
Project Loom

Disclaimer

- Changing prototype implementation
- Changing APIs (including deprecations)

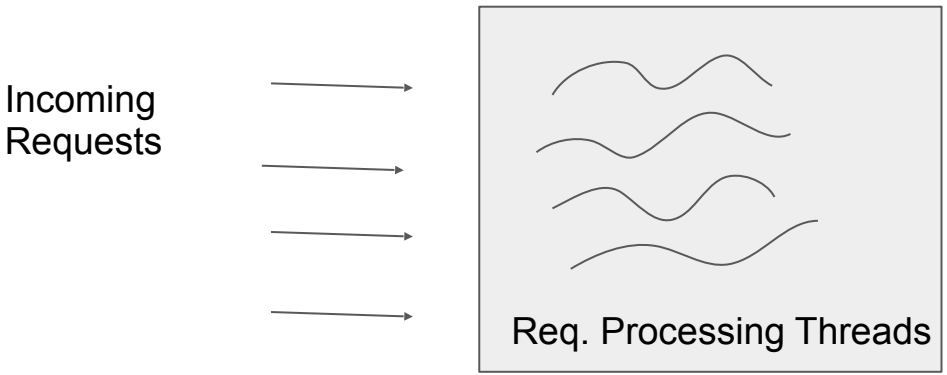
- I do not fully understand Loom

Nawazish Khan

Machine Vs Developer

- Cores in the machine has increased, dramatically.
- Threads as langugage construct to fully utilize machine cores.

Synchronous processing



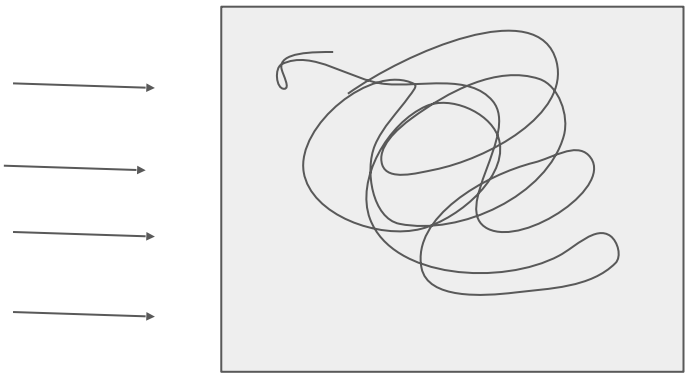
Developer :) | **Machine :(**

Machine Vs Developer

Asynchronous processing

- Callback hell

Incoming
Requests



Developer :(| **Machine :)**

Loom: Virtual Threads

- **Developer makes synchronous calls**
- **Runtime makes it asynchronous.**

- **Virtual Threads: Developer :) + Machine :)**
  - **Light weight/user-mode threads with ability to park/unpark.**

Loom: Virtual Threads

Virtual Threads = **Continuation** + Scheduler

```
public class Continuation implements Runnable{

        public Continuation (ContinuationScope, Runnable)

        public final void run()

        public static void yield (ContinuationScope)

        public boolean isDone()

}
```

LockSupport

```
Public class LockSupport{
  Public static void park(...){
        var t =
Thread.currentThread();

if(t.isLightweight())

Continuation.yield(...);
                  else

Unsafe.park(...)
    }
}
```

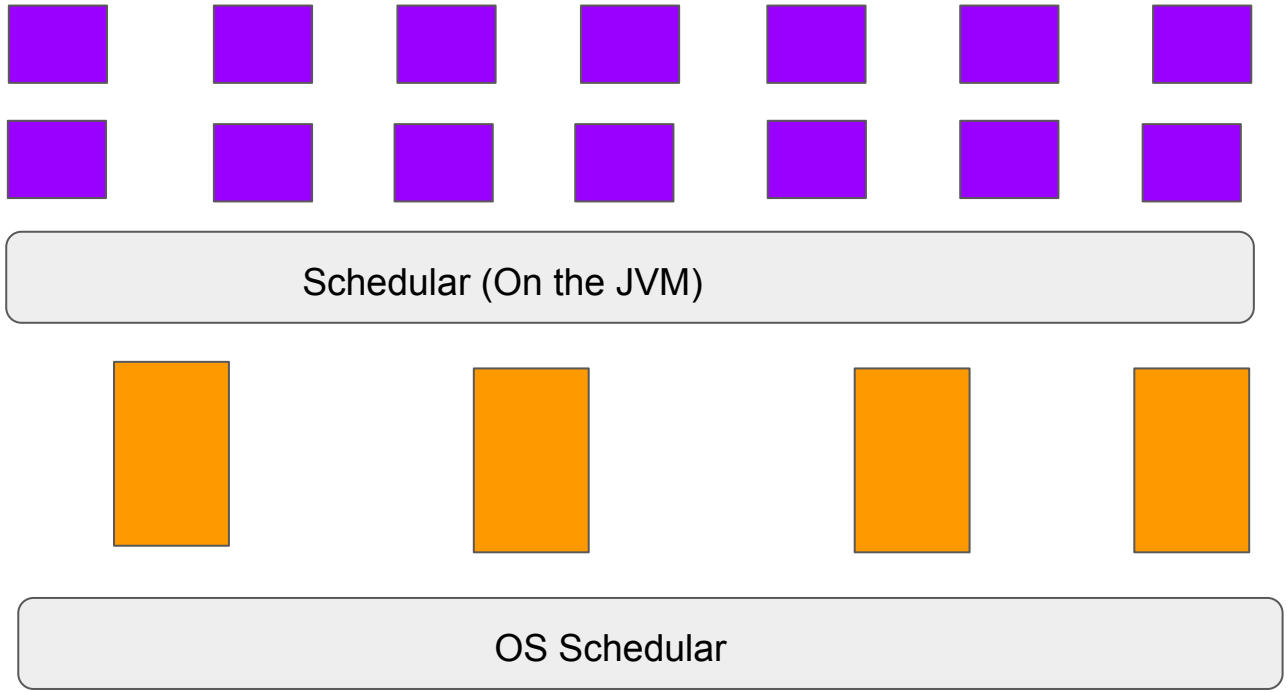
```
Public static void unpark(Thread t){
    if(t.isLightweight())
        t.scheduler.submit(t.continu
    ation)
        else
                Unsafe.unpark(t)
    }
```

Loom: Virtual Threads

Virtual Threads = Continuation + **Scheduler**

- **Fork/Join Pool**

Virtual Threads = **Continuation** + **Scheduler**



Thank you for your patience...