Основанное на анализе кода fuzzyпланирование потоков с применением lincheck для обнаружения ошибок в многопоточном коде

Проблема

Текущие проверки линеаризуемости алгоритмов, например с помощью lincheck, заключаются в случайном многопоточном исполнении кода и сравнеии результатов со всеми возможными последовательными исполнениями тех же операций. В ходе проверки даже самых простых студенческих lock-free алгоритмов выявилась полезность следующих функций:

- 1. Останавливать потоки после / перед CAS для повышения выроятностью некорректных ситуаций
- 2. Долгой остановки потока после / перед CAS с проверкой отработки остальных потоков по своим задачам

Пути решения

Внедриться в JRE или байт-код и на основе понимания использования CAS явно влиять на планирование потоков

Ближайшие задачи

- 1. Разобраться со способами внедрения через обёртку над Atomic*Reference с барьерами и соответствующий ClassLoader
- 2. Разобраться с возможностью влиять на планирование потоков в JRE через outline|приоритеты

Идеи на обсуждение

From:

http://wiki.osll.ru/ - Open Source & Linux Lab

Permanent link:

http://wiki.osll.ru/doku.php/etc:users:kel:jre_fuzzy_threads_planning?rev=1580678049

Last update: 2020/02/03 00:14

