

**© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.**

# Adaptive API for Real-Time Streaming Analytics as a Service

Catherine Inibhunu, Member IEEE, Roozbeh Jalali, Ian Doyle, Aaron Gates, John Madill, Carolyn McGregor, AM, Senior Member IEEE

**Abstract**— A significant amount of physiological data is generated from bedside monitors and sensors in neonatal intensive units (NICU) every second, however facilitating the ingestion of such data into multiple analytical processes in a real time streaming architecture remains a central challenge for systems that seek effective scaling of real-time data streams. In this paper we demonstrate an adaptive streaming application program interface (API) that provides real time streams of data for consumption by multiple analytics services enabling real-time exploration and knowledge discovery from live data streams. We have designed, developed and evaluated an adaptive API with multiple ingestion of data streamed out of bedside monitors that is passed to a middleware for standardization and structuring and finally distributed as a service for multiple analytical services to consume and perform further processing. This approach allows, (a) multiple applications to process the same data streams using multiple algorithms, (b) easy scalability to manage diverse data streams, (c) processing of analytics for each patient monitored at the NICU, (d) ability to integrate analytics that seek to evaluate multiple patients at the same point in time, and (e) a robust automated process with no manual interruptions that effectively adapts to changing data volumes when bedside monitors increases or the amount of data emitted by a monitor changes. The proposed architecture has been instantiated within the Artemis Platform which provides a framework for real-time high speed physiological data collection from multiple and diverse bed side monitors and sensors in NICUs from multiple hospitals. Results indicate this is a robust approach that can scale effectively as data volumes increase or data sources change.

## I. INTRODUCTION

Real-time analytics that seek to utilize live data streams captured from bedside monitors and sensors from Neonatal Intensive Care Units (NICUs) requires a robust framework that can ingest multiple data streams, transform the data and then distribute it in a synchronized manner for more complex analytical services to process, every second. Without such a framework, processing of every new data stream would be impossible to manage especially in architectures that need to accommodate changing data volumes, diverse data types and multiple algorithms for real time knowledge discovery and

analysis of physiological data streams. Generation of real time analytics is a phenomenon that has been explored by many organizations that seek data insights within minutes to hours of data capture [1]. However, most often the time aspects of real time analysis are actually performed using static data that is already generated, collected, stored and then provided for analytics processes to ingest. Within such a paradigm, most analytical processes follow the data pipeline as shown in Fig. 1.



Figure 1. Static Analytical Process Flow

The sequential pipeline in Fig. 1 has been demonstrated as successful for many predictive analytics and data mining case studies in understanding customer buying habits in market basket analysis where static data is the core [2]. There are many shortcomings in adopting the static process flow in Fig. 1 when faced with real time data streams as this process does not allow actual real-time analytics, but rather analytics is only possible after data is captured and stored. Within that paradigm, a significant gap exists from event time to analysis. As a result, important insights are discovered too late [3]. In critical care, especially in domains like NICUs, delayed dissemination of insights would affect their ability to be used for clinical decision making.

Recent research described in Wang et al [4] highlights the capabilities in systems developed to support big data analytics from the potential of the data once collected, however, not much has been addressed on the ability for data integration across many data services at the same time scaling to exponential data volume growth. Most systems built to support big data analytics lack interconnectivity given they are normally built as standalone applications which are not able to effectively interact with other systems. This approach leaves many data replication across many systems resulting in a huge strain on the computing and network infrastructure especially when trying to utilize all the data available for continuous analytics. As a result, data ingestion remains segregated in many clinical decision support systems.

C. Inibhunu, University of Ontario Institute of Technology, Oshawa, Ontario, Canada. (e-mail: cinibhunu@acm.org).

C. McGregor, University of Ontario Institute of Technology, Oshawa, Ontario, Canada, University of Technology Sydney, Ultimo, NSW, Australia (e-mail: c.mcgregor@ieee.org).

Cloud computing services can provide effective data replication and seamless data management as they can be designed with the ability to provide the same ground truth to multiple analytics services. This would enable the same data to be utilized by multiple analytics and machine learning algorithms at the same time point in time. Even though cloud computing has seen many excellent features for parallel data disseminate that propagate vast services, there is still more work required to support such a process especially in real-time physiological data streams.

Artemis is a platform that recognized the need for a more effective data acquisition process as described in [5]. The system was effective at collecting real-time data streams from NICU, however, physiological analysis was only possible for each patient at a time making the process cumbersome when analyzing multiple patients data. Such an approach is prone to human error, missing analysis during transitions, not possible to process this in real-time for all the patients in parallel and not able to propagate the same data for multiple analytic services especially when trying to understand patterns in multiple patients' data.

In recognition that a real-time streaming analytics service must have very limited or no manual intervention while processing all data flowing from multiple data sources every second, we have proposed an adaptive real-time streaming API. This is an innovative method that allows multiple processes to ingest the same data streamed out of multiple bedside monitors in parallel. The API adapts to multiple incoming data streams from a middleware that first standardizes and structures the data then provide the processed data for ingestion by other services. This approach allows; several applications to process the same data using multiple algorithms, making it easier to manage data flows, process analytics for each patient at each bedside, allow potential integration of analytics to evaluate multiple patients and provides a robust automated process with no manual intervention.

The rest of the paper is organized as follows: section II details the methods, section III the application to real-time physiological data streams, section IV details the results, section V the discussion and we conclude in section VI.

## II. METHODS

In this paper we present and evaluate an adaptive real time streaming as a service API for support of multiple machine learning services and analysis of real-time physiological data streams generated from medical devices. The architecture scales to multiple data sources and data consumers. A demonstration of this approach has been instantiated within the context of neonatal intensive care. The API has been developed as an application within the Artemis Cloud platform as described in [6] using data streaming from two hospitals, McMaster Children Hospital and Southlake Regional Hospital. The application receives multiple physiological data streams, processes the data and then provides standardized data as a service for other processes to consume. This approach allows an autonomous API for each

hospital system enabling clear segregation of data per hospital following a robust security mechanism as detailed in [5].

### A. Real Time Streaming Analytics API

Artemis Cloud instantiates a framework for real time data acquisition from monitors at the bedside [6]. We have enhanced this framework to include a robust component for multiple data ingestion augmented with an API that processes data tuples within data streams when captured live from bedside monitors. This API is embedded in a real time streaming cloud architecture where multiple data streams are processed and then made available for consumption by various other services such as storage systems, machine learning algorithms and other analytical services. The proposed approach as shown in Fig. 2 (in purple) adds a new component on the cloud computing architecture within Artemis Cloud where raw data streams are processed and then made available for other services.

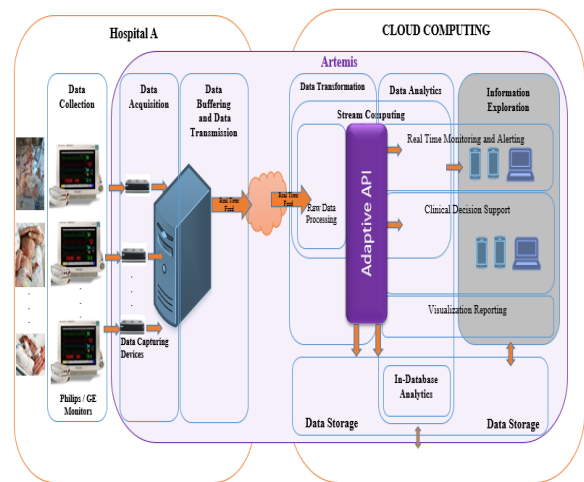


Figure 2. Adaptive API for Real-Time Streaming Analytics

The new component acts as an internal server that generates processed data for clients to consume. In our application the clients are the multiple algorithms that need to ingest data for multiple analytics purposes in parallel. Within Artemis, we have developed an application that is well defined such that a client can identify the data and format it needs to consume and therefore uses the appropriate rules to ingest the data. This is a robust approach that allows smooth flow of data from incoming streams (producers) to analytical services (consumers) in addition to allowing multiple services to integrate data flow effectively

To facilitate effective functioning of the proposed API, instantiation of some of the existing Artemis components is augmented by new ones to support the adaptive API as follows;

Data: As sensors continuously capture data about patients physiological features live at the bedside and then display this information on monitors so is the flow of this data to Artemis Cloud. For example if a single hospital can have as many as 60 patients in the NICU this would mean there are 60 monitors streaming live data from the bedside. Adopting Artemis cloud, we have put together an architecture that captures all the physiological features captured by the monitors such as heart rate, respiratory rate, blood pressure, blood oxygen saturation and ECG waves. This data is fully de-identified with no way of re-identifying a patient once the data is sent to the cloud system following the principles in [5]. This component is able to handle multiple data sets captured through the GE Dash 4000/5000 series or the Philips MP50 or MP70 monitors.

Collection: Each hour, a patient can have millions of records, as such the collection of such data from multiple patients and in multiple hospitals requires a streaming architecture that is stable and able to scale on demand. We realize the possibility of having higher volume of data in one hospital compared to the other based on the monitoring device utilized at a specific hospital and the number of patients in the NICU. As such, we have implemented a data collection component that scales the data collection process based on data source loads.

Processing: This component includes modules that ingest all the data from the collection component. Data is received as JSON messages and mechanisms for structuring and standardizing it are fully incorporated thereby effectively transforming the incoming data streams then making it ready for the API.

Adaptive API: This component acts as an agent for ingesting data and has modules that process all incoming data streams and publish it for other processes to consume. As there are multiple data points flowing from the monitors, this component is designed to take in the multiple data streams in parallel. This approach allows publishing of each data stream which then flow seamless to other processes as shown in Fig. 3, where multiple data producers send in data to the API which then makes processed data ready for consumption by multiple consumers. Figure 4 demonstrates the composition of the published datasets as topics after processing within the API.

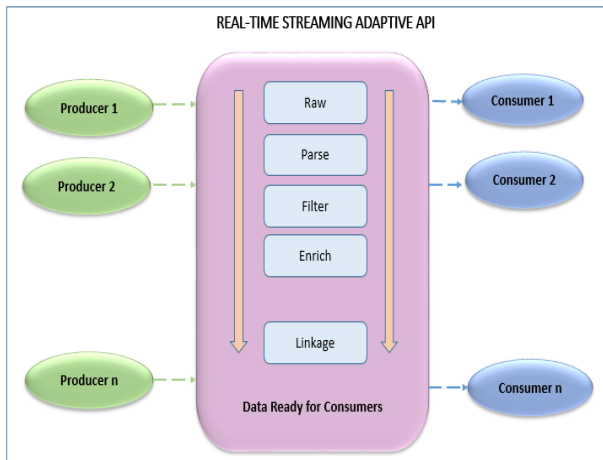


Figure 3. Real-Time Streaming Adaptive API

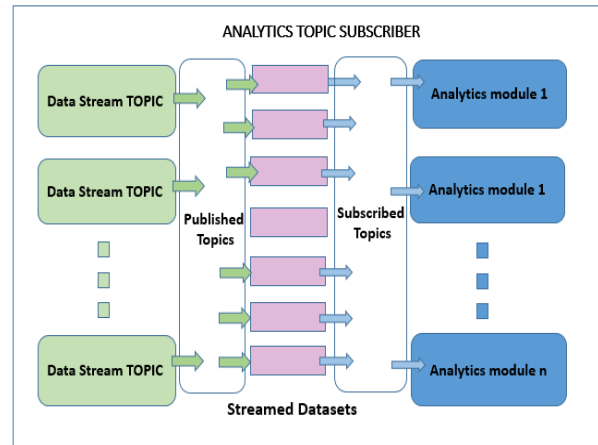


Figure 4. Analytics Subscriber Modules

Analyze: The Analytics component contains multiple modules which enables multiple complex algorithms to be processed in parallel while acting as consumers of the data streams published by the real-time API. The algorithms can communicate with each other as they consume whatever data types they need to accomplish their assigned tasks. This approach allows the same data to be processed by multiple algorithms as data streams are provided as topics which any analytical process can subscribe to as shown in Fig. 4. Such an approach ensures consistency in all the data processed across many analytics algorithms.

Storage: This component allows raw data generated from the API to be stored in standard data storage systems such as DB2, MongoDB or any other storage type. The ability to store both raw and enriched data from the analytics components enable other data management processes to be performed in parallel. This approach enhances other data management processes that may include data quality and evaluation of the real-time analytics. At the same time, data streamed live to storage spaces is made available for other systems that require historical data but not yet integrated for real time consumption of streaming data feeds.

Presentation: This component accommodates the presentation of the analysis generated from the Analysis modules from the vast amount of algorithms and standalone analysis.

### III. APPLICATION TO REAL-TIME PHYSIOLOGICAL DATA STREAMS

Following the framework described in [5], a health care cloud service is implemented to facilitate real-time streaming of physiological data from 17 GE Dash Monitors at Southlake Regional Health Centre and 51 Philips Intellivue MP 70 monitors at McMaster Children’s Hospital.

Using the IBM InfoSphere Streams computing platform, for the Online Analytics component of Artemis Cloud, incoming raw data streams are ingested, parsed, standardized and then provided as input to other services.

IBM InfoSphere Streams is utilized as it allows seamless flow of data as tuples and enables new streams to be received, processed and propagated downstream for consumption by multiple operators or exported for use by other services. The implemented design is represented in Fig. 5 within the Artemis Cloud computing paradigm.

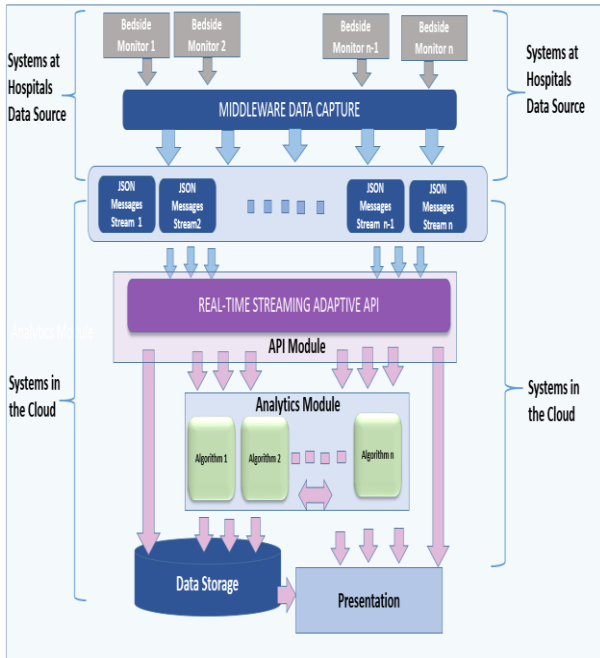


Figure 5. Implemented Real-Time Adaptive API

#### A. API Module

In each hour there are up to 68 patients across the two NICUs and data acquired from these patients is transmitted to Artemis Cloud in real-time. As shown in Fig. 5, multiple physiological data streams which are structured as JSON messages are ingested, processed and then published for consumption by other systems.

The API is able to handle multiple physiological data streams and adapts appropriately based on volume of data streams. As shown in Fig. 6 the API handles different data formats from metrics such as heart rate, SP02, respiratory rate to waveforms data such as electrocardiogram (ECG), which has orders of magnitude more data per second.

The API is able to adapt to ingest data from Philips MP50/70 and the GE Dash monitors and standardize this to a unified data element regardless of hospital source. This makes it very easy for downstream consumers not to worry about the data source but receive physiological data content for further processing in a consistent way.

#### B. Analytic Module

After the API has processed the incoming data tuples within the data streams, this is then provided for consumption by multiple algorithms. In the deployed architecture, we have algorithms that analyze data streams such as Heart Rate, Respiratory Rate and Blood Oxygen Saturation generating temporal abstractions. The analytic module allows the ability

for these algorithms to consume the same data for each patient. This process facilitates unique evaluation of each patient's analytics that are then published to other services such as storage, presentation or for further analysis.

## IV. RESULTS

The proposed architecture is demonstrated as a part of the Artemis Cloud project within the Health Ecosphere Innovation Project funded by FedDev Ontario and is deployed within the NICUs of two hospitals in Canada. It is deployed in the Level 3 NICU in McMaster Children Hospital where there are 48 bedspaces together with an addition 3 bedspaces in the delivery suite. The second deployment is in the Level 2b NICU in Southlake Regional Health Centre (SRHC) which is a 17 beds NICU. McMaster Children's Hospital (MCH) NICU has a Philips Intelivue MP70 monitor and Southlake Regional Health Center has GE Dash monitors. These bedside monitors generate real time physiological data that is streamed to the Artemis Cloud service. The API allows multiple data ingestion workflows then prepares the data for consumption by other services such as multiple data flows to the storage systems. We have implemented 9 analytics algorithms that subscribe to the adaptive API which then process the data streams and sends the resulting analytics for storage or presentation.

The results from the implemented system demonstrate the potential of our adaptive API that can facilitate real-time streaming analytics as a service where multiple analytical algorithms are run concurrently thereby subscribing in real-time to physiological data streams as they are captured from the bedside NICUs.

#### A. Computation of the API

In a single hour there could be up to 17 neonatal infants in the NICU at SRHC, as such there are 17 monitors that capture physiological data from those patients every second. In MCH, there can be up to 51 monitors capturing live data. This data is then processed and passed on to the cloud system as JSON messages.

Fig. 6 shows the pictorial flow of messages as they are processing and passed on to the API which then enables consumption by multiple analytical services.

During our study there are approximately 900 messages per second acquired from MCH. These messages are combination of metric data (Heart rate, SPO2, Respiratory rate and waves data i.e., 12 lead ECG that is comprised of the standard leads I, II and III, the augmented limb leads (aVR, aVL and aVF) and the 6 precordial leads (V1, V2, V3, V4, and V6) which provide the electrical activity of the heart over a period of time [7]. The API is able to process this data for each bedside monitor at the rate it is generated.



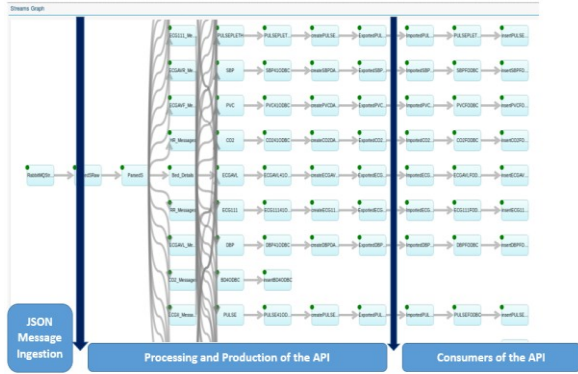


Figure 6. Data Streams Ingestion as JSON Messages

Fig. 7 shows the multiple algorithms that consume the streamed data in real time as it is generated in the API

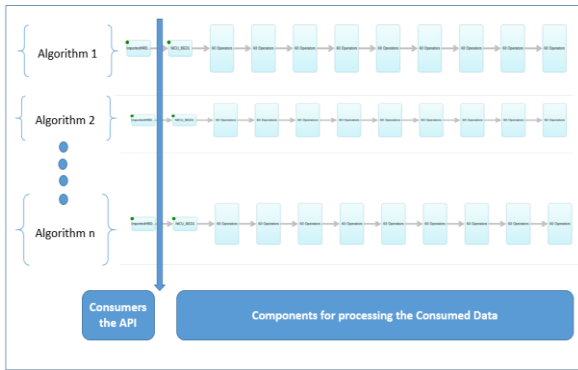


Figure 7. Multiple Algorithms Consuming Live Data Streams

### B. Data Volume Processed in the API

The volume of data processed grows as the number of patients increases in the NICUs. Fig. 8 shows the volume of data processed in 1 hour from 1 to 55 patients. In 1 hour approximately 180 million data points are processed for 55 patients, as each patient has about 3 million records within the hour. Fig. 9 and 10 shows the volume of data processed within 24 hrs for 55 patients.

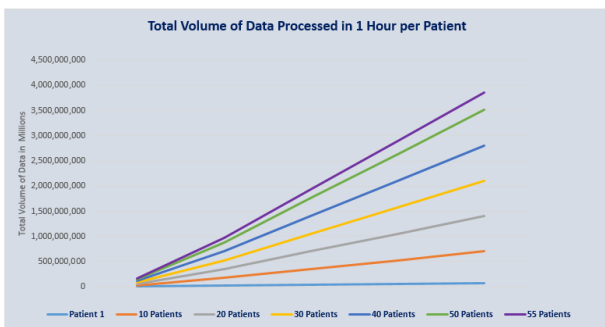


Figure 8. Volume of Data Streams Processed each Hour

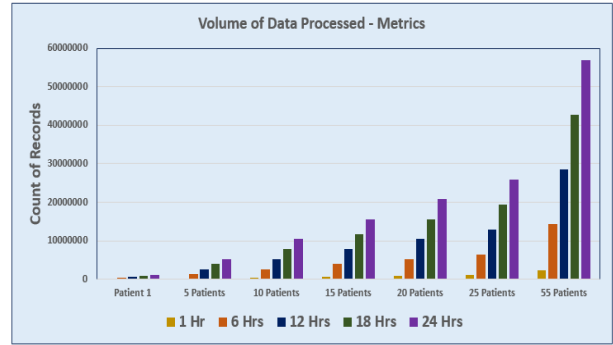


Figure 9. Metrics Data Processed for 55 Patients 1 to 24 Hrs.

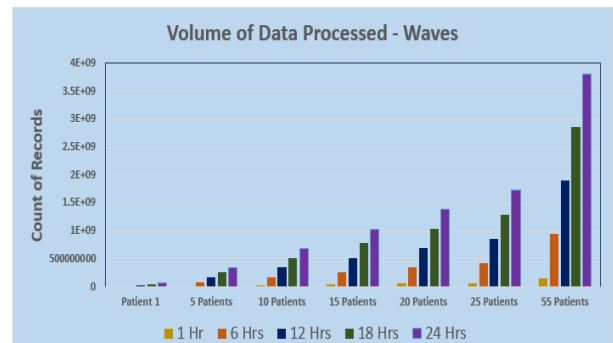


Figure 10. Waves Data Processed for 55 Patients in 1 to 24 Hrs.

Volume of data processed from 1 to 24 hrs. is shown in Fig. 10 and 11. For 55 patients the volume of metrics processed grows from 2.37 million to 57 million in 24 hrs. (Fig. 9). While the waves grow from 158.6 million records in 1 hour to over 3.8 billion records in 24 hrs. (Fig. 10). The overall volume of metric and waves data processed within the API for 55 patients is shown in Fig. 11 in a period of 1 to 24 hours.

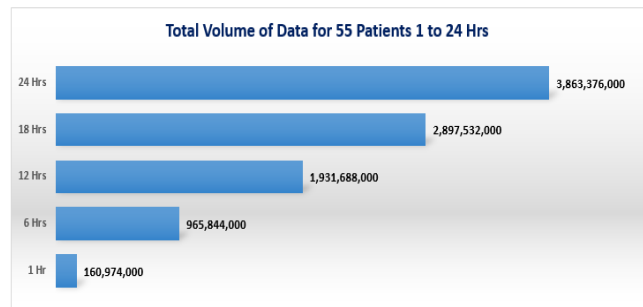


Figure 11. Volume Processed for 55 Patients 1 to 24 Hrs.

### C. Analytical Algorithms Processed in the API

There are 9 analytical algorithms that have been deployed as consumers of the implemented API. The algorithms process 3 physiological features for detection of Late Onset of Neonatal Sepsis (LONS) which generates temporal abstractions (TA)

that measures the heart rate and respiratory rate variabilities as detailed in [8], and TA of high fidelity blood oxygenation saturation (SpO<sub>2</sub>) levels and its correlation with Retinopathy of Prematurity (ROP) as detailed in [9]. Other consumers support our apnoea of prematurity, and anemia of prematurity analytics research. All these algorithms are able to process data for each patient without any manual intervention. The results generated are then provided for presentation or sent to the data storage for further processing.

## V. DISCUSSION

With the amount of data streams generated every second in NICU bedside monitors, a robust system is needed to ingest, process and distribute this data in multiple analytical services. We have developed and deployed such a framework that is able to process millions of records per hour. Unlike the framework detailed in [5], the developed system is able to scale to multiple data sources, effectively distributing processed data to multiple and diverse analytical services thereby allowing multi-patients analytics.

The developed API has been successfully implemented to handle data from two different hospitals in Ontario, Southlake Regional Hospital with 17 NICU beds and McMaster Childrens Hospital with 51 Patients in the NICU. Utilizing differing data captured from both hospitals, we have demonstrated the ability to process multiple algorithms to support concurrent research. As Artemis analytical capability advances, this framework provides a solid foundation for diverse real-time analytics services.

The scalability of the developed system is fully demonstrated by the ability to capture data from multiple Philips and GE Dash bedside monitors that vary on the type and size of data generated. As shown in Fig. 11, the API is able to successfully scale to exponential data volumes and within 24 hours we are able to process 3.8 billion records within the data streams. The API allows reduction of data duplication among multiple services as analytics systems consume only the data they require. This approach also allows the same ground truth of data to be available to multiple processes.

However, we have seen that as medical devices are developed by differing companies, so is the variation in the type of data generated by bedside monitors. As a result, more development work is needed in integrating data from multiple devices to be able to provide a unified adaptive API. Alternatively the new 11073 Medical Device Communication standards protocol provides potential for a more uniform data communication process. In future work we will demonstrate the integration of the 11073 standard within our API.

## VI. CONCLUSION

In the implementation of an adaptive real-time streaming API, we demonstrate the effectiveness of such a data pipeline for ingestion, processing and distribution of multiple analytics services. The system handles varying data sources, sizes and processing services and scaling for successful deployment in differing hospital systems. With such an architecture volumes of data captured is processed by multiple applications without

creating duplicates make it very effective for the same ground truth among multiple processes.

This architecture provides a foundation for accommodating advanced algorithms which continue to be developed and deployed as part of the enhancement to the Knowledge Discovery within Artemis. Subsequent results from analytical services that ingest data from the adaptive API will be provided in upcoming publications.

## ACKNOWLEDGEMENT

This research has been supported by: 1) the Health Ecosphere: Innovation Pipeline for Commercial Health Solutions Project (HEIPP) awarded by FedDev Ontario 2) the Canada Research Chairs program (#950-203427 and #950-225945); 3) the Canadian Foundation for Innovation (#203427); and McGregor's Research Excellence Chair in Health Informatics.

## REFERENCES

- [1] C. McGregor, "Big data in neonatal intensive care," *Computer*, vol. 46, pp. 54-59, 2013.
- [2] D. L. Olson, Y. Shi, and Y. Shi, *Introduction to business data mining* vol. 10: McGraw-Hill/Irwin Englewood Cliffs, 2007.
- [3] C. McGregor, "A cloud computing framework for real-time rural and remote service of critical care," in *2011 24th International Symposium on Computer-Based Medical Systems (CBMS)*, 2011, pp. 1-6.
- [4] Y. Wang, L. Kung, and T. A. Byrd, "Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations," *Technological Forecasting and Social Change*, vol. 126, pp. 3-13, 2018.
- [5] H. Khazaei, C. McGregor, J. M. Eklund, and K. El-Khatib, "Real-time and retrospective health-analytics-as-a-service: a novel framework," *JMIR medical informatics*, vol. 3, 2015.
- [6] C. McGregor, C. Catley, A. James, and J. Padbury, "Next generation neonatal health informatics with Artemis," in *MIE*, 2011, pp. 115-119.
- [7] M. B. Conover, *Understanding electrocardiography*: Elsevier Health Sciences, 2002.
- [8] C. McGregor, C. Catley, J. Padbury, and A. James, "Late onset neonatal sepsis detection in newborn infants via multiple physiological streams," *Journal of critical care*, vol. 28, pp. e11-e12, 2013.
- [9] K. S. Fernando, C. McGregor, and A. G. James, "CRISP-TDM 0 for standardized knowledge discovery from physiological data streams: Retinopathy of prematurity and blood oxygen saturation case study," in *2017 IEEE Life Sciences Conference (LSC)*, 2017, pp. 226-229.