Kocaeli Üniversitesi Yazılım Laboratuvarı Proje 2

Edanur Çövüt 190201035
Fedai Engin Can Yılmaz 200201042

21 Kasım 2022

1 Özet

Bu doküman Yazılım Laboratuvarı-1 dersinin 2. projesi için hazırlanmıştır. Dokümanda projenin tanımı, çözüme yönelik yapılan araştırmalar, kullanılan yöntemler, proje hazırlanırken kullanılan geliştirme ortamı ve kod bilgisi gibi programın oluşumunu açıklayan başlıklar bulunmaktadır. En sonda ise kaynakça kısmı bulunmaktadır.

2 Giriş

Bu projede müşteri şikayetleri kayıtlarının tutulduğu bir veri seti içerisindeki benzer kayıtlar tespit edilecek ve tespit edilen kayıtlar masaüstü uygulamasında gösterilecektir. Multithreading kullanarak benzerlik arama süresini düşürmek amaçlanmaktadır. Amaç:

1. Veri seti içerisindeki arama işlem süresini multithreading kullanılarak azaltmak.
2. Belirtilen sütun/sütunlar için her bir satırdaki kayıtların birbiriyle kelime bazlı karşılaştırılması ve aralarındaki benzerliğin tespit edilmesi.
3. Uygulama içerisinde istenen özelliklere göre kayıtları filtrelemek ve kullanıcıya göstermek.
4. Masaüstü uygulama geliştirme hakkında bilgi ve beceriye sahip olmak.

3 Yöntem

Bu projenin amacı müşteri şikayetleri kayıtlarının tutulduğu bir veri seti içerisindeki benzer kayıtlar tespit edilecek ve tespit edilen kayıtlar masaüstü uygulamasında gösterilecektir. Multithreading kullanarak benzerlik arama süresini düşürmek amaçlanmaktadır.

Thread aynı process ortamında birden fazla iş yürütme imkanı sağlar. Bir process'in çalışmaya başlaması ile birlikte bir thread (main thread) oluşturulur ve process içerisinde birden fazla (multi-thread) oluşturulabilir. Yaratılan threadler aynı networkde ve farklı networkde işlem yapabilirler. Threadler genel olarak yazılım testleri yapılırken simülasyon olarak kullanılabilir ancak bilişim sistemlerinde cpu üzerinden işlem yapan thread fiziki olarak işlem yaptığı için yazılım test senaryosundaki thread gruplardan farklı davranış gösterebilir. Threadler kullanıldığı alanlara göre farklılık gösterebilir. Kısaca Thread (iş parçacığı) kullanımı, birden fazla işlemin tek bir akışı paylaşarak neredeyse eşzamanlı bir şekilde gerçek-

leşmesini sağlar. Bilgisayar mimarisinde, MultiThreading (çok iş parçacıklı), bir merkezi işlem biriminin (CPU) (veya çok çekirdekli bir işlemcideki tek bir çekirdeğin) aynı anda işletim sistemi tarafından desteklenen birden çok yürütme iş parçacığı sağlama yeteneğidir. Bu yaklaşım, çoklu işlemden farklıdır. MultiThreading (çok iş parçacıklı) bir uygulamada, iş parçacıkları, hesaplama birimlerini, CPU önbelleklerini ve çeviri ön tampon tamponunu (TLB) içeren tek veya çok çekirdeğin kaynaklarını paylaşır. Çok işlemcili sistemler, bir veya daha fazla çekirdekte birden çok tam işlem birimi içerdiğinde, çok iş parçacıklı, iş parçacığı düzeyinde paralellik ve aynı zamanda talimat düzeyinde paralellik kullanarak tek bir çekirdeğin kullanımını artırmayı amaçlar. Bu iki teknik birbirini tamamlayıcı nitelikte olduğundan, bazen çoklu çok iş parçacıklı CPU'lara ve birden fazla çok iş parçacıklı çekirdeğe sahip CPU'lara sahip sistemlerde birleştirilirler.
MultiThread Nasıl Çalışır?

Multithreading aynı anda birden fazla iş parçacığı çalıştırmanıza izin verir. Çok çekirdekli bir makinede bu, iki iş parçacığının gerçekten paralel olarak çalışabileceği ve her seferinde bir tane çalıştırdıklarının iki katı işi yapabileceği anlamına gelir. İdeal olarak, 4 çekirdekli bir makinede, 4 iş parçacığı ile, tek bir iş parçacığına kıyasla neredeyse 4 kat daha fazla iş yaparsınız. Bunun işe yaraması için, birbirinden bağımsız çalışan birden çok iş parçacığı ile çözülebilecek bir soruna ihtiyacınız var. Programı nasıl iş parçacığına böleceğinizi bulmak için oldukça zeki olmanız gerekir. Ve çoğu zaman, bu iş parçacıklarının birbirlerinin verilerini çöpe atmasını (veya daha kötüsü, kurnazca sabote etmesini) engellemek için gerçekten zeki olmanız gerekir. Bazen konuları aynı programın farklı oturumları gibi neredeyse bağımsız olarak çalıştırmak mümkün olsa da, bunu yapabildiğiniz zaman güzel bir şeydir.

4 Sonuç

- Verilen veri seti istenen şekilde yeniden düzenlenmiştir.
- Düzenlenmiş veri setindeki kayıtlar arasında benzerlik kontrolü yapılmıştır. Kontrol sırasında multithreading kullanıldı. Multithreading için kullanılan thread sayısı uygulama arayüzünden giriliyordur.
- Her thread'in çalışma zamanı ve tüm thread'ler için toplam çalışma zamanı bilgileri uygulama arayüzünde gösteriliyor.
- İstenilen sütun ya da sütunlar arasındaki girilen benzerlik oranı (threshold) ve üzerinde benzerliğe sahip kayıtlar masaüstü uygulamasında gösteriliyor.
- Uygulamayı sunmak üzere basit bir arayüz geliştirildi. Bu arayüz aşağıdakileri içeriyor:
  - Benzerlik oranının (Threshold değeri) seçilebileceği / girilebileceği bir araç,
  - Benzerliklerinin araştırılması istenen sütun veya sütunların seçilebileceği bir araç,
  - Kaç tane thread kullanılacağının seçilebileceği / girilebileceği bir araç,
  - Her bir thread'in çalışma zamanını ve toplam çalışma zamanını gösteren araçlar
  - Sonuçların açıkça ekranda gösterilebileceği bir araç.

- Uygulamada aşağıdaki ve benzer senaryolardan elde edilen sonuçlar ekranda gösteriliyor:
  - Senaryo 1:Ürün (Product) sütununda ekranda gösteriyor.
  - Senaryo 2:Aynı ürünler (Product) için (Issue) içeren Şirket (Company) isimlerini ekranda gösteriyor.
  - Senaryo 3: 'Complaint Id' = 3198084 olan şikayet kaydı için benzerlikteki konuları (issue) içeren kayıtları ekranda gösteriyor.
  - Senaryo 4: 5 Thread ile Konular(Issue) sütununda kayıtları ekranda gösteriyor.

5 Deneysel Sonuçlar

Proje ile ilgili deneysel sonuçlar aşağıdadır:

5.1 Geliştirme Ortamı

Projeyi Windows sistemde, Visual Studio üzerinde geliştirip yine Visual Studio kullanarak derledik.

5.1.1 İstatistik

Program kodu 4 farklı c(sharp) dosyası içinde toplamda 537 satır koddan oluşmaktadır. Kodu derleyebilmek ve derlerken kolaylık olsun diye toplamda 20 yorum satırı kullanılmıştır. Programı derlemek için kullandığımız kütüphaneler:

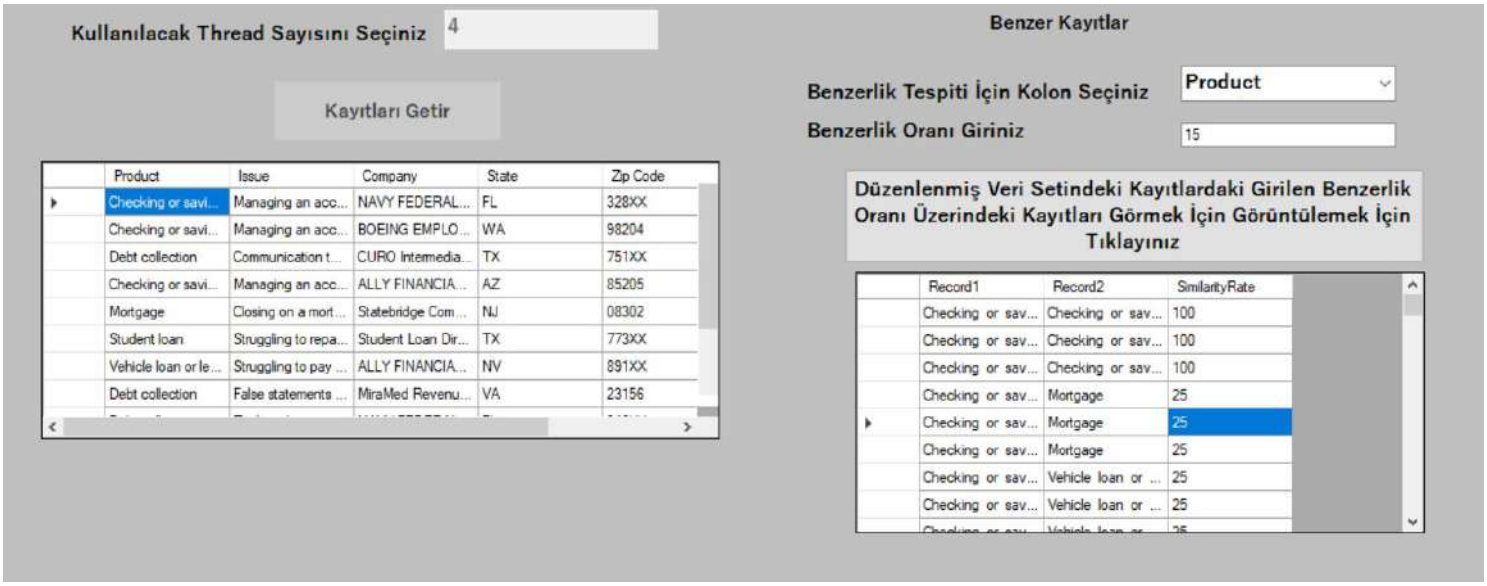
1. System.io=: Yaygın olarak kullanılan değer ve başvuru veri türleri, olaylar ve olay işleyicileri, arabirimler, öznitelikler ve işleme özel durumlarını tanımlayan temel sınıfları ve temel sınıfları içerir.
2. System.Collections.Generic: Kullanıcıların, genel olmayan türü kesin olarak belirlenmiş koleksiyonlardan daha iyi tür güvenliği ve performansı sağlayan kesin türü belirlenmiş koleksiyonlar oluşturmasına olanak tanıyan genel koleksiyonları tanımlayan arabirimler ve sınıflar içerir.
3. System.ComponentModel: Bileşenlerin ve denetimlerin çalışma zamanı ve tasarım zamanı davranışını uygulamak için kullanılan sınıflar sağlar. Bu ad alanı öznitelikleri ve tür dönüştürücülerini uygulamaya, veri kaynaklarına bağlamaya ve lisans bileşenlerini uygulamaya için temel sınıfları ve arabirimleri içerir.
4. System.Data: Birden çok veri kaynağından verileri verimli bir şekilde yöneten bileşenler oluşturmanızı sağlar.
5. System.Drawing: Temel grafik GDI+ erişim sağlar.
6. System.Linq: Language-Integrated Query (LINQ) kullanan sorguları destekleyen sınıflar ve arabirimler sağlar.
7. System.Text: ASCII ve Unicode karakter kodlamalarını temsil eden sınıfları içerir.
8. System.Threading: Bir iş parçacığını oluşturur ve kontrol eder, önceliğini ayarlar ve durumunu alır.
9. System.Threading.Tasks
10. System.Windows.Forms: Microsoft Windows işletim sisteminde bulunan zengin kullanıcı arabirimi özelliklerinden tam olarak yararlanan Windows tabanlı uygulamalar oluşturmak için sınıflar içerir.

5.1.2 Programın Derlenmesi

Programın kaynak kodu 4 dosyadan oluşmaktadır. Bu dosyayı Visual Studio ile derleyebilirsiniz.

6 Kaynakça

- https://mertmekatronik.com/thread-ve-multithread-nedir
- https://www.tutorialspoint.com/operating
- https://www.javatpoint.com/multithreading-in-java
- https://totalview.io/blog/multithreading
- https://www.geeksforgeeks.org/multithreading-python-set-1/
- https://dotnettutorials.net/lesson/multithreading-in-csharp/
- https://www.partech.nl/nl/publicaties/2021/02/multi-threading-in-c-sharp
- https://thecodeprogram.com/c–ile-multi-thread-calisma—-cok-kanalli-calisma
- https://www.tutorialspoint.com/csharp/csharp_multithreading.htm
- https://www.geeksforgeeks.org/c-sharp-multithreading/
- https://www.geeksforgeeks.org/how-to-create-threads-in-c-sharp/
- https://www.learncsharptutorial.com/threading-and-types-of-threading-stepbystep.php
- https://www.c-sharpcorner.com/article/multithreading-in-c-sharp-net2/


Kullanılacak Thread Sayısını Seçiniz
4
Kayıtları Getir
Benzer Kayıtlar
Benzerlik Tespiti İçin Kolon Seçiniz
Product
Benzerlik Oranı Giriniz
15
Düzenlenmiş Veri Setindeki Kayıtlardaki Girilen Benzerlik Oranı Üzerindeki Kayıtları Görmek İçin Görüntülemek İçin Tıklayınız
Product
Issue
Company
State
Zip Code
Checking or savi...
Managing an acc...
NAVY FEDERAL...
FL
328XX
Checking or savi...
Managing an acc...
BOEING EMPLO...
WA
98204
Debt collection
Communication t...
CURO Intermedia...
TX
751XX
Checking or savi...
Managing an acc...
ALLY FINANCIA...
AZ
85205
Mortgage
Closing on a mort...
Statebridge Com...
NJ
08302
Student loan
Struggling to repa...
Student Loan Dir...
TX
773XX
Vehicle loan or le...
Struggling to pay ...
ALLY FINANCIA...
NV
891XX
Debt collection
False statements ...
MiraMed Revenu...
VA
23156
Record1
Record2
SimilarityRate
Checking or sav...
Checking or sav...
100
Checking or sav...
Checking or sav...
100
Checking or sav...
Checking or sav...
100
Checking or sav...
Mortgage
25
Checking or sav...
Mortgage
25
Checking or sav...
Mortgage
25
Checking or sav...
Vehicle loan or ...
25
Checking or sav...
Vehicle loan or ...
25