Информационно-вычислительная система массивно-параллельной обработки радарных данных в среде Apache Spark

В. П. Потапов[1], С. Е. Попов[1,*], М. А. Костылев[2]
[1]Институт вычислительных технологий СО РАН, Новосибирск, Россия
[2]ООО "Программные технологии", Новосибирск, Россия
*Контактный e-mail: popov@ict.sbras.ru

Рассмотрена задача создания информационно-вычислительной системы обработки радарных снимков с возможностью визуализации, конфигурирования и запуска алгоритмов основных этапов процессинга интерферометрических данных методом Persistent Scatterer в интеграции с MPP-системой (Massive Parallel Processing) для высокопроизводительного мониторинга смещений земной поверхности участков аэрокосмической съемки. Приведены основные схемы маршрутизации потоков данных исполнения заданий. Представлена программная реализация в виде веб-портала на базе компонентов ReactJS, включая автоматизированную загрузку и обновление базы данных радарных снимков Sentinel-1A посредством технологии RESTful API.

Ключевые слова: мониторинг смещений земной поверхности, радарная интерферометрия, системы с массивно-параллельным исполнением заданий, высокопроизводительная обработка пространственных данных.

Введение

Изображения, получаемые с помощью космических средств дистанционного зондирования Земли, играют исключительно важную роль в научных исследованиях, связанных с мониторингом смещений земной поверхности. Метод дифференциальной радарной интерферометрии незаменим для своевременного выявления сдвигов земной поверхности над районами подземной добычи полезных ископаемых, картирования деформаций бортов и уступов карьеров, а также мониторинга природных и техногенных смещений и деформаций сооружений. Радарная интерферометрия выявляет малейшие смещения (вплоть до нескольких миллиметров), сводит к минимуму риск возникновения чрезвычайных ситуаций и значительно уменьшает их возможные последствия. Основное преимущество радарной интерферометрии — независимая дистанционная оценка изменений по всей площади снимка. Для расчета используется массив спутниковых радарных данных, полученных с периодичностью до восьми раз в месяц [1, 2].

Активное развитие методов дифференциальной интерферометрии и средств дистанционного зондирования требует создания проблемно-ориентированных программных

комплексов обработки больших объемов данных. При этом зачастую основная ценность космической информации, поступающей при мониторинге земной поверхности, заключается в возможности ее оперативной постобработки и анализа результатов. Для получения точных и непротиворечивых результатов требуется исходный массив данных радарных наблюдений, состоящий в среднем из 30 радарных съемок за 30 разных дат. Постобработка может включать в себя повторные этапы (формирование интерферограмм, расчет когерентности и оценку ее значений сигнал/шум и т. п.) для составления корректного временного стека сцен съемки с последующим расчетом методами SBAS или Persistent Scaterers [3, 4]. Таким образом, на отдельных стадиях расчетов может возникать резкая деградация производительности. Экспериментальные расчеты показывают время от 3 до 5 ч для 12 пар снимков небольшого разрешения в 3000×1000 пикселей для выявления динамики вертикальных смещений с погрешностью разности высот цифровой модели рельефа не более чем 3 мм/пиксел.

Мониторинг и анализ геодинамической ситуации отличаются высоким уровнем ответственности и сложности решаемых задач, так как наряду с мощными возмущениями из известных очаговых зон анализировать и классифицировать приходится разнородный поток событий, среди которых промышленные взрывы различной мощности и глубины заложения, горные удары и оползни [5–7].

В настоящее время реализовано большое количество различных систем мониторинга, основанных на данных дистанционного зондирования земли (ДДЗ) [8–13], использующих различные типы и форматы ДДЗ, как мульти- и гиперспектральные, так и радарные данные. Многие из них носят преимущественно информационный характер, включают набор ретроспективных данных и отчеты и по факту не предоставляют интерактивной расчетной части процессинга ДДЗ.

В области комплексной обработки радарных данных наиболее развитым в плане программного обеспечения, набора функционала и доступа к базам данных космоснимков является веб-портал Geohazard Tep [14]. Построенный на базе облачной архитектуры Amazon Web Service (AWS) содержит широкий пул процессинговых сервисов, ориентированных на различные прикладные направления радарной интерферометрии, обеспечивает PaaS-модели (Platform as a Service) облачных вычислений. Однако представленные веб-службы портала не дают реализации именно realtime-обработки в потоковом представлении предметных данных. Сервисы функционируют по модели доступа On-demand Processing Service, бóльшая часть из них использует коммерческое программное обеспечение (ENVI, SARScape и т. п.).

Реализация и широкое внедрение большого количества программных алгоритмов технологических этапов обработки радарных данных показывают целесообразность совместного их применения в инфраструктуре, предоставляющей массивно-параллельное исполнение расчетных заданий, где программный каркас (фреймворк) такой инфраструктуры выступает как интегратор распределенного исполнения программного кода на данных, получаемых в потоковом режиме, что является актуальной задачей современной радарной интерферометрии.

Требуется разработать информационно-вычислительную систему полного цикла процессинга радарных снимков (методом Persistent Scatterer) для мониторинга смещений участков земной поверхности. Мониторинг должен производиться с применением аэрокосмической съемки и возможностью массивно-параллельного исполнения расчетных заданий в потоке поступающих предметных данных, которое является основной функциональной характеристикой в парадигме распределенных технологий.

1. Концепция системы мониторинга смещений земной поверхности

По сравнению с традиционными подходами к обработке радарных данных (SNAP, ENVI, и т. д.), при которых высокопроизводительные вычисления не применяются, а оптимизация программной составляющей алгоритмов достигается за счет использования стандартных библиотек параллельных вычислений, предлагаемое концептуальное решение (рис. 1) ориентировано как на использование собственных расчетных пакетов модулей, так и на привлечение сторонних разработок за счет гибкой программной инфраструктуры кластера (Apache Spark), позволяющего использовать изолированные контейнеры объектов с возможностью запуска в среде JVM.

Анализ различных технологий параллельных, распределенных и облачных вычислений [14–26] показал, что на сегодняшний день стандартом прикладной разработки, в том числе и в области геоинформатики, являются программные каркасы (API) компонентов массивно-параллельной архитектуры на базе экосистемы Apache Hadoop. Данная архитектура относится к классу SN-систем, она предполагает модель разделения ресурсов, когда у каждого вычислительного узла свои собственная оперативная память, дисковые массивы и процессорные единицы. Для сравнения взяты различные реализации компо-

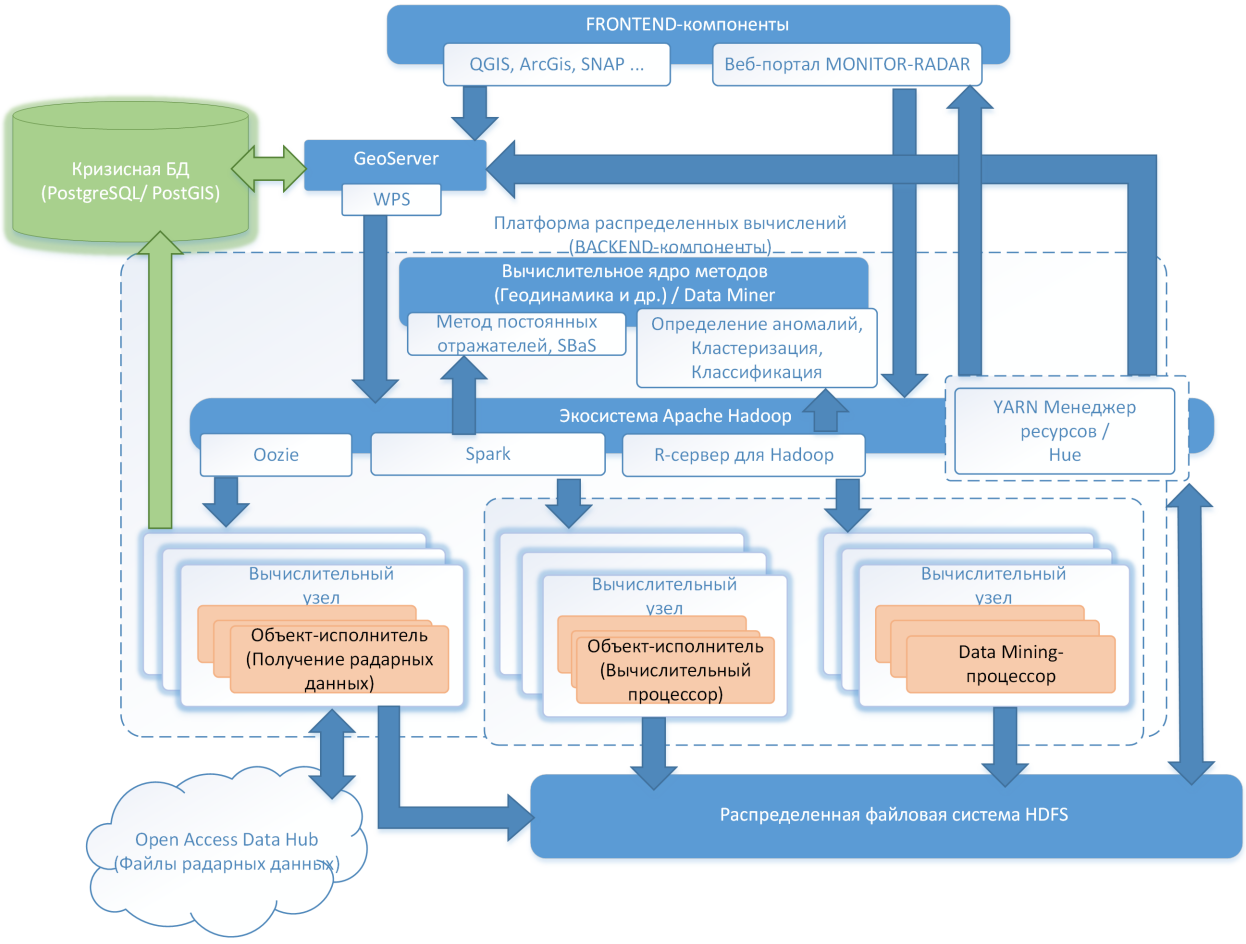

Рис. 1. Концептуальная модель распределенного программного комплекса мониторинга геодинамической ситуации на базе модели расчета смещений

нентов экосистемы Apache Ecosystem и рассмотрены с точки зрения их программных свойств применительно к задачам обработки радарных данных (табл. 1).

В плане выбранного подхода, где каждое задание является изолированным контейнером вычислений над распределенным набором данных, определяемых значениями точек одного или нескольких радарных снимков, преимущество отдается компоненту с поддержкой пакетной модели (batch-proccessing framework). Проведенный анализ позволил установить, что данному условию удовлетворяют компоненты Spark, Tez и Flink.

Каждый из выбранных компонентов имеет свои преимущества, однако в пользу выбора фреймворка Apache Spark указывают: 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 построить алгоритмы потоковой обработки радарных данных; большое количество вспомогательных программных решений, необходимых для организации системы (с системами управления контентом), и их постоянная поддержка разработчиками; поддержка Data Mining-функционала на базе компонентов R-Server и GraphX [21–23, 27–29].

Таким образом, в рамках выбранной парадигмы распределенного программирования Apache API разрабатываемая информационно-вычислительная система логически может быть представлена такими двумя составляющими, как графическая часть (FRONTEND) и вычислительное ядро с массивно-параллельным функционалом (BACKEND). Кроме стандартных требований, предъявляемых к клиент-серверным распределенным архитектурам (прозрачность, открытость, масштабируемость, аудит, логирование и др.), необходимо указать специфические в контексте имплементации методов обработки радарных данных и функциональности разрабатываемой системы. К ним относятся:

- Запуск, процессинг и корректное завершение заданий в массивно-параллельном стиле для многопользовательских запросов, в том числе в потоковом режиме и с использованием стандартов спецификации Web Processing Service (WPS).
- Автоматическая маршрутизация вычислительных потоков SN-системы в пуле поступающих заданий.

Т а б л и ц а 1. Компоненты экосистемы Apache Distributed Programming

Свойство компонента	Spark	Ignite	Tez	Flink	Apex	Storm	Beam
Возможность кеширования данных и хранения промежуточных результатов вычислений в оперативной памяти	+	+	–	+	+	+	+
Поддержка потоков	+	+	–	+	+	+	+
Пакетная система обработки данных	+	–	+	+	–	–	–
Поддержка распределенной файловой системы	+	+	+	+	+	+	–
Изолированные JVM-контейнеры	+	–	+	+	–	–	–
Создание вычислительных контекстных объектов посредством RESTful-запросов	+	–	+	+	–	–	–

- Автоматическое разделение заданий на основе аппаратной конфигурации кластера по узлам системы, их идентификация и логирование процесса выполнения. Поддержка возможности указания количества требуемых ресурсов (CPU Cores, JMV memory) для конкретных заданий, запускаемых пользователем.
- Поддержка распределенной файловой системы, доступной со всех узлов.
- Возможность комплексного управления заданиями в удаленном режиме посредством RESTful-запросов через протокол HTTP.
- Поддержка компонентной модели структуры графических элементов интерактивного пользовательского интерфейса (WebGUI). Представление и взаимодействие с радарными данными посредством электронной карты, таблицы параметров и методов, составляющих backend, базы данных космоснимков. Настройка графических объектов в соответствии с профилем аутентифицированного пользователя.

2. Модели и схемы маршрутизации потоков

Согласно сформулированным требованиям разрабатываемая система поддерживает следующий режим работы: пользователь FRONTEND-приложения задает параметры требуемого метода расчетной схемы (рис. 2) и запроса SELECT к базе данных космоснимков.

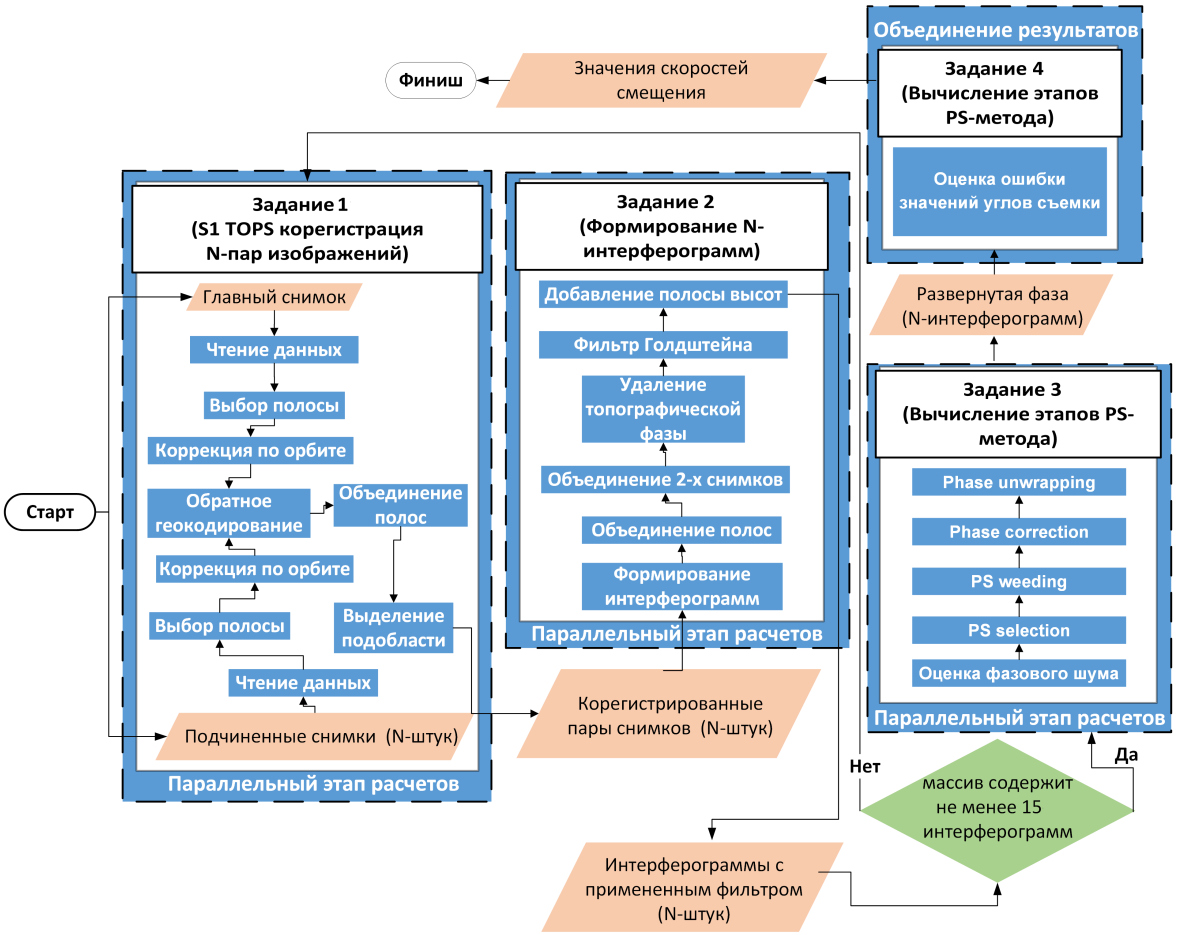

Рис. 2. Блок-схема полного процесса постобработки радарных снимков и расчета карты смещений земной поверхности методом Persistent Scaterers

Все заданные значения метода сохраняются в конфигурационном файле в распределенной файловой системе HDFS. Пользователь посредством HTTPS-протокола отправляет POST-запрос менеджеру заданий Spark на запуск соответствующего расчетного метода, получает уведомление, содержащее уникальный идентификационный номер задания, по которому впоследствии система ассоциирует аутентифицированного пользователя со своим заданием (Task). Расчетный метод исполняется системой Spark как независимый процесс на кластере, координируемый объектом SparkContext в основной программе, называемой программой-драйвером (Driver Program) (рис. 3). Для запуска на кластере SparkContext подключается к объекту Resource Manager, который распределяет ресурсы между приложениями. После подключения Spark Resource Manager инициализирует объект Executor на свободном узле кластера (Worker Node). Executor является процессом, который запускает вычисления и хранит данные для пользова-

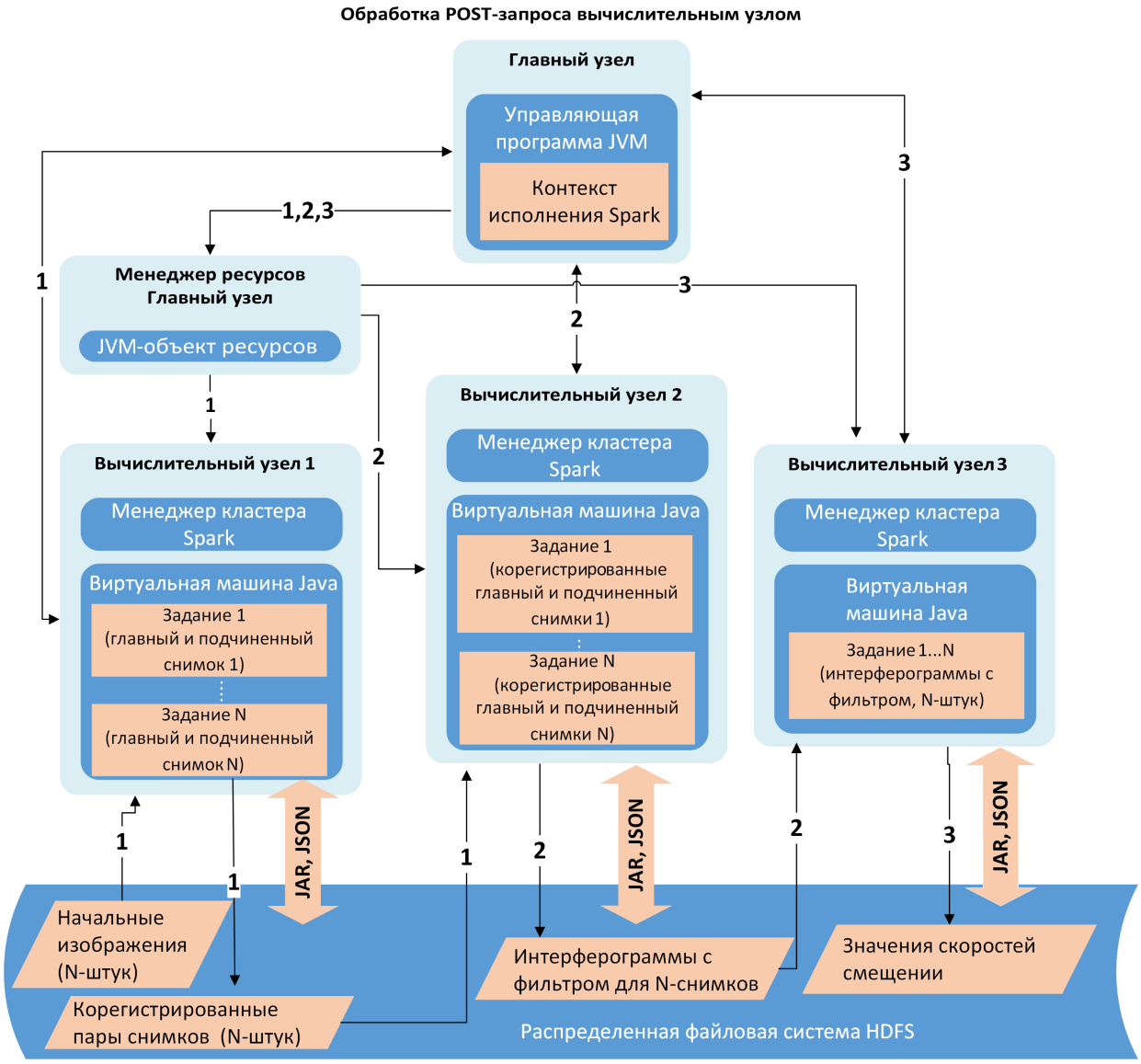

Рис. 3. Схема маршрутизации потоков данных исполнения задания (BACKEND) в кластерной архитектуре Apache Spark

тельского приложения. После создания Executor система Spark отправляет ему Java-код расчетного метода (JAR-файл), переданный объектом SparkContext. SparkContext отправляет задание (Task) объекту Executor для запуска JAR-файла в виртуальной машине Java (JVM, изолированный JVM-контейнер). Максимальное количество заданий на один объект Executor определяется параметром spark.executor.cores (см. рис. 1) в конфигурационном JSON-файле каждого расчетного модуля, равно как и другие параметры, передаваемые POST-запросом менеджеру заданий Spark.

Каждое задание получает свои собственные процессы-исполнители (Executors), которые остаются активными на все время жизни приложения (JAR-файла) и запускают задания (Task) в нескольких потоках. Это дает преимущество изолировать приложения друг от друга как на стороне драйвера, так и на стороне исполнителя (Executors из разных приложений выполняются в разных JVM).

Таким образом, согласно блок-схеме (см. рис. 2) задание Task 1 может быть запущено в стиле массивно-параллельного исполнения установкой параметра spark.executor.cores равному количеству пар, образуемых MASTER- и SLAVE-снимками (например, исходя из условия не менее 15 интерферограмм). Аналогичным способом реализуется запуск задания Task 2 (рис. 3). Следовательно, подобная схема исполнения расчетных методов блок-схемы дает возможность модифицировать функциональность из последовательного в частично параллельный вариант исполнения, значительно сократив время работы всего алгоритма построения карты смещений земной поверхности.

FRONTEND-составляющая разрабатываемой системы построена с применением технологий React (библиотека для создания компонентов графического интерфейса) и Redux (фреймфорк для управления состоянием приложения). В качестве среды выполнения используется платформа NodeJS. Архитектура веб-приложения основана на парадигме однонаправленного потока данных (Flux). Взаимодействие с распределенной файловой системой HDFS, Apache Spark и базой данных метаописаний космоснимков PostGIS происходит через интерфейсы REST API.

Распределенная файловая система используется как для хранения обрабатываемых в ней данных, так и для размещения вычислительных модулей отдельных этапов процессинга. Каждый модуль представлен в виде JAR-файла, исполняемого в системе Spark, а также конфигурационного файла в формате JSON. Веб-приложение поддерживает функцию аутентификации посредством API-модуля Apache Hue. Аутентифицированные пользователи получают доступ к конфигурациям вычислительных модулей из распределенной файловой системы, дифференцируемых по профилю пользователя. На основе файлов конфигураций в веб-интерфейсе создаются соответствующие элементы для настройки и запуска алгоритмов.

Запуск заданий обработки выполняется при помощи POST-запроса к Spark REST API с передачей выбранных пользователем параметров (обрабатываемые изображения, координаты территории и т. д.). Предоставляется возможность мониторинга исполняемых заданий, а также просмотра результатов обработки и их дальнейшего использования (рис. 4). Новизна данного подхода заключается в возможности хранения всех необходимых для работы приложения данных в едином хранилище в виде дерева объектов, а также описания всех возможных действий в системе и их влияния на текущее состояние. Компоненты графического веб-интерфейса создаются как функция от состояния, которая возвращает заданное представление. Такой подход позволяет отделить логику работы системы от ее отображения, упростить внесение изменений и дальнейшее масштабирование (рис. 4).

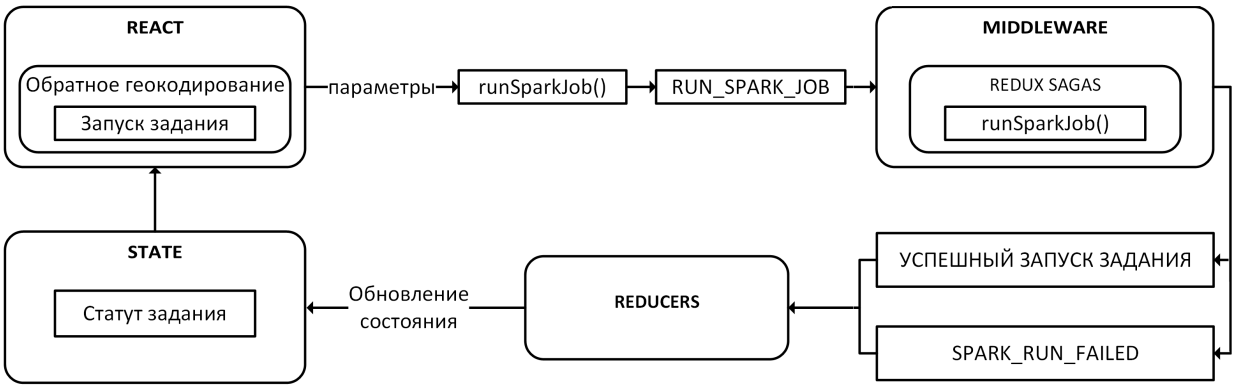

Рис. 4. Схема взаимодействия компонентов FRONTEND и BACKEND в процедурах запуска заданий

Полученное Spark REST API-задание обрабатывается на главном узле кластера. В теле запроса обязательным параметром является путь к JAR-файлу с кодом задания для выполнения, для запуска переданного файла на главном узле создается виртуальная машина Java. На рис. 5 представлен процесс исполнения задания Spark на примере алгоритма Back Geocoding. После запуска JAR-файла задания формируется объект конфигурации задания SparkConf на основе переданных параметров, а затем происходит инициализация контекста исполнения SparkContext.

На следующем этапе создается распределенный набор данных (RDD), состоящий из пар радарных снимков для обработки. Для каждой пары снимков применяется расчетная функция алгоритма как аргумент метода Map, созданного RDD. Переданная функция выполняется на отдельном узле кластера и может выполняться параллельно для нескольких пар при наличии свободных узлов в кластере. На данном этапе осуществляются получение данных исходных снимков по переданной ссылке из распределенной файловой системы, расчет и сохранение результатов. После обработки всех пар снимков происходит закрытие контекста исполнения методом Close объекта SparkContext.

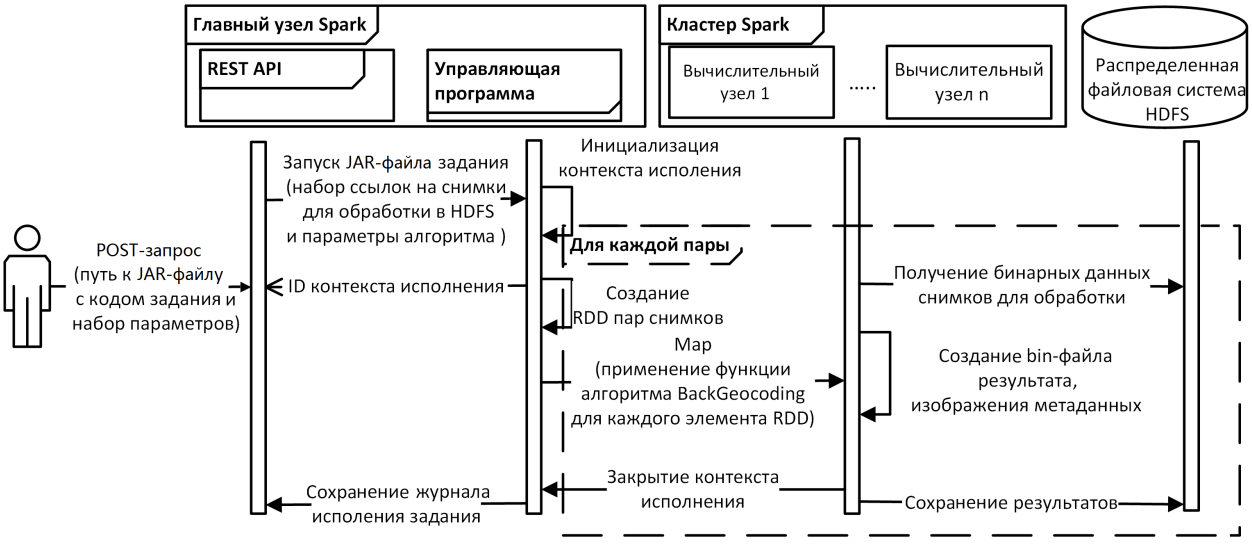

Рис. 5. Процесс обработки исходных радарных снимков методом Back Geocoding

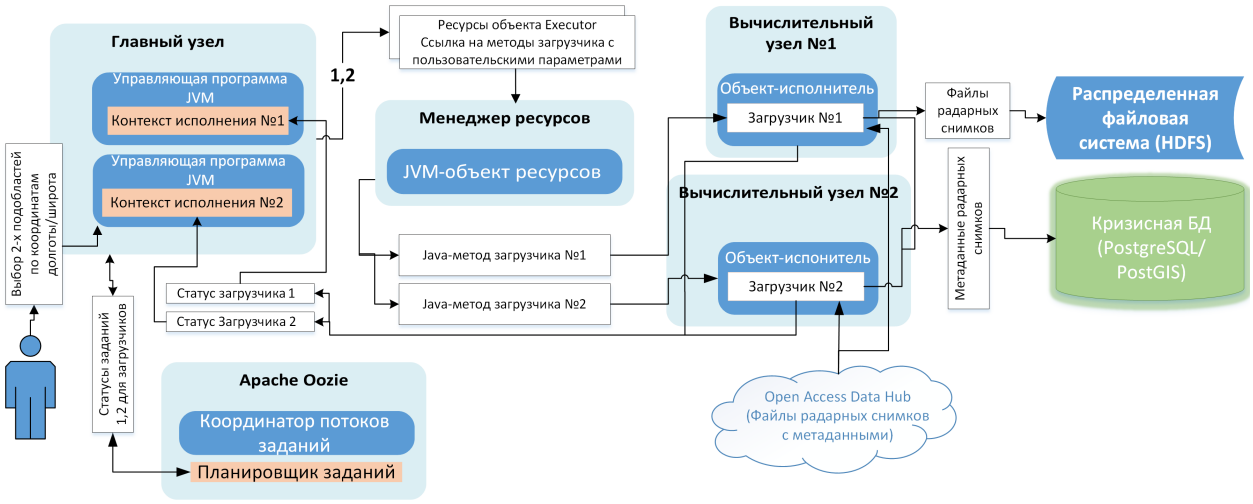

Рис. 6. Схема маршрутизации информационных потоков в процессе исполнения задания на получение радарных данных из открытых источников (Data-Hub)

Возможность инициализации потока поступающих радарных снимков исследуемой территории посредством координаторов заданий в среде Apache Hue и комплексного управления ими в удаленном режиме на основе RESTful-запросов через протокол HTTP реализуется по схеме, приведенной на рис. 5. Каждый отдельный элемент координатора рабочего процесса (workflow) позволяет запустить spark-задание на любом из узлов, доступных в данный момент времени. Метаданные каждого нового снимка размещаются в базе данных в виде записей в таблице, содержащей основные идентифицирующие элементы (id, product, swath, burst, geo coordinates, orbit и пр.). Отметим, полного скачивания снимка не происходит, для расчетов в схеме (см. рис. 3) используются только те фрагменты (burst), которые указаны в расчетном задании согласно выбранной географической территории. Расчетный инкапсулированный модуль в задании инициирует скачивание необходимой полосы (swath) и размещает ее в распределенной файловой системе HDFS, тем самым уменьшая как общий объем хранимой информации, так и время повторного использования данных другими расчетными заданиями. Любая метаинформация по текущему снимку может быть получена посредством RESTful-запроса к источнику данных Data-Hub (рис. 6), обработана и передана следующим модулям в виде параметров расчета по схеме, представленной на рис. 2.

3. Программный прототип информационно-вычислительной системы

На основе предложенной концепции реализован прототип системы полного цикла обработки радарных данных (методом Persistent Scatters) для задачи мониторинга смещений земной поверхности с применением массивно-параллельного подхода.

Система поддерживает радарные изображения, поступающие с космических аппаратов миссий Sentinel-1A и Cosmo-SkyMed. Доступ к данным Sentinel-1A осуществляется через открытый ресурс Copernicus Open Access Hub (OAHub)[1] посредством RESTful-

[1]https://scihub.copernicus.eu/userguide/WebHome

Т а б л и ц а 2. Характеристика компонентов Apache Distributed Programming

Метод	Время расчета, мин			
	Разработанная система		ESA SNAP Toolbox	
	Количество пар снимков			
	1	4	1	4
S1 TOPS Coregistration	4.5	5.3	4.86	17.2
Deburst	13	12	16.9	61.47
Interferogram Formation	1.43	1.73	1.71	9.45
TopoPhaseRemoval	0.56	0.57	0.53	4.68
	Параметры метода			
S1 TOPS Coregistration	Swath: IW1, Burst range: 1–4, DEM Resampling Method: NEAREST NEIGHBOUR, Resampling Type: NEAREST NEIGHBOUR			
Deburst	—			
Interferogram Formation	Orbit Interpolation degree: 3 Coherence Range/Azimuth Window Size: 10/10			
TopoPhaseRemoval	Orbit Interpolation degree: 3 Digital Elevation Model: SRTM3sec Tile Extension: 100 % Output Elevation Band: truel			

запросов согласно пользовательским параметрам региона интересов (ROI — Region of Interest). В зависимости от режима работы системы доступны опция выбора из базы данных (PostGIS) предварительно загруженных снимков либо загрузка по расписанию из OAHub.

Проведено тестирование прототипа системы с использованием методов расчетной схемы для построения карты смещений земной поверхности (см. рис. 2). Результаты тестирования приведены в табл. 2.

Заключение

В результате анализа различных подходов, применяемых при обработке радарных данных, а также обзора технологий распределенных вычислений была предложена и реализована распределенная информационно-вычислительная система на базе архитектуры массивно-параллельного исполнения заданий экосистемы Apache Hadoop (компонент Apache Spark) для потоковой постобработки радарных снимков и построения карты смещений. Программная реализация содержит многофункциональный веб-интерфейс, позволяющий пользователю взаимодействовать с кластером, получая доступ к распределенной файловой системе HDFS, взаимодействовать с открытыми ресурсами космоснимков посредством RESTful API, создавать и исполнять существующие задания, ориентируясь на схемы полного цикла процессинга интерферометрических данных, методом Persistent Scatterer.

Разработанный прототип системы ориентирован как на использование собственных расчетных пакетов модулей, так и на привлечение сторонних разработок за счет гибкой программной инфраструктуры кластера Spark, позволяющего применять изолированные контейнеры объектов Executors при запуске в среде JVM.

Новизна предложенного решения заключается в возможности взаимодействия разработанных алгоритмов на основе изолированного контекстного запуска заданий с данными в HDFS во время процедуры подготовки и на этапе полного цикла расчета смещений методом Persistent Scatters, где впервые применен интегральный подход к разработке масштабируемых FRONTEND- и BACKEND-составляющих программного комплекса на базе компонентов ReactJS+Redux и фрэймворк Apache Spark API, а также запуска заданий на основе RESTful API с поддержкой стандарта спецификации WPS, что позволяет использовать предлагаемое решение в практически любой ГИС, поддерживающей данный стандарт.

Результаты оценки научно-технического уровня рассматриваемой исследовательской работы показали высокие характеристики в плане производительности разработанной системы с сохранением требуемой точности результатов. В частности, адаптированные и интегрированные в систему Apache Spark модули прототипа системы и программного комплекса ESA SNAP Toolbox возвращали идентичные массивы обработанных интерферометрических данных в попиксельном сравнении при скорости работы первых в несколько раз быстрее.

Предлагаемое комплексное решение, веб-портал и MPP-кластер могут быть развернуты на большом количестве узлов с гибридной аппаратной архитектурой, не требующих дорогостоящих систем хранения данных и вычислительных серверов, за счет применения распределенной файловой системы и менеджера ресурсов отдельно функционирующих рабочих узлов.

Список литературы / References

[1] **Бондур В.Г., Савин А.И.** Концепция создания систем мониторинга окружающей среды в экологических и природно-ресурсных целях // Исследование Земли из космоса. 1992. № 6. С. 70–78.
Bondur, V.G., Savin, A.I. The concept for creating of environmental monitoring systems for environmental and natural resources purposes // Issledovanie Zemli iz Kosmosa. 1992. No. 6. P. 70–78. (In Russ.)

[2] **Кантемиров Ю.И.** Космический радарный мониторинг смещений и деформаций земной поверхности и сооружений. Опыт компании “СОФЗОНД” // Вестн. СибГАУ. 2013. № 5(51). С. 52–54.
Kantemirov, Yu.I. Space radar monitoring of displacements and deformations of the Earth's surface and structures // Scientific J. of Science and Technology. 2013. No. 5(51). P. 52–54. (In Russ.)

[3] Sbas Tutorial. Available at: `http://sarmap.ch/tutorials/sbas_tutorial_V_2_0.pdf` (accessed 12.02.2016)

[4] **Sousaa, J.J., Hooperc, J.A., Hanssenc, R.F. et al.** Persistent Scatterer InSAR: A comparison of methodologies based on a model of temporal deformation vs. spatial correlation selection criteria // Remote Sensing of Environment. 2011. Vol. 115, No. 10. P. 2652–2663.

[5] Деструкция земной коры и процессы самоорганизации в областях сильного техногенного воздействия / В.Н. Опарин, А.Д. Сашурин, А.В. Леонтьев и др. Новосибирск: Изд-во СО РАН, 2012. 632 с.
Destruction of the Earth's crust and the processes of self-organization in the areas of strong man-made impact / V.N. Oparin, A.D. Sashurin, A.V. Leontiev et al. Novosibirsk: Izd-vo SO RAN, 2012. 632 p. (In Russ.)

[6] **Адушкин В.В., Опарин В.Н.** От явления знакопеременной реакции горных пород на динамические воздействия — к волнам маятникового типа в напряженных геосредах. Ч. I // Физ.-техн. пробл. разработки полезных ископаемых. 2012. № 2. С. 3–28.
Adushkin, V.V., Oparin, V.N. From the alternating-sign explosion response of rocks to the pendulum waves in stressed geomedia. Pt I // J. of Mining Sci. 2012. Vol. 48, iss. 2. P. 203–222.

[7] **Адушкин В.В., Опарин В.Н.** От явления знакопеременной реакции горных пород на динамические воздействия — к волнам маятникового типа в напряженных геосредах. Ч. II // Физ.-техн. пробл. разработки полезных ископаемых. 2013. № 2. С. 3–46.
Adushkin, V.V., Oparin, V.N. From the alternating-sign explosion response of rocks to the pendulum waves in stressed geomedia. Pt I // J. of Mining Sci. 2013. Vol. 49, iss. 2. P. 175–209.

[8] **Simmons, A.D., Kerekes, J.P., Raqueno, N.G.** Hyperspectral monitoring of chemically sensitive plant sentinels // Proc. SPIE 7457. Imaging Spectrometry XIV, 74570G, San Diego, CA, 2003. P. 45–51.

[9] **Лупян Е.А., Савин И.Ю., Барталев С.А. и др.** Спутниковый сервис мониторинга состояния растительности ("Вега") // Соврем. пробл. дистанц. зонд. Земли из космоса. 2011. Т. 8, № 1. С. 190–198.
Lupyan, E.A., Savin, I.Yu., Bartalev, S.A. et al. Satellite monitoring service for vegetation ("Vega") // Sovrem. Probl. Distants. Zond. Zemli iz Kosmosa. 2011. Vol. 8, No. 1. P. 190–198. (In Russ.)

[10] **Lavrova, O.Yu., Loupian, E.A., Mityagina, M.I. et al.** See the sea — multi-user information system ocean processes investigations based on satellite remote sensing data // Bollettino di Geofisica Teorica ed Applicata. 2013. Vol. 54. P. 146–147.

[11] **Гордеев Е.И., Гирина О.А., Лупян Е.А. и др.** Изучение продуктов извержений вулканов Камчатки с помощью гиперспектральных спутниковых данных в информационной системе VolSatView // Соврем. пробл. дистанц. зонд. Земли из космоса. 2015. Т. 12, № 1. С. 113–128.
Gordeev E.I., Girina O.A., Lupyan E.A. et al. Studies of Kamchatka volcanic eruptions products using hyperspectral satellite data in VolSatView information system // Sovrem. Probl. Distants. Zond. Zemli iz Kosmosa. 2015. Vol. 12, No. 1. P. 113–128. (In Russ.)

[12] **Лупян Е.А., Барталев С.А., Ершов Д.В. и др.** Организация работы со спутниковыми данными в информационной системе дистанционного мониторинга лесных пожаров Федерального агентства лесного хозяйства (ИСДМ-Рослесхоз) // Соврем. пробл. дистанц. зонд. Земли из космоса. 2015. Т. 12, № 5. С. 222–250.
Lupyan, E.A., Bartalev, S.A., Ershov, D.V. et al. Organization of work with satellite data in the information system for remote monitoring of forest fires of the Federal Forestry Agency (ISDM-Rosleskhoz) // Sovrem. Probl. Distants. Zond. Zemli iz Kosmosa. 2015. Vol. 12, No. 5. P. 222–250. (In Russ.)

[13] **Takeuchi, S., Yamada, H.** Monitoring of forest fire damage by using JERS-1 InSAR // Geosci. and Remote Sensing Symp. (IGARSS '02). Toronto, Ontario, Canada. 2002. P. 3290–3292.

[14] Geohazards Tep — Geobrowser. Available at: `https://geohazards-tep.eo.esa.int/` (accessed 31.05.2017)

[15] **Маклин С., Нафтел Дж., Уильямс К.** Microsoft .NET Remoting: Пер. с англ. М.: Торгово-изд. дом "Русская редакция", 2003. 384 с.
McLean, S., Naftel, J., Williams, K. Microsoft .NET Remoting. 1st ed. N.Y.: Microsoft Press, 2003. 336 p.

[16] **Berman, F., Wolski, R.** Application-level scheduling on distributed heterogeneous networks // Supercomputing: Proc. of the ACM/IEEE Conf. Pittsburgh, Pennsylvania USA, May 25–28, 1996. IEEE Comput. Soc., 1996. P. 1–28.

[17] **Maheswaran, M., Ali, S.** Dynamic matching and scheduling of a class of independent tasks onto heterogeneous computing systems // J. of Parallel and Distributed Comput. 1999. Vol. 59, No. 2. P. 107–131.

[18] **Laszewski, G., Foster, I. et al.** CoG Kits: A bridge between commodity distributed computing and high-performance grids // Proc. of the ACM Java Grande 2000 Conf., San Francisco, CA, USA, June 3–5, 2000. ACM Press, P. 97–106.

[19] **Yang, T., Gerasoulis, A.** DSC: Scheduling parallel tasks on an unbounded number of processors // IEEE Transact. on Parallel and Distributed Syst. 1994. Vol. 5, No. 9. P. 951–967.

[20] **Qusay, H.M.** Distributed Java programming with RMI and CORBA // Oracle Techn. Network. `http://www.oracle.com/technetwork/articles/javase/rmi-corba-136641.html` (accessed 12.09.2017)

[21] **Reyes-Ortiz, J.L., Oneto, L., Anguita, D.** Big data analytics in the cloud: Spark on Hadoop vs MPI/OpenMPI // INNS Conf. on Big Data 2015: Conf. Proc. San Francisco, USA, 8–10 August, 2015. ACM Press. P. 121–130.

[22] **Mavridis, I., Karatza, H.** Performance evaluation of cloud-based log file analysis with Apache Hadoop and Apache Spark // J. of Syst. and Software. 2017. Vol. 125. P. 133–151.

[23] **Polato, I., Reginald, R., Goldman, A., Kon, F.** A comprehensive view of Hadoop research — a systematic literature review // J. of Network and Comput. Appl. 2014. Vol. 46. P. 1–25.

[24] **Chen, Xu.** Big data analytic frameworks for GIS (Amazon EC2, Hadoop, Spark) // Comprehensive Geographic Inform. Syst. 2017. Vol. 1. P. 148–152.

[25] **Verbesselt, J.** Big Data: Techniques and Technologies in Geoinformatics // Intern. J. of Appl. Earth Observation and Geoinform. 2015. Vol. 35. Pt B. P. 368–369.

[26] **Yang Chaowei, Yu Manzhu, Hu Fei, Jiang Yongyao, Li Yun.** Utilizing cloud computing to address big geospatial data challenges // Computers, Environment and Urban Syst. 2017. Vol. 61. Pt B. P. 120–128.

[27] Hadoop, Storm, Samza, Spark, and Flink: Big data frameworks compared // Digital Ocean. `https://www.digitalocean.com/community/tutorials/hadoop-storm-samza-spark-and-flink-big-data-frameworks-compared` (accessed 12.09.2017)

[28] Spark vs. Tez: What's the Difference? Xplenty. `https://www.xplenty.com/blog/2015/01/apache-spark-vs-tez-comparison/` (accessed 12.09.2017)

[29] Feature wise comparison between Apache Hadoop vs Spark vs Flink // TheServerSide. `http://www.theserverside.com/blog/Coffee-Talk-Java-News-Stories-and-Opinions/Feature-wise-comparison-between-Apache-Hadoop-vs-Spark-vs-Flink` (accessed 12.09.2017)

Поступила в редакцию 6 декабря 2017 г.,
с доработки — 24 апреля 2018 г.

The information and computational system for the massive parallel processing of radar data based on Apache Spark framework

Potapov, Vadim P.[1], Popov, Semen E.[1,*], Kostylev, Mikhail A.[2]

[1]Institute of Computational Technologies SB RAS, Novosibirsk, 630090, Russia
[2]Software Technologies, Novosibirsk, 630090, Russia
[*]Corresponding author: Popov, Semen E., e-mail: popov@ict.sbras.ru

The aim of the presented work is the development of an information computational system for processing radar images with the ability to visualize, configure and run algorithms for the main stages of processing interferometric data by the Persistent Scatterer method integrated with the MPP system (massive parallel processing) for high-performance monitoring of the Earth surface displacement of aerospace survey sites.

As a result of the analysis of the different approaches used in the processing of radar data and the review of distributed computing technologies, a distributed information system based on the architecture of massively parallel execution of the Apache Hadoop ecosystem processes the streaming post-processing of radar images and the construction of a displacement map was proposed and implemented. A software implementation is presented in the form of a web portal based on ReactJS components, including automated downloading and updating of the Sentinel-1A radar image database using RESTful API technology.

The innovation of suggested solution consists of the model of the interaction between developed processing modules based on the isolated execution context with HDFS data storage during the preparing procedure and the complete cycle for the processing of the Earth surface displacement.

An integrated approach to the developing scalable front-end and back-end software complex components with the use of ReactJS, Redux and Apache Spark framework was used for the first time. Supporting of WPS specification makes it possible using almost any GIS, which works with this standard.

The evaluation of a scientific and technological level of research shows high performance of the developed system while maintaining the results quality. In particular, the adapted and integrated ESA SNAP Toolbox returned identical arrays of processed interferometric data in the per-pixel comparison but the speed of the procedure is several times faster.

Keywords: monitoring of Earth surface displacements, radar interferometry, systems with massively parallel execution of tasks, high-performance processing of spatial data.

Received 6 December 2017
Received in revised form 24 April 2018