

Quasar, Kilim отчёт

Данный отчёт является скорее мнением т.к. полностью понять преимущества quasar ввиду отсутствия статей по нему и какой-никакой документации в коде (касательно части quasar-core, которая отвечает за weave процесс). Также в quasar обнаружилась aot компиляция, что сильно облегчило процесс понимания происходящего.

Принцип работы легковесных потоков Quasar такой же как у библиотеки kilim. Разница в том, что kilim поддерживает только концепцию message blocking, а quasar предоставляет работу с fiber как с обычными потоками имплементируя какую-то из JavaAPI. Принцип работы прост и одинаков и там и там (сложность в реализации). Все suspendable/pausable методы оборачиваются в так называемые “сэндвич” из сохранения состояния, до вызова метода (down), и возврата состояния (раскрутки стэка up) после. Подробнее(хотя и не достаточно, по моему, подробно) описывалось в статье kilim и часть в его коде. Kilim для сохранения состояния создаёт новые классы, в которых всё и хранит. Quasar, судя по всему, использует для этих целей массивы объектов. Kilim приводит разумность их подхода в статье. Кода quasar генерирует порядочно. Для примера я сделал просто два fiber и в одном из них вызвал park. И код который раньше занимал 10 байтковых строк стал занимать 50. Скорей всего всё это нужно, т.к. судя по коду, quasar очень эффективно работает со стекком состояний. Я даже сгенерировал граф управления, чтобы понимать происходящее, но он не сильно облегчает работу.

Тем не менее обе либы проводят оптимизации (kilim например понимает константы и не сохраняет их в состоянии) Изобретать свой потоковый легковесный велосипед смысла на начальных этапах не имеет. Т.к. всё-что будет написанное в лучшем случае будет сведено к пародии на kilim. А для того чтобы работать с потоками быстрее и всячески оптимизировать их работу для данной задачи, надо для начала понять общие ограничения на структуры обрабатываемых классов (возможно удастся понять общие шаблоны в них и тогда более эффективно с ними работать) и барьерное общение. Если производительность увеличится в разы, то quasar и kilim это покажут, всё остальное дополнительные микрооптимизации.

Park и unpark в квазаре работают так как нами и надо. Позволяют гибко блокировать текущий поток и разблокировать целевой таким образом создавая свой граф управления потоками. Да и вообще api квазара очень удобно и понятно сделано.

Ссылки:

Статья по kilim

<http://www2.seas.gwu.edu/~simha/NHCworkshop/2006/final/NHC06-Srinivasan.pdf>

Reddit где ведущий разработчик quasar-core отвечает на вопросы и описывает разницу с kilim.

https://www.reddit.com/r/programming/comments/1x0tfu/quasar_rewrites_java_bytecode_to_implement/

https://www.reddit.com/r/java/comments/3b9cvq/quasar_lightweight_threads_and_actors_for_the_jvm/

Граф управления, генерируемый квазаром, покажу при встрече.