

*Соколов Н.Д. Обработка информации с применением анализа методов и технологий параллельных вычислений // Академия педагогических идей «Новация». Серия: Студенческий научный вестник. – 2017. – № 07 (июль). – АРТ 330-эл. – 0,2 п.л. - URL: <http://akademnova.ru/page/875550>*

**РУБРИКА: ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**УДК 004.04**

**Соколов Никита Дмитриевич**  
студент 2 курса, группы 15КС-1  
*Научный руководитель:* Зорина М.  
А., преподаватель  
ОГАПОУ «УАвиаК-МЦК»,  
г. Ульяновск, Российская  
Федерация  
e-mail: [mary-hist@rambler.ru](mailto:mary-hist@rambler.ru)

**ОБРАБОТКА ИНФОРМАЦИИ С ПРИМЕНЕНИЕМ АНАЛИЗА  
МЕТОДОВ И ТЕХНОЛОГИЙ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ**

*Аннотация:* В статье рассмотрены основы параллельных вычислений и особенности организации обработки программных потоков в операционной системе Windows.

*Ключевые слова:* кэш-память; поток; процесс; ядро; CPU; CUDA.

**Sokolov Nikita**  
2nd year student, group 15CS-1  
*Supervisor:* Zorina M. A., lecturer  
AGAPO "Aviac-ICC",  
Ulyanovsk, Russian Federation  
e-mail: [mary-hist@rambler.ru](mailto:mary-hist@rambler.ru)

## **PROCESSING INFORMATION WITH THE USE OF ANALYSIS METHODS AND TECHNOLOGIES OF PARALLEL COMPUTING**

*Abstract:* The article covers the basics of parallel computing, and features processing the software threads in the Windows operating system.

*Keywords:* the cache memory; a thread; a process; a kernel; the CPU; CUDA.

На сегодняшний день все большую роль играют технологии, обеспечивающие анализ, хранение и обработку данных. Это связано с огромным, непрерывным ростом информации, который начался с конца XX века. 2017 год, время, когда предъявляются особые требования, связанные с анализом данных: к компьютерам, к их приложениям, и к вычислительным ресурсам. Эти требования зачастую значительно превышают мощность отдельных компьютеров.

В последние 30 лет одним из действенных методов повышения производительности компьютеров было увеличение тактовой частоты. А все началось с первых персональных компьютеров, появившихся в начале 1980-х годов, где генератор тактовых импульсов внутри CPU работал на частоте 1 МГц. В наше время у большинства настольных компьютеров тактовая частота процессора составляет от 1 до 4 ГГц, это примерно в 1000 раз быстрее своих родоначальников. Следует отметить, что самый надежный способ повышения производительности вычислений является увеличение частоты тактового генератора.

Сегодня инженеры оказались перед острой необходимостью искать замену данному традиционному источнику повышения быстродействия. Получение дополнительной производительности от существующих

архитектур уже невозможно рассчитывать на увеличение тактовой частоты процессора, как средство фундаментальных ограничений при производстве интегральных схем. Инженеры и производители ищут все новые и новые решения о преодолении ограничений: на потребляемую мощность, на тепловыделение и на быстро приближающийся физический предел размера транзистора.

Итак, графический процессор, рассмотрим его особенности функционирования. У графического процессора своя собственная архитектура вычислений, и совсем недавно было довольно трудно себе представить, чтобы в каких-либо вычислениях процессору помогали графические чипы, кроме как для них предназначенных (поддержка GUI, обработка графики, векторные вычисления). Но сегодня появились новые технологии, они позволяют производить вычисления с использованием графических процессоров, поддерживающих технологию GPGPU (произвольных вычислений на графических процессорах).

CUDA (Compute Unified Device Architecture) - одна из передовых технологий - осуществляющая вычисления связанные с математическими моделями и формулами в десятки, и даже в сотни раз быстрее, а также позволяющая обрабатывать трудоемкие алгоритмы и большие массивы данных. А также большим плюсом является то, что при этом не обязательно покупать огромные вычислительные центры и мощные сервера.

Чтобы выявить основные отличия архитектуры CUDA от других существующих технологий распараллеливания, необходимо понимать, чем графические процессоры отличаются от стандартных многопроцессорных систем.

Итак, рассмотрим стандартную архитектуру процессора:

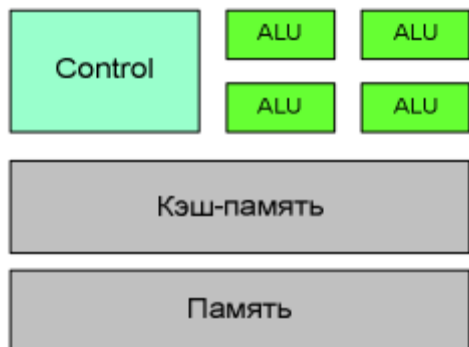


Рисунок 1. Архитектура центрального процессора

Как мы видим (Рисунок 1), кэш-память занимает большую часть площади кристалла, и лишь четверть кристалла занимают вычислительные модули (арифметико-логическое устройство, АЛУ). Каждый отдельный АЛУ является полноценным центральным процессором, который способен поддерживать все аппаратные прерывания, а также может работать со всеми устройствами ввода/вывода. Это несомненно полезная функция для его работы, как центрального процессора. При этом он становится излишним для использования в роли векторного вычислительного модуля.

Кэш-память - это высокоскоростное запоминающее устройство небольшой емкости для временного хранения данных. Кэш-память значительно более быстро действующая, чем основная память. Её основное отличие от оперативной памяти, не адресуемое и непосредственно «невидимое» положение для программиста (Рисунок 2).

Главным назначением кэш-памяти являются:

1. снабжение быстрого доступа к интенсивно используемым данным;
2. упреждающая загрузка данных;
3. согласование интерфейсов процессора и контроллера памяти;

#### 4. отложенная запись.

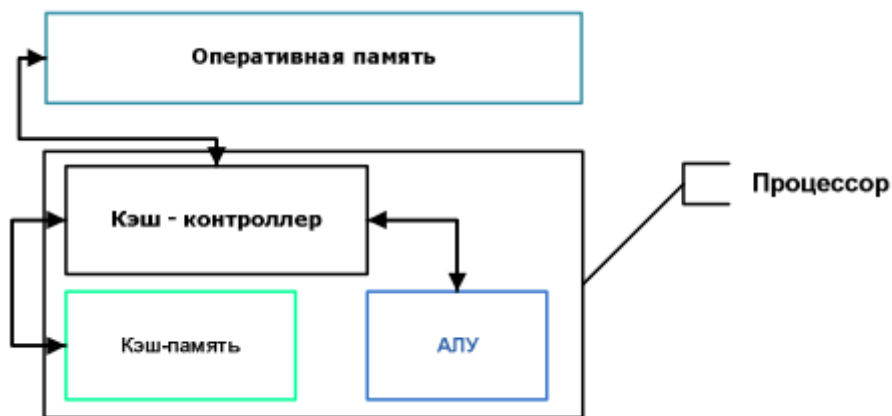


Рисунок 2. Кэш-память в иерархии оперативной памяти

Кэш-контроллер перехватывает запросы, направляемые к основной памяти, проверяет их на наличие копий затребованных данных в кэш-памяти. Если ответ положительный, то произошло так называемое кэш-попадание, если ответ отрицательный, то говорят о промахе, и тогда данные переадресуются к основной памяти. Помимо этого, потоки исполнения (код программы) являются очень «тяжеловесными», ими управляет операционная система. Таким образом их не может быть много (максимальное количество измеряется несколькими тысячами).

Рассмотрим более детально параллельное программирование в операционной системе Windows. Для этого мы разберем следующие понятия: процесс, поток исполнения и контекст переключения.

Процесс определяют как экземпляр выполняемой программы, он состоит из двух компонентов:

- 1) объект ядра, через него операционная система управляет процессом (в ядре хранится статическая информация о процессе);

2) адресного пространства, в нем содержится код и данные всех EXE и DLL библиотек.

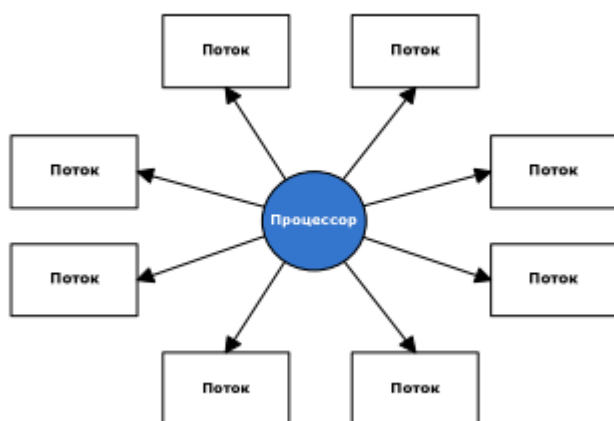


Рисунок 3. Операционная среда Windows и планирование потоков

Сами по себе процессы инертны. Для того чтобы процесс что-нибудь выполнил, в нем создается поток выполнения. За исполнение кода отвечают потоки, содержащегося в адресном пространстве процесса. Один процесс может владеть несколькими потоками. Тогда они одновременно исполняют код в адресном пространстве процесса. В идеале каждый поток выполнения должен располагать собственным стеком и собственным набором регистров процессоров. Минимум один поток выполнения есть в каждом процессе.

Квант – определенное процессорное время, которое отводит операционная система для работы всех потоков. По принципу карусели потокам выделяются отрезки времени, тем самым создается иллюзия одновременного выполнения потоков. На одной машине с одним процессором происходит распределение процессорного времени между потоками (Рисунок 3). При окончании времени потока, операционная система прерывает его выполнение и переключается на другой поток.

Этому сопутствует следующее:

1. в структуре контекста сохраняются значения регистров процессора исполняющегося в данный момент поток;
2. из всех потоков выбирается тот, к которому перейдет управление;
3. в регистры процессора загружаются значения из выбранной структуры контекста потока.

Затем происходит переключение контекста, процессор исполняет выбранный поток, до тех пор пока не истечет выделенное потоку время, после этого следует переключение контекста. Как мы видим, создаются разные парадигмы параллельного программирования на CPU из-за того, что потоки в операционной системе Windows, являются «тяжеловесными».

В помощь приходит технология CUDA. CUDA – это архитектура параллельных вычислений от NVIDIA. CUDA позволяет существенно увеличить вычислительную производительность за счет использования графического процессора. Поэтому, данная технология упрощает написание параллельного кода и предлагает использовать параллелизм на уровне данных. А также, вычислительная архитектура CUDA основана на концепции «одна команда – множество данных» (Single Instruction Multiple Data, SIMD).

Данная концепция SIMD подразумевает, что одна инструкция позволяет одновременно обработать множество данных. Мультипроцессор (многоядерный SIMD процессор) - позволяет в каждый определенный момент времени выполнять на всех ядрах только одну инструкцию. А модель программирования в CUDA предполагает группирование потоков, а также позволяет работать с блоками, содержащими от 64 до 512 потоков,

что позволяет одновременно обрабатывать множество данных по разным потокам.

Таким образом, в современных параллельных вычислениях становится нормой использование данной технологии в анализе, обработке данных и разработке крупномасштабных приложений. Без применения параллельного вычисления будут невозможны такие решения, как в области интеллектуального анализа данных, в разработке эффективных алгоритмов. А также, в ходе нашего анализа мы выявили, что технология CUDA позволяет облегчить написание параллельного кода, посредством использования концепции SIMD и переносит вычисления на графический процессор, что является неотъемлемой частью в современном программировании.

#### Список использованной литературы:

1. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. / А.В. Боресков, А.А. Харламов - М.: ДМК-Пресс, 2010. - 232 с.
2. Рихтер Дж. Windows для профессионалов: создание эффективных Win32-приложений с учетом спецификации 64-разрядной версии Windows, 4-е издание / Дж. Рихтер – Спб.: Питер, 2001. – 752 с.
3. Сандерс Дж., Кэндрот Э. Технология CUDA в примерах: введение в программирование графических процессоров: перевод с английского / Дж. Сандерс, Э. Кэндрот – М.: ДМК Пресс, 2011. – 234 с.
4. Старченко А.В. Практикум по методам параллельных вычислений / А.В. Старченко – М.: Издательство Московского университета, 2010. – 200 с.
5. Сухорослов О.В. Новые технологии распределенного хранения и обработки больших массивов данных // Всероссийский конкурсный отбор обзорно-аналитических статей по приоритетному направлению «Информационно-телекоммуникационные системы» - М.: 2008. - 40 с.

*Дата поступления в редакцию: 02.07.2017 г.*

*Опубликовано: 05.07.2017 г.*

*© Академия педагогических идей «Новация». Серия «Студенческий научный вестник», электронный журнал, 2017*

*© Соколов Н.Д., 2017*