Алгоримты обнаружения взаимныхблокировок в многопотчных приложениях

Мазунин Константин, Доронин Олег, Дергачёв Андрей

2020

Введение

Компания Google имеет ряд иструментов для нахождения ошибок в коде, так как Thread Sanitizer, Address Sanitizer и Memmory Sanitizer, которые позволяют находить ошибки в C/C++ коде, такие как: взаимную блокировку, гонку данных, переполнение данных, утечку данных и т.п.

Взаимоблокировка или Deadlock[1] — это ситуация в многозадачной среде, при которой несколько процессов или потоков находятся в состоянии ожидания ресурсов, занятых друг другом, при этом ни один из них не может продолжать свое выполнение. Данная проблема встречается часто в многопоточных приложения и может не только снижать производительность но и приводить к полному “зависанию” всей системы в целом.

Google Thread Sanitizer - инструмент для нахождения ошибок в многопоточных приложения на C/C++ и Go. Позволяет находить гонки данных и детектировать возможные взаимные блокировки между потоками. К сожалению, GTSAN не может корректно обрабатывать все виды взаимных блокировок и имеет ошибки первого и второго рода.

Актуальность темы исследования. Актуальность темы дипломно работы заключается в том...

Тут можно написать, что крупные компании как гугл и Яндекс используют C++ и что пишут многопточные программы. Разработка инструмента, который позволяет находить и исправлять ошибки в многопоточном коде, позволяет увеличить качество программ.

Языки C/C++(2 и 4 места в рейтинге TIOBE) являются довольно популярными и хорошы(заменить) для написания высокопроизводительных многопоточных приложений.

Deadlock является часто встречаемой ошибкой в многопоточных приложениях, которая приводит к деградации или полному простою всей системы. Существующих инструменты для выявления deadlock не позволяют...

Что думаю написать в актульности: (deadlock приводит к деградации системы) -> (deadlock лего допустить / часто встречается) -> (существующие инструменты не позволяют выявить все виды deadlock или не допускают большое количество ошибок) -> (проблема серьёзна, но инструменты не мог с ней адекватно справиться) -> (ВКР актуален)

Объект исследования - ЭВМ

Предмет исследование - алгоритм детектирования Deadlock в GTSAN

Гипотеза: Предполагается, что в ходе данной работы получиться улучшить один из параметров алгоритма, без ухудшения других: производительность(обратный параметр, процентное соотношение скорости исходной работы и после инструментации), количество ошибок(уменьшение ошибок первого и второго родов) или используемая память(обратный параметр, процентное соотношение используемой памяти исходной работы и после инструментации).

Цель - улучшение алгоритма детектирования Deadlock

В связи с поставленно целью были выявлены следующие **задачи**:

- Изучение алгоритма детектирования Deadlock в GTSAN
- Обзор алгоритмов и подходов к детектированию Deadlock в многопоточных средах
- Разработка алгоритмов для улучшения алгоритма детектирования Deadlock в GTSAN
- Реализация алгоритмов для улучшения алгоритма детектирования Deadlock в GTSAN
- Тестирование производительности разработанных алгоритмов

В процессе работы над дипломом были использованы следующие методы исследования: изучение материалов научных и периодических изданий по проблеме, анализ документации и (написание кода).

Научная новизна состоит в использование ...

1 Глава 1

Алгоритм GTSan базируется на построение графа ресурсов.

Для детектирования Deadlock в C/C++ существуют решения:

- Google Thread Sanitizer
- Valgrind
- GDB с плагином