

## **Доклад**

### **Слайд 1.**

Здравствуйте, уважаемая комиссия. Представляю вашему вниманию научно-исследовательскую работу по теме «Параллелизация логического вывода».

### **Слайд 2.**

Математическая логика представляет собой формализацию человеческого мышления и представления его знаний. В связи с этим появилась необходимость автоматизации процесса логического вывода, и была придумана новая парадигма программирования – логическое.

В широком смысле он понимается как семейство таких методов и решений проблем, в которых используются приемы логического вывода для манипулирования знаниями.

### **Слайд 3.**

Основной успешной реализацией данной области является Prolog. Его характерной особенностью является то, что используется метод «полного перебора» для вывода решений. В основе этой идеи была развита Aurora. К современным реализациям относятся Datalog, Yedalog (от корпорации Google) и др.

### **Слайд 4.**

Метод «полного перебора» характеризуется ожиданием решения, которое может выполняться на примере NP-трудной задачи «комивояжера» днями или годами. Поэтому для увеличения скорости подбора ключа применяют распараллеливание вычислений. К основным способам ускорения относят AND- и OR-параллелизм, которые можно представить в виде AND- и OR-деревьев.

### **Слайд 5.**

В последние два десятилетия были проведены новые исследования, в которых изучалась роль новых параллельных архитектур в ускорении и расширении масштабов логического программирования. С развитием технологий всё чаще стали прибегать к использованию GPU ввиду того, что графические процессоры имеют большее число ядер по сравнению с CPU. Так, у NVIDIA TITAN X (PASCAL) их число составляет 3584 штук.

### **Слайд 6.**

Для работы с GPU выделяют два основных ПО: CUDA и OpenCL.

В CUDA можно выделить такие особенности, что основой послужила SIMD-архитектура с некоторым отличием: только потоки в рамках одной группы могут выполняться одновременно. Потоки разных блоков могут находиться на разных стадиях выполнения программы. Но такое ПО применимо исключительно для видеокарт NVIDIA, и требуется специальный компилятор для host кода.

OpenCL применяется для различных видеокарт, исходный код преобразуется сначала в OpenCL программу, затем она компилируется для нужного устройства. По результатам некоторых исследований скорость вычислений проигрывает CUDA в полтора раза.

### **Слайд 7.**

Была предложена реализация Datalog на GPU. В основе этой идеи лежит использование операторов реляционной алгебры (select, join, projection), каждый из которых выполняется на своих ядрах.

Для select используются ядра для пометки всех удовлетворяющих выборке строк, суммирование префиксов для определения размеров буфера результата и записи этого результата.

Для join используют структуру дерева поиска с учетом кэша, что позволит использовать на графических процессорах. Выполнение проекции включает в себя взятие всех элементов каждого требуемого столбца и сохранение их в новой ячейке памяти. Использование высокой пропускной способности GPU памяти по сравнению с CPU позволит выполнять на нем.

### **Слайд 8.**

Таким образом, исследования по идеям параллелизма на GPU продолжаются и в настоящий момент. В основе распараллеливания лежит OR-метод, который позволяет с использованием графического процессора сократить время полного перебора путем установки для каждого ядра новую точку старта. Такой метод решения можно использовать на графическом процессоре.