Event Loops in Practice

July 19, 2017

Review Quiz

Which of these is not a common method for speeding up an application?

A. Threading

B. Forking / Subprocesses

C. Compression / Gzipping

D. Event Loops / Async programming

What is a benefit of threading?

A. When one thread is waiting for IO, another thread can be doing work

B. When one thread is waiting for IO, other threads can speed up that IO operation

C. When one thread is executing, another thread can be doing garbage collection

D. When one thread is executing, another thread can preserve memory

What is a disadvantage of threading?

A. Threads can be confusing for the kernel scheduler, causing segfaults

B. Threads take up too much memory to speed up single threaded programs

C. Multiple threads can modify the same memory, causing consistency errors

D. Threads can operate too quickly for some processors, harming hardware

What is an advantage of event loops / async programming?

A. Event loops allow multiple parts of your application to execute simultaneously, increasing responsiveness

B. Event loops efficiently use a single thread, blending responsiveness without the danger of threading

C. Event loops increase the efficiency of the language's looping operators (for, while, do) increasing speed

D. Event loops execute parts of a program on the GPU, using more hardware effectively

What is an disadvantage of event loops / async programming?

A. Event loops allocate too much stack memory, which can cause programs to terminate

B. Event loops use too many threads, which can interfere with other programs running on the system

C. Event loops require writing programs in a different, "inside out" way, which can be difficult to learn

D. Event loops can cause different parts of your program to corrupt program variables

Done!

Contributing to Open Source Projects

Github Example

- **Fork the project**
  Create a clone of the project in git

- **Create a branch**
  Isolate your changes from irrelevant changes

- **Fix the problem**
  Make the changes to the project locally

- **Suggest the code**
  Ask the main project with a "pull request"

https://github.com/snyderp/petes-problem-repo

Using Event Loops

Event Loop Review

- Single Threaded
- Handle different events on waiting events
- Inversion of control style

Comparison

Threads

- Few changes to existing code
- Possible performance increase (vs event loop)
- Applicable to most… applications

Event Loops

- Avoids concurrency related correctness issues
- Avoids deadlocks
- Efficient memory use
- Not always applicable

Conceptual Event Loop

- Based on `select`, `kqueue`, `epoll`
- Applications use these to register for call backs
- Single thread, single call stack

event-loop/{EventLoop, ReadAsync}.java –>

Event Loop Warnings

- Turns must be fast
- Turns can never block
- Long tasks will need
  - Threads
  - Subprocesses
  - Network / micro-services

If you block…

- Your function call will take a long time
- Event loop will freeze for you to finish
- Your whole application will freeze

event-loop/{AsyncServer}.java –>

Conclusion

- Event loops are great
  - Performance benefits of threads w/o the danger
    - Even better if program is IO bound
- Event loops are fragile
  - Any blocking call back can bring the system down
- Event loops are everywhere