Are the Challenges of IoT Frameworks being Resolved to Enable More Efficient Application Development?

Ryan C. Donovan
Data Analytics
NCI
Dublin, Ireland
Email: x16104269@student.ncirl.ie

***Abstract*—The IoT is becoming a very big part of the Data Analytics world by offering a vast amount of data to be collected and analysed for application development and more. There are challenges with IoT frameworks: creating real-time applications, difficult low-level programming, the large number of domains, the heterogeneity of a humongous number of sensor devices, and more. There are proposed frameworks that are working on these challenges: Mobile Fog, PatRICIA, COMPOSE API, SDG-Pro, LEONORE, and others. Programming abstraction and locating data processing closer to devices (rather than just in the cloud) are two of the biggest solutions to the challenges faced.**

I. Introduction

A. IoT - Internet of Things

IoT is an emerging new concept in the data analytics and developer applications world. The IoT is a very large amount of small, heterogeneous sensory-devices usually having small storage and processing capacities [1],[2]. [3] predicts that these devices will become better at processing more complex data by itself. 300 thousand IoT devices existed in 1990 and it's forecasted to be up to 20.6 billion in 2020[1]. These devices, also referred to as edges (leafs of IoT system) are internet-connected directly or via another channel [4] so that their data can be used by organizations from multiple domains to make useful applications [5],[6]. The data generated from these edge devices is high speed, high volume, and can be dispersed about a wide geographical area [2].

B. IoT Framework

[7] define a IoT framework as a set of standards and protocols to aid in the creation of IoT applications. They further state that a good framework makes the development of applications easier and more intuitive in the following ways: implementation, managing, security, and malleability for use on multiple differing platforms. IoT frameworks originally used large data centers, but as years went by the cloud was used more in order to centralize infrastructure [6]. They also indicate that there are new frameworks using edge devices and sensor gateways to do their own processing for higher efficiency. Fog Computing is a framework concept in which a some of the processing is done in the network, rather than strictly in the cloud infrastructure [8]. Routers and computing nodes in the Fog framework can process data closer to its creation.

[1]https://www.analyticsvidhya.com/blog/2016/08/10-youtube-videos-explaining-the-real-world-applications-of-internet-of-things-iot/ [Accessed March 12, 2017]

II. Challenges of IoT Frameworks

A. Time-Sensitivity

Most clouds are not distributed near the edge devices so an application can have inefficiency in offering real-time processing [4],[6]. [2] suggests that edges should be enabled with more processing power to overcome the latency in applications and [3] believes that future devices will have higher RAM and storage functionalities to enact this change. Since there are so many devices in a IoT framework, bandwidth is lowered, thus slowing an application [9].

B. Programming

Developers have a difficult task in writing applications due to programming complexities because they have to deal with low-level programming for many different types of devices [10],[1]. [6],[1],[11] believe that abstracting low-level programming languages through a new intermediate layer is key to overcoming issues involving heterogeneity, multiple domains, and multiple languages. There can also be a myriad of different programming languages that a developer needs to know when using some frameworks [12],[11] causing more delays in development. [13],[14] stipulated that an API should be enacted for frameworks to make easy access to data by the abstraction being enabled through it. Scaling applications is also slowed by this programming complexity [6].

C. Domains

There are many domains that span the IoT and a developer needs know them to make an application in a specific one [2],[6],[5],[11]. This becomes tedious because when a developer makes an application for a new domain they have to do a great deal of new research understand data to be processed. Bad metadata is one of the causes of a developer needing to

learn domain because they can not decipher the data coming from the devices intuitively [15].

D. Edges

All edge devices are not the same (heterogeneous) which makes them hard to deal without the idea of programming abstraction [6] and also makes it tough to integrate them[15]. Sometimes edges and sensors go idle because of malfunctions and this can cause issues in the framework [2],[5]. They're also hard to manage because of the vast number of them [6] of which sometimes need to be configured manually [3].

E. Other

There isn't a standard protocol for a frameworks which makes it tough to use data gathered from devices with platforms across domains [16], [13],[15]. [11] states there can be changes in business needs and devices themselves which make framework management more difficult. The speed and volume of data can also cause problems if framework can't keep up with them [2].

III. Proposed Frameworks

A. Mobile Fog

[4] proposed the Mobile Fog framework for applications that are time-sensitive and have a wide geographic distribution of edges. The physical structure is a cloud infrastructure that is connected to a bunch of Mobile Fogs (like sub-clouds) that are distributed near edge devices. The edges send their raw data to a local Fog that processes data and sends it to the cloud or an application. This saves bandwidth because an edges raw data doesn't have to be across network directly to the cloud to be processed. They abstracted low-level language details with a high-level model using an API so that developers could create applications more easily. The code used is the same in all areas of the framework: cloud, fog, and edges. Their model scales well by using load balancing. When there are a lot new device generated in an area, a new Fog node is created. If there is a decrease in data being created, then Fog nodes can be combined. They simulated a situation where cars on a highway could communicate with each other via Fog nodes. They found their Mobile Fog idea worked much faster than using a cloud on its own. In the future they plan to make their framework more secure and reliable for a real-world launch.

B. PatRICIA

[1] created PatRICIA and they also use programming abstraction to better the usefulness and intuitiveness of their framework. They state there is less flexibility in framework because of this abstraction, but it is traded for more scalability and ease of use. They did a case study involving electric golf carts to test their model. They were able to monitor the carts via an on-board gateway to see if there were any failures. If cart had a high-energy fault, it would automatically be put into a low energy state and its status would be sent to interested users. They used a cloud service as opposed to Fogs concept. They noted that there approach is great to use across many different domains due to its flexibility by programming abstraction.

C. COMPOSE API

The COMPOSE API project created by [17] proposed it could process edge data in real-time by using a REST API to transfer data. They discussed the Web of Things (WoT) being a place where devices and other objects could communicate with each other by using a standard set of communication protocols. All entities need to use the COMPOSE API protocols, causing the WoT to have a common communication language. They think their API is highly efficient because all the devices are integrated and API reduces network costs by using common resources like HTTP.

D. SDG-Pro

The SDG-Pro Framework (Software-Defined Gateways) was proposed by [6] which also uses abstraction to hide low-level tedious programming. Their model has an API that shows a system-wide view of platform and also enables governance and provenance control. The API can be used to audit the framework in order to trace the history of events inside itself. They did case studies on electric cars where they were able to code automatic maintenance checks for cars with a small amount of code. In the future they want to have more support for erratic edges and a better API that covers more use cases.

E. LEONORE

LEONORE is a framework proposed by [3] which allows for malleable provisioning on edge devices. It allows the edge to schedule its own provisioning during idle or off-peak times. This elastic provisioning can reduce network congestion as well. It also allows greater universal control by allowing system-wide software updates to devices and other infrastructure entities. Security issues can be dealt with universally as well. They state it is easily scalable because they put network nodes near devices (similar to Fog) and also because they allow some devices to self-process data. They saw the need in tightening up security issues like authorizations and authentications.

F. Other Noteworthy Frameworks

[18] talked about a delay-tolerant framework for RFID sensor networks (RSNs) which is very scalable, minimizes bandwidth, and is reliable. [10] proposed a framework that could be used by a non-programmer to make application development more accessible. Calvin, as proposed by [12], is a hybrid framework prototype written in Python that uses its own abstraction language and has a limited number of protocols for cross-platform ease. It also embraces heterogeneity, by using it to optimize its design. ScriptIoT, as proposed by [5] defines itself as a middleware (programming abstraction layer) and is polling and event-driven. [15]'s framework uses semantic web languages to get information out of raw sensor data which drastically decreases the domain knowledge a developer would need to know.

IV. Conclusion

It is easy to see that the IoT will become more engrained in our society because it allows more efficiencies in life: smart homes that save energy or smart cars that self-monitor themselves as show in PatRICIA's case study. There has been challenges in IoT frameworks that many people have been working on to fix. Mobile Fog, PatRICIA, and COMPOSE API all are worked towards solving the issue of real-time processing for applications. Most frameworks abstracted low-level programming in order to drastically shorten application development time through intuitive abstraction languages. The use of semantic web languages greatly eases the amount of domain knowledge a developer needs to know by deriving information out of raw data from sensor devices. Challenges have been greatly reduced over the past few years by the many researchers shown in this review. Two of the biggest aids have been programming abstraction and processing of data closer to edges by using Fog concept.

Acknowledgment

The authors would like to thank...

References

[1] S. Nastic, S. Sehic, M. Vogler, H.-L. Truong, and S. Dustdar, "Patricia–a novel programming model for iot applications on cloud platforms," in *Service-Oriented Computing and Applications (SOCA), 2013 IEEE 6th International Conference on*. IEEE, 2013, pp. 53–60.

[2] J. Krishnamurthy and M. Maheswaran, "Chapter 5 - programming frameworks for internet of things," in *Internet of Things*, R. Buyya and A. V. Dastjerdi, Eds. Morgan Kaufmann, 2016, pp. 79 – 102. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780128053959000058

[3] M. Vögler, J. M. Schleicher, C. Inzinger, and S. Dustdar, "A scalable framework for provisioning large-scale iot deployments," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 2, p. 11, 2016.

[4] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenw¨alder, and B. Koldehofe, "Mobile fog: A programming model for large-scale applications on the internet of things," in *Proceedings of the second ACM SIGCOMM workshop on Mobile cloud computing*. ACM, 2013, pp. 15–20.

[5] H.-C. Hsieh, K.-D. Chang, L.-F. Wang, J.-L. Chen, and H.-C. Chao, "Scriptiot: A script framework for and internet-of-things applications," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 628–636, 2016.

[6] S. Nastic, H.-L. Truong, and S. Dustdar, "Sdg-pro: a programming framework for software-defined iot cloud gateways," *Journal of Internet Services and Applications*, vol. 6, no. 1, p. 21, 2015.

[7] H. Derhamy, J. Eliasson, J. Delsing, and P. Priller, "A survey of commercial frameworks for the internet of things," in *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. IEEE, 2015, pp. 1–8.

[8] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[9] I.-Y. Ko, H.-G. Ko, A. J. Molina, and J.-H. Kwon, "Soiot: Toward a user-centric iot-based service framework," *ACM Transactions on Internet Technology (TOIT)*, vol. 16, no. 2, p. 8, 2016.

[10] H.-S. Choi and W.-S. Rhee, "Iot-based user-driven service modeling environment for a smart space management system," *Sensors*, vol. 14, no. 11, pp. 22 039–22 064, 2014.

[11] P. Patel and D. Cassou, "Enabling high-level application development for the internet of things," *Journal of Systems and Software*, vol. 103, pp. 62–84, 2015.

[12] P. Persson and O. Angelsmark, "Calvin–merging cloud and iot," *Procedia Computer Science*, vol. 52, pp. 210–217, 2015.

[13] J. Yun, I.-Y. Ahn, S.-C. Choi, and J. Kim, "Tteo (things talk to each other): Programming smart spaces based on iot systems," *Sensors*, vol. 16, no. 4, p. 467, 2016.

[14] S. Bendel, T. Springer, D. Schuster, A. Schill, R. Ackermann, and M. Ameling, "A service infrastructure for the internet of things based on xmpp," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*. IEEE, 2013, pp. 385–388.

[15] A. Gyrard, P. Patel, A. Sheth, and M. Serrano, "Building the web of knowledge with smart iot applications," *IEEE Intelligent Systems*, vol. 31, no. 5, pp. 83–88, 2016.

[16] F. Khodadadi, A. V. Dastjerdi, and R. Buyya, "Simurgh: A framework for effective discovery, programming, and integration of services exposed in iot," in *Recent Advances in Internet of Things (RIoT), 2015 International Conference on*. IEEE, 2015, pp. 1–6.

[17] J. L. Pérez, Á. Villalba, D. Carrera, I. Larizgoitia, and V. Trifa, "The compose api for the internet of things," in *Proceedings of the 23rd International Conference on World Wide Web*. ACM, 2014, pp. 971–976.

[18] F. M. Al-Turjman, A. E. Al-Fagih, W. M. Alsalih, and H. S. Hassanein, "A delay-tolerant framework for integrated rsns in iot," *Computer Communications*, vol. 36, no. 9, pp. 998–1010, 2013.