Верификация потокобезопасности структур данных с помощью управляемых точек синхронизации

Автор: Александр Сергеевич Озерцов, 444 группа
Научный руководитель: д. ф.-м. н., профессор А.Н. Терехов
Научный консультант: руководитель направления dxLab Д.И. Цителов
Рецензент: асистент СПбГПУ М.А. Беляев

Санкт-Петербургский государственный университет
Кафедра системного программирования

Lin-Check

- Генерирует многопоточный алгоритм
- Запускает алгоритм в однопоточном режиме возможными способами
- Запускает алгоритм в многопоточном режиме и сверяет результат

Постановка задачи

Целью работы является разработка и внедрение системы управления порядком исполнения многопоточных алгоритмов в проекте lin-check
Задачи:

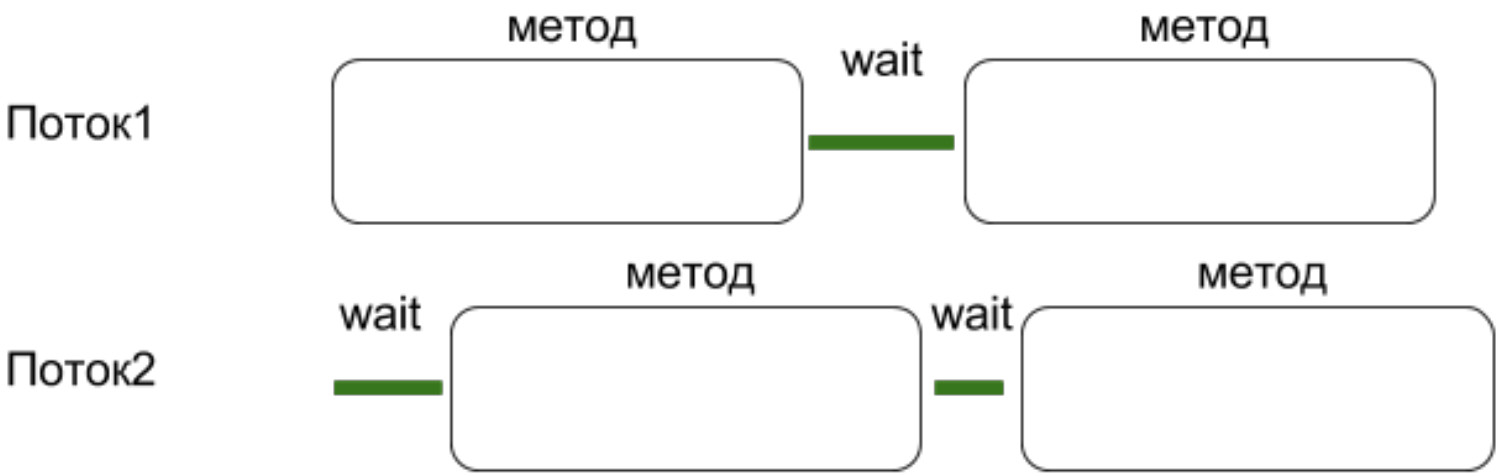
- Выполнить обзор методологий тестирования многопоточных программ
- Формализовать операции JVM, влияющие на поведение алгоритма
- Разработать архитектуру для создания стратегий управления ходом исполнения программ
- Доработать существующий алгоритм с учетом архитектуры
- Разработать алгоритм управления порядком исполнения многопоточных программ
- Сравнить полученные алгоритмы с нынешней реализацией

Существующие инструменты

- Не дающие никаких гарантий покрытия
  - jcstress
  - Penelope
- Предоставляющие гарантии покрытия пространства чередований
  - Chess
- Предоставляющие гарантии покрытия пространства чередований и входных данных
  - STORM

Предыдущая реализация

- Модификация стресс тестирования
- Не предоставляет гарантий нахождения ошибки

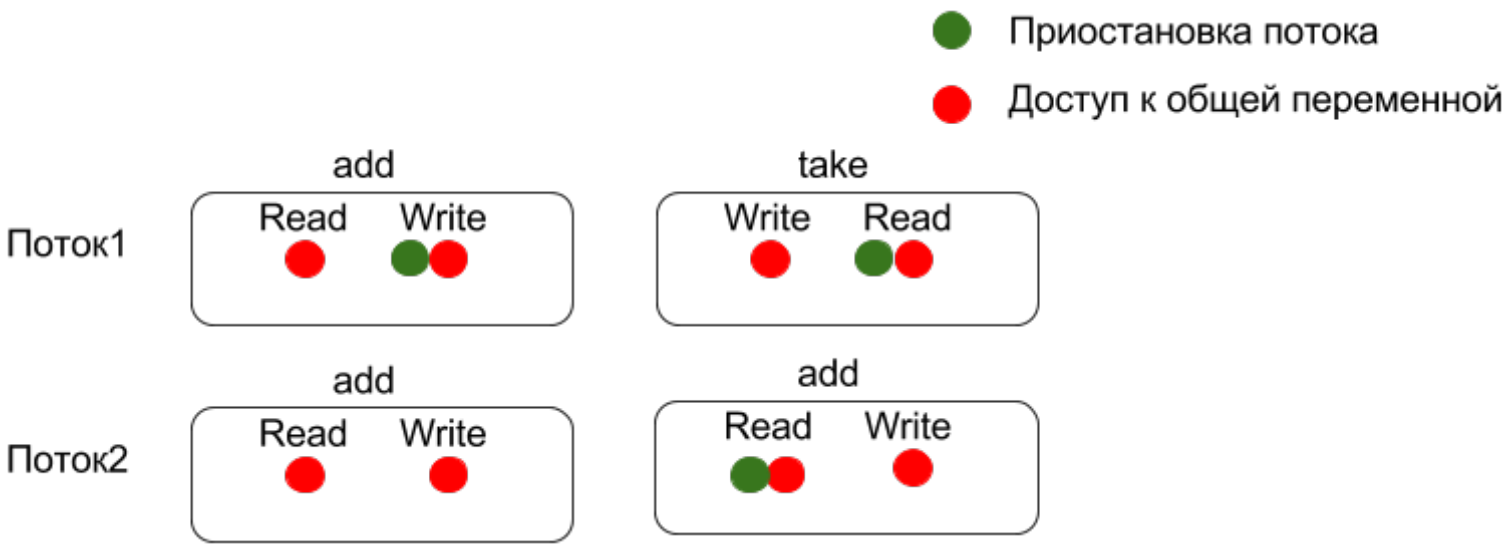


Архитектура системы

- Паттерн стратегия
- Управление ходом исполнения
  - оповещение о запуске/окончании потока
  - оповещение о доступе к общей переменной
- Вспомогательные методы
  - оповещение о создании тестового алгоритма
  - оповещение об окончании запуска тестового алгоритма
  - прекращение запусков тестового алгоритма

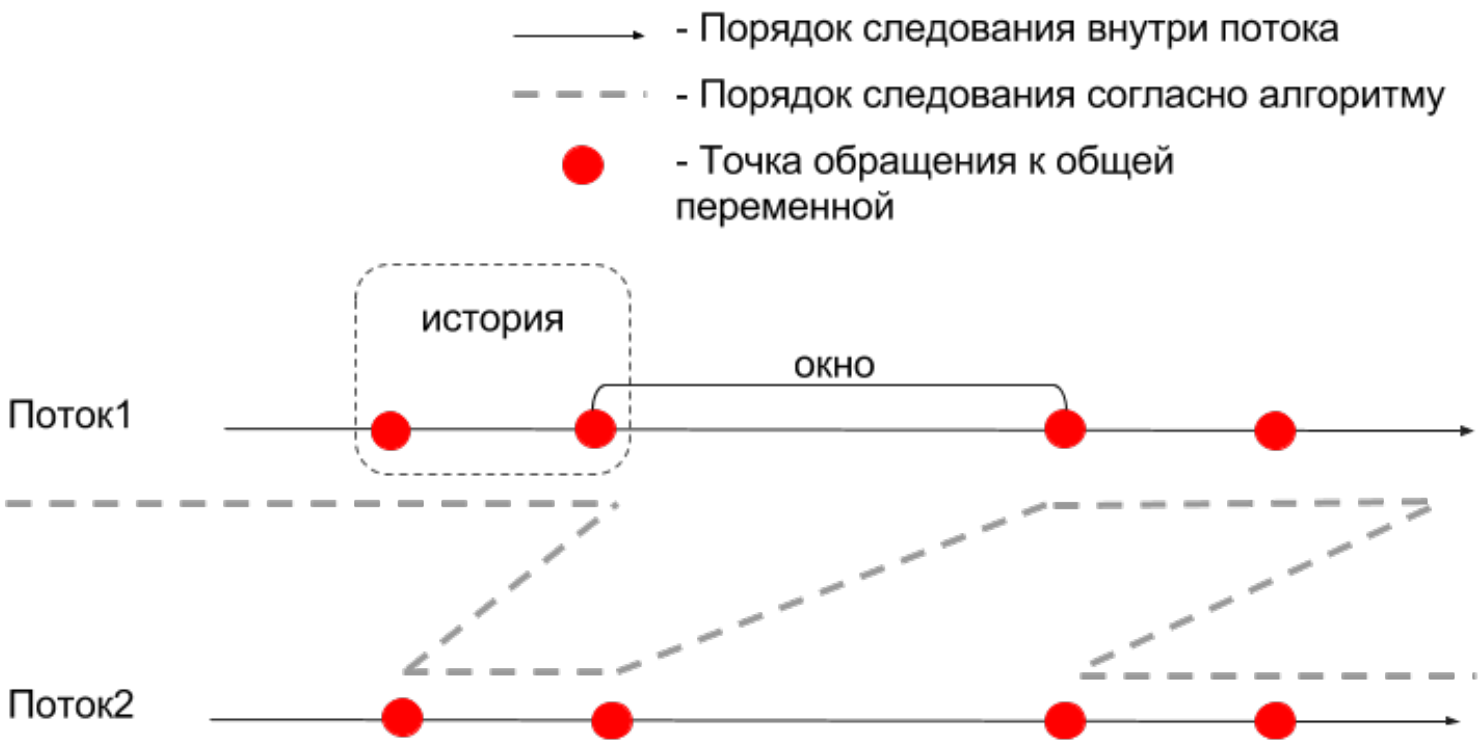
Доработка алгоритма

- Переключение в точках синхронизации
- Нет гарантий нахождения ошибки



Алгоритм

- Планировщик типа Round-robin[Delay-bounded scheduling, 2011]
- Проверяет устойчивость каждого окна
- Покрывает пространство переключений



Алгоритм. Расписание

Определение (Очередь исполнения)

Перестановка чисел от 0 до k, определяющая в каком порядке будут выполняться потоки.

Определение (Прерываемые потоки)

Пара чисел, обозначающая какие потоки будут прерываться между собой.

Определение (Глубина устойчивости окна)

Количество точек, которое может выдержать окно без нарушения линеаризуемости.

Алгоритм. Паттерны SMT[1]

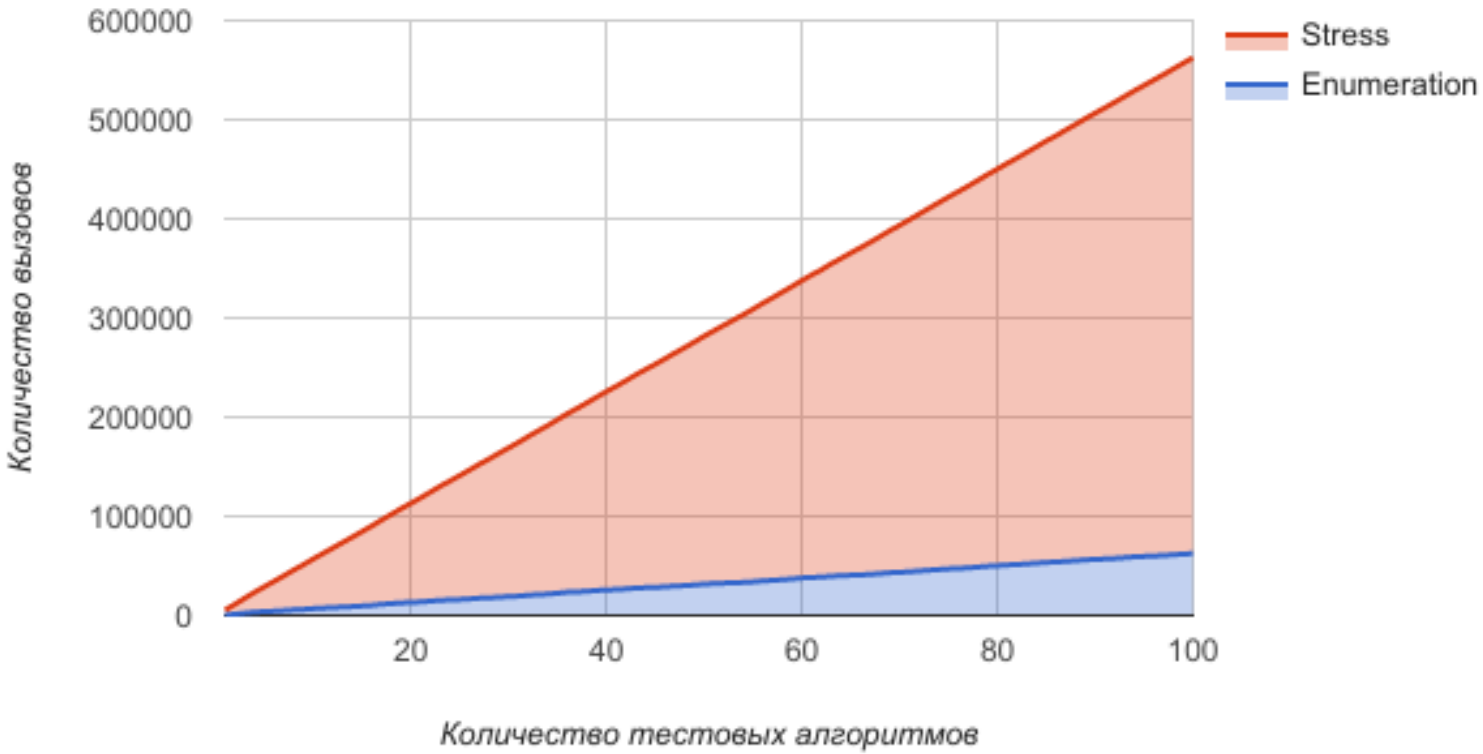
Определение (Паттерн SMT)

Трехэлементная последовательность событий многопоточной программы, влияющих на потокобезопасность. 1 и 3 события принадлежат одному потоку, а 2 — другому.

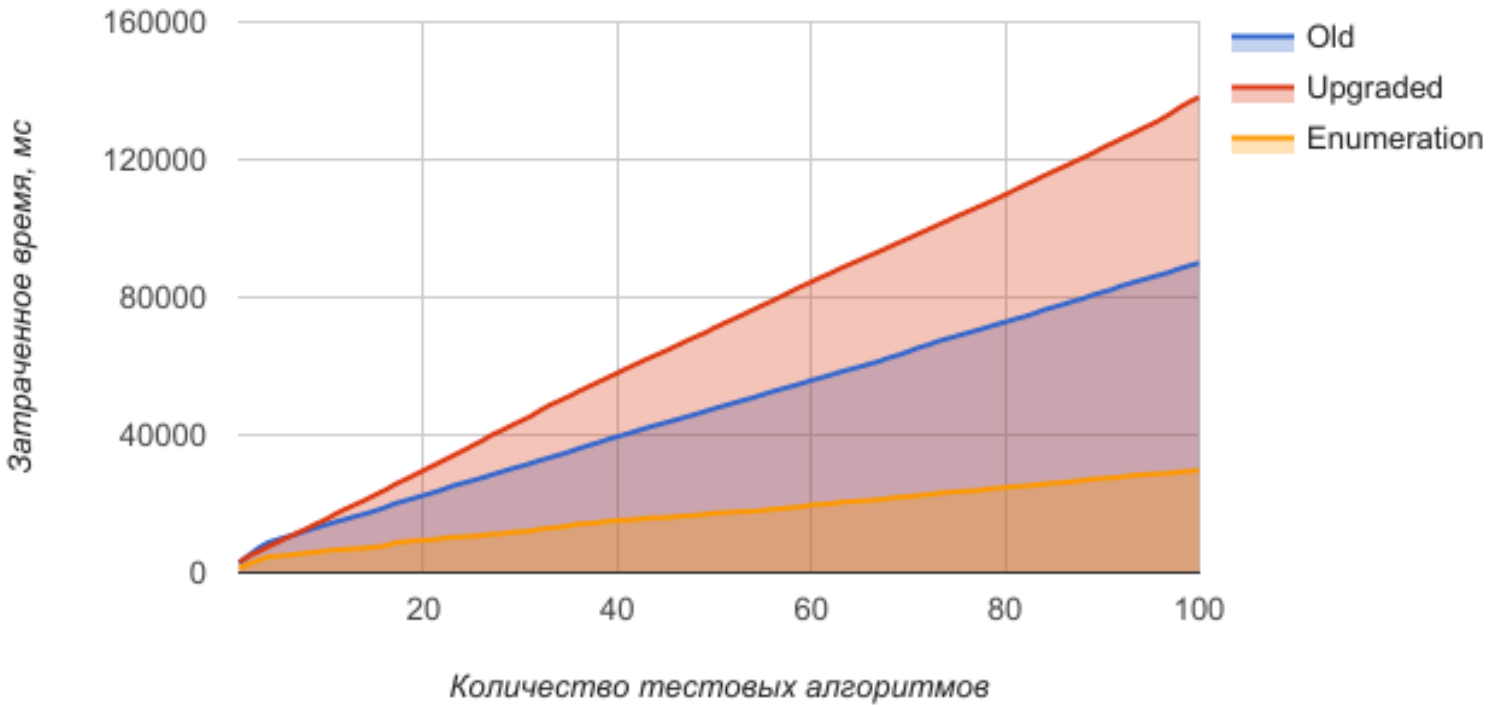
- Read-Write-Write
- Read-Write-Read
- Write-Read-Write
- Write-Write-Read
- Write-Write-Write

[1] Задача выполнимости формул в теориях (англ. satisfiability modulo theories, SMT)

Сравнение производительности



Сравнение производительности



Сравнение производительности

- Выявлены ошибки в существующих библиотеках

Структура данных	Метод тестирования	Количество тестовых алгоритмов	Количество вызовов
List	Old Upgraded Enumeration	260 260 260	1 295 243 1 295 017 118 852
Queue	Old Upgraded Enumeration	356 356 217	1 777 046 1 775 588 116 027
Deque	Old Upgraded Enumeration	3085 1322 1322	15 421 064 6 610 320 853 456

Результаты

- Выполнен обзор методологий тестирования многопоточных программ
- Формализованы операции JVM, влияющие на поведение алгоритма
- Разработана архитектура для создания стратегий управления ходом исполнения программы
- Доработанный алгоритм учитывает новую архитектуру
- Разработан алгоритм управления порядком исполнения многопоточных программ
- Сравнение подтвердило прирост производительности