

УДК 681.3.07

ОБЛАЧНЫЕ АРХИТЕКТУРЫ IOT: BIGDATA И МОНИТОРИНГ ПОДВИЖНЫХ ОБЪЕКТОВ



А.В. Сенько

*Кандидат физико-математических наук,
старший разработчик программного обеспечения в компании Fastdev AB*

*Fastdev AB, Стокгольм, Швеция
E-mail: alexandresenko@gmail.com*

А.В. Сенько

Закончил Минский государственный радиотехнический колледж по специальности «Проектирование и производство РЭС». Обучался в БГУИР на факультете компьютерного проектирования по специальности «Моделирование и компьютерное проектирование РЭС». В 2009 году закончил магистратуру на кафедре антенн и устройств СВЧ, а в 2012 году закончил аспирантуру на кафедре вычислительных методов и программирования. Диссертацию кандидата физико-математических наук по специальности «Радиофизика» защитил в 2013 году. Работал в институте ядерных проблем БГУ лаборатории сильноточной электроники и лаборатории ядерной оптики и космофизики. Работал программистом в компании ISSoft Solutions, (Минск), в настоящее время работает старшим программистом в компании Fastdev AB (Стокгольм, Швеция). Автор 2-х книг, 6 статей и 15 – ти тезисов докладов на международных конференциях. Область интересов – облачные технологии, интернет вещей, BigData, Computer Science, системы моделирование и оптимизации технических объектов. Женат, есть сын

Аннотация. Описаны современные облачные IoT архитектуры современных систем мониторинга автотранспортных средств. Рассмотрены облачные сервисы, позволяющие создавать надежные, масштабируемые и высоконагруженные системы мониторинга и показана экономическая целесообразность использования облачных платформ для развертывания их архитектур.

Ключевые слова: Большие данные, интернет вещей, спутниковый мониторинг, облачные вычисления.

В настоящее время, благодаря широкому развитию беспроводных телекоммуникационных систем, микропроцессорных встраиваемых систем и элементной базы абонентской аппаратуры спутниковых систем навигации, стало возможно создание полномасштабных систем мониторинга местоположения и технического состояния транспортных средств. Сбор и последующий анализ данных с тысяч и миллионов подвижных объектов позволяет оптимизировать затраты на топливо и техническое обслуживание, и повысить их надежность и безопасность путем своевременного выявления и даже предсказания возможной технической неисправности. Кроме того, возможно построение оптимальных транспортных сетей, не только минимизирующих расход топлива, но и максимизирующих среднюю загрузку при одновременной минимизации уровня износа элементов конструкции.

Чтобы достигнуть всего этого, требуется платформа, которая будет способна собирать и анализировать данные с большого количества подвижных объектов – система мониторинга подвижных объектов. Традиционно такие системы реализованы и широко используются для авиационно, морского и железнодорожного транспорта и только сравнительно недавно они стали применяться для мониторинга автомобильного транспорта. Кроме того, традиционно процесс мониторинга местоположения был отделен от процесса анализа технического состояния – телеметрическая информация для важнейших параметров передавалась и анализировалась только на этапе проектирования и испытания новых транспортных средств [1].

Прогресс в области телекоммуникации, элементной базы встраиваемых систем и в области вычислительной техники привел к тому, что уже сегодня созданы системы мониторинга автотранспортных средств, которые отслеживают не только местоположение автомобилей, но и важнейшие параметры их состояния – уровень топлива, операционные параметры двигателя, трансмиссии, приводных агрегатов и пр.

Далее будут рассматриваться системы мониторинга автотранспортных средств, основное отличие которых к примеру от авиационных или морских состоит в том, что измерение навигационной информации производится непосредственно во встраиваемой системе на борту автомобиля и передается для обработки через телекоммуникационную сеть общего пользования [2].

Итак, система мониторинга транспортных средств состоит из следующих основных компонент (рисунок 1):

1. **Система спутниковой навигации**, включая **орбитальную группировку** и **абонентские терминалы**.

Встраиваемая система, размещаемая на транспортном средстве, которая должна содержать как минимум **абонентский терминал системы спутниковой навигации** (именно поэтому такие системы называются «спутниковыми системами мониторинга») и **модуль телекоммуникационного приемопередатчика**, позволяющий передавать информацию по телекоммуникационным сетям. Дополнительно возможно использование **модуля сопряжения с бортовым компьютером транспортного средства** (как правило в виде коннектора к CAN шинам автомобиля) для съема и последующей передачи телеметрической информации или уведомления об изменении состояния технического объекта;

2. **Телекоммуникационная сеть**, обеспечивающая покрытие всей или большей части географической области перемещения автотранспортного средства. В настоящее время в качестве таковой чаще всего используются сотовые сети связи стандарта GSM, 3G, реже 4G / LTE. Кроме того, развитие получают специализированные сети и протоколы, оптимизированные под передачу данных IoT устройств – NB IoT, LoRa и другие. Для передачи информации с автотранспортных средств вне покрытия всех этих сетей используются спутниковые системы – Iridium, "Гонец". В перспективе возможен переход на использование специализированных спутников с глобальным интернет-покрытием, типа OneWeb, а также к межмашинным (M2M) системам передачи данных – 5G.

3. **Вычислительная платформа**, или собственно **платформа мониторинга**, предназначенная для сбора, хранения, анализа и отображения информации, поступающей с транспортных средств. Далее, эту компоненту рассмотрим более подробно.

4. **Подсистема служебных сервисов**, отвечающих за управления самими встраиваемыми устройствами с точки зрения обеспечения их безопасности, удаленного мониторинга, загрузки обновлений ПО, управления подписками пользователей и пр.

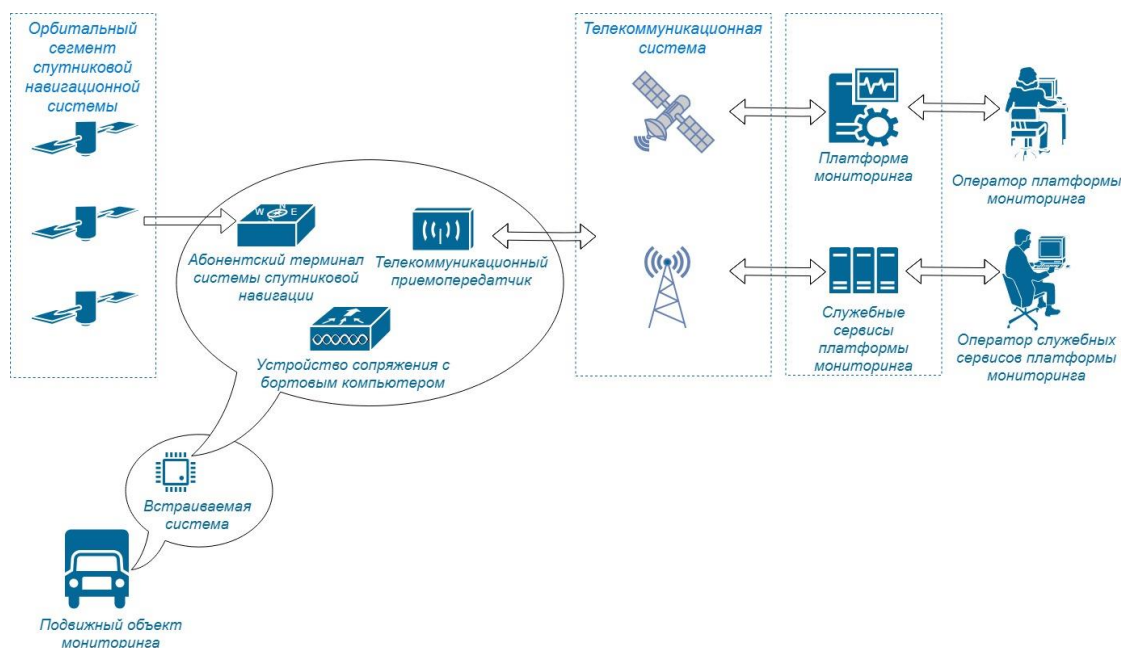


Рисунок 1. Общая структура системы мониторинга автотранспортных средств

Чтобы рассматривать возможные архитектуры вычислительной платформы, необходимо знать с какой информацией она будет иметь дела и о каких масштабах потоков данных и единиц хранения идет речь.

Как и у любого устройства IoT, транспортное средство может быть источником сигналов как минимум 4-х видов:

1. Сигналы о текущем местоположении. Они обязательно включают в себя географические координаты объекта (широта, долгота), опционально – параметры движения (курс, высота, скорость, ускорения).
2. Сигналы телеметрии – отсчеты измеряемой величины, например уровень топлива, давление масла, температура груза, количество оборотов в минуту двигателя и пр.
3. Сигналы о событиях на объекте мониторинга – возникают при смене состояния объекта и отображают конкретные события. Например, включение и выключение двигателя, открытие дверей, заправка или слив топлива и пр.
4. Служебные системы встраиваемой системы мониторинга.

Первые два вида сигналов представляют собой сигналы типа временных рядов, т.е. отсчетов изменяемой во времени величины, которые поступают в телекоммуникационную сеть через регулярные промежутки времени. Все эти сигналы – векторные, т.е. одному временному отсчету соответствует сигнал из набора величин, например, для сигнала о текущем местоположении – идентификатор устройства, временная метка, широта, долгота.

Поскольку сами отсчеты могут содержать погрешности и аномальные ошибки, то для этих величин необходима дополнительная фильтрация. Это может быть сделано лишь частично во встраиваемой системе, поскольку она обладает ограниченными вычислительными ресурсами и чаще всего не хранит историю величин, после их передачи по телекоммуникационной сети.

Кроме того, сама телекоммуникационная система может вносить ошибки и погрешности, проявляющиеся в том, что отсчеты могут прийти с существенной задержкой или в неверном порядке. Фильтрация сигналов для этого случая может быть выполнена только платформой мониторинга с учетом исторических данных и с применением достаточно сложных алгоритмов.

Информация о местоположении, в отличие от телеметрической информации, не может быть агрегирована. Агрегированию подвергаются опциональные параметры движения – средняя скорость, пройденный путь, среднее ускорение при торможении и пр. Возможно лишь частичное сглаживание и аппроксимация, если этого позволяют требования к отображению фактических координат.

Сигналы событий поступают в телекоммуникационную сеть нерегулярно и с гораздо меньшей частотой, чем отсчеты векторов измеряемых величин.

Итак, очевидно, что с каждого объекта мониторинга будут поступать как минимум сигналы о его местоположении с частотой, определяемой требованиями к точности и оперативности отображения этой информации, вплоть до одного сообщения в секунду. И эта информация должна храниться в платформе напрямую без агрегирования.

Чаще всего, системы мониторинга разрабатываются и поддерживаются либо специализированными компаниями, например Gurtam, либо крупными производителями транспортных средств, такими как Volvo, Mercedes, Scania и пр. В обоих случаях количество этих объектов, по мере выпуска новых транспортных средств или подключению к системе уже выпущенных, может расти и расти неравномерно. Например, компания, предоставляющая услуги мониторинга, «подключила» заказчика со значительным автопарком, что приведет к единовременному скачкообразному росту нагрузки.

Таким образом, можно сформировать четкие требования к платформе мониторинга, важнейшие из которых (с точки зрения применения ее по целевому назначению) следующие:

1. Работа нагрузкой, в виде большого количества разнообразных сообщений, поступающих в единицу времени. Эта нагрузка, вследствие особенностей работы телекоммуникационных систем и неравномерности поступления ряда сигналов, имеет флуктуирующий характер;

2. Возможность обработки сообщений «на лету» как с целью их сглаживания (для случая временных рядов), так и с целью непрерывного измерения параметров и определения достижения ими заданных уровней величин. Пример последнего – контроль скорости автомобиля и сообщение о превышении им заданной величины.

3. Возможность реакции на события как генерируемые на самом транспортном средстве, так и получаемые в результате достижения измеряемой величины предельного уровня;

4. Сохранение как сигналов-событий так и сигналов отсчетов местоположения и телеметрии с целью последующей их обработки или работы совместно с подсистемой отображения местоположения транспорта на карте.

Если добавить сюда требования хранения и обработки очень больших масштабов (а речь идет о терабайтах) данных, а также требование к возможности масштабирования, в том числе скачкообразной, то становится понятно, что наиболее естественными средами, для размещения платформ мониторинга, являются облачные среды. Размещение платформ мониторинга в облачных средах обладают следующими преимуществами [3]:

1. Отсутствие необходимости строить собственный ЦОД и привлечения существенных инвестиций в эту область. Крупнейшие облачные провайдеры (AWS, Microsoft Azure, Google, IBM) имеют глобальную сеть ЦОД с многократным резервированием физического оборудования, в том числе сетей питания и сетей передачи данных, а также с встроенной многократной репликацией данных. Этим обеспечивается надежность, недостижимая или крайне трудно достижимая для собственного ЦОД. С целью обеспечения требованиям законодательства, возможно создание гибридных решений, когда критическая информация или пользовательские данные хранятся локально.

2. Гибкость и масштабируемость. Ценовая политика большинства облачных провайдеров обеспечивает возможность оплаты за фактически использованные ресурсы, а резервы

масштабирования для глобальной сети ЦОД таковы, что с точки зрения клиента кажется, что он использует облачную среду, в которой он может «создать» столько вычислительных ресурсов, сколько ему нужно за минимальное время. Сценарии моментального масштабирования в ответ на скачкообразный рост нагрузки – типовой для облачных сред. Верно и обратный случай – де-масштабирование при падении нагрузки, в том числе скачкообразно.

3. Удобство и простота создания масштабируемой архитектуры. Во всех облачных провайдерах реализована платформа, позволяющая быстро создавать как виртуальную инфраструктуру, в виде «виртуальных сетей», файрволов и виртуальных машин – IaaS – Infrastructure as a Service. В крупных облачных провайдерах, реализована платформа, в виде сопрягаемых друг с другом сервисов, каждый из которых представляет собой широко используемый программный продукт (например, реляционную базу данных, файловое хранилище, управляемую очередь сообщений), представляемый в виде полностью управляемого и автоматически масштабируемого сервиса, для которого не нужно создавать экземпляры виртуальных машин – PaaS – Platform as a Service. Некоторые облачные провайдеры предоставляют сервисы, которые включают в себя практически все, что нужно, для создания платформы мониторинга, начиная от библиотек и SDK для встраиваемых систем и заканчивая встроенной службой аналитики телеметрических данных. Наиболее широкое распространение в настоящее время получили решения, созданные на базе первых двух – IaaS и набора сервисов PaaS, а потому сосредоточимся на них. Это обусловлено тем, что они дают наибольшую гибкость и возможность интеграции со сторонними сервисами, что чаще всего требуется для реальных систем, а потому рассмотрим их.

Общая архитектура платформы мониторинга выглядит следующим образом [5], [7], [8] (рисунок 2):

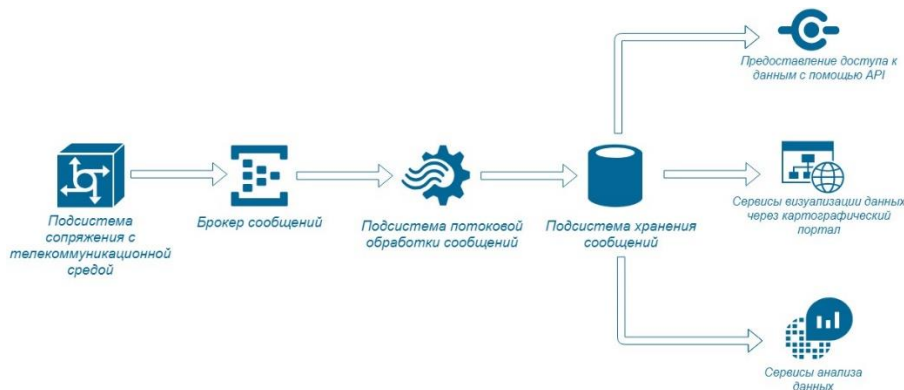


Рисунок 2. Общая архитектура платформы мониторинга

Система сопряжения с телекоммуникационной средой, предназначенная для приема сообщений из телекоммуникационной среды и трансляции их далее в платформу мониторинга. Она может включать в себя конечные точки REST сервисов, MQTT, AMQP и др. В случае REST конечной точки это может быть API шлюз и сервис сообщений в брокер сообщений.

Брокер сообщений (рисунок 3) – предназначен для приема, буферизации и распределения по источникам сообщений, поступивших из системы сопряжения с телекоммуникационной средой.

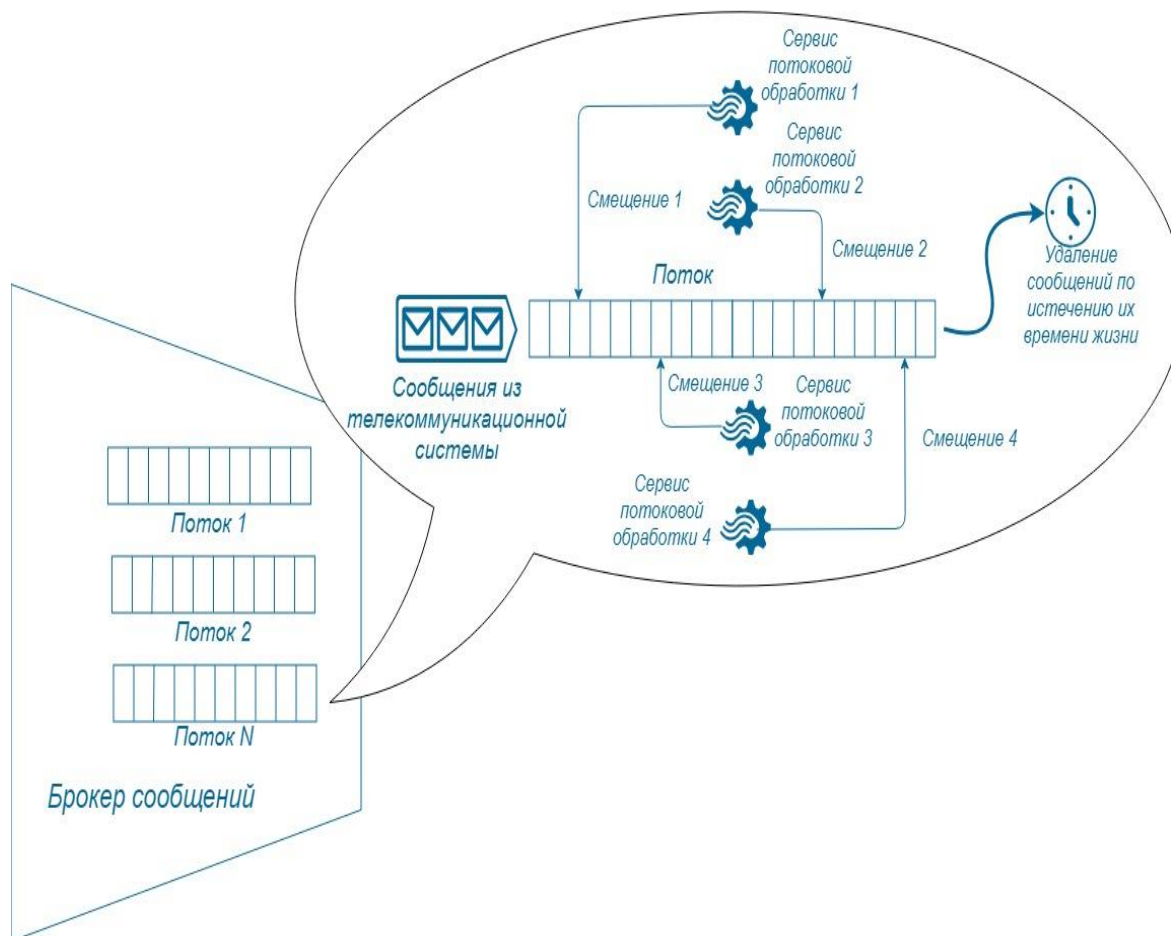


Рисунок 3. Общая структура брокера сообщений

Для обеспечения возможности буферизации сообщений, в нем реализована внутренняя система очередей (или «потоков» - в документации для различных брокеров используется различная терминология). Буферизация нужна для того, чтобы согласовать скорость поступления сообщений со скоростью их обработки последующими сервисами и повысить надежность [11]. В случае отказа сервиса обработки или возникновения в нем ошибки, он может повторно прочитать сообщение из очереди. Каждый такой поток работает по принципу «издатель - подписчик», когда поступившее в очередь сообщение читается несколькими потребителями - «подписчиками» независимо. Для этого каждый «подписчик» должен хранить позицию только-что прочитанного сообщения в потоке. Сами же сообщения не удаляются при чтении (как это имеет место в системах очередей), но удаляются по истечении заданного времени жизни. Одним из самых широко используемых брокеров является OpenSource сервис Apache Kafka. Специализированными брокерами для IoT приложений являются брокеры с поддержкой протокола MQTT – ZeroMQ, RabbitMQ и другие. Для MQTT брокеров возможна двунаправленная коммуникация в рамках одного стека протоколов. В качестве облачных сервисов PaaS можно привести Azure Event Hub (имеет встроенные конечные точки HTTPS и AMQP) и AWS Kinesis Data Streams.

Подсистема потоковой обработки сообщений (рисунок 4) служит для обработки сообщений со скоростью, равной их поступления.

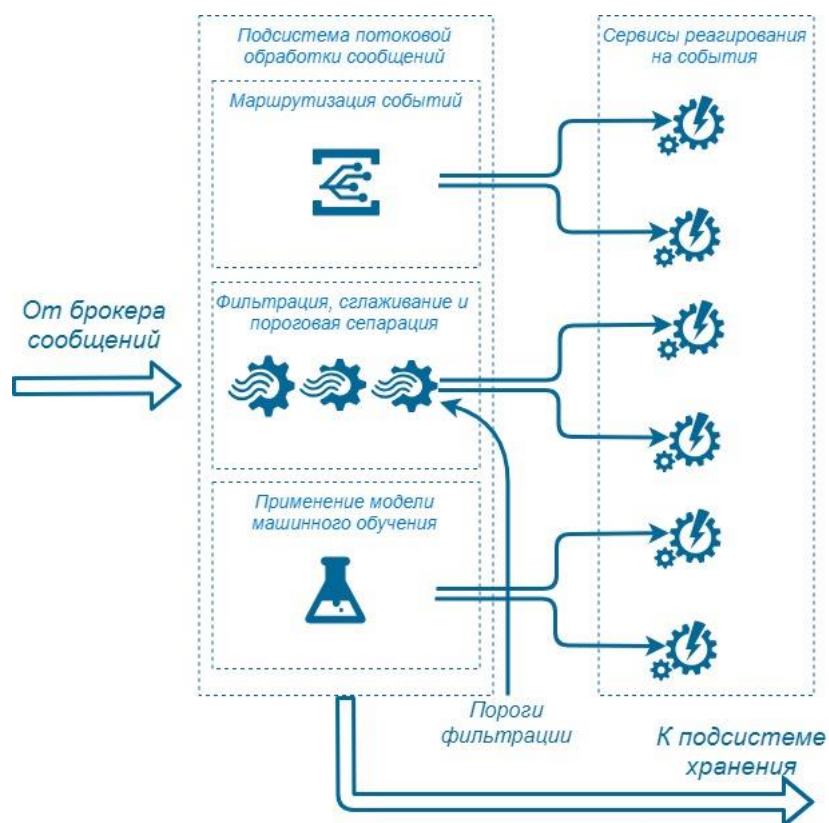


Рисунок 4. Общая структура подсистемы потоковой обработки сообщений

Для телеметрических данных и данных типа временных рядов обработка состоит в фильтрации, сглаживании и непрерывном сравнении параметров вектора величин с пороговыми. Кроме того, возможно непрерывная обработка с применением натренированной модели машинного обучения. Для сообщений типа событий потоковая обработка состоит из маршрутизации по типу в соответствующие **сервисы реагирования на события**. В качестве OpenSource продуктов можно отметить Apache Spark Streaming, представляющий собой программный API, который позволяет выполнять потоковый алгоритм обработки на языках Java, Python, Scala или Python. Еще один широко используемым продуктом является Apache Storm. В качестве PaaS сервисов можно отметить Azure Stream Analytics и AWS Kinesis Stream Analytics, которые предоставляют возможность писать алгоритмы обработки сообщений на языке близком к SQL непосредственно в веб-консоли управления. Эти же PaaS сервисы могут быть использованы и для маршрутизации сообщений, а также для интеграции с сервисами машинного обучения.

По поводу архитектуры подсистемы потоковой обработки сообщений следует сделать замечание - для случая обработки IoT данных с подвижных объектов мониторинга, практически всегда используется потоковая фильтрация сообщений, перед их сохранением в подсистеме хранения. Это обусловлено тем, что “сырые” координаты, получаемые с абонентского терминала системы спутниковой навигации, содержат множество аномальных ошибок, которые дают неверные координаты местоположения. Если терминал имеет возможность дополнительно контролировать свое местоположение, например работает в дифференциальном режиме GPS или использует сигналы телекоммуникационных сетей, то таких ошибок конечно меньше. Но такая ситуация для мониторинга автотранспорта пока не характерна и “сырые” данные всегда с ошибками [4].

В качестве **сервисов реагирования на события** могут выступать сервисы отсылки оповещений операторам (SMS, E-Mail, визуальные оповещения в веб портале и пр.), а также сервисы интеграции со сторонними службами.

Подсистема хранения и анализа сообщений (рисунок 5) служит для хранения поступивших в систему сообщений, а также для обработки их в пакетном или интерактивном режимах.

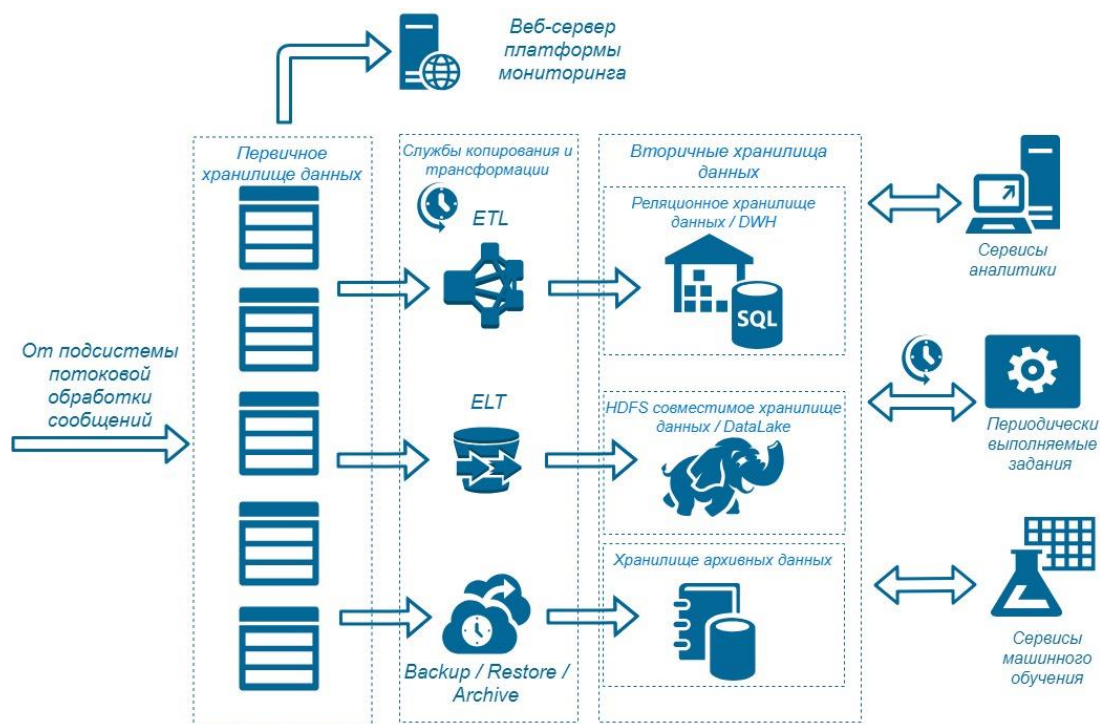


Рисунок 5. Подсистема хранения и анализа сообщений

К хранилищам данных в этой подсистеме предъявляются противоречивые требования. Так, например, с одной стороны, они должны иметь достаточную производительность и масштабируемость, чтобы обеспечивать сохранение всех поступивших сообщений, а с другой - иметь широкие возможности для выполнения аналитических запросов к данным. Если добавить сюда требования минимальной стоимости, а также возможности хранения очень больших объемов данных (терабайты) и предоставления возможности выполнения быстрых запросов к ним, то становится понятным, что одним сервисом хранения данных обойтись очень трудно. Эти противоречия решаются путем разделения одного хранилища на несколько разнотипных, имеющих различное назначение. Чаще всего в качестве **первичного хранилища данных** используется масштабируемое нереляционное (NoSQL) хранилище, например Redis, MemcacheD. Если входящие сообщения имеют формат JSON, то возможно применение документ-ориентированных баз данных, например MongoDB. Все эти хранилища позволяют строить высокопроизводительные службы хранения IoT данных и обеспечивать доступ к ним из сторонних приложений, например картографических веб-приложений или серверов REST API.

Облачные PaaS нереляционных данных достаточно разнообразны. Это как базы данных NoSQL общего назначения, например AWS DynamoDB, AWS DocumentDB, Azure CosmosDB, Azure Table Storage, так и специализированные базы данных для хранения временных рядов, например AWS Timestream и Azure Time Series Insight.

Подобные нереляционные базы данных, как правило, не могут быть использованы для целей выполнения аналитических запросов со сложным соединением данных из разных источников, агрегированием и пр. Поэтому данные из первичных хранилищ, должны быть перемещены во вторичные. Для этих целей служат сервисы трансформации и копирования (ETL / ELT).

Возможны два принципиально различных типа хранилища, допускающие выполнение сложных аналитических запросов - реляционные хранилища - Data Warehouse (DWH) и нереляционные, типа “озеро данных”, DataLake. В случае реляционного хранилища, данные из первичного хранилища преобразуются и приводятся к той схеме данных, которая принята в DWH (чаще всего это денормализованная типа “Звезда” или «Снежинка»). Поэтому процесс переноса данных носит название ETL - Extract Transform Load. Для случая же DataLake, данные переносятся без изменения их схемы, а уже потом подвергаются трансформации, поэтому процесс переноса называется ELT - Extract Load Transform.

Реляционные хранилища имеют в своем составе встроенный аналитический процессор и встроенные средства обеспечения массивно - параллельного выполнения запросов. Наоборот, для DataLake требуется дополнительные сервисы аналитики, например Apache Hive, Apache Spark, Presto и другие. Однако, у DataLake есть существенное преимущество в том, что помимо упрощения переноса данных, имеется возможность применения алгоритмов машинного обучения, встроенных в указанные программные продукты. Кроме того, фреймворк MapReduce и прямой программный доступ к данным позволяет строить сколь угодно сложные алгоритмы обработки данных [10].

Физически, все указанные сервисы размещаются на кластерах, состоящих из нескольких вычислительных узлов. Администрирование подобных систем может быть очень непростым делом, а потому в облачных средах имеются сервисы, упрощающие работу с ними. Так например, AWS и Azure предоставляют сервисы управляемых кластеров, которые упрощают развертывание на них наиболее популярных BigData фреймворков - Azure HDInsight [6] и AWS EMR [9], Для дальнейшего упрощения создания подобных архитектур, облачные провайдеры предоставляют PaaS сервисы как для DWH - Azure SQL DWH, AWS RedShift, так и для DataLake - Azure DataLake Storage / DataLake Analytics и AWS Lake Formation.

Необходимость в архивном хранилище данных обусловлено тем, что первичное хранилище данных не может хранить все сообщения за все время работы, поскольку это экономически нецелесообразно. Вместо этого, оно хранит данные в течении определенного времени (от нескольких месяцев, до нескольких лет), после чего переносит их в более дешевое хранилище архивных данных, из которого, в случае необходимости, они могут быть восстановлены. В ряде случаев, это хранилище может выступать в качестве нереляционного хранилища Data Lake. Так например, хранилище объектов AWS S3, совместно с сервисом аналитики AWS Athena могут использоваться для построения DataLake.

Заключение:

Современные облачные среды позволяют строить масштабируемые, высокопроизводительные архитектуры для систем мониторинга автотранспортных средств. Благодаря возможностям потокового и исторического анализа данных, подобные системы позволяют снизить затраты на эксплуатацию парка транспортных средств как за счет анализа возможностей снижения расхода топлива, так и за счет поиска оптимальных путей технического обслуживания и предсказания технического состояния. Для этого необходим сбор и сохранение телеметрических данных и последующая обработка их сервисами аналитики. В современных облачных средах имеются сервисы, позволяющие максимально упростить создания подобных архитектур, за счет автоматизации их развертывания, мониторинга и масштабирования. Это в свою очередь позволяет минимизировать расходы на создание всей

системы мониторинга и применять самые продвинутые и современные методики анализа без роста сложности системы.

Литература:

- [1] Козырев Н.А. Современная телеметрия в теории и на практике. Учебный курс / Н.А. Козырев, А.В. Назаров, И.В. Шитов, В.П. Обрученков, А.В. Древин. - СПб: Наука и техника, 2007. – 627 с.
- [2] Радиотехнические системы. Под ред. Казаринова Ю.М. – М. Академия, 2008. – 592 с.
- [3] Сенько А. Работа с BigData в облаках. Обработка и хранение данных с примерами из Microsoft Azure. – СПб. Питер, 2019. – 448 с.
- [4] Соловьев Ю.А. Спутниковая навигация и ее приложения. – М. Эко-Трендс, 2003. – 326 с.
- [5] Microsoft Azure IoT Reference Architecture // Microsoft 2018. URL: http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf
- [6] Avkash Chauhan, Valentine Fontama, Michele Hart, Wee Hyong Tok, Buck Woody. Introducing Microsoft Azure HDInsight: Technical Overview // Microsoft, 2014. URL: https://download.microsoft.com/DOWNLOAD/E/7/B/E7B25440-1569-40B5-989E-3951FC178214/MICROSOFT_PRESS_EBOOK_INTRODUCING_HDINSIGHT_PDF.PDF
- [7] Olawale Oladehin, Brett Francis. Core IoT Trends // Amazon Web Services, 2007. URL: <https://d1.awsstatic.com/whitepapers/core-tenets-of-iot1.pdf>
- [8] Olawale Oladehin, Brett Francis, Craig Williams, Catalin Vieru, Philip Fitzimons. AWS IoT Lens. AWS Well Architected Framework // Amazon Web Services, 2018. URL: <https://d1.awsstatic.com/whitepapers/architecture/AWS-IoT-Lens.pdf>
- [9] Parviz Deyhim. Best practice for Amazon EMR // Amazon Web Services, 2013. URL: <https://d0.awsstatic.com/whitepapers/aws-amazon-emr-best-practices.pdf>
- [10] Erik Swensson, Erik Dame, Shree Kenghe. Big Data Analytics Options on AWS // Amazon Web Services, 2018. URL: https://d1.awsstatic.com/whitepapers/Big_Data_Analytics_Options_on_AWS.pdf
- [11] Allan MacInnis, Chander Matrubhutam. Streaming Data solutions on AWS with Amazon Kinesis // Amazon Web Services, 2017. URL: <https://d1.awsstatic.com/whitepapers/whitepaper-streaming-data-solutions-on-aws-with-amazon-kinesis.pdf>

IOT CLOUD ARCHITECTURES: BIGDATA AND MONITORING OF MOBILE OBJECTS

S. V SENKO

*PhD, Senior Software Developer, Fastdev AB,
Stockholm, Sweden*

*Fastdev AB, Stockholm, Sweden
E-mail: alexandresenko@gmail.com*

Abstract. Described IoT cloud architectures for the modern fleet monitoring systems. Considered cloud services, which allow to create reliable, scalable and high-loaded fleet monitoring systems and shown economic efficiency of using cloud platforms for deploying their architectures.

Keywords: Big Data, internet of things, iot, fleet monitoring, cloud computing.