The Async Revolution

ASYNCHRONOUS PROGRAMMING, SWOOLE, AND PHP 8

By Andrew Caya, ZCE, ZCA

Who am I?

- I am Andrew Caya
- Started out with GW-BASIC and QBASIC in 1991
- C, C++ (Qt), Perl
- Linux System Administration
- PHP developer since 2009
  - Zend Certified Engineer since 2015
  - Zend Certified Architect since 2016
- « Mercenary » developer since 2010 (thanks to Tim Lytle for the term)
- CEO and CTO – Foreach Code Factory/Caya Technologies
- Instructor at Concordia University
- Author and technical reviewer for Packt Publishing since 2016
- Lead developer and maintainer of the Linux for PHP project
- Lead developer of the LightMVC Framework
- Upcoming projects:
  - PHP Continuous Learning (new online professional development platform)
  - Book on Atlas ORM (Paul M. Jones as technical reviewer)

What is Asynchronous Programming?

What is Async Programming

- Programming in an “asynchronous” way means to be able to interrupt a program’s normal flow of execution by using structures that allow for iteration, or looping, by temporarily yielding control back to the calling code (ex. generator functions).
  - Requirements:
    - Paused execution (non-blocking code)
      - Most PHP internal structures are blocking in nature
    - Main loop
  - Useful for:
    - Slow I/O calls (especially on the network, but also the hard drive)

A few examples please! ☺

So what about blocking code?

Yep, that's a problem…

What is Async Programming

- Asynchronous programming is not parallel programming in the sense of multithreading
  - Concurrent execution of different parts of the code on different processors for example
  - An example with the pthreads library
- Rather, it is parallel programming only in the sense of getting different parts of your code to run (apparently) simultaneously

What is Async Programming

- New PHP libraries can help:
  - ReactPHP
    - ReactPHP is an event-driven, non-blocking I/O library. This library relies essentially on an event loop that polls file descriptors, uses timers and defers callbacks by registering and executing outstanding ticks on each iteration of the loop.
    - ReactPHP is based on the Reactor pattern which, according to Douglas C. Schmidt, is a "*design pattern that handles service requests that are delivered concurrently to an application by one or more clients. Each service in an application may consist of several methods and is represented by a separate event handler that is responsible for dispatching service-specific requests. Dispatching of event handlers is performed by an initiation dispatcher, which manages the registered event handlers. Demultiplexing of service requests is performed by a synchronous event demultiplexer.*"

      - Schmidt. *Reactor: An Object Behavioral Pattern for Demultiplexing and Dispatching Handles for Synchronous Events*
  - Another async library:
    - amphp

Another example please! ☺

But, wait! There’s more!

PHP 8
and
Advanced Async Capabilities

Promises, Promises…!

☺

PHP 8 and Async

- PHP 8 should introduce "await" capabilities by using "Promises" to map expected return values (see Monads in functional programming)
- PHP 8 should allow for execution of different parts of blocking code through "workers" in order to speed up any code base
- It is actually possible to see some of these new features in action by using the Swoole extension right now!
  - Laravel: https://laravel-news.com/laravel-swoole
    - composer require swooletw/laravel-swoole
    - provider: SwooleTW\Http\LaravelServiceProvider::class,
  - LightMVC Framework https://lightmvcframework.net
  - Blink

Oh! Please tell me there are more examples… ☺

Thank you!

Andrew Caya, ZCE, ZCA

@AndrewSCaya

https://ca.linkedin.com/in/andrewscaya

https://andrewscaya.net

https://etista.com

https://linuxforphp.net

https://lightmvcframework.net