

Е. В. Шевчук, Ю. В. Шевчук

## Современные тенденции в области хранения и обработки сенсорных данных

*Аннотация.* Содержится обзор современных тенденций в области хранения и обработки сенсорных данных, различные подходы рассматриваются с точки зрения применимости к организации масштабируемых высокопроизводительных хранилищ данных, выделены наиболее перспективные решения с учетом специфики сенсорных данных и их использования.

*Ключевые слова и фразы:* временные ряды, масштабируемое распределенное хранение данных, онлайн сервисы хранения данных.

### Введение

По исследованиям IDC (International Data Corporation) [1, 2] одним из основных факторов роста объемов информации является увеличение доли автоматически генерируемых данных: с 11% от общего объема в 2005 г. до более 40% в 2020 г. Т.е. наблюдаемый экспоненциальный рост объемов мировой информации будет происходить в основном за счет поступления информации от таких источников данных, как спутники дистанционного зондирования Земли, GPS, мобильные устройства, датчики физических величин и т.п.

Также в своих исследованиях IDC делает вывод, что большая часть полезных данных теряется — используется менее 3 из 23% потенциально полезных данных, которые могли бы найти применение (в том числе с технологиями Big Data). Поскольку считается, что возможность анализа больших данных даст совершенно новый инструмент исследований в различных областях человеческой деятельности, то очевидно, что потеря значительной части входных данных для этого инструмента ограничит возможности его применения и снизит достоверность полученных результатов. Намечилась тенденция в области хранения данных — не выбрасывать ничего,

хранить все получаемые данные в надежде на то, что будут использованы имеющиеся или созданы новые технологии обработки больших данных, которые позволят проанализировать все собранные данные. Кроме того, не только надежное сохранение данных, но и возможность эффективного доступа к ним по запросу является необходимой предпосылкой для обработки и проведения анализа этих данных.

Не удивительно, что в современных подходах к хранению и обработке сенсорных данных уделяется внимание не только эффективному хранению и доступу к данным, но и созданию инструментария или инфраструктуры, позволяющих выполнять как простую обработку, так и аналитические исследования над собранными данными.

### **1. Особенности сенсорных данных**

Рассмотрим характерные черты, отличающие сенсорные данные от данных, поступающих от других источников:

- сенсорные данные от одного датчика представляют собой простые временные ряды;
- часто в течение длительного периода времени данные от датчика могут не менять значения;
- в некоторых случаях имеет место большая скорость увеличения объема данных (несколько точек в секунду на каждый датчик);
- базовые операции, которые выполняются над данными — это запись и считывание (один раз записанные, данные не подвергаются изменениям и перезаписи).

Основные варианты использования сенсорных данных:

- в системах мониторинга различных объектов для получения информации о текущем состоянии объекта;
- в системах мониторинга для получения агрегированной информации, исторической информации и выявления тенденций в поведении объекта;
- как источник данных для технологии data mining (аналитическая обработка больших данных) для выявления новых закономерностей.

Проведенный анализ современных решений по хранению и обработке сенсорных данных имел своей целью выявить подходы, наиболее полно учитывающие особенности сенсорных данных и возможности их применения.

## **2. Основные способы хранения и обработки сенсорных данных**

По архитектурным решениям, применяемым при организации хранения и обработки сенсорных данных, можно выделить следующие классы:

- (1) внешнее хранение и обработка данных вне сенсорной сети;
- (2) распределенное внутрисетевое хранение и обработка;
- (3) комбинированные решения.

Первая модель рассматривает сенсорные данные как непрерывный поток, который без потерь накапливается в сенсорной сети и затем передается и архивируется за пределами сенсорной сети. Однажды собранные, данные могут храниться в различных хранилищах, в том числе в традиционных базах данных, и запросы к ним формируются с помощью стандартных методов. Такую модель реализует большое количество уже развернутых сетей, предназначенных для мониторинга каких-либо явлений. Такая сеть легко разворачивается, но она может оказаться коротко живущей, особенно в случае, когда используются сенсоры с высокой скоростью передачи данных, такие как камеры, акустические или вибрационные сенсоры и т.д., так как требования к передаче данных сильно превышают доступные энергетические ресурсы.

Вторая модель работы с сенсорными данными — это рассматривать сенсорную сеть как распределенную базу данных, которая поддерживает запросы на предоставление и обработку данных. В такой модели запрос направляется внутрь сенсорной сети, возможно, к удаленным сенсорным узлам, которые хранят свои данные локально. Такая архитектура потенциально является более энергоэффективной, так как обработка запроса выполняется на сенсорном узле, и происходит передача результата запроса, а не сырых данных. Затраты энергии на сетевую передачу большого объема данных превосходят затраты на их локальную обработку.

### **2.1. Внешнее хранение и обработка сенсорных данных**

При централизованном методе обработки и хранения данные с сенсорных узлов собираются в системе хранения на сервере, находящемся вне сенсорной сети. Рассмотрим возможные способы организации систем хранения.

### 2.1.1. Реляционные базы данных

Для хранения данных, в том числе данных большого объема, по-прежнему активно используют реляционные базы данных. В качестве примера можно привести статистику [3] (со ссылкой на исследование: IDC and Computerworld BI and Analytics Survey Research Group IT Survey, 2012, N=111), по которой более 70% организаций США, внедряющих у себя решения в области Big data, пользуются РБД.

Тем не менее реляционные БД не позволяют эффективно работать с упорядоченным множеством элементов временного ряда и не являются оптимальным вариантом для высокопроизводительных систем хранения и обработки сенсорных данных. Действительно, сама структура РБД рассчитана на другой тип данных и другой тип взаимосвязи между данными. Поэтому накладные расходы на реляционный сервис в этом случае оказываются неоправданными.

### 2.1.2. In-memory технологии

Быстрый рост внедрения наблюдается у технологий in-memory вычислений и хранения данных. Технологии In-Memory Data Grid (IMDG) и In-Memory Data Base (IMDB) используют оперативную память компьютера (RAM) в качестве места хранения данных.

В отличие от размещения традиционных баз данных в памяти на RAM-диске (что в разы увеличивает производительность работы с данными), технологии in-memory подразумевают применение специально разработанных решений для оптимизации операций с данными в памяти, что позволяет повысить производительность чтения данных в десятки раз, а производительность записи — в сотни раз.

По отчетам [4] компании McObject LLC (разработчика известной IMDB eXtremeDB) тестирование производительности eXtremeDB показало практически линейную масштабируемость базы данных до размеров свыше 1 Тб (на 160 ядрах) без видимых ограничений для дальнейшего масштабирования. На простых запросах производительность достигла порядка 90 млн. запросов в секунду для собственного API и в 3 раза меньшую производительность для SQL-подобного eXtremeSQL.

Высокая производительность IMDB привлекает к себе внимание разработчиков всех приложений, работающих с большими объемами данных, для которых важно время отклика. Поэтому большое количество компаний-разработчиков, в том числе таких как Bell Labs, Microsoft, IBM, Oracle и т.д., предлагают свои решения для IMDB [5].

Технология IMDG отличается от IMDB тем, что она предполагает распределенное хранение и обработку данных, причем обработка данных происходит на узлах, где они хранятся, что снижает расходы на передачу по сети и повышает безопасность.

Хотя развитие микроэлектроники ведет к падению цен на память и повышению емкости памяти, тем не менее in-методу технологии еще нельзя назвать широко доступными. Все-таки мощные серверы, многоядерные процессоры и большие объемы оперативной памяти, требуемые для in-методу вычислений стоят недешево. Однако эта область признается аналитиками как одна из перспективных, и доля компаний, использующих этот подход, будет расти.

В плане работы с сенсорными данными in-методу технологии представляют большой интерес для приложений реального времени, осуществляющих мониторинг объектов или окружающей среды, и работающих с большими объемами живых (свежих, поступающих в режиме реального времени) данных. Также возможны комбинированные схемы, в которых архивное хранилище использует дисковую память, а в RAM обрабатываются и хранятся более свежие данные. В этом случае снижаются требования к объему RAM и снижается риск потери данных при аварийной остановке сервера.

### 2.1.3. *Нереляционные распределенные базы данных*

Это направление получило бурное развитие после публикации спецификаций двух хранилищ, разработанных большими интернет-компаниями для собственных нужд: Google BigTable [6] и Amazon Дупано [7]. В настоящее время насчитываются десятки нереляционных распределенных баз данных, наиболее известные из которых Apache HBase [8], Zvents Hypertable [9], Apache Cassandra [10], Apache CouchDB [11], Basho Technologies Riak [12].

Данный класс хранилищ использует в качестве аппаратной платформы сети недорогих компьютеров и обеспечивает возможность масштабирования системы на десятки тысяч узлов.

В распределенной системе реализация типичных для RDBMS надежных транзакций со свойствами ACID (атомарности, согласованности, изолированности и живучести) оказывается затруднительна. В то же время, для многих приложений (к которым относится и хранение сенсорных данных) надежные транзакции не являются необходимыми. Поэтому на базовом уровне распределенные хранилища не предоставляют надежных транзакций, хотя и могут их поддерживать в виде дополнительного сервиса.

Отказ от надежных транзакций позволяет получить взамен не только масштабируемость без потери эффективности, но также повышенную доступность системы (availability) и возможность сохранения работоспособности при нарушении связности системы (partition tolerance).

Отсутствие гарантий согласованности несколько усложняет задачу прикладного программиста. В частности, проблема восстановления согласованности после работы системы в состоянии нарушенной связности (split brain), не имеющая удовлетворительного решения на системном уровне, решается выносом на прикладной уровень — предоставлением прикладному программисту конфликтующих версий объекта данных для слияния их в единую согласованную версию.

Еще несколько свойств систем этого класса:

- возможность репликации данных (с конфигурируемым коэффициентом репликации) для обеспечения доступности данных на чтение при нарушении связности системы и живучести данных при отказах оборудования. Репликация происходит автоматически и обеспечивает высокую отказоустойчивость системы;
- возможность инкрементного масштабирования — добавления в работающую систему дополнительных узлов для увеличения объема и производительности существующего хранилища. Пропускная способность для операций чтения и записи растет линейно с добавлением новой машины (узла);
- гетерогенность — возможность присутствия в системе узлов с разной производительностью (объемом накопителя, другими ресурсами), при этом система балансирует нагрузку между узлами пропорционально их ресурсам;
- децентрализация управления — все узлы равноценны, отсутствует «главный» узел, играющий роль координатора. Это свойство повышает живучесть системы (нет единой точки отказа) и упрощает ее сопровождение. Этим свойством обладает Amazon Dynamo и ее последователи, но не Google BigTable, в архитектуре которой имеется выделенный узел — «сервер метаданных».

Распределенные базы данных хорошо сочетаются с работающими на той же аппаратной платформе моделями распределенной обработки данных: модель MapReduce и надстройки над ней (Apache Pig, Apache Hive) или более современные модели распределенной обработки (Apache Spark).

В сочетании с параллельной архитектурой аппаратной платформы БД этого класса представляют собой мощный инструмент для обработки и хранения большого количества данных.

#### 2.1.4. Циклические базы данных

Циклические базы данных (RRDTool [13], YAWNDB [14]) характерны тем, что они имеют постоянный объем и содержат данные за определенный промежуток времени, например 1 год. При поступлении свежих данных происходит «сдвиг», самые старые данные удаляются из базы. Для экономии пространства более старые данные могут агрегироваться — храниться со снижением разрешения.

Система RRDTool является наиболее распространенным примером циклических баз данных, но ее архитектура изначально не ориентирована на большие объемы живых данных (большое число датчиков). База для каждого датчика хранится в отдельном файле, при постоянной записи большого числа файлов ограничением становится производительность файловой системы.

Более современная система YAWNDB реализована как inmemory база данных для хранения временных рядов. Все данные хранятся в памяти (откуда могут быть быстро выданы по запросу клиента), но также периодически записываются на диск, что повышает шансы сохранить данные в случае аварийной остановки сервера.

#### 2.1.5. Базы данных временных рядов

Базы данных временных рядов (TSBD — time series database) разработаны специально для хранения сенсорных данных. TSBD предоставляют пользователям сервис по хранению, удалению, обновлению и модификации временных рядов. Реализуется поддержка некоторых вычислений над временными рядами (например, умножение, сложение и другие манипуляции, позволяющие преобразовать один или несколько рядов в новый ряд). Есть инструменты, позволяющие выполнять фильтрацию по произвольным образцам, при этом значения одного ряда могут являться фильтром для другого. Язык обработки запросов имеет простой синтаксис.

Базы данных временных рядов Open TSDB [15], InfluxDB [16], Geras [17], имеют свободную лицензию.

Общими чертами различных TSDB являются:

- хранение временных рядов без потери разрешения;
- высокая частота записи (до миллисекунд);

- сырые данные никогда не удаляются;
- масштабирование до миллионов записей в секунду;
- горизонтальное масштабирование (увеличение емкости хранилища добавлением узлов);
- возможность агрегирования данных;
- предоставление GUI для построения графиков;
- возможность чтения/записи по протоколу HTTP(S).

Укажем характерные особенности каждой из рассматриваемых баз данных временных рядов:

- **Open TSDB:** возможность выбора из нескольких фронт-ендов с открытым исходным кодом; работает поверх Hadoop и HBase; текстовый в стиле telnet протокол для взаимодействия с пользователем; HTTP API для привязки к внешним приложениям (системы мониторинга, приборные панели и т.п.).
- **InfluxDB:** агрегирование данных на лету; может работать как дисковая, так и в памяти; реал-таймовая и пакетная обработка; автоматическое фоновое выполнение непрерывных запросов; несколько способов записи данных: клиентские библиотеки, коллектор данных, интеграция со сторонними приложениями; управляемые политики хранения.
- **Geras:** может работать как облачный сервис, позволяющий быстрое наращивание емкости хранилища без прерывания работы сервиса; поддерживаются форматы для отправки, получения и сбора данных: HTTPS, JSON, RESTful, SenML, MQTT, HyperCat; полукоммерческая — бесплатная версия доступна без соглашения об уровне сервиса; выбор страны хранения данных.

Базы данных временных рядов обладают свойствами, дающими возможность использовать их в качестве высокопроизводительного архивного хранилища сенсорных данных. Реализованные поверх распределенных баз данных TSDB обладают высокой масштабируемостью и отказоустойчивостью. Выбор конкретной базы данных может зависеть от специфики реализации приложения, использующего сенсорные данные.

### 2.1.6. *Онлайн-сервисы для хранения временных рядов*

Интернет-сервисы для хранения и обработки сенсорных данных как правило представляют собой облачный сервис, который позволяет пользователю посылать в облако свои данные непосредственно от источников, хранить их, управлять жизненным циклом этих данных,

делать запросы к сервису, пользоваться предоставляемым API для различных видов обработки данных и их отображения в графическом виде.

Сервис работает в многопользовательском режиме, предоставляя каждому пользователю возможность безопасной работы с его набором данных. Различные сервисы придерживаются разной политики в отношении платы за сервис: ограничивается объем бесплатно хранимых данных, либо пользователь освобождается от платы за сервис при условии, что он делает свои данными публично доступными, либо сервис предоставляется бесплатно без гарантии об уровне сервиса и т.п.

Наиболее известные сервисы: Xively [18], TempoIQ [19], Nimbits Public Cloud [20].

Сервисная платформа Xively включает сервис каталогов (поиск объектов по каталогам и управление правами доступа), сервис хранения данных (хранение архивов временных серий), бизнес-сервис (инсталляция, активация и управление приборами), канал для сообщений (управление и маршрутизация сообщений в режиме реального времени), службу доверия, управляющие приложения, основанные на веб-технологиях. Платформа предоставляет инструменты разработчика, управляющую консоль для устройств — поставщиков данных, CRM (Customer relationship management) сервис и т.д.

Xively — открытая платформа с возможностью выбора формата взаимодействия. Xively поддерживает много вариантов комбинаций ПО и аппаратуры для создания пользовательских решений: существуют официальные реализации библиотек Xively для ряда языков программирования и платформ, таких как Objective C, C, Java, JavaScript, Ruby и т.д. Xively RESTful API поддерживает JSON, XML и CSV форматы данных.

Инструментальный набор разработчика Xively позволяет осуществлять интерактивную разработку и отладку продукта, включая мониторинг в реальном времени http сообщений, API для создания запросов, отслеживание и визуализацию.

Облачная инфраструктура, построенная на собственном сервисе данных Gravity data service, поддерживает сотни миллионов устройств.

Для обеспечения безопасности Xively использует стандартный защищенный протокол HTTPS при сборе информации. Права доступа к данным, хранящимся в Xively, тонко настраиваются.

Обеспечивается возможность использования сторонних устройств, приложений и сервисов. Список аппаратных платформ, поддерживаемых Xively, расширяется постоянно.

С помощью сервиса Xively были реализованы проекты по сбору добровольцами данных об уровне радиации в Японии после аварии в Фукусиме [21], о состоянии окружающего воздуха в различных регионах мира [22] и др. Данные со счетчиков собираются на серверах Xively и отображаются на публично доступных картах в режиме реального времени.

В другой сервисной платформе TempoIQ отказались от Hadoop в пользу создания собственного механизма хранения с большим коэффициентом сжатия. Пользователь может комбинировать, агрегировать и формировать сводные данные по своему желанию. Также в TempoIQ реализованы «виртуальные сенсоры», которые поставляют данные, производные от тех, которые собраны от настоящих сенсоров, и хранятся в БД.

В настоящее время разработчики TempoIQ ведут работу над реализацией прогнозирования и принятия решений (в сотрудничестве с компанией Numenta, выполняющей разработки в области искусственного интеллекта). Цель данных работ — формулировка для пользователя предложений, что они должны делать исходя из полученных данных.

Платформа Nimbits предоставляет веб-сервис, который обрабатывает данные с гео-или временными метками, предоставляет инструменты для расчетов, статистики, уведомлений и др. Сервис может быть установлен на системы на основе Debian, т.е. Raspberry Pi, Ubuntu Server, или облачные сервисы. Сервис реализован на основе Java библиотек с открытым кодом.

На базе Nimbits реализован облачный сервис Nimbits Public Cloud, который функционирует в настоящее время, и можно протестировать его работу, зайдя на него со своим Google Account.

Пользователь может сконфигурировать поведение Nimbits при поступлении новых данных в БД, например можно отфильтровать шумы, послать уведомление по электронной почте, если значение параметра превышает заданные пределы, выполнить вычисления.

Интернет-сервисы для хранения и обработки сенсорных данных активно развиваются и используются в настоящее время в связи с появлением концепции «Интернета вещей». Данный подход базируется на масштабируемом распределенном хранении данных и является

одним из самых эффективных способов хранения и обработки сенсорных данных. Хранилище «как сервис» и услуги по обработке и анализу данных «как сервис» позволяют поставщикам данных не разворачивать собственную инфраструктуру для хранения и обработки данных. Кроме того, появляется важная возможность использовать уже имеющиеся данные, предоставленные другими поставщиками в публичное пользование.

## 2.2. Распределенное внутрисетевое хранение

Еще одно направление в развитии систем хранения сенсорных данных рассматривает сенсорную сеть не только как место сбора данных, но и как распределенное хранилище этих данных. Работы в этом направлении нацелены на создание схемы хранения, которая позволила бы минимизировать энергетические затраты в сенсорной сети на пересылку, хранение и обработку данных. Этот подход особенно актуален для сенсорных сетей, в которых сенсорные датчики и узлы автономны (тем более если они расположены в труднодоступных для обслуживания местах) и вопрос энергосбережения стоит на первом месте, поскольку от него напрямую зависит длительность работы автономных источников питания узлов сети.

Применение на практике этого подхода в последнее десятилетие стало возможным благодаря развитию микроэлектроники и появлению энергоэффективных флэш-памяти и процессоров, размещаемых на сенсорных платформах. В исследовании [23] показано, что новое поколение флэш-памяти объемом в несколько Гб и высокопроизводительные процессоры делают локальное хранение в 2 раза более энергоэффективным, чем передачу данных по сети, и обеспечивают сравнимую стоимость вычислений.

### 2.2.1. Системы со сбором данных по запросу

Один из способов экономии энергии сенсорного узла — инициировать сбор данных только по запросу. По умолчанию узел находится в спящем состоянии. Такой подход реализован в системах класса AQP (Acquisitional Query Processing).

Известный пример AQP-системы — TinyDB [24]. TinyDB — распределенный обработчик запросов, который запускается на всех узлах сенсорной сети. TinyDB работает на сенсорной платформе Berkeley mote поверх операционной системы TinyOS [25]. TinyDB имеет как

много черт традиционных обработчиков запросов (возможность обработки запросов «select», «join», «project» и агрегирования данных), так и возможности, разработанные специально для экономии энергопотребления с помощью метода сбора данных по запросам.

Описываемый подход применен в многочисленных сенсорных сетях по мониторингу окружающей среды. Автономные сенсорные узлы в режиме постоянного функционирования израсходовали бы энергию батареек в течение нескольких дней. Применение техники сбора данных по запросам позволяет продлить время жизни таких узлов на месяцы и годы. Например, сенсорная платформа Mica, которая проводит только 2% времени в день в активном режиме, функционирует 6 месяцев на энергии двух батареек типа AA.

TinyOS не обеспечивает возможностей операционной системы в привычном понимании (изоляция процессов и управление ими, управление памятью, многопоточность и т.д.), она представляет собой библиотеку, которая предоставляет ряд удобных программных абстракций, включающих компоненты: модуляции сигналов для передачи по радио, чтения сенсорных данных для различных аппаратных платформ, синхронизации часов между отправителем и получателем, переключения аппаратной платформы в низкопотребляющее состояние. Однако даже с помощью этих абстракций разворачивание и настройка сенсорной сети требует написания программного кода в тысячи строк, что сдерживает широкое внедрение сенсорных сетей. Применение декларативной модели, предложенной разработчиками TinyDB, уменьшает объем необходимого программного кода до нескольких коротких утверждений на простом языке для каждого приложения, работающего с сенсорными данными.

Для сенсорного узла в TinyDB хранится таблица, каждая строка которой соответствует определенному моменту времени, а столбцы содержат значения собираемых параметров: температуру, влажность и т.д. Суть метода сбора данных по запросу заключается в том, что табличные значения «материализуются» (т.е. данные реально собираются с датчиков) только по запросу и хранятся, как правило, в течение короткого периода времени (до отправки за пределы сенсорной сети). Исходный запрос передается от приложения на базовую станцию сенсорной сети, где обрабатывается с составлением плана выполнения запроса, оптимизированного с точки зрения энергопотребления, и затем передается в узлы в виде бинарного кода.

Подход, реализованный в TinyDB, позволил организовать эффективную работу по сбору живых данных в сетях с ограниченными энергетическими ресурсами. Правда, вопрос архивирования данных, их длительного хранения и последующей аналитической обработки вынесен за рамки этого подхода.

### 2.2.2. Распределенная внутрисетевая база данных

Еще одно направление в методах хранения сенсорных данных предлагает использовать ресурсы сенсорных узлов для хранения архива данных, собранных этим узлом, и их обработки.

В работе [26] описана разработка data-ориентированной базы данных StonesDB, архитектура которой базируется на локальном хранении данных и обработке запросов на сенсорных узлах. Основные позиции: использование флэш памяти для локального хранения данных на сенсорных узлах, обработка на узлах сложных запросов и поддержка как традиционных запросов в стиле SQL, так и запросов в стиле data mining (интеллектуальный анализ данных).

Иерархическая структура внутрисетевой (внутри сенсорной сети) базы данных основана на иерархической структуре самой сенсорной сети. Нижний слой — локальные базы данных, работающие на «интеллектуальных» сенсорных узлах, верхний слой — слой управления — на богатом ресурсами прокси-сервере, на котором хранятся агрегированные данные, выполняются первичная обработка запросов и взаимодействие с локальными хранилищами на узлах.

Ключевые компоненты локального хранилища: механизм запросов, составляющий энергоэффективный план выполнения запросов и представления результатов с доверительным интервалом; агрегирование данных с различными уровнями разрешения; механизм старения данных, в соответствии с ограничениями флэш памяти узла. Конкретная реализация этих механизмов зависит от используемой аппаратной платформы сенсорных узлов. Для ограниченных в ресурсах (память и производительность процессора) архитектур работает минималистская версия, которая поддерживает простой декларативный интерфейс запросов и метод «наивного» старения данных. Обладающие большими ресурсами платформы поддерживают более богатый набор запросов и более изощренные техники хранения и индексирования.

В локальном хранилище данные могут храниться различными способами:

- как непрерывный поток, доступ к данным в котором осуществляется методом последовательного сканирования;
- индексная структура поверх потока данных;
- представления данных с меньшим разрешением.

При обработке запроса сначала сканируется более грубое представление данных для исключения подмножества, которое не может быть интересным, и для определения релевантного подмножества. Затем сканируется полное представление последнего подмножества.

Верхний слой управления распределенными данными содержит 2 ключевых элемента:

- кэш, который содержит агрегированные данные с низкого уровня слоя (т.е. образ данных с более низким разрешением). Также кэш содержит некоторые данные с сенсорных узлов, необходимые для обработки запросов.
- механизм обработки запросов, который определяет, как обрабатывать каждый входящий запрос. Запрос может обработаться локально на прокси с использованием кэшированных данных или дополнительных данных, запрошенных с узлов, либо запрос переправляется на узел для полной или частичной обработки; в случае частичной обработки прокси обрабатывает запрос из кэшированных данных, затем передает узлу для дальнейшей обработки и уточнения.

Для решения вопроса с каким разрешением делать агрегирование StonesDB адаптивно рассчитывает баланс между стоимостью передачи суммаризации с узла на прокси и стоимостью получения недостоверных ответов из-за низкого разрешения.

Предложенный подход позволяет использовать ресурсы сенсорных узлов для хранения и обработки сенсорных данных и представляется интересным для определенного круга задач. Однако в случае, если стоит задача хранить все данные с датчиков без потерь, в том числе без потерь разрешения и устаревания данных, этот подход неприменим. Все же хранить значительный объем данных без использования диска в настоящее время невозможно.

## **Заключение**

Современное разнообразие подходов к задачам хранения и обработки сенсорных данных свидетельствует как об актуальности этих

задач, так и об их многообразии. Специфика задачи определяет необходимое количество и частоту опроса датчиков, срок хранения данных и методы их обработки. С одной стороны, это поощряет дальнейший расцвет разнообразных подходов: для каждой задачи подбирается (или вновь разрабатывается) наиболее эффективное решение. Но в то же время есть и обратная тенденция: появление онлайн-сервисов для хранения временных рядов, которые специализируются на сборе, хранении и обработке чужих сенсорных данных и предлагают универсальное решение для всего многообразия задач. Ситуация напоминает историю развития сервиса e-mail: времена, когда каждый провайдер, каждая организация имели свой почтовый сервер уходят в прошлое; пользователи предпочитают облачные почтовые сервисы, такие как gmail, несмотря на их недостатки, такие как засоренность интерфейсов рекламой и сомнения в их конфиденциальности.

Онлайн-сервисы в сочетании с интернет-технологиями сбора данных способны обеспечить пользователю желаемый результат при минимуме затрат: достаточно подключить датчик к Интернет, соблюсти протокол, и вот уже данные собираются, надежно хранятся и постоянно доступны для анализа. Решается и проблема стартовых затрат: нет единоразовых вложений, оплата плавно растет по мере роста интенсивности использования сервиса.

В простейшем случае владелец датчика сам является и пользователем информации, но это не обязательно всегда будет так. Можно предугадать формирование рынка сенсорной информации с разделением ролей участников: поставщики данных, сервис хранения и обработки, и пользователи (приложения, использующие сенсорную информацию).

Дополнительные возможности для сервиса открываются в связи с тем, что в его распоряжении оказывается огромный объем разнообразных данных. Это позволит использовать накопленные данные не только для выявления временных тенденций в поведении отдельных наблюдаемых объектов, но и для интеллектуального анализа данных (data mining) в рамках концепции Big data.

Также проведенный анализ показал, что большинство решений в области хранения сенсорных данных являются разработками зарубежных компаний и университетов. Популярные облачные сервисы по хранению сенсорных данных располагают свои сервера за пределами России. Создание онлайн-сервисов хранения и обработки сенсорных данных в России является актуальной задачей ближайшего будущего.

## Список литературы

- [1] J. Gantz, D. Reinsel. *The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East–United States*, IDC Country Brief, February 2013 (data of access: 13.03.2015), URL: <http://www.emc.com/collateral/analyst-reports/idc-digital-universe-united-states.pdf><sup>↑157</sup>
- [2] V. Turner, J.F. Gantz, D. Reinsel, S. Minton. *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*, April 2014 (data of access: 13.03.2015), URL: <http://www.emc.com/leadership/digital-universe/2014iview/index.htm><sup>↑157</sup>
- [3] D. Vesset. *Transforming Massive Data into Better Outcomes. Big Data Trends and Opportunities* (data of access: 13.03.2015), URL: [http://www.afceabethesda.org/PDF%20Files/Vesset\\_Presentation\\_3282012.pdf](http://www.afceabethesda.org/PDF%20Files/Vesset_Presentation_3282012.pdf)<sup>↑160</sup>
- [4] Benchmark Report Scaling the eXtremeDB® In Memory Database System (IMDS) Beyond the Terabyte Size Boundary (data of access: 13.03.2015), URL: <http://www.mcobject.com/terabyte-plus-benchmark><sup>↑160</sup>
- [5] List of inmemory databases — Wikipedia, the free encyclopedia (data of access: 13.03.2015), URL: [http://en.wikipedia.org/wiki/List\\_of\\_inmemory\\_databases](http://en.wikipedia.org/wiki/List_of_inmemory_databases)<sup>↑160</sup>
- [6] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, R.E. Gruber. *Bigtable: A Distributed Storage System for Structured Data* (data of access: 13.03.2015), URL: <http://goo.gl/YAG4Pq><sup>↑161</sup>
- [7] Amazon DynamoDB Documentation (data of access: 28.10.2015), URL: <https://aws.amazon.com/documentation/dynamodb/><sup>↑161</sup>
- [8] Apache HBase™ Reference Guide (data of access: 13.03.2015), URL: [http://hbase.apache.org/book.html#\\_preface](http://hbase.apache.org/book.html#_preface)<sup>↑161</sup>
- [9] Hypertable — Big Data. Big Performance (data of access: 13.03.2015), URL: <http://hypertable.org/><sup>↑161</sup>
- [10] The Apache Cassandra Project (data of access: 13.03.2015), URL: <http://cassandra.apache.org/><sup>↑161</sup>
- [11] Apache CouchDB (data of access: 28.10.2015), URL: <http://couchdb.apache.org/><sup>↑161</sup>
- [12] Big Data Platform — Enterprise Distributed NoSQL & Object Storage — Basho (data of access: 28.10.2015), URL: <http://basho.com/><sup>↑161</sup>
- [13] RRDtool — About RRDtool (data of access: 13.03.2015), URL: <http://oss.oetiker.ch/rrdtool/><sup>↑163</sup>
- [14] YAWNDB. Time Series Database Written in Erlang (data of access: 13.03.2015), URL: <http://kukuruku.co/hub/erlang/yawndb-time-series-database><sup>↑163</sup>

- [15] OpenTSDB — A Distributed, Scalable Monitoring System (data of access: 13.03.2015), URL: <http://opentsdb.net/><sup>↑163</sup>
- [16] InfluxDB (data of access: 13.03.2015), URL: <http://influxdb.com/><sup>↑163</sup>
- [17] Geras TSDB — The 1248 time-series database (data of access: 13.03.2015), URL: <http://1248.io/geras.php><sup>↑163</sup>
- [18] Xively by LogMeInXively (data of access: 13.03.2015), URL: <https://xively.com/><sup>↑165</sup>
- [19] TempoIQ — IoT Application Platform (data of access: 13.03.2015), URL: [https://www.tempoiq.com](https://www.tempoiq.com/)<sup>↑165</sup>
- [20] Nimbits (data of access: 13.03.2015), URL: <http://www.nimbits.com/><sup>↑165</sup>
- [21] R. Courtland. *Radiation Monitoring in Japan Goes DIY*, Radiation Monitoring in Japan Goes DIY — IEEE Spectrum, 25 Mar 2011 (data of access: 13.03.2015), URL: <http://spectrum.ieee.org/tech-talk/energy/environment/radiation-monitoring-in-japan-goes-diy><sup>↑166</sup>
- [22] Air quality egg. Community-led sensing network (data of access: 13.03.2015), URL: <http://airqualityegg.com/><sup>↑166</sup>
- [23] G. Mathur, P. Desnoyers, D. Ganesan, P. Shenoy, “Ultra-low power data storage for sensor networks”, *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, IPSN’06 (April 19–21, 2006, Nashville, Tennessee, USA), pp. 374–381.<sup>↑167</sup>
- [24] S. R. Madden, M. J. Franklin, S. M. Hellerstein, W. Hong. “TinyDB: An Acquisitional Query Processing System for Sensor Networks”, *ACM Transactions on database systems*, **30**:1, pp. 122–173 (data of access: 13.03.2015), URL: <http://db.cs.berkeley.edu/papers/tods05-tinydb.pdf><sup>↑167</sup>
- [25] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. C. K. Pister, “System architecture directions for networked sensors”, *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS IX (November 12–15, 2000, Cambridge, MA, USA), pp. 93–104.<sup>↑167</sup>
- [26] Y. Diao, D. Ganesan, G. Mathur, P. Shenoy, “Rethinking Data Management for Storage-centric Sensor Networks”, *Proceedings of the Conference on Innovative Data Systems Research*, CIDR’07 (January 7–10, 2007, Asilomar, CA, USA), 2007, pp. 22–31.<sup>↑169</sup>

Рекомендовал к публикации

чл.-корр. РАН С. М. Абрамов

*Об авторах:*



**Елена Васильевна Шевчук**

Старший научный сотрудник Исследовательского центра мультипроцессорных систем Института программных систем имени А.К. Айламазяна РАН. Область интересов: распределенные вычисления, сенсорные сети, мониторинг и управление территориально распределенными объектами

*e-mail:*

[shev@shev.botik.ru](mailto:shev@shev.botik.ru)



**Юрий Владимирович Шевчук**

Зав. лабораторией телекоммуникаций Института программных систем имени А.К. Айламазяна РАН, к.т.н. Область интересов: системное программирование, цифровая электроника, сети компьютеров, сенсорные сети, мониторинг и управление территориально распределенными объектами

*e-mail:*

[shevchuk@botik.ru](mailto:shevchuk@botik.ru)

*Пример ссылки на эту публикацию:*

Е. В. Шевчук, Ю. В. Шевчук. «Современные тенденции в области хранения и обработки сенсорных данных», *Программные системы: теория и приложения*, 2015, **6:4(27)**, с. 157–176.

URL:

[http://psta.psiras.ru/read/psta2015\\_4\\_157-176.pdf](http://psta.psiras.ru/read/psta2015_4_157-176.pdf)

Elena Shevchuk, Yury Shevchuk. *Modern trends in sensor data storage and processing.*

ABSTRACT. The review contains overview of modern trends in sensor data storage and processing. A number methods are considered from the standpoint of applicability for scalable, high performance data storage. The most promising solutions are highlighted taking into account specificity of sensor data. (*In Russian*).

*Key Words and Phrases:* time series, scalable distributed data storage, online services of data storage.

## References

- [1] J. Gantz, D. Reinsel. *The digital universe in 2020: Big Data, Bigger Digital Shadows, and Biggest Growth in the Far East—United States*, IDC Country Brief, February 2013 (data of access: 13.03.2015), URL: <http://www.emc.com/collateral/analyst-reports/idc-digital-universe-united-states.pdf>
- [2] V. Turner, J.F. Gantz, D. Reinsel, S. Minton. *The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things*, April 2014 (data of access: 13.03.2015), URL: <http://www.emc.com/leadership/digital-universe/2014iview/index.htm>
- [3] D. Vesset. *Transforming Massive Data into Better Outcomes. Big Data Trends and Opportunities* (data of access: 13.03.2015), URL: [http://www.afceabethesda.org/PDF%20Files/Vesset\\_Presentation\\_3282012.pdf](http://www.afceabethesda.org/PDF%20Files/Vesset_Presentation_3282012.pdf)
- [4] Benchmark Report Scaling the eXtremeDB® In Memory Database System (IMDS) Beyond the Terabyte Size Boundary (data of access: 13.03.2015), URL: <http://www.mcobject.com/terabyte-plus-benchmark>
- [5] List of inmemory databases — Wikipedia, the free encyclopedia (data of access: 13.03.2015), URL: [http://en.wikipedia.org/wiki/List\\_of\\_inmemory\\_databases](http://en.wikipedia.org/wiki/List_of_inmemory_databases)
- [6] F. Chang, J. Dean, S. Ghemawat, W.C. Hsieh, D.A. Wallach, M. Burrows, T. Chandra, A. Fikes, R.E. Gruber. *Bigtable: A Distributed Storage System for Structured Data* (data of access: 13.03.2015), URL: <http://goo.gl/YAG4Pq>
- [7] Amazon DynamoDB Documentation (data of access: 28.10.2015), URL: <https://aws.amazon.com/documentation/dynamodb/>
- [8] Apache HBase™ Reference Guide (data of access: 13.03.2015), URL: [http://hbase.apache.org/book.html#\\_preface](http://hbase.apache.org/book.html#_preface)
- [9] Hypertable — Big Data. Big Performance (data of access: 13.03.2015), URL: <http://hypertable.org/>
- [10] The Apache Cassandra Project (data of access: 13.03.2015), URL: <http://cassandra.apache.org/>
- [11] Apache CouchDB (data of access: 28.10.2015), URL: <http://couchdb.apache.org/>
- [12] Big Data Platform — Enterprise Distributed NoSQL & Object Storage — Basho (data of access: 28.10.2015), URL: <http://basho.com/>
- [13] RRDtool — About RRDtool (data of access: 13.03.2015), URL: <http://oss.oetiker.ch/rrdtool/>

- [14] YAWNDB. Time Series Database Written in Erlang (data of access: 13.03.2015), URL: <http://kukuruku.co/hub/erlang/yawndb-time-series-database>
- [15] OpenTSDB — A Distributed, Scalable Monitoring System (data of access: 13.03.2015), URL: <http://opentsdb.net/>
- [16] InfluxDB (data of access: 13.03.2015), URL: <http://influxdb.com/>
- [17] Geras TSDB — The 1248 time-series database (data of access: 13.03.2015), URL: <http://1248.io/geras.php>
- [18] Xively by LogMeInXively (data of access: 13.03.2015), URL: <https://xively.com/>
- [19] TempoIQ — IoT Application Platform (data of access: 13.03.2015), URL: <https://www.tempoiq.com>
- [20] Nimbits (data of access: 13.03.2015), URL: <http://www.nimbits.com/>
- [21] R. Courtland. *Radiation Monitoring in Japan Goes DIY*, Radiation Monitoring in Japan Goes DIY — IEEE Spectrum, 25 Mar 2011 (data of access: 13.03.2015), URL: <http://spectrum.ieee.org/tech-talk/energy/environment/radiation-monitoring-in-japan-goes-diy>
- [22] Air quality egg. Community-led sensing network (data of access: 13.03.2015), URL: <http://airqualityegg.com/>
- [23] G. Mathur, P. Desnoyers, D. Ganesan, P. Shenoy, “Ultra-low power data storage for sensor networks”, *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks*, IPSN’06 (April 19–21, 2006, Nashville, Tennessee, USA), pp. 374–381.
- [24] S.R. Madden, M.J. Franklin, S.M. Hellerstein, W. Hong. “TinyDB: An Acquisitional Query Processing System for Sensor Networks”, *ACM Transactions on database systems*, **30**:1, pp. 122–173 (data of access: 13.03.2015), URL: <http://db.cs.berkeley.edu/papers/tods05-tinydb.pdf>
- [25] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. C. K. Pister, “System architecture directions for networked sensors”, *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS IX (November 12–15, 2000, Cambridge, MA, USA), pp. 93–104. “Rethinking Data Management for Storage-centric Sensor Networks”, *Proceedings of the Conference on Innovative Data Systems Research*, CIDR’07 (January 7–10, 2007, Asilomar, CA, USA), 2007, pp. 22–31.

*Sample citation of this publication:*

Elena Shevchuk, Yury Shevchuk. “Modern trends in sensor data storage and processing”, *Program systems: theory and applications*, 2015, **6**:4(27), pp. 157–176. (In Russian).

URL: [http://psta.psiras.ru/read/psta2015\\_4\\_157-176.pdf](http://psta.psiras.ru/read/psta2015_4_157-176.pdf)