Thread Pools and Work Queues

CS 272 Software Development

Professor Sophie Engle
Department of Computer Science

Thread States

```
┌─────────────┐      ┌─────────────┐      ┌─────────────┐
│     New     │ ───▶ │  Runnable   │ ───▶ │ Terminated  │
└─────────────┘      └─────────────┘      └─────────────┘

┌─────────────┐      ┌─────────────┐      ┌─────────────┐
│   Waiting   │      │Timed Waiting│      │   Blocked   │
└─────────────┘      └─────────────┘      └─────────────┘
```

https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/lang/Thread.State.html

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO
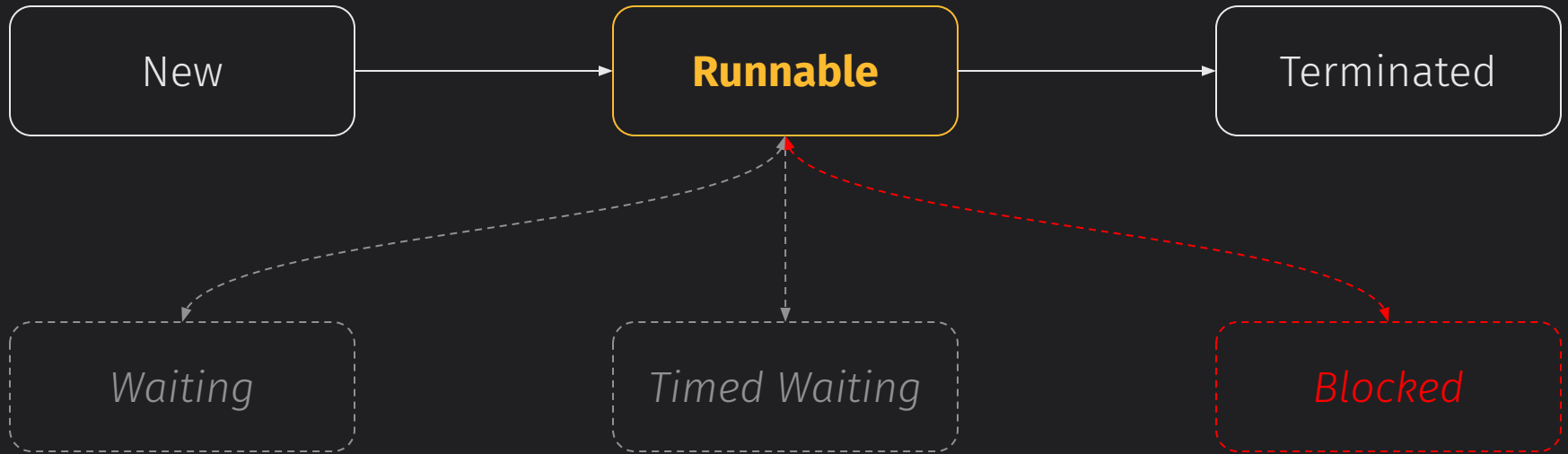
Motivation

- Goal: Web Server
  - Must handle multiple simultaneous requests
  - Must be **responsive** AND **efficient**
    (e.g. respond quickly, finish quickly)

- Implementation: Multithreading
  - One thread per request?

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Problems

- Overhead cost to **creating objects**
  - Initialization in constructor (and `super()` calls)

- Overhead cost to **destroying objects**
  - Garbage collection

- Overhead cost to **excessive memory usage**
  - Causes thrashing

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Solutions

- Keep threads around
  - Initialize a "wise" number of threads once
  - Reuse threads for other tasks instead of destroying

- Two-part approach
  - Thread pool (efficiency)
  - Work queue (responsiveness)

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Thread Pool

- Create a fixed number of worker threads

- When have work to do…
  - Get available thread from pool and assign task
  - Thread runs assigned task
  - Thread returns to pool of available threads

- What if there are no available threads?

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Thread Pool

```java
private final Worker[] workers;

public ThreadPool(int threads) {
  this.workers = new Worker[threads];

  for (int i = 0; i < threads; i++) {
    workers[i] = new Worker();
    workers[i].start();
  }
}
```

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Work Queue

- Add a work queue to thread pool
  - Queue of `Runnable` objects

- Workers check for work in FIFO queue
  - If work... worker removes and runs the work
  - If no work... worker waits until queue is not empty

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Work Queue

```java
private final Worker[] workers;
private final LinkedList<Runnable> tasks;

public WorkQueue(int threads) {
  this.workers = new Worker[threads];
  this.tasks = new LinkedList<Runnable>();

  …
}
```

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Work Queue

```
public run() { // VERY simplified logic
  while (true) { // infinite loop
    while (tasks.isEmpty()) { // wait for work
      tasks.wait();
    }

    Runnable task = tasks.removeFirst();
    task.run(); // run oldest task
  }
}
```

Worker run() logic

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Work Queue

- Main thread no longer manages worker threads

- Main thread only concerned about adding work to the queue and moving on to the next request

- Allows main thread to primarily respond to requests (but not fulfill those requests)

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Keeping Threads Around...

- Thread Pools
  - Basically an array of threads that sticks around
  - Simple, but causes blocking

- Work Queues
  - Adds a queue of "work" (runnable objects)
  - More complicated, but responsive

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Resources

Java Theory and Practice: Thread Pools and Work Queues

Brian Goetz on IBM Developer (2002)

https://www.ibm.com/developerworks/library/j-jtp0730/

Introduction to Java Threads

Brian Goetz on IBM Developer (2002)

https://developer.ibm.com/tutorials/j-threads/

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

Package java.util.concurrent.*

- Includes thread pool and work queue implementations

- Includes thread-safe data structures

- Related packages also include:
  - Read/write lock implementations
  - Atomic versions of Boolean, Integer, etc.

https://www.cs.usfca.edu/~cs272/javadoc/api/java.base/java/util/concurrent/package-summary.html

CS 272 Software Development
Professor Sophie Engle

Department of Computer Science
https://www.cs.usfca.edu/

UNIVERSITY OF
SAN FRANCISCO

USF ◈ UNIVERSITY OF SAN FRANCISCO

CHANGE THE WORLD FROM HERE