BIL 244 FİNAL RAPOR

Bu projede İki adet server programı ve bir adet clients programı implement edildi.Server programlarından biri Thread-per-request olarak tasarlandı, diğeri ise Tthrad-pool olarak tasarlandı.

Dosyalar:

server.c (thread-per-request-server)
server2.c(thread pool server)
clients.c
methods.c
methods.h

Kullanım:

./server <port #, id>
./server2 <port #, id> <thpoolsize , k>
./clients <port #, id> <# column , m> <# row, p> <#clients, q>

Algoritma:

Clients programı içerisinde argüman olarak grilen sayı kadar detachable thread oluşturulur.Ve bu threadlerin hepsi paralel olarak server'a istekte bulunur.Server'a istekte bulunduğu kısım için semafor kullandım.aslında bunu kullanmama gerek yoktu.Çünkü soketin kendi kuyruğu var ama kullanmamız istendiği için kullandım.
İki farklı server implement edildi.Thread per request serverda her her gelen istek için bir client oluşturur.Client için oluşan bu thread kendi içinde üç ayrı child process oluşturur.Bu prosesler aynı anda oluşturulur ve paralel olrak çalışırlar.Proseslerden ilki A matrsini ve b vektörünü random olarak oluşturur ve proses 1 ve proses 2 arasındaki shared memory bloğuna yazar.bu sırada 2.

proses 1. prosesin oluşturduğu matris ve vektörü shared memory bloğuna yazmasını bekler.Bu bekleme işini IPC semafor setleri kullanarak hallettim.2. proses shared memoryden aldığı A matrisi ve b vektörlerinden Ax = b denklemini üç farklı metodla çözmek için 3 ayrı thread oluşturur.Bu metodlar Singular Value Decomposition, QR decomposition ve Pseudo inverse metotlarıdır.QR decomposition implementasyonunu bulamadığım için ve kendim yazmaya da zamanım olmadığı için QR sadece rectangular matrisxin kare olan kısmını çözer ve diğer ksımları truncate eder.Bu yüzden error norm bu metod için biraz fazla çıkmaktadır.Bu üç metodla bulunan x vektörleri A matrsi ve b vektörü 2. ve 3. proses arasındaki shared memory bloğuna yazılır. 3 .proses ise shared memorydeki bu bilgileri alarak 3 metod içinde error hesaplar ve bu bilgileri her cilienta spesifik(process id ve thread id) olan
log fila yazar ve clienta gönderir.Clientda aldğı bu bilgileri kendi loguna yazar.
Clients programı ayrıca bütün clientların bağlanma zamanını ve standard sapmasını mikrosaniye resolutionunda hesaplayarak log file'a yazar. Server programına SIGINT sinyali geldiğinde server kapanır ve kapanma zamanını ekrana yazdırır.

Eksikler:

-- Thread per request serverından Ctr+C ile çıkıldığında herhangi bir memory leak kalmamaktadır fakat still reachable bir blok kalmaktadır.
--Thread pool serverı Ctrl+c sinyali aldığında kapnır.Fakat memory leak kalmaktadır.Bunu halledemedim.Yeterince zamanım yoktu.

Bazı testler

Thread-per-request 10 x 10 matris 5 client

Average Connection Time = 5119.0000 microseconds
Standard devition of overall run = 1392.1061 microseconds

Thread-per-request 10 x 10 matris 10 client

Average Connection Time = 11962.6000 microseconds
Standard devition of overall run = 5325.0472 microseconds

Thread-per-request 10 x 10 matris 15 client

Average Connection Time = 19136.2000 microseconds
Standard devition of overall run = 8258.5628 microseconds

Thread-per-request 10 x 10 matris 20 client

Average Connection Time = 20204.6000 microseconds
Standard devition of overall run = 10596.6629 microseconds

Thread pool server 10 x 10 matris thpool size 5 client

Average Connection Time = 6355.4000 microseconds
Standard devition of overall run = 1398.8238 microseconds

Thread pool server 10 x 10 matris thpool size 10 client

Average Connection Time = 10476.4000 microseconds
Standard devition of overall run = 3968.5456 microseconds

Thread pool server 10 x 10 matris thpool size 15 client

Average Connection Time = 17480.2667 microseconds
Standard devition of overall run = 6529.9791 microseconds

Thread pool server 10 x 10 matris thpool size 20 client

Average Connection Time = 26255.3000 microseconds
Standard devition of overall run = 12576.4720 microseconds