Sub-Sahara African Academic
Research Publications
June, 2021, Editions

JPAS
Journal of Pure and Applied Science
Vol..20, No.9, ISSN 2318-9315

PERFORMANCE ANALYSIS OF A CLOUD-BASED FDR

S. N. ARIH1; S. N. UDEH2; & G. N. JOLA3

1Computer Science Programme, National Mathematical Centre, Abuja, Nigeria. 2Computer Sceince, University of Lagos, Lagos. 3Electrical Electronics Engineering Department Federal Polytechnic Bauchi, Nigeria

ABSTRACT

Over the years, there has been the question of improving the mode of collecting flight data record from an aircraft which has a traditional device called the Flight data recorder (FDR) that records both mechanical data, electrical data as well as cockpit voice records popularly known as the "blackbox". It has been a subject of study over the years in research to have flight data records transmitted to a cloud repository in order to have a better-assured way of keeping flight data records in the case of an emergency situation where the data needs to be retrieved. Several research works have been carried out in this area to store flight data records to cloud repositories which grant faster access to such data for analysis. Others also in the areas of data compression in order to optimize the collection of data to the cloud. The only issue here is that these data collected are primarily for investigative purposes in case of a crash or an after flight analysis. None have been able to provide flight assistance using the cloud-based FDR while a flight is still air bound. Here in this work, we have implemented a system that collects flight data to a cloud repository and checks the condition of the flight based on such data in order to provide some form of assistance even while the flight is still air bound. Our system cross-checks the flight data collected and compares it with the active database created which have a set of rules for a normal flight condition and provides a clue to the condition of a flight that is air bound. All flights with an abnormal condition are flagged red and this allows flight control tower administrators to alert the aircraft concerned as well as alert some emergency units for a standby given the location of the flight. The system was implemented using Java Flight

Simulator. The two systems were implemented and evaluated using the metrics such as CPU time, Memory Usage and Bandwidth consumption. The results revealed that the proposed system uses less CPU time utilization of 40.7% as against the existing system's 52.71%. The Memory usage capacity of the proposed system is at 151.35mb which is lesser than the existing system with 194.8mb. The existing system makes use of 1.733mbps bandwidth specification while the proposed system uses 3.55mbps bandwidth specification which makes the proposed system to have a faster data transmission rate.

Introduction

A Flight Data Recorder (FDR) is a replaceable computer element used to record specific aircraft performance parameters in airplanes. It could either be digital or analog depending on the age of the aircraft. The purpose of an FDR is to collect and record data from a variety of aircraft sensors onto a medium designed to survive an accident. Its task is recording pilots' inputs, electronic inputs, sensor positions and information sent to any electronic systems on the airplane Li J. (et. al. 2014). Flight Data Recorder is informally called "black box". They are designed to be quite small and carefully manufactured to withstand the influence of high speed and the heat of an extreme temperature Endre B. (et. Al. 2012).

In fact, FAA regulations stipulate that FDRs retain just the last two hours of recorded information. Thus, many of the recorders have the capacity to store only 2 hours of data. From history, the two types of "flight recorder" carried on aircraft are the FDR and the cockpit voice recorder (CVR). Sometimes both types of the recorder are fitted or combined into a single unit. The design of an aircraft is built in such a way that the Flight Data Recorder, as well as the Cockpit Voice Recorder (CVR), is placed at the tail of the aircraft which has been said to be a most suitable place for its survival in case of a crash according to designers.

Usually, a flight-data acquisition unit (FDAU) receives various discrete, analog and digital parameters from a number of sensors and avionics systems and then routes them to a flight data recorder (FDR) and, if installed, to a Quick Access Recorder (QAR). Information from the FDAU to the FDR is sent via specific data frames, which depend on the aircraft manufacturer. Integration of FDAU functions into software required by other aircraft system components is now being seen, as in the case of the Enhanced Airborne Flight Recorder (EAFR) installed on the Boeing 787.

Literature Review

Ayeni and Yisah (2015) carried out research on developing a cloud-based model for flight data records. The results of their works show cloud-based services can be used as an alternative to the flight data recorder which may be lost in a crash as a result of extreme heat conditions. Some vital issues affecting cloud integration with the flight data recorder were also identified such as the unreliable internet connection and unavailability of large bandwidth to support real-time flight data transmission. The cloud computing concept could be extended to the aircraft system data network environment with every aircraft subscribing to the cloud resources to run their mission-critical or non-mission critical applications.

In this work, cloud-based aircraft Flight Data Recorder was modeled, simulated using MATLAB and implemented using Google private cloud computing. The result shows that FDR data stored on the cloud can serve as a backup or alternative data when FDR on board the aircraft could not be retrieved; retrieved but damaged or burnt beyond allowable temperature. With their model, aircraft manufacturers or airlines may not necessarily wait for disaster to occur or disaster recovery plans, because, for every flight, it is possible to retrieve flight data performance of the aircraft from cloud database independent of an on-board FDR equipment, this will enable them to take preventive or proactive action to resolve problems in order to avert disaster.

Research Methodology

Proposed Model

The proposed model is basically set to achieve its aim and objectives in the following steps:

- Designing an active database with events, conditions, and actions to be triggered by the control tower in case of an abnormal flight condition detected.
- Initiate a cloud-based service to record FDR from the aircraft

- Integrate the above-mentioned systems for flight monitoring and assistance.

The proposed model is hereby presented in figure 1 below which is an improvement upon Ayeni and Yisah (2015). The model was adapted from their work while the improvemen+t is on the inclusion of the active database section and its synchronization with the aircraft to achieve flight monitoring and assistance.

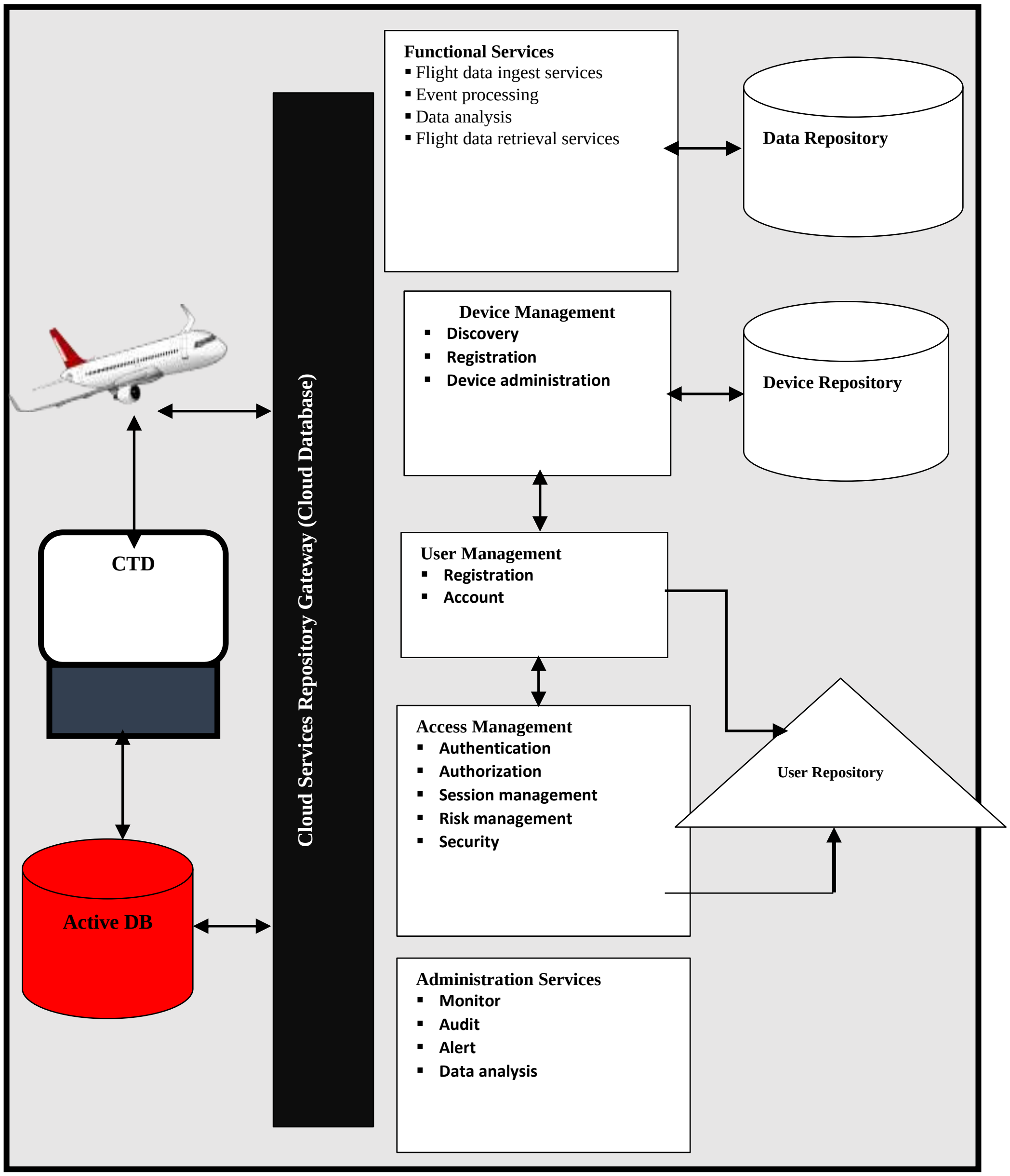

Figure 1: Proposed model

Data flow diagram for the proposed model

The proposed model aforementioned in this work can be seen in a summary form from the data flow diagram below. The diagram shows the real-time working principle and flow of data from one entity in the model to another. The data repository in the cloud collects the data generated by the aircraft which is stored on the FDR on board the aircraft while it is still air-bound and places the data in a ready form for analysis and further manipulation and control. The measures described above have been provided as an aid to data manipulation, storage, control, and general security. The active database which is embedded in the control tower system device interacts with the cloud repository to detect and indicate any abnormality in the flight. The control tower having been alerted then provides real-time assistance to the aircraft. Here we understand how the flow of data happens in the proposed system.

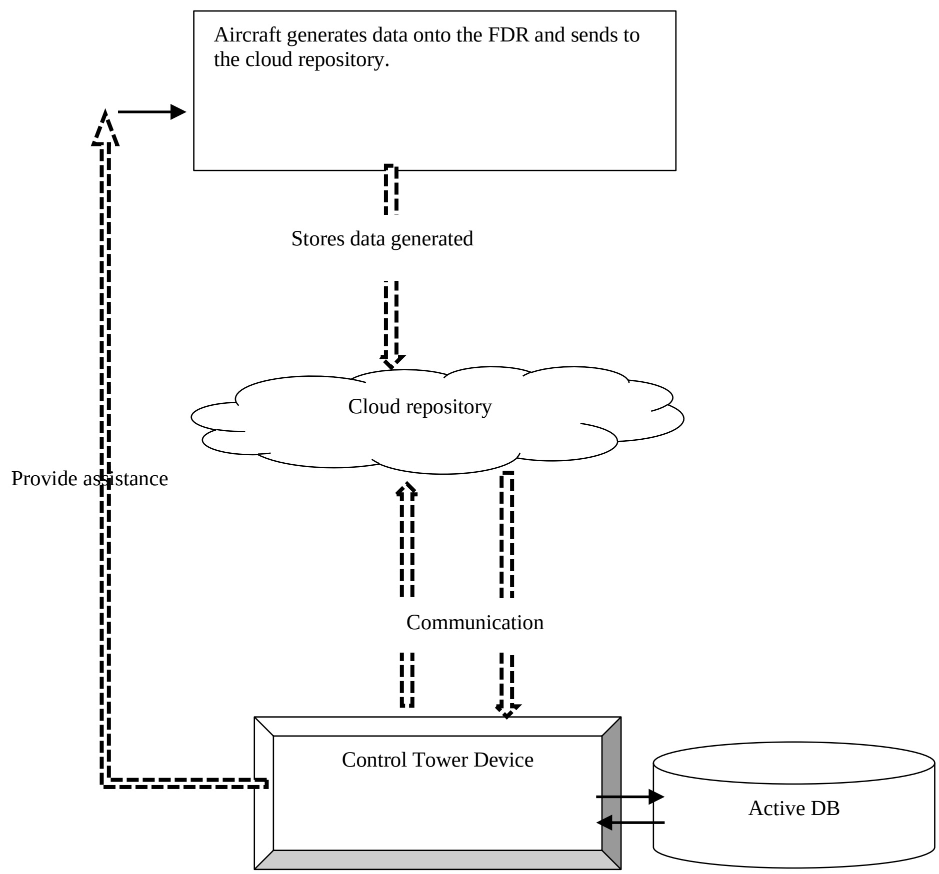

Figure 2: Working Principle model

System Analysis and Design

The approach used in the design of the required system is presented here. We make integration between the use of databases and its tools while linking with the front end application to provide the necessary assistance specified in this work. We first discuss the functional requirements and specifications of the system and then delve into the conceptual design methodology.

Functional Requirements/ Specifications

The functional requirements of a system specify explicitly what a system is supposed to achieve in its design and implementation. It summarizes or rather details on the objectives of the entire work and what the researcher seeks to achieve.

Building an Enhanced Flight Data Record system that provides flight assistance to flights that are even still air bound. It extends the power of transmitting flight data record to cloud to repositories for better access to flight data in case of an accident investigation and after flight analysis. The extension it seeks is to provide flight assistance where necessary in the case of identifying an abnormal flight condition from any of the aircrafts currently air bound.

Below are the specifications the proposed system seeks to achieve:

- Create an interface to allow administrators to log in to the front end of the application
- Provide a mechanism for authenticating user log in
- Redirect to the appropriate page in response to a login attempt depending on the success of the login process
- Provide an admin interface to select what operation to carry out
- An admin can choose to view air bound flight status
- An admin can choose to view a specific flight status
- An admin can choose to monitor a specific flight
- The interfaces communicate with the database and respond with appropriate results to queries

Application Blueprint

Here we show the internal structure and communication flow of the entire application in a diagrammatic format for a proper understanding of the specifications aforementioned in the work. The blueprint maps the implementation course of the application that implements the methodology of the approach to meet the problem stated earlier.

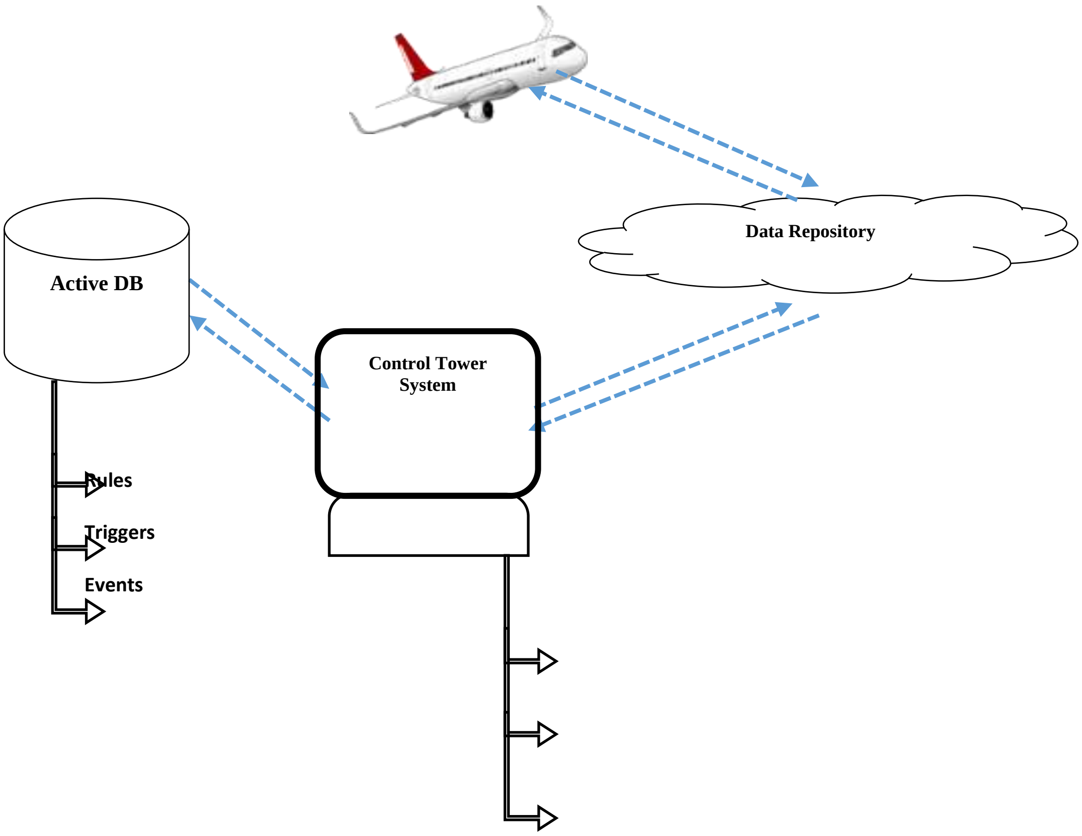

Figure 3: Application Blueprint

Active Databases

There are traditional database implementations which are primarily for the purpose of data storage which may be needed for retrieval purposes which could either be updating the record in the database or other manipulation actions that are allowed on the records in the database. There are, however, other forms of database implementations which are more robust in their approach in that they tend to make decisions with or without the supervision of an administrator. One such implementation is known as an Active Database.

The approach of an Active Database implements certain rules by creating events and specifies some conditions and associated actions if triggered. The model of an Active Database's implementation compared with a traditional database implementation is presented below.

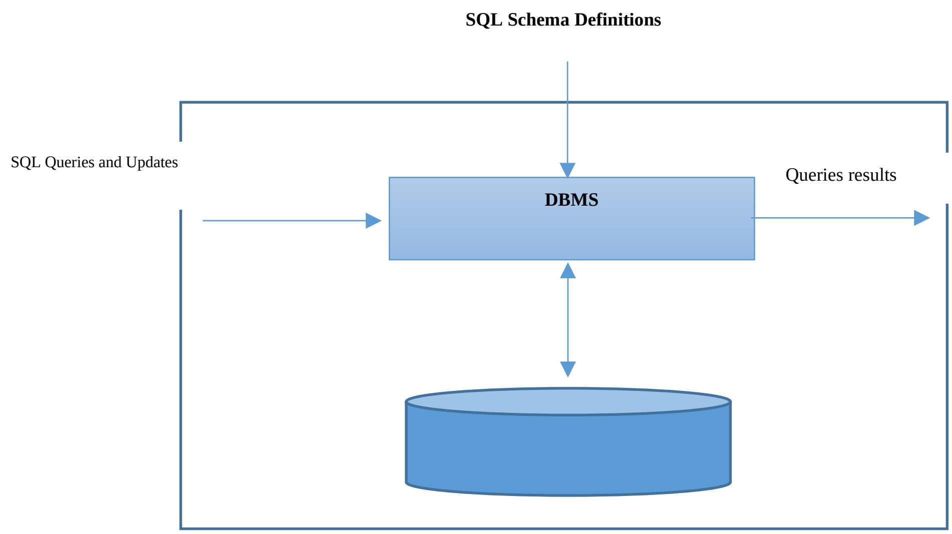

Figure 4: General Principles of Database Systems

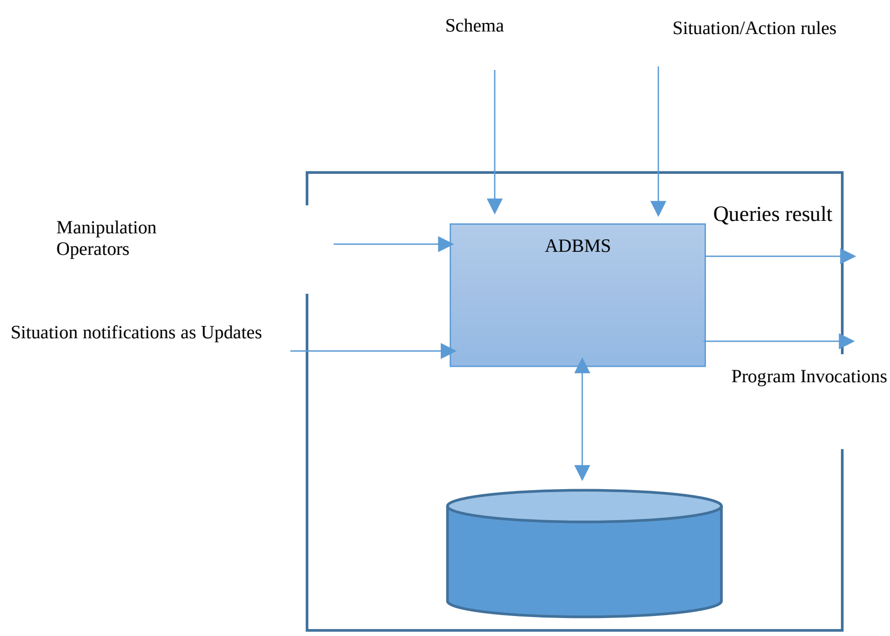

Figure 5: General Principles of an Active Database System

Results

Profiling Results of the proposed system

CPU Usage of the proposed system

Figure 6 below shows CPU utilization in terms of time of the proposed system.

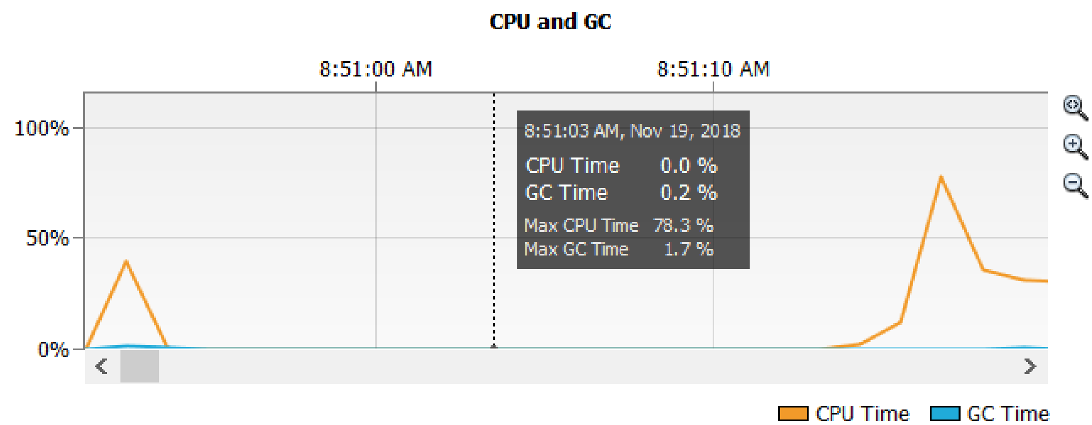

Figure 6: CPU Usage

Memory Usage of the proposed system

Figure 7 below shows the memory usage by the proposed system.

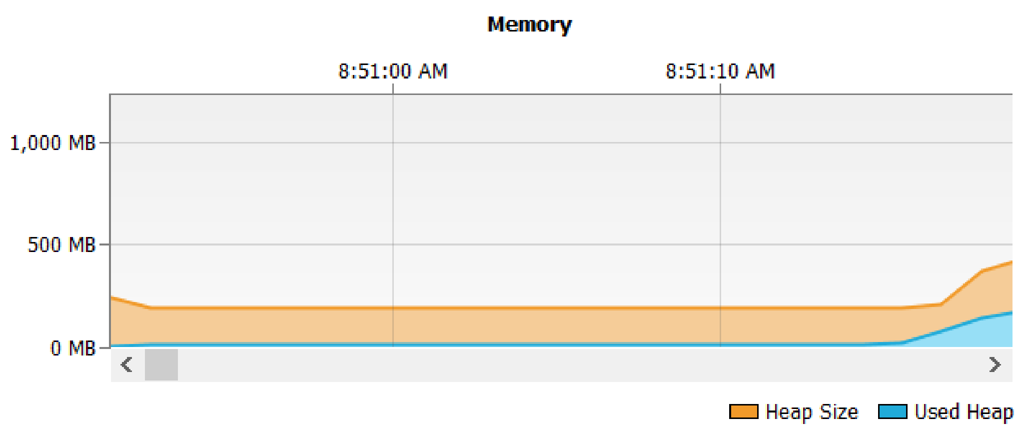

Figure 7: Memory Usage of the proposed system

Performance Evaluation

In this section, the existing and the proposed system were evaluated based on the CPU time, Memory usage and bandwidth utilization.

Performance Evaluation of proposed system

Table 1 below shows the performance evaluation of the proposed system in terms of CPU time, Memory Usage and Bandwidth.

Parameter	Test cases	CPU Time (%)	Memory Usage (Mb)	Bandwidth utilization (Mbps)
Speed (kts)	256.42	30.5	150	3.55
	266.42	35.1	149.8	3.55
	276.42	40	149.5	3.55
	286.42	50	148 7	3.55
	296.42	60.8	148.3	3.55
Altitude (ft)	3000	32.5	160	3.55
	3010	33.6	155.2	3.55
	3020	38.6	152.4	3.55
	3030	40	150	3.55
	3040	46	149.6	3.55

Table 1: Performance evaluation of the proposed system

Performance Evaluation of existing system

Table 2 below shows the performance evaluation of the existing system.

Parameter	Test cases	CPU Time (%)	Memory Usage (Mb)	Bandwidth utilization (Mbps)
Speed (kts)	256.42	36.1	200	1.733
	266.42	41.8	199	1.733
	276.42	45.2	197	1.733
	286.42	56	188	1.733
	296.42	64.7	183	1.733
Altitude (ft)	3000	38.5	210	1.733
	3010	40.2	200	1.733
	3020	42.4	198	1.733
	3030	45.1	187	1.733
	3040	52.4	186	1.733

Table 2: Performance evaluation of the existing system

Comparison of Experimental Result obtained from the existing and proposed system

Table 3 below shows the comparative analysis of the result obtained from averages of resources consumed by both the existing system and proposed system.

System	Parameter	Readings	CPU Time (%)	Memory Usage (Mb)	Bandwidth utilization (Mbps)
Existing system	Speed	276.42	48.76	193.4	1.733
	Altitude	3020	56.66	196.2	1.733
Proposed System	Speed	276.42	43.19	149.3	3.55
	Altitude	3020	38.14	153.4	3.55

Table 3: Comparison of Experimental Result obtained from the existing and proposed system

Analysis of the experimental result

This section shows the graphical representation of the evaluation result obtained from both the existing system and the prosed system.

System processing time used for airspeed by both the existing and proposed system

Figure 8 below shows the graphical representation of the CPU time used for the Airspeed of both the existing and proposed system obtained from table 1 and 2.

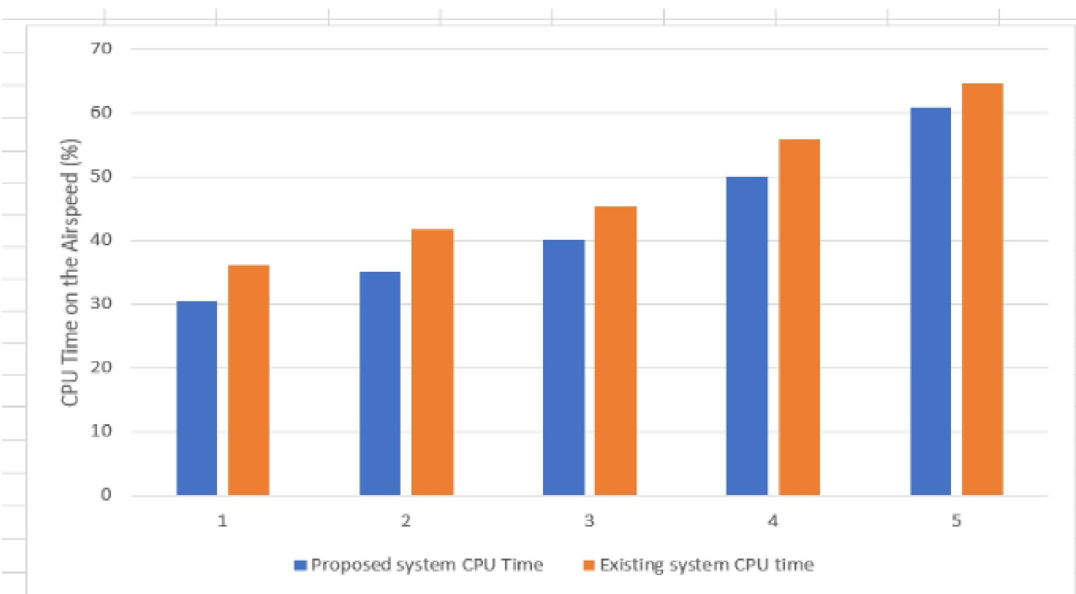

Figure 8: System processing time used for airspeed by both the existing and proposed system

System processing time used for Altitude by both the existing and proposed system

Figure 9 below shows the graphical representation of the CPU time used for the Altitude of both the existing and proposed system obtained from Table 2 and Table 3

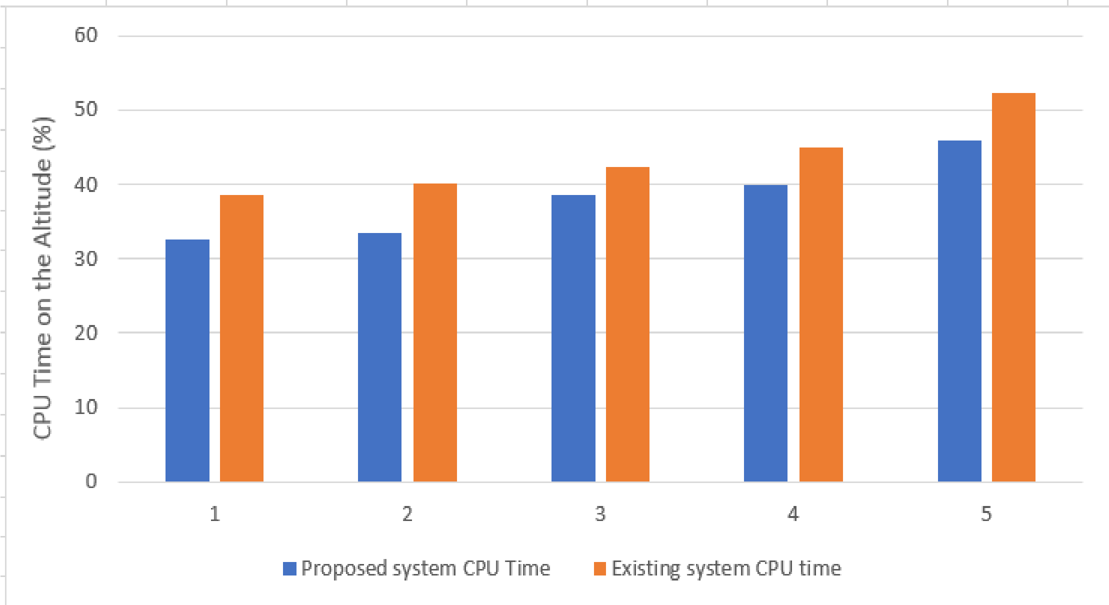

Figure 9: System processing time used for Altitude by both the existing and proposed system

Memory Usage for Airspeed by both the existing and proposed system

The figure below shows the graphical representation of the Memory Usage for the Altitude of both the existing and proposed system obtained from Table 1 and Table 2.

Figure 10: System processing time used for Altitude by both the existing and proposed system

Memory Usage for Altitude by both the existing and proposed system

Figure 11 below shows the graphical representation of the Memory usage for the Altitude of both the existing and proposed system obtained from table 1 and 2.

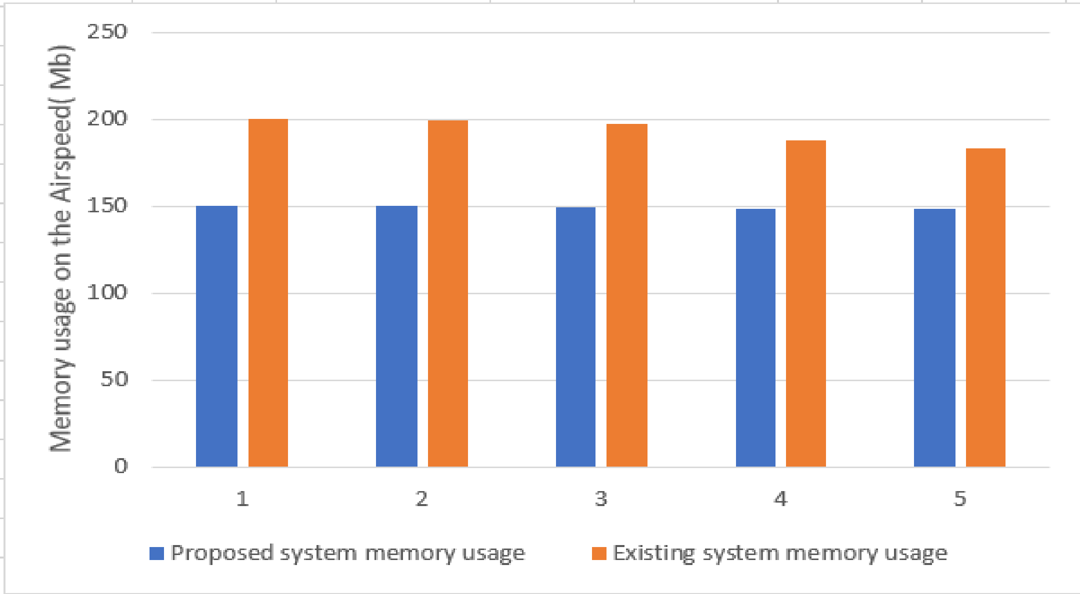

Figure 11: Memory Usage for Altitude by both the existing and proposed system

Conclusion

Enhancing the flight data record has been the main aim of this work. This research work has focused on detecting an abnormal flight condition even without the Pilot checking in on that in order to provide assistance to air-bound flights.

So far, this research work has shown that flight data record collected to the cloud repository can be used for more than just accident investigation but can be employed to provide real-time assistance. The implementation has been able to detect abnormal flight conditions and flag them with appropriate colors where necessary for further investigation.

With the advancement in technology of aircrafts today, flight data recorded to cloud repositories can be enhanced to perform more than what it is usually used for (crash investigations). This can be of great assistance to Pilots who may have lost communication with the control tower when experiencing some abnormal conditions.

Limitations of the Study

This research work has been implemented based on some assumptions and thus, some limitations. The limitations of this work include:

- Flight data collected are just limited to altitude, speed, latitude, and longitude.
- Flight data were simulated and not real due to the difficulty to access live flight data.
- Database and the entire application were hosted locally not online.

Areas for Further Research

As a recommendation for further research, an extension can be made on this work by improving on its available features in areas such as:

- Connecting with local authorities of a known flight location.
- Being able to place a direct call to authorities notifying them of the presence of an aircraft with some abnormal conditions within their jurisdiction.

REFERENCES

Li, J., Carlson, E. B., & Lacis, A. A. (2014). Application of spectral analysis techniques in the inter-comparison of aerosol data, Part 4: Synthesized analysis of multisensor satellite and ground-based AOD measurements using combined maximum covariance analysis. Atmos. Meas. Tech., 2531-2549.

Endre, B., & Winterhalter, M. (2012). The United States Patent No. 8121752.

Ayeni, B. K., & Yisah, S. O. (2015). Cloud-Based Architecture Solution for Aircraft Flight. British Journal of Applied Science & Technology.

Zachman, J. A. (1999). Framework for Information Systems Architecture. IBM Systems Journal, 38(2/3):454–470.

Zachman, J. A. (1987). The framework for information systems architecture. IBM Systems Journal, 26(3):276–292.

Putman, J. (2001). Architecting with RM-ODP. Prentice Hall, Upper Saddle River.

TOGAF. (2011). TOGAF. The Open Group: Version 9 Enterprise Edition. Retrieved from TOGAF: Available:https://www2.opengroup.org/ogsys/catalog/G116 Version 9.1 01

C4ISR. (1997). Architecture Working Group: C4ISR Architecture Framework Version 2; Retrieved from C4ISR: Available:http://www.afcea.org/education/courses/archfwk2.pdf

DODAF. (2010). DOD Architecture Framework Version 2.02. Retrieved from DODAF: http://dodcio.defense.gov/Portals/0/Documents/DODAF/DoDAF_v2-02_web.pdf