Towards an Integrated Solution for IoT Data Management

Anderson Chaves
Supervised by Fabio Porto
LNCC, Brazil
achaves@lncc.br

ABSTRACT

The emergence of Big Data and the Internet of Things (IoT) is increasingly affecting all areas of modern society, being characterized by a huge number of data streams that demand real-time processing and analysis. The development of systems to assist on the management of these data streams plays an important role for IoT applications. However, there are numerous challenges that must be taken into account when building an efficient data system for handling large scale, dynamic, semi-structured data such as IoT, and currently existing solutions only partially address the requirements of these scenarios. In this PhD research, we summarize some of the main challenges involved in building an efficient system for IoT data management and analysis, and how different data management approaches such as Actor oriented, Array and Active Databases fit together offering strong contributions to these requirements. We also examine the potential of performing Machine Learning inference and handling Concept Drift in IoT as an integrated database process. Through this work, we lay the structure for the development of a Database Management System to support large scale data stream based analysis capable of combining these different strategies.

1 INTRODUCTION

From smart homes control systems to transportation, healthcare and industrial automation, the Internet of Things has been enabling great benefits both for individual and businesses, being used for better decision making, planning and higher productivity [1]. The main characteristics behind this IoT paradigm is the exploration of different technologies such as communication, embedded systems and data analytics in order to create smart devices for intelligent monitoring, locating, tracking and so forth [9, 18].

The efficient management of sensor data from IoT devices is essential to perform IoT data analysis. Through Complex Event Processing (CEP) methods, it is possible to detect anomalies and meaningful events from data streams and perform real-time decision making. However, processing and analyzing continuous data streams from heterogeneous networks still leads to a number of different challenges, and requires the development of new techniques and strategies.

A major challenge in an IoT environment is related to its large scale data flows. Data in IoT can have its sources in a very big range of endpoints that generate masses of data, and is frequently semi-structured or unstructured, conforming it to the Big Data paradigm [9]. Traditional DBMSs, which need to store and index data before processing it, cannot fulfill the requirements of timeliness and scalability of IoT data streams [10]. Besides, in order to perform analysis and visualization, existing solutions are often inefficient, because they incur in an incompatibility between the structure of the source data and the analysis tool [7]. Finally, there are a number of privacy and security issues as well as resource constraints such as memory, bandwidth and energy that must be taken into account when building an IoT data management system.

Another challenge in IoT is the necessity for on-line processing of data streams as opposed to off-line analysis. Machine learning (ML) is one of the leading strategies to perform reliable, efficient real-time analysis of IoT data in tasks such as predictions or anomalies detection [1]. However, the lack of integration between the ML application and the data system is often a restraint to performance improvements, since optimizations such as query planning or lazy evaluation are not possible when the two processes are treated as completely isolated tasks [8]. Additionally, when dealing with dynamic stream data such as IoT, the nature of the data distribution tends to change over time, resulting in the phenomenon known as concept drift. It occurs when the statistical properties of the target variable, which the model is trying to predict, change over time in unforeseen ways [15]. When that happens, the learned patterns of past data may not be relevant to the new data, leading to poor predictions and incorrect decisions. Machine Learning based analysis needs to be able not only to detect the drift, but also understand and react to it.

We argue that data management systems demand efficient mechanisms to deal with large-scale, heterogeneous IoT data. A recent work [25] has demonstrated that the programming model aimed specifically at concurrency and inherent parallelism of actor-oriented databases such as Orleans [5] and ReactDB [22] is an adequate solution for systems focused on IoT data management. Reactive behavior and CEP techniques are also essential for evaluating complex patterns over high-throughput data streams such as IoT [13, 21]. Since a large part of data made available by IoT devices is multidimensional spatio-temporal [9, 19], multidimensional array data models could provide great advantages to its management [4]. However, managing several different platforms instead of one makes the resulting solution unnecessarily complex and potentially inefficient. To the best of our knowledge, no existing solution has been yet proposed to combine all these approaches for IoT Scenarios.

Therefore, to address the challenges involved in the development of an adequate IoT solution, we envision a Database Management System capable of offering scalable support for IoT data management as well as analysis through Machine Learning. In this work, we present the following contributions:

Proceedings of the VLDB 2021 PhD Workshop, August 16th, 2021. Copenhagen, Denmark.

System Features		Actor Oriented Databases	Array Databases	Active Databases	Proposed Solution
Actor-Based Programming	Dynamic Scalability	+	-	-	+
	Asynchronous primitives				
	Encapsulation				
Array Based Data Management	Array-Based Operations	-	+	-	+
	Flexible Storage Format				
Complex Event Handling	Event Detection	-	-	+	+
	Reactive Behavior				
Machine Learning Support	ML as first class operations	-	-	-	+
	Concept Drift Handling				

Table 1: Potential contributions from different models for IoT data management

- We propose the development of a new Database Management System that offers CEP primitives through actor-based programming in order to perform rule-based monitoring for real-time scalable IoT scenarios.
- We propose to further extend our solution to include ML inference as first class operators for CEP, enabling further integration between the data system and the Machine Learning tasks.
- We propose to investigate the challenges involved in concept drift handling specifically in an IoT environment, and how to address these challenges in a data management system.

The remainder of this paper is organized as follows. In Section 2 we present the base concepts for the highlighted problems and proposed solutions. In Section 3 we present our idea of leveraging array databases to a scalable, reactive and intelligent solution fit for IoT. We conclude and present our research directions in Section 4.

2 RESEARCH CONTEXT

In this section, we introduce the base concepts of IoT data and challenges related to it. Afterward, we present the different database models that serve as foundation to the proposed solution. Finally, we describe the problem of Concept Drift in IoT context.

2.1 IoT Big Data Challenges

According to [9], big data in IoT has three features that conform to the big data paradigm: (a) a very big range of endpoints that generate masses of data; (b) semi-structured or unstructured data; (c) it is only useful after being analyzed.

Data generated by IoT has usually a high number of parallel sources, being subject to inaccuracies and noise during acquisition and transmission. It can be streamed continuously or accumulated as a source of big data. When dealing with big data analytics, its possible to produce insights after several days of its generation, but in the case of streaming data IoT analytics, they must be delivered in at most a few seconds or less. This real-time constraint incur in the following challenges for IoT big data:

Data Management: Data management is a big challenge to be addressed in order to realize the full potential of IoT, and therefore has become a key research topic [17, 20]. Many IoT systems are processor-intensive and require processing a massive amount of highly concurrently generated data. How to perform the management of these data interactions while ensuring low latency?

Visualization: Visualization is important in big data analytics, specially for IoT systems [18]. How can we perform visualization in the case of heterogeneous and diversely structured data generated in IoT?

Data Mining: The realization of the potential of IoT depends on being able to gain the insights hidden in the vast and ever increasing available data. Current data mining approaches don't scale well to IoT volumes. What characteristics are the most essential for a system fit to such environments?

Resource Constraints: In the IoT data stream model, a high volume of data is produced at high speed. Therefore algorithms that process it must do so under very strict constraints of space and time. Addressing these constraints requires that a significant amount of data processing must happen on edge devices. How can we design algorithms that work efficiently in such environments?

Security: Being able to deal with dynamic scaling while guaranteeing protection of data from different entities is another significant challenge. What is the most effective way to ensure access control and protection of data from large volumes of devices and, at the same time, ensure the development of a dynamic and flexible application?

2.2 Data management solutions

2.2.1 Array Database Models. Most IoT environments are constituted by static or moving sensor devices placed in specific locations that produce data continuously. Each data item has space coordinates as well as a time-stamp associated, incurring in a high time and space correlation. Because of this multidimensional spatio-temporal nature of IoT data, multidimensional array database models, built using arrays as the primary data representation, offer advantages for an efficient data management.

Array databases were initially proposed to better represent sensor, image, simulation, and statistics data of tipically spatio-temporal dimensions [4]. They have special query languages built upon array-based algebraic formalizations that model different kinds of operations such as aggregations or subsetting. Cells in an array have an intrinsic ordering, making it easy to quickly lookup values by taking advantage of this ordering. Array indexes do not need to be stored and can be inferred by the position of a cell, saving storage space. Arrays can also be split into subarrays (called tiles or chunks)

that can be used as processing and storage units to help answering queries efficiently.

Recently, some research effort is being applied in order to integrate ML tools and array DMBSs [24]. The system Rasdaman [3] allows the implementation of machine learning algorithms through User Defined Types and Functions that implement the underlying linear algebra operations directly over the arrays. In the case of SciDB [23], users are provided with linear algebra operators that can be used as building blocks to implement the ML algorithms. In SAVIME [11], users can perform inference from machine learning models as part of the query expression, allowing the jointly optimization of the data preparation process and its input to the model.

2.2.2 Active Databases and Complex Event Processing. An event can be defined as an occurrence of significance in a system [16]. Historically, many different initiatives have studied event processing for different reasons. Active Databases intended to extend traditional DBMSs by enabling the specification of reactive behavior. The idea was to develop strategies to respond automatically to events and changes in the database state through mechanisms formalized as ECA rules [26]: if an *event* is detected, and any of previously defined *conditions* become true, then a corresponding *action* is taken without any external intervention.

Complex Event Processing extend the logic behind ECA rules, being understood as a set of techniques combined in order to perform real-time stream processing for monitoring and detection of arbitrarily complex patterns in massive data streams [16]. They are commonly used in IoT environments to enable real-time or near real-time decisions [13]. In CEP, each data item is abstracted as an event produced by a data source. A CEP engine combines multiple simpler events to produce more complex ones, that match previously defined patterns. It typically must process multiple data streams from different sources in order to track simultaneously hundreds or even thousands of different patterns through evaluation mechanisms such as non-deterministic finite automaton or tree-based plans [12].

2.2.3 Actor Oriented Databases. The actor programming model is a well-known model for distributed and concurrent programming, in which the actor is the fundamental computing unit. Its main principle is that in a system, the control flow and the data flow must be inseparable. Actors do not share state and communicate via asynchronous messages. Because of its characteristics, actors are a scalable solution to support the management of any number of independent and heterogeneous streaming data sources.

In recent works, it has been demonstrated the effectiveness of the integration of data management features such as transactions and indexing into actor runtimes [6]. The authors of [25] demonstrate that this solution is in fact very suitable to perform IoT data management. A similar approach has sought to integrate actor primitives into relational databases [22] by extending the programmability of stored procedures with actor objects, taking advantage of databases state management features.

2.3 IoT Concept Drift

Concept drift can be formally defined as follows [15]: given a time period $[0, t]$, a set of samples, denoted as $S_{0,t} = \{d_0, ..., d_t\}$, where $d_i = (X_i, y_i)$ is one observation or data instance, X_i is the feature vector, y_i is the label, and $S_{0,t}$ follows a certain distribution $F_{0,t}(X, y)$. Concept drift occurs at timestamp $t + 1$, if $F_{0,t}(X, y) \neq F_{t+1,\infty}(X, y)$.

Research on learning under concept drift presents three components beyond traditional Training/Prediction: Drift detection, drift understanding and drift adaptation. The first refers to whether or not a concept drift occurs in a stream set of data. Drift understanding is related to when, how and where it occurs. Finally, drift adaptation refers to reacting to the existence of a drift.

Recently, some works have been proposed to deal with concept drift specifically in IoT platforms. For example, the work of [14] proposes an ensemble learning method based on offline classifiers to address concept drifts and imbalance data concurrently. In [2], its proposed an unsupervised model-independent methodology to detect drifts in data generated from IoT devices. In [27], it is proposed a concept drift adaptive method to anomaly detection in IoT services that considers the time influence to change the sample distribution. However, this is a not fully explored topic and many research opportunities still exist.

3 LEVERAGE ARRAY DATABASES TO IOT COMPLEX EVENT PROCESSING

Historically, Database Management Systems have offered many benefits to data intensive applications, such as transactions, indexing, query planning and declarative query languages. An IoT data management solution must answer specific demands, such as encapsulation for isolating state and access control, asynchronous primitives and dynamic scalability, since in many scenarios, sensing devices can instantly enter and leave a system. It should be able to detect and react to predefined data patterns automatically, while providing quick data access and an efficient integration to ML analysis. Table 1 highlights the strong contributions offered by active, actor-oriented and array databases to each of these IoT demands.

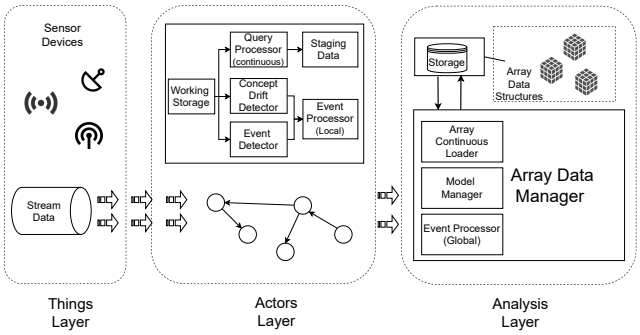

Figure 1: System Overview

By taking our inspiration in the approaches of Orleans [5], that added data-management functionality in a virtual actor runtime and ReactDB [22], which integrates actor features into a relational

database system, we investigate the potential of performing event detection and reactive behavior through actor-based primitives in an array database model. Figure 1 illustrates the proposed idea. At the things layer, data is collected from sensor devices and communicated to actor engines at the actor layer. Distributed actors manage these intermediate nodes that process and detect relevant (local) events based on attached sensors before sending them to the cloud based data center, along with relevant data in the form of array data structures. At the analysis layer, global queries and analysis that take into account alerts provided by actors can be made over the collected data. The intention is to provide a low latency environment, in which there is a reduced communication bottleneck.

The integration of ML-based analytics as part of the Data Management System may lead to powerful optimization opportunities since different parts of the ML process may be treated as operators of the query plan. To cope with the growing need for ML support in IoT data systems, we aim to provide both a local and a global event detector that supports ML inference from trained models as first class operators.

In IoT environments, communicated data from devices is usually collected and recorded by assuming a temporal relationship between records. As time goes on, concept drift is bound to occur, which may cause an accuracy drop to any methods that rely on long-term statistical data attributes. The proposed solution will count with a central drift detector that is able to determine if and when the drift occurred as well as the best reaction to it based on the local drift detectors.

4 CONCLUSION AND RESEARCH DIRECTION

In this paper, we discuss characteristics and challenges of IoT data management and summarize potential contributions from different strategies in addressing each of them. Our goal is to build an efficient, in-memory data management system that combines each of these different contributions into a single integrated solution, while offering a robust support for data analysis trough Machine Learning. As the next step in our study, we aim to focus on the design refinement and implementation of a prototype system as a foundation to our subsequent investigations. To evaluate the viability of our approach, we intend to submit it to a real use-case scenario that presents the IoT characteristics and challenges described. We also intend to perform comparative experiments with state-of-the-art big data frameworks in order to demonstrate the optimization opportunities that we envision.

5 ACKNOWLEDGEMENT

We would like to thank CAPES for its scholarships, and Petrobras for financing this work through the Gypscie project.

REFERENCES

[1] Furqan Alam, Rashid Mehmood, Iyad Katib, and Aiiad Albeshri. 2016. Analysis of eight data mining algorithms for smarter Internet of Things (IoT). *Procedia Computer Science* 98 (2016), 437–442.

[2] Mohsen Asghari, Daniel Sierra-Sosa, Michael Telahun, Anup Kumar, and Adel S Elmaghraby. 2020. Aggregate density-based concept drift identification for dynamic sensor data models. *Neural Computing and Applications* (2020), 1–13.

[3] Peter Baumann, Andreas Dehmel, Paula Furtado, Roland Ritsch, and Norbert Widmann. 1998. The multidimensional database system RasDaMan. In *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*. Association for Computing Machinery, Washington, USA, 575–577.

[4] Peter Baumann, Dimitar Misev, Vlad Merticariu, and Bang Pham Huu. 2021. Array databases: concepts, standards, implementations. *Journal of Big Data* 8, 1 (2021), 1–61.

[5] Phil Bernstein, Sergey Bykov, Alan Geller, Gabriel Kliot, and Jorgen Thelin. 2014. Orleans: Distributed virtual actors for programmability and scalability. *MSR-TR-2014–41* (2014).

[6] Philip A Bernstein, Mohammad Dashti, Tim Kiefer, and David Maier. 2017. Indexing in an Actor-Oriented Database.. In *CIDR*.

[7] Spyros Blanas, Kesheng Wu, Surendra Byna, Bin Dong, and Arie Shoshani. 2014. Parallel data analysis directly on scientific file formats. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. Association for Computing Machinery, Utah, USA, 385–396.

[8] Shaofeng Cai, Gang Chen, Beng Chin Ooi, and Jinyang Gao. 2019. Model slicing for supporting complex analytics with elastic inference cost and resource constraints. *Proceedings of the VLDB Endowment* 13, 2 (2019), 86–99.

[9] Min Chen, Shiwen Mao, Yin Zhang, Victor CM Leung, et al. 2014. *Big data: related technologies, challenges and future prospects*. Vol. 96. Springer.

[10] Gianpaolo Cugola and Alessandro Margara. 2012. Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)* 44, 3 (2012), 1–62.

[11] Anderson Chaves da Silva, Hermano Lourenço Souza Lustosa, Daniel Nascimento Ramos da Silva, Fábio André Machado Porto, and Patrick Valduriez. 2020. SAVIME: An Array DBMS for Simulation Analysis and ML Models Prediction. *Journal of Information and Data Management* 11, 3 (2020).

[12] Nikos Giatrakos, Elias Alevizos, Alexander Artikis, Antonios Deligiannakis, and Minos Garofalakis. 2020. Complex event recognition in the big data era: a survey. *The VLDB Journal* 29, 1 (2020), 313–352.

[13] Ilya Kolchinsky and Assaf Schuster. 2019. Real-time multi-pattern detection over event streams. In *Proceedings of the 2019 International Conference on Management of Data*. 589–606.

[14] Chun-Cheng Lin, Der-Jiunn Deng, Chin-Hung Kuo, and Linnan Chen. 2019. Concept drift detection and adaption in big imbalance industrial IoT data using an ensemble learning method of offline classifiers. *IEEE Access* 7 (2019), 56198–56207.

[15] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. 2018. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering* 31, 12 (2018), 2346–2363.

[16] David C. Luckham. 2001. *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems*. Addison-Wesley Longman Publishing Co., Inc., USA.

[17] Meng Ma, Ping Wang, and Chao-Hsien Chu. 2013. Data management for internet of things: Challenges, approaches and opportunities. In *2013 IEEE International conference on green computing and communications and IEEE Internet of Things and IEEE cyber, physical and social computing*. IEEE, 1144–1151.

[18] Mohsen Marjani, Fariza Nasaruddin, Abdullah Gani, Ahmad Karim, Ibrahim Abaker Targio Hashem, Aisha Siddiqa, and Ibrar Yaqoob. 2017. Big IoT data analytics: architecture, opportunities, and open research challenges. *IEEE Access* 5 (2017), 5247–5261.

[19] Mehdi Mohammadi, Ala Al-Fuqaha, Sameh Sorour, and Mohsen Guizani. 2018. Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials* 20, 4 (2018), 2923–2960.

[20] John Paparrizos, Chunwei Liu, Bruno Barbarioli, Johnny Hwang, Ikraduya Edian, Aaron J Elmore, Michael J Franklin, and Sanjay Krishnan. 2021. VergeDB: A Database for IoT Analytics on Edge Devices. In *CIDR*.

[21] José Roldán, Juan Boubeta-Puig, José Luis Martínez, and Guadalupe Ortiz. 2020. Integrating complex event processing and machine learning: An intelligent architecture for detecting IoT security attacks. *Expert Systems with Applications* 149 (2020), 113251.

[22] Vivek Shah and Marcos Antonio Vaz Salles. 2018. Reactors: A case for predictable, virtualized actor database systems. In *Proceedings of the 2018 International Conference on Management of Data*. 259–274.

[23] Michael Stonebraker, Paul Brown, Donghui Zhang, and Jacek Becla. 2013. SciDB: A database management system for applications with complex analytics. *Computing in Science & Engineering* 15, 3 (2013), 54–62.

[24] Sebastian Villarroya and Peter Baumann. 2020. On the Integration of Machine Learning and Array Databases. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*. IEEE, 1786–1789.

[25] Yiwen Wang, Julio Cesar Dos Reis, Kasper Myrtue Borggren, Marcos Antonio Vaz Salles, Claudia Bauzer Medeiros, and Yongluan Zhou. 2019. Modeling and Building IoT Data Platforms with Actor-Oriented Databases.. In *EDBT*. 512–523.

[26] Jennifer Widom and Stefano Ceri. 1996. *Active database systems: Triggers and rules for advanced database processing*. Morgan Kaufmann.

[27] Rongbin Xu, Yongliang Cheng, Zhiqiang Liu, Ying Xie, and Yun Yang. 2020. Improved Long Short-Term Memory based anomaly detection with concept drift adaptive method for supporting IoT services. *Future Generation Computer Systems* (2020).