РЕАЛИЗАЦИЯ ШИФРОВАНИЯ ПО СТАНДАРТУ AES НА ПРОЦЕССОРАХ CELL

К.С. Пан

Введение

Advanced Encryption Standard (AES) [1] представляет собой стандарт блочного симметричного шифрования, принятый в США в 2001 г. для шифрования данных с высшим уровнем секретности [2]. В соответствии с AES шифрование и расшифрование осуществляется с помощью одного и того же ключа над блоками данных фиксированного размера (16 байтов). Стандарт устанавливает различные режимы шифрования, каждый из которых определяет способ разбиения данных на блоки и шифрующие операции над данными.

В настоящее время, по-видимому, отсутствуют параллельные реализации стандарта AES для шифрования файлов. Одни из наиболее распространенных реализаций AES, библиотеки OpenSSL [3] и GnuPG [4] являются последовательными. В то же время имеются работы, посвященные разработке специализированных микропроцессоров для аппаратной реализации шифрования по стандарту AES [5, 6, 7].

Микропроцессорная архитектура Cell Broadband Engine [8] в настоящее время является одной из наиболее перспективных архитектур для построения параллельных вычислительных систем для решения различных научных и технических задач [9, 10].

В данной работе представлена параллельная версия алгоритма шифрования по стандарту AES, адаптированная для вычислительных систем на базе процессоров Cell.

Выбор режима шифрования для реализации на процессорах Cell

Для реализации шифрования по стандарту AES на процессорах Cell нами было выбрано шифрование в режиме счетчика [1]. Схема шифрования в режиме счетчика представлена на иллюстрации.

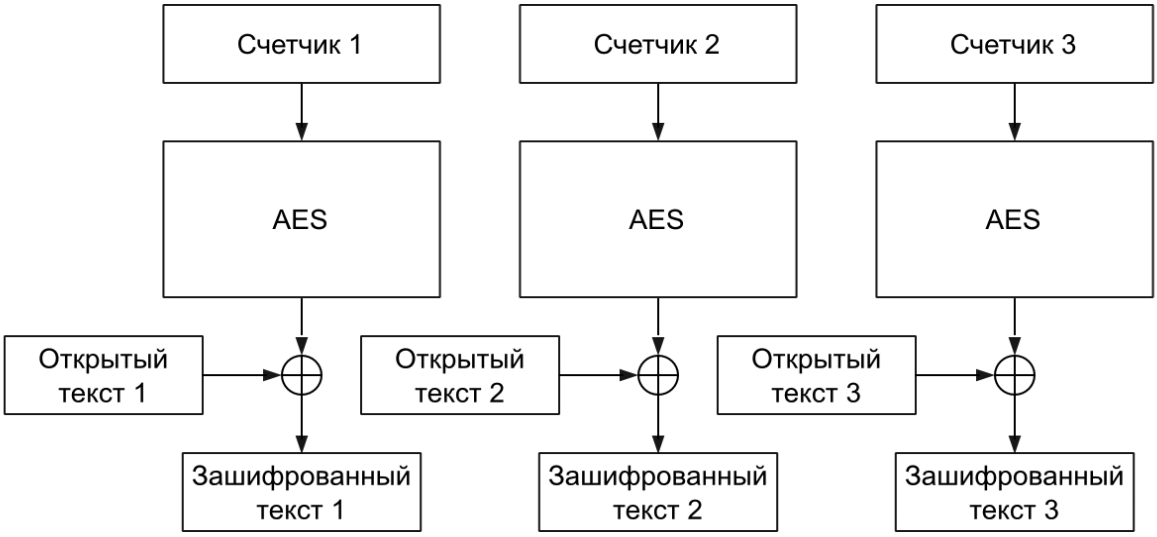

Рис. 1

Счетчик представляет собой переменную, размер которой равен размеру блока. Перед шифрованием данных счетчику присваивается случайное значение. Для каждого блока соответствующее значение счетчика подвергается шифрующим преобразованиям, определенным в стандарте. Измененное значение счетчика складывается с блоком с помощью побитовой операции XOR. После шифрования каждого блока счетчику присваивается новое уникальное значение (например, путем добавления единицы к текущему значению). Начальное значение счетчика сохраняется вместе с зашифрованными данными.

В параллельной версии шифрования в режиме счетчика может быть применена концепция параллелизма по данным: шифрование блоков текста может осуществляться независимо отдельными ядрами процессора Cell.

Архитектура процессора Cell

Процессор Cell имеет асимметричную многоядерную архитектуру [8], представленную на иллюстрации.

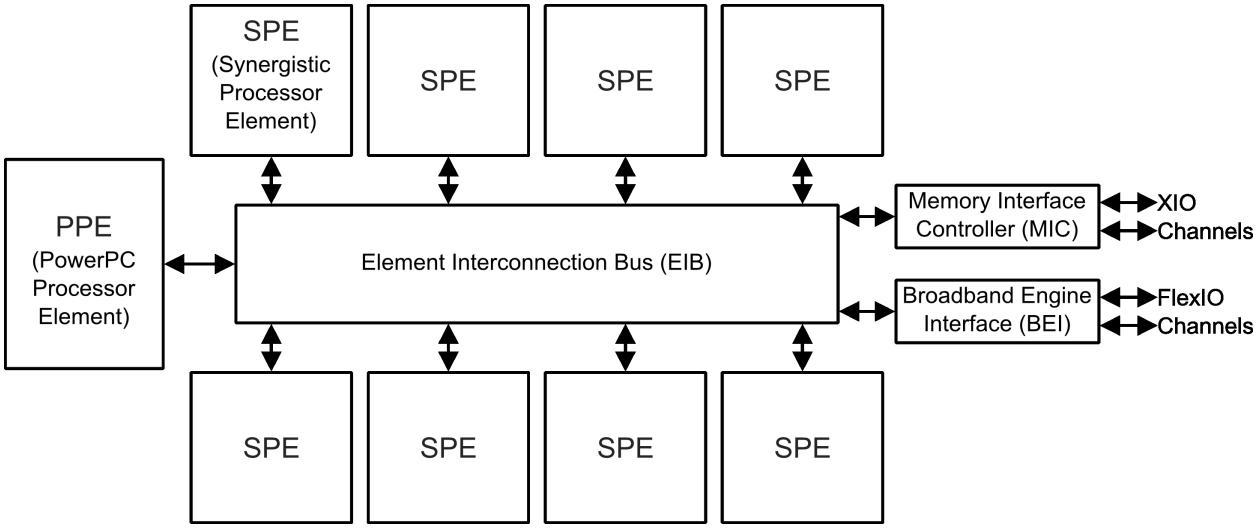

Рис. 2

Процессор Cell состоит из девяти ядер. Одно из них, Power Processor Element (PPE), является управляющим, а остальные, Synergistic Processor Element (SPE) — вычислительными. Вычислительные ядра не могут напрямую обрабатывать данные из основной памяти, но обладают собственной локальной памятью размером 256 Кбайт. Все ядра поддерживают набор векторных инструкций, позволяющих работать с 16-байтными векторами.

Проектирование алгоритма

В соответствии с особенностями архитектуры процессора Cell нами была разработана параллельная версия алгоритма шифрования по стандарту AES в режиме счетчика, названная нами PAES-CTR [11]. В основе PAES-CTR лежит концепция параллелизма по данным: исходные данные разбиваются на порции, которые параллельно шифруются на нескольких вычислительных ядрах SPE.

В разработке PAES-CTR нами также применена модель «мастер-рабочие». Приложение-«мастер» запускается на управляющем ядре PPE и занимается распределением заданий для «рабочих». «Рабочие» запускаются как нити на вычислительных ядрах SPE и выполняют шифрование блоков данных и передачу результата «мастеру».

На иллюстрации представлена диаграмма классов на языке UML, реализующих алгоритм PAES-CTR.

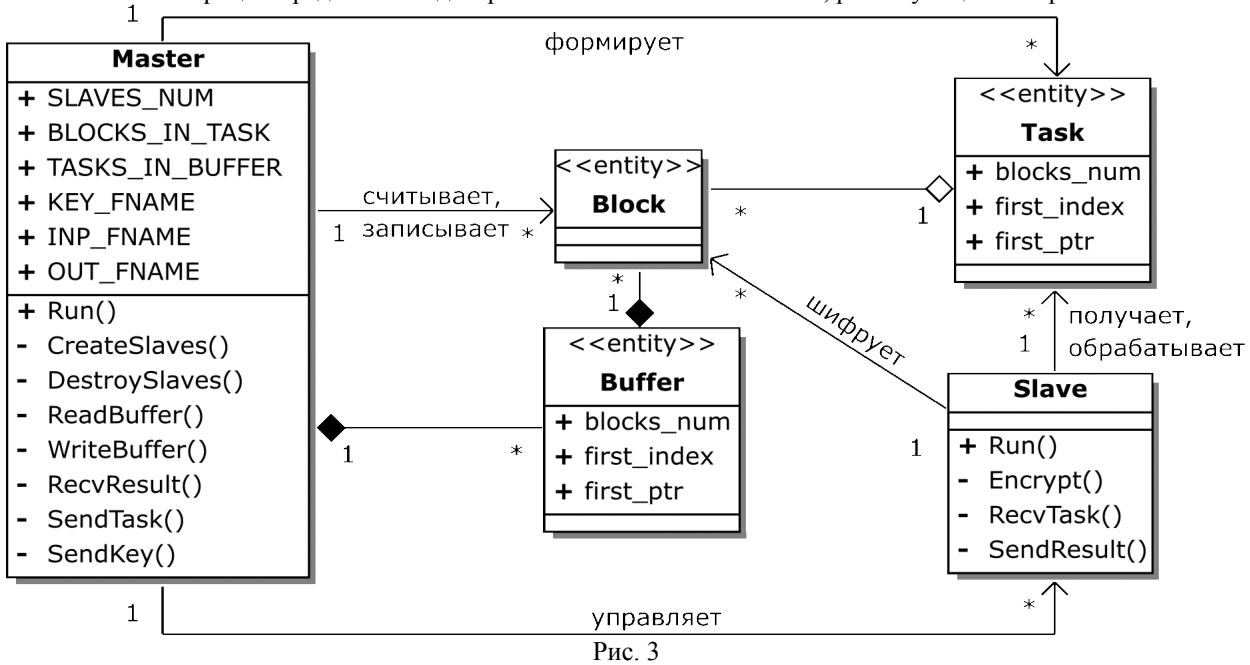

Рис. 3

Класс Master выполняет следующие основные функции: чтение и запись блоков, формирование и отправка заданий, управление экземплярами класса Slave. Задание реализуется как класс-сущность Task с атрибутами «адрес в основной памяти» и «количество блоков» данных для шифрования. Класс Slave выполняет следующие основные функции: получение блоков открытого текста от класса Master, шифрование блоков и передача зашифрованных блоков классу Master.

На иллюстрации представлены диаграммы деятельности классов Master и Slave на языке UML.

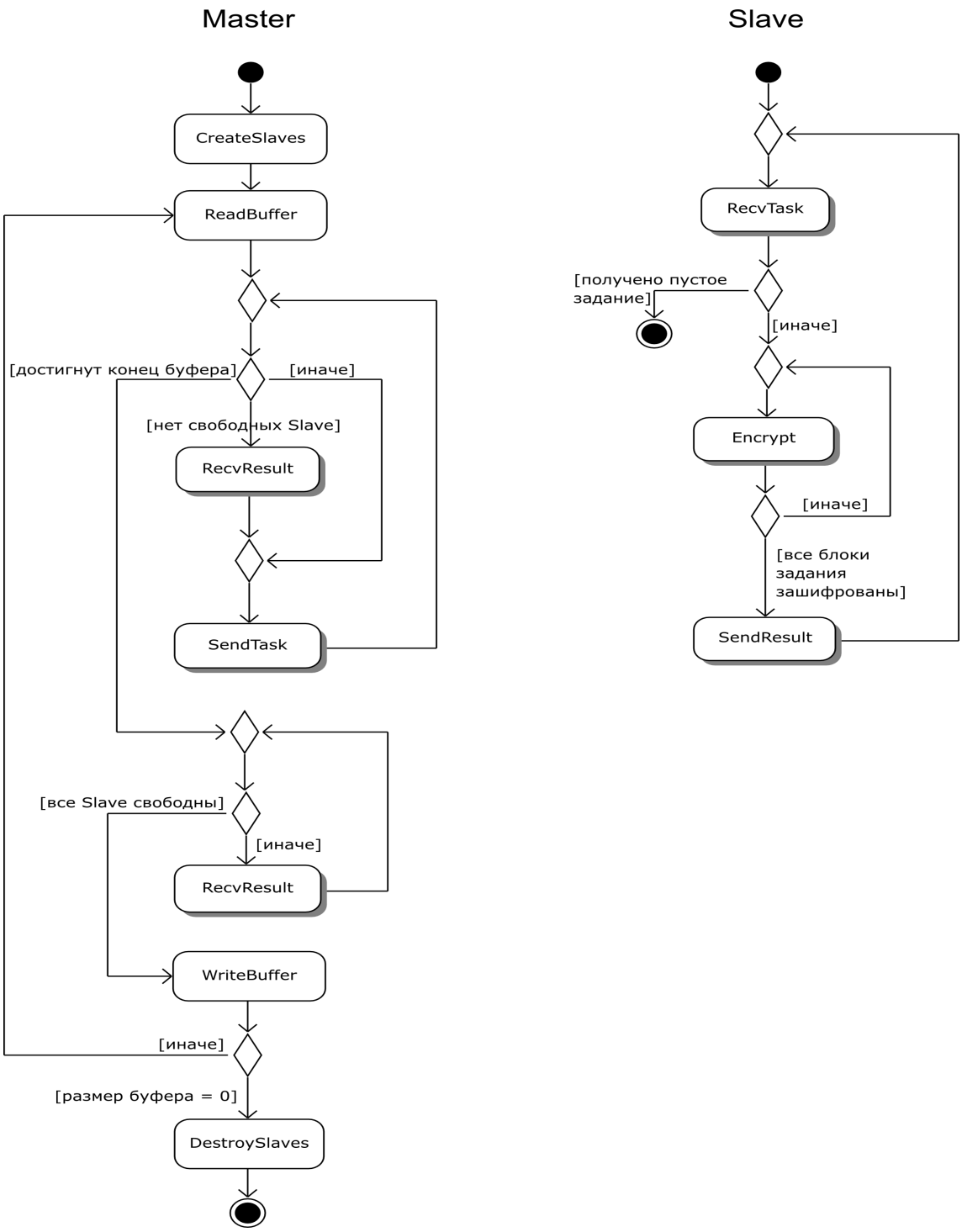

Рис. 4

Шифрование происходит следующим образом. Мастер загружает данные из файла в основную память, разбивает данные на порции и отправляет задания рабочим. После получения заданий рабочие загружают блоки в локальную память и выполняют их шифрование. По окончании шифрования рабочий отправляет зашифрованные блоки обратно в основную память и посылает сигнал готовности мастеру. После этого он получает от мастера следующее задание и приступает к его выполнению. Шифрование завершается, когда мастер получает сигналы от всех рабочих об успешном завершении работы. Зашифрованные данные записываются мастером на жесткий диск.

Реализация алгоритма

Описанный выше алгоритм был реализован нами на языке C++ с использованием IBM SDK for Multicore Acceleration [12]. Файловая структура программной системы в виде диаграммы компонентов на языке UML представлена на иллюстрации.

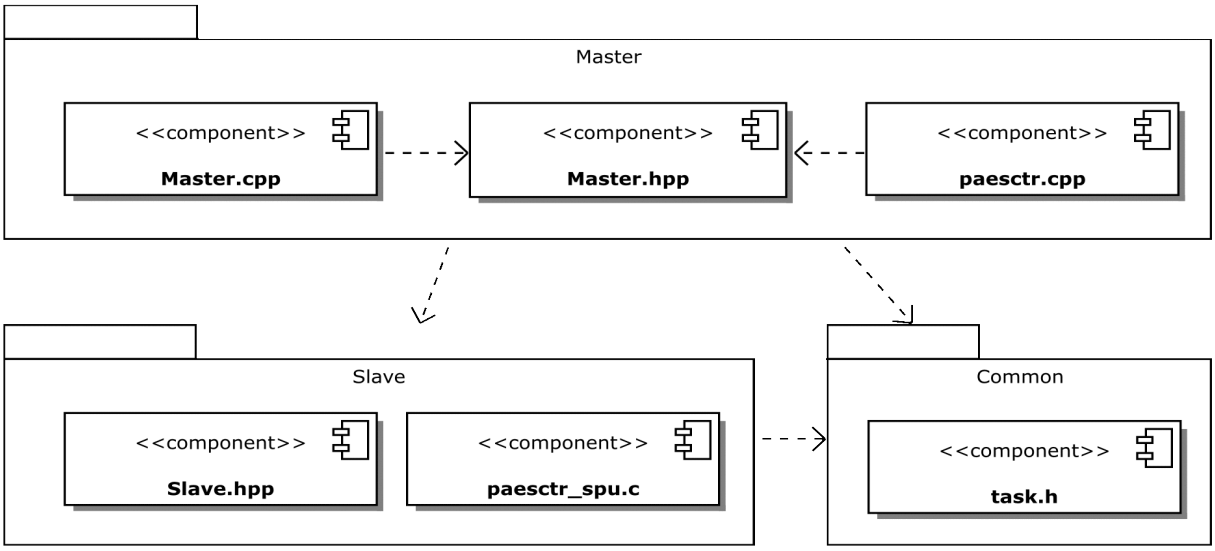

Рис. 5

Основные шифрующие преобразования стандарта AES ShiftRows, MixColumns и AddRoundKey [1] реализованы нами с помощью набора векторных инструкций, предоставляемых в IBM SDK for Multicore Acceleration [12]. В данном случае совпадение размера вектора, которым оперирует процессор Cell, с размером блока в стандарте AES (16 байт), позволяет существенно ускорить выполнение шифрования.

Для исключения простоев вследствие ожидания загрузки шифруемых данных нами реализована опережающая загрузка блоков данных в основную память. Здесь параметром выполнения шифрования является доля заданий, находящихся в основной памяти, после выполнения которых мастер осуществляет загрузку новой порции данных с диска.

Вычислительные эксперименты

Для исследования эффективности предложенного алгоритма нами были проведены вычислительные эксперименты на вычислительной системе на базе процессоров Cell, технические характеристики которой таковы:

Архитектура процессора — PowerXCell 8i

Количество процессоров — 2

Тактовая частота процессора — 3.2 GHz

Пиковая производительность процессора — 204.8 Gflops (25.6 Gflops на одно ядро SPE)

Эксперименты преследовали следующие основные цели. Во-первых, определить ускорение алгоритма PAES-CTR. Во-вторых, сравнить быстродействие PAES-CTR с другими реализациями стандарта AES вычислительных системах на базе процессоров Cell и на вычислительных системах на базе процессоров Intel. В экспериментах производилось шифрование данных объемом 1 ГБ с помощью ключа размером 128 битов.

Результаты экспериментов по определению ускорения представлены на иллюстрации.

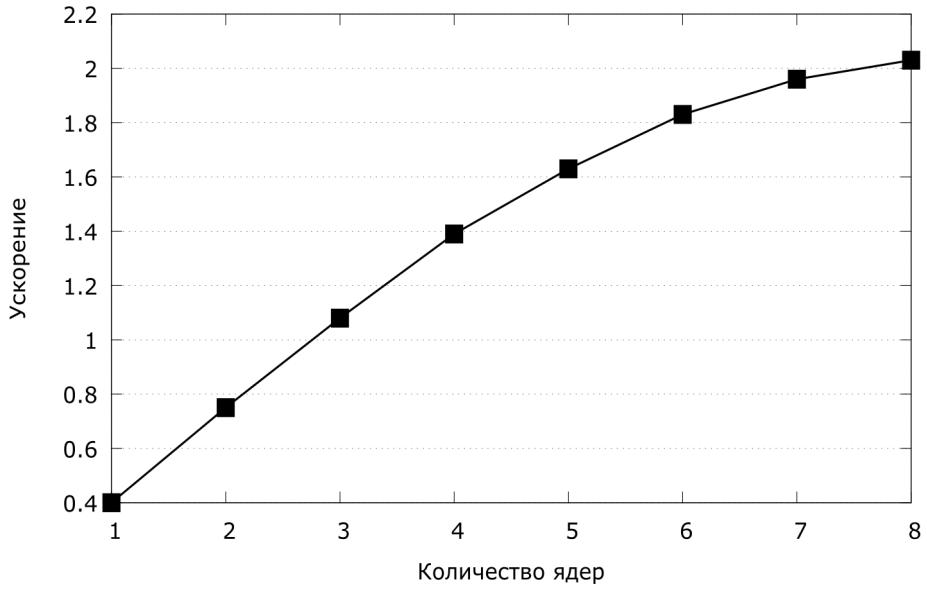

Рис. 6

Ускорение алгоритма PAES-CTR вычислялось относительно последовательной реализации OpenSSL [3]. Результаты экспериментов свидетельствуют, что алгоритм PAES-CTR демонстрирует линейное ускорение.

В сравнительных экспериментах быстродействие алгоритма PAES-CTR сравнивалось нами с быстродействием реализаций OpenSSL [3] и GnuPG [4]. Сравнение выполнялось на вычислительной системе на базе процессоров Cell (см. выше) и на узлах суперкомпьютера СКИФ Урал [13] на базе процессоров Intel, технические характеристики которых таковы:

Архитектура процессора — Intel Xeon E5472
Количество процессоров — 2
Количество ядер в процессоре — 4
Тактовая частота процессора — 3.0 GHz
Пиковая производительность процессора — 48.2 Gflops (12.0 Gflops на одно ядро)
Результаты сравнительных экспериментов представлены на иллюстрации.

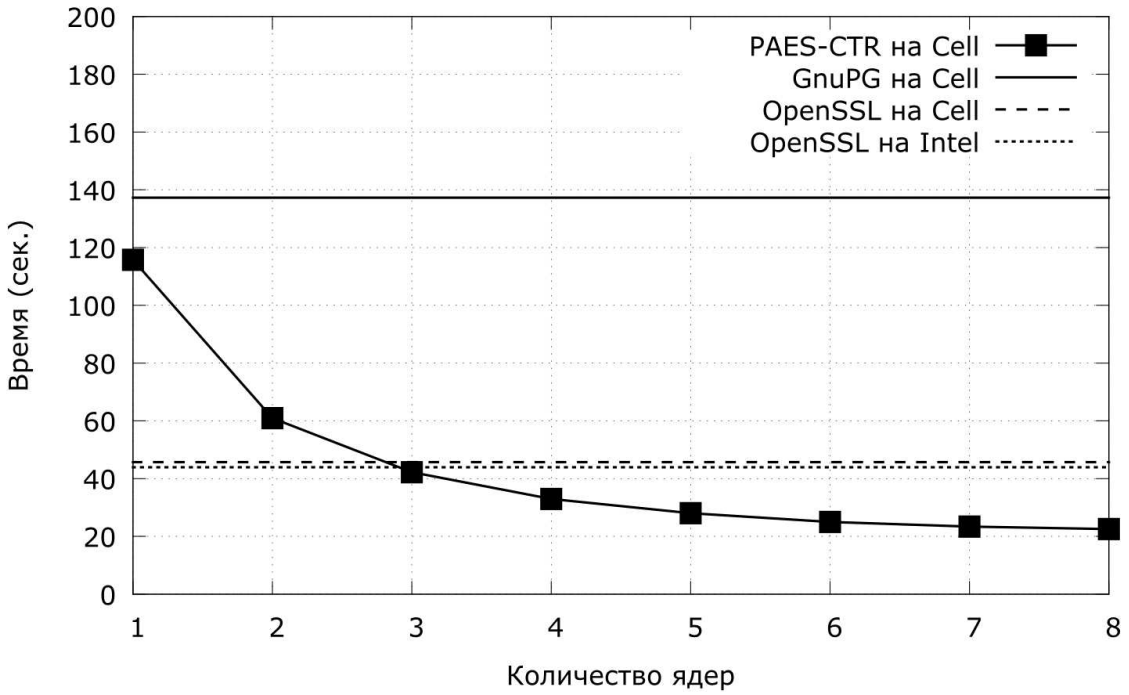

Рис. 7

Результаты экспериментов свидетельствуют, что при использовании трех и более вычислительных ядер, алгоритм PAES-CTR демонстрирует лучшее быстродействие по сравнению другими реализациями.

Заключение

В работе описана параллельная версия алгоритма блочного симметричного шифрования по стандарту AES, учитывающая особенности архитектуры многоядерного процессора Cell.

В реализации использована модель параллельного программирования «мастер-рабочие». Приложение-«мастер» запускается на ядре PPE и занимается распределением заданий для «рабочих». «Рабочие» запускаются как нити на ядрах SPE и выполняют шифрование блоков данных и передачу результата «мастеру».

Представлены результаты вычислительных экспериментов, подтверждающих эффективность разработанного алгоритма.

ЛИТЕРАТУРА:

1. Specification for the Advanced Encryption Standard (AES) / National Institute of Standards and Technology. URL: http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf (дата обращения: 01.06.2009).
2. CNSS Policy No. 15, Fact Sheet No. 1, National Policy on the Use of the Advanced Encryption Standard (AES) to Protect National Security Systems and National Security Information. URL: http://csrc.nist.gov/groups/ST/toolkit/documents/aes/CNSS15FS.pdf (дата обращения: 01.06.2009).
3. OpenSSL Project. URL: http://www.openssl.org (дата обращения: 01.06.2009).
4. GnuPG Project. URL: http://www.gnupg.org (дата обращения: 01.06.2009).
5. Damaj I. Parallel AES Development for Programmable Devices // PDCN 2009: Int. Conf. on Parallel and Distributed Computing and Networks (2009, Innsbruck, Austria).
6. Su Y.X., Mathew J., Singh J., Pradhan D.K. Pseudo parallel architecture for AES with error correction // SOC 2008: Int. Conf. (September, 17-20, 2008). P. 187-190.

7. Hua L., Chang N.Z., Jianzhou L. A CAM Based Associative Processor Array for Parallel Implementation of AES // ISCA: Int. Journal of Comput. Applications. No. 13(4). 2006. P. 176-181.
8. IBM Corporation. Cell Broadband Engine technology. URL: http://www.alphaworks.ibm.com/topics/cell (дата обращения: 01.06.2009).
9. Williams S., Shalf J., Oliker L. The Potential of the Cell Processor for Scientific Computing // Proceedings of the 3rd conference on Computing frontiers (2006, Ischia, Italy). 2006. P. 9-20.
10. Crawford C.H., Henning P., Kistler M., Wright C. Accelerating computing with the cell broadband engine processor // Proceedings of the 5th conference on Computing frontiers (2008, Ischia, Italy). 2009. P. 3-12.
11. Пан К.С. Адаптация алгоритма блочного симметричного шифрования AES для вычислительных систем на базе процессоров Cell: Дис. ... бакалавра информационных технологий: 010400.62 / Южно-Уральский государственный университет. Челябинск, 2009. 46 л. URL: http://omega.sp.susu.ac.ru/publications/bachelorthesis/09-Pan.pdf (дата обращения: 08.06.2009).
12. IBM Cell Broadband Engine SDK, Version 3.0 documentation. URL: http://www-01.ibm.com/chips/techlib/techlib.nsf/products/IBM_SDK_for_Multicore_Acceleration (дата обращения: 01.06.2009).
13. Вычислительный кластер СКИФ Урал. URL: http://supercomputer.susu.ru/computers/ckif_ural/ (дата обращения: 08.06.2009).