Библиотеки с поддержкой длинной арифметики на GPU

Собянин П.И.

Рассматриваются вычислительные библиотеки, использующие архитектуру CUDA для работы с целыми числами произвольной длины. Сравниваются возможности и производительность находящихся в открытом доступе таких библиотек.

Ключевые слова: длинная арифметика, GPU, CUDA.

1. Введение

Часто в задачах криптографии с открытым ключом требуется использование типов данных, не являющихся базовыми, в т.ч. чисел произвольной длины:

- существуют алгоритмы с открытым ключом, которые основаны на возведении в степень в группе или кольце. Например, в алгоритме RSA[1] – для шифрования передаваемого сообщения, в алгоритме DSA[1] – для генерации открытого ключа, в схеме Эль-Гамаля[1] и протоколе Диффи-Хеллмана[1] – для генерации секретного ключа шифрования.

- для криптографии важно исследовать решение задач факторизации[1] и дискретного логарифмирования[1] (например, для тех же алгоритмов RSA и DSA);

В работе исследуется возможность реализации "длинной" арифметики на GPU путем привлечения существующих решений.

Задача данной статьи – исследовать возможности и производительность библиотек, использующих архитектуру CUDA[2]. Насколько можно судить по имеющимся публикациям, сравнение имеющихся инструментов еще не проводилось.

Здесь мы сравним две таких библиотеки – **GARPREC**[3][4] и **CUMP**[5].

Дальнейшая часть статьи построена следующим образом: в разделе 2 рассматриваются приемы реализации библиотек и распараллеливания алгоритмов; раздел 3 посвящен описанию сравнительного эксперимента, его результатам и выводам; в разделе 4 подведены итоги текущей работы и освещены дальнейшие планы в предметной области.

2. Обзор реализаций

В настоящий момент разработано несколько библиотек, реализующих длинную арифметику на GPU: **GPUMP** [6] и уже упоминавшиеся **GARPREC** и **CUMP**. Вкратце опишем каждую.

GPU Multiple-Precision library (GPUMP). В 2010-м году Кайонгом Жао (Kaiyong Zhao) и Сяовенем Чжу (Xiaowen Chu) была представлена библиотека GPUMP, поддерживающая "длинную" арифметику для CUDA. Она способна выполнять арифметические операции с целыми числами произвольной фиксированной точности на архитектуре CUDA. Эта библиотека поддерживает операции сложения и вычитания (в т.ч. и по заданному модулю), умножения, деления, умножения и нахождения остатка от деления методами Монтгомери[7]. Все операции используют последовательные алгоритмы вычислений. Реализовано параллельное покоординатное выполнение операций для векторов операндов.

Авторы сравнивают производительность библиотек GPUMP и GNU MP, что поддерживает "длинные" вычисления на CPU. Проводятся тесты производительности для каждой операции над большим количеством пар чисел разного размера. В каждой паре размер чисел одинаков – 512, 1024 либо 2048 бит. Производительность измеряется в количестве операций в секунду. Авторы показывают, что производительность библиотеки GPUMP в 7-9 раз выше в зависимости от операции и размеров чисел. Использованное для расчетов оборудование: CPU – Intel Core i7 с рабочей частотой 2.8 ГГц, видеокарта – XFX GTX280 с GPU NVIDIA GT200 (240 вычислительных ядер, работающих на частоте 1.24 ГГц).

GARPREC. Эта библиотека основана на библиотеке **ARPREC** [8], что поддерживает арифметические операции с "длинными" числами на CPU. GARPREC также поддерживает основные арифметические опера-

ции, такие как сложение, умножение, взятие частного, корня, экспоненты, а так же логарифма, синуса и косинуса. Как и у первой библиотеки, все операции используют последовательные алгоритмы вычислений. Вычислительная сложность алгоритма сложения линейная по размеру операндов, а умножения – $O(n \log n)$.

Тесты производительности аналогичны тем, что применяются к библиотеке GPUMP – бинарные и унарные операции применялись к большому множеству чисел разных размеров; вычисления производились как на GPU, так и CPU. Прирост производительности расчетов на GPU по сравнению с CPU многократный: для сложения – в 12 раз, для умножения – в 8-9 раз в зависимости от размера чисел, для деления – в 10 раз.

CUMP. Эта библиотека была разработана Такатоши Накаямой (Takatoshi Nakayama) в 2012 году. Она основана на библиотеке **GMP** [9]. CUMP поддерживает операции сложения, вычитания и умножения. Вычислительная сложность алгоритма сложения – $O(n)$, сложность операции умножения зависит от алгоритма, который автоматически выбирается самой библиотекой в зависимости от размеров операндов, и может составлять как $O(n^2)$ (для алгоритма умножения "столбиком"), так и $O(n \log n)$ (для алгоритма быстрого преобразования Фурье[10]). Число n здесь – размер операндов в битах.

Во всех библиотеках каждая арифметическая операция выполняется на одной "нити" (вычислительном ядре) GPU, т.е. соответствующие процедуры не распараллелены. Реализовано параллельное покоординатное выполнение операций для векторов.

Необходимо отметить, что только две из описанных библиотек находятся в открытом доступе – это GARPREC и CUMP. В силу этого обстоятельства тестирование библиотеки GPUMP не проводилось.

3. Эксперимент

Описание. Мы тестировали две операции: сложение и умножение пары чисел. Такой выбор обусловлен тем, что именно эти две операции поддержаны одновременно в каждой из библиотек. В качестве входных данных эксперимента генерировались два случайных вектора. Для каждой операции проводилось два теста: в первом варьировался размер вектора при константном размере чисел (он был равен 1000 десятичных зна-

ков), во втором – размер чисел при константном размере вектора (100000 компонент).

Мы тестировали библиотеки на видеокарте Tesla C2070, на которой установлен графический процессор Tesla T20 с 448 вычислительными ядрами, работающих на частоте 1.15 ГГц.

Результаты. На рис. 1-2 по оси X отложена длина операндов, на рис. 3-4 – длина векторов. По оси Y на всех рисунках отложено время выполнения операции в секундах. На всех изображениях пунктирный график отражает результаты работы библиотеки GARPREC, сплошной – библиотеки CUMP.

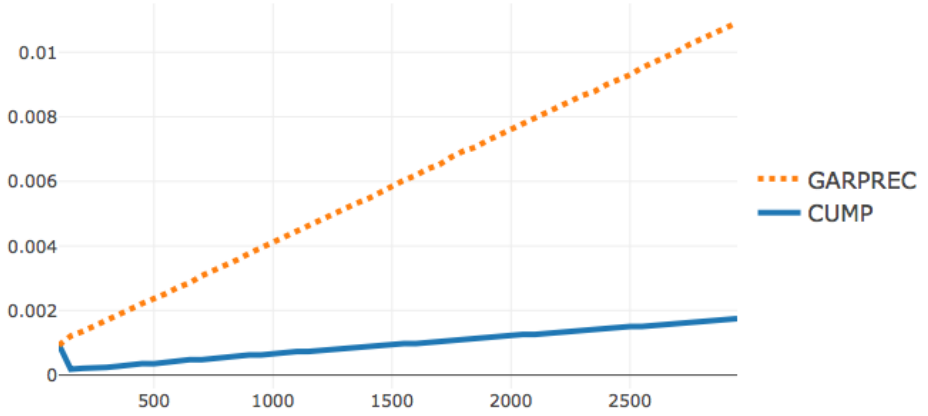

Рис. 1. Скорость сложения в зависимости от размера чисел.

Результаты тестов показывают, что порядок роста функций совпадает с теоретическим: в трех случаях из четырех рост линейный, а в оставшемся – порядка $O(n \log n)$, как и ожидалось.

Также видно, библиотека CUMP является более производительной, чем GARPREC: сложение в указанных условиях производится в 5 раз быстрее, умножение – в 20 раз при одинаковой сложности используемых алгоритмов. Причины этого отмечены в [11].

Но нельзя не отметить очень ограниченную функциональность CUMP: всего поддержаны лишь три простейшие арифметические операции, и это вряд ли делает ее пригодной для применениях в каких-то серьезных областях. Надеяться на развитие функционала этой библио-

теки также не приходится – последнее обновление её исходного кода датировано 2012-м годом[5].

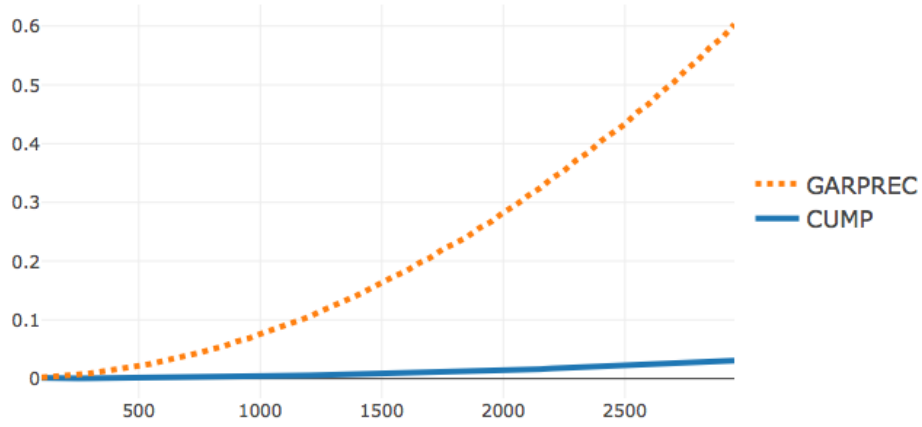

Рис. 2. Скорость умножения в зависимости от размера чисел.

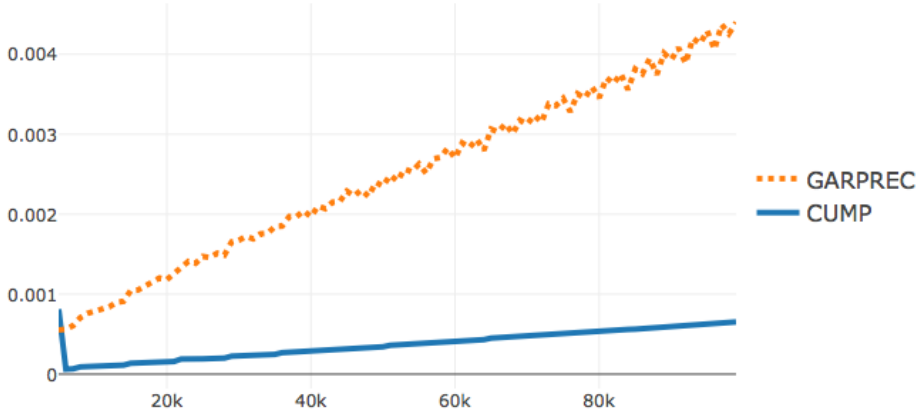

Рис. 3. Скорость сложения в зависимости от размера векторов.

4. Заключение

Операции с целыми числами произвольной длины являются важной составляющей криптографии с открытым ключом. В статье мы описали и сравнили возможности и производительность существующих библиотек,

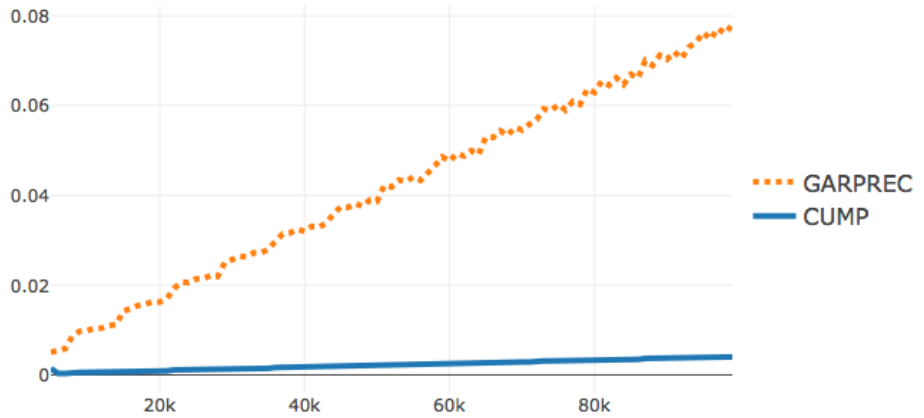

Рис. 4. Скорость умножения в зависимости от размера векторов.

поддерживающих работу с таким типом данных. В дальнейшем планируется реализовать собственную библиотеку с более широким функционалом.

Автор выражает искреннюю признательность А. В. Галатенко за постановку задачи и обсуждение результатов работы.

Список литературы

[1] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone. Handbook of Applied Cryptography. *CRC Press, Fifth Printing, August 2001.*

[2] NVIDIA CUDA. http://developer.nvidia.com/object/cuda.html.

[3] GARPREC: https://code.google.com/archive/p/gpuprec/.

[4] Mian Lu, Bingsheng He, Qiong Luo. Supporting Extended Precision on Graphics Processors.

[5] CUMP Library. https://github.com/skystar0227/CUMP.

[6] Kaiyong Zhao, Xiaowen Chu. GPUMP: a Multiple-Precision Integer Library for GPUs.

[7] Peter Montgomery. Modular Multiplication Without Trial Division. *Mathematics of Computation, vol. 44 no. 170, pp. 519–521, April 1985.*

[8] ARPREC. http://crd-legacy.lbl.gov/ dhbailey/mpdist/.

[9] GMP Library. https://gmplib.org/.

[10] Brigham, E. Oran. The Fast Fourier Transform. *New York, USA: Prentice-Hall, 2002.*

[11] https://mail.haskell.org/pipermail/glasgow-haskell-users/2006-September/010963.html

GPU Multiple-Precision Arithmetic Libraries

Sobyanin P.I.

CUDA libraries supporting multiple-precision integer arithmetic are considered. Features and performance of such open-source libraries are compared.

Keywords: multiple-precision arithmetic, GPU, CUDA.