

А.В. СЕНЬКО

ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ IoT ДЛЯ ПОСТРОЕНИЯ СИСТЕМ МОНИТОРИНГА ТЕРРИТОРИАЛЬНО РАСПРЕДЕЛЕННЫХ ПРИРОДНЫХ ОБЪЕКТОВ

*Fastdev AB
Sveavägen, 9, vån 17, Stockholm, 11157, Швеция*

Поступила в редакцию

Аннотация. В статье описано применение современных технологий и сервисов IoT облачного провайдера Microsoft Azure, использованных для построения системы удаленного мониторинга территориально распределенных природных объектов. Описана облачная архитектура системы, приведены примеры использования различных сервисов хранения и анализа сообщений для целей создания системы мониторинга и предотвращения чрезвычайных ситуаций.

Ключевые слова: Интернет вещей, системы дистанционного мониторинга, облачные технологии, Microsoft Azure.

Abstract. In article described application of modern IoT technologies, provided by Microsoft Azure, used for architecting the remote monitoring system for geographically distributed nature objects. Described cloud architecture of system, described examples of using different services for message storing and analysis for purpose of creating monitoring and emergency case preventing system.

Keywords: IoT, remote monitoring systems, Cloud Computing, Microsoft Azure.

Doklady BGUIR. 201*, Vol. *, No. *, pp. **-****

Application of modern IoT technologies for architecting monitoring systems of geographically distributed nature objects

A.V. Senko

Введение

Одним из важнейших факторов минимизации ущерба от природных и техногенных катастроф является фактор времени: чем раньше обнаружен факт наступления предельного состояния техногенного объекта или природного комплекса, тем больше времени будет у служб ГО и ЧС на адекватное реагирование. А поскольку количество потенциальных природных комплексов очень велико и все они географически разнесены, то единственно приемлемыми с экономической точки зрения системами анализа их состояния являются системы дистанционного мониторинга.

Текущее состояние природного объекта может быть определено путем измерения параметров физических величин в конкретных его географических точках (измерительных пунктах) или путем анализа его изображений, полученных с борта летательных или космических аппаратов. Однако построение систем мониторинга традиционно сопряжено с большими техническими трудностями и высокими материальными затратами. Развитие современных технологий интернета вещей и облачных технологий позволяют существенно сократить затраты на создание систем дистанционного мониторинга благодаря использованию готовых сервисов и

компонентов. В данной статье рассказывается как с помощью сервисов облачных сервисов Microsoft Azure можно построить масштабируемую, надежную и защищенную систему мониторинга природных объектов.

Обзор существующих систем мониторинга природных объектов

Все системы дистанционного мониторинга природных объектов можно разделить на 3 большие группы:

1. Системы телесигнализации [2];
2. Дистанционного зондирования Земли (ДЗЗ), включая спутниковые системы и системы на базе пилотируемых и беспилотных ЛА [4];
3. Дистанционного мониторинга параметров состояния окружающей среды, включая состав атмосферы, уровня воды, радиации и пр. [2]

Системы *телесигнализации* относятся к чисто реактивным системам, то есть они реагируют на факт наступления некоего события (например, существенного повышения температуры измеряемого объекта, задымления и пр).

Системы *ДЗЗ* и *дистанционного мониторинга параметров* в ряде случаев могут быть использованы для проактивного реагирования и предотвращения природных катастроф до их фактического возникновения. Для этого, необходимо комплексирование в единую сеть пунктов измерения и центров обработки информации [1].

Системы *дистанционного мониторинга* состоят из сети пунктов наблюдения / замеров параметров физических величин природных полей, значения оценок которых по каналам связи поступают в единый диспетчерский центр (ДЦ) для последующей обработки. На основе их анализа и сравнения с пороговыми уровнями, можно создавать системы оповещения о ЧС. Кроме того, зная характер территориального распределения величин параметров возможно оценить масштабы стихийного бедствия. В тоже время, анализ исторических данных измерений позволяет строить математические модели, на основе которых возможно предсказание будущего состояния природного объекта.

Традиционные системы дистанционного мониторинга создавались с использованием выделенных физических вычислительных средств, сетей передачи данных и аппаратных средств пунктов замеров, конфигурируемые и зачастую создаваемые специально для данной системы. Это требовало больших капитальных затрат и приводило к созданию немасштабируемых, трудных и дорогих в сопровождении систем. Современные облачные среды предоставляют ряд сервисов и SDK для доступа к ним, позволяющих существенно упростить, ускорить и удешевить создание систем мониторинга [3], [5].

Архитектура облачного сервиса мониторинга

Обобщенная архитектура системы мониторинга территориально-распределенных природных объектов, созданную на основе облачных сервисов Azure IoT приведена на рисунке 1 [10], [11]. Сигналы с электронных измерительных датчиков, поступают в каналы аналоговой обработки, которая может включать в себя фильтрацию сигнала, усиление или аттеюацию, нормализацию, синхронное детектирование и пр. [6]. После прохождения тракта аналоговой обработки, сигналы подаются на вход АЦП встраиваемой микропроцессорной системы.

Программа, запущенная в виде фонового процесса операционной системы (не обязательно операционной системы реального времени) встраиваемой системы пункта

замера, периодически считывает показания с АЦП, отправляя результаты измерений в виде сообщений JSON в облачный сервис Azure IoT Hub.

В настоящее время имеется достаточно много аппаратных платформ, совместимых с IoT сервисами Microsoft Azure, например Raspberry Pi, Arduino, Espressif ESP32 и другие [7], [8], [9]. В качестве сети в IoT устройствах может быть использована мобильная сеть 2G / 3G / 5G, NB IoT, специализированная IoT сеть типа LoRa WAN или локальная сеть, соединенная с интернет - шлюзом, например Wi-Fi, Bluetooth или Zigbee.

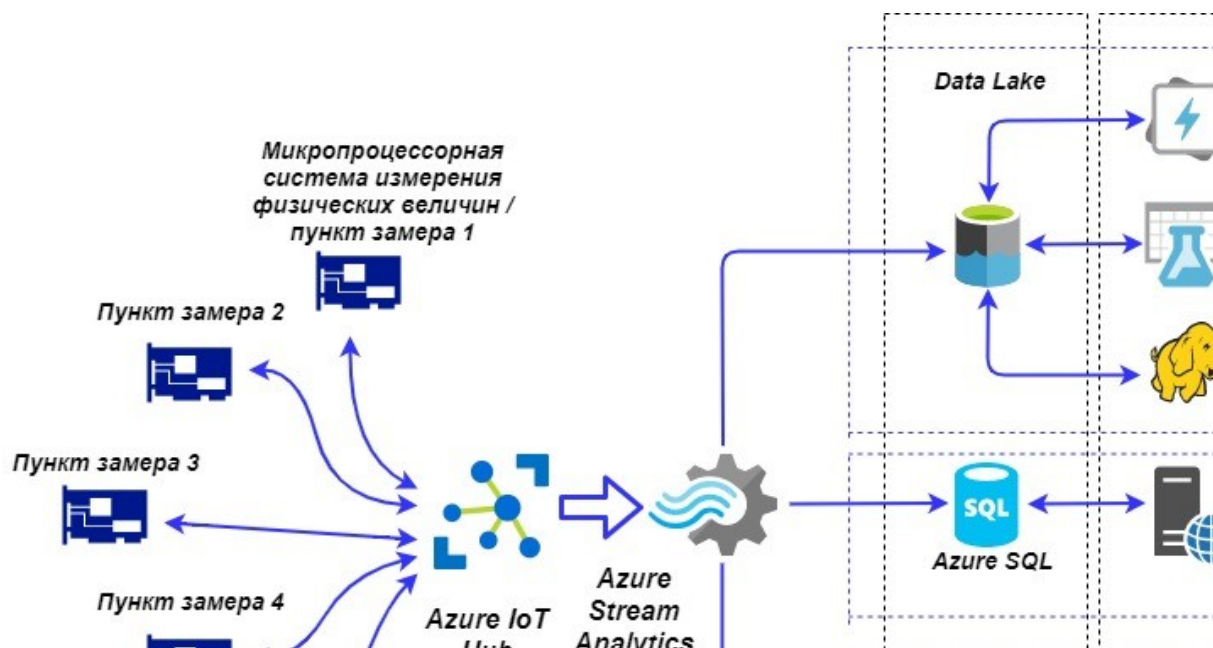


Рисунок 1. Архитектура системы мониторинга территориально-распределенных природных объектов, созданная на базе облачных сервисов Azure IoT.

Для того, чтобы пункты замеры могли удаленно взаимодействовать с центром обработки данных, в современных системах используют специальные протоколы, наиболее распространенным из которых является MQTT [12].

Azure IoT Hub является брокером сообщений, предоставляющим защищенный доступ к MQTT топикам, обеспечивающий авторизацию и аутентификацию устройств, а также хранение их учетных данных и метаданных.

Протокол MQTT обеспечивает взаимодействие между программными клиентами на основе модели “издатель - подписчик” (pub / sub, publisher subscriber). Программные клиенты могут выступать в качестве издателей, посылая при этом сообщения в топики, или подписчиком, принимая сообщения из топиков. Каждый клиент может посылать сообщения в один или несколько топиков, в то же время каждый клиент может быть подписан на один или несколько топиков. Также, клиент может выступать только подписчиком или только издателем. MQTT брокер представляет программным клиентам адреса конечных точек топиков и обеспечивает надежное хранение сообщений, доступных программным клиентам (рисунок 2).

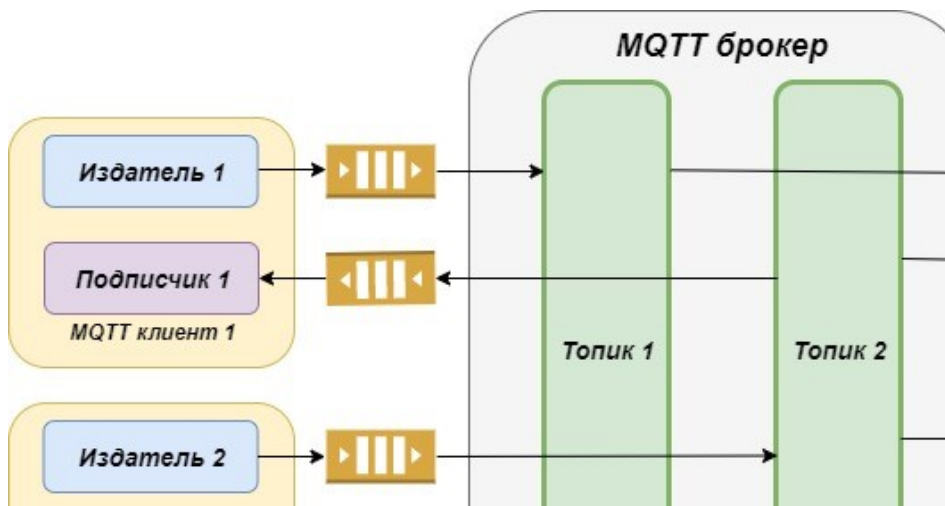


Рисунок 2. Архитектура обмена сообщениями между клиентами на основе протокола MQTT

Фрагмент кода Python программы программного MQTT клиента, использующий Azure IoT SDK, показан на рисунке 3:

```
import os
from azure.iot.device import IoTHubDeviceClient, Message

def get_data_from_sensor():...
def process_exception(e):...

def mqtt_client_main():
    try:
        client_cnn_string = os.environ["MQTT_BROKER_CNN_STRING"]
        client = IoTHubDeviceClient.create_from_connection_string(client_cnn_string)

        while True:
            sensor_data = get_data_from_sensor()
            message = Message(sensor_data)

            print( "Message content: {}".format(message) )
            client.send_message(message)

            sleep_delay = int(os.environ["SLEEP_DELAY"])
            time.sleep(sleep_delay)

    except Exception as e:
        process_exception(e)

if __name__ == '__main__':
    mqtt_client_main()
```

Рисунок 3. Пример кода программного клиента, посылающего сообщения в Azure IoT Hub сообщения в формате JSON и использующего строку подключения для аутентификации.

Сообщения с измерительных пунктов, в формате JSON, поступают в сервис Azure IoT Hub, который играет роль брокера сообщений. Он предоставляет защищенный доступ к MQTT топикам, обеспечивает авторизацию и аутентификацию устройств, хранение их учетных данных и метаданных.

Потоковая обработка сообщений в облачной системе мониторинга

Сообщения, принятые сервисом Azure IoT Hub, могут быть подвергнуты потоковой обработке, которая может включать в себя: осреднение показателей измеряемой

величины в течении заданного временного интервала, агрегирование, отсеечение аномальных значений, непрерывное применение к сообщениям натренированной машинным обучением модели и распределении самих сообщений в зависимости от их содержания по разным направлениям. В облачной платформе Microsoft Azure это реализуется при помощи сервиса Azure Stream Analytics, подключенного к выходу Azure IoT Hub. Сервис Azure Stream Analytics представляет собой непрерывный фильтр, алгоритм работы которого описывается при помощи синтаксиса ANSI SQL. Каждый источник входных сообщений (например MQTT топик сервиса Azure IoT Hub) в SQL выражении интерпретируются как таблица, с которой можно выполнять различные операции.

Все входные сообщения поступают в подсистему хранения данных, состоящую из хранилищ трех типов:

- Очереди сообщений для подсистемы оповещения, построенной на основе сервиса Azure Service Bus;
- Хранилища оперативной информации, построенного на основе экземпляра сервиса реляционной базы данных Azure SQL;
- Хранилища исторической информации, построенного на основе HDFS совместимого сервиса Azure Data Lake Store.

The screenshot shows the Azure Stream Analytics Query Editor. On the left, there is a sidebar with 'Inputs (1)' containing 'device-telemetry' and 'Outputs (5)' containing 'archive-data-lake', 'archive-table', 'dashboard', 'over-temperature', and 'over-temperature-dashboard'. The main area displays a SQL query:

```

1 SELECT * INTO dashboard FROM [device-telemetry]
2 SELECT * INTO [archive-table] FROM [device-telemetry]
3 SELECT * INTO [archive-data-lake] FROM [device-telemetry]
4
5 SELECT * INTO [over-temperature]
6 FROM [device-telemetry]
7 WHERE temperature > 100
8
9
10 SELECT * INTO [over-temperature-dashboard]
11 FROM [device-telemetry]
12 WHERE temperature > 100

```

Below the query, there is a section for 'Input preview' and 'Test results'. It shows a table of events from 'device-telemetry' with columns 'temperature', 'humidity', and 'EventProcessedAt'.

temperature	humidity	EventProcessedAt
34.853649814865456	74.39830277983393	"2019-11-29T21:
28.83502078909903	70.18926069639588	"2019-11-29T21:

Рисунок 4. Окно редактора непрерывного потокового фильтра. Слева видны выходы, к которым подключен сервис Azure IoT Hub, справа - сам выполняемый запрос и десериализованные данные, поступающие на вход.

Подсистемы обработки и анализа данных системы мониторинга

Подсистема рассылки сообщений служит для оповещения потребителей о превышении измеряемыми величинами заданных порогов. Очередь сообщений в ней необходима для того, чтобы синхронизировать скорость отсылки сообщений со скоростью их поступления.

Подсистема отображения оперативной обстановки необходима для визуализации информации о текущем состоянии наблюдаемого природного комплекса. Для этого сообщения из сервиса Azure Stream Analytics, должны сохраняться в “горячем” хранилище (реляционная – Azure SQL - или нереляционная – CosmosDB - база данных), которое оперативно обновляется новыми сообщениями и также оперативно обновляет визуальное представление. Если для отображения информации используется сервис PowerBI, то он подключается к сервису Azure Stream Analytics напрямую.

Подсистема анализа исторических данных предназначена для долговременного хранения истории значений физических величин, отосланных пунктами замера с последующим ее изучением. В качестве долговременного хранилища может выступать как реляционное (Azure SQL DWH) так и нереляционное (Azure Data Lake) хранилища данных.

Подсистема анализа данных включает в себя сервисы, которые выполняют программный анализ данных, сохраненных в подсистемах хранения. Самый сложный и комплексный случай - подсистема анализа исторических данных, которая может включать в себя [3], [5]:

- Пакетный анализ и трансформацию, включая ETL, выполняемый, например, сервисом Azure Data Lake analytics. Этот вид анализа запускается периодически и служит для регулярной агрегации, группировки и фильтрации данных;
- Пакетную процедуру перетренировки моделей машинного обучения, нужную для “подстройки” моделей под изменившиеся входные данные;
- Выполнение аналитических запросов научным персоналом в интерактивном режиме с целью отыскания новых закономерностей в данных.

Подсистема рассылки оповещений реализована на основе сервиса Azure Logic App, позволяющего автоматизировать выполнение рабочих процессов анализа входящих сообщений о превышении пороговых значений и определения канала рассылки оповещений.



Рисунок 5. Реализация логики рабочего процесса подсистемы рассылки оповещений, созданной на основе сервиса Azure Logic App.

Отфильтрованные сообщения, поступившие в очередь сообщений Azure Service Bus, запускают выполнение экземпляра Azure Function, представляющей собой код, исполняемый в безсерверном окружении. Этот код должен иметь доступ к хранилищу состояний оповещений, показывающих отправлено ли оповещение реципиенту или

нет. Если сообщение еще не было отправлено или используется более сложная логика обработки (отправлять сообщение каждые пол часа до тех пор, пока состояние природного объекта не изменится), то запускается экземпляр сервиса Azure Logic App, осуществляющий отправку оповещения.

Также, экземпляр сервиса Azure Stream Analytics Job может быть размещен непосредственно на аппаратной платформе, на которой будет также реализовываться логика подсистемы рассылки оповещений. Для этого может быть использован сервис Azure IoT Edge, разворачиваемый на аппаратуре пункта замера. В случае отсутствия связи с Azure IoT Hub, этот сервис может хранить в себе исторические данные мониторинга и отправлять их в хаб как только появится соединение.

Заключение

В статье описано построение облачной архитектуры системы мониторинга природных объектов с использованием сервисов облачного провайдера Microsoft Azure. Показано, что для создания ключевых элементов подобной системы не требуется разработка с нуля, развертывание на отказоустойчивых кластерах и сложное конфигурирование программных продуктов. Относительная простота конфигурирования облачных сервисов позволяет существенно снизить затраты на разработку и эксплуатацию системы мониторинга в целом.

Список литературы

1. Главный информационно-аналитический центр Национальной системы мониторинга окружающей среды Республики Беларусь (ГИАЦ НСМОС) // Официальный портал. [Электронный ресурс]. URL: <http://www.nsmos.by>.
2. 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.
3. Облачные архитектуры IoT: BigData и мониторинг подвижных объектов / BIG DATA and Advanced Analytics = BIG DATA и анализ высокого уровня: сб. материалов V Междунар. науч.-практ. конф. (Республика Беларусь, Минск, 13–14 марта 2019 года). В 2 ч. Ч. 1 / редкол. : В. А. Богуш [и др.]. – Минск: БГУИР, 2019. – 403 с.
4. Подсистема Белорусской космической системы дистанционного зондирования Земли. Мониторинг земельного фонда /Николай Бобер [и др.]// Земля Беларуси – 2018. – № 2. – С. 23-24.
5. Сенько А. Работа с BigData в облаках. Обработка и хранение данных с примерами из Microsoft Azure. – СПб. Питер, 2019. – 448 с. 9
6. Хоровиц П., Хилл У. Искусство схемотехники: Пер. с англ. - Изд. 2-е. - М.: Издательство БИНОМ . -. 2014. - 704 с., ил. ISBN 978-5-9518-0351-1.
7. Azure Certified for IoT device catalog [Электронный ресурс]. URL: <https://catalog.azureiotsolutions.com/alldevices>.
8. Azure IoT Device SDKs Platform Support. // Microsoft Azure IoT Hub documentation. [Электронный ресурс]. URL: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-device-sdk-platform-support>.
9. Azure IoT Git repo // APIs, SDKs and open source projects from Microsoft Azure. [Электронный ресурс]. URL: <https://github.com/Azure/iot>.
10. Azure IoT reference architecture // Microsoft Azure architecture center. [Электронный ресурс]. URL: <https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot/>.

11. Microsoft Azure IoT reference architecture // e-book, version 2.1. [Электронный ресурс] . URL: http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf.
12. MQTT community official page. [Электронный ресурс]. URL: <http://mqtt.org/>.

References

1. Glavnij informatsionno-analititseskij tsentr Natsionalnoj sistemi monitoringa okruzajuschej sredi Respubliki Belarus (GAITZ NSMOS) // Oficialnij portal. URL: <http://www.nsmos.by>.
2. Kozirev N.A. Sovremennaja telemetrija v teorii i na praktike. Uchebnij kurs / N.A. Kozirev, A.V. Nazarov, I.V. Schitov, V.P. Obruchenkov, A.V. Drevin. - SPb: Nauka I tehnika, 2007. – 627 s.
3. Oblatsnije architekturi IoT: BigData i monitoring podvijnich objectov / BIG DATA and Advanced Analytics = BIG DATA i analiz visokhogo urovnja: sb. materialov V Mezhdunarodnoj nautsno-praktitseskoj konferentsii (Respublika Belarus, Minsk, 13–14 marta 2019 goda). V 2 ch. Ch. 1 / redcollegija: V. A. Bogush [i dr.]. – Minsk: BSUIR, 2019. – 403 p.
4. Podсистема Belorusskoj kosmitseskoj sistemi distantsionnogo zondirovanija Zemli. Monitoring zemeljnogo fonda /Nikolaj Bober [i dr.] // Zemlja Belarusi – 2018. – № 2. – p. 23-24.
5. Senko A. Rabota s BigData v oblakah. Obrabotka i chranenije dannich s primerami iz Microsoft Azure. – SPb. Piter, 2019. – 448 s.
6. Horwitz P., Hill W. Isskustvo skhemotehniki: Perevod s anglijskogo. - Izd. 2. - M.: Izdatelstvo BINOM . -. 2014. - 704 pages., ill. ISBN 978-5-9518-0351-1.
7. Azure Certified for IoT device catalog. URL: <https://catalog.azureiotsolutions.com/alldevices>.
8. Azure IoT Device SDKs Platform Support. // Microsoft Azure IoT Hub documentation. URL: <https://docs.microsoft.com/en-us/azure/iot-hub/iot-hub-device-sdk-platform-support>.
9. Azure IoT Git repo // APIs, SDKs and open source projects from Microsoft Azure. URL: <https://github.com/Azure/iot>.
10. Azure IoT reference architecture // Microsoft Azure architecture center. URL: <https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot/>.
11. Microsoft Azure IoT reference architecture // e-book, version 2.1. URL: http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf.
12. MQTT community official page. URL: <http://mqtt.org/>.

Сведения об авторах

Сенько Александр Васильевич, кандидат физико-математических наук, старший разработчик программного обеспечения, Фастдев АБ.

Information about the authors

Senko A.V., PhD, senior software engineer, Fastdev AB

Адрес для корреспонденции

147 45, Швеция,
Тумба, Минтвэген, 12,
Фастдев АБ
тел. +46729226011;
e-mail: alexandresenko@gmail.com
Сенько Александр Васильевич

Address for correspondence

147 45, Sweden, Tumba, Myntvägen, 12,
Fastdev AB
tel. +46729226011;
e-mail: alexandresenko@gmail.com
Senko Alexandre