

CS 580 Client-Server Programming
Spring Semester, 2010
Doc 12 Thread Pools
8 March, 2010

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

Thread Pool Pattern

Thread Pooling

Group of threads created to perform a number of tasks

A thread

- Reads a task from a queue

- Performs the task

- Repeat

Server Options

Iterative Server - server handles one client at a time

Concurrent Server with Thread creation

 Create new thread for each client

Concurrent Server with Thread Pool

Concurrent Server with expandable Thread Pool

Single thread handles multiple clients concurrently

Iterative Server - When to use

Iterative Server

When usable

```
while (true)
{
  Socket client = serverSocket.accept();
  Sequential code to handle request
}
```

TP = Time to process a request

A = arrival time between two consecutive requests

Then we need $TP \ll A$

Concurrent Server with Thread creation

Basic Concurrent Server

```
while (true)
{
    Socket client = serverSocket.accept();
    Create a new thread to handle request
}
```

When usable

Let TC = time to create a thread

Let A = arrival time between two consecutive requests

We need $TC \ll A$

Often this is good enough

Problem with Threads

Thread consume resources

- Memory

- CPU cycles

A program has a limit of

- Threads it can productively support

- Sockets it can have open

We need to insure we don't create too many threads

Concurrent Server with Thread Pool

```
Create N worker threads
while (true)
{
    Socket client = serverSocket.accept();
    Use an existing worker thread to handle
request
}
```

When usable

TP = Time to process a request

A = arrival time between two consecutive requests

N = Thread Pool size

Then we need $TP \ll A * N$

Concurrent Server - expandable Thread Pool

```
Create N worker threads
while (true)
{
    Socket client = serverSocket.accept();
    if worker thread is idle
        Use an existing worker thread to handle
request
    else
        create new worker thread to handle the
request
}
```

When usable

Number of requests we can handle in a unit of time

$$TP / N + 1/TC$$

where N is not constant

Thread Pool Issues

How many threads?

When to create some threads?

When to destroy some threads?

Java ThreadPool Classes

`java.util.concurrent.ExecutorService`

Simple interface

Uses 3 common configurations for the pool

`java.util.concurrent.ThreadPoolExecutor`

Used by `ExecutorService`

Configurable

ExecutorService Example

```
class Server extends Thread {
    private final ServerSocket serverSocket;
    private final ExecutorService pool;

    public Server(int port, int poolSize)
        throws IOException {
        serverSocket = new ServerSocket(port);
        pool = Executors.newFixedThreadPool(poolSize);
    }

    public void run() {
        try {
            for (;;) {
                pool.execute(new Handler(serverSocket.accept()));
            }
        } catch (IOException ex) {
            pool.shutdown();
        }
    }
}
```

```
class Handler implements Runnable {
    private final Socket socket;
    Handler(Socket socket) {
        this.socket = socket;
    }

    public void run() {
        // process request
    }
}
```