SecQSON: Secure Query Scheduling and Ontology-Based Searching in Map-Evaluate-Reduce-Enabled Grid Environment

N. Nalini and G. M. Kiran

Abstract Task scheduling and resource allocation are the major issues in grid environment. Based on grid user's requirements such as deadline, cost, and service type, tasks must be scheduled and appropriate resources are allocated for each user task. Previous works in this topic is failed to analyze all criteria for timely scheduling and resource allocation. Further, scalability and storage issues are other drawbacks in grid computing. Grid over Hadoop is a great solution for solving scalability and storage issues. When adding security to the system, we can address the traditional issues of grid computing high response and retrieval time, resource searching time, low scalability, and storage issues. In this paper, we proposed a new framework called as SecQSON which is a secure query scheduling and ontology-based searching in Map-Evaluate-Reduce model. There are three processes which are applied in this paper as authentication, scheduling, and data retrieval phases. In authentication phase, data owners (DO) and data users (DU) are authenticated by trusted authority (TA). For this purpose, Dual Bio-Key-based Random Authentication (DUBK-RA) algorithm is proposed. The security credentials are fingerprint, finger vein, ID, and password for authentication. For bio-key generation, BLAKE-3 hashing algorithm is generated in TA and this key is verified to validate whether the request is authorized or not. Then scheduling phase processes the authorized requests (DU's) for query scheduling. In this task, Map-Evaluation-Reduce model is proposed that maps the users to optimum grid resources. The evaluation of the resources for user queries is evaluated using Spotted Hyena Optimizer algorithm. For evaluation purpose, various criteria are considered such as trust level, resource score (available bandwidth and queries), and time score (response time and execution time). Final phase is a data retrieval phase in which authorized DO's records are stored in the grid-connected Hadoop server. In grid server, name node is processed and it constructs the index values for DO's

N. Nalini (✉)
Department of CSE, NITTE Meenakshi Institute of Technology, Yelahanka, Bengaluru, Karnataka 560064, India
e-mail: nalini.n@nmit.ac.in

G. M. Kiran
Department of CSE, Shridevi Institute of Engineering and Technology, Tumkuru, Karnataka 572106, India

N. R. Shetty et al. (eds.), *Advances in Computing and Information*, Lecture Notes in Electrical Engineering 1104, https://doi.org/10.1007/978-981-99-7622-5_33

records by means of Dendrimer Order Statistic (Den-OS) index. Further, ontology is constructed for records stored in data nodes. Grid resources for user queries are dynamically searched, and the optimum search results are retrieved for grid users. Experiments are conducted and the performance is evaluated using several metrics such as response time, search accuracy, retrieval time, authentication time, latency, energy consumption, precision, recall, and f-measure.

Keywords Grid environment · Hadoop · Map-Evaluation-Reduce · Secure query scheduling · Resource selection · Index construction

1 Introduction

Grid computing which is the distributed computing paradigm is an important evolving architecture. In grid computing, security is the major challenging issue to be concentrated [1]. In general, unauthorized user access increases the resource consumption, which is not efficient in grid environment. In order to improve security level, multi-factor authentication scheme is proposed [2]. In multi-factor authentication, ID, password, and smartcard are considered. However, it is vulnerable to smartcard loss attack. In addition, a two-factor authentication scheme uses password and smartcard [3]. To ensure security, XOR operations and one-way hash function are involved. The smartcard plays a pivotal role in anonymous authentication scheme also [4]. The smartcard-based authentication is effective to server forgery attack. However, smartcard is likely to be tampered which affects the security level in the grid computing. Trust-based scheduling is presented with the optimization technique in grid computing [5]. To enable scheduling process, ant colony optimization (ACO) algorithm is presented with Firefly Algorithm (FFA). In this work, the Min-Min algorithm is used for pheromone initialization. In general, ACO algorithm has slow convergence rate which results in non-optimal scheduling.

The ACO algorithm is also used with cuckoo search algorithm to enable optimal scheduling [6]. The cuckoo search algorithm is used to form clusters of resources based on the load value and ACO algorithm performs scheduling process. However, execution of cuckoo search and ACO increases time consumption since both algorithms have higher time consumption. Besides other techniques, MapReduce which is the big data processing procedure is utilized in grid computing [7]. Integration of MapReduce and grid computing results in minimized time consumption. A fuzzy-based security scheme is proposed for grid environment [8]. The fuzzy algorithm computes the trust value of resources based on trust value and security level required. However, this work is unable to validate the users since it only validates the resources. To enable search over the distributed environment (i.e.) grid environment, a Boolean search is enabled [9]. Here, the users are authenticated and authorized by the data owners. Then, the user queries are searched by Boolean search mechanism. However, the Boolean search is ineffective and not suitable for secure search. In addition, authentication by data owner limits the security level. Thus, secure and semantic

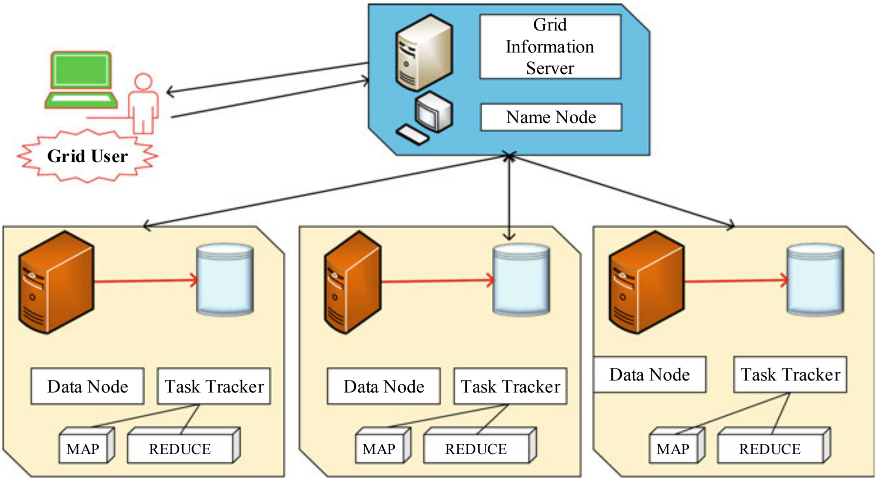


Fig. 1 Grid over hadoop architecture

search over grid environment is still challenging issue. Further, grid servers use big data analytics framework for processing queries/tasks. One of the emerging big data frameworks is Hadoop that provides huge amounts of data processing among data nodes through name node. Figure 1 depicts the grid over Hadoop architecture.

1.1 Research Motivation and Contributions

In grid-based big data processing, query search and data retrieval have been applied in many research works. However, searching unauthorized queries over distributed big data increases higher time consumption and also degrades the overall system performance. Besides, scheduling authorized user queries lacks with poor algorithm design and higher time consumption. The main research issues determined in this paper are as follows:

1. Authentication schemes often use ineffectual authentication credentials.
2. Fast and efficient scheduling is necessary since it has to handle large-scale users.
3. Lack of semantic representation in keyword search decreases the search accuracy.

In searching contents to the semantic database, there are several problems as (1) user query is too ambiguous, (2) keywords in the query are not match with the repository, (3) the retrieved results are not properly ranked, and (4) lack of knowledge in sematic meaning of keywords. To address the issues of previous works, this paper considers the following objectives.

- To prevent resources from unauthorized users by eliminating unauthorized users in the grid environment.

- To support multi-query scheduling such that retrieval and response times are minimized in the grid environment.
- To enable secure search and semantic search over multi-server grid environment.

Contributions: The main contributions of this work are follows:

- We present a novel Dual Bio-Key-based Authentication (DuBK-RA) algorithm that works upon dual biometrics such as finger vein and fingerprint which make the system as secure. We present BLAKE-3 algorithm for authentication to ensure high-level security with minimum computational time.
- A novel Map-Evaluate-Reduce-based trusted scheduling algorithm which is fast and efficient is proposed. We present SHO algorithm to evaluate the fitness value and to schedule the queries to the resources in Map-Reduce model. It has good convergence and also has lower time consumption.
- We present novel ontology-based search with Den-Order Statistics indexing (Dendrimer) structuring. It improves the search efficiency and also minimizes searching time. And also semantic ontology is constructed for improving the searching and retrieving time.

1.2 Paper Organization

The remainder of this paper is described as follows: Sect. 2 presents the previous works in secure query scheduling, resource allocation, and data retrieval in grid computing. The major problems and open challenges of grid computing are described in Sect. 3. Section 4 shows the proposed work in detail which presents the algorithm procedure and explanation. Section 5 gives the experiments description and the results for the performance are depicted in terms of graphs and tables. Finally, Sect. 6 concludes the paper and summarizes the future directions.

2 Related Work

In this section, existing works of secure query processing and scheduling are presented in grid environment. There are two classifications of works presented in this section in terms of scheduling and resource selection and secure service provisioning and ontology construction for grid users. We studied these works and presented the deficiencies and limitations.

2.1 Scheduling and Resource Allocation

A trust-based resource selection by optimization technique is presented in [10] under grid computing. In grid computing, scheduling is an important process. Here, the main aim is to minimize the make span in grid environment. For trust-based scheduling, ant colony optimization (ACO) algorithm is presented. In ACO, the pheromones' update is performed by Min-Min algorithm. The results are then initialized as fireflies in firefly algorithm (FFO). Then, the FFA algorithm obtains the optimal scheduling. Trust value and make pan parameters are considered as objective functions. In general, ACO algorithm has slow convergence rate which is ineffectual in scheduling. In [11], authors present load-based scheduling in grid computing environment. The aim of this work is to balance the load among grid resources. At first, the cuckoo search algorithm forms clusters of resources. Then, optimal scheduling is performed by ant colony optimization (ACO) algorithm. The cuckoo search algorithm uses load as the metric to form clusters. The ACO algorithm considers make span, miss ratio, and throughput for performing optimal scheduling. Both ACO and cuckoo search have slow convergence rate which increases time consumption for scheduling. An Integrated Grid and Spatially Indexed MapReduce (IGSIM) is presented to enable fast search [12]. For fast search, R-tree and R*-tree spatial indexes are constructed in the MapReduce environment. Based on them, Hilbert TGS R-Tree index is constructed to enable parallel processing. This paper highlights that the involvement of MapReduce boosts up the parallel processing and minimize the time consumption. Though time consumption is minimized, the search accuracy is low due to the absence of semantic ability to search.

In [13], authors aimed to address the high waiting time and inaccurate resource allocation issues for user's submitted jobs. Initially, users submit their job with the predefined limit and then scheduling policy, i.e., first come first service (FCFS) is used to schedule the jobs. FCFS is worked by left, right, and hole basis. Based on the time limit, user's jobs are scheduled to the distributed grid environment. Real-time tasks' waiting time is higher, and also, shorter jobs only are frequently executed and longer jobs' retransmission rate is very high. These causes high SLA violation and poor QoS. A comprehensive survey in resource allocation and task scheduling is presented in [14]. The objective of the proposed model is to make the task execution by scheduling and resource allocation for sending the scheduled jobs for heterogeneous tasks. Further, total execution cost is reduced in resource selection and reduces the average waiting time in scheduling. Further scheduling and resource allocation parameters must be optimized such as response time, penalty ratio, and time constraint issues.

2.2 Secure Service Provisioning and Ontology Construction

Authors of this paper [15] present a secure architectural model for distributed cloud and grid computing. In general, identity theft, identity management, authentication, data theft, and trust computation are major security concerns. As of now, security assessment markup language (SAML), open authentication, and open identity are presented as security solutions. However, these security solutions are insufficient for assuring security in distributed computing. Thus, this paper presents a secure architecture for distributed computing environment. A multi-factor authentication scheme is analyzed and proposed in [16]. The main of this work is to analyze the multi-factor authentication in multi-server environment. In majority of multi-factor authentication schemes are uses smartcard, ID and passwords. The main pitfalls of the multi-factor authentication scheme are smartcard loss attack and there is no forward secrecy which is achieved. This paper highlights that the smartcard and password alone are insufficient to authenticate the users in multi-server environment.

Authors in [17] revisit the two-factor authentication scheme for cloud users in multi-server environments. The two-factor authentication scheme uses smartcard and password to validate the users. In that, multiple operations such as bitwise XOR operation, string concatenation and one-way hash function. Majorly, the two-factor authentication scheme is constructed upon the strong assumption that smartcard cannot be tampered. In practical, there is high possibility for smartcard tampering. Thus, the authentication scheme is inefficient.

In [18], authors propose an improved anonymous authentication scheme which is proposed to authenticate users in distributed cloud computing. This paper builds upon smartcard-based authentication procedure. Initially, a smartcard generator selects two cyclic groups over addition and multiplication. To embed the secret keys in smartcard, a one-way hash function is used. The authors highlighted that the proposed scheme is efficient for server forgery attack.

In [19], authors aim to improve security of distributed computing environment such as grid computing. In precise, this work designs a trusted grid (T-grid) computational model for secure computations. For security purpose, fuzzy logic is proposed. The fuzzy algorithm finds the security level needed to be employed in the grid computing. The fuzzy algorithm considers final trust value and security level as input and provides secured final trust value as the final output. This final trust value is considered as reputation value for the future purpose. However, trust-based computation alone is not sufficient to assure desired security level in the grid environment since value evaluates the reputation of resources, but the users are not validated in this work.

In [20], secure multi-search methodology is presented. In this work, the data are stored in distributed manner across multiple servers. Here, the data owner encrypts the data with searchable index construction. Then, the data and index are stored in the distributed databases. When a user search is enabled, Boolean search is performed for data retrieval. Here, the data users are authenticated based on non-interactive authorization scheme. In order to improve retrieval time, the user queries are searched in parallel. The drawbacks in this system are as follows:

1. Boolean search is performed only based on the query terms (i.e.) it does not consider semantic representations of the query which degrades the efficiency.
2. The authorization is performed between owners and users which results in unauthorized access since the data owner is not authenticated.

Semantic service retrieval based on the domain specific knowledge is a crucial operation. Automatic query processing and reformulation support for accurate retrieval [21]. A unified search engine is proposed in this paper that integrates both semantic ontology search engine and keyword-based architecture. Fuzzy membership values based ranking results are presented that determines semantic relationship among keywords. The five unique properties of this paper are as follows: Query Words, Ontology Support, Ranking Method, Semantic Score, and Keyword Expansion. Searching query keywords to the single ontology causes higher response time. In some cases, query retransmissions' demand is required in this paper. Hadoop MapReduce-based retrieval framework is proposed in this paper for query processing and retrieval of similar records from Hadoop database. Feedbacks from the user are collected and stored in the database that achieves higher retrieval precision and good user experience. The various types of multimedia data are used for semantic-based ontology construction [22].

Natural language processing (NLP)-aided query processing framework is introduced in ontology-based search engine. For accurate semantic information retrieval, query processing operations are used such as tokenization and part of speech (PoS) tagging [23]. The proposed ontology method is computationally simple that does not suited for heterogeneous tasks.

3 Problem Statement

Running services over distributed computing grid environment supports rapid provisioning of services and increases the availability of resources for incoming jobs. In this case, many factors are taken into account for it and they are low response time, high resource allocation efficiency, low retrieval time, and less energy consumption from grid users and so on. This section summarizes the main problems existed in the current scheduling, resource selection, and security issues in grid environment.

To improve security level, Zero-Knowledge Proof (ZKP)-based authentication and Intrusion Detection System (IDS) are proposed. Here, the IDS is deployed in the mobile agents to detect the malicious activities. The ZKP algorithm intends to defend against man in the middle attack. Thus, the authentication credentials such as ID and password are encrypted by RSA algorithm. Besides, enhanced Diffie Hellman algorithm is used for key exchange process [24]. The major problems in this work are as follows:

- The ZKP algorithm only considers ID and PW that are insufficient to validate the grid users. Also, RSA algorithm takes large time for computation even for lower security level. Thus, the authentication procedure is inefficient.
- ZKP-based authentication algorithm consumes large time and also involves high complexity.
- Involvement of random scheduling is ineffective since it makes large time for user requests. It also leads to schedule the user tasks to untrusted resources.

The Map-Reduce model is optimized with the non-dominated sorting genetic algorithm (NSGA-II) algorithm [25]. This work improves the quality of service (QoS) by optimum scheduling. The objective is formulated as minimization of flow time and maximization of throughput. Initial population is initialized in the mapper phase and optimal scheduling is performed in the reducer phase. This work has several issues as (1). The NSGA-II algorithm has higher time consumption and the complexity is also high. Thus, user scheduling takes large time. Besides, the convergence is restricted by the involvement of sorting process and consideration of limited metrics and lack of security further affects the overall performance of the grid system. Since, this work performs scheduling for all unauthorized user tasks. Query analysis ontology-based cluster (QAOC) architecture for Hadoop big data environment is presented in [26]. The overall architecture includes query manager, scheduler, and data management. The ontology is constructed to form clusters of relevant data in data management process. When the user query is initiated then, the searching is carried over the ontology based binary index. Although ontology is constructed, here it is used to form clusters. Searching is performed over binary search time which increases retrieval time and has limitations in update and delete. Thus, searching process is inefficient. Scheduling by neuro-fuzzy algorithm only considers the query-related metrics. But, the resource-related metrics are also important.

4 Proposed Work

This research work focuses on secure and semantic search over grid environment. For that we present a novel secure query scheduling and ontology-based searching (SecQSON) in the Map-Reduce-enabled grid environment. The overall environment includes Data Owners (Dos), Data Users (DUs), Grid Information System (GIS), Grid Scheduler (GS), Trusted Authority (TA), and Grid Resources. The overwork involves three major phases as, (i) Authentication Phase, (ii) Scheduling Phase, and (iii) Data Retrieval Phase (Fig. 2).

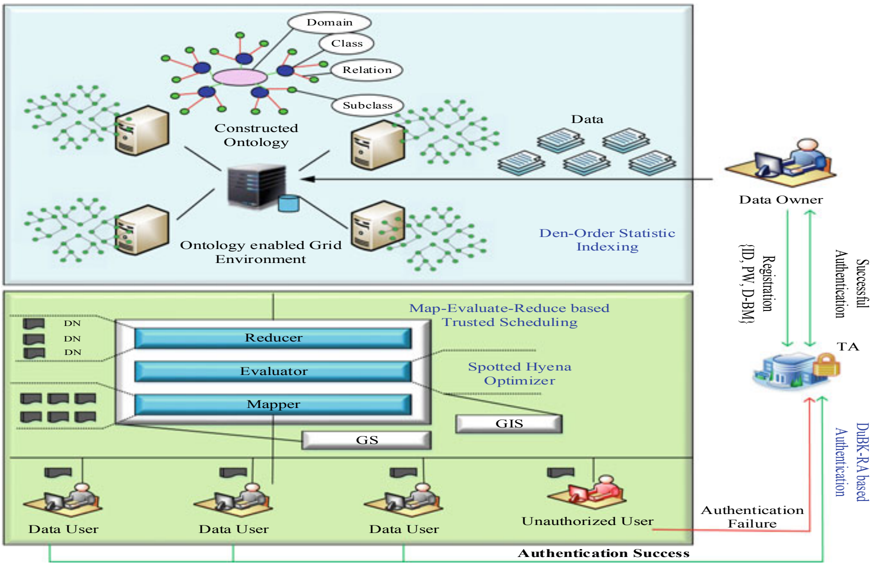


Fig. 2 Proposed system architecture

4.1 Authentication Phase

In first phase, the DOs and DUs are authenticated by TA. For authentication, we propose a novel Dual Bio-Key-based Random Authentication (DuBK-RA) algorithm. Initially, all DUs and DOs are registered their ID, Password (PW), and Dual Biometrics such as Fingerprint (BM1) and Finger Vein (BM2) at TA. These are main security credentials for protecting users and data owners' privacy. Further, we considered multi-contextual attributes for authentication that are follows:

- User / Owner Real-Time Location: GPS coordinates of user and owner is taken and also IP address is considered.

Based on Dual Biometrics, a bio-key is generated for all DOs and DUs. For bio-key generation, we present BLAKE-3 hashing algorithm. In authentication, the bio-key and random bits of biometrics are validated. In this phase, the unauthorized requests are eliminated. BLAKE-3 is a hashing algorithm that uses cryptographic hash tree function that security strength is stronger than SHA-1, BLAKE2, SHA-2, and SHA-3. In general, BLAKE 2 is different from BLAKE-3 version that reduces the number of rounds from ten to seven. Various block_ciphers determined by a Round. It comprised number of building blocks that are composed together to form a cryptographic function that runs in multiple times (Fig. 3).

Firstly, BLAKE-3 splits the input into 1 KB of chunks and arranges them as the binary tree structure. All chunks are compressed in a parallel way. This implies

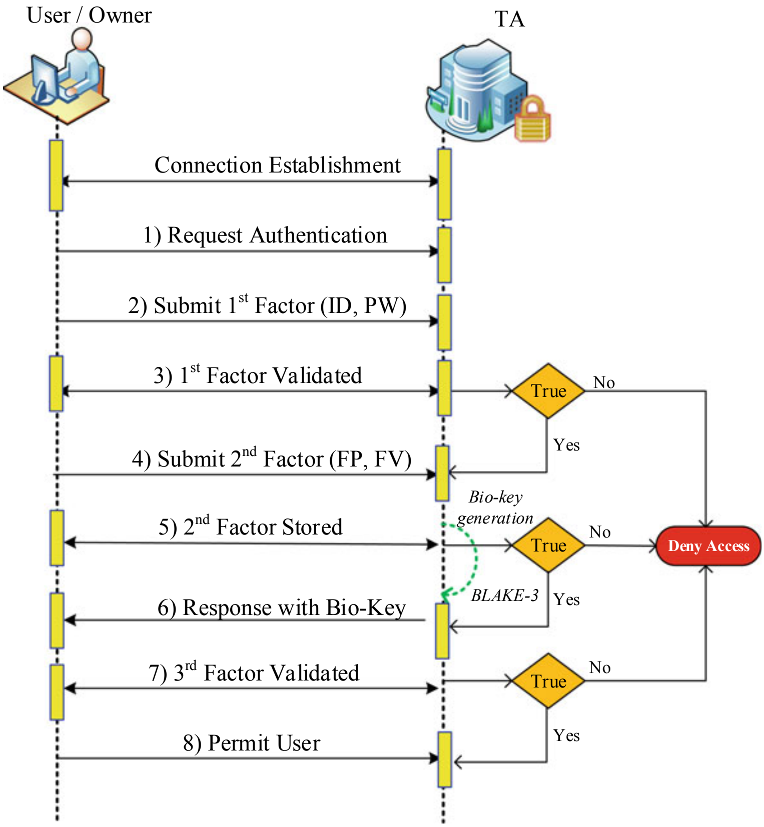


Fig. 3 Authentication of user and owner

that BLAKE-3 leverages 128-bit of security for achieving all privacy goals and also mitigates differentiability or preimage collision attacks.

// Get Input from User (Biometric)

Let convert input matrix into binary values (0's and 1's).

```
// Hash an Input All at Once
Let hash1 = blake3::hash(b “000,111,011,110,111,100,111”);
// Hash an Input Incrementally
Let mut HASHER = blake3::HASHER::New ();
HASHER.update (b “000,011,101”).
HASHER.update (b “1,110,111”).
HASHER.update (b “100,111”).
Let Hash2 = Hasher.Finalze();
assert_eq! (hash 1, hash2);
// Apply Output_Reader for Extending the Output
```

```
Let mut output = [0; 1000];
Let mut output_reader = hasher/finalize_xof();
Output.reader.fill (&mut output);
Assert_eq! (&output[..32], hash1.as_bytes());
// Print a Hash as HEX ());
```

In authentication, TA verifies the quality or trustworthiness of the user is identified for a user which depends on the number of factors considered during registration. Access time and duration are taken into account that determines the user trustworthiness in the system. Finally, user and owner access history is determined that assess the past history of the person related to the location and the type of file access in the grid server. Based on the authentication, data owners and users are classified into the following classes,

- *Legitimate user & owner*—Owner and user who are registered properly to the TA such persons are subjected to the legitimate and those will be affected by attackers. However, attackers use legitimate user data and their parameters.
- *Unauthenticated user & owner*—These persons are the entities which are not registered properly to the TA. These types of attackers purposely enter into the system for crashing the servers with unnecessary authentication request and also it tries to extract data from other servers. This causes high resource wastage for grid servers.
- *Compromised user & owner*—The compromised user and owner are the legitimate the users and owners who are affected by attackers and compromised the legitimate users and owners for make it to participate the system. Hereby, attackers collect information from such compromised attackers.

4.2 Scheduling Phase

The authorized DU requests are scheduled by GS in order to access distributed servers. For query scheduling, we present novel Map-Evaluation-Reduce procedure. In this algorithm, the user queries are mapped to optimal grid resources. To evaluate the resources for user queries, we enable Spotted Hyena Optimizer (SHO) in evaluation phase. The SHO evaluates the resources based on multiple criteria such as trust level, resource score (available bandwidth and queries), and time score (response time and execution time). GIS provides this information for scheduling. In the following, we described the preliminary knowledge about the grid over Hadoop framework with MapReduce paradigm.

Hadoop is a popular open source scalable, reliable distributed computing platform for analyzing large datasets [8]. Apache Hadoop software library is an essential framework that allows for distributed processing of huge datasets across clusters of computers using a simple programming model. The two main components involved in Hadoop system which is used in our research work are listed below,

- Hadoop Distributed File System (HDFS).
- MapReduce.

HDFS [5] is a storage that is enabled to hold on with large amounts of data such as terabytes and petabytes. HDFS provides high aggregated data bandwidth and scales up to hundreds of nodes in single cluster which supports tens of millions of files in single instance. In this system, files are stored in redundant fashion through the presence of multiple machines. HDFS architecture is comprised name node/master nodes and Data nodes/Slave nodes based on the storage and processing requirement of an application. The main feature of HDFS is its "Streaming access" which performs on the motive that the most frequent data processing patterns are write-once and read-many times. In HDFS, the files are divided into multiple segments and they are stored in individual data nodes that are called as blocks. Blocks can also be defined as the minimum amount of data that HDFS can either read/write. 64MB is considered as the default size of blocks; the sizes of these blocks are increased based on the HDFS configuration. HDFS provides high throughput access to application data and is suitable for applications that have large datasets.

(1) Name Node

HDFS consists of single name node, a master server that maintains the file system namespace and regulates access to file by users. Name node server is a significant entity present in HDFS which is comprised files and directories. This reduces the amount of disk space consumed by the log file on the name node, which also reduces the restart time in primary name nodes. The HDFS namespace can be represented as hierarchy of files and directories.

- Manages file system namespace.
- Regulates client's access to files.
- Performs operations like renaming, opening and closing.
- Mapping of blocks to Data Nodes.

(2) Data Node

Data nodes are considered to be the horses of file system, since it stores and retrieves blocks when requested by either client or name nodes. Each and every block of HDFS data is stored in separate file which does not create all files in the same directory. Data node stores two files in each block over its native local file system, which has been configured by HDFS: (i) data itself (ii) block metadata, checksum, and block's generation stamp. The communication of data node is held directly between clients to read and write blocks, by which communication avoids data replication.

- Serves read and write requests.
- Effectively performs block creation, deletion, and replication.
- Block replication is actively maintained.

(3) Features of HDFS

HDFS system performs efficiently due to its key features and the important features involved in this framework are enlisted below,

- Achievement of Maximum throughput.
- Higher fault tolerance.
- Applicable for huge dataset-based applications.
- Suitable for distributed storage and processing.
- Ease to verify the status of clusters with the support of built-in data nodes and name nodes.
- Data reliability, availability, and scalability support.

Map-Reduce is Yet another Resource Negotiator (YARN)-based software programming paradigm for processing of large amount of data in parallel way. The main objective of Map-Reduce is to map dataset into a collection of <key, value> pairs and then reducing overall pairs with similar key. The process of Map-Reduce job includes Map phase and Reduce phase. It solves the problems which existed in previous processes such as fault tolerance, data distribution, load balancing, and task parallelization.

The main advantage of Map-Reduce is that it is easy to scale data processing over multiple computing nodes. Figure 4b represents the Map-Reduce programming model consisting of “n” number of HDFS file storage database. This MapReduce involves two major functions they are,

- Mapping function.
- Reducing function.

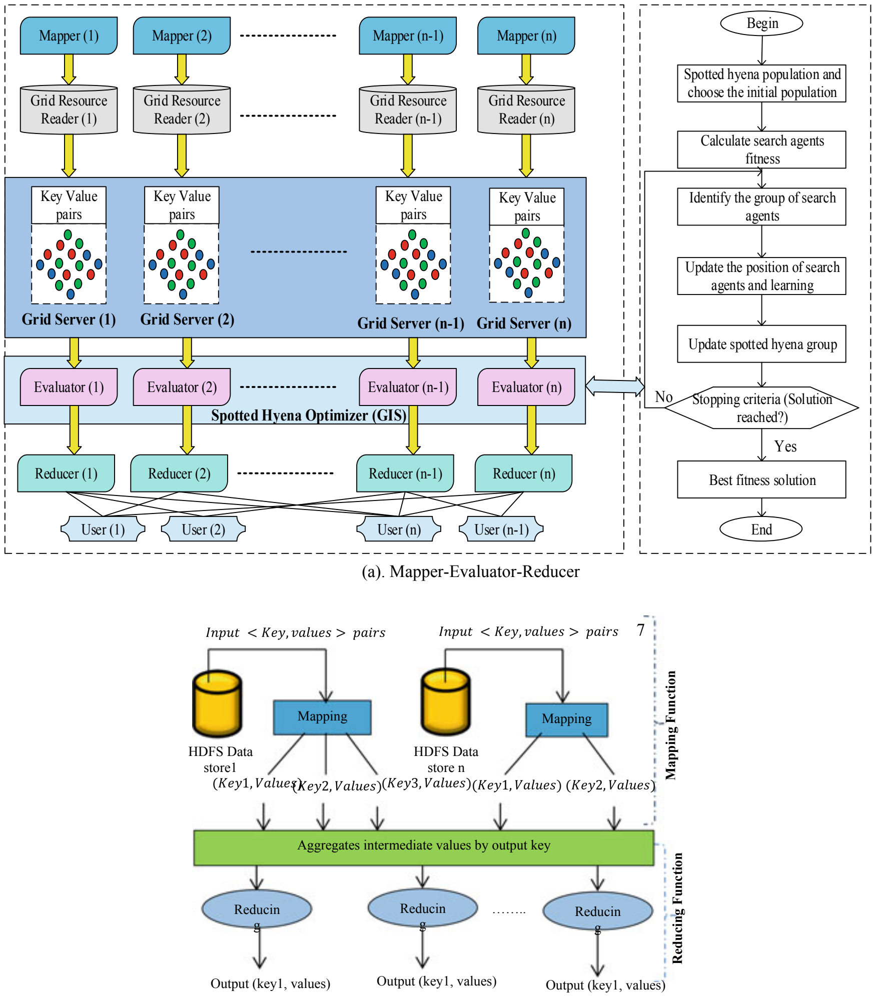
Mapping function initially consider a set of data and it is separated into another format (i.e.) individual elements are considered as tuples (key/pair values). The Map function is applied parallel for every pair in input dataset.

$$Map : (K1, V1) \rightarrow K2, V2. \tag{1}$$

Reducing function is the output from mapping function which is taken as input for Reduce function, which combines the data tuples into smaller set of tuples. The Reduce function is then applied parallel in each group, which produces collection of values in same domain

$$Reduce : (K2, list(V2)) \rightarrow V3. \tag{2}$$

As shown in Fig. 5, the Mapping function processes a <key, value> pair to generate another <key, value> pair. A number of mapping functions run in parallel on the data that is partitioned across cluster, produce a set of intermediate <key, values> pairs. The intermediate key values are stored and grouped by keys. Then, reducing function is applied to merge all the intermediate values that are associated with same intermediate key.



(a). Mapper-Evaluator-Reducer

(b). MapReduce Programming Model

Fig. 4 **a** Mapper-evaluator-reducer. **b** Map reduce programming model

The performance of the MapReduce model is enhanced with the use of optimization algorithm. Since previous works have used genetic algorithm-based optimization that requires higher processing time for evaluating the results of the mapper function. Hence, improving the performance of the MapReduce is very important in this case. Therefore, we proposed a Map-Evaluate-Reduce model to further optimize the performance of the traditional MapReduce functions. In particular, we added the Evaluate function into the Map and Reduce functions. The Evaluate function produces higher performance than MapReduce model. Figure 4a demonstrates the

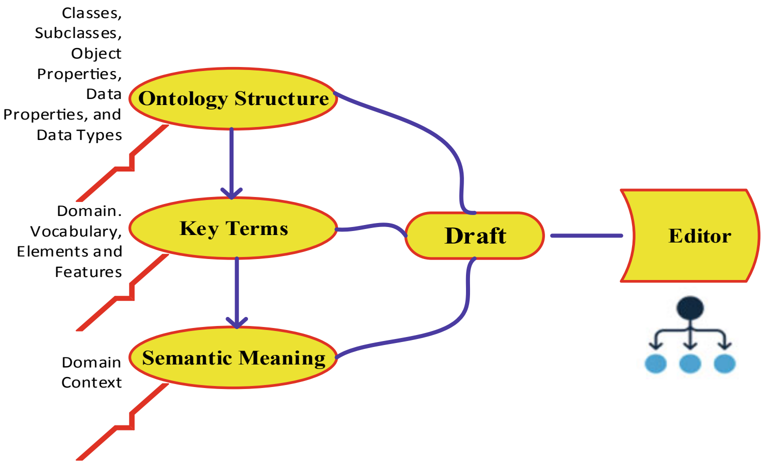

Classes, Subclasses, Object Properties, Data Properties, and Data Types
Ontology Structure
Domain. Vocabulary, Elements and Features
Key Terms
Domain Context
Semantic Meaning
Draft
Editor

Fig. 5 Semantic ontology in grid server

proposed Map-Evaluator-Reduce framework. The processes involved in the proposed Mapper-Evaluator-Reducer framework are discussed briefly as follows:

(a) Mapper

Mapper is the first execution block in the MapReduce which maps the given input user requests present in the other transaction to generate its occurrence value. The mapper acquires input as one of the chunks in the database, i.e., request along with the input parameters as waiting time and arrival time. These chunks are split using the InputSplit operation of the Hadoop.

Pseudocode for Mapper Operation

```
Require: Set of resources $(R_1, \ldots R_n)$,
Grid Server Id's $(GS_{id1}, \ldots GS_{idn})$
Ensure: $< K_n, V_n >$
// Map Operation
For all $R \in [R_1, \ldots R_n]$ do
    If $(R_i \in GS)$
        write $(R_i, 1)$;
    End If
End For
emit $(K_n \rightarrow R, V_n \rightarrow GS(I))$
End
```

Pseudocode Depiction

The pseudocode 1 illustrates the mapper operation in the Map-Evaluate-Reduce framework. Here, the mapper obtains input as a set of requests associated with their service lists required for users. For which, it generates the key and value pair represented as the $< K_n, V_n >$. The mappers assign the key as the input items and resource location as its value. The mapper output is given as input to the optimizer to further reduce the requests moved between the mapper and reducer.

(b) Evaluator

The proposed method introduces the evaluator block between the mapper and reducer. Most of the works have concentrated on the combiner between the mapper and reducer. By contrast, we introduce the evaluator which performs better than the traditional evaluator incorporated between the mapper and reducer. The introduced evaluator enhances the MapReduce framework in the following way:

- It reduces the scanning time of the optimal resources identification.
- It decreases the searching time required to mine the servers from the database.
- It provides the optimal result in finding the services that close to the users request for each request present in the database effectually.

Pseudocode for Evaluator Operation

```
Require: Set of Resources ($I_1, \ldots I_n$), Server ($\mathcal{S}$)
Ensure: $< K_n, V_n >$
// Evaluator function—SHO algorithm
Generate population P = $[P_1, \ldots P_n]$;
For all $P \in [P_1, \ldots P_n]$ do
Compute $\rightarrow$ fitness $f(i)$ for $P_i$
If $(f(i) > f(j))$
$\mathcal{S}(P_i) \leftarrow f(i)$;
End If;
End For
emit $(K_n \rightarrow P, V_n \rightarrow \mathcal{S}(P))$
End
```

The pseudocode 2 illustrates the evaluator operation in the proposed Map-Evaluate-Reduce framework. Here, the key is represented as the resources and the value is defined as the server obtained using the SHO algorithm. The proposed method improves the performance of the Map Reduce framework by exploiting aforesaid operation in evaluator and thus increases the efficacy of the proposed scheduling mechanism for large number of users.

(c) Reducer

The reducer attains input as the evaluator output $< key, value >$ pair. Here, the key represents the requests $(I_1, \ldots I_n)$ and the value represents the $(S_{c1}, \ldots S_{cn})$. The reducer implements the pruning process to reduce the services.

Pseudocode for Reducer Operation

```
Require: Set of Requests ($R_1, \ldots R_n$), Service list for each request $S_c(R)$
Ensure: All requests are scheduled
For all $R \in [R_1, \ldots R_n]$ do
Search Grid Server _name node($R, \mathcal{S}(R)$);
If ($\mathcal{S}(R)$<min.$\mathcal{S}(R)$) then
return;
End If
End For
End
```

4.3 Data Retrieval Phase

The data from authorized DOs is stored in the grids. Then for stored data, Dendrimer-Order Statistic (Den-OS) index is constructed. Besides, we constructed ontology to enable semantic search for user queries. All grid resources search semantically for all user queries. At last, the grid resources return the optimum results for the user queries.

Application Study: The proposed research is applicable are many areas. Some of them are,

- Healthcare Data Storage.
- Big data Analytics.
- Smart City Data Storage.

In particular, Pollution Data Storage and Retrieval are an emerging application in smart city data storage. For ontology construction, three aspects of semantic factors are analyzed such as,

- Ontology structure.
- Key terms in data.
- Context meaning.

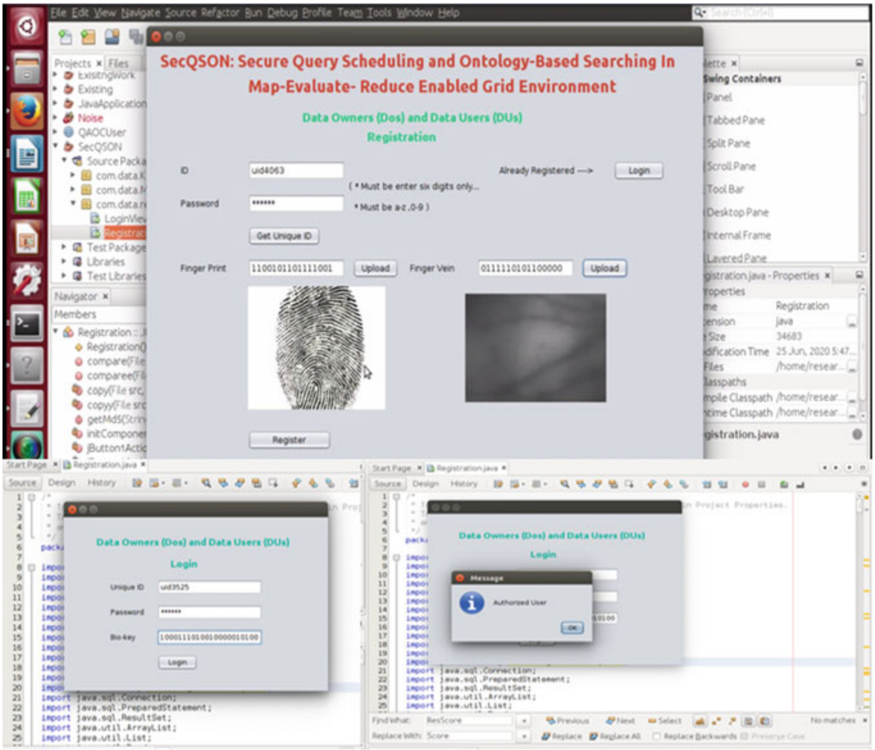
Key terms used in data refer to meaning of terms observed from different terms such as rain, rainfall, and precipitation.

5 Experimental Results

In this section, we discussed the experiments of our proposed work and comparison is made with the previous works such as QAOC and MR-NSGA-II.

5.1 Experiment Setup

The proposed SecQSON model is implemented over grid-assisted Hadoop environment. Java programming language is used to implement this work (Fig. 6). The system configurations required for the implementation of the proposed approach are discussed in Tables 1, 2.



SecQSON: Secure Query Scheduling and Ontology-Based Searching In Map-Evaluate- Reduce Enabled Grid Environment

Data Owners (Dos) and Data Users (DUs)
Registration

ID: uid4063 (* Must be enter six digits only...
Password: ****** * Must be a-z ,0-9)
Already Registered —> Login
Get Unique ID
Finger Print: 1100101101111001 Upload
Finger Vein: 0111110101100000 Upload
Register

Data Owners (Dos) and Data Users (DUs)
Login

Unique ID: uid3525
Password: ******
Bio-key: 1000111010010000010100
Login

Message: Authorized User
OK

```
import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.util.ArrayList;
import java.util.List;
```

Find What: ResScore
Replace With: Score

Fig. 6 Authentication results

Table 1 Software configuration

	Operating system	Ubuntu 14.04 LTS
System outline	IDE	Netbeans 8.0
	Hadoop	2.7.2
	Hard disk	1 TB
	Connectivity	100 Mbps ethernet LAN

Table 2 Hardware configuration

	Development toolkit	JDK 1.8	
Configuration components	Number of users	100	
	Number of nodes	Primary name node	1
		Secondary name node	1
		Number of data nodes	3
	Node configuration	Memory	1.00 GB
		Processor	Pentium (R) Dual-Core CPU E5200 @ 2.50 GHz

5.2 *Comparative Analysis*

In this section, the comparison between the proposed and existing works such as QAOC and MR-NSGA-II. In QAOC, query analysis and ontology construction is implemented and map reduce with genetic algorithm is implemented for scheduling user requests for service assignment. Various performance metrics are considered for evaluation such as response time, search accuracy, retrieval time, authentication time, latency, energy consumption of grid users, precision, recall, and f-measure. The performance results of the proposed work versus previous works are discussed in the following sections.

5.2.1 Response Time

The response time is an essential metric to analyze the performance of the proposed work when processing the services and resources for user query. It is described as the length of the time occupied to complete the user given mining request. It is measured using the below expression,

$$\mathcal{R}_T = T_i\left(\frac{F_{r,q}}{S_r}\right) \tag{3}$$

where T_i represents time interval between first response to the given query $F_{r,q}$ and submission of the response S_r (Fig. 7).

However, the response time increases when workload increases and the service deadline varies little due to this effect. In NSGA-II, processing time is very higher that introduces more overhead in scheduling. Further, QAOC uses default schedulers for user requests. Both result high response times compared to the proposed work. To avoid the higher workload in the system, we presented authentication for eliminate the compromised and unauthenticated devices. As a result of pruning, unauthenticated users' workload of our proposed work gets minimized and also historical authentication history is evaluated frequently by the trusted authority.

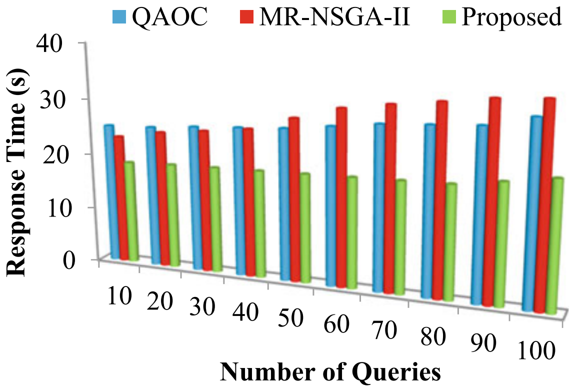


Fig. 7 Number of queries versus response time (s)

5.2.2 Latency

Latency is a time delay that takes during the user requests submitted to the grid over Hadoop environment. However, there are three factors which affect the performance of latency such as request size, type and number of requests forwarded from the particular region. Hence, optimizing the performance of service applications and web processing is essential, but also it is challenging. Hadoop data node stores the input data and processing takes into account in name node and searching process using MapReduce and the output is written to the database (metadata of NameNode). To get the data stored in Hadoop, users request a server via query, which scans the NameNode metadata information. For searching, filters, roll-ups, or drill downs process is applied and returning result. In distributed grid computing environment, data get from the grid servers. Both individual processing of grid servers and Hadoop servers increase the time for resulting a single query. To address this issue, we presented a unified architecture for handling massive requests from the users. Grid scheduler is deployed in Hadoop MapReduce framework for handling requests for users. The evaluation module is additionally included in the MR, which finds the fitness value by SHO algorithm. With the use of fitness of user requests, the optimum resource is determined and services are provisioned for user before the service delay is reached. Figure 8 shows the performance of the latency with respect to the number of queries.

5.2.3 Search Accuracy

In this paper, relevance measure is estimated for each request from users. A relevance measure must be a positive value that ranges between 85 and 95% for proving with the better performance. Figure 9 shows the performance of the search accuracy with respect to the number of queries.

Fig. 8 Number of queries versus latency

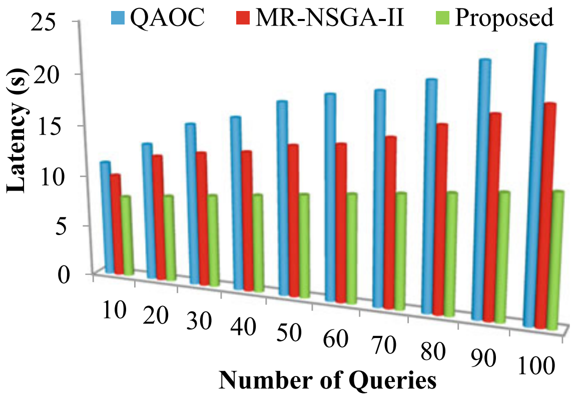
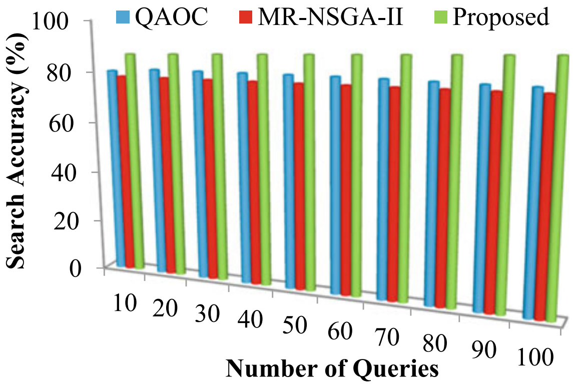


Fig. 9 Number of queries versus search accuracy



Our proposed work is highly accurate which is due to the semantic ontology construction. Ontology is constructed using OWL graphical editor. Further, we constructed ontology based on the geospatial information. In this work, domain knowledge is extracted from the data stored by the data owners. From the records, domain, class, sub-class, and properties are determined and stored in constructing the ontology. Based on the results of the ontology and index construction by the dendrimer index functions, search accuracy is improved.

5.2.4 Retrieval Time

The retrieval time is a significant metric to estimate the query processing and scheduling performance. It is referred as the time required to mine the optimal resources from the database. It must be low as much as possible, to enhance the proficiency of the system. It is expressed in mathematical form as follows:

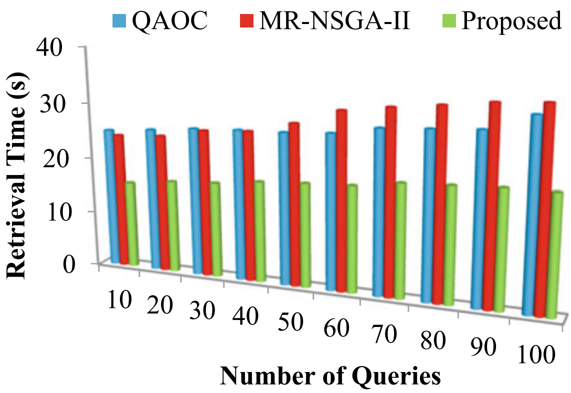


Fig. 10 Number of queries versus retrieval time

$$\mathcal{E}_T = \mathrm{T}_R \frac{\mathrm{FP}}{D} \tag{4}$$

where $\mathrm{T}_R \frac{\mathrm{FP}}{D}$ signifies the time necessary to mine the optimal resources from the Hadoop nodes. The performance of the retrieval time for the proposed and previous works such as QAOC and MR-NSGA-II is compared with the number of users which is depicted in Fig. 10

Retrieval time is varied in the proposed work as well as the previous works due to the time of arrival, location of request, and service type. In this work, multiple parameters are considered for service retrieval. To get the immediate access in query processing, semantic meaning-based ontology and index are constructed, which are used to execute user queries in a faster way. Lack of domain and geo knowledge from grid users cause higher retrieval time for the earlier works.

5.2.5 Authentication Time

As already discussed, time is an important performance evaluation metric while providing security for the system. The time taken for each process is analyzed here. However, the time taken for registration, authentication, and login increases when there is increase in the number of owners and users. Based on the three common processes for security, the time consumption is studied. Here, the authentication of multiple devices and users can be performed at a time.

The proposed system is compared with prior research works that have been incorporated in distributed grid and Hadoop environment for security. The number of security credentials that are used for authentication does not reflect on time for processing. Since, the time for processing depends on the computations performed.

Figure 11 illustrates the comparative chart for authentication time. From this comparison, authentication time is higher due to the participation of devices and users. Authentication in previous work was established based on the validation of

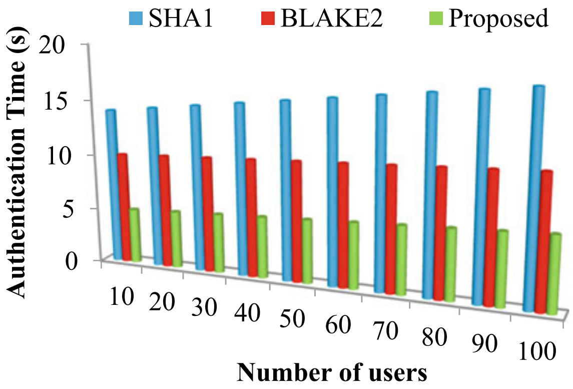


Fig. 11 Number of users versus authentication time

identity, password, and random number. The entities were not authenticated sequentially by submitting each security credential. These three security credentials in this work were fed into smartcard of the particular. By this, remembering of certain security credentials is not required. It is enough to swipe the smart card, and then, the entity will be authenticated and permitted to access.

Most of the previous works have followed to use smartcard as one of the factors. The process handled with smartcard was equipped with a special smartcard revocation procedure by which the lost smartcard can be recovered and old card could be blocked. However, the presence of such smart card for authentication was not advisable in recent days.

5.2.6 Precision

In this paper, we firstly introduced the precision metric in grid environment which is derived from the true positives and false positive values. To show the higher performance of precision, it must be higher as much as possible. Figure 12 indicates the performance of precision with respect to the number of queries.

In this paper, precision is defined by the number of queries correctly processed than the total number of queries generated by the users. In ontology construction process, each query semantic meaning is evaluated and assessed based on the meaning to the database. Thus, we obtained higher precision performance than previous works.

5.2.7 Recall

Recall metric is computed using true positives and false negatives. It is defined by the sum of queries processed correctly in a server to the total number of queries processed in the server. With respect to a given query, index searching is performed which is a notion of relevant and not relevant responses related to the query. Our

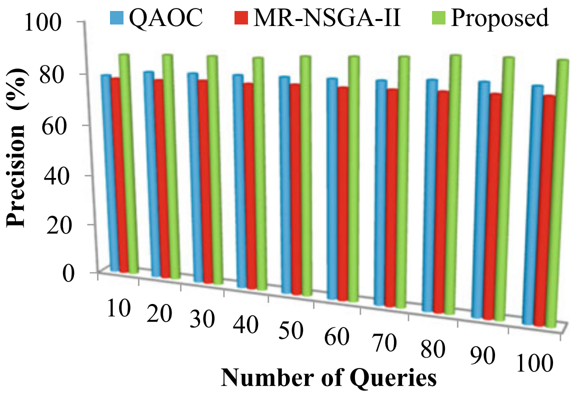


Fig. 12 Precision versus number of queries

proposed work attempts to retrieve the services correctly which it believes to be the relevant for the user query.

5.2.8 F-measure

It is a harmonic measure between the precision and recall. However, users are not retrieving the reliable results for their query. This is due to the index and ontology was not constructed or translated from one domain to another. While handling queries under large-scale environment, difficulties are retrieving the accurate response for the user query. F-measure means the same idea, i.e., how users retrieved the most relevant results for a given query. The performance is indicted in Fig. 13. It denotes that the performance of the proposed work is higher than the previous works. Adding security is one of the solutions, and also ontology and index processing gives the higher performance of f-measure (Fig. 14).

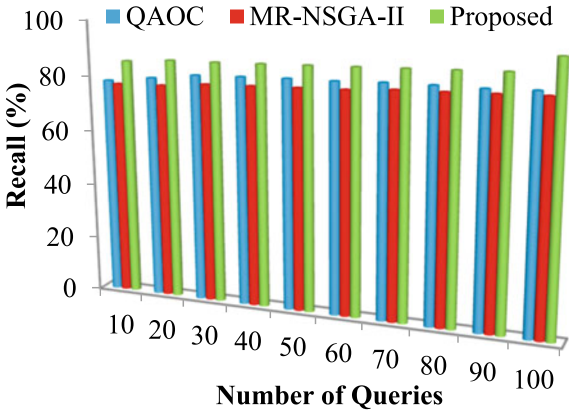
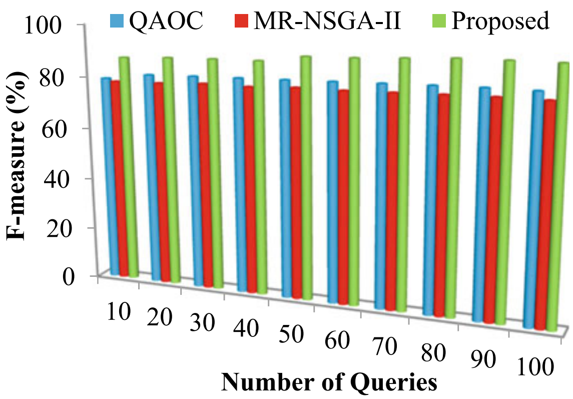


Fig. 13 Recall versus number of queries

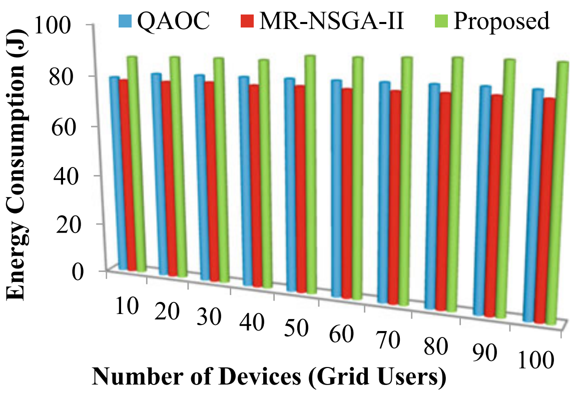
Fig. 14 F-measure versus number of queries



5.2.9 Energy Consumption

Energy consumption is a recent metric since current users and their associative devices are Internet of Things (IoT), which are resource constrained. Based on the energy level of devices, query scheduling process is presented. Achieving higher energy efficiency is a way of managing the system and accesses the results. Otherwise, all key features of the device are affected. The performance of energy consumption related to the number of queries for previous works and the proposed work is highly deviated. In previous works, query scheduling is performed only the waiting time and arrival time and also the job type, but user's device energy level is not analyzed. Due to this effect, energy consumption is higher for the previous works (Fig. 15).

Fig. 15 Energy consumption versus number of queries



5.3 Result Highlights

This section describes how the results obtained from the experiments for the proposed SecQSON model. The proposed model addresses the following research questions.

- How to improve the scalability of the proposed model for large number of data owners and users?
- How to improve the performance of the MapReduce mechanism for speedup the processing and make the effective solutions?
- How to handle the queries from users in a semantic manner?
- How to decrease the response time, authentication time, latency, and retrieval time for the data owners and users?

The above mentioned questions are addressed by the succeeding solutions.

- Proposed DuBK-RA algorithm takes lower time consumption since we use lightweight encryption algorithm. Presented DuBK-RA algorithm eliminates all unauthorized users at initial stage. Thus, resource consumption and time consumption are optimized.
- Scheduling by Map-Evaluate-Reduce with SHO considers all major metrics. Involvement of Map-Reduce also minimizes scheduling time.
- Semantic ontology with index is constructed to access the data from the given query of users

6 Conclusion

This research work mainly focuses on authorized semantic search over grid computing environment. The major objective is to minimize the search time and to improve the search accuracy for authorized user queries. For that, this work covers authentication, query scheduling, and retrieval processes. Firstly, we proposed a BLAKE-3 algorithm with multiple parameters for authentication such as ID, password, biometrics (fingerprint and vein). From the biometric data, bio-key is generated by the trusted authority. Bio-key is generated by hashing the biometric features of binary values. Then, user requests are scheduled using Map-Evaluation-Reduce framework. Here, evaluation is implemented by SHO algorithm. Then, semantic ontology is constructed in the grid server which consists of name node (meta data information of data nodes). In each data node, dendrimer index is constructed which is a lightweight tree that performance is higher than the binary tree, binary search tree, and AVL tree. Finally, the performance of the proposed SecQSON model is analyzed with respect several metrics and also compared to the QAOC and MR-NSGA-II.

In the future, we focus on blockchain-based technology which is more user-friendly and collect attacker's information based on geo-location.

References

1. Garg R, Singh A (2015) Adaptive workflow scheduling in grid computing based on dynamic resource availability. Eng Sci Technol Int J 18:256–269
2. Pujiyanta A, Nugroho LE, Widyawan E (2018) Planning and scheduling jobs on grid computing. In: 2018 international symposium on advanced intelligent informatics (SAIN)
3. Sathish K, RamaMohan Reddy A (2016) Workflow scheduling in grid computing environment using a hybrid GAACO approach. J Inst Eng (India):Series B 98(1):121–128
4. Hafaiedh IB (2019) A generic formal model for the comparison and analysis of distributed job-scheduling algorithms in grid environment. J Parallel Distribut Comput
5. Younis MT, Yang S (2018) Hybrid meta-heuristic algorithms for independent job scheduling in grid computing. Appl Soft Comput. https://doi.org/10.1016/j.asoc.2018.05.032
6. Chauhan A, Singh S, Negi S, Verma SK (2016) Algorithm for deadline based task scheduling in heterogeneous grid environment. In: 2016 2nd international conference on next generation computing technologies (NGCT)
7. Pandey R, Srivastava A, Rathore R (2017) An efficient resource scheduling algorithm using dynamic priority in grid computing. In: 2017 international conference on current trends in computer, electrical, electronics and communication (CTCEEC)
8. Kaya K, Aykanat C (2006) Iterative-improvement-based heuristics for adaptive scheduling of tasks sharing files on heterogeneous master-slave environments. IEEE Trans Parallel Distrib Syst 17(8):883–896
9. Kaya K, Aykanat C (2006) Iterative-improvement-based heuristics for adaptive scheduling of tasks sharing files on heterogeneous master-slave environments. IEEE Trans Parallel Distrib Syst 17:883–896
10. Singh H, Bawa S (2019) An improved integrated Grid and MapReduce-Hadoop architecture for spatial data: Hilbert TGS R-Tree-based IGSIM. Concurr Comput Pract Exp 31
11. Kumar ES, Vengatesan K (2018) Trust based resource selection with optimization technique. Clust Comput 22:207–213
12. Mahato DP, Sandhu JK, Singh NP, Kaushal V (2019) On scheduling transaction in grid computing using cuckoo search-ant colony optimization considering load. Clust Comput 1–22
13. Pujiyanta A, Nugroho LE, Widyawan W (2020) Resource allocation model for grid computing environment. Int J Adv Intell Inform 6:185–196
14. Shukla A, Kumar S, Singh H (2019) An improved resource allocation model for grid computing environment. Int J Intell Eng Syst 12:104–113
15. Mohamed MI, Hassan MF, Safdar S, Saleem MQ (2019) Adaptive security architectural model for protecting identity federation in service oriented computing. J King Saud Univ Comput Inform Sci
16. Wang D, Zhang X, Zhang Z, Wang P (2020) Understanding security failures of multi-factor authentication schemes for multi-server environments. Comput Secur 88
17. Wang P, Li B, Shi H, Shen Y, Wang D (2019) Revisiting anonymous two-factor authentication schemes for IoT-enabled devices in cloud computing environments. Secur Commun Netw 2516963:1–2516963:13
18. Chaudhry SA, Kim IL, Rho S, Farash MS, Shon T (2019) An improved anonymous authentication scheme for distributed mobile cloud computing services. Clust Comput 22:1595–1609
19. Kumar PS, Ramachandram S (2019) Fuzzy-based integration of security and trust in distributed computing. Adv Intell Syst Comput
20. Yuan X, Yuan X, Zhang YH, Li B, Wang C (2020) Enabling encrypted boolean queries in geographically distributed databases. IEEE Trans Parallel Distrib Syst 31:634–646
21. El-Gayar MM, Mekky NE, Atwan A, Soliman H (2019) Enhanced search engine using proposed framework and ranking algorithm based on semantic relations. IEEE Access 7:139337–139349
22. Guo K, Liang Z, Tang Y, Chi T (2017) SOR: An optimized semantic ontology retrieval algorithm for heterogeneous multimedia big data. J Computat Sci
23. Kaur N, Aggarwal H (2020) Query reformulation approach using domain specific ontology for semantic information retrieval. Int J Inform Technol

24. Ennahbaoui M, Idrissi H (2018) Zero-knowledge authentication and intrusion detection system for grid computing security
25. Devarajan R, Prakash M, Suresh J (2019) Computational grid scheduling architecture using mapreduce model-based non-dominated sorting genetic algorithm. Soft Comput 1–13.
26. Pradeep D, Sundar C (2020) QAOC: Novel query analysis and ontology-based clustering for data management in Hadoop. Futur Gener Comput Syst 108:849–860
27. Selvi S, Manimegalai D (2015) Task scheduling using two-phase variable neighborhood search algorithm on heterogeneous computing and grid environments. Arab J Sci Eng 40(3):817–844
28. Natesan G, Chokkalingam A (2019) Multi-objective task scheduling using hybrid whale genetic optimization algorithm in heterogeneous computing environment. Wireless Person Commun
29. Sahu T, Verma SK, Shakya M, Pandey R (2018) An enhanced round-robin-based job scheduling algorithm in grid computing. Lect Notes Data Eng Commun Technol 799–807
30. Entezari-Maleki R, Bagheri M, Mehri S, Movaghar A (2016) Performance aware scheduling considering resource availability in grid computing. Eng Comput 33(2):191–206
31. Poonam P, Shivani S, Satish R (2016) A genetic algorithm based scheduling algorithm for grid computing environments. In: Proceedings of fifth international conference on soft computing for problem solving, pp 165–173
32. Khajemohammadi H, Fanian A, Gulliver TA (2014) Efficient workflow scheduling for grid computing using a leveled multi-objective genetic algorithm. J Grid Comput 12(4):637–663
33. Rajan CD (2020) Design and implementation of fuzzy priority deadline job scheduling algorithm in heterogeneous grid computing. J Amb Intell Human Comput 1–8
34. Sahu DP, Singh K, Prakash S (2015) Maximizing availability and minimizing markesan for task scheduling in grid computing using NSGA II. In: Proceedings of the second international conference on computer and communication technologies, pp 219–224
35. Hassan SR, Pazardzievska J, Bourgeois J (2012) Fast attack detection using correlation and summarizing of security alerts in grid computing networks. J Supercomput 62:804–827
36. Chin J, Zhang N, Nenadic A, Bamasak O (2008) A context-constrained authorisation (CoCoA) framework for pervasive grid computing. Wireless Netw 16(6):1541–1556
37. Gnanaraj JW, Ezra K, Rajsingh E (2013) Smart card based time efficient authentication scheme for global grid computing. HCIS 3(1):16