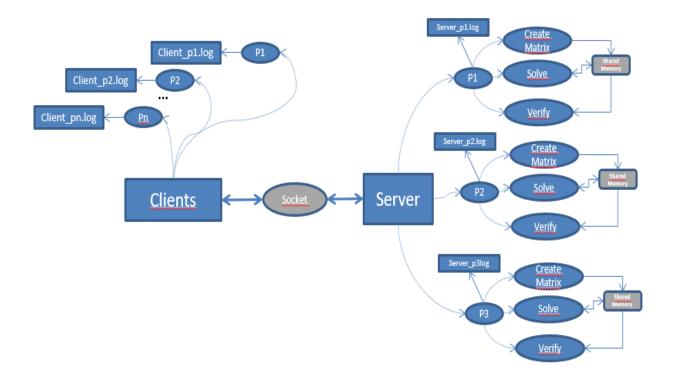
Gebze Technical University Computer Engineering

CSE 244 2017 Spring

FINAL PROJECT REPORT

Elif Şeyma ARMAĞAN 151044042

Course Lecturer: **Prof. Dr. Erkan ZERGEROĞLU**



Projeyi Client modülü ve Server modülü olmak üzere 2 modül olarak tasarladım.

Client uygulaması kullanıcıdan matrix boyutunu önce row sonra col olarak, client uygulamasındaki başlatılacak alt process sayısını ve server ile haberleşmelerini sağlayacak port bilgisini almaktadır. Client uygulaması bu şekilde server ile tanımlanan port üzerinden haberleşmektedir. Bu port üzerinden server ile bağlantı kurulduktan sonra her bağlantı için bir thread oluşturulmaktadır. Thread ler arasındaki erişim koordinasyonunu sağlayabilmek için mutex kullanılmaktadır. Oluşturulan her thread öncelikle process id ve matris boyutuna ilişkin bilgileri socket üzerinden server uygulamasına iletmektedir. Daha sonra kendisi için yaratılan server processinden oluşturulan matrisi ve b vektörünün elemanlarını socket üzerinden almaktadır. Aldığı bu bilgileri oluşturulan log dosyasına yazmaktadır.

Server uygulaması ise dinleme yapılacak port bilgisi ve thread pool sayısını kullanıcıdan almaktadır. Eğer aldığı bu sayı 1 veya 0 ise thread-per-request, daha fazlaysa worker pool tabanlı kurulan server çalıştırılmaktadır. Öncelikle kullanıcı tarafından belirtilen port üzerinden dinleme yapabilmek için bir socket oluşturulmaktadır.

Oluşturulan bu socket per- request yöntemi ile oluşturulduysa sonsuz bir döngü ile dinlenerek her bağlantı talebi için ayrı bir thread oluşturulmaktadır. Worker pool yöntemi ile oluşturulduysa threadler başta oluşturulur ve her thread request bekler.

Oluşturulan her thread öncelikle matris boyutunu ilgili client processinden socket aracılığıyla almaktadır. Bu bilgiyi aldıktan sonra thread e özel shared memory alanı oluşturmaktadır. Thread shared memory alanını oluşturduktan sonra matris oluşturan P1, matrisi çözen P2 ve doğruluğu kontrol eden P3 processleri oluşturulmaktadır.

P1 process i ile çözüm oluşturulacak A matrisi ve B vektörü rastgele değerlerden oluşturulmaktadır.. Bu process oluşturduğu matris ve vektörü shared memory alanına ve socket üzerinden client process ine bildirmektedir. Bu yazma işlemi sırasında erişim kontrolü thread için oluşturulan mutex aracılığyla yapılmaktadır. Ayrıca oluşan bu matrisi her client process i için ayrı ayrı oluşturulan log dosyasına yazmaktadır.

P2 process i ile P1 process i tarafından shared memory alanı üzerinde oluşturulan A matrisi ve B vectörü okunmakta ve yerel bir array e atanmaktadır. Bu işlem sonrasında bu problemi üç farklı şekilde çözen uygulamalar çağrılmaktadır.

P3 process i P2 tarafından oluşturulan çözümlerin doğruluğu kontrol ederek oluşan «e» değerini hesaplamaktadır.

Eğer server daha oluşmamışsa clientlar çıkmaktadır. CTRL-C sinyali hem server için hem de client için handle edilmiştir.

Server a CTRL-C sinyali geldiğinde tüm clientlar öldürülmektedir.

Hem thread-per-request hem de worker-pool yöntemleri için client sayısını 50 ye kadar deneyince bir sorun çıkmadan çalışmaktadır.

Eksik ve hatalı kısımlar

- Uzun araştırmalarım sonucu SVD, QR Factorization ve pseudo inverse yöntemlerine ait kodu internette bulamadım. Kendim implement etmeye de zamanım kalmadığı için bu yöntemleri kodumda kullanamadım. Bunun yerine o kısımların boş kalmaması için belli matematiksel işlemler yapıp random sonuçlar ürettim. Bu nedenle bazı durumlarda inf ve nan lar ile karsılaştım.
- Bu yöntemleri implement edemediğim için hata hesabı yapmamın bir anlamı olmayacağından dolayı P3 de de random sonuçlar ürettim.
- Client ta handle edilen CTRL-C sinyalinin tüm clientlara iletilmesi kısmı tam olarak çalışmamaktadır.
- Client log dosyasında standart deviation bulunmamaktadır.

Örnek Çalışma Durumu ve Oluşturulan Log Dosyaları

