Билет 40. Методы эффективной организации параллельных вычислений на графических процессорах

Коротко:

1. Утилизация латентности памяти
   Цель: эффективно загружать ядра
   Решение CPU: сложная иерархия кэшей
   GPU: За счет наличия сотен ядер и миллионов нитей на GPU легче утилизировать всю полосу пропускания. Быстрое переключения контекста – эффективное перераспределение вычислений
2. SIMT(Single instruction, multiple thread) и масштабирование
   Виртуальное: Поддержка миллионов нитей в GPU; независимость виртуальных блоков
   Аппаратное: Мультипроцессоры независимы -> Можно “нарезать” GPU с разным кол-вом SM (streaming multiprocessor)
3. Асинхронность в CUDA
   Чтобы GPU больше времени работало в фоновом режиме, параллельно с CPU, некоторые вызовы являются асинхронными. Отправляют команду на устройство и сразу возвращают управление хосту
4. Работа с глобальной памятью
   a. Расположена в DRAM GPI
   b. Объем до 4 Гб
   c. Использование кэшей L1, L2

Общие принципы эффективной работы:

- Обращения нитей варпа* в память должны быть пространственно-локальными
- Начала строк матрицы должны быть выровнены
- Использование массивов структур
- Выбор режима кэша L1
- Избегать косвенной адресации
- Избегать обращений нитей варпа* к столбцу матрицы

*Варп – группа из 32 потоков и является минимальным объемом данных, обрабатываемых SIMD-способом в мультипроцессорах CUDA.

Полную версию билета см в gosy.pdf, all.pdf (билет 40) или билеты_скиподы.pdf (билет 16)