

Метод обработки радарных данных на базе системы массово-параллельного исполнения заданий Apache Spark

В. П. ПОТАПОВ, С. Е. ПОПОВ, М. А. КОСТЫЛЕВ*

Институт вычислительных технологий СО РАН, Новосибирск, Россия

*Контактный e-mail: 5999ft@gmail.com

Представлен современный подход к созданию распределенного программного комплекса для потоковой пред- и постобработки радарных снимков. Отличительной особенностью системы является применение массово-параллельной технологии Apache Spark, что позволило использовать аппаратное обеспечение общего назначения, а также предоставило возможность исполнения существующих алгоритмов, не предназначенных для распределенной обработки на множестве узлов без изменения реализации алгоритма.

В работе приводится сравнение технологий распределенных вычислений, представлено общее описание кластера и механизма выполнения задач пре- и постпроцессинга радарных данных, также приведены особенности имплементации конкретных задач в рамках предложенного подхода. Представлены результаты тестирования разработанных алгоритмов на демонстрационном кластере.

Ключевые слова: Apache Spark, Apache Hadoop, распределенные информационные системы, радарная интерферометрия, алгоритмы обработки.

Введение

Данные дистанционного зондирования Земли получили широкое распространение практически во всех отраслях как науки, так и промышленности, и они используются для решения самых разнообразных задач, число которых постоянно растет. Исключением не стали и радарные данные, получаемые с космических аппаратов радиолокационной съемки.

Основным аппаратом анализа радарных данных является метод радарной дифференциальной интерферометрии (DInSAR), который заключается в построении цифровых моделей рельефа (ЦМР) на основе расчета разности фаз нескольких радарных снимков одной территории [1, 2]. В то же время пред- и постобработка радарных данных обусловлена повышенными требованиями к вычислительным ресурсам даже на современном аппаратном уровне. Тем не менее математические модели и алгоритмизация, заложенные в процедуры пред- и постобработки радарных данных, корректно могут быть перенесены на технологию параллельных вычислений. К основным процедурам обработки можно отнести, например, корегистрацию снимков [3], расчет когерентности и формирования интерферограммы [4], а также развертку фазы и последующий расчет [4–7], которые являются наиболее ресурсоемкими этапами построения ЦМР.

Оптимизации и ускорению работы, в том числе и за счет распараллеливания расчета математических алгоритмов в процедурах обработки радарных данных посвящено большое количество научных работ [8].

В [6, 7, 9–11] используется технология CUDA для улучшения производительности процедуры развертки фазы на базе различных математических моделей. В частности, рассматриваются уравнение Пуассона с весовыми коэффициентами, метод сопряженных градиентов [6, 7], метод роста регионов и отсечения ветвей [8, 9], компенсации фазового набегания [10] и совмещения (корегистрации) радарных снимков [11]. В работе [13] представлен алгоритм генерации нативных (синтетических) радарных данных, симулирующих различные ошибки/шумы принимающей/передающей части космических аппаратов в зависимости от наблюдаемой земной поверхности, с целью выработки программных методов коррекции.

Наряду с GPU-имплементацией существуют программные реализации алгоритмов обработки радарных данных на базе параллельных вычислений на CPU. В работе [14] предложена интеграция технологии параллельных вычислений MPI [16] с системой комплексной обработки радарных данных Doris (Delft object-oriented radar interferometric software) [17]. Рассмотрены основные этапы обработки InSAR/PS-InSAR-данных, включая наиболее ресурсоемкие: корегистрация, формирование интерферограммы и развертка фазы. Предложена стратегия распараллеливания декомпозиции главного/подчиненного изображений с большим количеством сегментов и частичным перекрытием границ, обеспечивающих минимальное межпроцессорное взаимодействие.

Преимуществом подходов, показанных в [7–15], является алгоритмизация решений задач пре- и постобработки, которые легко переносятся на параллельные вычисления, с применением уже широко известных процедур, реализованных в библиотеках для GPU и CPU. Так, в упомянутых выше работах [7, 11–15] за основу взяты элементы пакетов CuFFT, CuBLASS, CuSPARSE, PBLASS, FFTW, ScaLAPACK [18–20].

В последнее время для стандартного алгоритмического аппарата DInSAR получил широкое распространение метод интерферометрии малых базовых линий (SBAS), основанный на совместном использовании длинных временных серий радарных изображений одной территории, полученных в повторяющейся геометрии съемки и хронологически упорядоченных моментах времени. SBAS относят к одной из наиболее ресурсоемких технологий обработки радарных данных [14]. Так, например, только на начальном этапе формируются интерферограммы на базе комплексного перемножения всех возможных пар главного/подчиненного изображений. Затем для каждой парной интерферограммы выполняется развертка фазы с целью получения полного смещения в направлении на спутник. При проведении развертки чаще всего сначала применяют алгоритмы “минимальной стоимости”, а затем алгоритм “роста регионов”. Учитывая тот факт, что для выявления динамики и средней скорости изменения вертикальных смещений земной поверхности с погрешностью разности высот ЦМР не более чем 3 м/пиксель необходимо как минимум 30 изображений, на отдельных стадиях расчетов возникает резкое уменьшение производительности (экспериментальные расчеты показывают время от 3 до 5 ч для 12 пар снимков небольшого разрешения в 3000×1000 пикселей).

Так как сегодня существует огромное количество программно реализованных высокопроизводительных алгоритмов технологических этапов обработки радарных данных, целесообразно применять их совместно в облачной инфраструктуре. Основными преимуществами такой интеграции являются:

- возможность хранения и работы с большими объемами радарных данных за счет применения распределенной файловой системы;
- высокая масштабируемость, достигаемая за счет добавления дополнительных узлов в вычислительный кластер без необходимости внесения изменений в алгоритмы обработки;
- возможность организации работы с данными, поступающими в потоковом режиме.

Сама облачная инфраструктура выступает как интегратор распределенного исполнения программного кода на данных, получаемых в потоковом режиме. Технологии распределенных вычислений в облачной среде получили широкое распространение в различных областях науки, некоторые работы проводятся и в сфере тематической обработки радарных данных [21, 22], однако авторам не удалось найти примеры применения технологий потоков обработки в системах с массово-параллельным исполнением заданий на базе открытых распределенных технологий. Поэтому задача разработки интегрального программного комплекса для пре- и постпроцессинга радарных изображений в распределенной среде с массово-параллельными механизмами (на примере Apache Spark) является, на наш взгляд, весьма актуальной в современной радарной интерферометрии.

1. Сравнение технологий распределенных вычислений

Все бóльшую популярность приобретают облачные системы распределенной обработки данных на базе технологии MapReduce. Такие системы включают в себя распределенную файловую систему, обеспечивают отказоустойчивость при выходе из строя отдельных узлов кластера, повышающих надежность хранения данных и выполнения их обработки, а также возможность использования оборудования широкого назначения с низкой стоимостью.

Для сравнения взяты различные реализации технологии MapReduce в рамках экосистемы Apache Hadoop и рассмотрены применительно к задаче обработки радарных данных (табл. 1). Наиболее важный критерий при выборе вычислительной платформы — возможность хранения промежуточных результатов в оперативной памяти, что позволяет получить существенно бóльшую производительность при обработке радарных данных в сравнении со стандартным подходом, например Hadoop MapReduce [22], что позволяет сделать Apache Spark.

Т а б л и ц а 1. Сравнение технологий распределенных вычислений

Характеристика	Apache Spark	Tez	Hadoop MapReduce
Интерактивный режим	Да	Да	Нет
Языки программирования	Java, Scala, Python, R	Java	Java
Работа в потоковом режиме	Да	Нет	Нет
Хранение промежуточных результатов	RAM, ROM	RAM, ROM	ROM

Технологии Apache Spark и Tez предлагают более высокую производительность по сравнению с Hadoop MapReduce (в основном за счет активного использования оперативной памяти в процессе вычислений), что делает их более пригодными для применения совместно с алгоритмами обработки радарных данных.

Также авторы учли наличие программного API для языков Java, Scala, Python, R у технологии Apache Spark, в то время как Apache Tez и Hadoop MapReduce имеют API только для языка Java. Это важное преимущество как при разработке конечного решения, так и при прототипировании, когда значительно повышается скорость разработки различных алгоритмов за счет интерактивного режима работы, доступного на языках Scala, Python и R. Еще одним достоинством технологии Apache Spark перед Tez является наличие большого сообщества, участвующего в развитии проекта, а также значительное количество вспомогательных технологий, предоставляемых как внутри проекта Apache Spark (Spark Streaming, Spark ML, Spark GraphX, Spark SQL), так и за счет интеграции с другими решениями.

Важным преимуществом системы Apache Spark также является возможность потоковой обработки данных, которая упрощает создание автоматической системы пре- и постпроцессинга радарных данных из-за отсутствия необходимости создавать собственную программную реализацию для работы с потоком радарных данных [23].

На основании результатов сравнения авторами выбрана технология массово-параллельных вычислений Apache Spark. К преимуществам этой технологии относятся:

- высокая масштабируемость, достигаемая за счет добавления новых узлов в вычислительный кластер, без необходимости внесения изменений в применяемые алгоритмы;
- встроенная возможность работы в режиме реального времени, позволяющая создавать алгоритмы потоковой обработки радарных данных;
- большое количество вспомогательных технологий, необходимых для организации системы, которая будет поддерживать полный цикл задач пре- и постпроцессинга радарных данных.

2. Постановка задачи

Проведенный анализ методов обработки радарных данных и сравнение технологий параллельных распределенных вычислений позволили установить, что для организации потоковой пре- и постобработки радарных снимков целесообразно создать распределенный программный комплекс на базе массово-параллельной технологии Apache Spark со следующими функциональными особенностями:

- возможность интеграции в программный код существующих высокопроизводительных решений для отдельных этапов обработки радарных данных дистанционного зондирования Земли (например, расчета когерентности, формирования интерферограмм, развертки фазы);
- возможность автоматического выбора высокопроизводительной параллельной технологии (например, CUDA, MPI и пр.) в расчетном ядре реализуемых алгоритмов;
- использование общего распределенного пространства для хранения промежуточных результатов расчетов в виде бинарных данных стандартов BSQ с возможностью доступа к ним отдельных алгоритмов, применяемых на последующих этапах обработки с различных узлов экосистемы Apache Spark.

3. Программная реализация

Для решения задач пред- и постобработки радарных снимков создан распределенный программный комплекс на базе технологий обработки больших данных Apache с применением языков программирования Java и Scala, также была использована технология распределенных наборов данных (Spark RDD), операции над элементами которых могут выполняться параллельно. На рис. 1 представлена архитектура программного комплекса.

Разработанное решение обладает следующими функциональными особенностями.

Распределенная отказоустойчивая файловая система на базе технологии HDFS позволяет хранить данные и их производные на всех этапах обработки и предоставляет доступ к данным с различных вычислительных узлов кластера. Такой подход позволил отказаться от применения дорогостоящих систем хранения и предоставляет возможность увеличения доступного пространства за счет простого добавления новых узлов кластера. Отвечает за компонент “Файловый менеджер” (рис. 1). Данная файловая система является базовым компонентом Apache Hadoop и имеет тесную интеграцию с платформой Apache Spark.

Распределенная вычислительная платформа на базе технологии Apache Spark позволяет производить параллельную обработку потоковых радарных данных на множестве узлов с максимально возможным объемом обрабатываемых данных, ограничиваемым только количеством узлов кластера. Масштабирование программного комплекса

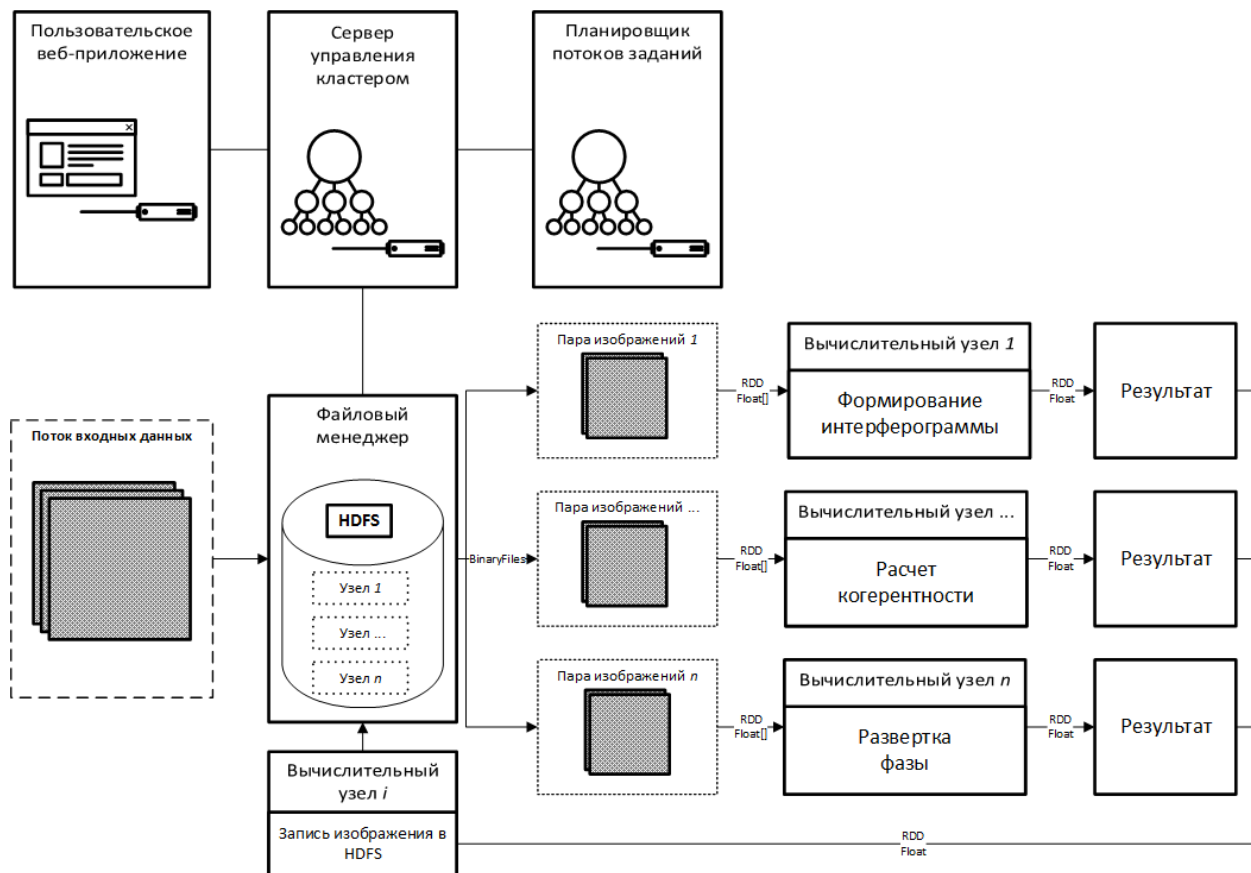


Рис. 1. Архитектура распределенного программного комплекса

осуществляется добавлением новых узлов без необходимости изменения программной реализации и используемых алгоритмов. Отвечает за “вычислительные узлы”.

Система планирования потоков работ на базе технологии Apache Oozie [24] позволяет организовать процесс обработки потока радарных данных в виде последовательности действий, каждое из которых выполняется после успешного завершения предыдущего. Последовательности могут исполняться либо автоматически при поступлении новых данных, либо по запросу пользователя, либо по расписанию. Отвечает за компонент “Планировщик потоков заданий”. Эта система выбрана для организации процесса пре- и постобработки радарных данных в виде последовательности заданий, которые необходимо выполнить для входных данных в зависимости от различных условий.

Пользовательское веб-приложение на базе технологии Apache Hue [25] предоставляет возможность взаимодействия с программным комплексом. Приложение состоит из файлового менеджера, позволяющего загружать новые файлы, а также просматривать и редактировать уже существующие файлы в распределенной файловой системе. Вторым важным компонентом системы является интегрированная среда управления потоками работ с возможностью их создания и редактирования, а также запуска и мониторинга. Отвечает за компонент “Пользовательское веб-приложение”. Выбор данной технологии обусловлен наличием интеграции со всеми указанными выше компонентами, что предоставляет пользователю возможность просматривать и изменять операции, выполняемые над радарными данными, осуществлять мониторинг состояния системы (просмотр выполняемых и выполненных работ), а также получать доступ к исходным данным и результатам их обработки через встроенный файловый менеджер, интегрированный с распределенной файловой системой HDFS.

В зависимости от реализации конкретного алгоритма деление входных данных производится следующими способами:

- 1) отдельные файлы радарных данных, предназначенные для обработки, делятся на области, вычисление результата для каждой области производится на отдельном узле, затем полученные данные объединяются и сохраняется готовое изображение;
- 2) отдельное изображение обрабатывается целиком на вычислительном узле без деления на области, параллелизация достигается за счет обработки большого количества изображений на множестве узлов.

Рассмотрим реализацию приведенной выше обобщенной диаграммы потоков данных на примере алгоритма расчета фазы. На рис. 2 представлена блок-схема предложенного алгоритма. В процессе работы алгоритма производится параллельная развертка интерферометрической фазы, а количество одновременно выполняемых расчетов определяется количеством узлов в кластере. Входными данными являются массив значений интерферограммы в каждой точке изображения и массив значений когерентности каждого значения интерферограммы. В общем виде этап развертки фазы состоит из следующих шагов:

- Выбираются начальные точки с наибольшей когерентностью, они объявляются развернутыми и образуют регионы. Региону с наибольшей когерентностью начальной точки присваивается номер 1 и т. д., при уменьшении значения когерентности увеличивается номер региона.
- Для каждой развернутой точки ищутся соседние неразвернутые (целевые) точки, образующие кольца роста регионов. Назовем этот процесс итерацией роста.
- Для каждой целевой точки кольца роста вычисляется предсказываемое значение по формуле

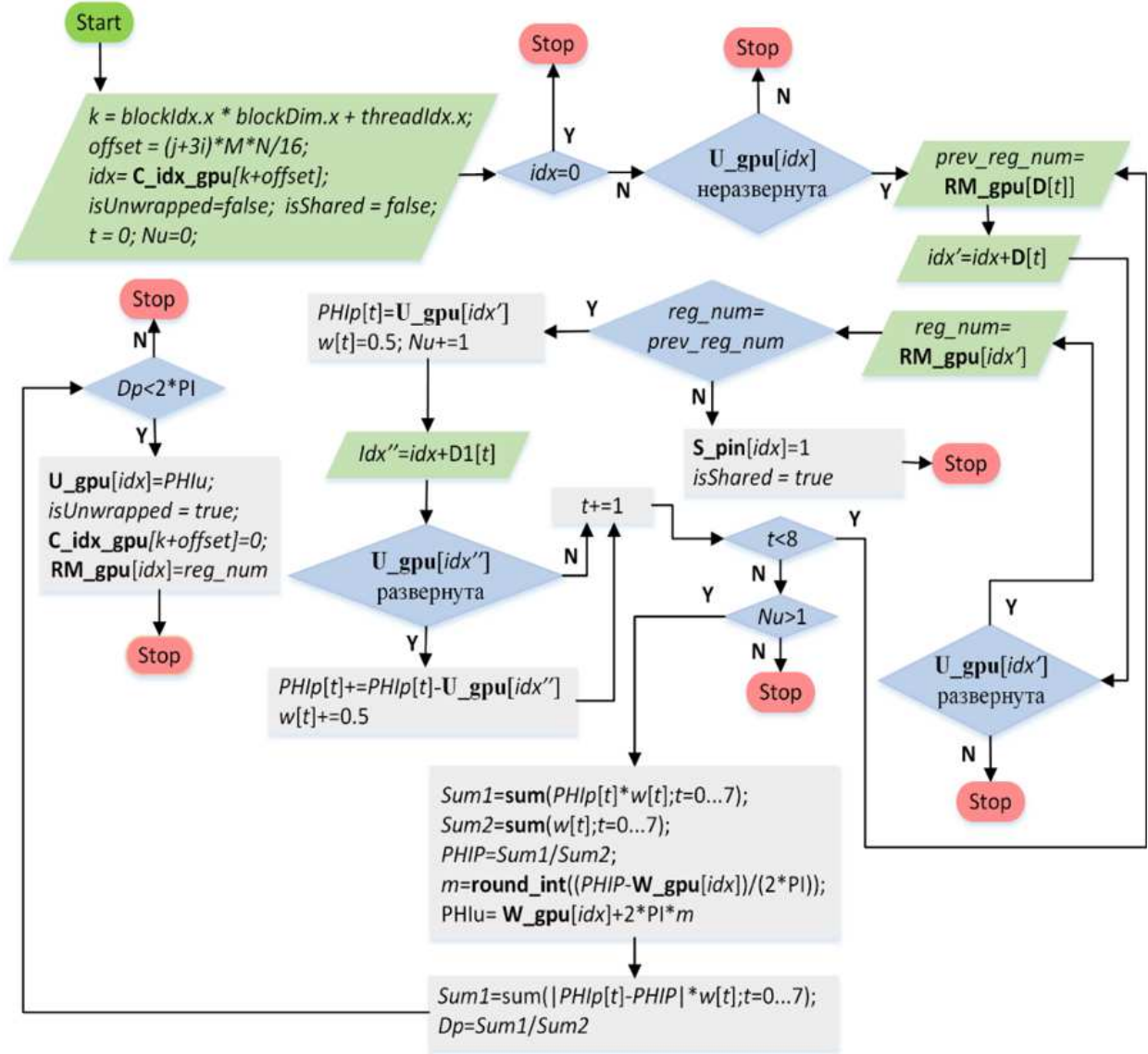


Рис. 2. Распределенный алгоритм развертки фазы

$$\phi^p = \frac{\sum_{\kappa=1}^{N_u} \omega^\kappa \phi_\kappa^p}{\sum_{\kappa=1}^{N_u} \omega^\kappa}, \quad (1)$$

где $N_u > 1$ (за исключением области начальных точек), N_u — количество соседних пикселей с развернутой фазой; $\phi_k^p = 2\phi[k] - \phi[k']$, $\omega^k = 1.0$, если на пути предсказания к точке лежат два развернутых пикселя, если один развернутый пиксель $\phi_k^p = \phi[k]$, $\omega^k = 0.5$; $\phi[k]$ и $\phi[k']$ — значения развернутых фаз на пути предсказания соответственно.

- Рассчитывается число неоднозначности

$$m = \text{nint}((\phi^p - \phi_\omega)/2\pi), \quad (2)$$

где ϕ_ω — значение свернутой фазы, целевого пикселя; nint — оператор взятия ближайшего целого.

- Рассчитывается значение развернутой фазы $\phi_u = \phi_\omega + 2\pi/m$.
- Значение развернутой фазы в целевой точке принимается, если выполняются условия надежности $d^p < T_p$, $d_u < T_u$ и значение когерентности в целевой точке больше параметра релаксации $T_c = 0.9 \dots 0.2$, где

$$d^p = \frac{\sum_{k=1}^{N_u} \omega^k |\phi_k^p - \phi^p|}{\sum_{k=1}^{N_u} \omega^k}, \quad d_u = |\phi_u - \phi^p|, \quad T_p = T_u = \pi/2.$$

Если два региона имеют точки пересечения, то запускается процедура их объединения.

- Определяется количество общих точек N_{ov} .
- Для каждой общей точки вычисляется разность их чисел неоднозначности ($D_m d = m_{R1} - m_{R2}$), где m_{R1} , m_{R2} — числа неоднозначности общей точки в регионах с номерами R1 и R2 соответственно.
- Находятся мода значений разностей $D_m d$ и количество точек, образующих ее (N_c).
- Если $\frac{N_c}{N_{ov}} \geq T_r r$ и $N_{ov} \geq T_r n$, где $T_r r = 0.75$, $T_r n = 3$, то регионы объединяются. В точках склейки значение развернутой фазы берется для региона с наименьшим номером, остальные точки исключаются из алгоритма. Если условия не выполнены, все общие точки исключаются.

В табл. 2 дано описание объектов (массивов), используемых в программной реализации усовершенствованного алгоритма роста регионов. На рис. 3 представлена блок-схема этапа развертки интерферометрической фазы. В табл. 3 приведены реализованные алгоритмы обработки радарных данных с указанием особенностей их распараллеливания.

Расчетная часть представленных алгоритмов реализована в соответствии с аналогичными схемами в основных системах обработки радарных данных. Так, расчет

Т а б л и ц а 2. Основные объекты (массивы) алгоритма

Название объекта, массива	Описание
isUnwrapped	Флаг останова процедуры итераций роста
isShared	Флаг запуска процедуры объединения регионов
Nu	Счетчик количества развернутых соседних точек
D = -N-1, -N, -N+1, -1, 1, N-1, N, N+1 и D1 = -2N-2, -2N, -2N+2, -2, 2, 2N-2, 2N, 2N+2	Массивы констант для получения значений индексов соседних точек к целевой
sum(arg, index) и round_int(arg)	Функции нахождения суммы аргументов с указанными индексами и округления до ближайшего целого соответственно
C_idx_gpu	Массив индексов целевых точек W_gpu, удовлетворяющих текущим параметрам надежности (когерентность > T_c)
U_gpu	Массив значений развернутых фаз
RM_gpu	Массив номеров регионов
S_pin	Массив общих точек

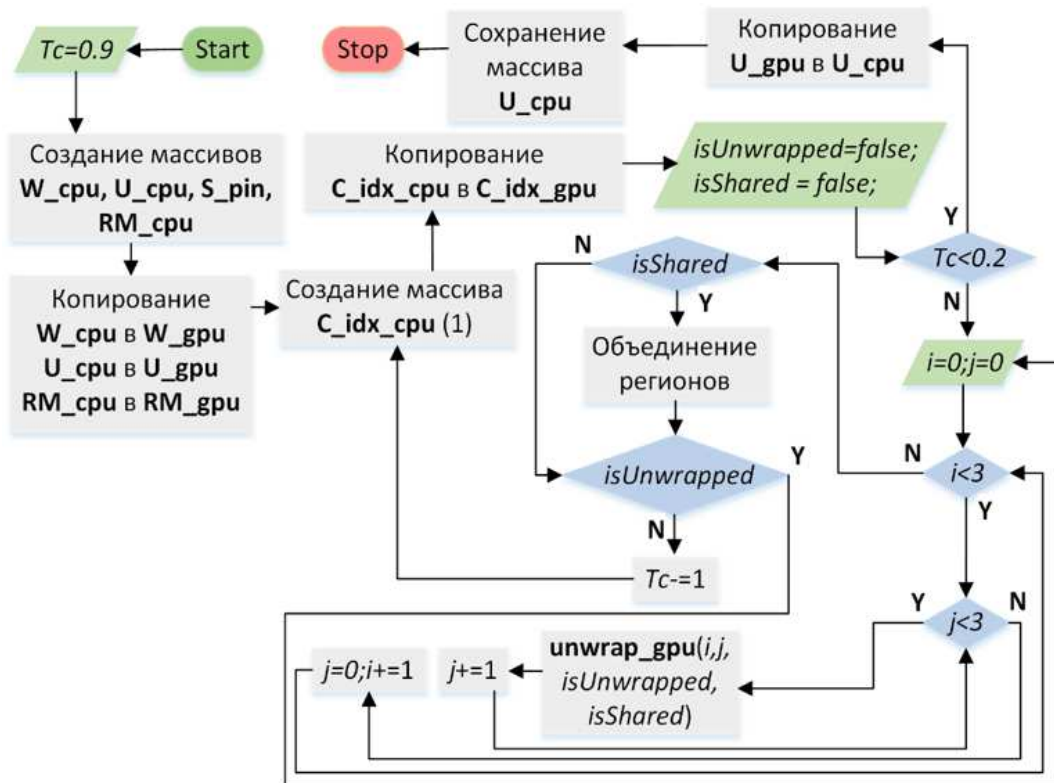


Рис. 3. Алгоритм развертки интерферометрической фазы

фазы и формирование интерферограммы производятся согласно выражениям (3) и (4), а расчет когерентности — выражению (5). Для развертки интерферометрической фазы применяется алгоритм роста регионов:

$$Phase_{i,j} = \arctan \left[\frac{(float)I_{i,j}}{(float)(I_{i,j} \gg 32)} \right], \quad (3)$$

Т а б л и ц а 3. Особенности реализации различных алгоритмов

Алгоритм	Входные данные	Особенности реализации
Расчет фазы	Бинарные представления радарных данных аппарата Cosmo-SkyMed уровня L1A в формате (.hdr + SingleLookComplexbinary) на примере Exelis	Изображение делится на отдельные значения (точки) последовательно, каждое изображение рассчитывается на множестве узлов по частям
Формирование интерферограммы		Изображение делится на регионы заданного размера (7 × 7 точек), каждое изображение рассчитывается на множестве узлов по частям
Расчет когерентности		То же
Развертка фазы		Изображение обрабатывается целиком на отдельном вычислительном узле, производится параллельная обработка множества изображений

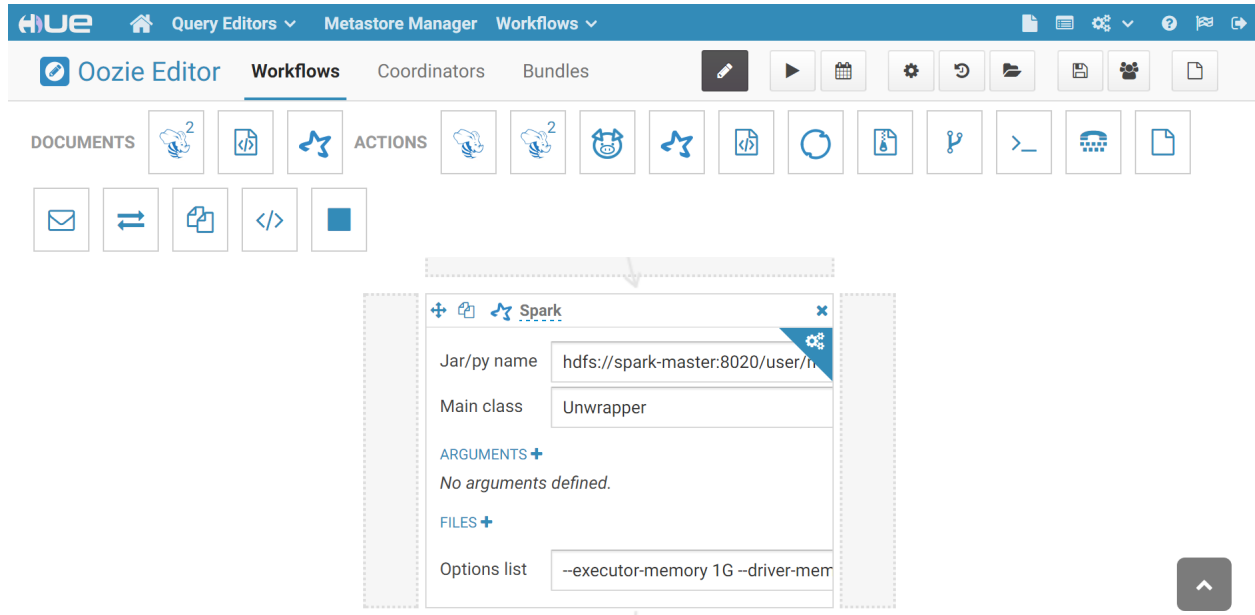


Рис. 4. Графический интерфейс workflow-задания развертки интерферометрической фазы

$$\begin{aligned}
 I_{i,j} &= (float)(SLC_{i,j}^M \cdot SLC_{i,j}^S) + (float)(SLC_{i,j}^M \gg 8)(SLC_{i,j}^S \gg 8), \\
 I_{i,j} \gg 32 &= (float)(-(SLC_{i,j}^M(SLC_{i,j}^S \gg 8)) + (float)((SLC_{i,j}^M \gg 8)SLC_{i,j}^S)), \\
 I_{i,j} &= (I_{i,j})|((I_{i,j}) \ll 32),
 \end{aligned} \tag{4}$$

$$C_{i,j} = \frac{\frac{1}{49} \sum_{i=0}^7 \sum_{j=0}^7 |I_{i,j}|}{\left(\frac{1}{49} \sum_{i=0}^7 \sum_{j=0}^7 |SLC_{i,j}^M|^2 \right) \left(\frac{1}{49} \sum_{i=0}^7 \sum_{j=0}^7 |SLC_{i,j}^S|^2 \right)}. \tag{5}$$

Особенностью приведенного способа обработки радарных данных является использование распределенных технологий (HDFS и Apache Spark), что позволяет применять данные алгоритмы для потоков радарных данных. Графический интерфейс разработанного алгоритма в виде workflow-задания развертки интерферометрической фазы представлен на рис. 4.

4. Результаты тестирования

Разработанные алгоритмы протестированы на кластере из четырех и восьми узлов, созданном с использованием CDH5 — дистрибутива среды Hadoop, вычисления произведены на отдельном узле без применения распределенного подхода.

При создании кластера использована версия среды CDH 5.7.0, которая включает Apache HDFS 2.6, Apache Spark 1.6.0, Apache Hue 3.9, Apache Oozie 1.7. Компоненты хранения и вычисления данных установлены на каждом узле кластера, в то время как система управления потоками работ и пользовательское веб-приложение имеются только на одном узле, выбранном как основной. Все узлы кластера были подключены

к локальной вычислительной сети с пропускной способностью 1 Гбит/с. При проведении вычислений на отдельном компьютере использован только вычислительный компонент представленного программного комплекса (Apache Spark), так как остальные компоненты целесообразно использовать только для кластера.

В качестве тестовых данных взяты радарные изображения размера $16\,384 \times 16\,384$ точек с объемом одного файла ~ 2 Гбайт. Для расчета фазы было использовано одно изображение, а для формирования интерферограммы, расчета когерентности и развертки фазы — пара изображений, что обусловлено особенностями работы представленных алгоритмов обработки радарных данных.

Для оценки эффективности распределенного подхода тестирование проводилось в условиях необходимости одновременного исполнения восьми заданий каждым из алгоритмов. Входные данные для каждого задания были одинаковыми, так как использовались копии одних и тех же исходных изображений. Такой подход позволил избежать излишних погрешностей при оценке производительности алгоритма в зависимости от количества узлов. Каждый узел кластера соответствует следующей конфигурации: Ubuntu 16.04; Oracle JDK 1.8; Intel Xeon E5-2620 v2 @ 2.10GHz; 6GB RAM. Результаты тестирования представлены на рис. 5.

Важно отметить, что для алгоритмов расчета фазы, формирования интерферограммы и расчета когерентности применяется разделение на уровне изображения, за счет чего и достигается ускорение при обработке отдельного изображения по сравнению с локальным подходом. Алгоритм развертки фазы предполагает проведение всех необходимых вычислений для одной пары изображений на одном вычислительном узле, а ускорение достигается за счет возможности одновременной обработки изображений, количество которых равно числу узлов кластера. При этом для алгоритма развертки фазы максимальное количество узлов кластера, при котором будет достигнуто ускорение,

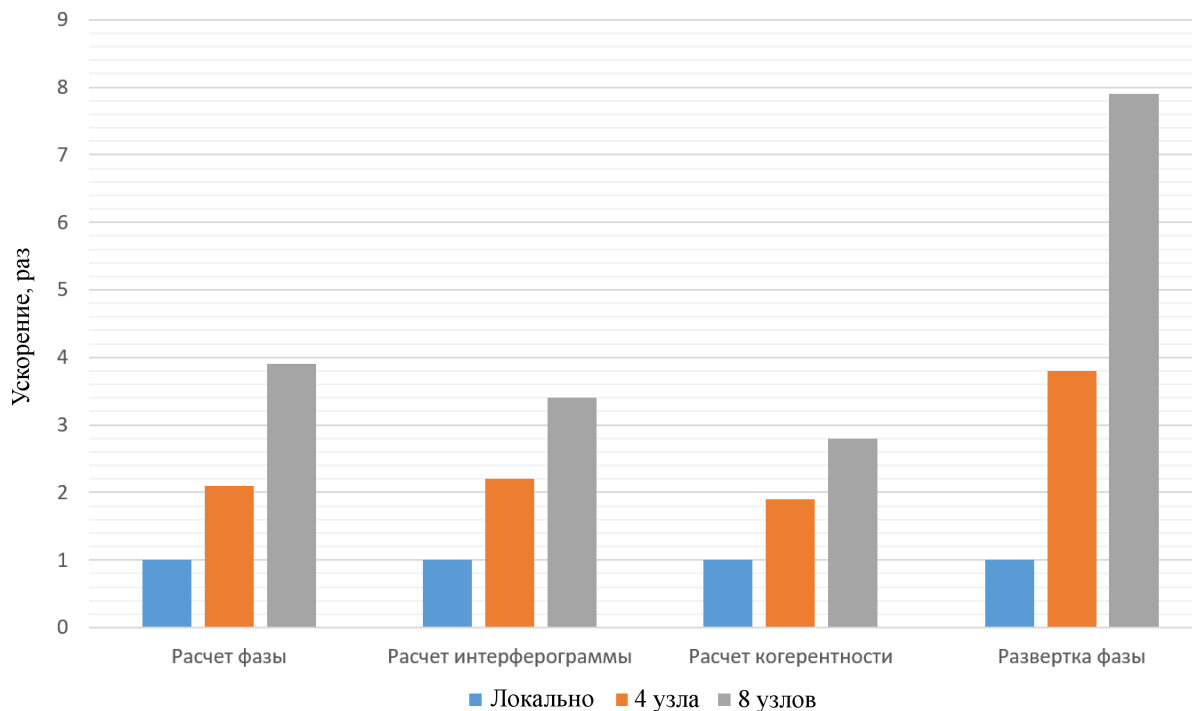


Рис. 5. Результаты тестирования

равно количеству одновременно исполняемых заданий, так как дополнительные узлы не будут участвовать в процессе обработки данных. Также для алгоритмов с разделением на уровне изображения более эффективным может оказаться подход, при котором одновременно обрабатывается множество изображений с дополнительным разделением каждого изображения на части.

Выводы

В результате анализа различных подходов, применяемых при обработке радарных данных, а также обзора технологий распределенных вычислений предложен и реализован распределенный программный комплекс на базе массово-параллельной технологии Apache Spark для потоковой пре- и постобработки снимков. По сравнению с традиционными подходами к обработке радарных данных, в которых параллельные вычисления либо не применяются, либо применяются только для повышения производительности расчетов, предложенное решение ориентировано на обработку большого количества потоковых данных. Программная реализация содержит веб-интерфейс, позволяющий пользователю взаимодействовать с кластером, получая доступ к распределенной файловой системе, а также создавать и запускать существующие потоки работ, существенно уменьшая вычислительные затраты.

Благодарности. Работа выполнена в рамках базового проекта IV.38.2.8 “Геоинформационные технологии и системы распределенного типа для природно-техногенного мониторинга и контроля объектов угледобывающего региона”.

Список литературы / References

- [1] **Елизаветин И.В., Шувалов Р.И., Буш В.А.** Принципы и методы радиолокационной съемки для целей формирования цифровой модели местности // Геодезия и картография. 2009. № 1. С. 39–45.
Elizavetin, I.V., Shuvalov, R.I., Bush, V.A. Principles and methods of SAR Interferometry for the purpose of forming a digital elevation model // Geodeziya i Kartografiya. 2009. No. 1. P. 39–45. (In Russ.)
- [2] **Ferretti, A., Monti-Guarnieri, A., Prati, C., Rocca, F., Massonnet, D.** InSAR principles: guidelines for sar interferometry processing and interpretation. ESA Publications, 2007. TM-19. Available at: http://www.esa.int/esapub/tm/tm19/TM-19_ptA.pdf (accessed 02.08.2016).
- [3] **Li, Zh., Bethel, J.** Image coregistration in sar interferometry // The Intern. Archives of the Photogrammetry. Remote Sensing and Spatial Inform. Sci. Vol. XXXVII. Pt B1. Beijing, 2008. P. 433–438.
- [4] **Massonnet, D., Feigl, K.L.** Radar interferometry and its application to changes in the earth’s surface // Reviews of Geophysics. 1998. Vol. 36(4). P. 441–500.
- [5] **Costantini, M., Farina, A., Zirilli, F.** A fast phase unwrapping algorithm for SAR interferometry // IEEE Trans. GARS. 2002. Vol. 37, No. 1. P. 452–460.
- [6] **Mistry, P., Braganza, S., Kaeli, D., Leeser, M.** Accelerating phase unwrapping and affine transformations for optical quadrature microscopy using CUDA // Proc. of 2nd Workshop on General Purpose Proc. on Graphics Proc. Units, GPGPU 2009. USA, Washington, DC: ACM, 2009. P. 28–37.

- [7] **Karasev, P.A., Campbell, D.P., Richards, M.A.** Obtaining a 35x speedup in 2D phase unwrapping using commodity graphics processors // Radar Conference. IEEE, 2007. P. 574–578.
- [8] **Верба В.С., Неронский Л.Б., Осипов И.Г., Турук В.Э.** Радиолокационные системы землеобзора космического базирования. М.: Радиотехника, 2010. 675 с.
Verba, V.S., Neronskiy, L.B., Osipov, I.G., Turuk, V.E. Radar systems of ground-based space survey. Moscow: Radiotekhnika, 2010. 675 p. (In Russ.)
- [9] **Wu, Z., Ma, W., Long, G., Li, Y., Tang, Q., Wang, Z.** High performance two-dimensional phase unwrapping on GPUs // Proc. of the 11th ACM Conf. on Computing Frontiers – CF'14. 2014. New York, USA: ACM, 2014. P. 35:1–35:10.
- [10] **Xin-Liang, S., Xiao-Chun, X.** GPU acceleration of range alignment based on minimum entropy criterion // Radar Conference 2013, IET Intern. 14–16 April 2013. P. 1–4.
- [11] **Guerriero, A., Anelli, V.W., Pagliara, A., Nutricato, R., Nitti, D.O.** High performance GPU implementation of InSAR time-consuming algorithm kernels // Proc. of the 1st Workshop on the State of the art and Challenges of Research Efforts at POLIBA. Bari, Italy: Politecnico di Bari, 2014. 383 p.
- [12] **Zhang, F., Wang, B., Xiang, M.** Accelerating InSAR raw data simulation on GPU using CUDA // Geoscience and Remote Sensing Symp. (IGARSS). 2010 IEEE Intern. 25–30 July 2010. Bari, Italy: Politecnico di Bari, 2010. P. 2932–2935.
- [13] **Marinkovic, P.S., Hanssen, R.F., Kampes, B.M.** Utilization of parallelization algorithms in InSAR/PS-InSAR processing // Proc. of the 2004 Envisat ERS Symp. (ESA SP-572). 6–10 Sept. 2004. Salzburg, Austria: ESA, 2004. P. 1–7.
- [14] **Sheng, G., Qi-Ming, Z., Jian, J., Cun-Ren, L., Qing-xi, T.** Parallel processing of InSAR interferogram filtering with CUDA programming // Science of Surveying and Mapping Engineering. 2015. No. 1. P. 54–68.
- [15] **Gabriel, E., Fagg, G.E., Bosilca, G., Angskun, Th., Dongarra, J.J., Squyres, J.M., Sahay, V., Kambadur, Pr., Barrett, B., Lumsdaine, A., Castain, R.H., Daniel, D.J., Graham, R.L., Woodall, T.S.** Open MPI: Goals, concept, and design of a next generation MPI implementation. Available at: <https://www.open-mpi.org/papers/euro-pvmmpi-2004-overview/euro-pvmmpi-2004-overview.pdf> (accessed 23.09.2014).
- [16] **Kampes, B., Hanssen, R., Perski, Z.** Radar interferometry with public domain tools presentation. Available at: http://doris.tudelft.nl/Literature/kampes_fringe03.pdf (accessed 23.09.2014).
- [17] **Frigo, M., Johnson, S.G.** FFTW: An adaptive software architecture for the FFT // ICASSP Conf. Proc. 15 May 1998. Seattle, Washington, USA: IEEE, 1998. Vol. 3. P. 1381–1384.
- [18] **Larkin, J.** Fast GPU development with CUDA libraries. Available at: https://www.olcf.ornl.gov/wp-content/uploads/2013/02/GPU_libraries-JL.pdf (accessed 23.09.2014).
- [19] **Demmel, J., Dongarra, J.** ST-HEC: Reliable and scalable software for linear algebra computations on high end computers. Available at: <https://people.eecs.berkeley.edu/demmel/Sca-LAPACK-Proposal.pdf> (accessed 23.09.2014).
- [20] **Феокистов А.А., Захаров А.И., Гусев М.А., Денисов П.В.** Исследование возможностей метода малых базовых линий на примере модуля SBaS программного пакета SARscape и данных PCA ASAR/ENVISat и PALSAR/ALOS. Ч. 1. Ключевые моменты метода // Журн. радиоэлектроники. 2015. № 9. С. 1–26.

- Feoktistov, A.A., Zaharov, A.I., Gusev, M.A., Denisov, P.V. Investigation of abilities for the method of small baselines technique on the example of SBaS module of SARscape software package and the data of the RSA ASAR/ENVISat and PALSAR/ALOS. Pt 1. Key points of the method // Zhurn. Radioelektroniki. 2015. No. 9. P. 1–26. (In Russ.)
- [21] Zinno, I., Mossucca, L., Elefante, S., De Luca, C., Casola, V., Terzo, O., Casu, F., Lanari, R. Cloud computing for earth surface deformation analysis via spaceborne radar imaging: a case study // IEEE Trans. Cloud Computing. 2016. No. 4. P. 104–118.
- [22] Zinno, I., Elefante, S., Mossucca, L., De Luca, C., Manunta, M., Terzo, O., Lanari, R., Casu, F. First assessment of the P-SBAS DInSAR algorithm performances within a cloud computing environment // J. of Selected Topics in Applied Earth Observations and Remote Sensing. 2015. Vol. 8. P. 4675–4686.
- [23] Kannan, P. Beyond Hadoop MapReduce Apache Tez and Apache Spark. San Jose State Univ. Available at: <http://www.sjsu.edu/people/robert.chun/courses/CS259Fall2013/s3/F.pdf> (accessed 02.08.2016).
- [24] Nathan P. Real-time analytics with Spark Streaming. Available at: <http://viva-lab.ece.virginia.edu/foswiki/pub/InSAR/RitaEducation/InSAR%20Technology%20Literature%20Search.pdf> (accessed 02.08.2016).
- [25] Nagler, E. Introduction to Oozie. Apache Oozie Documentation. Available at: <http://www.cse.buffalo.edu/~bina/cse487/fall2011/Oozie.pdf> (accessed 02.08.2016).
- [26] Jhajj, R. Apache Hadoop Hue Tutorial. Available at: <https://examples.javacodegeeks.com/enterprise-java/apache-hadoop/apache-hadoop-hue-tutorial/> (accessed 02.08.2016).

Поступила в редакцию 7 марта 2017 г.

Method for SAR data processing based on massively parallel Apache Spark system

POTAPOV, VADIM P., POPOV, SEMEN E., KOSTYLEV, MIKHAIL A.*

Institute of Computational Technologies SB RAS, Novosibirsk, 630090, Russia

*Corresponding author: Kostylev, Mikhail A., e-mail: 5999ft@gmail.com

The paper presents a novel approach to development of distributed software for streaming pre- and post-processing of radar images. A distinctive feature of the system is employment of Apache Spark mass-parallel processing, which allowed operations with general-purpose hardware as well as the ability to execute existing algorithms that are not designed for distributed processing on multiple nodes without changing the implementation of the algorithm.

The paper compares distributed computing technologies and presents a general description of the cluster and the mechanism for performing the tasks of pre- and post-processing of radar data, as well as specifies the implementation of tasks within the framework of the proposed approach. The parallel computations methods based on GPUs, Java-multithreading and the application development based on a massively parallel architecture are used. In particular, in the phase unwrapping algorithm, the

construction of the growth ring, the estimating of absolute phase values and the performance of reliability tests performed on the side of the GPU are presented. The number map of the regions which is monitored and updated after each iteration allows monitoring all regions along the optimal unwrapping path simultaneously and tracking the moments of their encounters and intersections. In the massively parallel algorithm execution, we apply a pairing deployment scheme for the input data in separate HDFS directories. As result of reading of the input data, a Resilient Distributed Datasets object is created. It, consists of pairs which include the file name and the binary data stream.

The binary sequence is passed to the map-function, which executes the phase unwrapping procedure. The implementation of the map-function checks the node hardware and runs the code on the CPU or GPU. The interferometric image is processed entirely locally at the computational node without dividing into parts.

Parallelization is achieved by processing a large number of images on a set of nodes. In conclusion, the test results of cluster execution of the developed algorithms are presented. 7x speedup performance is shown with parallel start of eight computational tasks in the phase unwrapping algorithm. Compared to traditional approaches to radar data processing, in which parallel computations either do not apply, or are used only to improve the performance of calculations, the proposed solution is focused on processing a large number of packet data.

Keywords: Apache Spark, Apache Hadoop, distributed information systems, sar interferometry, processing algorithms.

Acknowledgements. The work was carried out within the framework of the Fundamental Project IV.38.2.8 “Geoinformation technologies and distributed type systems for natural and technogenic monitoring and control of the objects of the coal-mining region”.

Received 7 March 2017