

1 First Section

(Конспект книги параллельный и конкурентный хаскелл товарища Симона Марлоу)

Во многих случаях не делают различия между параллелизмом и многопоточностью.

Parallelism - разделяем работу между несколькими отдельными вычислительными единицами (процессоры, ядра, ГПУ) в надежде что расходы на поддержку параллелизма будут ниже профитов от него

Concurrency - она же многопоточность - исполнение в одной программе нескольких потоков управления. Потоки выполняются одновременно в том смысле что побочные эффекты потоков перемежаются между собой. Количество вычислительных ядер не имеет значения. Используется в задачах ИО (считывание - запись БД и пр). Происходит ли выполнение действительно одновременно - остается вопросом реализации.

В Хаскеле нет и не может быть никакого "Потока управления" (нет побочных эффектов, порядок выполнения не имеет никакого значения)

Единственный профит от Параллелизма - скорость. Многопоточность же позволяет реализовать например асинхронный ввод-вывод.

Детерминистская модель программирования - при каждом запуске получаем одинаковый результат. Недетерминистская модель программирования - при каждом запуске разный результат. Многопоточные программы недетерминированные тк взаимодействуют с внешними агентами. Недетерминированные программы намного сложнее тестировать.

Лучше всего писать детерминированные параллельные программы. Большинство процессоров предлагают детерминированный параллелизм в виде конвейерного исполнения (pipelining) и множественных единиц исполнения.

Старая нерешаемая проблема автоматического параллелизма (см также закон Амдала) заключается в том, что даже в функциональных ЯП компилятор не знает как "разрезать" программу так чтобы затраты на распаралеливание не сводили на нет профит от параллельных вычисле-

ний.