
ПАРАЛЛЕЛЬНОЕ И РАСПРЕДЕЛЕННОЕ
ПРОГРАММИРОВАНИЕ

УДК 004.42

**СПОСОБ УСКОРЕНИЯ ВЫЧИСЛЕНИЙ ЦЕЛЕВЫХ ПОКАЗАТЕЛЕЙ
ФУНКЦИОНИРОВАНИЯ КОСМИЧЕСКИХ СИСТЕМ АВТОМАТИЧЕСКОЙ
ИДЕНТИФИКАЦИИ И ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ
ПОДВИЖНЫХ ОБЪЕКТОВ
НА ОСНОВЕ ПРИМЕНЕНИЯ ТЕХНОЛОГИИ CUDA**

© 2019 г. Я. А. Скороходов*

*Военно-космическая академия имени А.Ф. Можайского
197198 Санкт-Петербург, ул. Ждановская, д. 13, Россия*

**E-mail: yaroslavskor@gmail.com*

Поступила в редакцию 12.12.2018 г.

После доработки 04.03.2019 г.

Принята к публикации 30.04.2019 г.

Сложность организации и условий применения в космосе систем автоматической идентификации и определения местоположения подвижных объектов, к которым следует отнести AIS (от англ. “Automatic Identification System” — автоматическая идентификационная система) и систему ADS-B (от англ. “Automatic dependent surveillance-broadcast” — автоматическое зависимое наблюдение — вещание), определили выбор моделей для математической формализации функционирования исследуемых систем в пользу имитационных. Имитационное моделирование орбитальных группировок космических аппаратов с возможностью приема, обработки и ретрансляции сигналов AIS и системы ADS-B с целью обоснования схемно-технических решений при их проектировании и планировании применения в дальнейшем, учитывая большое число источников излучений (например, для AIS свыше 500 тыс.), может занимать достаточно продолжительное время. Одним из методов решения указанной проблемы является распараллеливание вычислений с использованием технологии CUDA (от англ. “Compute Unified Device Architecture” — программно-аппаратная архитектура параллельных вычислений). Однако, в силу особенностей исполнения машинных инструкций на графическом процессоре NVidia качество программного обеспечения во многом зависит от эффективности распределения памяти видеокарты и алгоритмов выполнения программного кода. В статье предложен способ оценивания целевых показателей функционирования космических систем автоматической идентификации и определения местоположения подвижных объектов, основанный на массивно-параллельной организации вычислений с использованием графических вычислительных устройств, и позволяющий существенно сократить длительность моделирования, что особенно актуально для многоспутниковых орбитальных группировок. Выводы подтверждены практическими результатами модельно-кибернетических экспериментов, выполненных на различных программно-аппаратных платформах.

DOI: 10.1134/S0132347419060074

1. ВВЕДЕНИЕ

В настоящее время ряд государств, в том числе и РФ, проявляют интерес к созданию и наращиванию орбитальной группировки космических аппаратов с возможностью приема, обработки и ретрансляции сигналов автоматической идентификационной системы (АИС) и системы автоматического зависимого наблюдения — вещания (АЗН-В) [1–9]. Несмотря на наличие специализированных интернет-ресурсов, предоставляющих возможность следить за перемещением воздушных и морских судов [10–15] гарантии полноты, оперативности и достоверности получаемых

сведений фактически отсутствуют, в связи с чем использование перечисленных сервисов в критически важных отраслях народного хозяйства сопряжено с определенным риском и многие суверенные государства планируют иметь независимый информационный доступ к передаваемым в радиоэфир данным.

Моделирование сложных технических систем является одним из этапов в процессе их проектирования и создания, предназначенный для обоснования схемно-конструкторских решений, от которых в дальнейшем зависит эффективность целенаправленного применения создаваемых образцов техники.

Сложность формализованного описания работы космических систем наблюдения за движением воздушных и морских судов обусловила выбор моделей для оценивания целевых показателей их функционирования в пользу имитационных [16–24]. Главным недостатком разработанных имитационных моделей является большая длительность производства вычислений целевых показателей функционирования анализируемых систем.

Целью статьи является разработка алгоритмов моделирования космических систем контроля движения воздушных и морских судов на основе массивно-параллельной организации вычислений с использованием технологии CUDA и графических устройств.

Представляется краткая характеристика и анализ алгоритма моделирования функционирования космических систем автоматической идентификации и определения местоположения подвижных объектов с целью поиска возможных мест и путей распараллеливания вычислений.

2. АЛГОРИТМ МОДЕЛИРОВАНИЯ ФУНКЦИОНИРОВАНИЯ КОСМИЧЕСКИХ СИСТЕМ АВТОМАТИЧЕСКОЙ ИДЕНТИФИКАЦИИ И ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ

Алгоритм разработан с целью унификации частных алгоритмов моделирования космических систем контроля движения морских и воздушных судов [16, 17], поскольку принципы функционирования AIS и системы ADS-B являются одинаковыми, несмотря на ряд существенных отличий в их технической реализации. В то же время он является отправной точкой для последующих исследований, связанных с разработкой технологии оценивания показателей качества и синтеза космических систем указанного класса на основе имитационного моделирования, распределенной обработки и применения гетерогенных систем.

Исходными данными для алгоритма являются:

- время t_0 начала и интервал T моделирования;
- количество N_S моделируемых источников радиоизлучения (ИРИ);
- координаты c_i ($i = 1, \dots, N_S$) ИРИ в геодезической системе координат (СК), $c_i \in C = \{(b, l, h): -90 \leq b \leq 90, -180 \leq l \leq 180, h \in \mathfrak{R}\}$ (в случае статической модели распределения ИРИ) или множество трасс $\mathcal{T}_i = \{t_{ij} : i = 1, \dots, N_S, j = 1, \dots, n_i\}$, $n_i = \text{card}(\mathcal{T}_i)$ движения ИРИ, где трасса t_{ij} — множество координат $\{c(t_k)\}$, соотнесенных с моментом времени $t_k \in [t_0, t_0 + T]$ (в случае динамической модели);
- распределение ИРИ по их динамическим характеристикам (скорости и курсу движения) — только для морских судов [25];

- распределение различных типов сигналов (MODE A/C, MODE S, MODE S ADS-B) по частоте (количество раз в секунду) передачи — только для воздушных судов [26];

- распределение ИРИ по мощности излучений, мощность сигналов радиопередатчиков для воздушных судов определяется типом судна (удовлетворяющих стандарту ICAO или нет, с ADS-B или без) и выбирается из множества $E = \{21 \text{ дБВт}, 24 \text{ дБВт}, 27 \text{ дБВт}, 29 \text{ дБВт}\}$ [26], для морских судов — классом судна (грузовое, пассажирское и др.) и массово-габаритными характеристиками в соответствии с рекомендациями МСЭ и выбирается из множества $E = \{2.5 \text{ Вт}, 12.5 \text{ Вт}\}$ [27];

- частота излучений, для АЗН-В множество частот состоит из одного элемента $F = \{1090 \text{ МГц}\}$, для АИС — $F = \{161.975, 162.025 \text{ МГц}\}$;

- количество N_R КА, входящих в состав орбитальной группировки (ОГ);

- начальные значения оскулирующих элементов орбит $c_j(t_0)$, $c_j \in C = \{(\Omega, i, r_a, r_n, \omega, \theta(t_0)) : 0 < \Omega < 360, 0 < i < 180, r_a, r_n \in \mathfrak{R}, 0 < \omega < 360, 0 < \theta < 360\}$, $j = 1, \dots, N_R$;

- максимальный коэффициент G_0 направленного действия (КНД), ширина $\theta_{\text{зdB}}$ диаграммы направленности приемной антенны, направление максимального усиления относительно оси антенны;

- чувствительность q радиоприемного устройства (РПУ).

Алгоритм состоит из нескольких этапов. На предварительном этапе формируется множество ИРИ $S \subset I \times \mathcal{T} \times E \times F$, где $I = \{1, \dots, N_S\}$ — индексное множество, т.е. каждому из ИРИ s_i ($i = 1, \dots, N_S$) в соответствии с заданными распределениями назначаются координаты c_i (трассы $\mathcal{T}_i$), мощность e_i и частота f_i излучений.

В процессе моделирования для каждого момента времени $t_k \in [t_0, t_0 + T]$ выполняется следующая последовательность действий:

1) рассчитывается текущее положение КА $r_j \in R$, $j = 1, \dots, N_R$, входящих в состав ОГ, по заданным начальным значениям оскулирующих элементов орбиты и координаты активных ИРИ s_i (в случае динамической модели распределения ИРИ), активным называется ИРИ, для которого индикатор $a_i(t_k) = 1$, т.е. его радиопередатчик включен. В любой момент времени $t_k \in [t_0, t_0 + T]$ выполняется условие $\sum_{i=0}^{N_S} a_i(t_k) < N_S$;

1*) (пункт выполняется только для источников сигналов АИС) проверяется условие $t_k \% t_u(i) = 0$, $i = 1, \dots, N_S$, где $t_u(i) \in T_u = \{(3, \dots, 7) * 60\}$ — время резервирования временных слотов [25]. Для ИРИ $s_i | t_k \% t_u(i) = 0$ выполняется алгоритм планирования использования тайм-слотов — SOTDMA,

значение t_u интервала обновления для каждого объекта генерируется датчиком случайных чисел, результатом выполнения алгоритма является множество $T_i = \{t_1, \dots, t_{n_i}\}$, $n_i = \text{card}(T_i)$ индексов слотов, принадлежащее объекту s_i ;

2) для КА $r_j \in R, j = 1, \dots, N_R$ определяется пространственно-энергетическая доступность $v_r(s_i, r_j)$ источников излучений s_i ($i = 1, \dots, N_S$) исходя из условий превышения минимально допустимого угла места $\beta(s_i, r_j)$ КА r_j в топоцентрической системе координат, связанной с местоположением анализируемого источника излучений s_i , и требуемого потока мощности сигналов на входе приемника r_j с учетом энергетических потерь в линии связи, диаграмм направленности типовой антенны ИРИ и спутниковой антенны, мощности излучений и чувствительности РПУ;

3) для всех объектов наблюдения $s_i, v_r(s_i, r_j) = 1$, находящихся в зоне покрытия КА r_j , выполняется алгоритм генерирования сигналов и формируется список $M_j = \{M_{ji} : v_r(s_i, r_j) = 1\}$ сообщений с привязкой к источнику излучения s_i и его характеристикам (мощности передатчика), алгоритм генерирования имеет особенности в зависимости от типа ИРИ (источник сигналов АИС или авиационных систем связи);

3.а) для источников сигналов АИС – проверяется условие $T_i \cap \{\lceil (t_k \% 60) / 0.0267 \rceil \div \lceil ((t_k + 1) \% 60) / 0.0267 \rceil\}$ задействования слотов для передачи сообщений в анализируемом секундном интервале t_k , где $\lceil x \rceil$ – операция округления в большую сторону значения x . Если $t_i \in T_i$ и условие выполняется, сообщение заносится в список M_{ji} , момент начала его передачи вычисляется как произведение $t_i * 0.0267$, $t_i \in T_i \cap \{\lceil (t_k \% 60) / 0.0267 \rceil \div \lceil ((t_k + 1) \% 60) / 0.0267 \rceil\}$;

3.б) для источников сигналов авиационных систем связи – количество сигналов типов

MODE A/C, MODE S генерируется случайным образом, используя статистическую модель их распределения, количество сигналов MODE S ADS-B фиксировано, время излучения сигналов выбирается случайно в интервале $[0, 1)$ таким образом, чтобы сигналы, переданные одним объектом, не накладывались друг на друга и были равномерно распределены в секундном интервале t_k ;

4) для каждого из сообщений в списке M_j проверяется условие наличия коллизии с другими сообщениями m_{ji}, m_{jk} , где $m_{ji}, m_{jk} \in M_j, i, k = 1, \dots, |M_j|, i \neq k$, учитывая задержки распространения сигналов от различных источников, потери мощности и т.д., вычисляется вероятность безошибочного приема сообщения (вид модуляции и длительность сигнала зависят от моделируемой системы) для отношения сигнал/шум, где энергия шума при наличии интерференции сигналов соответствует максимальной энергии одного из “мешающих” сообщений. Решение о безошибочном приеме сообщения принимается в случае попадания значения равномерно распределенной величины от 0 до 1 в интервал, длина которого зависит от вычисленной на предыдущем шаге вероятности правильного декодирования;

5) в случае правильного детектирования сообщения источник s_i , его передавший, считается идентифицированным, кроме того, если сообщение содержит координаты (сигналы AIS или ADS-B) – локализованным;

6) формируется информация как интегрированная за интервал моделирования $t \in [t_0, t_k]$, так и текущая, относящаяся к секунде t_k .

Результатом работы алгоритма являются целевые показатели Y функционирования систем контроля движения воздушных и морских судов [16, 17]. Псевдокод алгоритма с последовательной организацией вычислений представлен в листинге 1.

```
function run sequential(R, S, Ns, C, t0, T)
```

Input:

R – объект типа Satellite, включающий алгоритмы орбитального движения и характеристики приемного тракта

S – объект типа SenderList, включающий трассы движения источников излучений, алгоритмы распределения сигналов и характеристики передающих устройств

C – объект типа Channel, содержащий характеристики канала передачи информации и алгоритмы вычисления ослабления мощности сигнала

t0 – время начала моделирования

T – длительность моделирования

Ns – количество источников сигналов

Output:

SD – объект типа SimulationData, содержащий целевые показатели функционирования КС

```
for all t in [t0, t0 + T] {
```

```

if (type(S) = AisSenderList && (t - t0) % 60 == 0) {
    S.MsgList ← SOTDMA // Алгоритм планирования слотов
}
R.coord ← координаты КА в соответствии с заданной моделью
орбитального движения в момент времени t
S.coord ← координаты источников излучений в момент времени t и статус их
активности
for all sender in S { // шаг 2 и 3 алгоритма
    if (!sender.enable) { // Проверка активности sender
        continue
    }
    b ← угол места КА в топоцентрической СК, связанной с sender
    bm ← минимальный угол места КА, при котором обеспечивается прямая его
    видимость для источника с высотой h
    if (b > bm) {
        SNR ← отношение сигнал/шум исходя из заданной модели С среды рас-
        пространения сигналов, мощности передатчика и чувствительности РПУ
        if (SNR > 0) {
            R.msgList += сгенерированный список сигналов в соответствии с за-
            данной моделью и SNR
        }
    }
}
for (i = 0, i < count(R.MsgList), i++) { // шаг 4 алгоритма
    sender[R.MsgList[i].sndIdx].isMsgSent = true
    for (j = i + 1, j < count(R.MsgList), j++) {
        hasCollision ← факт коллизии R.MsgList[i] и R.MsgList[j]
        if (!hasCollision) {
            continue
        }
        SIR = R.MsgList[i].SNR - R.MsgList[j].SNR
        k ← R.MsgList[i].SNR > R.MsgList[j].SNR ? i : j
        R.MsgList[k].SNR = abs(SIR)
        R.MsgList[k == i ? j : i].collisionStatus = true
    }
    if (R.MsgList[i].collisionStatus == false) {
        P ← вероятность безошибочного приема сообщения, состоящего из n бит (за-
        висит от типа) для рассчитанного значения SNR
        h ← значение СВ, распределенной равномерно от 0 до 1
        if (h > P) {
            sender[MsgList[k].sndIdx].isDetected = true
        }
    }
}
SD ← count(s ∈ S | isMsgSent = true), count(s ∈ S | isDet = true),...
out ← координаты КА и SD
}

```

Листинг 1. Алгоритм оценивания целевых показателей КС с последовательной организацией вычислений

Listing 1. Algorithm for estimation targets with sequential calculations organization

В случае большого числа моделируемых объектов, составляющих радиоэлектронную обстановку, этапы оценивания энергетической доступности ИРИ и определения безошибочно принятых сообщений могут занимать достаточно продолжительное время, несмотря на распараллеливание операций для каждого КА в отдельности при наличии соответствующих процессорных ресурсов (количества ядер), а учитывая, что в перспективных КС наблюдения за движением воздушных и морских судов предполагается использование нескольких десятков и даже сотен КА, требуется разработать подход, позволяющий существенно сократить время вычислений.

Решением изложенной выше задачи может быть использование графических карт для проведения вычислений общего назначения.

3. ОРГАНИЗАЦИЯ МАССИВНО-ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ NVIDIA ПРИ МОДЕЛИРОВАНИИ ФУНКЦИОНИРОВАНИЯ КОСМИЧЕСКИХ СИСТЕМ АВТОМАТИЧЕСКОЙ ИДЕНТИФИКАЦИИ И ОПРЕДЕЛЕНИЯ МЕСТОПОЛОЖЕНИЯ ПОДВИЖНЫХ ОБЪЕКТОВ

Учитывая программную модель и особенности организации вычислений на ГП [28–30], можно сформулировать направления для распараллеливания алгоритма и сокращения общего времени моделирования:

- На шаге № 2 анализ ослабления переданных сигналов на входе РПУ и оценку энергетической доступности ИРИ, т.е. проверку условий пространственной доступности для КА, входящих в состав ОГ, и превышения мощности сигнала заданного порогового значения производить для каждого объекта в отдельном потоке. Для этого требуется создать количество нитей равное числу N_S моделируемых объектов и проинициализировать их значениями элементов массива с индексами соответствующим абсолютному номеру нити.

- На шаге № 4 наличие наложений сообщений m_{ji} и m_{jk} , где $m_{ji}, m_{jk} \in M_j$, $i, k = 1, \dots, |M_j|$, $j = 1, \dots, N_R$, проверять в отдельных потоках. С этой целью необходимо создать пул нитей размерностью не меньшей числу $C_{\text{card}(M_j)}^2$ сочетаний из $\text{card}(M_j)$ элементов по 2 без повторений. Таким образом, перестает существовать двойной цикл с попарной проверкой коллизий сигналов, однако такой подход не является самым оптимальным, поскольку не исключает возможность одновременного обращения к одним и тем же участкам глобальной памяти из разных нитей.

Разработан алгоритм моделирования функционирования космической системы с использова-

нием гетерогенных вычислений. Исходные данные для проведения моделирования не отличаются от представленных выше. Схема алгоритма представлена на рис. 1.

Шаг 1. Задать исходные данные (в соответствии с п. 2 настоящей статьи).

Шаг 2. Выделить память на ГП и произвести копирование информации из памяти ЦП в память ГП: массивов координат c_i ($i = 1, \dots, N_S$), $c_i \in C$, $C = \{(b, l, h) : -90 \leq b \leq 90, -180 \leq l \leq 180, h \in \mathbb{R}\}$ в геодезической СК источников сигналов; мощностей e_i ($i = 1, \dots, N_S$) их передающих устройств; и значений диаграммы направленности a_z ($z = 1, \dots, k$).

Шаг 3. Начать цикл по множеству $t_k \in [t_0, t_0 + T]$.

Шаг 4. Произвести моделирование орбитального движения КА [31]

$$c_j(t_k) = f[c_j(t_0), t_k - t_0], \quad j = 1, \dots, N_R.$$

Шаг 5. Выполнить на ГП ядро, в котором производится анализ энергетической доступности источников сообщений, и скопировать результаты вычислений (массив значений индикаторов v_i^j энергетической доступности и потерь L_i^j передачи для каждого источника s_i , $i = 1, \dots, N_S$) из памяти ГП в память ЦП. Координаты $c_j = (x_j^{\text{НГР}}, y_j^{\text{НГР}}, z_j^{\text{НГР}})$ КА в НГРСК, полученные на предыдущем шаге, передаются в функцию ядра в качестве параметров.

Шаг 6. Рассчитанные значения индикаторов v_i^j и потерь L_i^j скопировать из памяти ГП в память ЦП.

Шаг 7. Сформировать список $M_j = \{M_{ji} : v_i^j = 1\}$ сообщений, переданных судами s_i ($i = 1, \dots, N_S$), которые находятся в зоне покрытия КА r_j , ($j = 1, \dots, N_R$) с координатами $c_j = (x_j^{\text{НГР}}, y_j^{\text{НГР}}, z_j^{\text{НГР}})$. Порядок выполнения пункта определяется шагами 3.а и 3.б алгоритма из п. 2 настоящей статьи.

Шаг 8. Выполнить на ГП ядро, осуществляющее проверку наличия интерференций сигналов, оценивания безошибочно принятых сообщений, и определения объектов $s_i | \rho_i = 1$, передавших сигналы, идентифицированных $s_i | \kappa_i = 1$ и локализованных $s_i | \mu_i = 1$ объектов, где ρ_i, κ_i, μ_i — индикаторы, обозначающие факт передачи сообщения источником s_i , идентификации и определения местоположения источника s_i соответственно.

Шаг 9. Выполнить на ГП ядро расчета целевых показателей, относящихся к интервалу $[t_0, t_k]$, так и к моменту времени t_k [16, 17]. Например, количество $\rho(S)$ ИРИ, передавших хотя бы одно сообщение, количество $\kappa(S)$ идентифицированных и локализованных $\mu(S)$ ИРИ за интервал моделирования $[t_0, t_k]$ определяются выражениями:

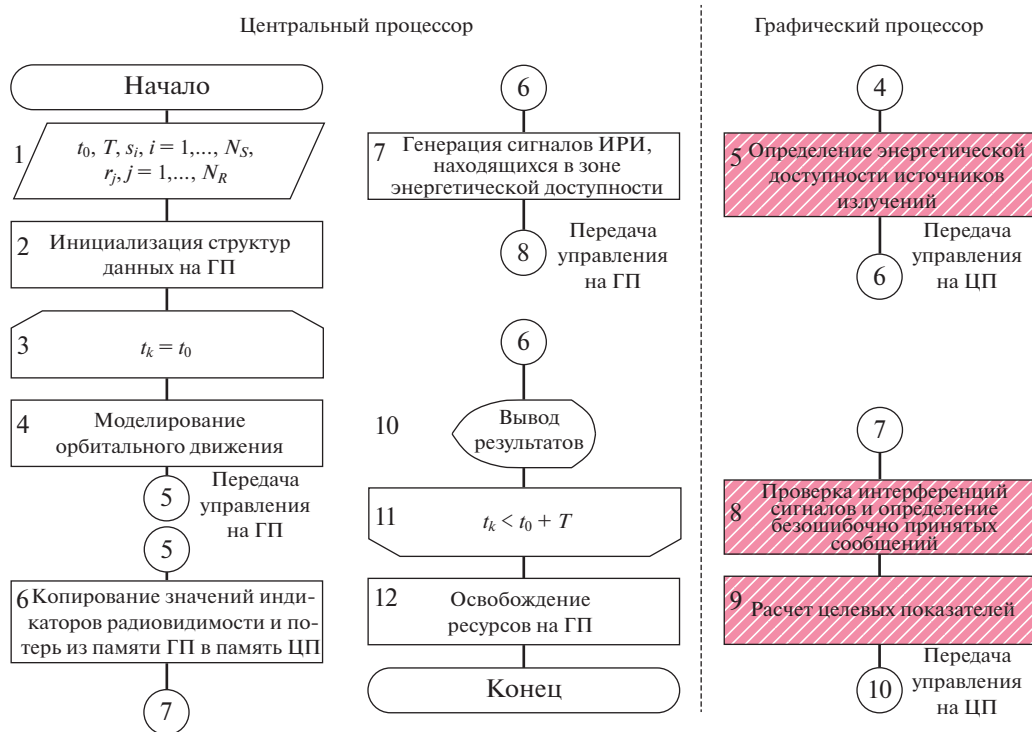


Рис. 1. Схема алгоритма оценивания целевых показателей функционирования КС с использованием гетерогенных вычислений.

$$\rho(S) = \sum_{i:s_i \in S} \rho_i; \quad \kappa(S) = \sum_{i:s_i \in S} \kappa_i; \quad \mu(S) = \sum_{i:s_i \in S} \mu_i.$$

Приведенные показатели и им подобные могут быть вычислены с помощью параллельного суммирования элементов соответствующих массивов.

Шаг 10. Передать в поток вывода значения текущего времени t_k , количества $\rho(S)$ воздушных судов, передавших хотя бы одно сообщение, количества $\kappa(S)$ идентифицированных судов, количества $\mu(S)$ обнаруженных судов, количества полученных КА сообщений, количества детектированных сообщений, количества коллизий.

Шаг 11. Проверить условие $t_k < t_0 + T$ окончания заданного промежутка времени, если условие выполняется, перейти на шаг 4, иначе на шаг 9.

Шаг 12. Освободить память на ГП.

Учитывая ограничения программной модели графического процессора nVidia, сложность выполненной работы заключалась в том, чтобы, во-

первых, преобразовать вычисления на этапах, предполагающих исполнение на ГП, к линейному виду без циклов и условных переходов, минимизировать количество и объем передаваемых данных между вычислительными устройствами, во-вторых, организовать эффективные алгебраические и матричные вычисления на графическом процессоре, а также разработать способ хранения исходных и промежуточных данных, а также результатов в различных типах памяти ГП в зависимости от их объема и необходимой скорости доступа. Программный код CUDA C разделен на ряд сегментов в виде функций, псевдокод которых представлен в листинге 2, что обусловлено различными параметрами (размерность грида, количество блоков, количество нитей в блоке), с которыми должно запускаться ядро для разных функций, а также передачей управления на ЦП между вызовами. Названия функций соответствуют названию блоков на рисунке 1.

```
function определение энергетической доступности источников (R, S, Ns,
C, dev_snr, dev_vis)
```

Input:

R – объект типа Satellite, размещенный в памяти ГП
 S – объект типа SenderList, размещенный в памяти ГП
 C – объект типа Channel

N_s – количество источников сигналов

Output:

dev_snr – указатель на массив в памяти ГП, содержащий значения SNR для каждого источника

dev_vis – указатель на массив индикаторов в памяти ГП, содержащий индикаторы радиовидимости

$idx = blockIdx.x * blockDim.x + threadIdx.x$

if ($idx < N_s$) {

$dev_snr[idx] \leftarrow$ отношение сигнал/шум для $sender[idx] \in S$ и R с учетом заданной модели среды C , диаграмм направленности и т.д.

$dev_vis[idx] = dev_snr[idx] > 0$

}

function проверка коллизий сообщений($nMsg$, M , dev_col)

Input:

$nMsg$ – количество сообщений

M – объект типа `MessageList`, размещенный в памяти ГП

Output:

dev_col – указатель на массив в памяти ГП, содержащий индикаторы коллизии сообщений

$i = blockIdx.x * blockDim.x + threadIdx.x$

$j = blockIdx.y * blockDim.y + threadIdx.y$

if ($i < nMsg \ \&\& \ i > j$) {

$hasCollision \leftarrow$ факт коллизии сообщений $M[i]$ и $M[j]$

if ($\neg hasCollision$) {

`continue`

}

$dev_sigCol[i] = \text{true};$

$dev_sigCol[j] = \text{true};$

}

function определение безошибочно принятых сообщений($nMsg$, M , $dev_msgSent$, dev_det , dev_locDet)

Input:

$nMsg$ – количество сообщений

M – объект типа `MessageList`, размещенный в памяти ГП

Output:

$dev_msgSent$ – указатель на массив в памяти ГП индикаторов источников, передавших хотя бы одно сообщение

dev_det – указатель на массив в памяти ГП индикаторов источников, идентифицированных за время наблюдения

dev_locDet – указатель на массив в памяти ГП индикаторов источников, локализованных за время наблюдения

$idx = blockIdx.x * blockDim.x + threadIdx.x$

if ($idx < nMsg$) {

$sndIdx = M[idx].sndIdx$

$dev_msgSent[sndIdx] = \text{true}$

if ($\neg dev_sigCol[idx]$) {

$P \leftarrow$ вероятность безошибочного приема сообщения, состоящего из n бит (зависит от типа) для рассчитанного значения dev_snr

$h \leftarrow$ значение СВ, распределенной равномерно от 0 до 1

$dev_det[sndIdx] = h > P$

$dev_locDet[sndIdx] = M[idx].loc \ \&\& \ dev_det[sndIdx]$

```

// Производится вычисление аналогичных характеристик за период
// обращения КА
}
}

```

Листинг 2. Фрагменты алгоритма оценивания целевых показателей КС с параллельной организацией вычислений

Listing 2. Fragment of algorithm for estimation targets with parallel calculations organization

Поскольку в расчетах используются массивы данных большой размерности, а также многочисленные операции вращения и перемещения в трехмерном пространстве, эффективность разработанного алгоритма зависит от распределения памяти ГП и скорости выполнения матричных операций.

4. РАСПРЕДЕЛЕНИЕ ПАМЯТИ И МОДЕЛЬ ВЫПОЛНЕНИЯ МАТРИЧНЫХ ОПЕРАЦИЙ НА ГРАФИЧЕСКОМ ПРОЦЕССОРЕ

Схема, поясняющая размещение в памяти ГП исходных данных и других служебных структур данных, изображена на рис. 2.

В глобальной памяти ГП размещаются массивы значений координат $c_i = \{(b, l, h)\}$ в геодезической СК, мощностей e_i транспондеров источников излучений $s_i \in S, i = 1, \dots, N_S$, а также создаваемые на каждом шаге выполнения программы массивы значений отношения сигнал/шум и булевых переменных v_i^j , определяющих радиовидимость источников излучений КА r_j .

Выбор размещения перечисленных выше массивов в пользу глобальной памяти определяется их достаточно большой размерностью, однако, глобальная память, находящаяся в DRAM ГП, характеризуется высокой латентностью (низкой скоростью доступа).

Диаграмма направленности передающей антенны, заданная в табличном виде, размещается в константной памяти. Поскольку константная память, которая физически располагается также как и глобальная в DRAM ГП, является кешируемой, время чтения из нее значительно меньше, что делает ее подходящей для хранения неизменяемых данных небольшого размера.

Буфер, состоящий из объектов класса матрицы, размещается в разделяемой памяти. Общий объем буфера рассчитывается исходя из суммы требуемого количества BUFSIZE экземпляров класса матрицы на один поток и указателя idx на текущий элемент массива, умноженной на количество создаваемых потоков. Объем требуемой shared памяти рассчитывается с использованием выражения

$$\text{size} = \text{THREADS} * (\text{BUFSIZE} * \text{sizeof}(\text{cuMatrix}) + \text{sizeof}(\text{int})),$$

где sizeof – оператор вычисления размера структуры данных в байтах.

Например, при $\text{BUFSIZE} = 10$, $\text{size} = 28416$ байт.

С учетом особенностей задачи, поскольку при моделировании (а именно для определения взаимного расположения источников и приемников сообщений, вращения и преобразования систем координат) требуется производить большое количество операций над матрицами, целесообразно спроектировать отдельный класс матриц, включающий методы, исполняющиеся на устройстве (device), а также механизм, позволяющий распределять память с результатами промежуточных вычислений из программно-управляемого кеша (shared memory) ГП по мере необходимости.

На рис. 3 схематически изображена последовательность действий при поступлении запроса на выделение разделяемой памяти для выполнения методов класса матрицы.

При поступлении требования на распределение памяти для экземпляра класса матрицы, используя вызов функции ядра (соответствующий вызов обозначен на рисунке как *get_matrix_temp()*) выполняется следующая последовательность действий. Определяется относительный индекс нити внутри блока (шаг 1), происходит чтение счетчика с номером k , соответствующим индексу нити в блоке (шаг 2).

Поскольку доступ к разделяемой памяти имеют только нити внутри данного блока, коллизий с нитями из других блоков с аналогичными индексами k не произойдет. В свою очередь, общее количество счетчиков равно количеству нитей в блоке, т.е. каждая нить имеет свой собственный счетчик. Имея значения счетчика k и размер массива BUFSIZE, ограничивающий количество выделяемых каждой из нитей в блоке матричных структур, определяется абсолютный адрес разделяемой памяти, который передается в вызывающий процедуру код (шаг 3). Значение счетчика увеличивается на единицу (шаг 4). В случае если значение в результате инкрементирования становится равным максимальному значению BUFSIZE, то счетчик обнуляется.

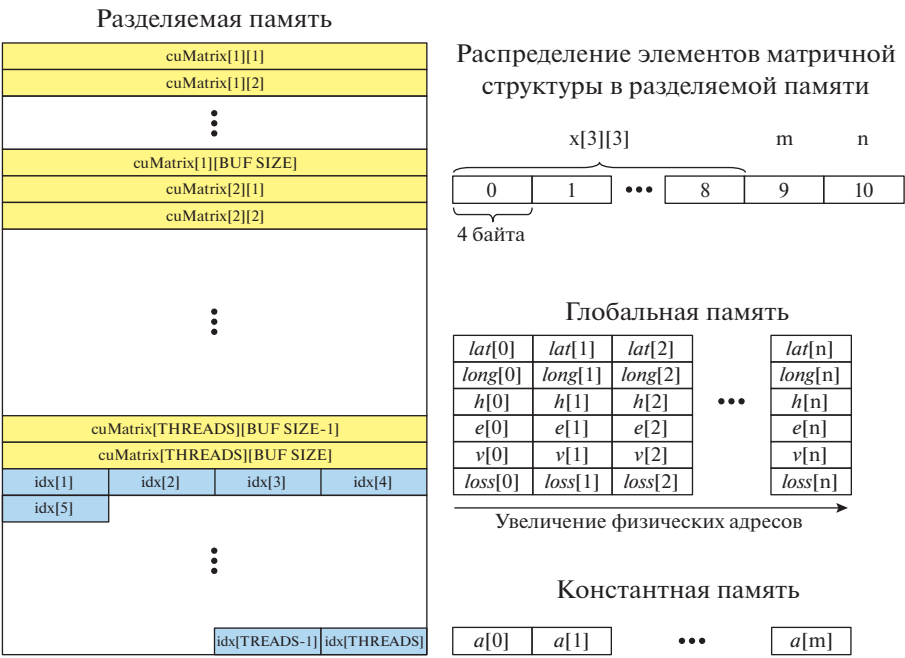


Рис. 2. Размещение структур данных в памяти ГП.

Таким образом, повышается скорость вычислений и исключаются задержки времени с постоянным распределением и освобождением памяти, однако значение `BUFSIZE` должно выбираться исходя из максимального количества требуемых экземпляров

класса матриц с результатами промежуточных вычислений, иначе результаты работы программы будут некорректными (количество экземпляров класса `cuMatrix`, содержащих промежуточные результаты, в одном выражении не превышает 10).

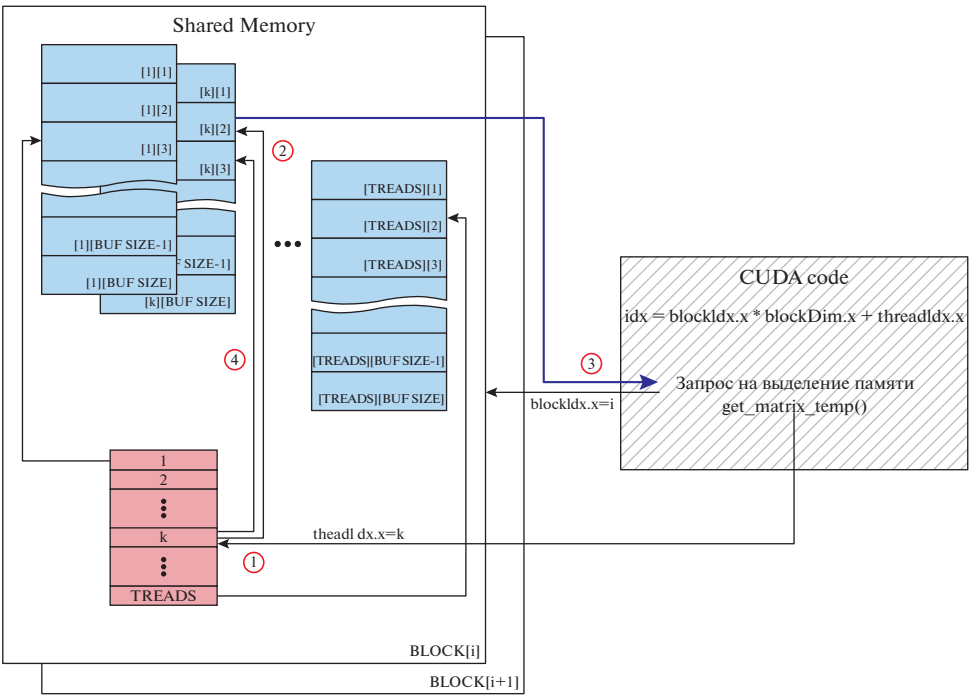


Рис. 3. Схема организации матричных вычислений на графическом процессоре.

Таблица 1. Характеристики ПЭВМ

№ п/п	Характеристика	Значение
1	ЦП	Intel(R) Core (TM) i3-2120 CPU @ 3.30GHz
2	ОЗУ	4.00 Гб DDR3

Таблица 2. Характеристики модуля Jetson TX2

№ п/п	Характеристика	Значение
1	ГП	NVIDIA Pascal, 256 CUDA cores
2	ЦП	HMP Dual Denver 2/2 MB L2 + Quad ARM® A57/2 MB L2
3	ОЗУ	8 GB 128 bit LPDDR4 59.7 GB/s

5. СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВРЕМЕНИ ВЫЧИСЛЕНИЯ ЦЕЛЕВЫХ ПОКАЗАТЕЛЕЙ ФУНКЦИОНИРОВАНИЯ КОСМИЧЕСКИХ СИСТЕМ НА РАЗЛИЧНЫХ АППАРАТНО-ПРОГРАММНЫХ ПЛАТФОРМАХ

С целью сравнительного анализа времени моделирования при различных вариантах организации вычислений (последовательном и параллельном), а также оценивания достигаемого эффекта производился модельно-кибернетический эксперимент, включающий два этапа. На первом этапе эксперимента определялся прирост производительности отдельно на каждом из оптимизируемых шагов (т.е. подлежащих распараллеливанию шагов). Для этого, во-первых, варьировалось количество моделируемых объектов в зоне обслуживания КА и оценивалось время анализа энергетической доступности источников излучений без последующей проверки интерференций переданных сигналов, во-вторых, измерялась длительность проверки наложений сигналов системы ADS-B и определения безошибочно принятых сообщений при различном количестве ИРИ в зоне покрытия КА. При этом предполагались допущения: модель КА — статическая (без учета орбитального движения), распределение объектов наблюдения — равномерное, время моделирования — 1000 с.

Параметры запуска функций (ядер), исполняющихся на графическом вычислительном устройстве:

Этап 1.1 — одномерная сетка с количеством нитей в блоке равным 64 и количеством блоков в сетке равным $(N + 64)/64$, N — число моделируемых объектов.

Этап 1.2 — двумерная сетка с количеством нитей в блоке равным 32×32 и количеством блоков в сетке равным $[(M + 31)/32] \times [(M + 31)/32]$, где M — количество анализируемых сигналов. Сетку можно представить матрицей, в которой элементы — булевы переменные, принимающие значение равное 1, если между сигналами с условными номерами i и j есть наложение, иначе 0. Фактически задействуются только нити с индексами, удовлетворяющими условию $i > j$, где i и j определяются исходя из выражений

$$i = blockIdx.x * blockDim.x + threadIdx.x;$$

$$j = blockIdx.y * blockDim.y + threadIdx.y,$$

где $blockDim$, $blockIdx$, $threadIdx$ — встроенные переменные языка программирования CUDA C, определяющие размерность сетки и порядковые номера блоков в сетке и нитей в блоке соответственно.

На втором этапе моделировался процесс функционирования КА в целом.

В табл. 1 приведены характеристики ПЭВМ, на которой выполнялся программный код с последовательной организацией вычислений («классической» схеме).

Исполнение гетерогенного программного кода производилось на модуле Jetson TX2, оснащенном графическим процессором с архитектурой NVIDIA Pascal. Основные характеристики модуля Jetson TX2 представлены в табл. 2.

Время работы программного комплекса при последовательной организации вычислений на ЦП измерялось в операционных системах Windows 7 SP1 и Ubuntu Linux 16.04 LTE. Моделирующий комплекс с параллельной организацией вычислений апробировался на ОС Ubuntu Linux 16.04 LTE.

Генерация программного кода, исполняемого на ЦП, в ОС Windows осуществлялась компилятором Microsoft C++, в ОС Linux — GCC C++, кода, исполняемого на ГП, — компилятором NVCC.

В силу того, что алгоритмы моделирования реализованы с использованием языка программирования C++ и кроссплатформенных библиотеки классов и инструментария разработки прикладных программ Qt, обеспечивается совместимость комплекса для разных ОС на уровне исходного кода, т.е. для запуска приложения на разных программных платформах требуется его простая перекомпиляция в соответствующей операционной системе без необходимости перекодирования.

Экспериментальные данные, полученные в результате выполнения первого этапа исследования, представлены в табл. 3 и 4. В табл. 3 приведены значения требуемого времени для расчета пространственно-энергетической доступности ИРИ при различном количестве моделируемых объектов без

Таблица 3. Время выполнения вычислений на ГП и ЦП для определения энергетической доступности источников излучений без проверки интерференции сигналов при различных значениях числа моделируемых объектов

№ п/п	Количество моделируемых ВС	Время на ЦП, с (Windows)	Время на ЦП, с (Linux)	Время на ГП, с
1	1000	2.4	2.5	1.8
2	10000	20.0	21.2	2.0
3	50000	97.3	105.4	3.0
4	100000	193.9	206.1	4.7
5	500000	975.2	1031.7	18.8

Таблица 4. Время выполнения вычислений на ГП и ЦП для определения энергетической доступности источников излучений и проверки интерференции сигналов

№ п/п	Количество ВС в зоне покрытия	Среднее количество сообщений в канале	Время на ЦП, с (Windows)	Время на ЦП, с (Linux)	Время на ГП, с
1	100	3574	21.3	21.7	7.8
2	200	7065	84.3	85.1	11.9
3	400	14182	337.1	344.5	36.6
4	800	28116	1332.7	1339.5	133.7
5	1600	56476	5249.1	5445.9	507.2

проверки интерференции сигналов в заданном моделируемом интервале времени ($T = 1000$ с).

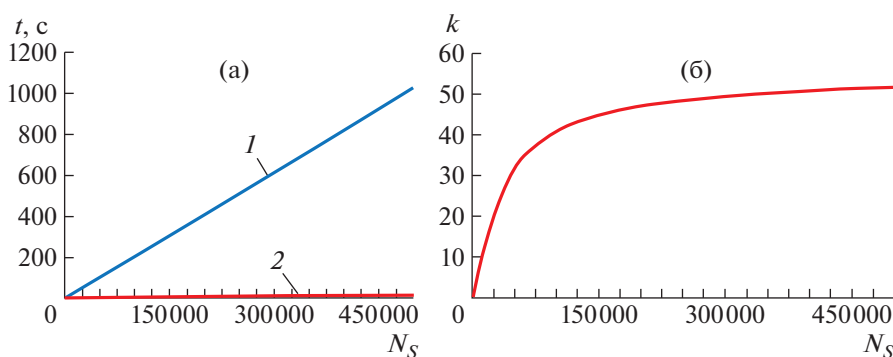
Оценки времени T выполнения вычислений для заданного числа N моделируемых объектов получены путем усреднения соответствующих значений $T_i(N)$, $i = 1, \dots, 5$, измеренных в результате проведения не менее пяти экспериментов для каждого числа N . Разброс значений времени T_i выполнения программы обусловлен системными прерываниями ОС, на обслуживание которых расходуется дополнительное время, поскольку используемые в эксперименте Windows и Linux не

являются ОС реального времени, и пользовательские процессы имеют самый низкий приоритет.

Результаты интерполяции приведенных в табл. 3 данных полиномом Лагранжа изображены на рис. 4.

На рис. 4, а цифрой 1 обозначена зависимость времени работы приложения от количества моделируемых объектов в случае последовательной организации вычислений (ОС Windows), цифрой 2 – в случае параллельной организации вычислений.

Из анализа представленных практических данных очевидно, что использование графических карт

**Рис. 4.** Результаты эксперимента: а – длительность вычислений на ЦП и ГП; б – выигрыш в производительности.

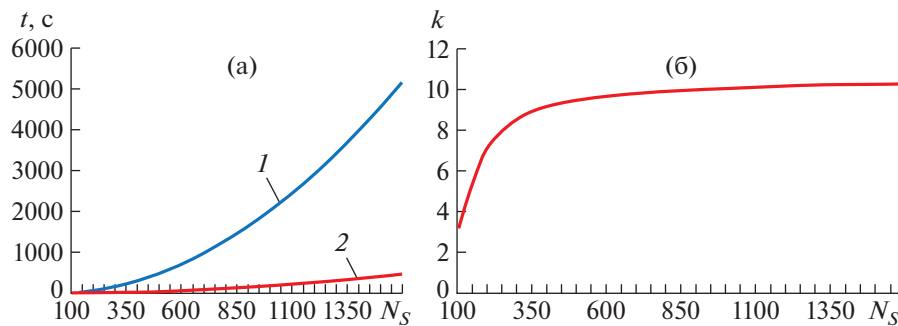


Рис. 5. Результаты эксперимента: а – зависимость длительности проверки интерференций сигналов на ЦП и ГП; б – выигрыш в производительности.

для определения энергетической доступности источников излучений приводит к существенному сокращению времени моделирования, при этом чем больше количество анализируемых объектов, тем больше достигается эффект от их применения (свыше 50 раз при числе моделируемых объектов превосходящим 500 тыс.). Также отмечается, что в ОС Windows вычисления производятся незначительно быстрее, чем в ОС Linux.

В табл. 4 приведены значения требуемого времени для проверки интерференции сигналов при различном числе моделируемых объектов, составляющих радиоэлектронную обстановку (РЭО), в зоне покрытия КА.

Время T работы программного комплекса с учетом проверки интерференций сигналов от различных источников, помимо системных прерываний, влияющих на продолжительность вычислений, также зависит от числа M сообщений в канале, являющегося случайной величиной и определяемого используемой статистической моделью распределения сигналов Mode A/C, Mode S, ADS-B (в данном случае используется модель, представленная ICAO [26]).

Результаты интерполяции полученных данных, представленных в табл. 4, полиномом Лагранжа второй степени изображены на рис. 5. Смысловое содержание обозначений на рис. 5, а осталось неизменным.

Анализ данных в табл. 4 показывает, что, во-первых, имеет место незначительное различие в длительности времени работы программного комплекса при последовательной организации вычислений на разных ОС (Windows и Linux), во-вторых, время в случае массивно-параллельной организации вычислений значительно сокращается (более чем в 11 раз при количестве моделируемых объектов превосходящим 1.5 тыс.).

В целом результаты первого этапа эксперимента показывают целесообразность применения графических карт как для определения энергетической доступности объектов, так и для проверки интерференций сигналов.

На втором этапе эксперимента использовалась динамическая модель и модель воздушной РЭО, соответствующая реальному размещению источников излучений сигналов систем авиационного радиотехнического обеспечения (MODE A/C, MODE S, MODE S ADS-B) в мировом воздушном пространстве.

Производилось моделирование функционирования КА с параметрами орбиты: долгота восходящего узла $\Omega = 0$ град; наклонение $i = 98$ град; апогей $r_a = 500$ км; перигей $r_n = 500$ км; широта перицентра $\omega = 0$ град. Интервал моделирования составляет сутки ($T = 86400$ с), время t_0 начала моделирования соответствует положению КА на орбите с истинной аномалией $\theta(t_0) = 0^\circ$.

Оценки длительности вычислений, полученные путем усреднения не менее пяти измерений, для различных значений порога чувствительности РПУ приведены в табл. 5.

Увеличивая или уменьшая значение порога чувствительности РПУ, изменяется количество $v_r(S, t)$ доступных с энергетической точки зрения объектов в зоне обслуживания КА, а, соответственно, число и время на обработку переданных сигналов.

Интерполированные данные табл. 5 графически представлены на рис. 6.

С целью исключения влияния различной мощности авиационных радиопередатчиков на количество $v_r(S, t)$ воздушных судов в зоне покрытия КА, и, соответственно, на количество принятых сигналов, длительность проверки их интерференций и вычислений в целом, мощ-

Таблица 5. Время моделирования функционирования КА

№ п/п	Чувствительность РПУ, дБВт	Время на ЦП, с (Windows)	Время на ГП, с	Сокращение времени
1	–124	3560.0	837.2	4.3
2	–126	3696.2	952.7	3.9
3	–128	5057.4	1089.6	4.6
4	–130	7088.4	1307.3	5.4
5	–132	10151.5	1606.8	6.3
6	–134	14714.2	2030.1	7.2

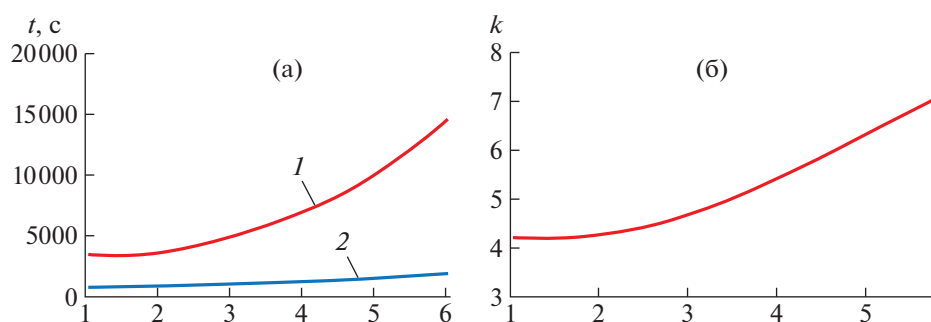


Рис. 6. Результаты эксперимента: а – время моделирования на ЦП и ГП; б – выигрыш в производительности.

ность всех передатчиков в эксперименте принималась равной 29 дБВт.

Анализ полученных данных показывает более чем семикратное сокращение времени проведения вычислений, при этом модель воздушной РЭО включает менее 10 тыс. объектов. В случае моделирования надводной РЭО, состоящей из более чем 500 тыс. морских судов, выигрыш в производительности вычислений будет несколько больше. С целью сокращения объема излагаемого материала результаты сравнительного анализа длительностей моделирования функционирования КС автоматической идентификации морских судов в работе не приводятся.

Результаты представленного эксперимента в целом явно свидетельствуют о значительном сокращении времени моделирования в случае массивно-параллельной организации вычислений, что подтверждает целесообразность использования гетерогенного программирования при разработке программных средств имитационного моделирования функционирования космических систем наблюдения за движением воздушных и морских судов.

Полученное значение ускорения вычислений близко к теоретическому значению максимального ускорения для соответствующей доли парал-

лельных вычислений, что позволяет говорить об успешном решении поставленной задачи.

В дальнейшем планируется использование разработанных алгоритмов в методе синтеза космических систем, основанном на оптимизационно-имитационном подходе, который предполагает многократное моделирование функционирования орбитальной группировки для различных параметров ее баллистической структуры и характеристик бортовой аппаратуры приема сигналов.

6. ЗАКЛЮЧЕНИЕ

В статье рассматривались вопросы, связанные с оцениванием целевых показателей функционирования космических систем контроля движения воздушных и морских судов. Главным недостатком используемых имитационных моделей является большая длительность вычислений в случае, когда требуется воспроизвести радиоэлектронную обстановку, близкую к реальной (число моделируемых объектов может достигать свыше десятков и сотен тысяч). Предлагается подход к моделированию, основанный на массивно-параллельной организации вычислений и использовании современных графических карт с возможностью производить вычисления общего назначения.

Результаты модельно-кибернетического эксперимента показали целесообразность примене-

ния гетерогенных систем для оценивания целевых показателей функционирования систем автоматической идентификации и определения местоположения подвижных объектов.

СПИСОК ЛИТЕРАТУРЫ

1. *Андреев А.М.* Анализ развития спутниковой автоматической идентификационной системы мониторинга движения судов. Ч. 2. Средства мониторинга. Труды Военно-космической академии имени А.Ф. Можайского. 2016. Вып. 652. С. 7–15.
2. *Романов А.А., Романов А.А., Тюлин А.Е.* Малоразмерные космические аппараты мониторинга подвижных объектов ОАО “Российские космические системы”: состояние и перспективы. Ракетно-космическое приборостроение и информационные системы. 2015. Т. 2. № 1. С. 3–10.
3. *Hoye G., Eriksen T., Meland B.J., Narheim G.* Space-Based AIS for Global Maritime Traffic Monitoring. *Acta Astronautica*. 2008. V. 62. P. 240–245.
4. *Trong T.V., Dinh Q.T., Van T.D., Quang H.P., Nguyen H.* Constellation of Small Quick-Launched and Self-Deorbiting Nano-Satellites with AIS Receivers for Global Ship Traffic Monitoring. Proc. 2nd Nano-Satellite Symp., March 2011, Tokyo, Japan (online). <https://uu.divaportal.org/smash/get/diva2:424564/FULLTEXT01.pdf>, 05.09.2016.
5. *Van Der Pryt R., Vincent R.* The CanX-7 Nanosatellite ADS-B Mission: A Preliminary Assessment // Positioning. 2017. V. 8. P. 1–11.
6. *Brodsky Y., Rieber R., Nordheim T.* Balloon-borne air traffic management (ATM) as a precursor to space-based ATM. *Acta Astronautica*. 2012. V. 70. P. 112–121.
7. *Barson J.V.* Automatic Dependent Surveillance-Broadcast – The First Step in the FAA’s Next-Generation Air Transportation System. *Aviation, Space and Environmental Medicine*. 2009. V. 80. № 4. P. 422–423.
8. *Werner K., Bredemeyer J., Delovski T.* ADS-B over satellite: Global air traffic surveillance from space. Tyrhenian International Workshop on Digital Communications – Enhanced Surveillance of Aircraft and Vehicles (TIWDC/ESAV). 2014. P. 47–52.
9. N2YO.com (online). <https://www.n2yo.com>.
10. Marine Traffic (online). <https://www.marinetraffic.com/ru>.
11. VesselFinder (online). <https://www.vesselfinder.com>.
12. MyShipTracking (online). <https://www.myshiptracking.com>.
13. Flightradar24 (online). <https://www.flightradar24.com>.
14. ADS-B Exchange (online). <https://www.adsbexchange.com>.
15. Planefinder (online). <https://planefinder.net>.
16. *Скорыходов Я.А.* Моделирование функционирования космических и наземных систем наблюдения за воздушными движениями. Труды СПИИ РАН. 2018. Вып. 6 (61). С. 29–60. <https://doi.org/10.15622/sp.61.2>
17. *Скорыходов Я.А., Андреев А.М.* Моделирование функционирования космического сегмента системы автоматической идентификации морских судов. Информационно-управляющие системы. 2018. Вып. 2 (93). С. 36–48. <https://doi.org/10.15217/issn1684-8853.2018.2.36>
18. *Скорыходов Я.А., Махров К.Б., Малышев Д.В.* Имитационная модель функционирования космической системы контроля движения морских судов. Труды ВКА имени А.Ф. Можайского. 2017. Вып. 657. С. 23–33.
19. *Кузнецов А.М., Романов А.А., Романов А.А.* Моделирование приема коллизий сигналов АИС на борту КА. Ракетно-космическое приборостроение и информационные системы. 2015. Т. 2. № 1. С. 25–36.
20. *Mendes S., Amado S., Teresa V., Scorzolini A., Perini V., Sorbo A.* Satellite AIS – an End-to-End Simulation Approach. Proc. 11th Intern. WS on Simulation & EGSE Facilities for Space Programmes, Sept. 28–30, 2010, Noordwijk, Netherlands (online). Доступно по ссылке: <https://indico.esa.int/indico/event/109/session/15/contribution/48/material/0/0.pdf>.
21. *Chen Y.* Research on Detection Probability of Space-based AIS for Real Scenarios. <https://doi.org/10.5013/IJSSST.a.17.30.03> (online). <http://ijssst.info/Vol-17/No-30/paper3.pdf>.
22. *Menghui Y., Yongzhong Z., Li F.* Collision and Detection Performance with Three Overlap Signal Collisions. Space-Based AIS Reception: Proc. of 11th Intern. Conf. on Trust, Security and Privacy in Computing and Communications, Jun. 25–27 2012, Liverpool, United Kingdom. P. 1641–1648.
23. *Van Der Pryt R., Vincent R.* A Simulation of Signal Collisions over the North Atlantic for a Spaceborne ADS-B Receiver Using Aloha Protocol. Positioning. 2015. V. 6. № 3. P. 23–31.
24. *Van Der Pryt R., Vincent R.* A Simulation of the Reception of Automatic Dependent Surveillance-Broadcast Signals in Low Earth Orbit. *International Journal of Navigation and Observation*, 2015, 11 p.
25. Technical Characteristics for an Automatic Identification System Using Time-Division Multiple Access in the VHF Maritime Mobile Band (online). Доступно

- по ссылке: <https://www.itu.int/rec/R-REC-M.1371-3-200706-S/en>.
26. Dynamic Statistical Studies on Satellite Reception of the ADS B Signal for Global Flight Tracking for Civil Aviation. World Radiocommunication Conference (WRC-15). Geneva, 2–27 November 2015, 19 p. (online). <https://www.itu.int/md/R15-WRC15-C-0100/en>.
27. REPORT ITU-R M.2084. Satellite Detection of Automatic Identification System Messages. (online). <https://www.itu.int/pub/R-REP-M.2084>.
28. NVIDIA. Developer zone (online). <https://developer.nvidia.com>.
29. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. М.: ДМК Пресс, 2010 г., 232 с.
30. Sanders J., Kandrot E. CUDA by Example: An Introduction to General-Purpose GPU Programming. Addison-Wesley Professional, 1st edn., Boston, 2010, 320 p.
31. Roy A.E. Orbital Motion. Bristol (UK): Institute of Physics Publishing, 2005, 526 p.