УДК 52.08:004.62:681.518.3
DOI: 10.30533/0536-101X-2022-66-6-27-36

МАРШРУТИЗАЦИЯ ПРОСТРАНСТВЕННО-КООРДИНИРОВАННЫХ СООБЩЕНИЙ В УСЛОВИЯХ РАСПРЕДЕЛЕННОГО ОКРУЖЕНИЯ

И.Н. БОНДАРЕВ[1] ✉, **О.Г. ГВОЗДЕВ**[1] ✉

[1] Московский государственный университет геодезии и картографии, Москва, Россия
✉ ib.miigaik@gmail.com
✉ gvozdev@migaik.ru

ЦИТИРОВАНИЕ Бондарев И.Н., Гвоздев О.Г. Маршрутизация пространственно-координированных сообщений в условиях распределенного окружения // Известия вузов «Геодезия и аэрофотосъемка». 2022. Т. 66. № 6. С. 27–36. DOI:10.30533/0536-101X-2022-66-6-27-36

КЛЮЧЕВЫЕ СЛОВА пространственный анализ, пространственные предикаты, адаптивные геоинформационные технологии

АННОТАЦИЯ Построение высокопроизводительных распределенных информационных систем предполагает выбор схем распределения задач и обрабатываемых данных по узлам в их составе. При этом, рациональное использование вычислительных ресурсов достигается путем минимизации передачи данных между узлами системы и выполнения их обработки в максимальной близости к обрабатываемым данным. В работе представлен метод решения проблемы пространственной маршрутизации, путём

формирования её декларативного описания, с последующим его автоматическим преобразованием различные исполняемые формы, пригодные для встраиваемых устройств, интеграции с СУБД, интеграции в API-шлюзы и др. Предложенный метод позволяет получить гибкость и наглядность, характерную для универсальных систем, одновременно с производительностью на уровне специализированных решений. Кроме того, данный метод расширет набор теоретических и инструментальных основ построения адаптивных геоинформационных технологий.

1 ВВЕДЕНИЕ

Задачу организации доступа к пространственно-временным данным в условиях распределенных вычислительных окружений и распределенного хранилища данных можно разбить на три подзадачи:
— организация низкоуровневого хранения пространственных данных,
— описание правил распределения данных по узлам,
— обеспечение консистентности поведения узлов.

С ростом объемов пространственных данных, а также увеличением необходимых операций над пространственными данными, понадобилась модификация систем управления базами данных, в результате чего появились системы управления пространственными базами данных (Spatial Database Management System — SDBMS) [1].

Для обработки потоков пространственно-временных данных высокой интенсивности, вычислительной мощности отдельных серверов становиться недостаточно, что приводит к необходимости построения распределенных систем обработки. Соответственно, необходимо рассматривать способы хранения и обработки пространственных данных, учитывающие специфику распределенных систем. В основе построения распределенных информационных систем, обеспечивающих оперативную обработку больших объемов данных, лежит концепция разбиения данных (data partitioning), включающая в себя всевозможные способы распределения данных по узлам системы.

В результате анализа публикаций было выявлено, что наиболее популярным методом является внедрение поддержки пространственных данных и операций над ними в программные решения, решающие проблемы разработки и выполнения распределенных информационных систем. Так, авторы статьи [2] предложили структуру распределенного хранения и обработки геопространственных данных для крупномасштабной WebGIS, в том числе разработали прототип фреймворка VegaCI, являющегося надстройкой над Apache Hadoop[1]. VegaCI может быть интегрирован со специфичными для ГИС системами, протоколами, сервисами, например, WMS и WFS. По оценке авторов, производительность доступа к геопространственным данным

1 The Apache™ Hadoop® project develops open-source software for reliable, scalable, distributed computing. 2022 [Электронный ресурс]. Режим доступа: https://hadoop.apache.org/ (дата обращения: 19.12.2022).

предложенным способом примерно на 57–153 % выше, чем у SDBMS с задействованной репликацией.

Ощутимо позже, чем Apache Hadoop, вышел Apache Spark[2], фреймворк для реализации распределенной обработки неструктурированных и слабоструктурированных данных, который также применяется для решения проблемы распределенного доступа к пространственным данным. В статье [3] приведено сравнение этих двух фреймворков применимо к задачам, требующим определения расстояния между объектами. Авторы сравнивают наиболее популярные пространственные расширения для Apache Hadoop и Apache Spark: SpatialHadoop и LocationSpark. Экспериментальное исследование показало, что LocationSpark является абсолютным победителем по времени выполнения благодаря эффективности обработки в оперативной памяти, обеспечиваемой Spark, и дополнительным улучшениям планировщика запросов. Об эффективности LocationSpark также свидетельствует, что авторы реализации SpatialHadoop перестали поддерживать проект и посоветовали использовать реализацию на базе Spark.

Несмотря на возможность использования этих инструментов для работы с большими пространственными данными в распределенных системах, решения на основе Hadoop и Spark не поддерживают обработку пространственно-временных данных в потоковом режиме. Одним из инструментов, предлагающих решение данной проблемы, является GeoFlink [4], который представляет из себя расширение Apache Flink[3]. Чтобы обеспечить эффективную обработку непрерывных пространственных запросов и эффективное распределение данных между узлами кластера Flink, строится пространственный индекс на основе сетки (Grid-Based Spatial Index), обеспечиювающий возможность отсекать пространственные объекты, которые не могут быть частью результата пространственных запросов, что позволяет добиться их эффективной обработки. Точно так же это помогает в распределении данных по узлам распределенной системы.

Наиболее популярным из семейства решений, являющихся надстройкой над существующими решениями для работы в распределенных средах, является GeoMesa[4] [5]. Отличительная особенность GeoMesa — пространственно-временная индексация поверх баз данных Accumulo, HBase, Google Bigtable и Cassandra. GeoMesa также обеспечивает потоковую обработку пространственно-временных данных практически в реальном времени за счет наложения пространственной семантики поверх Apache Kafka. Также GeoMesa обеспечивает возможности интеграции с геоинформационными системами за счет поддержки стандартных протоколов OGC, в том числе WFS и WMS.

2 Apache Spark. Unified engine for large-scale data analytics. 2022 [Электронный ресурс]. Режим доступа: https://spark.apache.org/ (дата обращения: 19.12.2022).

3 Apache Flink. Stateful Computations over Data Streams. 2022 [Электронный ресурс]. Режим доступа: https://flink.apache.org/ (дата обращения: 19.12.2022).

4 GeoMesa. 2022 [Электронный ресурс]. Режим доступа: https://www.geomesa.org/ (дата обращения: 19.12.2022).

Перечисленные выше методы позволяют оптимальнее использовать разнородные доступные ресурсы, а также повысить надежность системы хранения и обработки данных в целом, но не учитывают специфику данных, требующих географически-распределенной обработки. Например, в географически-распределенных системах, обслуживание пользователей из различных стран и регионов, предполагает хранение данных пользователя на узлах системы более близких к нему в телекоммуникационном смысле — для обеспечения скорости доступа.

Геошардинг (geosharding) или пространственное сегментирование — метод хранения и обработки пространственных данных, который дает возможность партиционирования данных по их пространственным признакам. Встроенные в MongoDB[5] механизмы пространственного сегментирования, применяемые для оптимизации доступа к данным и балансировки нагрузки на узлы сети, предполагают пространственную локальность доступа к данным, определяя границы наборов, хранимых каждым узлом, используя сегментацию и построение пространственных индексов, что, как правило, используется для географического распределения узлов БД и балансировки нагрузки между ними.

Однако, авторы статьи [6] отмечают, что стандартные механизмы не всегда эффективны и могут быть оптимизированы. Авторы используют диаграммы Вороного [7] для построения пространственного индекса, что позволяет избежать фиксированного разделения пространства и адаптировать пространственную сеть к неравномерности наборов данных с географической привязкой, размещая каждый сегмент там, где его обработка будет осуществляться максимально быстро (это позволяет оптимизировать балансировку нагрузки).

Решение таких задач, как поиск ближайших объектов или мониторинг подвижных объектов, в распределенных системах основывается на геошардинге. Он позволяет дифференцировать узлы и группы узлов системы по обрабатываемой ими нагрузке, например обеспечить большие вычислительные мощности для регионов с большей нагрузкой и меньшие — для более отдаленных.

2 МАТЕРИАЛЫ И МЕТОДЫ

2.1 Задача пространственной маршрутизации

При организации хранилищ пространственных данных, предполагающих выполнение пространственных запросов, для которых необходима возможность обращения к пространственно-смежным данным, имеет смысл организация сегментации данных, обеспечивающая физическую и логическую близость размещения

5 MongoDB. 2022 [Электронный ресурс]. Режим доступа: https://mongodb.org/ (дата обращения: 19.12.2022).

данных о близких в пространстве и времени явлениях и процессах, и наоборот — позволяющая разделить нагрузку, связанную с процессами и явлениями, на существенном пространственном и (или) временном удалении.

В контексте обработки потоков пространственно-временных данных высокой интенсивности это приводит к необходимости определения признаков, по которым каждое сообщение может быть направлено в различные подмножества узлов распределенной вычислительной системы.

Эту задачу можно назвать «задача маршрутизации сообщений, координированных в пространстве и времени». Или — для краткости — «задача пространственной маршрутизации».

Задача пространственной маршрутизации заключается в определении признаков, по которым сообщение (пространственный запрос, запрос на добавление или обновление сведений) может быть направлено к компонентам системы, отвечающим за затрагиваемые сегменты данных. Решения данной задачи должны:

— минимизировать влияние на пропускную способность и латентность системы в целом,
— исключить ложно-отрицательные срабатывания (пропуски и пропадания сообщений),
— допускать ложно-положительные срабатывания (избыточные сообщения) в небольших количествах (до 10 %),
— обеспечить детерминированность функционирования и воспроизводимость результата.

Выделение и обособление задачи пространственной маршрутизации позволяет выполнять её в разных вычислительных средах в составе распределенной системы:

— на периферийных устройствах в рамках парадигмы граничных (периферийных) вычислений[6],
— на входных узлах кластеров системы, например в рамках API-шлюзов,
— в системах ретроспективного анализа.

Данные возможности диктуют следующие дополнительные требования:

— возможность построения легковесных реализаций, пригодных в том числе для встраиваемых и периферийных систем,
— переносимость реализации, в частности путём исключения зависимостей от сложных сторонних программных компонентов, например СУБД.

Отдельные программные реализации задачи пространственной маршрутизации могут быть построены с применением стандартных инструментальных средств геоинформатики, но наибольший интерес, с позиции развития теоретической базы построения распределенных геоинформационных технологий, вызывает выявление частных случаев и построение специализированных для их решения

6 Edge Computing. What is Edge Computing: The Network Edge Explained. 2022 [Электронный ресурс]. Режим доступа: https://www.cloudwards.net/what-is-edge-computing (дата обращения: 19.12.2022).

операций пространственной маршрутизации, использование которых по отдельности или в режиме композиции позволяет решить широкий спектр конкретных случаев задачи пространственной маршрутизации. Рационализация самих процессов композиции также представляет существенный теоретический интерес.

2.2 Реализация системы моделирования процедур пространственной маршрутизации

Для моделирования задач пространственной маршрутизации авторами данной статьи предлагается объектная нотация, представляющая собой гибридизацию деревьев решений [8] и паттерна проектирования «спецификация» [9].

В целях наглядности и удобства проверки работоспособности концепции нотация ограничена тремя типами объектов:

— Many — множество правил, каждое из которых должно быть проверено,
— Result(T) — правило, присваивающее проверяемой точке метку T,
— DistanceFrom(P) — правило, поведение которого зависит от значения пространственного предиката вида «расстояния текущей точки до точки T».

С помощью этой нотации простая задача пространственной маршрутизации может быть записана следующим образом (язык программирования Python 3):

```
spatial_routing = ( md.Many()

  # г. Москва
  .branch( md.DistanceFrom( md.Point( 55.755841, 37.617563 ) )

    # г. Москва ("Старая Москва")
    .lower_than( 30*KM, md.Result( 'city.russia.moscow' ) )

    # Московская область
    .lower_than( 200*KM, md.Result( 'city.russia.moscow-with-suburbs' ) )

  )

  # Санкт-Петербург
  .branch( md.DistanceFrom( md.Point( 59.939090, 30.315831 ) )
    .lower_than( 30*KM, md.Result( 'city.russia.saint-petersburg' ) )
  )

  # г. Новосибирск
  .branch( md.DistanceFrom( md.Point( 55.030194, 82.920447 ) )
    .lower_than( 30*KM, md.Many()

      .branch( md.Result( 'city.russia.novosibirsk' ) )

      # г. Новосибирск, Калининский район, микрорайон Пашино
      .branch( md.DistanceFrom( md.Point( 55.177972, 82.973500 ) )
        .lower_than( 3.6*KM, md.Result( 'city.russia.novosibirsk.pashino' ) )
      )

    )
  )
)
```

Данная система моделирования может быть расширена за счет введения других правил пространственной маршрутизации. Наиболее перспективными с точки зрения задачи пространственной маршрутизации являются:

— принадлежность точки к ячейке регулярной сетки,
— принадлежность точки к ячейке регулярной сетки с возможностью пересечения ячеек сетки,
— пространственное отношение точки и прямой,
— пространственное отношение точки ограничивающего прямоугольника,
— пространственное отношение точки и полигона.

Перспективным направлением исследований является разработка оптимизационных алгоритмов, позволяющих по набору строгих пространственных предикатов (не предполагающих ложных срабатываний) определить минимальную структуру дерева решений из упрощенных пространственных предикатов, которая допускает заданную долю ложно-положительных срабатываний.

Приведенная объектная нотация является зависимой от высокоуровневого интерпретируемого языка программирования Python 3, что явным образом противоречит требованию о переносимости. Для сохранения удобства моделирования на высокоуровневом языке, а также возможности машинной обработки разрабатываемых и получаемых моделей, при обеспечении переносимости и высокой производительности, предлагается прибегнуть к техникам метапрограммирования [10]. Это предполагает использование объектной модели в качестве исходных данных для автоматической генерации исходного кода процедуры, семантически-аналогичной объектной модели, но построенной с помощью низкоуровневого языка программирования.

Важной особенностью такого подхода является возможность обеспечения строгого соответствия и одновременного обновления множества реализаций одной процедуры пространственной маршрутизации, что может быть необходимо, например, в случае ее востребованности в различных вычислительных средах:

— на встраиваемом периферийном оборудовании (я/п C, C++),
— во встроенных в СУБД средах выполнения (я/п Lua, JavaScript),
— для ретроспективного анализа на вычислительных кластерах (я/п Java, Python).

Таким образом может быть обеспечена совместимость с популярными СУБД с пространственными расширениями: PostgreSQL, MongoDB и др.

3 РЕЗУЛЬТАТЫ И ОБСУЖДЕНИЕ

Для реализации системы моделирования процедур пространственной маршрутизации был использован язык программирования Python 3.10, процедура вычисления расстояния на основе

координат в WGS-84, реализующая метод, разработанный Таддеусом Висенти[7] [11] и библиотека CFFI[8] для интеграции с кодом на языке C.

Для тестирования использовался набор из трех миллионов точек, построенный с помощью генератора случайных чисел вокруг центров городов (Таблица 1).

Таблица 1 ▶
Параметры генерации тестового набора точек.

№	Город	Координаты		Радиус (км)	Количество точек
		Широта	Долгота		
1	Москва	55,755841	37,617563	205	1000000
2	Санкт-Петербург	59,939090	30,315831	35	1000000
3	Новосибирск	55,030194	82,920447	35	1000000

Тестирование проводилось для трех реализаций:

- **noop** — не выполняющая полезных вычислений (для учета влияния тестовой среды выполнения на эксперимент),
- **object** — непосредственное выполнение объектной нотации, реализованной на языке программирования Python 3 (для вычисления расстояний использовалась функция, реализованная на низкоуровневом языке программирования C),
- **compiled** — выполнение высокопроизводительной реализации на языке программирования C, полученной с помощью техник метапрограммирования.

Вычислительные эксперименты были проведены на сервере с 2 CPU типа Intel Xeon E5-2690 с тактовой частотой 2,90 ГГц и 512 Гб оперативной памяти.

Результаты тестирования представлены в Таблице 2.

Таблица 2 ▶
Результаты тестирования реализаций процедуры пространственной маршрутизации.

№	Реализация	Производительность		
		Общее время (с)	Пропускная способность (Точек/с)	Латентность (мкс/Точку)
1	noop	5,63	532597,64	1,88
2	object	21,19	141575,52	7,06
3	compiled	9,37	320616,86	3,12

Тестирование подтвердило работоспособность предложенной концепции. Полученная автоматически реализация обеспечила прирост производительности в 2,26 раза и может быть использована, в том числе во встраиваемых системах и окружениях периферийных вычислений.

7 Формулы Винсенти. Vincenty's formulae. 2022 [Электронный ресурс]. Режим доступа: https://github.com/dariusarnold/vincentys-formula/blob/master/vinc.cpp (дата обращения: 19.12.2022).

8 CFFI documentation. C Foreign Function Interface for Python. 2022 [Электронный ресурс]. Режим доступа: https://cffi.readthedocs.io/ (дата обращения: 19.12.2022).

4 ВЫВОДЫ

Построение высокопроизводительных распределенных информационных систем предполагает выбор схем распределения задач и обрабатываемых данных по узлам в их составе. При этом рациональное использование вычислительных ресурсов достигается путем минимизации передачи данных между узлами системы и выполнения задач их обработки в максимальной близости к обрабатываемым данным.

Построение распределенных систем обработки пространственных данных, и, в особенности, потоков пространственно-временных данных высокой интенсивности, невозможно без учета пространственной семантики обрабатываемых данных. В частности — в логике распределения данных между узлами системы.

В данной работе проблема рассматривается более широко, как проблема маршрутизации сообщений (запросов к системе, данных для добавления или обновления), координированных в пространстве и времени, заключающаяся в определении признаков сообщения, на основе которых оно может быть направлено для дальнейшей обработки.

В работе предложен авторский подход к проблеме, заключающийся в создании системы моделирования (декларативного описания) правил пространственной маршрутизации, с последующим их автоматическим преобразованием в различные исполняемые формы, пригодные для встраиваемых устройств, интеграции с СУБД, интеграции в API-шлюзы и др.

Предложенный метод позволяет получить высокую гибкость и наглядность характерную для универсальных систем, одновременно с производительностью на уровне специализированных решений.

Кроме того, предложенный метод расширяет набор теоретических и инструментальных основ построения адаптивных геоинформационных технологий.

БЛАГОДАРНОСТИ

Результаты исследования были получены в рамках государственного задания № 0708-2020-0001 Минобрнауки РФ.

БИБЛИОГРАФИЯ

1. Nyerges T. Spatial Database Management Systems // The Geographic Information Science & Technology Body of Knowledge. 2021. DOI:10.22224/gistbok/2021.1.11.
2. Zhong Yunqin, Han Jizhong, Zhang Tieying, et al. A distributed geospatial data storage and processing framework for large-scale WebGIS // 20th International Conference on Geoinformatics. 2012. P. 1–7. DOI:10.1109/Geoinformatics.2012.6270347.
3. García-García F., Corral A., Iribarne L., et al. A Comparison of Distributed Spatial Data Management Systems for Processing Distance Join Queries // Advances in Databases and Information Systems. 2017. P 214–228. DOI:10.1007/978-3-319-66917-5_15.

4. Salman A., Komal M., Hiroyuki K., et al. A Distributed and Scalable Framework for the Real-time Processing of Spatial Streams // Preprint arXiv.org, 2020. [Электронный ресурс]. Режим доступа: https://arxiv.org/abs/2004.03352 (дата обращения: 19.12.2022).
5. Hughes J.N., Annex A., Eichelberger C. N., et al. A distributed architecture for spatio-temporal fusion // Geospatial Informatics, Fusion, and Motion Video Analytics. 2015. Vol. 9473. DOI:10.1117/12.2177233.
6. Graca João. GeoSharding: Optimization of data partitioning in sharded georeferenced databases // Computer Science, 2016. [Электронный ресурс]. Режим доступа: https://fenix.tecnico.ulisboa.pt/downloadFile/1407770020544988/Extended%20Abstract.pdf (дата обращения: 19.12.2022).
7. Aurenhammer F. Voronoi diagrams — a survey of a fundamental geometric data structure // ACM Computing Surveys, 1991. Vol. 23(3). P. 345–405. DOI:10.1145/116873.116880.
8. Kamiński B, Jakubczyk M, Szufel P. A framework for sensitivity analysis of decision trees // Central European Journal Operations Research. 2018. Vol. 26(1). P. 135–159. DOI:10.1007/s10100-017-0479-6.
9. Evans E. Domain- Driven Design: Tackling Complexity in the Heart of Software // Addison-Wesley, 2004. 560 p.
10. Майоров А.А., Гвоздев О.Г. Композиция методологий обобщенного и метапрограммирования как средство увеличения адаптивности процесса разработки специализированных программных систем обработки пространственных данных // Известия вузов «Геодезия и аэрофотосъемка», 2015. № 5. С. 85–89.
11. Karney C.F.F., Deakin R.E. The calculation of longitude and latitude from geodesic measurements // Astronomische Nachrichten, 2010. Vol. 331. P. 852–861.

АВТОРЫ

Бондарев Илья Николаевич
Московский государственный университет геодезии и картографии, Москва, Россия
Кафедра информационно-измерительных систем
Факультет геоинформатики и информационной безопасности
0000-0003-2777-8982

Гвоздев Олег Геннадьевич
Московский государственный университет геодезии и картографии, Москва, Россия
Кафедра информационно-измерительных систем
Факультет геоинформатики и информационной безопасности
Кандидат технических наук
Доцент
0000-0002-1917-3206

Поступила 25.11.2022. Прошла рецензирование 10.12.2022. Принята к публикации 29.12.2022.