ISSN 2347 - 3983
Volume 8. No. 9, September 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter243892020.pdf
https://doi.org/10.30534/ijeter/2020/243892020

Enhancing the Semantic Web of Things: A fog-centric Architecture and a Geospatial Indexing Method

Ismail Nadim[1], **Yassine El Ghayam**[2], **Abdelalim Sadiq**[3]
[1]Ibn Toufail University Faculty of science Kenitra, Morocco, ismail.nadim.gi@gmail.com
[2]SMARTiLab EMSI Rabat Honoris Universities, Morocco, yassine.elgh@gmail.com
[3]Ibn Toufail University Faculty of science Kenitra, Morocco, a.sadiq@uit.ac.ma

ABSTRACT

The deployment of interoperability, mobility, security and scalability mechanisms is essential for IoT. While, these mechanisms are resources consumers, IoT devices are resources constrained. This dilemma requires the use of an IoT architecture capable of meeting compute and storage needs while remaining close to data sources. In this article, we propose a distributed IoT architecture based on Fog computing. This architecture does not marginalize the cloud and edge computing. But, tries to balance the role of each paradigm according to the resource needs dictated by the IoT challenges. Besides, we propose a distributed indexing structure constructed by encoding the loT resource locations in form of geohashes that we use in conjunction to the Prefix Hash Tree (PHT) structure to index and store the semantic description of the IoT resources in the fog layer. The features of our index are many: (1) it allows to aggregate resources with similar geohash in order to reduce the size of the index. (2) it takes advantage of the flexibility of the P2P structure. (3) it improves the speed and accuracy of the discovery leveraging the performance of the PHT structure. Last but not least, the index allows the efficient discovery of the mobile objetcs.

Key words : Internet of Things, Semantic Web of Things, Fog Computing, Edge Computing, Interoperability, Indexing.

I.INTRODUCTION

Information and Communication Technology (ICT) has witnessed a remarkable development in recent years. Especially, with regard to reducing the size of electronic devices, reducing their prices and improving communication protocols between them. As a result, the number of devices connected to the Internet increased significantly, which contributed in forming the Internet of Things (IoT) [1]. The Internet of Things is used in many fields such as the industry, healthcare, agriculture, etc. However, it is still facing many challenges particularly the interoperability between heterogeneous data and devices, the scalability, the mobility and the security.

With the emergence of the Semantic Web of Things (SWoT) as a promising approach for addressing the interoperability, suitable environment for processing and storing semantic IoT data becomes essential. Such environment will not only helps in handling the integration of the semantic technologies, but will improves the design and the performance of mobility and security solutions.

As the IoT devices are, in general, resource constrained, three potential solutions are currently leveraged; the cloud computing, the fog computing and the edge computing. The cloud computing paradigm is widely used in the IoT, for the coordination between the end devices as well as for the data management and storage. However IoT devices are geographically distributed, consequently, data transmission latency between cloud and smart objects remains a critical issue to the applications that have sensitive delay requirements. Besides, the data being hosted far from the user is still raising security and privacy issues.

The fog computing paradigm [2] was recently proposed by the industry, enables real-time interaction and location-based services. In particular, the local processing capacity of fog computing significantly reduces the volume of data sent to the cloud.

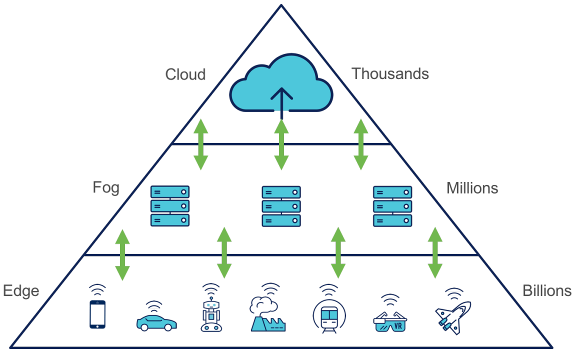

Figure 1: Cloud, Fog and Edge Computing

Similarly to the fog computing, the edge computing [3] concern is to leverage the local computing capabilities. The main difference between fog and edge is at where the data processing takes place. Edge concerns the computing carried out at the devices to where the sensors are connceted. Fog is different in this aspect because the data processing is moved to the processors which are connected to local area network making it a little farther from the sensors and actuators.

In general the three prementioned paradigms (Figure 1) are powerful enough to handle IoT application needs. However, the processing performance is impacted by different factors such as the distance between the processing unit and the data resource, the size of the data, the mobility and the security measures.

In this paper we adopt a fog-centric architecture, suitable for the use of semantic interoperability mecanisms. This architecture takes into account the scalability, mobility and security needs by leveraging the edge and the cloud computing. Besides, we propose an IoT spatial indexing approach which describes both an indexing data structure (registering operation) and the query processing operations (discovery operation). The indexing structure is constructed by encoding the loT resource locations in geohashes, and then using the Prefix Hash Tree [4] (PHT) structure to index and store them. The indexing method indexes the semantic description of IoT objects processed at the fog level. The remainder of this paper is organized as follows:
Section 2 provides an overview of the problem. Section 3 presents our fog based architecture. Section 4 present our indexing method. Section 5 discusses the obtained results. And Section 6 concludes the paper.

2. PROBLEM AND REQUIREMENTS

2.1 IoT Issues

Different IoT challenges [5] can be analyzed based on three characteristics of the IoT: heterogeneity, resource constraint and dynamic environment. The following issues can be formulated:

- The scalability which is a the quality of a system that allows it to meet current and future needs by automatically adapting to changes surrounding it.
- The interoperability which is the quality of a system of heterogeneous elements, allowing it to communicate comprehensible data between its components. For an IoT system, interoperability allows devices to collaborate automatically and transparently, regardless of their manufacturer.
- The mobility which is the dynamic nature of the IoT devices and gateways as well as the frequent change of the IoT data.
- The security issues which can result in damage and threat to human life and property. The authors of [6] name some of them such as confidentiality, integrity, availability etc. [7] highlight four security issues which are: information confidentiality, security of physical devices, information confidentiality and network security. The lack of a security vision that addresses all of these issues, given the number of networks, devices, data, and end users in the future IoT, will be a big issue in the face of IoT scalability.

Besides, four SWoT specific challenges can be captured:

- SWoT Modeling: In addition to the devices description, ontologies should model other characteristics of the IoT. To name few, the data, the services, the quality of these services, the mobility etc. Particularly, three categories of ontologies can be taken into account: device ontologies, service ontologies and data ontologies.
- Semantic Annotation: Even in the presence of a good SWoT model, the manual annotation of WoT with such model remains inefficient. For instance, the Sense2Web [8] gives the opportunity to manually annotate sensor data leveraging the SSN and Dbpedia ontologies. Practically, the semantic annotation should be automatic because of the huge number of the WoT data to be annotated.
- Semantic Discovery: The semantic discovery [9] attempts to use semantics to meet user requirements against offered IoT services. For this end, efficient search techniques should be used to discover and compose WoT resources as desired by the client applications. [10] has studied and elaborated a taxonomy of the state-of-the-art of search methods for the Web of Things.
- Services Composition: Service composition is a very beneficial tool used to generate new services with new features that an individual service cannot present. Since multiple IoT web services will be available, the purpose of semantic description is to enable automated composition of content. However, this is a difficult task due to the heterogeneous nature of the services and the required processing resources.

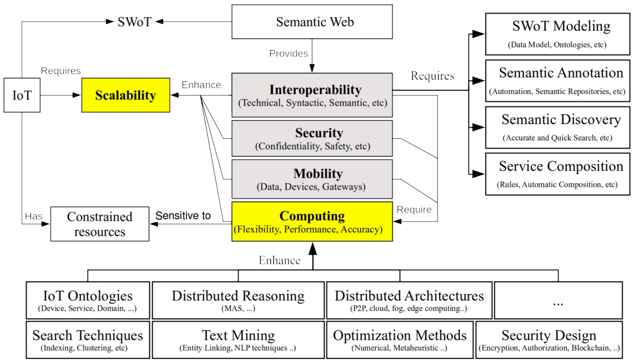
SWoT
Semantic Web
Provides
Requires
IoT
Scalability
Enhance
Interoperability
(Technical, Syntactic, Semantic, etc)
Security
(Confidentiality, Safety, etc)
Mobility
(Data, Devices, Gateways)
Requires
SWoT Modeling
(Data Model, Ontologies, etc)
Semantic Annotation
(Automation, Semantic Repositories, etc)
Semantic Discovery
(Accurate and Quick Search, etc)
Service Composition
(Rules, Automatic Composition, etc)
Has
Constrained resources
Sensitive to
Computing
(Flexibility, Performance, Accuracy)
Require
Enhance
IoT Ontologies
(Device, Service, Domain, ...)
Distributed Reasoning
(MAS, ...)
Distributed Architectures
(P2P, cloud, fog, edge computing..)
...
Search Techniques
(Indexing, Clustering, etc)
Text Mining
(Entity Linking, NLP techniques ..)
Optimization Methods
(Numerical, Metaheuristic ..)
Security Design
(Encryption, Authorization, Blockchain, ..)

Figure 2: IoT Challenges & Solutions

It is worth to note that the scalability can be considered as the principal IoT challenge. Besides, the problem of computation optimization constitutes the link connecting the scalability to the other challenges. In other words, finding new solutions to optimize processing and data storage in the IoT will open the door to solutions that optimize security, mobility and interoperability and consequently the scalability of IoT.

2.2 Requirements

In the following, we present a summary of some solutions that can help to address the computation optimization required by the interoperability, mobility and security issues. Moreover figure 2 summarized the discussed challenges and solutions.

- **Distributed Architectures and Reasoning** [11][12]: Semantic data are sensitive to resource constraints. Therefore, the convenient partitioning of the semantic data in distributed manner may increase considerably the storage and the search performance. In addition, providing efficient indexing, clustering, ranking and discovery mechanisms are important solutions. Besides, using known P2P architecture for IoT systems enhance the IoT scalability in term of mobility [13]. Effectively, P2P networks are fault tolerant and more suitable for mobile nodes. In addition, they are federated which means that only suitable nodes will be requested during the semantic discovery. Besides, mobile nodes can be used to visit lower power devices, collect and send their data to static stations which have higher computation and storage capacity.

- **Security Measures**: When it comes to security, unauthorized access and data integrity remain more critical issues than physical damage of IoT devices. Therefore, authentication, authorization and encryption mechanisms are still required. To design such solutions, the IoT must have significant resources for processing. In this sense, emerging technologies such as the edge computing which migrates huge compute and storage resources to the network edge, which forms an edge layer close to IoT terminals, can be leveraged [14]. This new paradigm alleviates resource constraints, optimizes system performance and offers a new place to deploy new security solutions, such as blockchain which stores and transmits the resulting data stream in a secure, transparent, distributed, auditable and effective way [15].

- **Optimization methods**: There are many optimization methods, inspired from other disciplines, that can improve the processing of IoT data. For example, the Entity Linking (EL) process, is part of the natural language processing techniques. The EL describes a method to automate semantic annotation. In addition, mathematical approaches e.g statistics, probabilistic, numerical and meta-heuristic give the opportunity to address delicate optimization issues such as NP complex problems.

3. PROPOSED ARCHITECTURE

3.1 Computation Requirements

Processing optimization is an important lever for the scalability of IoT, and is in close interaction with the other challenges.

- **Interoperability Requirements**

The interoperability challenge presents four main challenges (SWoT modeling, semantic annotation, semantic discovery and service composition). While the SWoT modeling requires only standardization, modeling and mapping efforts, the other SWoT challenges require computation optimization to enhance the flexibility, the performance and the accuracy of the automated tasks such as the annotation, the search or the service composition. The interoperability mechanisms require Cloud or Fog environment.

- **Mobility Requirements**

The device, gateway and data mobility generate a huge number of costly frequent computations in terms of update operations etc. These operation processing impact the performance, the accuracy and consequently the scalability of the IoT system. This dynamic aspect of the IoT should be treated near to the data source: at the fog or edge layer.

- **Security Requirements**

According to [14], resource constraints and insufficient security design are the major causes of many IoT security problems. Effectively, many of the existing security solutions, require the device to have a high level of computation power and memory space to run them. While the cloud offers a high level of security to the user data, the fact that these data are externalized is still rising privacy and security issues in many IoT applications.

Table 1 shows the distribution of the prementioned challenges in terms of computing and storage needs. Accordingly, the Fog Computing seems to be the central pillar of the architecture.

Table 1: Computing possibilities by Challenges

	Cloud	**Fog**	**Edge**
Scalability	C	F	E
Semantic Annotation		F	
Semantic Discovery	C	F	
Semantic Integration	C	F	
Data Mobility		F	E
Devices Mobility		F	E
Gateway Mobility		F	
Security	C	F	E

3.2 Architecture Design

In this section, we present a fog-centric computing architecture for IoT applications as illustrated in Figure 3.

The fog-centric IoT architecture contains five major parties, the cloud, the edge, the end devices, the fog, and the users.

- **End User:** The end user (human or application) queries the system in order to get IoT services. According to our architecture this can be done through application interfaces at the cloud, fog or edge layers.
- **Cloud Layer:** This level contains discovery and indexing servers. The role of this layer is, on one side, to locate the fog servers required by the user query (this can be done leveraging a geo-spatial index). On the other hand, it redirects the query to the found servers, gathers and ranks the results of the this query according to the user preferences.
- **Fog Layer:** This layer contains the semantic description of the IoT services distributed in different servers or gateways, and is responsible for processing the discovery requests for its geographical scope. The main functions of this layer are: storing, managing andprocessing IoT services using mechanisms such as the semi-automatic semantic annotation to process semantic descriptions of the IoT services, and the semantic-based clustering to dynamically manage the services in order to speed up their discovery.
- **Edge Layer:** To cope with the various networking protocols and processing limitations of connected objects, we are often resorting to IoT gateways (home router, raspberry Pi , smart phones …). In general, the devices in this layer are not as powerful as those in the fog. However, these devices are nearest to data source. Consequently, they are responsible of sending useful and real time data to the higher layers. Further, most of the edge devices are mobile and should inform the other layers about their novel locations. For this reason, Fog and Edge gateways are interconnected through a P2P network. The mobile gateways are indexed using the PHT indexing method described in section 4.
- **End Devices:** Sensors and actuators which are in interaction with the phyical world. These devices are mostly connected to the edge gateways.

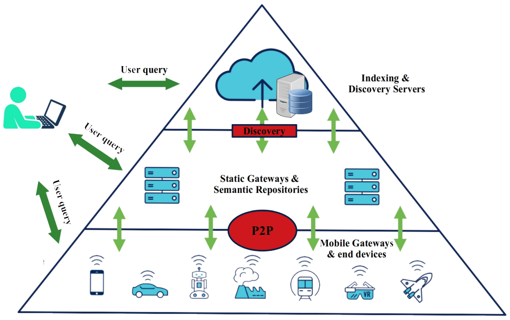

Figure 3: Architecture Overview

4. INDEXING THE FOG

Our architecture leverages two geo-spatial indexes. The first index, allows the discovery of the static gateway, and is implemented in the cloud. Where the second allows the discovery of the mobile gateways through the static ones, and is distributed on the fog and edge servers. In what follows we present how the mobile gateways can be discovered using the PHT index.

4.1 Prefix-Hash-Tree

The Prefix hash Tree (PHT) is a binary trie indexing data structure over Distributed-Hash-Table (DHT) based P2P networks (Figure 4). Keys of objects to be indexed are within the domain $\{0,1\}^D$, where D is the length of the string. The left branch of a node is labeled 0 and the right branch is labeled 1. Each node n of the trie is identified with a chain of P bits produced by the concatenation of the labels of all branches in the path from the root to n. PHT builds a prefix tree in which objects are stored at leaf nodes. Hence, an object with key K is stored at a leaf node with a label that is a prefix of K. The trie is completely distributed among the peers in the network. This is achieved by hashing the prefix labels of the PHT nodes over the underlying DHT identifier space. As a consequence, each node of the trie will have an assigned node in the DHT. The following properties are invariant in a PHT:

- Universal prefix: Each node has either 0 or 2 children.
- Key storage: A key K is stored at a leaf node whose label is a prefix of K.
- Split: Each leaf node stores atmost B keys.
- Merge: Each internal node contains atleast (B + 1) keys in its sub-tree.
- Threaded leaves: each leaf node maintains a pointer to the leaf nodes on its immediate left and immediate right respectively.

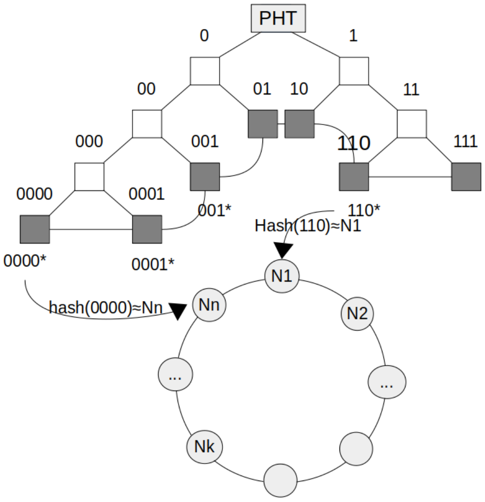

Figure 4: The Prefix Hash Tree structure.

4.2 Geohash technique

Table 2: Near geohashes are stored in the same leaf node

Geohashes	**LEAF NODE PREFIX**	**PHT Keys**
9rhyu	010011011	0100110111100001111111010
9rhyt	010011011	0100110111100001111011010
9rhzt	010011011	0100110111100001111110101

Similarly, geohash [16] is a geocoding system invented by Gustavo Niemeyer, which encodes a geographic location into a short string of letters and digits. It is a hierarchical spatial data structure which subdivides space into buckets of grid shape, which is one of the many applications of the linearization method known as a Z-order curve.

Our approach is a distributed PHT and Geohashes based architecture. The use of PHT is explained by the necessity to be able to make range queries which is not allowed by the distributed hash tables (DHTs). The DHT will allow, in addition to that, a self management and a nodes failure tolerance. As far as the geohashes are concerned, they offer properties like arbitrary precision, similar prefixes for nearby positions etc.

4.3 How does the index work ?

At the fog level, each gateway is considered as a spatial object that is characterized by two location coordinates (latitude, longitude). At the edge level, the gateways can be static at fixed place or mobile (for example mobile phones or gateways attached to mobile objects like cars). Our indexing method indexes only the mobile gateways. Particularly, we encode the geographical locations (latitude, longitude) of the gateways into their corresponding binary representations. These later save the same property of the string representation of the geohashes such that the closer two geographical locations are, the longer their common geohash prefix binary representation is (Table 2). After that, the obtained geohashes are stored in the PHT data structure.

The fact that each PHT leaf node stores the data that has the same prefix as the PHT node label makes that nearby gateways locations (geohashes) are stored in the same leaf node. When performing a search query for a certain location, the PHT prefix of this location is calculated, if a PHT leaf node corresponds to the calculated location then all the data stored in this node are returned. The following step-by-step description summarizes the registration and the discovery processes through our indexing method:

A. Registeration

- Heterogeneous IoT devices communicate through different communication protocols (ZigBee, Zwave, Bluetooth, Wi-Fi or Ethernet) to an edge gateway.
- The gateway publishes the communicated IoT data in form of services (e.g JSON format) leveraging different messaging protocols (CoAP, MQTT, XMPP or HTTP) .
- Each gateway of the edge has location coordinates (latitude, longitude). And can be either static at a fixed place (Home gateway) or mobile from a place to another (mobile phone).
- The location coordinates of gateways are converted to their corresponding geohashes. These later are converted to their corresponding binary representations (same properties of the string geohash).
- The binary representation (e.g 30 bits) is concatenated to a binary security code (e.g 20 bits) to form the PHT key.
- The data to be archived is the semantic description of the IoT devices.
- While the PHT structure stores the PHT keys of the mobile gateways. The data corresponding to each PHT key is distributed over the DHT servers.

B. Discovery

- The queries supported by the PHT structure are exact match, range and proximity queries.
- A location coordinates (latitude, longitude) or a geohash is given in the query. (Mainly the location of a static gateway)
- The distributed indexing identifies gateways concerned by the query based on the location coordinates.
- The unique names of the nearby mobile gateways are returned and can be leveraged to obtain the semantic files or addresses of the semantic repositories.

Our approach makes it possible to improve search speed in range and nearby queries. Given two keys L and H (L ≤ H), a range query returns all keys K contained in the PHT satisfying L ≤ K ≤ H, while a nearby query algorithm returns all keys near to a given key K. Following our approach, for both range or nearby queries, we have just to find the leaf nodes corresponding to the query and then return all the keys stored in the leaf nodes. Effectively, all the keys stored in the same leaf node are closer to each other with a precision equal to the prefix of the leaf node. It's worth to note that the PHT structure uses for nearby and range queries two failover search solutions (binary/linear) and (sequential/parallel) for some reasons detailed here [4].

5. EVALUATION AND RESULTS

We performed a set of experimental activities to simulate our approach using a PHT implementation for the PeerSim simulator [17][18]. We considered the following scenario: A geographical area (e.g a state) containing 200 smart spaces, each smart space can hold 500 mobile gateways (edge layer). To evaluate our approach we have set two essential criteria : the first is the precision of the search and the second is the impact of our approach on the search speed. To do this, we have simulated the insertion of a number of randomly generated objects (mobile gateways) in form of key value tuples whose key is the binary encoding of a geohash. After the insertion of 10,000, 20,000, 30,000 up to 100,000 objects. We executed a set of queries to search for objects that are geographically close to a certain randomly generated object. We recorded for each insertion the precision of the search. This accuracy is quantified by the number of bits in the prefix in the PHT in which the searched objects were found. We noticed that the accuracy of the search increases proportionally to the number of the recorded objects, see Figure (Figure 5).

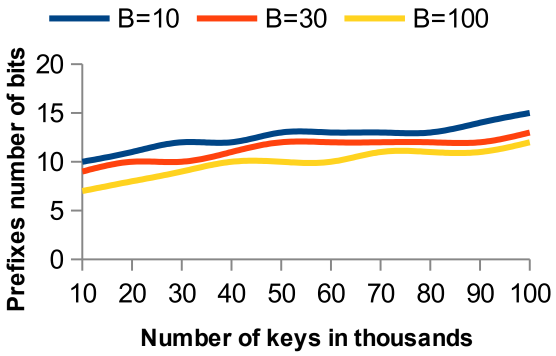

Figure 5: The search precision (in number of prefix bits) in function of the number of PHT keys.

6. CONCLUSION

IoT is highly dynamic , huge and presents a distributed nature. Many approaches have been proposed to enhance the semantic discovery of the IoT services. However, few of them take into consideration the precited critera. In this paper we have proposed an efficient fog-centric architecture enforced by a geospatial indexing based on DHTs and geohashes. The results of the simulations have shown that our approach improves both accuracy and the discovery speed. Our future work is to show the smart aspect (knowledge, reasoning, cooperation between components) in our system by designing and implementing a multi-agents IoT framework for a smart space.

REFERENCES

1. L. Atzori, A. Iera, and G. Morabito, **The Internet of Things: A survey**, *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.
2. P. Hu, S. Dhelim, H. Ning, and T. Qiu, **Survey on fog computing: architecture, key technologies, applications and open issues**, *Journal of Network and Computer Applications*, vol. 98, pp. 27–42, Nov. 2017.
3. Y. Ai, M. Peng, and K. Zhang, **Edge computing technologies for Internet of Things: a primer**, *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, Apr. 2018.
4. Sriram Ramabhadran, Sylvia Ratnasamy, M. Hellerstein Joseph and Scott Shenker, **Prefix Hash Tree: An Indexing Data Structure over Distributed Hash Tables**, *PODC 2003 Proc. of the 23rd ACM Symposium on Principles of Distributed Computing*, pp. 368, 2004.
5. I. Nadim, Y. El ghayam, and A. Sadiq, **Towards a semantic web of things framework**, *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 4, p. 443, Dec. 2019.
6. S. Mishra, S. Jain, C. Rai, N. Gandhi, **Security challenges in semantic web of things**, in: *International Conference on Innovations in Bio-Inspired Computing and Applications, Springer*, pp. 162–169. 2018.
7. R. Mehta, J. Sahni, and K. Khanna, **Internet of Things: Vision, Applications and Challenges**, *Procedia Computer Science*, vol. 132, pp. 1263–1269, 2018.
8. P. Barnaghi, M. Presser, **Publishing linked sensor data**. In: *Taylor K, Ayyagari A, Roure DD (eds) Proceedings of the 3rd International Conference on Semantic Sensor Networks (SSN'10)*, vol 668. CEUR-WS.org, Aachen, Germany, pp 1–16, 2010.
9. I. Nadim, Y. Elghayam, and A. Sadiq, **Semantic discovery architecture for dynamic environments of Web of Things**, *2018 International Conference on Advanced Communication Technologies and Networking (CommNet)*, Apr. 2018.
10. Y. Zhou, S. De, W. Wang, and K. Moessner, **Search Techniques for the Web of Things: A Taxonomy and Survey,***Sensors*, vol. 16, no. 5, p. 600, Apr. 2016.
11. S. B. J, **Enhancing Performance of IoT Networks through High Performance Computing**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 3, pp. 432–442, Jun. 2019.
12. R. N. S., **Optimal Reactive Power Control Using Compensating Capacitor Based on Artificial Immune System**, *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 8, no. 1.3, pp. 381–386, Aug. 2019.
13. I. Nadim, Y. El Ghayam, and A. Sadiq, **Mobility of Web of Things: A Distributed Semantic Discovery Architecture**, *Big Data, Cloud and Applications*, pp. 249–260, 2018.
14. K. Sha, T. A. Yang, W. Wei, and S. Davari, **A survey of edge computing-based designs for IoT security**, *Digital Communications and Networks*, vol. 6, no. 2, pp. 195–202, May 2020.
15. A. Rhayem, M. B. A. Mhiri, and F. Gargouri, **Semantic Web Technologies for the Internet of Things:Systematic Literature Review**, *Internet of Things*, vol. 11, p. 100206, Sep. 2020.
16. Available online: http://geohash.org/
17. https://github.com/nongbottom/Peersim-Pht
18. A. Montresor and M. Jelasity. Peersim: **A scalable p2p simulator**. *In Proc. of P2P'09*, pages 99–100. IEEE, 2009.