Privacy Preservation in Big Data with Data Scalability and Efficiency Using Efficient and Secure Data Balanced Scheduling Algorithm

A Elsir\textsuperscript{1}, O Elsier\textsuperscript{2}, A Abdurrahman\textsuperscript{3} and A Mubarakali\textsuperscript{4}

\textsuperscript{1}Information System Department, King Khalid University, Saudi Arabia
\textsuperscript{2,3,4}College of Computer Science, Department of CNE, King Khalid University, Saudi Arabia

Received 8 December 2018; revised 9 June 2018; accepted 2 October 2019

In the current circumstance, the dimension of information in many kind of cloud atmosphere increases enormously along with the trend of Big Data. Subsequently making, it is a test for common software programming tools to get, supervise, and huge scale process information inside an explicit spans. Thus, it is a test for previous anonymization ways to agreement protection preservation on credential datasets because of their insufficiency of adaptability. To overcome these issues, efficient and Secure data balanced scheduling (ESDBS) algorithm is introduced to data maintain large-scale credential data sets using the MapReduce framework in big data aspects. The proposed system designs efficient MapReduce tasks to concretely achieve the specialization computation in a scalable manner. Based on the experimental evaluation, proposed ESDBS algorithm reduces 20 seconds encryption time and improves 15\% algorithm efficiency and 6\% system throughput compare than convention techniques.

Keywords: Big data privacy, Efficient job balanced scheduling, MapReduce, Merging, Level-level privacy, clustering efficiency, scalability

Introduction

Author\textsuperscript{1} presented to encoded deduplication information stored in distributed storage system relied upon proprietorship challenge and proxy re-encryption. It fuses the cloud information deduplication with access control. In\textsuperscript{2} designed an Attribute-Based Encryption (ABE) algorithm for de-duplicating encrypted data stored in the cloud storage server and supported protected data. The cloud service supplier was a storage service and performed truthfully on information storage and organization to gain commercial revenue. In\textsuperscript{3} presented an attribute-based storage framework with protected de-duplication in a hybrid cloud environment, where a private cloud is responsible for duplicate detection, and a public cloud manages the storage capacity. In\textsuperscript{5} introduced Deduplication Aware Resemblance Detection and Elimination (DARE), which exploits duplicate-adjacency information. The idea of DARE is to employ, call Duplicate-Adjacency based Resemblance Detection. In\textsuperscript{7} introduced Over Flow, a uniform data management system for scientific workflows. This solution is environment-aware, as it monitors and offering high and predictable data handling performance for transfer cost and time. In\textsuperscript{6} discussed on confidentiality and safety concerns, distinguishes between confidentiality, security and privacy requirements in big data. In\textsuperscript{8} designed a secure protocol for the deduplication of horizontally partitioned datasets with deterministic record linkage algorithms.

Proposed methodology

The MapReduce computation paradigm is stills a challenge to perform efficient and secure data balanced scheduling technique for maintaining the large scale of secure data. Existing data anonymization technique has less privacy. The approach applies to the small number of data sets. To overcome these issues, efficient and secure data balanced scheduling (ESDBS) algorithm is designed to maintain large-scale credential data sets utilizing the MapReduce framework in big data attributes. The ESDBS approach designs a group of efficient MapReduce tasks to achieve the specialization computation to improve the data scalability concretely.

The paper contribution is following as:

- Design efficient and Secure data balanced scheduling (ESDBS) approach to data maintain
large-scale credential data sets using the MapReduce framework

- Propose an algorithm to grow high scalability via permitting specializations to be conducted on multiple data partitions in a parallel way.
- Improved the experimental result of data execution time, scalability compare than existing approaches.

The rest of the paper contribution is followed as literary works which closest to proposed mechanisms. The proposed methodology framework exaction and implemented algorithm details. The implementation setup and achieved the result of proposed approaches.

Conclusion section elaborates the proposed method, implemented modules and proposed algorithm details. The proposed design main objective is to provide a framework which produced efficient data portioning, job scheduling with efficient privacy for large scale of a dataset. The diagrammatic representation of the proposed mechanism is explained in figure 1.

Data Pre-processing
Data processing loads the input dataset in the framework to perform the task. Here, this design uploads the student academic records unstructured dataset to perform efficient job balanced scheduling for performing a task with privacy.

Data Distribution
Data distribution is a process to perform data distribution in Hadoop. It distributes a large volume of collected un-structured data to a respective node for performing the task as per given schedule.

Data Splitting
Data splitting is the process to break the dataset into small data sets to perform a task effectively. It calculates the length of a file, attribute, and records separately. Hence, it assigns the random number for every split datasets. Next, its move for the clustering process as per task schedule.

Data Merging
Data merging is a process to integrate all small packet cluster data in a single dataset. The process utilizes to combine the small clustered intermediate result. In the process, a weight of attributes can be calculated and displayed for a data field. Hence, it proceeds to apply level by level security. Finally, the intermediate clustered results are processed for the specialization process.

Level-level Privacy
Once, the design receives individual data sets, and it applies level by level privacy. It protects sensitive attribute in data sets. Hence, this process is integrated with merging cuts. The level by level privacy designed to ensure the privacy of combined sensitive data. This approach protects credential information of reduced data sets. It dedicates to maintain adequate privacy during data transformations. Data Table T which contains identifiers A_i (i=1, 2…n) and credential attribute C. Initially, this approach is applied all over selected apparently identifiers A_i and then extracted and visible by a proposed novel procedure. The proposed approach presents unique data extraction operation with hidden attribute table T. After data extraction, the records in the table T are clustered and formed in many groups like G1, G2, G3…Gn. Each group is arranged in category wise with many attributes value of apparently identifiers attribute P_i (i=1, 2, … z). Among the apparent identifiers A_i, one with more distinct values is

---

![Fig. 1 — Diagrammatic representation of Apply Efficient and Secure data balanced scheduling Approach](image-url)
selected. In every group Gi, the next closest and lower integer value Li, and the next closest highest integer value Hi is obtained.

**Specialization**

Specialization is processed to integrate clustered intermediate result in a single database structured formats. Once again, level-level privacy works to protect credential attributes from merged data. It contains two types of task like as update and Initialization.

**Efficient and Secure data balanced scheduling (ESDBS) Algorithm**

Efficient and Secure data balanced scheduling (ESDBS) approach is designed to data maintain large-scale credential data sets utilizing the MapReduce framework in big data attributes. The ESDBS algorithm develops a group of efficient MapReduce tasks to achieve the specialization computation to improve the data scalability concretely. This main design focuses on schedule the job based on time and size of the dataset. Here, data sets a split into the small packet to perform a task effectively. Before starting the task, this design calculates the length of file, records, and number of attributes. Hence, it measures the weight of every attribute by using cut functions. Here, MapReduce has two levels of parallelization such as job level and task level. The job level performs numerous jobs concurrently to create full utilization of Hadoop infrastructure assets. Task level utilizes the parallelization process to assign multiple mapper/reducer tasks to execute the parallel from a split dataset. Random sampling technique is used to split the data set D. Where, this method calculates each attributes value of the dataset. Hence, this approach calculates the weight of each attributes then it proceeds to categorize the credential attributes for total attributes. A proposed approach is capable of handling the large volume of a dataset. The pseudo of the proposed algorithm is given below in details: Initially, an original data set D is split into small-small packages. Where, Di, 1 ≤ i ≤ p, denote the dataset distribution from original dataset D. Where, p is the total number of split packages, and D = Σi=1p Di, Di ∩ Dj =Ф 1 ≤ i < j ≤ p. Then, proposed techniques conduct subroutine over every partitioned data set in the sequential mode to make full utilization of the job level parallelization of MapReduce.

**Input:** Load un-structured Original Dataset D, and the number of partitions p.

**Output:** Structured visualize data with weighted attributes.

**Process:**

- Load the un-structured Original Dataset D;
- Partition D into Di, 1 ≤ i ≤ p using random sampling methods;
- Apply level-level privacy un-structured Original dataset D;
- Calculate the weight of each attribute value (Di, ki, AL0) → A L’i, 1 ≤ i ≤ p in parallel as multiple MapReduce tasks.
- Merge all portioned data into one, merge (AL1′ AL2′ . . .A Lp′)→A LI.
- Visualize (D, k, ALI) → A L*.

**Data Visualization**

Data visualization processes visualize the clustered result with privacy based on the requirement of applications. This module displays the data with privacy where a user can apply multiple types of attributes privacy. It produces the details for a completed task and reduced task details. It also displays the clustering efficiency and execution time of given dataset.

**Result and Discussion**

**Experimental Setup**

To evaluate the proposed mechanism with the existing method, experiments were conducted with Intel Core i7 6700 CPU 3.40 GHz Processor with 16 GB RAM running with Ubuntu 12.0, JDK 1.7, Eclipse, Virtual box 50.12, Cloudera Hadoop 5.8.0 with Hive database.

**Dataset**

The academic dataset is utilized with student id, name, college, attendance and district attributes with 100 user-generated records to evaluate the performance of proposed approaches.

**Performance Matrix**

The performance matrix expresses the evaluation parameter namely as system throughput, encryption time and algorithm efficiency on academic data to measure the performance of the proposed approach.

**System Throughput**

System throughput is the average of successful data distributed in HDFS. The average system throughput is calculated using Equation (1).

\[
\text{Throughput System} = \sum_{\text{Dataset}}^n \text{Distributed \ (n)*Record \ Size/1000} \quad \ldots (1)
\]
Encryption Time

Encryption time is computed to evaluate the encoding efficiency of the proposed approach where encryption time is computed based on the size of records and total time to perform the encoding process.

The encryption time is expressed in equation (2).

\[ ET = \frac{T_{size\ of\ records}}{T_{end} - T_{start}} \]  \hspace{1cm} (2)

Where \( T_{size\ of\ records} \) represents the size of records, \( T_{end} \) is encryption complete time and \( T_{start} \) is encryption initialization time.

Algorithm Efficiency

Algorithm efficiency is computed based on time, memory and complexity efficiency of the algorithm. Here, cost and time function also varies to evaluate the performance of the proposed approach.

\[ f(n) = O(g(n)) \]  \hspace{1cm} (3)

\[ |f(n)| \leq c|g(n)| \]  \hspace{1cm} (4)

Where \( f \) & \( g \) are two positive functions, \( g \) is asymptotically dominated by \( f \) and \( n \) is the size of records. \( f = O(g) \) if and only if \( g \) asymptotically dominates \( f \) then \( f \) is a sequence of \( g \). Where is \( c \) represents positive constant for a large number of records? Table 1 display encryption time in seconds, algorithm efficiency in percentage and system throughput in percentage for an academic dataset with existing approaches to evaluates the efficiency of the proposed approach. The proposed method is evaluated with Proxy (Proxy Re-encryption) KP-ABE (Key-Policy Attribute-Based Encryption) AES (Advanced Encryption Standard) techniques to measure encryption time, algorithm efficiency and system throughput. Table 1 displays the average value of overall parameters along with academic dataset. According to Table 1 comparative analysis, it states that the proposed method performs well on each respective parameter. According to Table 1 observations, the ESDBS technique is estimated based on encryption time, algorithm efficiency and system throughput with an existing algorithm. The ESDBS algorithm is calculated with Proxy, KP-ABE and AES methodologies behalf of on encryption time, algorithm efficiency and system throughput to evaluate the efficiency of the proposed algorithm. Proxy Re-encryption provides access control to a protected file framework, and it sends secure key from the user. Therefore, the key is unwanted to be located on the cloud storage server. However, it consumes more storage space, and it takes a long time to encrypt and decrypt the files. The ESDBS algorithm takes less time to encrypt and decrypt files and consumes less storage space. The Public Key Attribute-Based Encryption (PK-ABE) allows information to encrypt under the public key of the recipient like nobody except a legitimate receiver could get access the information. It doesn't fulfill the requirements of the client in the big data. In this Decryption, encryption and re-encryption time utilization is more. The primary disadvantage of KP-ABE has increased the size of the key. The ESDBS key size is small. AES is the nearest competitor algorithm for the overall parameter. The AES is a block cipher (Advanced Encryption Standard) is extensively utilized and has been analyzed lengthily. With 128-bits key length AES, the symmetric key encryption algorithm is utilized. AES algorithm is used to re-encode the information once more. In various public clouds, the information should be secure and protected. However, the AES algorithm provides low system efficiency and throughput. The proposed method also provides efficiency and high throughput of data, and it takes less time to predict using academic dataset. Proposed ESDBS reduced 20sec encryption time and improve 15% (algorithm efficiency) and 6% system throughput. Finally, the paper claims that the proposed ESDBS algorithm performs best on every evaluation matrix and respective input parameters.

### Conclusion

This paper presents efficient and secure data balanced scheduling (ESDBS) approach is designed to data maintained large-scale credential data sets utilizing the MapReduce framework in big data features. The ESDBS approach developed a group of efficient MapReduce tasks to achieve the specialization computation to improve the data scalability concretely. Proposed design improves scalability and efficiency. This approach provides level wise data privacy for credential attributes. Here, the overall performance of providing privacy is high. Its ability to handles a large number of data sets. This main design focus to schedule the job based on time...

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Proxy</th>
<th>KP-ABE</th>
<th>AES</th>
<th>ESDBS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encryption time</td>
<td>120</td>
<td>80</td>
<td>60</td>
<td>40</td>
</tr>
<tr>
<td>Algorithm efficiency</td>
<td>60</td>
<td>70</td>
<td>80</td>
<td>95</td>
</tr>
<tr>
<td>System Throughput</td>
<td>67</td>
<td>78</td>
<td>89</td>
<td>95</td>
</tr>
</tbody>
</table>
and size of the dataset. Here data sets a split into small pack packet to perform the task effectively. Before starting the task, this design calculates the length of file, records, and number of attributes. Hence, it measures the weight of every attribute by using cut functions. The ESDBS method is evaluated with data scalability, encryption time, algorithm efficiency, and system throughput for an academic dataset with existing approaches. This system improves the, 6% system throughput, 20sec encryption time, and 15% algorithm efficiency. In the future, this paper can be extended to execute the moving live data like traffic, medical weather data for performing MapReduce in an efficient and secure way.

Acknowledgement
The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through General Research Project under grant number (G.R.P- 14-1441). I would like to thank King Khalid university for the necessary support to lead this paper, we thank our colleagues who sustained greatly assisted this research. We would also like to show our gratitude to for sharing their pearls of wisdom with us during this research, and we thank “anonymous” reviewers for their so-called insights.

References