слайд 1.

В современном мире существует множество задач, в которых требуется параллельная обработка данных. Ярким примером могут быть вычислительные задачи, вроде математических операций над матрицами, поскольку в них требуется получать результат как можно скорее. Другой пример это системы сбора и обработки данных, в которых требуется повышать пропускную способность системы.

слайд 2

Традиционно, параллелизм реализуется внутри операционной системы с помощью механизма потоков. Однако, у потоков есть минусы.Они предлагают достаточно "тяжеловесный" механизм: их создание и переключение несет в себе крупные накладные расходы. Избежать накладных расходов на использование потоков можно, применяя вместо них сопрограммы.

слайд 3

<пересказ>

слайд 4

Сопрограммы уже реализованы в языках программирования С++ стандарта 20, С# и в относительно молодом языке Go. К сожалению, на сегодняшний день сопрограммы не реализованы в языке Java, но работы в этом направлении ведутся проектом "Loom".

слайд 5

<пересказ>

слайд 6

Целью же моей работы является изучение применимости сопрограмм как альтернативу потокам в программах Java. Для этого были поставлены следующие задачи: <слайд>

слайд 7

Первым шагом работы стал анализ существующих реализаций. Для этого был создан набор тестов производительности различных реализаций сопрограмм и потоков. Анализ предметной области показал, что не существует подходящего набора тестов, который бы измерял 3 интересующих параметра:

1) Скорость переключения

2) потребление физической памяти.

Тесты доступны по ссылке на слайде.

слайд 8

Следующим шагом работы стала реализация переключения контекста сопрограмм. Переключение - это процесс передачи управления от одной сопрограммы другой в точках явного планирования, помеченных на слайде функцией yield(). Так же переключение происходит при завершении исполнения сопрограммы.

слайд 9

к переключению контекста есть несколько подходов. Для реализации в HuaweiJDK использовался механизм как в языке Go, поскольку он показался более эффективным.

слайд 10

Наконец были измерены скорости переключения сопрограмм в управляемых средах HuaweiJDK, OpenJDK и языка Go. Как видно на слайде, HuaweiJDK выигрывает в скорости переключения у OpenJDK на всех измерениях, но проигрывает языку Go.

слайд 11

Так же была измерена скорость переключения потоков. Как видно из слайда, сопрограммы имеют лучшую скорость переключения, чем потоки на всем диапазоне измерений.

слайд 12

По сравнению с другими реализациями, сопрограммы в HuaweiJDK имеют неплохой результат по потреблению памяти: они обходят OpenJDK, но проигрывают сопрограммам в Go.

cлайд 13

Измерение потребления физической памяти потоков показало, что сопрограммы имеют меньшее потребление памяти, чем потоки при любом их числе.

слайд 14

Так же были проведены измерения производительности сопрограмм и потоков. Для этих целей была модифицирована библиотека Colt Parallel. Эта библиотека используется для многопоточной обработки данных. В ней потоки были подменены практически один к одному. Результаты измерений представлены на слайде <пересказ>

слайд 15

После проведения замеров можно сформулировать ключевые отличия потоков от сопрограмм. <пересказ>

<остальные слайды пересказываются>