

GEBZE TEKNİK ÜNİVERSİTESİ

Sistem Programlama Final Raporu

Rıdvan Demirci

141044070

Prof. Dr. Erkan ZERGEROĞLU

Problem Çözüm Yolu

./server <portNumarası> <poolSize>

Öncelikler pool size sıfıra eşit veya küçük ise Server'ın **thread-per-request** implementasyonu çalışır ,sıfırdan büyük olduğu durumlarda ise **worker pool** aktif olur.

.thread-per-request:

bu kısım için socket den gelen her request için bir tane thread oluşturulur ve bu thread portdan özel socket numarası alır.Üç adet proccess oluşturup matrix işlemlerini yapar.

.worker pool:

Bu bölüm ise verilen pool size kadar thread oluşturulur bu thread sonsuz döngüdedir.bir tane global olarak Queue tutulur gelen her request bu Queue içine atılır.Threadler ise sürekli sürekli bu Queue size kontrolü yapar ve size sıfırdan büyük ise oradaki socket numarasını alıp oradan işlemler yapar.

./client <row> <col> <threadSize> <portNum>

Bu programda verilen size kadar thread oluşturulur ve hepsi semaphore içinde connect olur.servera kendi thread numarası ve satır sutun numaralarını gönderir.thread id gönderme sebebi ise log dosyalarında karşılaştırma yapılabilmesidir.

Her client çalışma süresi global bir arraye alınır ve threadlerin işi bittikten sonra avarage time ve standart sapma hesaplanır.ayrıca servera kend pid numarasını da gönderir ve bu sayede server bittiginde kill sinyali gönderebilir.

Test Edilen Durumlar

Test Case	Sonuc	süre
Server thread-per-request ile çalıştırmak,yani poolSize=0, client da ise 5X4 lük 4 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	85.42 ms
Server worker pool ile çalıştırmak,yani poolSize=4, client da ise 5X4 lük 4 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	53.95 ms
Server worker pool ile çalıştırmak,yani poolSize=2, client da ise 5X4 lük 4 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	54.55 ms
Server worker pool ile çalıştırmak,yani poolSize=2, client da ise 5X4 lük 8 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	113.04 ms
Server thread-per-request ile çalıştırmak,yani poolSize=0, client da ise 5X4 lük 8 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	127.1 ms
Server worker pool ile çalıştırmak,yani poolSize=10, client da ise 5X4 lük 8 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	125.72 ms

Server thread-per-request ile çalıştırmak,yani poolSize=0, client da ise 7X7 lük 10 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	257.90 ms
Server worker pool ile çalıştırmak,yani poolSize=1, client da ise 7X4 7 lük 10 tane thread bağlamak	Program sadece bir kaç tane requeste cevap verdi.tam çalışmadı. (deadlock durumu)	-
Server worker pool ile çalıştırmak,yani poolSize=2, client da ise 7X7 lik 10 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	254.01 ms
Server worker pool ile çalıştırmak,yani poolSize=2, client da ise 2X2 lik 1 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	10.13 ms
Server thread-per-request ile çalıştırmak,yani poolSize=0, client da ise 2X2 lik 1 tane thread bağlamak	Program düzgünce çalıştı ve Log dosyaları oluşturuldu.	10.34 ms
Server worker pool ile çalıştırmak,yani poolSize=2, client da ise 7X7 lik 10 tane thread bağlamak ve client çalışırken SIGINT sinyali gönderme	Program düzgünce çalıştı ve Log dosyaları oluşturuldu. Bekleneni verdi.	-

LOG DOSYALARI

SERVER LOG DOSYASI

```
1
2 newClient
3 threadNum 139775383107328
4 A matrixi
5 19.000000 18.000000 30.000000 31.000000 19.000000 59.000000 25.000000
6 4.000000 57.000000 13.000000 69.000000 35.000000 70.000000 68.000000
7 1.000000 38.000000 19.000000 12.000000 5.000000 33.000000 53.000000
8 15.000000 69.000000 26.000000 52.000000 8.000000 68.000000 57.000000
9 31.000000 45.000000 55.000000 49.000000 5.000000 14.000000 9.000000
10 35.000000 3.000000 45.000000 39.000000 1.000000 69.000000 49.000000
11 35.000000 11.000000 46.000000 47.000000 48.000000 6.000000 58.000000
12
13 B matrixi
14 17.000000
15 57.000000
16 57.000000
17 20.000000
18 32.000000
19 41.000000
20 39.000000
21
22 result1
23 -4.543764
24 -1.090788
25 3.010042
26 1.116475
27 -0.523714
28 -0.196076
29 0.782912
30
31 result2
32 -4.543764
33 -1.090788
34 3.010042
35 1.116475
36 -0.523714
37 -0.196076
38 0.782912
39
```

CLIENT LOG DOSYASI

```
1 Gonderilen Thread 139775383107328
2 matrixOfA
3 19.000000 18.000000 30.000000 31.000000 19.000000 59.000000 25.000000
4 4.000000 57.000000 13.000000 69.000000 35.000000 70.000000 68.000000
5 1.000000 38.000000 19.000000 12.000000 5.000000 33.000000 53.000000
6 15.000000 69.000000 26.000000 52.000000 8.000000 68.000000 57.000000
7 31.000000 45.000000 55.000000 49.000000 5.000000 14.000000 9.000000
8 35.000000 3.000000 45.000000 39.000000 1.000000 69.000000 49.000000
9 35.000000 11.000000 46.000000 47.000000 48.000000 6.000000 58.000000
10
11 matrixOfB
12 17.000000
13 57.000000
14 57.000000
15 20.000000
16 32.000000
17 41.000000
18 39.000000
19
20 result1
21 -4.543764
22 -1.090788
23 3.010042
24 1.116475
25 -0.523714
26 -0.196076
27 0.782912
28
29 result2
30 -4.543764
31 -1.090788
32 3.010042
33 1.116475
34 -0.523714
35 -0.196076
36 0.782912
37
38 result3
39 0.000000
```

server log dosyasına da client dan gonderenin Id si basıldı böylece karışlaştırılabilir.

Ekran çıktısı:

```
oem@oem-X555UB ~/Desktop/c/Sistem Programlama/final/code $ ./server 8080 2
portNo 8080 size 2
Hizmet Edilen Thread Sayısı 0
WORKER POLL
Hizmet Edilen Thread Sayısı 1
Hizmet Edilen Thread Sayısı 2
  Hizmet Edilen Thread Sayısı 3
Hizmet Edilen Thread Sayısı 2
  Hizmet Edilen Thread Sayısı 2
Hizmet Edilen Thread Sayısı 1
  Hizmet Edilen Thread Sayısı 2
Hizmet Edilen Thread Sayısı 1
  Hizmet Edilen Thread Sayısı 2
Hizmet Edilen Thread Sayısı 1
  Hizmet Edilen Thread Sayısı 3
Hizmet Edilen Thread Sayısı 2
  Hizmet Edilen Thread Sayısı 2
Hizmet Edilen Thread Sayısı 1
  Hizmet Edilen Thread Sayısı 1
Hizmet Edilen Thread Sayısı 2
  Hizmet Edilen Thread Sayısı 3
Hizmet Edilen Thread Sayısı 2
  Hizmet Edilen Thread Sayısı 1
  Hizmet Edilen Thread Sayısı 0
  Hizmet Edilen Thread Sayısı 0

fname = ClientLog/19988
37.000000
26.000000
19.000000
13.000000
60.000000
33.000000
64.000000

fname = ClientLog/19988
fname = ClientLog/19988
fname = ClientLog/19988
Gecen Süre 421.972992
err 1 0.000082

err 2 0.000082

err 3 2187.682335

fname = ClientLog/19988
fname = ClientLog/19988
36.000000 70.000000 11.000000 50.000000 60.000000 49.000000 51.000000
43.000000 5.000000 30.000000 14.000000 20.000000 55.000000 58.000000
65.000000 41.000000 57.000000 31.000000 5.000000 49.000000 19.000000
60.000000 69.000000 1.000000 64.000000 47.000000 28.000000 46.000000
16.000000 2.000000 41.000000 52.000000 14.000000 63.000000 43.000000
15.000000 42.000000 24.000000 58.000000 58.000000 53.000000 13.000000
19.000000 49.000000 13.000000 14.000000 32.000000 69.000000 56.000000

fname = ClientLog/19988
62.000000
1.000000
33.000000
52.000000
30.000000
16.000000
37.000000

fname = ClientLog/19988
fname = ClientLog/19988
fname = ClientLog/19988
Gecen Süre 467.557007
fname = ClientLog/19988
ortalama zaman 257.834899
Standarsapma 301.565740oem@oem-X555UB ~/Desktop/c/Sistem Programlama/final/code $
```

Programlar Hakkında

programlar kendi içinde bulunan makefile lar ile çalıştırılabilir.

İçersinde SVD hesaplama var hazır aldığım bir kod.

*** Programlar yukarıda ki test caseler ile denendi.**

***ancak program çalıştırılırken bazen kitlenip karlıyo bazende port Numarasını silmek gerekiyo.**

***ilk durumda çalıştırılırken genelde çalışıyo.**

***ama bazen kitlene biliyo bu durumda tekrar başlatılmalı**

Log dosyaları için client ve server için ayrı ayrı oluşturuldu.
***program başlatılmadan önce bu directoriler silinirse daha iyi olur.**

Kaynakça:

<https://github.com/baiyangbupt/SVD-Decomp>

<http://bilgisayarkavramlari.sadievrenseker.com/2010/04/02/matris-carpimi-matrix-multiplication>

<http://bilgisayarkavramlari.sadievrenseker.com/2008/11/19/matrisin-tersinin-alinmasi-mantrix-inverse/>

<http://bilgisayarkavramlari.sadievrenseker.com/2008/04/16/sira-queue>