Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 55-11 (2022) 25–30

Architectures of cost-effective system for COVID-19 patient monitoring

Eng. Angel Ivanov*, Kocho Hrisafov*, Assoc. prof. Eng. Nayden Chivarov*,

Prof. Ivana Budinska**

**Institute of Information and Communication Technologies, Bulgarian Academy of Sciences Sofia, Bulgaria*

(e-mail: angel.ivanov@iict.bas.bg; kocho.hrisafov@iict.bas.bg; nayden.chivarov@iict.bas.bg)

*** Institute of Informatics, Slovak Academy of Sciences Bratislava, Slovakia*

(e-mail: budinska@savba.sk)

Abstract: In this paper are presented a Proof of Concept (POC) architectures of a cost-effective system for COVID-19 patient monitoring. While the hardware used by the system remains the same, two different approaches are shown and compared. This gives freedom and flexibility to hospitals and/or healthcare practitioners to choose with budget and available IT support in mind.

Keywords: COVID-19, patient monitoring, cost-effective, cyber-physical system, internet of things, low-cost, open source

1. INTRODUCTION

The COVID-19 pandemic became the new "normal" way of living. According to statistical data from the World Health Organization (WHO, 2022), the virus's spreading is far from over. With new variants emerging, the need for telemedicine is constantly rising (Abedini, Calton, & Fratkin, 2020). It is becoming even more important to provide health care to the public and at the same time to protect the healthcare practitioners from the disease, to preserve personal protective equipment, and to minimize the impact of patient surges and overcrowding in hospitals (CDC, 2020). In 2020 was noted a significant increase of the telehealth visits compared to 2019. It is important to say that 75% of the U.S. telehealth patients voted for satisfactory and very satisfactory telehealth sessions and treatments, while only 8% of U.S. consumers were unsatisfied with the service (Gartner, 2021). With that said, different medical cyber-physical systems (MCPS) emerged, each with its own opinion on architecture, complexity and overall cost (Dey, Ashour, Shi, Fong, & Tavares, 2018). (Din & Paul, 2019) show in their work that the future of telehealth monitoring systems is tightly coupled with big data analysis. They define medical data processing stages as equal to the processing of any other type of data:

- Data Cleaning – the process of preparing the data to be processed. This includes error detection, duplicate removal etc.
- Data Compression – the process of minimizing the data. This includes time correlation compression, data block compression etc.
- Data Mining – The process of analyzing the prepared data. This includes data characterization, pattern analysis, classification, clustering etc.

It is only natural that big tech companies with medical subsidiaries are prepared for the telemedical needs surge. GE Healthcare's digital centralized monitoring unit (GE Healthcare, 2022) brings monitoring to the next level, simplifying care-team collaboration, introducing mobile visualization, intelligent event notification and advanced analytics. Mindray's M-Connect IT solution (Mindray, 2022) focuses on being a universal and centralized monitoring platform, which can seamlessly integrate medical devices to third party information systems. The outcome is clinical workflow optimization and improved efficiency for the clinical staff. While being great commercial telemonitoring systems, they easily become quite costly. Unfortunately, a lot of hospitals, even in developed countries, struggle with very tight budgets, and with the COVID-19 pandemic, the struggle is even greater. Therefor they are not able to implement such solutions. The situation is the same or worse with hospitals in developing countries.

In this paper we propose two solutions for low and cost-effective system for telemonitoring COVID-19 diagnosed patients. The focus is to keep the budget for the system as low as possible, using off-the-shelf smart hardware and free or open-source software to run the engine.

2. PROOF OF CONCEPT ARCHITECTURES

2.1 Used hardware

The most crucial vital signs that are monitored on COVID-19 patients are:

- Oxygen saturation
- Heart rate
- Temperature

10.1016/j.ifacol.2022.08.043

- Lung operation

For the purpose of this paper and the design of the system we have focused on relatively cheap but still medical grade quality hardware. For oxygen saturation levels and heart rate, we use the widely available pulse-oximeter BM1000C, shown in (Fig.1), by Berry Medical (Berry Medical, 2022).

Figure 1. Berry Medical BM1000C Pulse Oximeter

It is lightweight and easy to use. It has a bright TFT LCD screen, making it easily visible even during bright daylight. The working voltage is between D.C. 2.2V-3.4V and it operates with standard batteries. The oxygen saturation levels (SPO2) that the device supports are in the range of 35-100% with accuracy of ±2% in the 80%-100% range and ±3% in the 70%-79% range. The field tests showed less than 1% error compared to professional medical pulse-oximeters. The pulse rate measurement range is between 25 and 250bpm with accuracy ±2bpm. With its small size it is highly portable and convenient for use. It provides one-touch operation – with a single button touch, the device starts to measure and visualize on its built-in display the measurements of oxygen saturation in the blood and pulse rate. The BM1000C supports Bluetooth Low Energy standard and provides a free mobile application that can visualize all current measurements. It can also store historical data for future reference. The manufacturer supplies freely available documentation about the protocol used by the device, so it can be implemented in complex scenarios.

A wireless thermometer by Tucky, shown in (Fig. 2) (Tucky, 2022) measures the temperature.

Figure 2. Tucky smart thermometer

It provides a Bluetooth interface, hypoallergic patch to attach to the patient's body and does a continuous measurement of the temperature. The body of the thermometer is thin, flexible and biocompatible. It adapts to all body types and should not cause any discomfort. Using the continuous measurements taken from the patient, Tucky can send an alert if the temperature exceeds a certain threshold, set by the healthcare practitioner. It is approved to be used on babies as well. In this case, it also provides monitoring for the baby's sleeping position. It can send an alert if the baby rolls over onto its stomach. Tucky is proud of its low wave emissions. It transmits only 0.05% of the time via Bluetooth Low Energy and the emission power is around 1000x lower than compared with a smartphone. At the same time, it has two antennas, which provide a wider transmission radius of 5-10m. Compared to other wearable thermometers, which have just a single antenna, it has a 2-3x wider signal radius. Tucky comes equipped with a rechargeable battery, which provides more than 5 days of autonomous work and charges quickly with a standard smartphone charger. In hospitals, usually continuous temperature monitoring is delivered by restrictive and invasive devices such as rectal or esophageal probes. Therefore, it is often limited only to general anesthesia and intensive care for emergency situations. In the rest of the cases, the patients are monitored on an ad-hoc basis, a few times per day. This way a fever episode might not be detected quickly enough or at all. Continuous temperature monitoring without human intervention becomes extremely useful to improve the quality of care due to more reliable diagnostics. It also improves the patient's comfort. Budget-wise, it can reduce the cost of hospitalization with early detection of patient's health deterioration.

For the lung operation, we chose Bluetooth-ready digital stethoscope from EKO, shown in (Fig. 3) (Eko Devices Inc., 2022).

Figure 3. Eko Core digital stethoscope

It supports most of the professional diaphragms and provides up to 40x sound amplification. It supports both analog and amplified listening modes. Supplies active noise cancelation for clearer sound reception. Supports wireless listening for easier use with personal protective equipment in COVID zones. It also provides automated cardiac murmur detection with their proprietary AI platform, to speed up medical diagnostics. Based on a 3M Littmann survey (Eko Devices Inc., 2022), nine out of ten healthcare professionals indicate that they can hear more clearly and identify sounds faster with the Eko digital stethoscope than compared with a traditional analog one.

To gather all measurements in one place and to send it towards the central database, a Raspberry Pi is used (Raspberry Pi Foundation, 2022). This is a minicomputer running Linux-based operating system. This unlocks the flexibility and freedom to choose and evaluate several types of open-source software packages to fulfill the goal. The Raspberry Pi has a 64-bit quad core ARM processor running at 1.4GHz, 1GB of LPDDR2 RAM, and supports dual-band 2.4GHz and 5GHz wireless LAN. It has Bluetooth 4.2 and Bluetooth Low Energy support. It has a Gigabit Ethernet over USB 2.0 interface providing up to 300mbit/s throughput. Supports input power via micro-USB connector, via the GPIO header or Power over Ethernet (PoE) with a separate PoE HAT. It has a built-in

HDMI interface, so it can be connected directly to an external screen for real time monitoring and configuration. In our use case the HDMI interface is not being used and the Graphical User Interface on the Linux OS has not been installed. With this approach we reduce the OS footprint on the device, making it more responsive, easier to support, and providing more resources for its primary use as IoT hub and gateway.

The common architectural scheme for both solutions is shown on (Fig. 4).

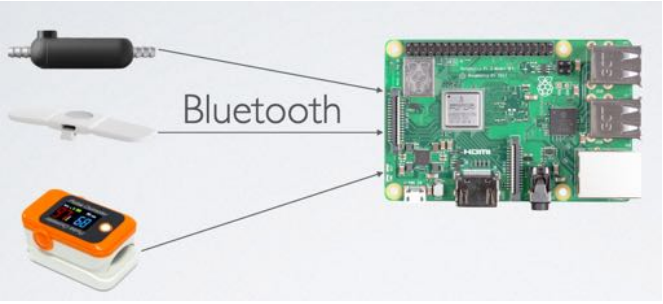

Figure 4. Common architecture for monitoring COVID-19 patients.

On the receiving side, there is a virtual or physical machine with Microsoft SQL Server 2019 Express installed (Microsoft, 2022). This is the database that will host the patient data.

2.2 Full on-premises solution

As shown in (Fig. 4) the Raspberry Pi acts as a Bluetooth gateway for the endpoint devices (stethoscope, thermometer, pulse-oximeter). As such, it is presumed that it will be connected to a network, preferably with internet access – depending on the location of the database. For the sake of being scalable, we will assume that the database is in a different physical location.

Figure 5. Full on-premises solution architecture

On (Fig. 5) is shown the remaining part of the full on-premises solution architecture. Since the idea of the system is of being event driven, it is needed to use an event streaming platform. Event streaming is called of being the digital equivalent of the central nervous system in the human body. In the "always-on" world, where more and more businesses are automated and "software-defined", this is the technological foundation. Event streaming is the workflow starting with capturing data in real-time from various data sources, such as databases, sensors, services etc., storing those events for later retrieval, reacting to events, or processing the data in real-time or retrospectively. It also includes routing of the events to different destination technologies when needed. Therefore, the event streaming ensures a continuous flow and accurate interpretation of data so that right information is at the right place, at the right time. Event streaming is widely used in different business applications. From financial companies to process financial transactions, to logistic monitoring of cars, trucks, or entire fleets; to capturing continuous data from IoT devices and patient monitoring. The event streaming is also the foundation of the microservice-oriented programming. In our case, the event streaming platform is Apache Kafka (Apache, 2022). It is an open-source distributed event streaming platform, which is used by thousands of companies for high-performance data pipelines, streaming services, data integration and mission-critical applications. Apache Kafka is always built as a cluster, even if it's a single or multi-node cluster. This way it provides the possibility to scale the cluster up to thousand brokers, delivering trillions of messages per day in a highly available and stable fashion. The Apache Kafka works as a client-server. In the context of the product, they are called producer and consumer. The producer sends messages in a predefined topic where all consumers, that are interested, are subscribed. In our case the producer is the IoT gateway.

A VPN server from OpenVPN (OpenVPN, 2022) is deployed to secure the communication between the gateway and the database. The OpenVPN was chosen because it is known as the de-facto standard in the open-source networking space. It supports all major operating systems like Windows, Linux, MacOS, as well as the major mobile operating systems like Android and iOS. The OpenVPN protocol provides significant security and is highly adaptable for third-party software. It includes encryption via the OpenSSL library to secure end-to-end connections between different networks. OpenVPN supports the use of both TCP and UDP protocols when transmitting data. The recommended choice is UDP; however, when the UDP protocol it is restricted, then TCP can be utilized. The recommendation from the vendor is to only use TCP when really needed, as is likely to observe performance degradation. The OpenVPN solution supports both client-server and site-to-site configurations. In our case we use client-server. The authentication is secured with a private/public key pair known only to the devices.

As an additional layer of security an SSL certificate is being used for message broker service. This way the information is being encrypted with the SSL certificate and only the intended receiver will be able to decrypt the information.

After the data gets inserted into the database, it is then visualized using a custom-made Graphical User Interface (GUI). There each patient's vital signs are shown. In case of emergency, the system will send back a message to the message broker and the respective medical personnel will be instantly notified to act.

2.3 Cloud-based solution

In the cloud-based solution, the transit part is replaced with cloud-based products. Since we are more familiar with Google Cloud as a cloud provider, we will use their services. There are analogical alternatives in Amazon's AWS; however, they were not evaluated by us. In (Fig. 6) we show the cloud-built part of this architecture. After the data is gathered by the gateway, it's sent towards a Google Cloud function (Google, Cloud functions, 2022). The Google Cloud Functions is a serverless execution environment for building and connecting cloud

services. The main purpose of a Cloud Function is for the developer to write simple, single-purpose functions that are attached to events. These events can originate of existing cloud or on-premises infrastructure, IoT devices, another Cloud Functions, or third-party services. The main advantage is that there is no need to provision any infrastructure for them to run, therefore, no need for infrastructure maintenance.

Figure 6. Cloud part of the cloud-based solution architecture

Cloud Functions can be written using JavaScript, Python 3, Go or Java runtimes on Google Cloud Platform. Another advantage is that a function developer can take the Cloud Function and run it directly on any standard Node.JS, Python 3, Go or Java environment. This makes for an extreme ease for portability and local testing. Being serverless automatically removes the work of managing servers, configuring environment, updating frameworks and OS patching. This work is being done by the cloud provider – in our case Google. The provisioning of the resources is fully automatic and transparent to the developer and happens in a response of an event. This automatically grants the ability of the function to scale from few invocations per day to multimillion invocations without any additional work required. In our case the function translates data into event and submits an event to the message broker.

In Google Cloud, the role of message broker is server by Google Pub/Sub (Google, Pub/Sub, 2022). Pub/Sub allows services to communicate asynchronously. Google claims that the latencies are on the order of 100ms. The Pub/Sub service is used for streaming analytics and data integration pipelines to ingest and distribute data. It is equally effective in the following cases:

- Messaging-oriented middleware – often used for multi-service integration
- Queue – parallelize different tasks

It is worth mentioning that the asynchronous method of work gives more flexibility and robustness of the overall system compared to synchronous RPC calls. This is because the broker sends the event without regards how or when it will be processed. Pub/Sub delivers the events to all internal or external services that need to react to them and does not wait to finish processing synchronously. Common use cases are data streaming from IoT devices, applications or services, ingestion of user interaction and server events, parallel processing and workflows etc.

The Pub/Sub works similarly compared with Apache Kafka. Key difference is the per-message parallelism. Pub/Sub sends individual messages to subscriber clients, then keeps track of whether a given message has been successfully processed or not. In contrast, Apache Kafka uses partition-based parallelism. This forces subscribers to process messages in each partition in order and limits the number of concurrent clients to the number of partitions.

After the message broker broadcasts the message, another cloud function receives it. It has the task to categorize the message's data. If the data contains no urgency, it just forwards it to the database. If there's a need for urgency, a message is sent to Pub/Sub, where another Cloud function reacts to it and pages the respective healthcare practitioner. In order to secure the transmission of data we use the embedded mechanisms provided by Google. We use a dedicated service account with the appropriate permissions. The communication is sent by default over an SSL encrypted channel with additional authentication via private and public key pair. The final piece is the database. In this scenario we are using Google BigQuery (Google, BigQuery, 2022). BigQuery is a petabyte scale, fully managed enterprise data warehouse. It is designed to help manage and analyze data with built-in features like machine learning, geospatial analysis and business intelligence. BigQuery also have serverless architecture – it lets you use standard SQL queries to get information about any question on the stored data with zero infrastructure management. The scalable and distributed analysis engine enables to run queries on terabytes of data in the matter of seconds or on petabytes of data in the matter of minutes.

It is worth mentioning that the usage of all mentioned services from Google cloud are within the free tier.

3. RESULTS

Our expectations were for huge difference performance wise between both architectures. However, the results surprised us. In (table 1) are shown the results from tests being ran on both architectures.

Table 1. Test results from both architectures using one broker and one database

	On-premises	Cloud
Data ingestion (single gateway)	500ms	600ms
Data ingestion (10 gateways)	600ms	600ms
Data ingestion (100 gateways)	2s	650ms
Data ingestion (1000 gateways)	15s	900ms

The tests were conducted in a simulated environment due to the lack of such large number of physical IoT gateway devices. In order to simulate the gateways, application containers were used with similar resource limitation as the IoT gateway in terms of compute power and storage IOPS. For the incoming device data, a sample stream was used and replicated programmatically to meet the gateway count. From (Table 1) we can see that for small number of gateways – around 10-20

there is basically no difference performance wise. For single gateway, the on-premises solution is faster. However, the time difference is so small, that it's negligible. The difference becomes more visible when we dramatically increase the number of gateways. The resource limitation of the SQL Server 2019 Express plays its role and becomes a bottleneck to the whole system. It results in large queues inside the Apache Kafka broker with messages waiting of being processed by the database server. On the other hand, the cloud-based solution almost doesn't suffer from the number of gateways throwing data towards it. Even with 1000 gateways simultaneously sending data, it was loaded into BigQuery in less than a second. The Pub/Sub queues were within normal levels and the Cloud Functions easily scaled the invocation count to support the increased load coming towards them. In (Table 2) is shown another test with multiple SQL Server 2019 Servers ingesting data from multiple Apache Kafka brokers.

Table 2. Tests on multi-on-premises setup compared with Cloud setup

	On-premises	Cloud
Data ingestion (single gateway)	500ms	600ms
Data ingestion (10 gateways)	600ms	600ms
Data ingestion (100 gateways)	900ms	650ms
Data ingestion (1000 gateways)	2s	900ms

The gateways were balanced equally between the Kafka brokers. All databases are stand-alone, all receiving databases replicate asynchronously data with one "master" database, used by the custom Graphical User Interface. As shown, we still see the expected single gateway data ingestion results. However now with a distributed load, the performance is much closer to the respective Cloud measurements.

4. CONCLUSIONS AND FUTURE WORK

In this paper we have set our goals to provide proof of concept architectures for tele-monitoring cyber-physical system. The architectures we propose, are simple enough, robust and easy to maintain. Given the options for entirely on-premises or cloud-based solution, the IT staff at the hospitals are free to choose what works best for their setup and needs.

The results from out tests conclude that in small installations, there are no significant differences in performance. However, in larger installations – for example multiple branches of the same hospital, multiple stay-at-home patients etc., it becomes visible that performance-wise the cloud solution is faster. In order to match the cloud performance, a load balancing method must be implemented with multiple temporary databases which then replicate to a "master". In conclusion it all depends on the IT capabilities and budget of the respective hospital and both architectures will do the work to provide a cost-effective COVID-19 patient monitoring system. Future work includes adding advanced analytics on the measurement data using AI methods, to provide per patient thresholds, predictive alerts, and increased levels of healthcare service.

ACKNOWLEDGMENTS

This research was carried out as part of the bilateral BAS-SAS project: № IC-SK/01/2021-2022 "Cyber-Physical System for smart monitoring and tele-medicine for patient with COVID-19"

REFERENCES

Abedini, N., Calton, B., & Fratkin, M. (2020). Telemedicine in the Time of Coronavirus. *Journal of Pain and Symptom Management, 60*(1), E12-E14.

Apache. (2022). *Apache Kafka*. Retrieved January 2022, from Apache Kafka: https://kafka.apache.org/

Berry Medical. (2022). *Finger Pulse Oximeter*. (BerryMedical) Retrieved January 2022, from Finger Pulse Oximeter: https://shberrymed.com/finger-pulse-oximeter-p00015p1.html

CDC. (2020). *Centers for Disease Control and Prevention*. Retrieved March 2022, from https://www.cdc.gov/coronavirus/2019-ncov/global-covid-19/telehealth-covid19-nonUS.html

Dey, N., Ashour, A., Shi, F., Fong, S. J., & Tavares, J. (2018, March 10). Medical cyber-physical systems: A survey. *Journal of Medical Systems, 42*(74).

Din, S., & Paul, A. (2019, February). Smart health monitoring and management system: Toward autonomous wearable sensing for Internet of Things using big data analytics. *Future Generation Computational Systems, 91*, 611-619.

Eko Devices Inc. (2022). *Eko CORE Digital Attachment*. (Eko Devices Inc.) Retrieved January 2022, from https://shop.ekohealth.com/products/core-digital-attachment?variant=32764121251936

Gartner. (2021). *The rise of telehealth*. (Gartner) Retrieved March 2022, from https://www.gartner.com/en/marketing/insights/daily-insights/the-rise-of-telehealth

GE Healthcare. (2022). *Digital Centralized Monitoring Unit*. (General Electric Healthcare) Retrieved January 2022, from https://www.gehealthcare.com/products/patient-monitoring/connectivity/digital-centralized-monitoring-unit

Google. (2022). *BigQuery*. Retrieved March 2022, from https://cloud.google.com/bigquery

Google. (2022). *Cloud functions*. Retrieved February 2022, from https://cloud.google.com/functions

Google. (2022). *Pub/Sub*. Retrieved February 2022, from https://cloud.google.com/pubsub

Microsoft. (2022). *SQL Server 2019*. Retrieved March 2022, from https://www.microsoft.com/en-US/sql-server/sql-server-2019

Mindray. (2022). *M-Connect IT solution* . (Mindray Bio-Medical Electronics Co., Ltd.) Retrieved January 2022, from

https://www.mindray.com/en/solutions/hospitalwide-solution/

OpenVPN. (2022). *OpenVPN*. (OpenVPN, Inc., Producer) Retrieved January 2022, from https://openvpn.net/

Raspberry Pi Foundation. (2022). *Raspberry Pi 3 Model B+*. Retrieved January 2022, from https://www.raspberrypi.com/products/raspberry-pi-3-model-b-plus/

Tucky. (2022). *Tucky, Smart Wearable Thermometer*. (e-TakesCare) Retrieved January 2022, from https://www.e-takescare.com/en/produit/tucky

WHO. (2022). *World Health Organizatoin*. Retrieved March 19, 2022, from WHO: https://covid19.who.int