Protecting running multithreaded applications from races without a restart

Technology #m11-009

Large software applications such as web and database servers employ multiple threads of execution in order to facilitate development or improve performance. Multithreaded software often contain races, or situations in which the execution time variability of one thread may adversely affect the execution of another thread and thereby lead to problematic application behavior. This technology is a live-update system that enables rapid deployment of temporary workarounds for races in running multithreaded software applications without necessitating a disruptive restart of the software. Using an algorithm, it can prevent an identified race from adversely affecting a running application’s state after the workaround is installed.

**Enabling live updates of running multithreaded applications with execution filters that prevent race effects increases software uptime without requiring immediate conventional patching**

Diagnosing and fixing races without degrading software performance or introducing new bugs can be time consuming and leave applications in an unprotected state until they are patched to prevent occurrence of the races. By enabling immediate deployment of execution filters, the technology enables the effects of races upon a multithreaded application to immediately be prevented without patching and restarting the application. The system has very low overhead and does not impact the performance of the protected application.

The software has been implemented on Linux and successfully tested on a range of server, desktop, and scientific software applications.

**Lead Inventor:**

[Junfeng Yang, Ph.D.]
**Applications:**

- Increase uptime of high-availability/mission critical software installations by reducing need to restart the software to clear or patch bugs.
- Patching of running multithreaded software
- Improve performance of multithreaded applications by reducing the effects of possible race conditions in applications

**Advantages:**

- Low performance overhead
- Scales well with increasing numbers of threads
- Modification of original application source code not required
- Reduces interval between race discovery and deployment of fix
- Automatically assesses execution filters to prevent introduction of new races and bugs
- Easy to implement

**Patent information:**

Patent Pending (US 20120096449)

Tech Ventures Reference: IR M11-009

**Related Publications:**


**Inventors**

Junfeng Yang