Санкт-Петербургский государственный университет
Кафедра компьютерного моделирования и
многопроцессорных систем

Свешникова Светлана

Выпускная квалификационная работа бакалавра

Разработка инструментария для работы с данными реанализа гидрометеорологических данных с использованием параллельных примитивов функционального программирования

Направление 010300
«Фундаментальная информатика и информационные технологии»

Научный руководитель
д.т.н. проф. Дегтярев А.Б.

Санкт-Петербург
2016

Содержание

Введение

Данная работа посвящена использованию данных батиметрических исследований [1, 2] для задачи моделирования цунами в распределённой вычислительной среде. Батиметрия изучает рельеф подводной части водных бассейнов (мировой океан, реки, озера и т.д). На основе полученных данных составляется батиметрические карты или карты глубин.

Первая «Генеральная батиметрическая карта океанов» была издана в 1899 году. Она содержала всего 18 400 замеров глубин. Рельеф глубоководных областей был изображен лишь в общих чертах на основе около 7000 измерений.

Следующим шагом к повышению точности батиметрических карт стало появление эхолотов-самописцев в 1935 году. Теперь вместо дискретных значений глубин, научному обществу стали доступны непрерывные данные о глубинах на протяжении всего курса корабля. Это помогло значительно улучшить знания о рельефе дна в популярных для судовождения районах. В других же районах, напротив, маршруты кораблей пролегали достаточно разреженно. «Белые» (необследованные) пятна составляли до 500 км в диаметре. В конце 70-х годов появились многолучевые эхолоты. Такой прибор посылает не единичные звуковые лучи, а сразу пучок, состоящий из десятков или сотен сигналов. Сигналы расходятся веером, перпендикулярно курсу корабля и вместо одиночной линии зондирования получается полоса шириной от 3 до 7 глубин.

Глобальное заполнение «белых пятен» на батиметрических картах стало возможно благодаря появлению спутниковой альтиметрии. Принцип работы спутникового альтиметра показан на Рис. 1. Системы спутниковой

альтиметрии включают в себя радар для измерения высоты спутника над земной поверхностью и систему слежения для определения высоты спутника в геоцентрической системе координат. Альтиметры измеряют расстояние от спутника, до морской поверхности. Так как форма морской поверхности взаимосвязана с формой морского дна, то из альтиметрических данных возможно получение сведений о подводном рельефе.

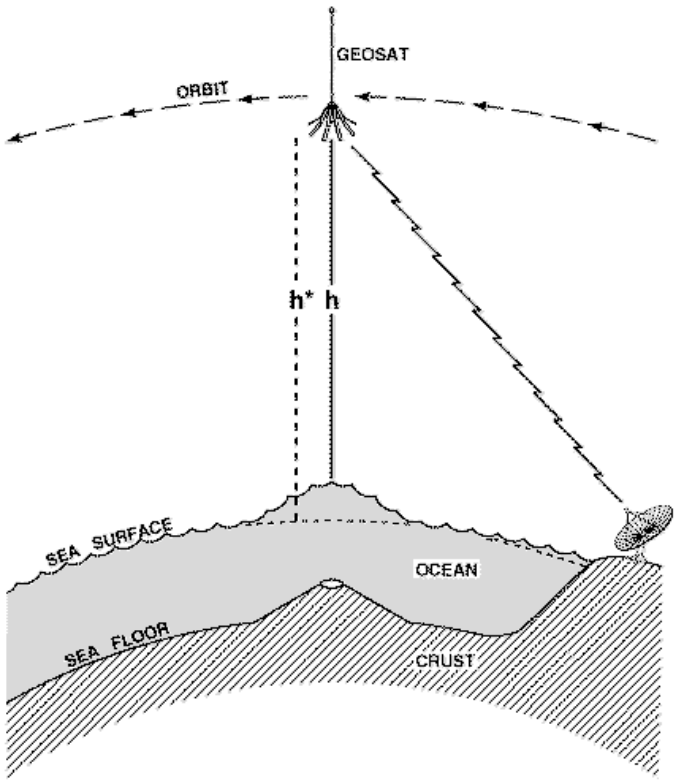

Рис. 1: Схема работы спутникового альтиметра

Рассмотрим подробнее, как осуществляется измерение эхолотами и спутниковыми альтиметрами.

Эхолоты (эхолокаторы). Эхолот состоит из передатчика, преобразователя, приемника и экрана. Передатчик с определенной частотой генерирует электрический импульс, который преобразователь преобразует в

звуковую волну и посылает в воду. Звуковая волна, ударяясь о дно, отражается и движется в обратном направлении, улавливаясь преобразователем. Преобразователь преобразует отраженную звуковую волну обратно в электрический импульс, усиливая с помощью приемника. Таким образом, зная скорость звука в воде и время между передачей сигнала и получением ответного эха, можем вычислить глубину по формуле $h = \frac{v*t}{2}$ (v – скорость звука в воде – 1500м/с, t – время прохождения сигнала). Данный метод хорош, но имеет некоторые недостатки:

1. Скорость звука в воде не является константной величиной. Она зависит от температуры и солёности воды, гидростатического волнения и меняется примерно на 4 процента в большую либо меньшую сторону [3]. Таким образом, для получения точных данных необходимо знать характеристики воды в каждой точке замера, что является достаточно трудоемкой задачей.

2. При большом скоплении зоопланктона или косяков рыб, звуковая волна может отразиться от них, не достигнув дна, и мы получим неверные данные.

3. Ошибки навигации. Спутниковая навигация, обеспечивающая достаточно хорошую точность определения местоположения судна, стала появляться лишь в 60-х годах XX века. Следовательно, полученные эхолотом в некоторой точке измерения могли быть неточно нанесены на карту. В некоторых регионах эти ошибки могли составлять десятки километров.

4. Неравномерное распределение данных. Данные есть в тех районах, где проходят маршруты кораблей. Чем плотнее сеть маршрутов в некотором районе, тем лучше он картографирован. Но есть районы океана размером до 500 километров, необследованные эхолокатора-

ми.

Спутниковые альтиметры. Система спутниковой альтиметрии включает в себя радар для измерения высоты спутника над земной поверхностью и систему слежения для определений высоты спутника в геоцентрической системе координат. Таким образом система измеряет расстояние от центра масс Земли до уровня моря, получая, тем самым форму морской поверхности. Возникает вопрос: как знание формы морской поверхности помогает составлению батиметрических карт? Есть такое понятие как «уровневая поверхность» — морская поверхность в спокойном состояни. По физическим законам, вектор силы тяжести должен быть перпендикулярен уровневой поверхности. В свою очередь, сила тяжести не является константной величиной. В районах подводных гор сила тяжести увеличивается, в местах океанических впадин — наоборот уменьшается. Вектор силы тяжести отклоняется в сторону горы. Но он должен быть перпендикулярен уровневой поверхности. Значит морская поверхность тоже имеет выпуклости там, где под водой находится гора. Эти выпуклости регистрируются альтиметрами, а значит, данные альтиметрии могут использоваться при составлении карты морского дна. Они соотносятся с данными, полученными с эхолотов и экстраполируются на те районы, которые не были промеряны эхолотами. При этом учитывается, что топография и сила тяжести коррелируют не везде, а лишь в местах с молодой земной корой. Также по альтиметрическим данным составляются карты аномалий сил тяжести.

Знание топографии морского дна и аномалий сил тяжести применяется во многих областях [4], например:

- Навигация
- Составление карт подводного рельефа и батиметрических карт
- Поиск подводных вулканов

- Поиск нефти
- Определение границ тектонических плит
- Структура литосферы
- Уточнение фигуры геоида

Для своей работы я выбрала область, связанную с моделированием волн цунами. Расположенные в море и на берегу станции постоянно контролируют различные параметры окружающей среды. Если какие-то из них содержат признаки, указывающие на возможное возникновение цунами, то координаты передаются в специальный центр, где выполняется математическое моделирование. Особенностью данной задачи является необходимость переключения между сетками с разными разрешениями. Переинтерполяция должна выполняться в зависимости от того, как далеко предполагаемая волна находится от береговой линии.

Скорость вычислений — крайне важный фактор в задаче моделирования. Ведь чем быстрее будет получен прогноз стихийного бедствия, тем больше времени у служб МЧС на принятие мер для защиты населения и стратегически важных объектов. Большим подспорьем в решении данного вопроса может стать использование фреймворков для распределенных вычислений, таких как Hadoop [5–7] и Spark [8–10]. Они могут быть развернуты на обычном вычислительном кластере, легко масштабируются, имеют высокий уровень отказоустойчивости и являются свободно распространяемыми программными продуктами. Препятствием для активного внедрения этих систем в область работы с географическими данными является их нацеленность на работу с потоковыми данными и, как следствие, поддержка только последовательного чтения файлов. В то время, как географические данные хранятся в сложных форматах, содержащих метаданные и многомерные массивы, при работе с которыми важна возможность произвольно-

го доступа. На настоящий момент нет решений, являющихся стандартом де-факто в данной области.

Обзор литературы

Знания, необходимые для выполнения моей работы можно разделить на две основные предметные области.

Первая охватывает спектр знаний о цунами, причинах их образования, способах обнаружения и предсказания, методах их моделирования. Сюда же можно отнести информацию об организации и работе центров по предупреждению цунами и наводнений, порядке действий сотрудников МЧС и сопутствующих служб при объявлении тревоги.

Вторая предметная область включает в себя особенности работы с форматами, используемыми для хранения научных данных; знания об устройстве и специфике работы систем распределённых вычислений.

Книга Поплавского и Храмушина “Оперативный прогноз наводнений на морских берегах Дальнего Востока России” [11] охватывает большой спектр тем, связанных с предупреждением цунами. Книга состоит из двух частей. Первая часть “Анализ сейсмической и гидрофизической подсистем оперативного предупреждения о цунами с позиции заблаговременности” содержит информацию об методах, используемых при обнаружении волн цунами, а также для оценки уровня опасности; исследуются условия заблаговременности предупреждения о цунами.

Особый интерес для специалистов в области прикладной математики представляет 2 часть книги “Вычислительное моделирование длинноволновой динамики океана в оперативном режиме и при выполнении экспертных работ”, где описаны вычислительные эксперименты, применяемые для прогноза и моделирования волн цунами, также рассмотрены возможности свободно распространяемых программно-вычислительного комплексов

Ani (моделирование цунами, штормовых нагонов, экстремальных течений и приливного режима в открытом океане и вблизи побережья) [12] и Mario (ведение и анализ цифровых приливных архивов и мореографных записей колебаний уровня моря) [13]. Также в книге даны указания по техническому оснащению служб предупреждения цунами (на примере Сахалина) по состоянию на 2 половину 90-х годов.

Кроме ознакомления с материалом книги, также был проведен анализ исходного кода программно-вычислительных комплексов Ani и Mario. Обе программы имеют очень грамотную реализацию с точки зрения учёного-океанолога. Учитывается множество ньюансов, например, различные проекции карт, реализована работа с несколькими форматами файлов. Разработка программы была начата в конце 80-х — начале 90-х годов. Используются простые, но вместе с тем надежные технологии. Программной код написан на языке программирования C, для реализации графического интерфейса используется библиотека BGI, разработанная компанией Borland. Там, где это возможно, выполнено распараллеливание вычислений с помощью технологии OpenMP. Программа реализована для операционной системы Windows. Установка производится на локальную машину, поддержка распределённых систем отсутствует, таким образом, несмотря на реализацию параллелизма, его потенциал не может быть полностью использован, так как ограничен ресурсами одной машины. Файлы с данными целиком загружаются в оперативную память, там же производятся все операции над ними. Вместе с невозможностью запуска программы в распределённой сети, это накладывает ограничения на размер обрабатываемых файлов, связанные с объемом оперативной памяти. Решением данной проблемы может стать использование вычислительного кластера. В этом случае, оперативная память всех компьютеров, подключенных к сети используется как общее пространство, что существенно увеличивает доступный для работы объем памяти. Именно по такой схеме и планируется построить

работу разрабатываемой мной программы.

Далее рассматриваются решения, применяемые для обработки файлов с географическими и метеорологическими данными в распределённых системах.

Возможность применения стека технологий Hadoop для обработки научных данных сразу заинтересовала IT сообщество. Основная проблема состоит в том, системы для обработки больших объемов данных, такие как Hadoop, Spark и другие в первую очередь предназначены для работы с данными, которые легко разбить на независимые части (split'ы) для параллельной обработки.

Первая попытка создать инструментарий для эффективной работы с научными данными была предпринята в 2011 году, спустя буквально пару лет после появления фреймворка Apache Hadoop. SciHadoop – плагин для Hadoop, разработаный группой учёных из Калифорнийского университета [14]. Основная задача SciHadoop, по мнению его разработчиков, выполнение логических запросов к массивам научных данных на основе их модели.

Был разработан формат файла, ориентированный на работу с многомерными массивами. Ключевыми понятиями которого являются форма массива (shape) и угловая точка (corner point). Форма массива определяется длиной каждого из его измерений. Угловая точка – определяет положение вложенного массива внутри основного. Эти понятия используются при составлении запросов для SciHadoop. Для написания запросов также был определен свой собственный синтаксис. Кроме того SciHadoop корректирует работу планировщика задач в Hadoop. Основываясь на знании о том, как представлены данные, производится корректировка разбиения данных на партиции для повышения оптимальности выполнения запросов.

Работа SciHadoop реализована в виде стандартного MapReduce приложения. SciHadoop инициализируется с помощью класса FileInputFormat,

где происходит разделеление входных данных на логические части и распределение их по физическим узлам. Разделение данных на партиции в SciHadoop представлено в виде Hadoop-класса InputSplit и представляется во фреймворке как набор вложенных массивов. Каждая партиция обрабатывается классом RecordReader, который загружает логический раздел из файловой системы с помощью библиотеки NetCDF и направляет его на выполнение в map функцию.

Ещё одно решение было представлено в 2015 году [15]. Это первая работа, обеспечивающая нативную поддержку научных форматов данных в системах big-data. Созданный инструментарий рассматривается в сравнении с вариантом, когда NetCDF файлы для обработки в Hadoop и Hive преобразуются в давно известный формат CSV. Это достаточно надёжный вариант, но он требует больших затрат как по количеству используемой на жёстком диске памяти, так и по времени обработки. Авторы приводят пример: NetCDF файл размером в 20 гигабайт в формате CSV занимает уже целых 119 гигабайт, в 6 раз больше, а на саму конвертацию уходит почти 1,5 часа процессорного времени. Данные хранятся в HDFS, проведена оптимизация MapReduce для работы с ними. Функции стандартного API переопределены для прозрачной работы пользователя с NetCDF файлами наравне с другими форматами. Для выполнения запросов к данным, хранящимся в HDFS используется Hive. Созданный инструментарий, в сравнении с подходом, основанным на конвертации в CSV позволяет увеличить производительность в 20 раз и до 83% снизить использование дискового пространства.

Для обеспечения нативной поддержки NetCDF файлов был написан драйвер на основе библиотеки NetCDF, использованный для собственной реализации классов InputFormat (NetCDFInputFormat) и RecordReader (NetCDFRecordReader). RecordReader использует этот драйвер для чтения данных из NetCDF файла, где каждая запись — строка многомерного мас-

сива. Таким образом обеспечена поддержка Hadoop'ом всех тех структур данных, которые могут быть в NetCDF файле. Реализованный драйвер также используется для соотнесения границ записей в файле с границами блока в HDFS, поскольку для максимально быстрой работы необходимо, чтобы логическая структура файла совпадала с физической.

Узким местом в системах больших данных является подсистема ввода-выода. Для оптимизации чтения NetCDFRecordReader не производит чтение отдельно взятых переменных, а загружает за одну операцию сразу ряд переменных, например, строку массива.

Среди недостатков решения можно отметить поддержку устаревшего формата NetCDF3 и отсутствие поддержки новых, более сложных форматов NetCDF4 и HDF5.

1 Постановка задачи

Процесс моделирования волн цунами подразумевает динамическую переинтерполяцию расчётной сетки в зависимости от расстояния до береговой линии. Для хранения данных используется сложный формат, использующий многомерные массивы и метаданные. Использование технологии распределённых вычислений помогает существенно ускорить процесс моделирования, что является крайне важным фактором для решения задачи предсказания цунами. Кроме того, распределённый кластер позволяет работать с большими объёмами данных, содержащими информацию не только о прибрежной зоне, а в целом о Мировом океане, обработать которые посредством одного единственного компьютера невозможно.

В связи с вышеуказанными условиями становится ясно, что применение распределённых вычислений в задаче моделирования цунами является важной и актуальной задачей, однако на данный момент нет информации о существовании таких программных решений. Необходимо разработать инструментарий, позволяющий производить переинтерполяцию расчётной сетки в распределённой вычислительной среде для решения задачи моделирования волн цунами. Программное решение должно быть реализовано на языке программирования Scala с использованием фреймворка для распределённых вычислений Apache Spark.

Система должна содержать следующий функционал:

- поддержка работы с многомерными массивами данных (чтение, обработка, запись)
- возможность работы с распределённой файловой системой

- переинтерполяция заданного участка карты (координаты и требуемая точность задаются пользователем)

2 Теоретическая часть

Существует 2 основных способа предсказания цунами.

1. Сейсмический. Основан на регистрации сейсмических волн, распространение которых предваряет приход цунами. Определяются координаты эпицентра и магнитуда землетрясения – по этим данным и основываясь на анализе исторических данных оценивается вероятность возникновения цунами. Существенным недостатком такой модели является большой процент ложных срабатываний (до 80%). Ложная тревога приводит к экономическому ущербу: затраты на эвакуацию жителей и подготовку промышленно-хозяйственных комплексов к приходу цунами порой превышают стоимость устранения ущерба от небольшой волны.

2. Гидрофизический. Подразумевает анализ волновых процессов океана с помощью установленных в различных местах гидрофизических станций. На основе этих производится быстрое численное моделирование для расчета времени прихода волны [16].

Возникновение и движение волн цунами – сложный процесс, состоящий из нескольких фаз, для каждой из которых определены математические модели. Вот некоторые примеры задач для моделирования [17]:

1. Генерация волн цунами

2. Расчёт максимальной высоты волны при подходе к берегу

3. Распространение и трансформация волн цунами

Каждой из этих задач для проведения расчётов требуются батиметрические данные. Важным параметром вычислений является точность расчётов, но если по всему фронту движения волны использовать мелкие расчётные сетки, то вычисления займут неоправданно много времени. Применение высокоточных сеток оправдано лишь при моделировании волны, входящей в прибрежную зону, до этого момента целесообразно использовать более грубую расчётную сетку. Необходимая точность определяется специалистом, выполняющим моделирование, поэтому необходим инструмент, генерирующий из большого массива данных расчётную сетку для некоторого региона с заданными параметрами точности.

Для создания новой расчётной сетки необходимо извлечь часть массива данных и произвести переинтерполяцию, согласно заданным условиям. Переинтерполяция в программе осуществляется по формуле Лагранжа:

$$f(x,y) = \sum_{j=1}^{n}\sum_{k=1}^{m} z_{jk} \cdot \frac{W_j(x)}{x - x_j} \cdot \frac{W_k(y)}{y - y_k}$$

$$\text{где } W_j(x) = (x - x_1) \times \cdots \times (x - x_{j-1}) \times (x - x_{j+1}) \times \cdots \times (x - x_n)$$

Формула для $W_k(y)$ считается аналогично.

Для оперативного выполнения численного моделирования цунами целесообразно использовать распределённые вычисления. На текущий момент существует несколько решений для осуществления таких вычислений. Одним из них является фреймворк Apache Spark. Как уже отмечалось в предыдущих разделах данной работы, основной проблемой является разный подход к чтению данных.

Файл с данными батиметрии в формате NetCDF имеет следующую структуру:

- блок метаданных (используемые измерения и их размерность; пере-

менные, их имя, диапазон значений; дополнительные атрибуты)

- блок данных (содержит массивы значений указанных переменных)

В качестве примера можно рассмотреть файл со значениями батиметрии, имеющий следующие параметры:

- измерение x содержит 7200 значений; значения соответствующей переменной лежат в диапозоне от 60 градусов западной долготы до нулевого меридиана
- измерение y содержит 3600 значений; значения соответствующей переменной лежат в диапозоне от 60 градусов южной широты до 90 градусов южной широты
- переменная z зависит от переменных x и y; её значения – значения глубины в точке с координатами (x, y); диапазон значений от -7323 до 3095

```
netcdf w060s60 {
dimensions:
    x = 7200 ;
    y = 3600 ;
variables:
double x(x) ;
        x:long_name = "x" ;
        x:actual_range = -60., 0. ;
    double y(y) ;
        y:long_name = "y" ;
        y:actual_range = -90., -60. ;
    short z(y, x) ;
        z:long_name = "z" ;
        z:_FillValue = -32768s ;
        z:actual_range = -7323., 3095. ;
```

```
// global attributes:
        :Conventions = "COARDS/CF-1.0" ;
        :title = "w060s60.nc" ;
        :history = "xyz2grd -V -R-060/000/-90/-60 -I30c
         w060s60.Bathymetry.srtm
        -Gw060s60.nc=ns -ZTLhw -F" ;
        :GMT_version = "4.5.12 [64-bit]" ;
        :node_offset = 1 ;
data:
 x = -59.9958333333, -59.9875, -59.9791666667, -59.9708333333,
     ...
    -0.0208333333333357, -0.0125000000000028, -0.00416666666666667 ;

 y = -89.9958333333333, -89.9875, -89.9791666666667, -89.9708333333333,
     ...
    -60.0291666666667, -60.0208333333333, -60.0125, -60.0041666666667 ;

 z = 2775, 2775, 2775, 2775, 2775, 2775, 2775, 2775, 2775, 2775,
    ...
    -5330, -5329, -5328, -5327, -5328, -5329, -5329, -5329, -5329 ;
}
```

Организовать работу с такими файлами помогает библиотека SciSpark. Более подробное её описание дано в следующем разделе.

3 Реализация алгоритма

Прежде чем производить реализацию алгоритма, необходимо сначала организовать и произвести настройку системного окружения вычислительной среды. Поддержку распределённых вычислений планировалось реализовать с помощью фреймворка Apache Spark. Для работы с данными в Spark используется модель RDD (Resilient Distributed Dataset) - устойчивый распределённый набор данных. RDD хранит набор пар типа ключ-значение (key, value) и позволяет проводить над ними параллельные операции. Такое решение хорошо подходит для работы с неструктурированными данными. Как показано выше (см. пример NetCDF файла), файлы с научными данными имеют определённую структуру и для организации работы Spark с файлами NetCDF необходима специальная библиотека с API, позволяющим работать с научными данными в терминах RDD. Предполагалось, что такое API придется разработать самостоятельно, но оказалось, что оно уже существует. В середине 2015 года на конференции IEEE был представлен проект SciSpark [18].

В SciSpark реализована структура sRDD (Scientific Resilient Distributed Dataset). Это sRDD обладает свойствами RDD, но при этом ориентирован на формат научных данных. Схема sRDD представлена на Рис. 2. sRDD состоит из объектов класса SciTensor. SciTensor содержит поле для метаданных (они представляются в виде пар ключ-значение) и объекты класса AbstractTensor, либо наследуемого от него класса. AbstractTensor предоставляет интерфейс для работы с одной из билиотек линейной алгебры – Breeze или ND4J. Breeze – библиотека, реализованная на языке Scala, является частью проекта ScalaNLP – набора библиотек для научных вычис-

лений на Scala [19–21]. Она существует достаточно давно, но поддерживает операции только с двумерными массивами. Для работы с n-мерными массивами в SciSpark используется библиотека ND4J [22].

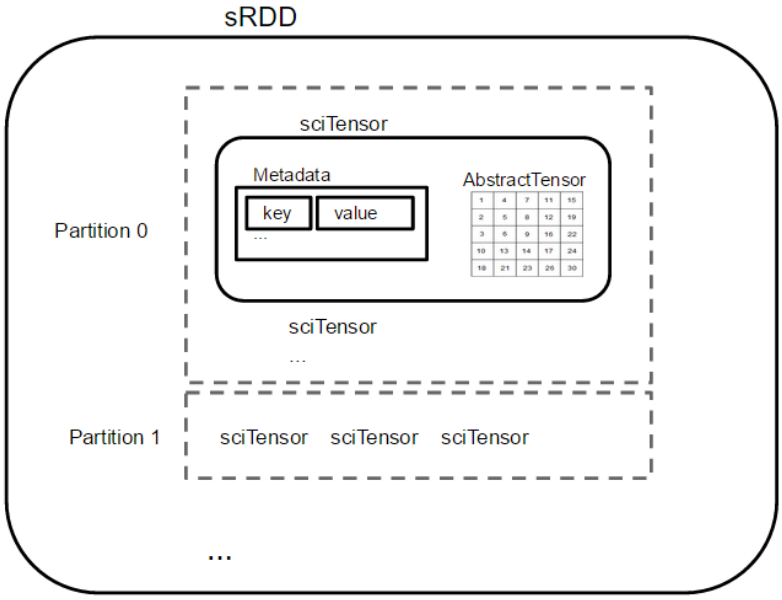

Рис. 2: Схема класса sRDD. Источник: [18]

На вход программе дается файл с координатами – двумя противоположными точками прямоугольника, задающего участок карты, который необходимо интерполировать, – и параметрами новой сетки – количество точек по каждому из измерений. Загрузка данных из распределённой файловой системы происходит с помощью функции `NetcdfDFSFile`. Затем, по заданным пользователем координатам, для каждой переменной многомерного массива с исходными данными извлекается необходимый кусок данных по следующей схеме:

1. В массивах переменных x и y производится поиск заданных пользователем координат и их "порядковых номеров". Распределённые коллекции данных не поддерживают итераторы, поэтому для определения номера позиции элемента в массиве используется разделяемая переменная типа `accumulator`. Разделяемые переменные – особый класс переменных в Spark, доступ к ним имеют все вычисли-

тельные узлы, но при этом, в зависимости от типа переменной, она доступна либо только для чтения, либо только для записи [23].

2. По вычисленным в пункте 1 координатам массивов x и y выделяется необходимую область из массива, содержащего значения глубины.

3. Выделенные массивы со значениями координат x и y и глубин объединяются в один sRDD.

Наконец, к полученному sRDD применяется функция интерполяции. Формула Лагранжа состоит из сумм произведений, каждую сумму можно посчитать независимо с помощью функции map. Результатом выполнения функции становится новый объект sRDD, содержащий значения новой сетки, который можно записать в NetCDF файл или использовать для дальнейших расчётов в SciSpark.

Выводы

Была поставлена задача разработать инструментарий, позволяющий производить переинтерполяцию заданного участка батиметрической карты для задачи моделирвоания волн цунами в распределённой вычислительной системе с использованием фреймворка Apache Spark. Основная проблема состояла в чтении формата научных файлов Spark'ом. Согласно своей задаче – моделирование волн цунами – был взят формат NetCDF, который используется для хранения данных о батиметрии. Формат NetCDF содержит метаданные и данные, содержащиеся в многомерных массивах.

Итоговая программа производит переинтерполяцию области карты, согласно заданным пользователем параметрам. Для организации работы с многомерными массивами и метаданными была использована библиотека SciSpark. Также была реализована функция, позволяющая выделять часть данных из загруженных в память массивов. Выполнение интерполяции происходит по методу Лагранжа.

Поставленная задача была решена в полной мере.

Заключение

В результате проведённой работы разработан инструментарий для работы в распределённой вычислительной среде, позволяющий произвести интерполяцию заданного участка карты в другую сетку с указанной точностью. Освоена работа с библиотекой SciSpark, позволяющей решить проблему с параллельным чтением и обработкой файлов, содержащих метаданные и многомерные массивы.

Список литературы

[1] Bathymetry from Space: White paper in support of a high-resolution, ocean altimeter mission / David T Sandwell, Walter HF Smith, Sarah Gille [и др.] // Int. Geophys. Ser. 2001. Т. 69.

[2] Bathymetry from space is now possible / David Sandwell, Sarah Gille, John Orcutt [и др.] // EOS Transactions. 2003. Т. 84. С. 37–44.

[3] Скорость звука в морской воде. URL: http://www.akin.ru/spravka/s_-i_svel.htm.

[4] Sandwell David T., Smith Walter H.F. Marine gravity anomaly from Geosat and ERS 1 satellite altimetry // Journal of Geophysical Research: Solid Earth. 1997. Т. 102, № B5. С. 10039–10054.

[5] Apache Hadoop, официальный сайт. URL: http://hadoop.apache.org/.

[6] Dean Jeffrey, Ghemawat Sanjay. MapReduce: Simplified data processing on large clusters // Communications of the ACM. 2008. Т. 51, № 1. С. 107–113.

[7] Bhandarkar Milind. MapReduce programming with apache Hadoop // Parallel & Distributed Processing (IPDPS), 2010 IEEE International Symposium on / IEEE. 2010. С. 1–1.

[8] Apache Spark, официальный сайт. URL: http://spark.apache.org/.

[9] Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing / Matei Zaharia, Mosharaf Chowdhury, Tathagata Das

[и др.] // Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation / USENIX Association. 2012. С. 2–2.

[10] Spark: Cluster Computing with Working Sets. / Matei Zaharia, Mosharaf Chowdhury, Michael J Franklin [и др.] // HotCloud. 2010. Т. 10. С. 10–10.

[11] Оперативный прогноз наводнений на морских берегах Дальнего Востока России / А.А. Поплавский, В.Н. Храмушин, К.И. Непоп [и др.] // Южно-Сахалинск: ДВО РАН. 1997.

[12] Храмушин В.Н. Программно-вычислительный комплекс Ani. 2010. Номер гос. регистрации 2010615848. URL: http://shipdesign.ru/SoftWare/2010615848.html.

[13] Храмушин В.Н. Программно-вычислительный комплекс Mario. Номер гос. регистрации 2010615847. URL: http://shipdesign.ru/SoftWare/2010615847.html.

[14] SciHadoop: Array-based query processing in Hadoop / Joe B Buck, Noah Watkins, Jeff LeFevre [и др.] // Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis / ACM. 2011. С. 66.

[15] Enabling scientific data storage and processing on big-data systems / Saman Biookaghazadeh, Yiqi Xu, Shujia Zhou [и др.] // Big Data (Big Data), 2015 IEEE International Conference on / IEEE. 2015. С. 1978–1984.

[16] Шевченко Г.В. История исследований цунами (ИМГиГ ДВО РАН) // Вестник дальневосточного отделения российской академии наук. 2011. № 6 (160).

[17] Методика расчета максимальных высот волн цунами в защищаемых пунктах побережья Дальнего Востока Российской Федерации / В.С. Косых, Л.Б. Чубаров, В.К. Гусяков [и др.] // Результаты испытания новых и усовершенствованных технологий, моделей и методов гидрометеорологических прогнозов. 2013. № 40. С. 115–134.

[18] SciSpark: Applying in-memory distributed computing to weather event detection and tracking / Rahul Palamuttam, Renato Marroquin Mogrovejo, Chris Mattmann [и др.] // Big Data (Big Data), 2015 IEEE International Conference on / IEEE. 2015. С. 2020–2026.

[19] Bugnion Pascal. Scala for Data Science. Packt Publishing Ltd, 2016.

[20] An overview of the Scala programming language: Tech. Rep.: / Martin Odersky, Philippe Altherr, Vincent Cremet [и др.]: 2004.

[21] Odersky Martin, Spoon Lex, Venners Bill. Programming in scala. Artima Inc, 2008.

[22] ND4J, официальный сайт. URL: http://nd4j.org/.

[23] Learning Spark: Lightning-Fast Big Data Analysis / Holden Karau, Andy Konwinski, Patrick Wendell [и др.]. O'Reilly Media, Inc., 2015.