alan bazı gerekli sınıflar şunlardır:

1. Parser (Dağıtılmış yöntemleri tespit etmek için istemci kodunu küçük parçalara ayırır.)
2. Kod Jeneratör (İstemci ve Sunucu iletişimi için arayüz, uygulama ve ana yöntemleri oluşturur).
3. RMI sunucusu başladıktan sonra istemci kodunu engellemek için kullanılan yöntem, bu yöntem komut penceresinde yürütülmeye başlar ve sunucu sınıfından istemciye döner.
4. Birden çok sunucu ve çoklu istemci iletişim sistemi, çoklu thread yaratılmıştır.

4.Uygulama Ve Testler

Projemizi uygulama kapsamında, 3.Tasarım bölümünde belirttiğimiz metodları kullandık. Ardından projemizi, virtual box üzerinden sanal işletim sistemleri kurarak test ettik. Test işlemlerimizi hem Windows 10 hem de Ubuntu işletim sistemleri üzerinden gerçekleştirdik. Örneğin, bazı JAVA kodları, Eclipse programı kullanılarak, istemci tarafında yazıldı, ardından istemci arayüz programı çalıştırılarak, bu kodlar sunucuya gönderildi, ve ardından RMI Sunucusunda bu kodlar çalıştırılıp, istemci arayüz programına, sonuçları istemcinin makinesine gönderildi. Testlerimizin sonucunda, istemci kodlarının, sunucu üzerinden ne kadar sürede çalıştırılıp geri döndürüldüğü, yazılan kodun uzun ve karmaşık olmasına bağlı olduğunu gözlemledik.

5.Sonuçlar

Sonuç olarak, projemizi daha önce bitirmeyi planladığımız şekilde bitirdik. Öncelikle, COMP4910 dersi kapsamında, danışmanımızla toplantılar yaparak, projemizin ne olması gerektiğini belirledik, daha sonra yazılım ve donanım gereksinimlerimizi belirledik, gerekli yöntemlerin neler olabileceğini araştırdık ve ikinci yarıyılı DJAVAC projesini uygulamaya geçirdik. DJAVAC birden fazla istemciyle çalışmayı sağlar ve bu istemciler birden çok sunucuya bağlanabilir. Bu sistemi uygularken, istemci ve sunucular arasındaki bağlantılarda daha önce «dağıtılmış» pragma olarak adlandırılan bir yöntemi çağırarak için Java RMI kullanılmaktadır. 'Dağıtılmış' pragmaya sahip işlev sunucuya aktarılabilir. Sunucu, istemcinin kodunu derler ve bu kodları JAVA RMI üzerinden istemcinin makinesine döndürmeye hazırdır. Sistemimizi test ettikten sonra, DJAVAC'ın yürütme süresinin, istemcinin kodunun ne kadar uzun ve karmaşık olduğuyla ilgili olduğunu gözlemledik. İstemci kodu TCP / IP ağı üzerinden dağıtıldığında, Sunucu kodunu çalıştırır ve İstemciye geri döndürür. İstemci / Sunucu iletişimi başarıyla sağlandı. İstemciler çok fazla zaman harcamadan daha kaliteli Java kodları yazabilir. Kod üreticileri ve ayrıştırıcı istemciden almak için ek açıklamayı algılamaya izin verir.. Ayrıca, hangi RMI Sunucusunun müsait olduğunu, istemciye o sunucunun IP adresini sorunsuz bir şekilde gönderildiğini gözlemledik. Eğer, Java programcılar çok fazla zaman harcamadan daha kaliteli Java kodları yazmak istiyorsa, DJAVAC kullanmalıdır.