

Flash

When picking a platform to work with, it is important to analyze the benefits and drawbacks of each architecture. The Flash journal paper introduced four different methods of implementing a web server, each with their own benefits and drawbacks. Out of the four, Multi-Process (MP) and Asynchronous Multi-Process Event Driven (AMPED) display the strongest contrast.

In terms of bottlenecks, an MP architecture will fork off a process per user initiated connection resulting in a one-to-one mapping of connections and processes. This allows multiple connections to be sustained and serviced even when other connections are blocking (on disk or network operations). However, this approach needs inter-process communication to make efficient data accounting possible; this can be slow. Also the sheer number of connections (and thus processes) will force an OS to do many costly context switches. In terms of programming ease, each additional connection is handled by a simple fork; however, communication and resource sharing between the processes can be programming intensive.

The AMPED architecture adds an event loop on top of a multi process scheme. This allows the AMPED architecture to out perform the MP scheme by initiating connections in a main process and calling helper functions only on blocking operations; therefore, context switches occur only when disk operations are needed. Also, since a connection is served by the main event loop, data accounting and algorithm optimization is much faster - no inter-process communication needed. Programming an AMPED architecture is harder than a simple MP architecture (no inter-process communication) because the event-loop needs to take care of multiple clients and multiple helper processes communicating job completions. If, however, an MP architecture includes inter-process communication, then this data management can result in the AMPED scheme to be the less complicated.

When choosing between speed or ease of use, the decision depends on many factors; consequently, I cannot place one factor higher relative to the other. Speed is important because it can save money in the real world by allowing companies to purchase less hardware to support software that utilizes the hardware better. Ease of programming on the other hand allows software to be more flexible. An easy programming scheme allows programmers to fix bugs faster and make software revisions easier. The overall ease of implementation allows software developers to hire cheaper workers and to reduce software turn-around time. The real decision depends on specific project needs.