doi: 10.14498/tech.2020.3.2

УДК 004.056

МЕТОДИКА РАСПРЕДЕЛЕНИЯ ЗАДАЧ ЗАЩИТЫ ИНФОРМАЦИИ МЕЖДУ ВЫЧИСЛИТЕЛЬНЫМИ УСТРОЙСТВАМИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ НА ОСНОВЕ МЕТОДА ВЕТВЕЙ И ГРАНИЦ

М.С. Гнутов, А.Б. Сизоненко

Краснодарское высшее военное орденов Жукова и Октябрьской Революции Краснознаменное училище имени генерала армии С.М. Штеменко
Россия, 350063, г. Краснодар, ул. Красина, 4

Аннотация. *Проанализировано влияние программных средств защиты на различные информационные системы. С помощью теории множеств описано задействование вычислительных ресурсов для совместного решения задач прямого назначения и задач защиты информации в автоматизированной системе. Предложена гибридная реализация вычислений в системе CPU+GPU. Рассмотрена актуальность использования метода ветвей и границ для составления минимального расписания задач защиты информации в гибридной многопроцессорной системе. Обозначены особенности обработки структур данных различными видами вычислителей. Проанализированы вычислительные стратегии метода ветвей и границ с наибольшей возможностью ускорения при ограниченных ресурсах. Выбран критерий эффективности, рассмотрены показатели эффективности при применении фронтальной и односторонней стратегии ветвления в зависимости от сложности вычислений и объема занимаемой памяти. Определен обобщенный показатель эффективности. Подчеркивается перспективность применения рассмотренного метода в распределенных системах посредством распределенного программирования.*

Ключевые слова: *метод ветвей и границ, графический процессор, центральный процессор, расписание для многопроцессорной системы, жадный алгоритм, распределенное программирование.*

Введение

В составе современных автоматизированных систем используются сложные средства вычислительной техники, обеспечивающие обработку задач и выработку решений в разнообразных сферах жизнедеятельности человека. Для графических вычислений в данных системах традиционно используются графические процессоры (GPU), для остальных расчетов применяются более многофункциональные, но менее многоядерные центральные процессоры (CPU).

Графические процессоры – это высокопроизводительные многоядерные процессоры. Возможность использования потенциала графического процессора в неграфических задачах привела к росту интереса пользователей и технических специалистов к графическим GPGPU и гибридным CPU+GPU вычислениям. Использование гибридных CPU+GPU систем для решения задач оптимизации систем защиты информации является перспективным направлением: при небольшом

Гнутов Максим Сергеевич, адъюнкт отдела научной работы и подготовки научно-педагогических кадров.

Сизоненко Александр Борисович (д.т.н., доцент), начальник кафедры «Защита информации от несанкционированного доступа».

энергопотреблении обеспечивается высокая производительность – радикально сокращается время, необходимое для вычислений и получения оптимальных решений. Сложность вычислений на GPU заключается как в обработке нерегулярных структур данных, которые не очень подходят для вычислений на графическом процессоре, так и в передаче информации между группами процессоров. Простая структура, скудный набор инструкций и отсутствие кэш-памяти графических ядер накладывают свои ограничения.

Как известно, существуют следующие основные программные методы обнаружения компьютерных вирусов:

– сканирование;
– обнаружение изменений;
– резидентные "сторожа";
– эвристический анализ;
– вакцинирование программных средств.

Такие действия, как, например, сканирование программных средств с использованием известных вирусных сигнатур (или их контрольных сумм), достаточно цикличны и позволяют с высокой эффективностью реализовать частичную обработку таких данных на графическом процессоре. Таким образом, использование распределенных вычислений на комбинированных CPU+GPU системах поможет перераспределить незадействованные системой ресурсы и в перспективе увеличить общее быстродействие автоматизированной системы. Задача центрального процессора будет состоять в обработке непараллельных вычислений, требующих более сложных наборов инструкций и организации обмена данных и вычисленных решений между множеством ядер графического ускорителя. Однако организация обмена данных также может создать очередь и нивелировать все преимущества комбинированных вычислений, поэтому использование CPU+GPU систем в той или иной степени сводится к задаче составления расписания для многопроцессорных систем. В статье [1] автор на примере описывает преимущества метода ветвей и границ над полным перебором множества всех имеющихся вариантов решения задачи коммивояжера. Начиная с восьми городов в маршруте преимущества метода ветвей и границ становятся явными. В современных же моделях видеоадаптеров количество процессоров измеряется не десятками, а тысячами – GeForce RTX 2080 Ti Gaming Z Trio, например, имеет 4352 универсальных процессора [2]. Поэтому составление расписания в многопроцессорных системах будем рассматривать на основе метода ветвей и границ.

1. Совместное использование ресурсов автоматизированной системы. Преимущества CPU+GPU систем

По результатам сравнительного исследования 18 продуктов безопасности (антивирус Касперского, Avast, AVG, McAfee, Norton Security, Total Security, Windows Defender и др.), проведенного в декабре 2019 года немецкой независимой лабораторией, специализирующейся на проверке и тестировании ведущих образцов антивирусного и защитного программного обеспечения, было выяснено: все средства защиты информации (в том числе предустановленный в Windows 10 защитник Windows Defender) в базовых настройках в среднем замедляют загрузку сайтов на 12–21 %, загрузку (запуск) приложений – на 13–30 %, установку стандартных приложений – на 18–51 %, а копирование и перенос файлов – на 5–25 % (см. таблицу). Для примера, защитник Windows Defender может замедлять установку приложений до 51 % [3]. Тестирование производилось на двух видах систем:

Среднепроизводительная система (standart) – на базе Intel i3-6100, 8 ГБ оперативной памяти, SSD-накопителем на 256 ГБ на операционной системе Windows 10 Professional 64bit;

Высокопроизводительная система (high-end) – на базе Intel i7-7770, 16 ГБ оперативной памяти, SSD-накопителем на 256 ГБ на операционной системе Windows 10 Professional 64bit.

Влияние средств защиты информации с базовыми настройками на скорость работы информационной системы, %

Вид замедления	Усредненные данные 18 продуктов безопасности		Антивирус Касперского	
	standart	high-end	standart	high-end
Замедление загрузки веб-сайтов	17–21	12–17	18–23	16–21
Замедление загрузки файлов	1–6	1–5	0–1	0–1
Замедление загрузки (запуска) стандартных приложений	15–30	13–14	10	7–8
Замедление установки популярных приложений	18–34	12–51	11–30	12–29
Замедление копирования файлов (локально и в сети)	5–13	5–25	1–2	1–3

В стандартах и инструкциях, разрабатываемых в интересах защиты информации в практически любых крупных организациях, имеется в том числе и ряд дополнительных требований к настройкам средств защиты информации. В результате программные средства защиты информации могут реализовывать усложненные методы проверки и анализа, что еще более негативно влияет на быстродействие конечной автоматизированной системы [4].

Для формализации описания вариантов задействования ресурсов автоматизированной системы введем следующие обозначения:

PU (Protection Unit) – подмножество необходимых ресурсов, которое необходимо задать для выполнения функции защиты информации автоматизированной системы за интервал времени Δt.

BU (Base Unit) – подмножество необходимых ресурсов, которое необходимо задать для решения задач в интересах автоматизированной системы за интервал времени Δt.

M (Memory) – ресурс памяти (совокупность физической и операционной памяти автоматизированной системы).

CPU (Central Processor Unit) – вычислительный ресурс центрального процессора.

GPU (Graphics Processing Unit) – вычислительный ресурс графического процессора.

На рис. 1 мы видим, что пересечение множеств $PU \cap BU$ при повышении нагрузки на систему в интервал времени Δt происходит на ресурсах CPU и M.

Следовательно, одновременное вычисление задач в интересах защиты информации и задач автоматизированной системы за время Δt при классическом подходе невозможно.

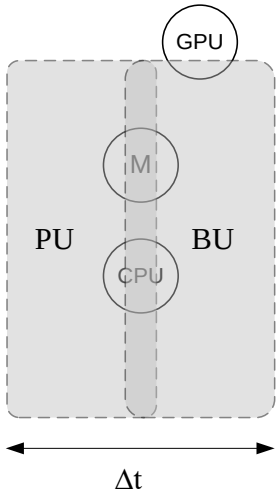

Рис. 1. Совместное использование ресурсов автоматизированной системы в промежуток времени Δt при классическом подходе

Перейдем к формализованному описанию. Подмножество $PU \subset (CPU \cup M)$ и $BU \subset (CPU \cup GPU \cup M)$. Для сохранения оперативности автоматизированных систем и сохранения должного уровня информационной безопасности необходимо, чтобы $PU \cap BU = 0$. Отсутствия пересечений данных множеств можно добиться двумя способами – путем увеличения времени Δt (что недопустимо по условию оперативности) или путем приращения неиспользуемых вычислительных ресурсов системы. За счет распараллеливания логико-математических задач защиты информации и подключения GPU возможно снятие части вычислительной нагрузки с CPU (рис. 2). Тогда $PU \subset (CPU \cup GPU \cup M)$.

Уменьшение использования ресурса M в интересах защиты информации также может происходить за счет приращения вычислительных возможностей путем подключения различных видов процессорных вычислителей. Например, при дефиците вычислительных мощностей и имеющейся свободной памяти повысить производительность логических вычислений можно представив логическую функцию таблицей и разместив ее в памяти.

Из проведенного анализа мы видим, что проблема влияния средств защиты информации на производительность автоматизированных систем является достаточно серьезной и требует системного решения. Увеличение вычислительных мощностей автоматизированной системы традиционно решалась наращиванием аппаратных ресурсов, однако это приводит к удорожанию системы и не всегда возможно на практике. Одним из способов решения указанной проблемы является разработка методик эффективного использования незадействованных ресурсов автоматизированных систем для решения задач защиты информации программными средствами. Рассмотрим их ниже.

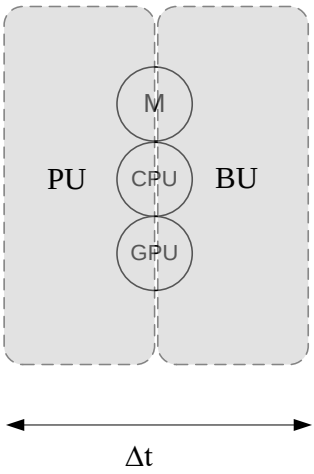

Рис. 2. Совместное использование ресурсов автоматизированной системы в промежуток времени Δt при использовании ресурсов GPU в интересах ЗИ

2. Составление оптимального расписания для гибридной многопроцессорной системы

Необходимо составить оптимальное по быстродействию расписание, т. е. назначить действия условного процесса системы защиты информации на процессоры таким образом, чтобы длина расписания была минимальной.

Формализуем данную проблему. Разобьем условный процесс системы защиты информации на множество задач $P\{p_1, p_2 ... p_n\}$. Т. к. рассматриваются параллельные вычисления, определим, что на конечный итог работы системы защиты информации не влияет последовательность выполнения задач на процессорах. Количество рассматриваемых процессоров примем равное m. m является собственным подмножеством множества процессоров CPU и множества процессоров GPU, т. е. $m \subset (CU \cup GU)$ и $m \neq (CU \cup GU)$. Длительность выполнения задачи p_i на процессоре j обозначим $t_{p_i j}$, где $i = \overline{1,n}$, $j = \overline{1,m}$. L_j –длина задач на процессоре j.

Таким образом, длина расписания выражается как $\max L_j$.

Минимальную длину расписания тогда можно представить следующим образом:

$$\min \max L_j,$$

т. е. ищется такое распределение задач, при котором наибольшая величина L_j принимает минимальное значение.

Метод ветвей и границ подразумевает решение подобных задач как разбиение множества всех допустимых решений на подмножества, для которых определяются верхние и нижние границы минимального времени выполнения задач. Если при сравнении двух подмножеств нижняя граница F_l первого подмножества оказывается больше, чем верхняя граница F_h второго подмножества, то первое подмножество в дальнейших расчетах игнорируется, т. к. оптимальная длина расписания там отсутствует.

Данные расчеты производятся для отдельно взятого процесса на гибридном CPU+GPU вычислителе. Следовательно, определенная часть задач этого процес-

са может быть уже назначена на конкретные процессоры существующими инструкциями. Обозначим часть назначенных задач – k. Вершина k может находиться в произвольном месте дерева решений между корнем (уровнем 0) и листьями (конечным уровнем p_n). Для k задач, соответственно, известно L_j. Тогда

$$F_l = (\sum_{j=1}^{m} L_j + \sum_{p_i=k+1}^{p_n} \min_{j=1,m} t_{p_i j}).$$

Т. е. за нижнюю оценку можно принять совокупность сумм длительностей назначенных задач и сумм длительностей оставшихся задач, выполненных на самом быстром для них процессоре.

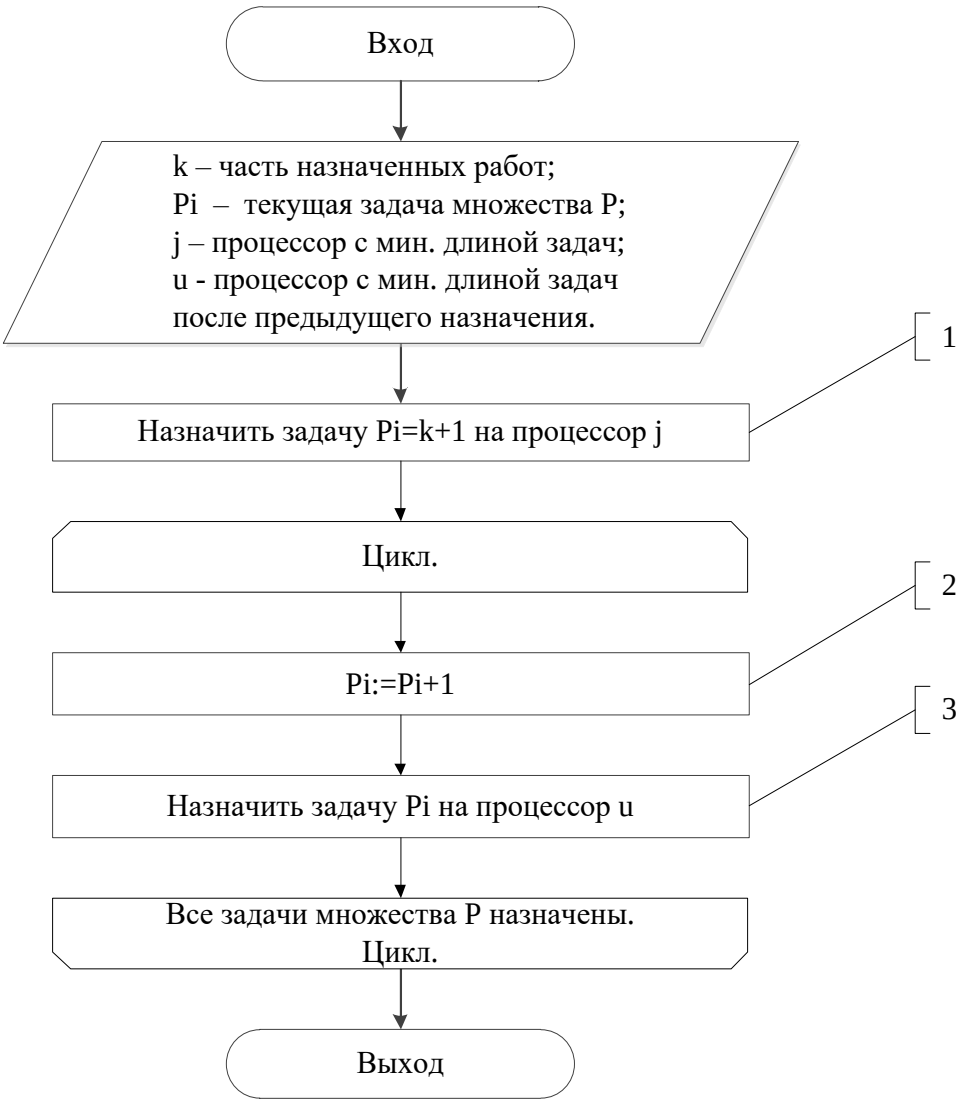

Рис. 3. Блок-схема работы жадного алгоритма

Для поиска F_h можно применить жадный алгоритм (рис. 3). Заполнение задачами процессоров по этому алгоритму будет выглядеть следующим образом:

1. Назначить первую из задач $p_i = \overline{k+1, p_n}$ на процессор j с минимальной длиной задач $min\, L_j$.

2. Перейти к следующей задаче $p_i := p_i + 1$. Если все задачи назначены, то далее следует выход из алгоритма.

3. Назначить задачу p_i на процессор u (с минимальной длиной задач после предыдущего назначения Q), $u = \overline{1, m}$, $Q_j = \max\left(L_1, \ldots, L_{j-1}, L_j + t_{p_i j}, L_{j+1}, \ldots\right)$, $\min\limits_{j=\overline{1,m}} Q_j = Q_u$, возврат в п. 2.

Когда все задачи назначены, находим максимальную длину расписания и принимаем ее как верхнюю оценку F_h для выбранной вершины уровня *k*:

$$F_h = \max\limits_{j=\overline{1,m}} L_j .$$

Оптимальное расписание F_o находится в диапазоне $F_l \le F_o \le F_h$.

3. Выбор стратегии ветвления метода ветвей и границ. Критерий эффективности, показатели эффективности

Существуют две основных стратегии построения дерева в методе ветвей и границ – одностороннее ветвление и фронтальное ветвление.

При выборе стратегии одностороннего ветвления на каждом уровне выбирается вершина с минимальной нижней оценкой, остальные вершины этого уровня игнорируются. Ветвление происходит только из вершины вышестоящего уровня с $\min F_l$ (рис. 4).

При выборе стратегии фронтального ветвления дерево решений строится полностью, т. е. на каждом уровне определяются все вершины (рис. 5). Отсев выполняется после вычисления верхних и нижних оценок и сравнения подмножеств (см. выше).

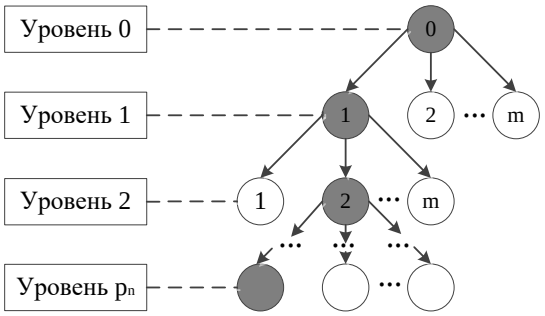

Рис. 4. Стратегия одностороннего ветвления

Как известно, более компактной форме представления, содержащей меньшее количество логических операций, будет соответствовать более производительная программная и менее громоздкая аппаратная реализация [5]. Соответственно, при выборе стратегии одностороннего ветвления времени для вычислений необходимо больше, памяти – меньше, при выборе фронтальной стратегии ветвления, наоборот, времени для вычислений необходимо меньше, памяти – больше.

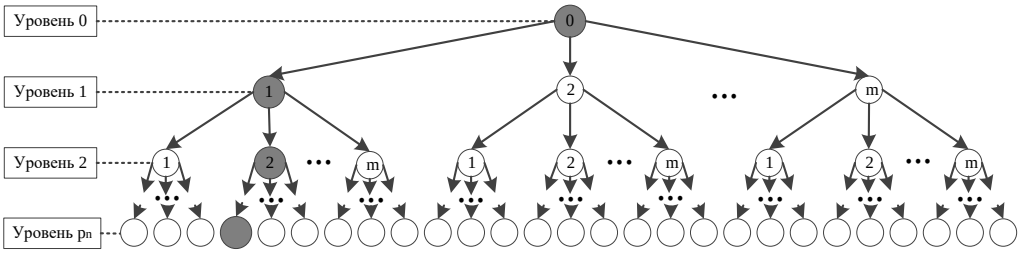

Рис. 5. Стратегия фронтального ветвления

Для выбора стратегии в конкретной автоматизированной системе необходимо произвести анализ физических и вычислительных возможностей системы, а также ряда требований к быстродействию и защите информации. В любом случае основным показателем эффективности работы рассматриваемых систем будет время t.

Критерием эффективности может быть минимальное время вычисления составления расписания: $\min t$ при имеющихся ограничениях на ресурсы системы $r \to lim$. Ресурсами системы являются объем оперативной и физической памяти, тактовые частоты процессоров, количество логических элементов и т. д.

Показателем объемной эффективности будет являться отношение количества бит, используемых для хранения всех возможных вершин дерева решений $N_{фв}^{кб}$, к количеству бит, используемых для хранения вершин при одностороннем ветвлении $N_{ов}^{кб}$:

$$v = \frac{N_{фв}^{кб}}{N_{ов}^{кб}}.$$

Показателем эффективности по сложности вычислений примем отношение количества выполняемых действий (тактов вычислителя), необходимых для вычисления оптимального расписания при фронтальном ветвлении $N_{фв}^{m}$, к количеству выполняемых действий (тактов вычислителя), необходимых для вычисления оптимального расписания при одностороннем ветвлении $N_{ов}^{m}$:

$$d = \frac{N_{фв}^{m}}{N_{ов}^{m}}.$$

Обобщенным показателем эффективности тогда будет:

$$c = v \cdot d.$$

Критерием эффективности будет $\max c$ при всех возможных вариантах расчета расписания.

4. Распределенное программирование в распределенных автоматизированных системах

Помимо параллельных вычислений существует и другой метод, позволяющий повысить эффективность использования имеющихся ресурсов. Он основан на кардинально ином подходе к самому программированию – так называемом распределенном программировании [6]. Он позволяет, в том числе, реализовать использование удаленных ресурсов иной вычислительной системы сети в интересах пользователя. Распределенное программирование в той или иной степени

включает в себя сетевое программирование и подразумевает общение посредством сетевого соединения между программами клиента и сервера [7]. Т. е. существует возможность использовать для параллельных вычислений, выполняемых на машине пользователя, вычислительные возможности, например, графического ускорителя соседней машины или сервера в распределенной автоматизированной системе.

Выводы

В современных реалиях, предписывающих распределенным автоматизированным системам максимальное быстродействие и минимальное время отклика посылаемого сигнала, остро встает проблема эффективности программной подсистемы защиты информации. Чем выше требования к настройке политики безопасности, тем сильнее средство защиты информации влияет на общее быстродействие. Перенаправление циклических вычислений системы защиты информации на пригодные к таким расчетам ядра графического процессора позволит существенно снизить нагрузку на информационную систему в целом. В зависимости от мощности имеющихся вычислительных ресурсов и объема памяти можно рассчитать оптимальную стратегию применения метода ветвей и границ.

Дополнительное преимущество, получаемое за счет перераспределения вычислений конкретной ЭВМ на ресурсы всей распределенной автоматизированной системы, дает распределенное программирование.

Грамотно скомбинированное использование рассмотренных в статье методов и алгоритмов открывает перед разработчиками огромные возможности для написания и реализации программ, позволяющих многократно увеличить вычислительные возможности распределенных автоматизированных систем путем перераспределения имеющихся ресурсов.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Решение задачи коммивояжера алгоритмом Литтла с визуализацией на плоскости. [Электронный ресурс]. – URL: https://habr.com/ru/post/332208 (дата обращения 28.05.2020).
2. Видеокарта MSI GeForce RTX 2080 Ti Gaming Z Trio получила более быструю память GDDR6 [Электронный ресурс]. – URL: https://www.hardwareluxx.ru/index.php/news/hardware/grafikkarten/49422-videokarta-msi-geforce–rtx-2080-ti-gaming-z-trio-poluchila-bolee-bystruyu-pamyat-gddr6.html (Дата обращения 03.06.2020).
3. Antivirus & Securuty software Test [Электронный ресурс]. – URL: http://spkurdyumov.ru/economy/informacionnye-texnologii-sostoyanie-i-perspektivy (Дата обращения 10.02.2020).
4. Сравнение сертифицированных средств защиты информации от несанкционированного доступа для серверов и рабочих станций (СЗИ от НСД) [Электронный ресурс]. – URL: https://www.anti-malware.ru/compare/information-protection-unauthorized-access-fstek-certified#part1 (Дата обращения 05.02.2020).
5. *Сизоненко А.Б.* Логико-математическое моделирование и синтез алгоритмов функционирования средств и систем защиты информации: монография. – Краснодар: КрУ МВД России, 2013. – 146 с.
6. *Камерон Х., Трейси Х.* Параллельное и распределенное программирование на C++. – М.: Вильямс, 2004. – 672 с.
7. Достоинства и недостатки параллельного программирования [Электронный ресурс]. – URL: http://web.snauka.ru/issues/2016/06/69538 (Дата обращения 06.06.2020).
8. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности ГОСТ Р ИСО/МЭК 27005-2010; введ. 01.12.2011. – М.: Стандартинформ, 2011. – 47 с.
9. *Kumar Jain A., Singh Y., Updhyay S.* Information systems security: A review. Ind Jour Math & Comp Sc. Jhs., 2013, № 2. P. 26–30.

10. *Боресков А.В., Садовничий В.А.* Параллельные вычисления на GPU. Архитектура и программная модель CUDA: учеб. пособие. – М.: Изд-во Московского университета, 2012. – 336 с.
11. Nvidia's Next Generation CUDA Compute Architecture: Kepler TM GK110 [Электронный ресурс]. – URL: http://www.nvidia.ru/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf
12. Доктрина информационной безопасности Российской Федерации, утв. Указом Президента Российской Федерации № 646 от 05.12.2016 // СПС «Консультант Плюс» [Электронный ресурс]. – URL: http://www.consultant.ru
13. *Сизоненко А.Б.* Модели и алгоритмы синтеза логиковычислительных подсистем защиты информации систем критического применения. – Воронеж, 2016. – 32 с.
14. Информационная технология. Методы и средства обеспечения безопасности. Менеджмент риска информационной безопасности. Критерии оценки безопасности информационных технологий. Часть 1 – Введение и общая модель. ГОСТ Р ИСО/МЭК 15408-1-2012; введ. 01.12.2013. – М.: Стандартинформ, 2013. – 75 с.
15. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 2 – Функциональные компоненты безопасности. ГОСТ Р ИСО/МЭК 15408-2-2013; введ. 01.09.2014. – М.: Стандартинформ, 2014. – 242 с.
16. Информационная технология. Методы и средства обеспечения безопасности. Критерии оценки безопасности информационных технологий. Часть 3 – Компоненты доверия к безопасности. ГОСТ Р ИСО/МЭК 15408-3-2013; введ. 01.09.2014. – М.: Стандартинформ, 2014. – 214 с.

Статья поступила в редакцию 25 июня 2020 г.

METHODOLOGY FOR DISTRIBUTING INFORMATION PROTECTION TASKS BETWEEN COMPUTER DEVICES OF AUTOMATED SYSTEMS BASED ON THE BRANCH AND BORDER METHOD

M. Gnutov, A. Sizonenko

Krasnodar higher military school
4, Krasina str. Krasnodar, 350063, Russian Federation

Abstract. *The influence of software protection on various information systems is analyzed. Using set theory, the use of computational resources for the joint solution of direct tasks and information protection tasks in an automated system is described. A hybrid implementation of computing in a CPU + GPU system is proposed. The relevance of using the branch and bound method to compile a minimum schedule of information security tasks in a hybrid multiprocessor system is considered. The features of processing data structures by various types of calculators are indicated. The computational strategies of the branch and bound method with the greatest possibility of acceleration with limited resources are analyzed. The efficiency criterion is selected, the performance indicators are considered when applying the frontal and one-sided branching strategies depending on the complexity of the calculations and the amount of occupied memory. The generalized indicator of efficiency is defined. The prospects of applying the considered method in distributed systems through distributed programming are emphasized.*

Keywords: *branch and bound method, GPU, CPU, schedule for a multiprocessor system, greedy, distributed programming*

REFERENCES

1. Resheniye zadachi kommivoyazhera algoritmom Littla s vizualizatsiyey na ploskosti [Solving the traveling salesman problem using the Little algorithm with visualization on the plane]. https://habr.com/en/post/332208 (accessed May 28, 2020).

Maxim S. Gnutov, Adjunct.
Alexander B. Sizonenko (Dr. Sci. (Techn.)), Associate Professor.

2. Videokarta MSI GeForce RTX 2080 Ti Gaming Z Trio poluchila boleye bystruyu pamyat' GDDR6 [The MSI GeForce RTX 2080 Ti Gaming Z Trio graphics card received faster GDDR6 memory]. https://www.hardwareluxx.ru/index.php/news/hardware/grafikkarten/49422 - videokarta - msi - geforce - rtx - 2080 - ti - gaming - z - trio - poluchila - bolee-bystruyu - pamyat - gddr6.html (accessed June 03, 2020).
3. Antivirus & Securuty software Test. http://spkurdyumov.ru/economy/informacionnye-texnologii-sostoyanie-i-perspektivy (accessed October 02, 2020).
4. Sravneniye sertifitsirovannykh sredstv zashchity informatsii ot nesanktsionirovannogo dostupa dlya serverov i rabochikh stantsiy (SZI ot NSD) [Comparison of certified means of protecting information from unauthorized access for servers and workstations (SZI from NSD)]. https://www.anti-malware.ru/compare/information-protection-unauthorized-access-fstek-certified#part1 (accessed February 05, 2020)
5. *Sizonenko A.B.* Logiko-matematicheskoye modelirovaniye i sintez algoritmov funktsionirovaniya sredstv i sistem zashchity informatsii: [Logical-mathematical modeling and synthesis of algorithms for the functioning of means and systems of information protection]. Krasnodar, Institute of the Ministry of the Interior of Russia, 2013.146 p. (In Russian).
6. *Cameron H., Tracy H.* Parallel'noye i raspredelennoye programmirovaniye na C++ [Parallel and distributed programming in C ++]. Williams, 2004. 672 p.
7. Dostoinstva i nedostatki parallel'nogo programmirovaniya [Advantages and disadvantages of parallel programming]. http://web.snauka.ru/issues/2016/06/69538 (accessed June 06, 2020).
8. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Menedzhment riska informatsionnoy bezopasnosti GOST R ISO/MEK 27005-2010 [Information technology. Security methods and tools. Information Security Risk Management GOST R ISO / IEC 27005-2010]. Standartinform, 2011. 47 p. (In Russian).
9. *Kumar Jain A., Singh Y., Updhyay S.* Information systems security: A review. *Ind Jour Math & Comp Sc. Jhs.*, 2013, no. 2. Pp. 26-30.
10. *Boreskov A.V., Sadovnichiy V.A.* Parallel'nyye vychisleniya na GPU. Arkhitektura i programmnaya model' CUDA : ucheb. Posobiye [GPU parallel computing. Architecture and software model CUDA] Publishing house of Moscow University, 2012. 336 p. (In Russian).
11. Nvidia's Next Generation CUDA Compute Architecture: KeplerTM GK110. http://www.nvidia.ru/content/PDF/kepler/NVIDIA-Kepler-GK110-Architecture-Whitepaper.pdf (accessed June 06, 2020).
12. Doktrina informatsionnoy bezopasnosti Rossiyskoy Federatsii, utv. Ukazom Prezidenta Rossiyskoy Federatsii [The doctrine of information security of the Russian Federation, approved. Decree of the President of the Russian Federation No. 646 dated 12.05.2016] http://www.consultant.ru (accessed June 10, 2020).
13. *Sizonenko A.B.* Modeli i algoritmy sinteza logikovychislitel'nykh podsistem zashchity informatsii sistem kriticheskogo primeneniya [Models and algorithms for the synthesis of logic-computing subsystems of information protection of critical application systems]. Voronezh, 2016. 32 p. (In Russian).
14. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Menedzhment riska informatsionnoy bezopasnosti. Kriterii otsenki bezopasnosti informatsionnykh tekhnologiy. Chast' 1 - Vvedeniye i obshchaya model' [Information technology. Security methods and tools. Information Security Risk Management. Criteria for assessing the security of information technology. Part 1 - Introduction and general model. GOST R ISO / IEC 15408-1-2012] Standartinform, 2013. 75 p. (In Russian).
15. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Kriterii otsenki bezopasnosti informatsionnykh tekhnologiy. Chast' 2 – Funktsional'nyye komponenty bezopasnosti [Information technology. Security methods and tools. Criteria for assessing the security of information technology. Part 2 – Functional safety components. GOST R ISO / IEC 15408-2-2013] Standartinform, 2014. 224 p. (In Russian).
16. Informatsionnaya tekhnologiya. Metody i sredstva obespecheniya bezopasnosti. Kriterii otsenki bezopasnosti informatsionnykh tekhnologiy. Chast' 3 – Komponenty doveriya k bezopasnosti [Information technology. Security methods and tools. Criteria for assessing the security of information technology. Part 3 – Security Trust Components. GOST R ISO / IEC 15408-3-2013] Standartinform, 2014. 214 p. (In Russian).