

Kotlin STM

Автор: Доронин О.В.

Руководитель: Иготти Н.Н.

Цель

- Создание реализации библиотеки для Software Transactional Memory на языке Kotlin для упрощения разработки многопоточных приложений

Задачи

1. Изучение существующей реализации STM в Scala
2. Изучение различных статей в рассматриваемой предметной области
3. Разработка STM на языке Kotlin
4. Создание тестов
5. Анализ производительности

Что такое STM?

- Технология синхронизации потоков. Она упрощает параллельное программирование, выделяя группы инструкций в логически атомарные транзакции
- Для своей реализации не требует аппаратной поддержки со стороны процессора

Преимущества

- Относительная простота использования (заключение целых методов в блок транзакции)
- Полное отсутствие блокировок и возможности взаимных блокировок (deadlock, livelock)
- Повышение уровня параллелизма, а следовательно и производительности

Недостатки

- При неправильном использовании возможно падение производительности и некорректная работа
- Ограниченность применения – в транзакции нельзя выполнять операции, действия от которых невозможно откатить (ввод/вывод)
- Сложность отладки

Результаты

Тестовый стенд

- **Brand:** Intel Core i7 3610QM

- **CPU:**

Byte Order: Little Endian

Model Name: Intel(R) Core(TM) i7-3610QM CPU @ 2.30GHz

CPU(s): 4

L1d cache: 32 KBytes

L1i cache: 32 Kbytes

L2 cache: 256 Kbytes

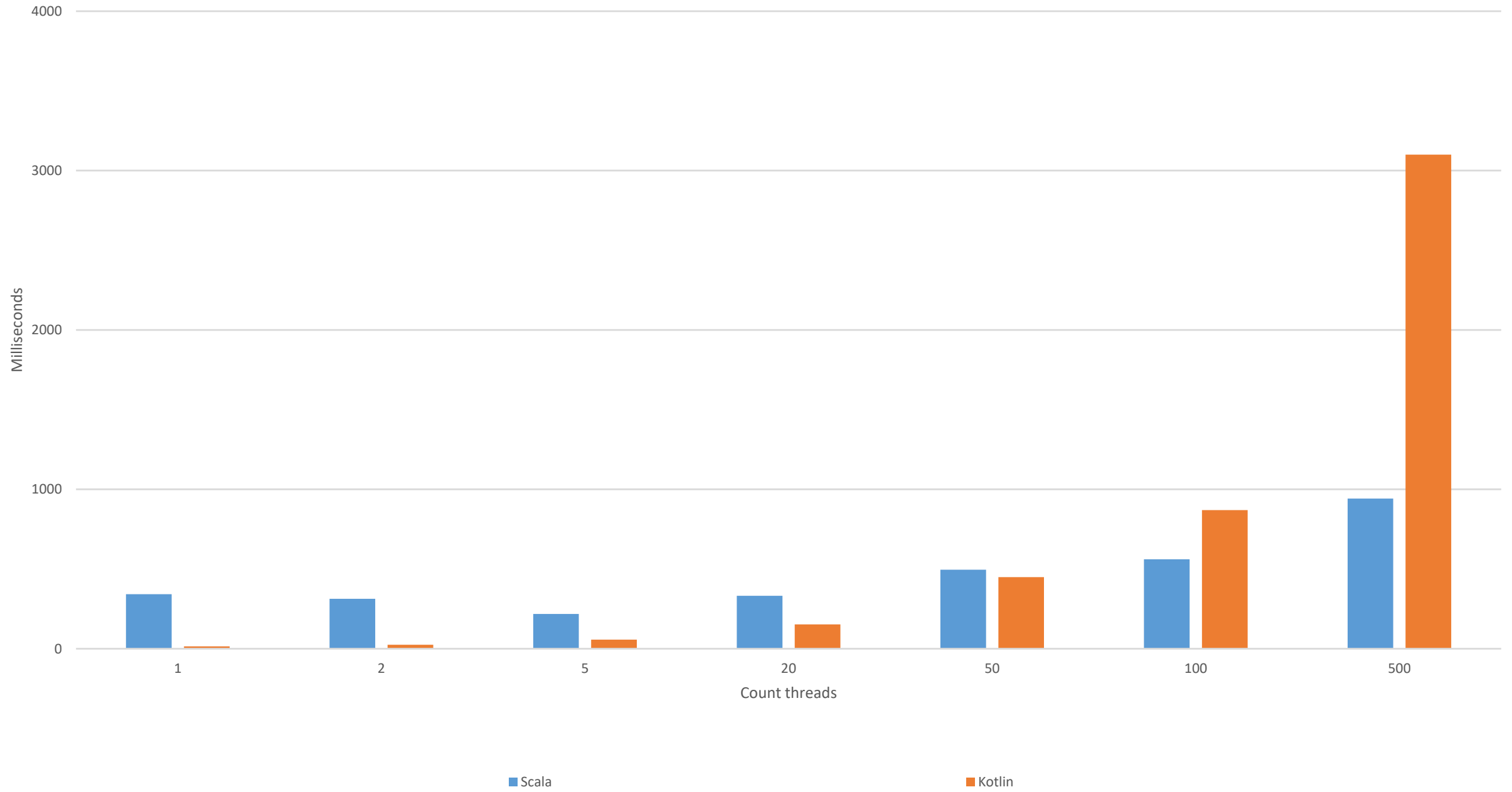
L3 cache: 6 MBytes

- **RAM:** 8 GBytes

Тест 1

- Односвязный список
- Поток удаляет и добавляет в начало списка.
50 % 50
- 1000 операций вставка/удаление на поток
- Много столкновений потоков

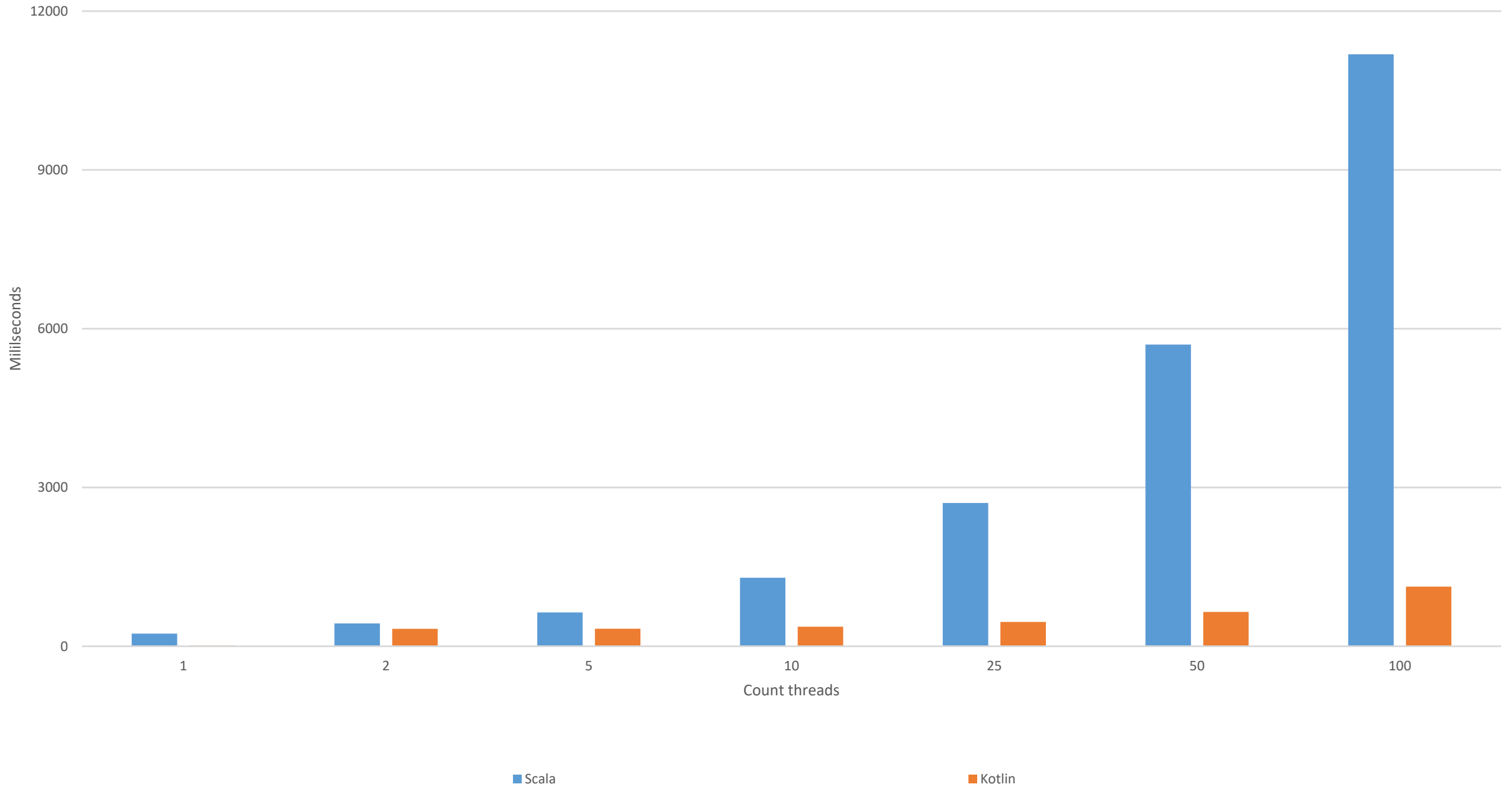
List measurements



Тест 2

- Два банковских счета
- Половина потоков переводит деньги со счета 1 на счет 2 , а вторая половина со счета 2 на счет 1
- Много столкновений потоков

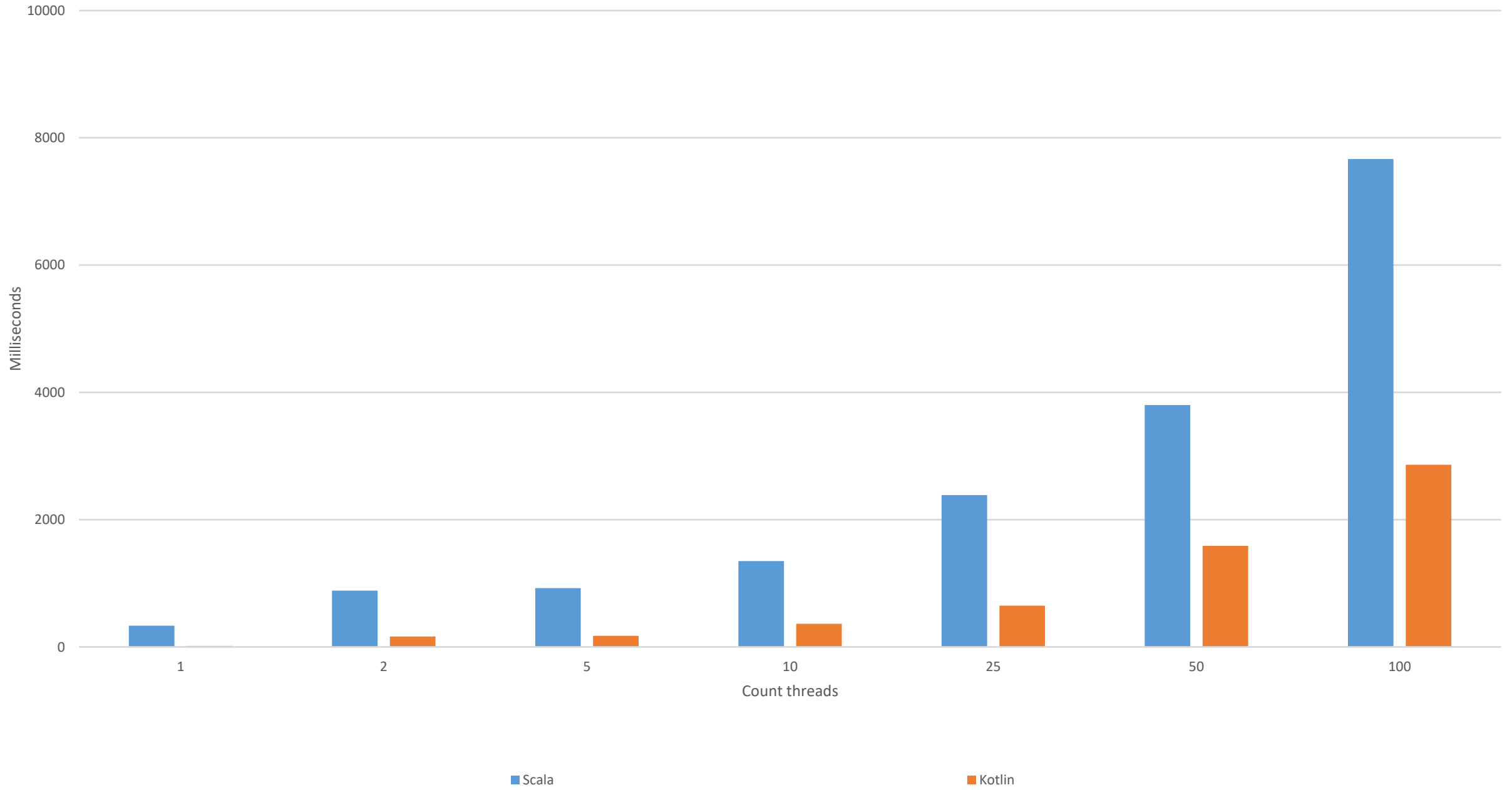
Account measutements



Тест 3

- Массив банковских счетов
- Потоки переводят деньги с одного счета на другой
- Мало столкновений потоков

Array measurements



Заключение

- Получены базовые знания о возможностях языков Kotlin и Scala
- Разработана STM на языке Kotlin
- Проведен сравнительный анализ реализаций STM в Kotlin и Scala
- Разработаны тесты
- Измерена производительность и масштабируемость реализации

Спасибо за внимание!