

ГЕОИНФОРМАТИКА

DOI: 10.30533/0536-101X-2020-64-3-341-348

О проблеме построения распределенных децентрализованных обобщенных геоинформационных технологий реального времени

© 2020 г. А.А. Майоров, А.В. Матерухин, О.Г. Гвоздев*

Московский государственный университет геодезии и картографии, Москва, Россия
*gvozdev@miigaik.ru

On design of distributed decentralized generic realtime geoinformation technologies

A.A. Maiorov, A.V. Materukhin, O.G. Gvozdev*

Moscow State University of Geodesy and Cartography, Moscow, Russia
*gvozdev@miigaik.ru

Received November 14, 2019

Revised Mars 2, 2020

Accepted June 1, 2020

Keywords: event-driven systems, functional scalability, geosensor network, highload systems, realtime GIS, realtime systems, smart geosensor, software architecture, streaming analytics.

Summary. A geosensor networks allows to gather large volume of spatiotemporal data. In near future this volume will hit a ten million events per second mark. There are few proprietary highly-specialized solutions, shows ability to proceed such amount of data. General purpose geospatial technologies can't afford required performance. Highly-specialized solutions lacks of adaptability and functional scalability. Authors propose to overcome this contradiction combining several conceptions. First — generic programming as a base for high performance and compile-time functional scalability. Second — distributed and decentralized structure, as a base for horizontal scalability and fault tolerance. Third — domain driven error handling. Detailed description of this concepts and their rationale are given. Several perspective research directions are described.

Citation: Maiorov A.A., Materukhin A.V., Gvozdev O.G. On design of distributed decentralized generic realtime geoinformation technologies. *Izvestia vuzov «Geodesy and Aerophotosurveying»*. 2020, 64 (3): 341–348. [In Russian]. DOI:10.30533/0536-101X-2020-64-3-341-348.

Поступила 14 ноября 2019 г.

После доработки 2 марта 2020 г.

Принята к печати 1 июня 2020 г.

Ключевые слова: высоконагруженные системы, геосенсорная сеть, интеллектуальный геосенсор, программная архитектура, системы потоковой аналитики, системы реального времени, событийно-ориентированные системы, функциональная масштабируемость.

Рассматривается проблема построения геоинформационных технологий, способных в реальном времени обрабатывать пространственно-временные данные, поступающие от сетей интеллектуальных геосенсоров. Проводится обзор существующих решений и их ограничений. Предлагается альтернативный существующим подход, основанный на методах обобщенного программирования, распределенной децентрализованной структуре системы, подходе к обработке ошибок предполагающем использование особенности предметной области для разрешения противоречий. Приводится описание и обоснование компонентов предлагаемого подхода.

Для цитирования: Майоров А.А., Матерухин А.В., Гвоздев О.Г. О проблеме построения распределенных децентрализованных обобщенных геоинформационных технологий реального времени // Изв. вузов «Геодезия и аэрофотосъемка». 2020. Т. 64. № 3. С. 341–348. DOI:10.30533/0536-101X-2020-64-3-341-348.

Введение

Геоинформатика — междисциплинарная область знаний о применении информационных технологий для моделирования систем, элементы которых координированы в пространстве и времени.

В последние десятилетия моделируемые средствами геоинформатики системы претерпевают значительные изменения, обусловленные, в частности: развитием встраиваемых и одноплатных (SoC) компьютеров; развитием стационарных и мобильных (беспроводных) сетей связи; увеличение доступности ГНСС (GPS, ГЛОНАСС) и приемников для них, что, в совокупности, привело к появлению нового класса устройств — интеллектуальных геосенсоров [1].

Интеллектуальные геосенсоры могут существовать как самостоятельные устройства, так и в виде компонентов в составе устройств интернета вещей (IoT), потребительских «клиентских» устройств (главным образом смартфонов) и специализированных измерительных приборов. Появление интеллектуальных геосенсоров создало потенциал для осуществления и сбора измерений, координированных в пространстве и времени, с большого количества объектов (агентов) в реальном времени и привело к появлению концепции геосенсорных сетей. Количество агентов (геосенсоров) и частота вы-

полнения измерений определяются, главным образом, особенностями предметной области и пределами доступных в её рамках технологий.

Примеры предметных областей, в которых геосенсорные сети применяются или могут быть применены в ближайшее время: логистика (мониторинг автомобильных заторов и общественного транспорта, агрегаторы услуг такси, мониторинг и оптимизация работы экстренных служб, диспетчеризация больших парков БПЛА, беспилотные автомобили); экологический и метеорологический мониторинг; телекоммуникации (например, мониторинг сетей связи); коммерция (геомаркетинг); сбор данных и поддержка принятия решений в условиях чрезвычайной ситуации.

Грубо оценить верхний предел требований к частоте обновления данных можно воспользовавшись теоремой отсчетов (теорема Котельникова, теорема Найквиста — Шеннона, sampling theorem) [2, 3]: любую функцию $F(t)$, состоящую из частот от 0 до f_1 , можно непрерывно передавать с любой точностью при помощи чисел, следующих друг за другом через $1/2f_1$ с, то есть с частотой $2f_1$. Если положить $f_1 = 1/t_m$, где t_m — минимальное время накопления значимых изменений в секундах, то для их передачи необходима частота измерений не менее $2/t_m$.

В табл. 1 приведены значения минимальных частот измерений для различных скоро-

Таблица 1

Минимальная частота измерений, Гц

Скорость передвижения, км/ч	Минимальнозначимое изменение положения, м										
	0,5	1	2	5	10	15	20	25	30	35	40
1	0,11	0,56	0,28	0,11	0,06	0,04	0,03	0,02	0,02	0,02	0,01
5	5,56	2,78	1,39	0,56	0,28	0,19	0,14	0,11	0,09	0,08	0,07
10	11,11	5,56	2,78	1,11	0,56	0,37	0,28	0,22	0,19	0,16	0,14
20	22,22	11,11	5,56	2,22	1,11	0,74	0,56	0,44	0,37	0,32	0,28
40	44,44	22,22	11,11	4,44	2,22	1,48	1,11	0,89	0,74	0,63	0,56
60	66,67	33,33	16,67	6,67	3,33	2,22	1,67	1,33	1,11	0,95	0,83
80	88,89	44,44	22,22	8,89	4,44	2,96	2,22	1,78	1,48	1,27	1,11
100	111,11	55,56	27,78	11,11	5,56	3,70	2,78	2,22	1,85	1,59	1,39
120	133,33	66,67	33,33	13,33	6,67	4,44	3,33	2,67	2,22	1,90	1,67
140	155,56	77,78	38,89	15,56	7,78	5,19	3,89	3,11	2,59	2,22	1,94
160	177,78	88,89	44,44	17,78	8,89	5,93	4,44	3,56	2,96	2,54	2,22

Оценка динамики размеров, частоты обновления и общего объема измерений в геосенсорных сетях

Эпоха		Свойства геосенсорных сетей		
		Агентов, тыс.	Частота обновлений	Событий в минуту, тыс.
Позавчера	До 2000-х г.	Обновления в ручном режиме (дни, недели, месяцы, годы)		
Вчера	2000-е	1–10	До 1 изм./мин	1–10
Сегодня	2010-е	10–100	0,1–1 Гц	60–6000
Завтра	2020-е	100–1000	1–10 Гц	6000–600000

стей перемещения геосенсора и минимально-значимых расстояний.

В табл. 2 приведены оценки авторов относительно динамики размеров существующих геосенсорных сетей, частоты сбора данных и общего объема измерений, требующих обработки. Из данных видно, что уже сейчас перед специалистами в областях телекоммуникаций, информатики и геоинформатики стоит проблема передачи и обработки данных геосенсорных сетей, генерирующих десятки миллионов событий в секунду.

Обзор существующих решений

Существующие решения в этой области носят, главным образом, узкоспециализированный характер. Среди общедоступных можно отметить продукты Яндекс.Пробки и GoogleTraffic, а также системы агрегации услуг такси (Яндекс.Такси, Uber, CityMobil, GettTaxi). Среди решений общего назначения можно отметить разработки ESRI [4, 5], GIS Cloud [6] и платформу с открытым исходным кодом GeoMesa [7]. Особенного внимания заслуживает теоретизация систем и процессов обработки потоков пространственно-временных данных высокой интенсивности, проведенная А.В. Матерухиным [1].

Кроме того, при обработке данных геосенсорных сетей активно используются программные решения не специализированные для задач обработки пространственных данных, в частности —

СУБД

реляционные (PostgreSQL, MySQL...);
колоночные и их производные
(Cassandra, Hupertable, ...);

документо-ориентированные
(MongoDB,...);

«Ключ-значение» (Redis, ...).

Очереди сообщений (ApacheKafka, NATS, AcitveMQ, RabbitMQ, ...).

Системы потоковой обработки (Apache Storm/Heron, Spark Streaming, AWS Lambda, ...).

Для этих систем также свойственно развитие средств (сторонних или встроенных) поддержки пространственных типов данных [8–10]. К сожалению, все упомянутые решения обладают особенностями или явными недостатками, не позволяющими говорить о решенности обозначенной проблемы.

Значительная часть этих решений или их базовых компонентов (СУБД, очереди сообщений и системы потоковой обработки) изначально не предусматривала специальных процедур для эффективного хранения и обработки пространственных данных, в частности — пространственной индексации [11]. Как следствие, горизонтальное масштабирование геоинформационных технологий, построенных на базе этих систем, невозможно без определенных свойств плотности распределения геосенсоров в пространстве: априорной известности; стационарности; неравномерности. Для них также свойственны проблемы совместимости режимов реального времени и ретроспективного анализа. СУБД, главным образом, ограничены неприспособленными для задач реального времени моделями данных и транзакционными моделями (ACID), ограничивающими вертикальную и горизонтальную масштабируемость [12].

Централизованная природа этих систем противоречит пространственной природе данных, что исключает (затрудняет) применение

многих очевидных оптимизаций производительности, создает единую точку отказа и определяет предел масштабируемости, как функцию максимальной мощности отдельного узла.

Среди подходов к построению геоинформационных технологий, как в целом, так и применительно к данной проблеме, доминируют два подхода.

Первый — построение узкоспециализированных решений, оптимизированных для обеспечения высокой пропускной способности и (или) низкой латентности в ущерб функциональной масштабируемости.

Второй — создание гибких универсальных решений, легко приспособляемых для конкретных условий задачи, но значительно отстающих от специализированных по показателям производительности.

Кроме того, доминирующие на рынке решения в высокой степени импортозависимы (Microsoft Windows, Oracle Java, ESRI, Google, Amazon, ...), что ограничивает или полностью исключает их применение, например, в образовании, научно-исследовательской деятельности и объектах критически важной информационной инфраструктуры.

Таким образом, по состоянию дел на сентябрь 2019 года, крупные геосенсорные сети, содержащие сотни тысяч агентов и более, поддерживаются только закрытыми узкоспециализированными решениями, только в отдельных предметных областях.

Предлагаемый подход

Анализ существующих решений показал, что рассматриваемая проблема не имеет общего решения или подхода к нему, в то же время, представители как мировой, так и отечественной индустрии преуспели в создании узкоспециализированных геоинформационных технологий, обеспечивающих потребности отдельных предметных областей. Анализ также показал, что универсальные решения данной проблемы возможны, но не способны достичь требуемых показателей производительности.

Авторами предлагается исследовать альтернативный подход к построению геоинформационных технологий, способных обрабатывать

данные современных и перспективных геосенсорных сетей, основанный на привлечении последних достижений фундаментальной и прикладной информатики, в области построения систем реального времени, событийно-ориентированных систем, систем потоковой аналитики, распределенных и децентрализованных систем.

Данный подход заключается в *совместном применении трех концепций*.

1. *Обобщенное программирование* как средство обеспечения высокой производительности, позволяющее сохранить статическую функциональную масштабируемость.

2. *Распределенная децентрализованная структура*, обеспечивающая горизонтальную масштабируемость и отказоустойчивость, без существенного повышения требований к отдельному узлу системы.

3. *Подход к обработке ошибок* и погрешностей измерений, учитывающий пространственную природу данных, распределенный характер системы, и использующий особенности предметной области для разрешения противоречий.

Обобщенное программирование

Обобщенное программирование предлагает альтернативный подход к компромиссу между производительностью и функциональной масштабируемостью. Авторами предлагаются следующие рабочие определения.

Функциональная масштабируемость (ФМ) — поддержка архитектурой и реализацией системы расширения её функциональных возможностей.

Статическая функциональная масштабируемость — ФМ, реализуемая на этапе разработки.

Динамическая функциональная масштабируемость — ФМ, реализуемая на этапе эксплуатации, в том числе без прерывания выполнения, в частности, путем приспособления существующих функциональных блоков для новых задач или условий функционирования.

Применение обобщенного подхода и технологий метапрограммирования как способа его реализации к задачам построения геоин-

формационных технологий различного назначения и задачам интеграции систем обработки пространственных данных, предложено в работах О.Г. Гвоздева [13–15] согласно которым:

специализированное решение — решение, определенное для отдельного частного случая, обладающее высокой производительностью и низкой функциональной масштабируемостью;

универсальное решение — решение, определенное для широкого диапазона случаев, обладающее высокой статической и динамической функциональной масштабируемостью при меньшей производительности;

обобщенное решение — решение, выраженное в общем виде, для которого определены процедуры получения реализаций отдельных частных случаев, позволяющее достичь высокой производительности с сохранением статической функциональной масштабируемости.

В качестве примера, приведем процедуры вычисления Евклидовой метрики, реализованные на языке C++, с использованием каждого из этих подходов. Выбор языка C++ обусловлен его поддержкой всех трех подходов к разработке (рис. 1–4).

```
double euclidean_2d(
    double x1, double x2,
    double y1, double y2
)
{
    return sqrt( pow( x1 - x2, 2 ) + pow( y1 - y2, 2 ) );
}
```

Рис. 1. Специализированное решение для двухмерного случая

Fig. 1. Specialized implementation for two dimensional case.

```
double euclidean_3d(
    double x1, double x2,
    double y1, double y2,
    double z1, double z2
)
{
    return sqrt(
        pow( x1 - x2, 2 )
    + pow( y1 - y2, 2 )
    + pow( z1 - z2, 2 )
    );
}
```

Рис. 2. Специализированное решение для трехмерного случая

Fig. 2. Specialized implementation for three dimensional case

```
double euclidean(
    std::vector< double > x,
    std::vector< double > y
)
{
    double agg = 0.0;
    assert x.size() == y.size()
    for ( auto i = 0 ; i != x.size() ; ++i )
        agg += pow( x[i] - y[i], 2 )
    return sqrt( agg );
}
```

Рис. 3. Универсальное решение для N-мерного случая

Fig. 3. Universal implementation for any number of dimensions

```

template <int N, class T >
T euclidean( T* x, T* y )
{
    double agg = 0.0;
    for ( auto i = 0 ; i != N ; ++i )
        agg += pow( x[i] - y[i], 2 )
    return sqrt( agg );
}

```

Рис. 4. Обобщенное решение для N -мерного случая

Fig. 4. Generic implementation for any number of dimensions

Очевидно, что специализированный подход требует написания дополнительного кода для каждого частного случая; универсальный подход позволяет процедуре обрабатывать точки в пространстве произвольной размерности N , но требует чтобы N было известно на этапе компиляции, что в свою очередь позволяет компилятору выполнить множество оптимизаций, таких как разворачивание цикла или векторизацию после мономорфизации кода. Таким образом, машинный код, сгенерированный компилятором для двух- и трехмерных случаев будет аналогичным для специализированного и обобщенного случаев, а значит будет обладать равной производительностью.

Применение этого подхода зарекомендовало себя в задачах кодирования данных для хранения и передачи [16], а также в решении вычислительноемких задач [17], в частности, при обучении искусственных нейронных сетей [18]. Встроенных возможностей языка, реализуемых его компилятором или интерпретатором, не всегда достаточно, что может быть компенсировано применением технологий метапрограммирования.

Распределенная и децентрализованная структура

В рамках данной работы авторами используются следующие определения:

распределенная система — система, в которой логические связи обусловлены местоположением элементов или их физическими связями;

децентрализованная система — система, в которой элементы не связаны иерархическими или декларативными отношениями.

Термин «централизованность» противопоставляется как термину «распределенность», так и термину «децентрализованность», тем не менее, распределенность и децентрализованность являются ортогональными понятиями, например:

интернет — распределенная и децентрализованная система;

ВОINC [19] — распределенная, но централизованная система;

Apache Kafka (в кластерном случае) [20] — децентрализованная система, но не распределенная пространственно;

Redis (в случае единственного узла) [21] — централизованная система (как в иерархическом, так и в пространственном смысле).

Системы, являющиеся распределенными и децентрализованными, согласно приведенным определениям, также являются параллельными. Совместное применение этих концепций позволяет добиться высокой горизонтальной масштабируемости и отказоустойчивости, формирует основу для множества оптимизаций, таких как перемещение вычислений к данным, что особенно значимо в силу пространственной природы обрабатываемых данных. Негативная сторона данного подхода — значительное усложнение проектирования, анализа и разработки систем.

Обработка ошибок

В рамках данной работы ошибки рассматриваются как широкий спектр явлений: от логических ошибок в оборудовании и ПО, нехватки ресурсов, аппаратных и коммуникационных сбоев, защитных отказов отдельных компонентов до погрешностей в измерениях координат и времени, и флуктуаций (jitter) скорости передачи и обработки данных.

Обработка ошибок основана на гарантиях, даваемых каждым поставщиком функций её потребителю. Например, протокол TCP гарантирует порядок доставки сообщений, отсутствие дубликатов, повторный запрос пропущенных пакетов и т.д.

Разработка традиционных геоинформационных систем и технологий полагается на систему гарантий, обеспечиваемую базовыми технологиями. В отечественной геоинформатике такими технологиями чаще всего являются реляционные СУБД с пространственными расширениями, например, PostgreSQL и PostGIS, реализующие систему гарантий ACID. Это обеспечивает простоту проектирования и анализа разрабатываемой системы, исключая возможные противоречивые (not consistent) состояния системы.

Реализация «жестких» систем гарантий, подобных ACID, возможна и в распределенных окружениях [22–23], однако её поддержание приводит к деградации производительности и многократному усложнению системы. По этой причине, разработчики распределенных систем прибегают к ослаблению гарантий. В СУБД это чаще всего реализуется посредством концепции *eventual consistency*, реализуемой с помощью версионирования блоков данных, с последующим разрешением возникающих противоречий на стороне прикладного приложения [24]. Аналогично при переходе к распределенным вычислениям разработчики отказываются от гарантии доставки сообщений «строго 1 раз» в пользу «не менее чем 1 раз», а также от гарантии упорядоченности сообщений в потоке. В этом случае, возникающие противоречия также разрешаются на стороне прикладного приложения.

Распределенный децентрализованный характер системы приводит к потребности еще большего ослабления гарантий, главным образом, в части сетевой связности узлов системы. В таких условиях ошибки необходимо рассматривать, как частный случай нормального состояния, корректная обработка которого возможна только с учетом свойств и требований предметной области, путем выработки коллективного поведения.

«Византийский» случай (возможность наличия злоумышленников среди узлов системы) в рамках данного исследования не рассматривается.

Выводы

Теоретический анализ предлагаемого подхода показывает возможность достижения пропускной способности до 200 тыс. событий в секунду на узел, при общей пропускной способности распределенной системы более 10 млн событий в секунду. Достижение этого результата потребует введения в научный оборот отечественной и мировой геоинформатики и разработки множества новых теоретических и прикладных направлений, в частности:

- принципы маршрутизации пространственно-координированных сообщений в распределенных окружениях;

- принципы построения интеллектуальных геосенсоров для работы в распределенных окружениях;

- принципы адаптации топологии распределенных геоинформационных технологий к изменениям плотности распределения событий в пространстве;

- алгоритм решения традиционных задач геоинформатики (k -ближайших, восстановление геополей и т.п.) в распределенном окружении;

- методы высокопроизводительной аппроксимации решений традиционных задач геоинформатики;

- методы определения доверительных интервалов для вычислений над разновременными измерениями с подвижных объектов.

ЛИТЕРАТУРА

1. *Матерухин А.В.* Теоретические основы и методология обработки потоков пространственно-временных данных: дисс. на соиск. уч. степ. доктора техн. наук: 25.00.35, М.: МИИГАиК. 2018. 173 с.
2. *Nyquist H.* Certain topics in telegraph transmission theory // Trans. AIEE. 1928. V. 47. P. 617–644.
3. *Котельников В.А.* О пропускной способности эфира и проволоки в электросвязи — Всесоюзный энергетический комитет // УФН. 2006. Т. 176, №7. С. 762–770.
4. Электронный ресурс: <https://www.esri.com/esri-news/arcnews/spring13articles/arcgis-enables-real-time-gis>.
5. Электронный ресурс: <https://www.esri.com/en-us/landing-page/product/2018/geodev-webinar-series/real-time-gis-mapping-and-analytics>.
6. Электронный ресурс: <https://www.giscloud.com/>.
7. Электронный ресурс: <https://www.geomesa.org/>.
8. Электронный ресурс: <https://docs.mongodb.com/manual/geospatial-queries>.
9. Электронный ресурс: <https://postgis.net/>.
10. Электронный ресурс: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0045-4>.
11. Электронный ресурс: <http://www.refractions.net/products/postgis/history/>.
12. Электронный ресурс: <https://pdfs.semanticscholar.org/e639/2f786ea3687448a257778e190cc64e58178c.pdf>.
13. *Майоров А.А., Гвоздев О.Г.* Композиция методологий обобщенного и метапрограммирования как средство увеличения адаптивности процесса разработки специализированных программных систем обработки пространственных данных // Изв. вузов. «Геодезия и аэрофотосъемка». 2015. № 5. С. 85–89.
14. *Майоров А.А., Гвоздев О.Г.* О задаче построения архитектуры обобщенной открытой программной платформы для обработки и хранения пространственных данных // Геодезия и картография. 2016. № 6. С. 57–60.
15. *Гвоздев О.Г.* Исследование принципов построения и разработка архитектуры обобщенной открытой программной платформы для обработки и хранения пространственных данных: Дисс. На соиск. уч. степ. канд. техн. наук: 25.00.35. М.: МИИГАиК, 2015. 140 с.
16. Электронный ресурс: <https://developers.google.com/protocol-buffers>.
17. Электронный ресурс: <http://numba.pydata.org/>.
18. Электронный ресурс: <https://www.tensorflow.org/>.
19. Электронный ресурс: <https://boinc.berkeley.edu/>.
20. Электронный ресурс: <https://kafka.apache.org/>.
21. Электронный ресурс: <https://redis.io/>.
22. Электронный ресурс: <https://pdfs.semanticscholar.org/5f5e/a0bda200b72c48219e5fda4f8df48a1d4a42.pdf>.
23. Электронный ресурс: <https://blog.couchbase.com/distributed-multi-document-acid-transactions-in-couchbase/>.
24. Электронный ресурс: <https://guide.couchdb.org/draft/consistency.html>.

REFERENCES

1. *Materukhin A.V.* Theoretical basis and methodology of spatiotemporal data stream processing: the dissertation of the doctor of Technical sciences. PhD. Moscow: State University of Geodesy and Cartography, 2018: 173 p. [In Russian].
2. *Nyquist H.* Certain topics in telegraph transmission theory. Trans. AIEE. 1928 (47): 617–644.
3. *Kotelnikov V.A.* On the bandwidth of ether and wire in telecommunications — All-Union Energy Committee. *UFN*. 2006, 176(7): 762–770. [In Russian].
4. URL: <https://www.esri.com/esri-news/arcnews/spring13articles/arcgis-enables-real-time-gis>.
5. URL: <https://www.esri.com/en-us/landing-page/product/2018/geodev-webinar-series/real-time-gis-mapping-and-analytics>.
6. URL: <https://www.giscloud.com/>.
7. URL: <https://www.geomesa.org/>.
8. URL: <https://docs.mongodb.com/manual/geospatial-queries>.
9. URL: <https://postgis.net/>.
10. URL: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-016-0045-4>.
11. URL: <http://www.refractions.net/products/postgis/history/>.
12. URL: <https://pdfs.semanticscholar.org/e639/2f786ea3687448a257778e190cc64e58178c.pdf>.
13. *Maierov A.A., Gvozdev O.G.* Generic programming and metaprogramming methodologies composition as a way to enhance adaptivity of development processes of specialized spatial data processing systems. *Izvestia vuzov «Geodesy and Aerophotogrammetry»*. 2015, 5: 85–89. [In Russian].
14. *Maierov A.A., Gvozdev O.G.* On problem of design of open generic software platform architecture for spatial data storage and processing. *Geodezia i kartografiya*. 2016, 6: 57–60. [In Russian].
15. *Gvozdev O.G.* Methodology research and design of open generic software platform architecture for spatial data storage and processing: PhD: Moscow State University of Geodesy and Cartography, 2015: 140 p. [In Russian].
16. URL: <https://developers.google.com/protocol-buffers>.
17. URL: <http://numba.pydata.org/>.
18. URL: <https://www.tensorflow.org/>.
19. URL: <https://boinc.berkeley.edu/>.
20. URL: <https://kafka.apache.org/>.
21. URL: <https://redis.io/>.
22. URL: <https://pdfs.semanticscholar.org/5f5e/a0bda200b72c48219e5fda4f8df48a1d4a42.pdf>.
23. URL: <https://blog.couchbase.com/distributed-multi-document-acid-transactions-in-couchbase/>.
24. URL: <https://guide.couchdb.org/draft/consistency.html>.