Using schedules for improving determinism in multithreaded computation

Technology #m11-010

Consumer and industrial computing needs continue to increase while processing components are approaching their physical limit. Multithreading is a useful technique for increasing computational capability by processing multiple requests in parallel. Problems may arise when the program returns differing results for the same input data, known as nondeterminism. This nondeterminism makes multithreaded programs more difficult to test, debug and maintain than sequential programs. This technology is a method that provides robust thread execution schedules for input data, improving the determinism for multithreaded programs. With this technology more efficient processing can be run among single core and multi core processor systems.

Faster processing by comparing input data to cached schedules

Existing methods for improving determinism in multithreaded programs focus on the way the processor determines the schedules to run but are unable to prevent input nondeterminism. By storing and reusing a cache of past working schedules, this technology causes the program to become less dependent on the input data, resulting in greater consistency as well as increased performance rates. It is readily implementable with existing systems, as the method is adaptable for any type of suitable hardware such as PCs or servers.

This technology has been tested on a wide variety of programs, including Apache and MySQL.

Lead Inventor:

Junfeng Yang, Ph.D.

Applications:

- Development of multithreaded software
- Debugging and improving determinism of existing systems

Advantages:

- Reduces both schedule and input nondeterminism in multithreaded programs
• Implementable with a wide range of applications and hardware

**Patent information:**
Patent Pending (US 20120102470)
Tech Ventures Reference: IR M11-010

**Related Publications:**


- [Cui H, Wu J, Tsai C, Yang J. “Stable Deterministic Multithreading through Schedule Memoization” 9th USENIX Symposium on Operating Systems Design and Implementation, October 4-6, 2010.](#)

**Inventors**

Junfeng Yang