

ПРИМЕНЕНИЕ ГРАФИЧЕСКИХ УСКОРИТЕЛЕЙ ДЛЯ ОБРАБОТКИ ЗАПРОСОВ НАД СЖАТЫМИ ДАННЫМИ В ПАРАЛЛЕЛЬНЫХ СИСТЕМАХ БАЗ ДАННЫХ

С.О. Приказчиков, П.С. Костенецкий

Южно-Уральский государственный университет

Введение

В настоящее время известно большое количество исследований обработки запросов к базам данных в оперативной памяти [3], но использование графических ускорителей (ГПУ) открывает новые перспективы исследований в этой области [1]. Целью данной работы является оценка эффективности выполнения запросов к сжатой базе данных без предварительной распаковки с использованием графических ускорителей, поддерживающих технологию CUDA. Объем внутренней памяти ГПУ на порядки меньше, чем объем оперативной памяти современных вычислительных систем. Это ограничивает размер базы данных, которую можно загрузить в память ГПУ и как следствие не позволяет раскрыть весь вычислительный потенциал ГПУ. Для снижения влияния объема видеопамати на возможность хранения в ней больших баз данных применяется сжатие. Предполагается, что сжатие позволит не только увеличить объем хранимой базы данных, но и повысит производительность по сравнению с обработкой несжатых данных.

Реализация

На текущий момент разработан эмулятор СУБД, позволяющий выполнять запрос SELECT над сжатыми данными, непосредственно на ГПУ с использованием технологии CUDA и, для сравнения, эмулятор для ЦПУ, работающий по технологии OpenMP.

Для сжатия был выбран и реализован алгоритм RLE [4]. Основным преимуществом использования данного алгоритма в СУБД является возможность выполнения многих реляционных запросов без предварительной распаковки данных.

Реляционное отношение базы данных хранится в сжатом виде в памяти ГПУ. Кортежи отношения имеют по 2 атрибута, первый атрибут – номер кортежа (идентификатор), второй атрибут – значение (строка символов). Первым этапом алгоритм производит поиск строк, длина которых совпадает с размером искомой строки. Так, каждый блок производит выборку из нескольких подряд идущих кортежей. Каждый поток блока выполняет сравнение одного символа искомой строки с символом в строке базы данных. При такой организации распараллеливания соблюдается принцип memory coalescing [2], что позволяет свести к минимуму влияние латентности доступа к памяти ГПУ. Для распараллеливания на ГПУ используется разбиение кортежей реляционного отношения по блокам CUDA. На конечном этапе номера кортежей, содержащих совпавшие строки записываются в выходной массив. Передаются не кортежи целиком, а только их номера с целью уменьшить объем передаваемых данных по шине PCI Express. Шина PCI Express имеет на несколько порядков меньшую скорость передачи данных, чем оперативная память и внутренняя память ГПУ, поэтому данный подход позволяет значительно повысить общую производительность при обработке данных.

Вычислительные эксперименты

Вычислительные эксперименты проводились с использованием разработанного эмулятора СУБД на оборудовании, приведенном в таблице 1.

Табл. 1. Оборудование для экспериментов.

Оборудование	Характеристики	
Процессор	Intel Core i7-3610QM 2.3 ГГц	
ОЗУ	8 Гб DDR3-1600	
Твердотельный накопитель	Plextor M5 Pro PX-128M5P, 128Гб, SATA III	
Системная шина	PCI Express 2.0	
Графический ускоритель	Модель	NVIDIA GeForce GTX 670M
	Объем видеопамати	3072 Мб
	Ядер CUDA	336
	Тактовая частота ядра	620 МГц
	Тактовая частота памяти	3000 МГц
	Пропускная способность памяти	72 Гб/с
Разрядность памяти	192-bit	

Для проведения вычислительных экспериментов было сгенерировано 4 тестовых отношения. Размер отношений, а также коэффициент сжатия строк в кортежах представлены в таблице 2.

Табл. 2. Характеристики тестовых отношений

№ тестовой БД	Размер отношения (количество кортежей)	Коэффициент сжатия	Объем в сжатом виде, Мб
1	33 000 000	4	519
2		2	1039
3		1.33	1562
4		1	2077

Исследование эффективности операции выборки над сжатыми данными.

Было произведено исследование эффективности операции выборки над сжатыми данными в сравнении с обработкой несжатых данных.

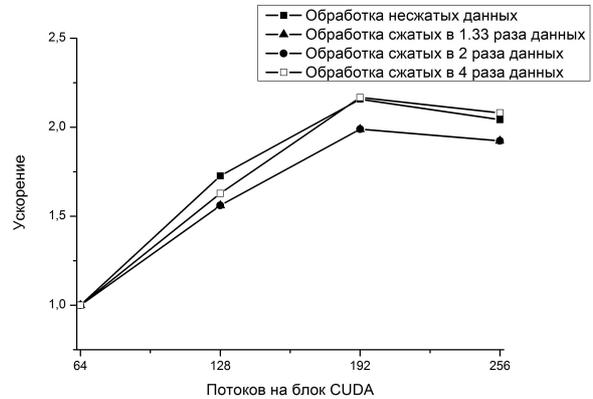
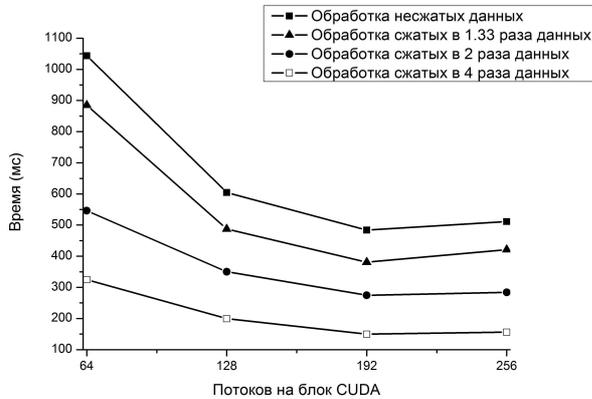


Рис. 1. Время выполнения алгоритма ГПУ SELECT на одном вычислительном узле

Рис. 2. Ускорение выполнения алгоритма ГПУ SELECT на одном вычислительном узле

Как видно из графиков времени выполнения операции выборки из тестовой базы данных с использованием ГПУ (см. рис. 1) и с использованием ЦПУ (см. рис. 3), при применении сжатия мы получаем максимальный прирост скорости в 3.2 раза на ГПУ и в 3 раза на ЦПУ.

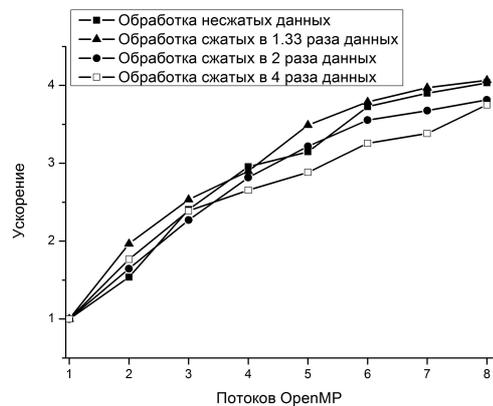
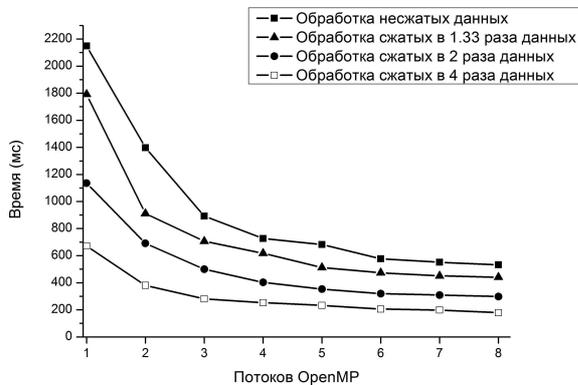


Рис. 3. Время выполнения алгоритма ЦПУ SELECT на одном вычислительном узле

Рис. 4. Ускорение выполнения алгоритма ЦПУ SELECT на одном вычислительном узле

На графике сравнения времени выполнения операции выборки на ГПУ и на ЦПУ (см. рис 5) видно, что при использовании ГПУ выборка выполняется быстрее независимо от размера базы данных.

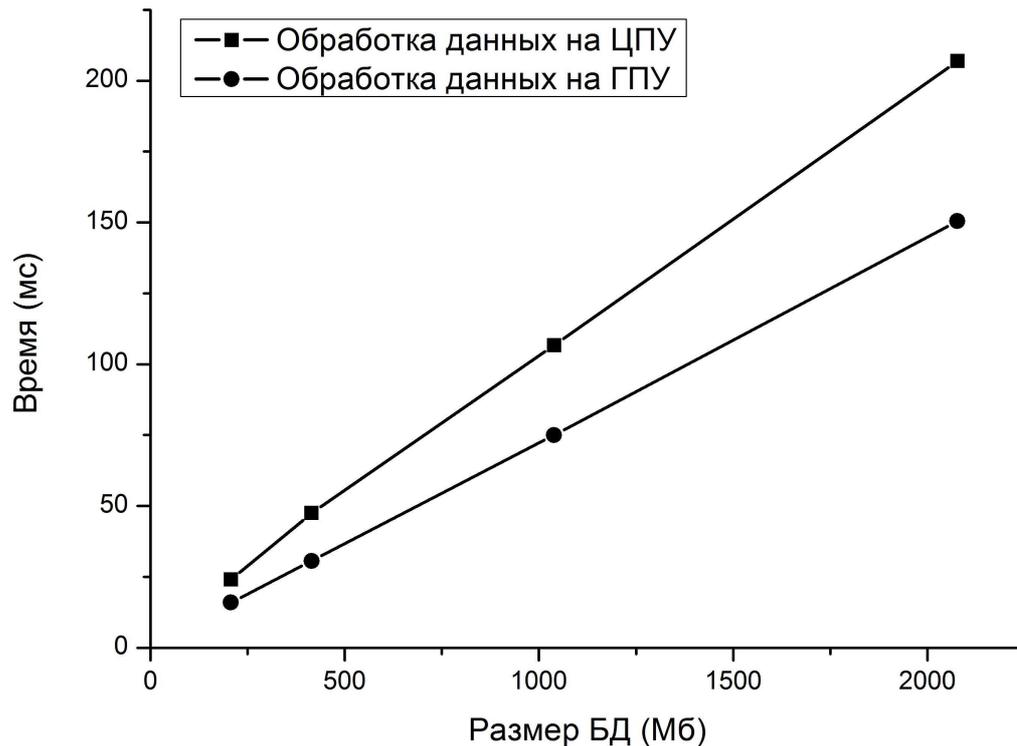


Рис. 5. Время выполнения алгоритмов ЦПУ и ГПУ SELECT на одном вычислительном узле над сжатыми в 4 раза данными

4. Заключение

Установлено, что выполнение выборки над сжатыми данными на ГПУ эффективнее, чем на ЦПУ в 1.2 раза. Небольшое преимущество ГПУ объясняется тем, что в качестве тестовой конфигурации имелся достаточно производительный процессор Intel Core i7-3610QM и не очень производительный графический ускоритель NVIDIA GeForce GTX 670M. На следующем этапе исследования тестирование будет произведено на новейших ГПУ NVIDIA Tesla K40 и многоядерных сопроцессорах Intel Xeon Phi.

Работа выполнена при поддержке гранта Президента РФ № МК-3711.2013.9 (2013-2014 гг.)

ЛИТЕРАТУРА:

1. Беседин К.Ю., Костенецкий П.С. Моделирование обработки запросов на гибридных вычислительных системах с многоядерными сопроцессорами и графическими ускорителями // Программные системы: теория и приложения: электрон. научн. журн. Института программных систем им. А.К. Айламазяна РАН. 2014. Т. 5, № 1(19), с. 91–110 2014.
2. Боресков А.В., Марковский Н.Д., Микушин Д.Н., Мортиков Е.В., Мыльцев А.А., Сахарных Н.А., Фролов В.А., Харламов А.А. Параллельные вычисления на GPU. Архитектура и программная модель CUDA. -М.: Издательство Московского университета, 2012. 58 с.
3. Костенецкий П.С., Соколинский Л.Б. Моделирование иерархических многопроцессорных систем баз данных // Программирование. Москва: МАИК "Наука/Интерпериодика". Vol. 39 No 1. 2013.
4. D. Abadi, S. Madden, M. Ferreira. Integrating compression and execution in column-oriented database systems // Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data, 2006. P. 671-682.