Эссе 2

Автор статьи Эдвард Ли весьма детально описывает принцип работы потоков и главную проблему, связанную с ними. Поначалу было сложно читать длинные термины, а также математические доказательства эквивалентности программ. Однако, немного подумав над темой, можно уловить главный посыл: потоковая модель вычислений – не самая лучшая концепция с точки зрения безопасности и детерминированности. Автор подчёркивает проблему непредсказуемости в работе потоков, из-за чего программисты могут получать серьёзные ошибки. Это действительно так, но мне кажется, что главный недостаток потоков вылезает как раз из их главного достоинства – возможности параллельной работы с общим адресным пространством. Безусловно потоки необходимо контролировать, иначе при доступе к общей памяти они могут создать множество конфликтов. Для этого программисты придумали инструменты синхронного доступа к памяти и блокировок. Однако, по заверению Эдварда Ли, таким способом люди смогли лишь сократить количество недетерминированности, но не избавиться от неё полностью. Как пример: при синхронном доступе в память появляется другая проблема - deadlock`и. Автор показал какие конфликты могут возникнуть с шаблоном проектирования «Наблюдатель» и рассказал о том как однажды он и его команда разработчиков сами столкнулись с непредвиденной «мёртвой блокировкой», хоть они и рассчитывали всевозможные риски. Эдвард Ли смог привести в качестве альтернативы различные способы уменьшения недетерминизма: концепция рандеву, PN director, SR director, PR director. Все они являются частью координационных языков программирования, которые как раз созданы для работы с распределёнными системами. По сути, это подобие высокоуровневых шаблонов (MapReduce) для уже существующих языков программирования (Java, C++ и тд). Являются ли они панацеей от недетерминированности работы потоков? Нет, но они используют немного другой подход к параллельному выполнению задач – программа начинается с полного детерминизма, недетерминизм вводится только там, где это необходимо. То есть, происходит сокращение неопределённостей в программе. Мне понравилось, что Эдвард Ли приводит сильные аргументы в пользу того, почему потоки проигрывают другим концепциям, но по моему мнению, у каждой идеи есть свои области применения, преимущества и недостатки одного над другим. Я считаю, что программирование, как область, движется в правильном направлении и уже существующие языки модифицируются для работы с потоками и уменьшения недетерминизма. Очевидно, что от костылей или появления новых ошибок никуда не денешься, но совершенствовать нынешние инструменты взаимодействия с потоками определённо стоит