Zhenyu Yan

Why Events Are A Bad Idea (for high-concurrency servers)

From

Rob von Behren

In this paper, rob believe event drivern system is a bad idea for high concurrency servers. He believes that threads can achieve all the advantages of events. In addition, he discussed threading allowing for a simpler and more natural programming style. He examine the claimed strengths of events over thread. He also refined the dual arguments of Raul and Joseph Needham. Finally, he discussed compiler support for threaded systems as a productive area for future research and discussed several enhancements implemented with relatively simple compiler changes.

First rob talks about debate between event and threads. He compares events with threads and rebuts the common arguments against threads. Threads can execute at least high concurrent events and events without substantial qualitative advantages. The lack of scalable user-level threads has provided the biggest impetus for event style, but he has shown that this flaw is an artifact of the available implementation, not the basic property of thread abstraction. Secondly, Rob explains why threads are particularly natural for writing high-concurrency servers. The reason is Control Flow is more natural. And Thread is better rxception handling and state management. Also, existing systems demonstrate a need for threads. Third, Rob explores the value of compiler support for threads. Then he validate an approach with a simple web server. He found out a well designed thread package can achieve the same scaling and event systems require various forms of run-time dispatch which affects performance by reducing opportunities for compiler optimizations and by increasing CPU pipeline stalls. behavior as a well-designed event system. At the end He covers some relative work to enhance threading.

For result, While event systems have been used to achieve good performance in high-concurrency systems, we have demonstrated that using threads can achieve similar or even higher performance. In addition, the simpler programming model provided by the threading system and the rich compiler analysis make the thread have a significant advantage over the event when writing highly concurrent servers.

I think this article is very useful even it is fifteen years ago. Although right now we still have a lot of event driven system for many different language, I still agree threads is better. I had a virtual reality class with C# coding in Unity3d and mobile application development class with java coding in android studio. When I was working on them, the event ordering is a mess in event driven system. I have a lot of troubles when I deal with the bugs.

The presenter Robert Stone and Daniel were Great. Their slide were detail and have graphs, but it is lack of the part of event driven that Daniel talks about. Also Robert talks about the event driven system in the mobile phone application that I am also taking. I feel the same as him when I was doing my project because it is hard to debug when I am working with fragment and activity together. Overall I learn a lot form them.