Overview of the Current Status of NoSQL Database

Article · April 2019

CITATIONS
4

READS
1,135

3 authors:

Yasmin Hisham Rasheed
Applied Science Private University
1 PUBLICATION **4** CITATIONS
SEE PROFILE

Mahmoud H. Qutqut
Applied Science Private University
20 PUBLICATIONS **175** CITATIONS
SEE PROFILE

Fadi Almasalha
Applied Science Private University
23 PUBLICATIONS **274** CITATIONS
SEE PROFILE

Overview of the Current Status of NoSQL Database

Yasmin Rasheed, Mahmoud H. Qutqut and Fadi Almasalha

Faculty of Information Technology, Applied Science Private University, Amman, 11931 Jordan

Summary
Nowadays with the accelerated development of the Internet and Cloud computing, the fast growth of technology generates a massive amount of data. Businesses and people generate these data by using web apps, social media, and new technologies. These data, in general, could be structured, semi-structured or unstructured. Because of the different types of data and the big data that is generated, there is a need for a database to be able to store and process these data effectively to enhance the performance when reading and writing. So, there is a need for a new design for the database, and it is not suitable for storing, analyzing and performing data in a relational database for big data. In addition to that, many new challenges faced the traditional relational database; especially in the applications that required large scale and high concurrency such as search engines. In response to that, NoSQL has developed to solve these types of problem. NoSQL database has many advantages that make it gain significant popularity over the last few years and used widely. It read and write the data quickly, expands easily, low cost and many other features. In this paper, we overview the NoSQL database and its characteristics in the field of the Internet of Things (IoT). We also provide two representative use cases of using the NoSQL database in current technologies.

Keywords:
NoSQL; database; relational database; IoT.

1. Introduction

Relational database management system (RDBMS) has been adopted since the'70s. That is why it is a mature technology for storing data and their relationships [1]. Also, every RDBMS must ensure four properties or characteristics in the transaction that is known as an ACID (Atomicity, Consistency, Insolation, Durability). Atomicity where all the tasks of a transaction are executed or none of them will be executed. Consistency is the operation takes of the database from one consistent state to another equally consistent. Where Insolation describes the effect of a transaction is not visible to other transactions until it is committed. Durability defines the changes made by authorized transactions are permanent. These characteristics have a cost, and it will generate a cost so that they are guaranteed [2]. In addition to that, recently big data analysis becomes the core of modern science and commercial. It is generated by users from online uses such as emails, videos, audios, images, logs, posts, search queries, health records, social networking interactions, science data, sensors and mobile phones and their applications. These data are stored in databases; they are either structured or unstructured form of data, so we face some problems like as how to capture, store, manage, share, analyze and visualize them via common database [3].

The main issues for researchers are that the data growth rate exceeds their ability to design appropriate cloud computing platforms to evaluate data and update the workload problem. Because of these problems, there is a need to modify the database as traditional relational databases have proved to be weak for the distributed environment. To solve the main issues and for better performance and scalability, a new database released by scientists [3] called NoSQL. NoSQL has appeared because it is a flexible, scalable and schema-free database. NoSQL means "Not Only SQL." It provides storage and retrieval mechanism with less constrained consistency models than traditional relational databases [3].

Regarding NoSQL design, some concepts must be taken into consideration such as the CAP theorem or Brewer's theorem. The designers of distributed systems suffer a fundamental tradeoff limitation that they must choose only two out of these three properties CAP, which are data consistency, system availability or tolerance to network partitions. This theorem states that it is impossible in a distributed system to guarantee the following properties simultaneously [1].

- Consistency: If this attribute is satisfied, once the data are written, it guarantees that this data is available and up to date for all users using the system.
- Availability: This attribute means that the service is offered continuously and without interrupt or degradation within a specific time.
- Partition Tolerance: This property means that a transaction or a process can be done entirely even when there is a part of the network failed.

Eric Brewer in 2000 estimates that at any specific time only two out of the three properties that we mentioned can be guaranteed. After that, Gilbert and Lynch proved this estimation; they conclude that in distributed systems only a specific combination can be created; AP (Availability-Partition Tolerance), CP (Consistency-Partition Tolerance) or AC (Availability-Consistency). To this end, we provide an overview of the NoSQL database and its characteristics

Manuscript received April 5, 2019
Manuscript revised April 20, 2019

in the field of the Internet of Things (IoT). We also present two representative use cases of using the NoSQL database in current technologies.
The structure of this paper is as follows. We will start with a brief background of the NoSQL database and its categories in Section 2. We will determine the difference between the RDMS and NoSQL and the benefits of the NoSQL database over RDBMS in Section 3. In Section 4, we provide two representative use cases and applications of the NoSQL database in current technologies followed by a conclusion in Section 5.

2. Background

Over the last few years, NoSQL databases have gained high popularity with both, developers who make new systems and organizations who want to improve their business. Both are trying to adapt their information systems to meet today's data requirements. The leaders of NoSQL databases were massive web companies such as Google, Amazon, and Facebook to promote them build and support their businesses. After they made the NoSQL public and open source, other giant web companies such as Twitter, Instagram and Apple started to use them [4].

- Developers are dealing with applications that create high volumes of rapidly changing data types; structured, semi-structured, and unstructured.
- There are no more twelve to eighteen months in the waterfall development cycle. Now a group of teams works on sprint-agile, which has iteration and generates code every week or two.
- Applications used by broad audiences required to be always-on, accessible from many different devices and scaled globally to millions of users.
- Instead of large monolithic servers and storage infrastructure organizations are now moving to scale-out architectures using open source software, commodity servers, and cloud computing.

In these situations, relational databases were not designed to deal with the scale and agility challenges that face modern applications [3]. There are different reasons to move to NoSQL database. Some reasons are mentioned below [3].

- The growing of Big Data and high data velocity, data variety, data volume, and data complexity.

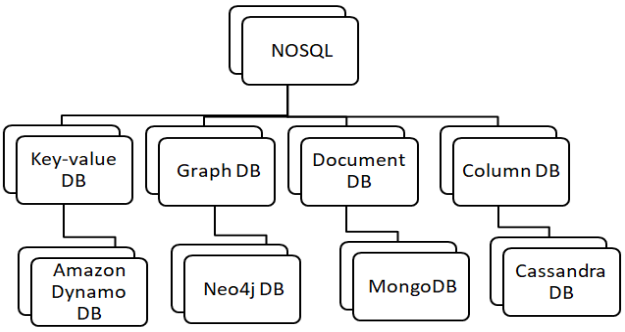


Fig. 1 Types of NoSQL Database.

- Data is always available.
- Reallocation transparency.
- A new era of transactional capabilities.
- Data architecture is flexible.
- High-performance architecture.
- Highly Intelligence.

There are different types of NoSQL database which are represented in Figure 1 and discussed below with an example in each kind.

2.1 Key-Value Database

These databases assign a key to a value or set of values [5]. The keys are unique and atomic. These keys are used to query for entries in the key value storage databases. The Key-value store provides a hash table structure with key-value pairs spread across several remote servers in a distributed cluster. So, they can achieve the required efficiency by providing fast random read/write requests and flexibility to store data in the schema-less format. Since pairs of different key values save a group from irrelevant data, it avoids SQL join and group by operations as well as foreign key references. Twitter use key-value stores to store tweets using a unique Twitter ID. The corresponding values may include the original message, User ID and time of sharing. One of the cases of key-value stores Amazon Dynamo DB. Figure 2 shows the key-value database system.

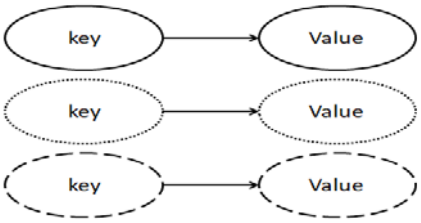


Fig. 2 Key-Value Database.

Amazon Dynamo DB - Dynamo DB uses key-value data structures that are designed to scale quickly with a flexible schema. Also, it supports querying, updating and storing documents. We can write applications that store JavaScript Object Notation (JSON) documents directly in Dynamo DB tables by using the Amazon Web Services software development kit (AWS SDK). Each item in Dynamo DB (row) is a key-value pair which has a primary key attribute that uniquely identifies each item. Dynamo DB is fast and has high adaptability. Here, data encryption is not supported; instead of that, they use https protocol for the communication between the client and the server. Dynamo and requests control authentication and approval must be marked using Hash-based Message Authentication Code (HMAC)-SHA256. Amazon dynamo gives a consistent replication mechanism with consistency levels that are what makes it a different approach [6]. Dynamo DB has a unique feature which is that the task of the database administrator is automated. It monitors issues of scalability, provisioning, load balancing, reliability, elastic map reduce integration and makes sure that there is replication synchronously to ensure no data is lost [7]. Dynamo DB has many features as described below [7].

- The users can specify how they want their performance to be in terms of the number of reads and writes per second. Dynamo DB will provide services based on their requirement.
- Eventual consistency and durable consistency are to types of readings offered by Dynamo DB. The reads are eventually consistent by default, but the user can choose strong consistency that after a write operation the updated value will be visible.
- When there is any modification or change in the level of provisioning, this does not lead to data loss or disruption on the application program.
- Because Dynamo DB stores data using solid-state drives rather than hard disk drives; retrieving the stored data is much faster than other NoSQL databases.

In the following, we describe cases that we can use Dynamo DB in them.

- Applications that have simple create, update, delete operations performed over an extensive data set (e.g., online gaming).
- In Amazon's cart, they use it while doing online shopping to store items.

However, Dynamo DB is not recommended for the following cases.

- For applications that have many relational JOIN operation or normalization of data.
- When the number of reads and writes per second in any application get change very fast in a way that passes the read/write specified limit.

2.2 Graph Database

These databases store the data in a graph structure. Data is represented by edges and nodes, each with their features and attributes. Most graph databases provide a useful graphical traversal, even when the nodes are on separate physical vertices. Lately, the graph database has received much attention because of its applicability to social media data. This opens the way for new implementations to accommodate the current market. However, several authors exclude graph databases from NoSQL because they do not fully align with the relaxed model constraints typically found in NoSQL implementations. However, others include it because they are mostly non-relational databases and have many applications nowadays [1]. Graph database includes Neo4j. Figure 3 shows the concept of the graph database.

Neo4j-Nowadays many companies and organizations used neo4j in different industries such as government, financial services, technology, energy, retail, and manufacturing. Neo4j is an open-source NoSQL database that provides an ACID-compliant transactional back-end for applications [8]. Neo4j uses graph model. Nodes and edges have associated properties with them. The nodes can also be linked with labels, and they classify according to their roles [9]. They have used Neo technology for developing Neo4j, the implementation was in Java, and other software are written in other languages can access using Cypher Query Language (CQL) by using a transactional HTTP endpoint, or through the binary “bolt” protocol. The main features of CQL are described below [4].

- The way it works to extract information or modify the data that matches patterns of nodes and relationships in the graph.
- It deals with parameters, restricted elements, and variables that indicate named.
- CQL can create, update, and remove nodes, relationships, labels, and properties.
- CQL manages indexes and constraints.

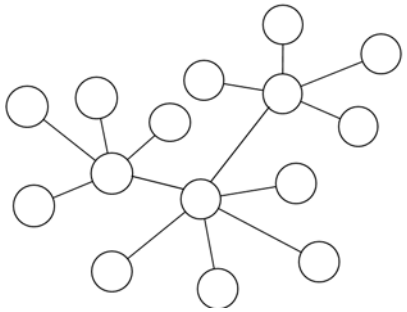


Fig. 3 Graph Database.

2.3 Document Database

Document-oriented model is used to store data as a record with its linked data as a single data structure called a document [4]. Each document contains multiple related attributes and values. Documents can be retrieved based on attribute values by using the various APIs or query languages provided by the DBMS system their schema-free organization of data characterizes them. Which mean that the record does not follow a specific structure, i.e., different records may have different attributes. The types of values of individual attributes can be different for each record. To store the records; document stores use a format such as JSON or XML. This helps the records to be processed directly in applications. The documents are stored and retrieved by using a key [4]. One of the examples of a document database is MongoDB. Figure 4 shows the document databases system.

MongoDB - MongoDB is document free and open source database published under the GNU Affero General Public License [10]. The data in Mongo is stored in flexible JSON-like documents. The fields could be different in each document, and over time the structure of the data can be modified [10]. Replication in MongoDB called replica set; it is a collection of MongoDB servers that maintain the same data set and provide data redundancy [4]. We summarize the features of MongoDB below [7].

- MongoDB provides high performance by offer indexing of every attribute in a document.
- It can scale-out without disrupting application, and it supports sharding mirroring and load balancing of data across nodes.
- It uses capped collections which are like the concept of circular buffers that provide high throughput performance.

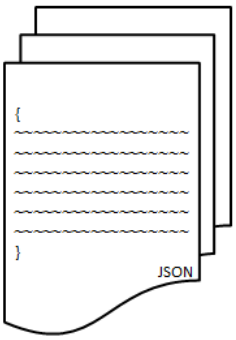


Fig. 4 Document Database.

2.4 Column Database

This type of databases instead of saving data by row (as in relational databases), they store data by column [1]. Therefore, some rows may not contain a portion of the columns, providing flexibility in data definition and allowing to apply data compression algorithms for each column. Columns that are not frequently used or queried together can be distributed across different nodes. Cassandra DB is an example of a column database.

Cassandra-Cassandra is free and distributive open-source broad column database developed by Apache [3]. The language used to write Cassandra is java, so it can be used by any platform that has a Java virtual machine (JVM) [3]. Cassandra is intended to handle enormous amounts of data across several commodity servers [3]. So, this will provide high availability with no single point of failure. To handle fault tolerance, the data in Cassandra replicated automatically to multiple nodes. The replication is performed via various data centers. There is no downtime when replacing failed nodes [11]. The following are some features of Cassandra DB [7].

- Linear scalability is offered by Cassandra even if the workload is enormous and the throughput performance will not change while crunching an extensive set of data.
- Gossip protocol is used by Cassandra to communicate an update message to all replicas simultaneously.
- In Cassandra reading, writing and updating are simple. It uses built-in queries; hence, it provides a good experience for the user.

Cassandra DB is highly recommended and should be used in the following cases.

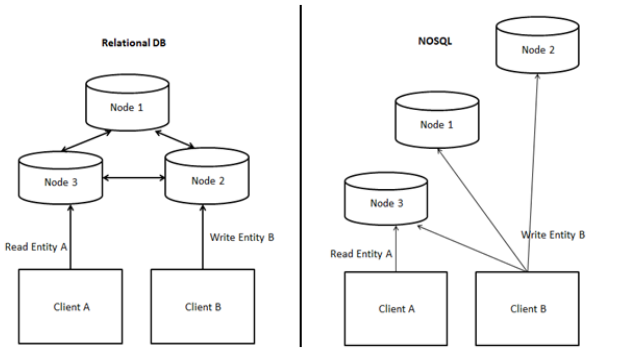
- In the case of apps that the number of reads is more than the number of writes such as Twitter.
- There are some applications where immediate consistency is not a significant concern. In this case, it is recommended to use Cassandra.
- An application needs high maintenance of code.
- There are web applications that provide dynamic schema and content to users such as Netflix.

3. NoSQL and the Relational Database

The NoSQL database maintains consistency models that are constrained only to the relational database contrast to collect and retrieve data [3]. Because of real-time web applications and big data; NoSQL has been optimized for use in various fields of engineering and traditional industries. The primary process for NoSQL is to simplify retrieval and to attach extensive data using data processing

operations. NoSQL key features are horizontal scalability, the simplicity of design and more delicate control over the availability. These technical features demonstrate effective and reliable results [12]. The relational databases load the data vertically as shown in Figure 5. This is not the case in NoSQL since it does not distribute data in this style. The relational database does not spread logical entities across multiple tables, as they are stored in one place. Relational databases do not guarantee referential integrity between logical objects. This feature enables them to distribute data across a significant number of database nodes and write independence [3].

NoSQL is an appropriate approach to deal with big data. Companies like IBM, Amazon, Facebook, Twitter, Google, Oracle, are now applying high performance conveyed NoSQL arrangements. Comparing NoSQL with RDBMS, NoSQL DB is more scalable and provide high performance, and their data model addresses several issues that the RDBMS is not designed to address [3]. Regarding the Internet of Things (IoT) domain, over the last few years until now the IoT has been used in many areas. IoT concept indicates numerous smart devices that are connected to the Internet [12]. IoT applications need to serve a high number of users, quick response to all users that are globally distributed, available all the time (no downtime), deal with different types of data, semi- and unstructured data [13]. These applications will generate a massive amount of data and with the heterogeneous data that are created; the problems to store, transfer and manage the data efficiently will appear. If we use the RDBMS that uses Structured Query Language (SQL) with these applications, we will face its static schema which is the main limitation that makes the RDBMS not suitable for IoT applications. So, the NoSQL database is a schema-free, no joins, and horizontally scalable database [7].



t hat

Fig. 5 Relational DB vs. NoSQL.

SQL Databases: to store the data in the SQL database; the relational data model is used. This model uses tables and stores the data in rows and columns, and these tables can be linked.

NoSQL Databases: NoSQL use non-relational data model which is schema-free that store the data in different forms as we mentioned earlier; document, graph, key value, and column. NoSQL has gained significant reputation because the features provided, such as high scalability, easy access, and distributed architecture [12]. Table 1 shows the differences between the SQL and NoSQL databases based on the IoT point of view. We describe the aspects in the following.

- Scalability: vertical scalability means that more resources will be added to the node such as memory or processors to enhance the performance of that node. But in horizontal scalability, the system load is divided into several nodes (servers) without the need to have more resources. Necessarily, in IoT application, the database will need to expand, therefore choosing the database that can grow will be a practical choice.
- Data retrieval: In SQL the tables are connected. To retrieve or to search for data from several tables, the JOINS statements are used by the user to view the data. On the other hand, the data in NoSQL is stored in the form of objects that contain all the related data. In this case, the process of combining and then view the data will be eliminated.
- System Maturity: SQL has been used for a long time, and it is practised technology. Therefore, most of the obstacles and issues have been solved. Security features like authentication, data confidentiality and integrity are incorporated in SQL. On the other hand, NoSQL considered as a new and not mature technology hence security issues not solved yet, and that may generate more security issues.

Table 1: SQL vs. NoSQL Databases.

Aspects	SQL	NoSQL
Scalability	Vertical scalability	Horizontal scalability
Data Retrieval	Time consuming process	Save response time
System Maturity	Mature technology hence highly secured	Not mature technology, hence lack of security

4. Current Use Cases and Applications of NoSQL Database

In this section, we present two current representative applications and use cases of using the NoSQL database.

4.1 Using a NoSQL graph-oriented database to store accessible transport routes

Some organizations aim to enhance public transport accessibility for people with disabilities, and this is what was proposed by the World Health Organization in its "World Report on Disability 2011" [14]. Different websites and applications solve the problem of public transport and its accessibility. But none of them offers general mechanisms that achieve available transportation data. There was a significant shortage of open and reusable data concerning public transport and its accessibility. Based on that, they proposed to develop a technological framework that can process manage and exploit open data that aim to raise the ability to access to city public transport under the scope of the Access@city project [14]. They concentrated on the design and storage of accessible transport routes, by using crowdsourcing techniques, in a NoSQL graph-oriented database. So, they defined an open data repository for convenient public transport under the Access@city project. They used a NoSQL database to develop it because of its high ability to deal with and manage vast volumes of information in addition to its scalability and flexibility. They have chosen a NoSQL graph-oriented database specifically; because they are going to deal with highly connected data and there will be queries that are more efficient in a graph-oriented database. By using a methodological approach, they developed and designed the graph-oriented database from scratch. They have selected Neo4j which is the most popular graph-oriented database according to the database ranking. For that, they have developed a native Android application; the application users can register for the generation of accessible routes. There is starting a route option that they can use which have choices about the particular need (wheelchair, bike, baby stroller, etc.) they will have in the journey. The application will periodically register the GPS position. There are two options the users have when the trip is finished either discard the route or save it. Comments about the paths taken may be added by the users in addition to that they can tell about possible incidents and/or including photos. In this case, they have decided to choose the graph-oriented database for the development of big data repository, that is because the data of the routes is highly connected and there are many ways will be used for querying the data [14].

4.2 Distributed architecture of mobile GIS application using the NoSQL Database

As a college student that on campus, you would like to know about every event, or any occasion happened on the campus. The idea here is to benefit from the distributed architecture of a mobile GIS (geographic information system) application [15]. A group of students built a GIS-based app for their campus. The app aims to provide location-based information to the students on the campus, information such as events, maps or any other useful or related information. The application keeps track of the user (student) current location. Once the user enters a predefined polygon structure such as a building that is already stored in the database, any relevant information about that building will be shown to the user. It could be an event in that building or a workshop hosted in a particular floor or a room. For the database design, they looked for a database that has high flexibility, ease of use and quick deployment. So, they chose NoSQL MongoDB because of the significant popularity of NoSQL databases. They decided to benefit from the advantages of this relatively new technology for their application. In their case, the college campus is periodically changing the layout of building they require a flexible database to store the building location information so that is why the choose NoSQL database because it will offer a straightforward future change to their data models as the campus changed over time. They used MongoDB, and it was preferable over SQL because of easy scalability, quick startup development, and ease of creating flexible data models [15].

5. Conclusion

In this paper, we overview the current status of the NoSQL database. We described all NoSQL database types and presented their applicability to different domains along with an illustrative example of each type. Also, the applicability of the NoSQL database for the Internet of Things (IoT) domain is discussed. We detailed two current representative use cases of the NoSQL database. Furthermore, we have noticed that people and organizations are moving on SQL to NoSQL because of the high performance and scalability and other features provided by NoSQL. NoSQL database has many features in the perspective of the massive amount of storage management and their utilization. However, security is a significant concern for IT infrastructure. Security in NoSQL databases is weak; authentication and encryption are non-exist or very weak. There should be a new solution to enhance security to improve the use of resources in the future.

Acknowledgment

This work made possible by financial support from Applied Science Private University in Amman, Jordan.

References

[1] A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S. Schiaffino, Persisting big-data: The NoSQL landscape, Information Systems, 63, pp. 1–23, 2017.

[2] F. Oliveira, A. Oliveira, and B. Alturas, Migration of relational databases to NoSQL-methods of analysis, Mediterranean Journal of Social Sciences, 9 (2), pp. 227–235, 2018.

[3] A. Haseeb and G. Pattun, A review on NoSQL: Applications and challenges. International Journal of Advanced Research in Computer Science, 8 (1), 2017.

[4] W. Hauger and M Olivier, NoSQL databases: Forensic attribution implications, SAIEE Africa Research Journal, 109 (2), pp. 119–132, 2018.

[5] Amazon official website, Amazon web services (AWS) - cloud computing services, https://aws.amazon.com/.

[6] A. Corbellini, C. Mateos, A. Zunino, D. Godoy, and S. Schiaffino, Persisting big-data: The NoSQL landscape, Information Systems, 63, pp. 1–23, 2017.

[7] P. Srivastava, S. Goyal, and A. Kumar. Analysis of various NoSQL database, in International Conference on Green Computing and Internet of Things (ICGCIoT), pp. 539–544, IEEE, 2015.

[8] Neo4j official website, what is a graph database and property graph - neo4j, https://neo4j.com/developer/graph-database/.

[9] A. Gupta, S. Tyagi, N. Panwar, S. Sachdeva, and U. Saxena, NoSQL databases: critical analysis and comparison, in International Conference on Computing and Communication Technologies for Smart Nation (IC3TSN), pp. 293–299, IEEE, 2017., 2015.

[10] MongoDB official website, MongoDB for giant ideas, https://www.mongodb.com/.

[11] Apache Cassandra by intracluster official website: https://www.instaclustr.com/.

[12] S. Rautmare and D. Bhalerao. Mysql and NoSQL database comparison for IoT application, in IEEE International Conference on Advances in Computer Applications (ICACA), pp. 235–238, 2016.

[13] Couchbase official website, why NoSQL database? https://www.couchbase.com/resources/why-nosql.

[14] B. Vela, J. Cavero, P. C´aceres, A. Sierra-Alonso, and C. Cuesta, Using nosql graph oriented database to store accessible transport routes, in EDBT/ICDT Workshops, pp. 62–66, 2018.

[15] J. Rodriguez, A. Malgapo, J. Quick, and C. Huang, Distributed architecture of mobile gis application using NoSQL database, Journal of Computing Sciences in Colleges, 33 (3), pp. 68–68, 2018.

Yasmin Rasheed is an MSc student at the Faculty of Information Technology at Applied Science Private University in Amman, Jordan. Yasmin received her BSc degree in computer science from Applied Science Private University in 2017. Her research interest focus on security for massive IoT.

Mahmoud H. Qutqut is an Assistant Professor at the Faculty of Information Technology at Applied Science Private University in Amman, Jordan since October 2014. He received his Ph.D. degree from the School of Computing at Queen's University in Canada 2014, under the supervision of Prof. Hossam Hassanein. He received his MSc degree in Telecom Systems from DePaul University at Chicago, Illinois in 2007 and BSc degree in computer systems from Applied Science University in 2015. He has served as a TPC co-chair and a technical program committee member for several IEEE international conferences. His research interests include mobile heterogeneous small cells networks, Internet of Things (IoT) enabling technologies, and smart cities enabling services.

Fadi Almasalha is an Associate Professor at the Faculty of Information Technology at Applied Science Private University in Amman, Jordan; received his M.S. in Computer Science from New York Institute of Technology, in 2005 and Ph.D. in Computer Science from the University of Illinois at Chicago, in 2011. In fall of 2011, he joined the Department of Computer Science at the Applied Science University. Dr. Fadi Almasalha received his Associate rank in 2016, during his appointment as the head of the computer science department. Dr. Fadi has published more than ten technical papers, journals and book chapters in refereed conferences and journals in the areas of multimedia systems, data mining, and cryptography.