International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 15 (2018) pp. 11888-11900
© Research India Publications. http://www.ripublication.com

Generalized Lasso Prediction Based Lyapunov Weighted Round Robin Seamless Scheduling for Big Stream Data Processing

N. Arunadevi[1], **Dr. Vidyaa Thulasiraman**[2]

[1]*Part-Time Research Scholar, Department of Computer Science, Periyar University, Salem, Tamilnadu, India.*

[2]*Assistant Professor, Department of Computer Science, Govt Arts & Science College for Women, Bargur, Tamilnadu, India.*

Abstract

Streaming data analysis is significant in big data processing. The scheduling of streaming data is difficult due to its fast data arriving speed and huge size. Few research works have been designed for predictive scheduling of big stream data processing. The performance of existing scheduling technique was not effectual. In order to solve this drawback, a Generalized Lasso Prediction Based Lyapunov Weighted Round Robin Seamless Scheduling (GLP-LWRRSS) Technique is proposed. The GLP-LWRRSS technique is designed with the key objective of improving scheduling performance of big stream data processing with higher prediction accuracy and lesser scheduling time. The GLP-LWRRSS technique employs Generalized LASSO Predictive Model to accurately predict average data processing time by using selected features of processing unit with higher accuracy. After prediction, GLP-LWRRSS technique used Lyapunov Weighted Round Robin Seamless Scheduling algorithm where stream data tasks are scheduled to appropriate processing units with lower time. This assists for GLP-LWRRSS technique to increases the scheduling efficiency of big stream data processing. As a result, GLP-LWRRSS Technique attains improved performance of predictive scheduling for big stream data processing as compared to state-of-the-art works. The GLP-LWRRSS technique conducts the experimental works using metrics such as prediction accuracy, false positive rate, scheduling efficiency and scheduling time with respect to different number of big stream data. The experimental results show that GLP-LWRRSS technique is able to increase the prediction accuracy and scheduling efficiency of big stream data processing as compared to existing works.

Keywords: Features, Generalized LASSO, Lyapunov Function, Stream Data, Weighted Round Robin Seamless Scheduling, Processing Unit

INTRODUCTION

Big data is commonly used for business operations in today's competitive scenario. Big data stream is a continuous data streams with huge size. With the development of Internet of Things (IoT), personal computing, and electronic commerce, streaming data analysis has become a significant topic in science and commercial areas. As an important part of big data, streaming data is not easy to be analyzed in real time owing to the data speed and huge size of data set in stream model. Predictive resource scheduling is used to improve the performance by leveraging big data analytics. A lot of research works has been developed in big data analytics of predictive scheduling for stream data. But, predictive scheduling of big stream data processing is still a challenging task. Therefore, there is a requirement for novel technique for improving predictive scheduling performance of big stream data processing.

A prediction scheduling algorithm was presented in [1] for scheduling threads to machines based on the prediction results. The prediction accuracy was lower. A Dynamic Assignment Scheduling (DAS) algorithm was intended in [2] for big data stream processing in mobile Internet services. The scheduling efficiency of this algorithm was not improved.

A new predictive scheduling framework was designed in [3] allows fast stream data processing. The computational complexity of this framework was not reduced. An energy-efficient scheduling was introduced in [4] to increase the performance results of big-data streaming applications. The memory consumption was higher.

Re-Stream framework was presented in [5] to minimize response time of big data stream processing. An adaptive moving window regression was used in [6] for predictive analytics of complex IoT data streams. The error rate and time complexity was not solved.

A stable online scheduling strategy was introduced in [7] for efficient system stability and minimizing response time of big data streams. A node scheduling model was developed in [8] with application of Markov chain for analyzing big streaming data in real time. Scheduling time for big stream data analysis was not solved.

Streaming analytics of Real-time big data was presented in [9].However predictive scheduling was remained unaddressed. Incremental partial least squares analysis technique was introduced in [10] for big streaming data. The time complexity was more.

In order to resolve the above said existing drawbacks, GLP-LWRRSS Technique is developed. The main contribution of GLP-LWRRSS Technique is formulated as follows,

- ❖ To increase the performance of predictive scheduling for big stream data processing as compared to state-of-the-art works, GLP-LWRRSS Technique is designed by using Generalized LASSO Predictive Model and Lyapunov Weighted Round Robin Seamless Scheduling (LWRRSS) algorithm.
- ❖ To get improved performance for predicting average data processing time, Generalized LASSO Predictive Model is applied in GLP-LWRRSS Technique. The Generalized LASSO Predictive Model performs feature selection in which minimal numbers of features are chosen to attain higher prediction accuracy for big stream data processing.
- ❖ To enhance the scheduling efficiency of big stream data processing with higher efficiency and lower time as compared to conventional techniques, Lyapunov Weighted Round Robin Seamless Scheduling (LWRRSS) algorithm is utilized in GLP-LWRRSS Technique. To handle stream data during scheduling, Lyapunov function is employed in GLP-LWRRSS Technique on the contrary to traditional scheduling schemes.

The remaining structure of the paper is formulated as follows. Section 2 portrays the different techniques developed for predictive scheduling of big stream data analysis. In Section 3, the proposed GLP-LWRRSS Technique is explained with help of architecture diagram. The experimental settings and comparative results analysis of GLP-LWRRSS Technique is shown in Section 4 and Section 5. Section 6 presents the conclusion of the paper.

RELATED WORKS

A genetic algorithm was used in [11] with intention of enhancing performance of scheduling for streaming applications. The scheduling time was very higher. A novel technique was designed in [12] to predict the data characteristics such as volume, velocity, variety, variability, and veracity and to perform resource allocation for big data streams. The predictive accuracy was not adequate. Runtime-aware adaptive scheduling was applied in [13] for processing stream data with less computational resources. The scheduling performance was poor. An Elastic and Scalable Data Streaming System was introduced in [14] for dynamic load balancing and lessening the computational resources usage. The scheduling time was remained unaddressed.

An online energy-efficient scheduler was developed in [15] for enhancing the QoS of big-data streaming with minimal energy and resources utilization. The efficiency of scheduling was lower. An efficient algorithm was designed in [16] for scheduling streaming data warehouses. The computational complexity of this algorithm was higher. A review of different techniques intended for distributed data stream processing was analyzed in [17].

A Game-Theoretic Approach was presented in [18] for data stream processing. Predictive scheduling was not solved in this approach. A novel technique was designed in [19] to solve the problem related with data stream processing applications. Quality-based Workload Scaling for Real-time Streaming Systems was predicted in [20]. The scheduling issues of stream data were not solved efficiently.

To overcome above mentioned existing problems of stream data predictive scheduling, GLP-LWRRSS Technique is introduced which is explained in below section.

OPTIMIZED FEATURE SELECTION BASED PREDICTIVE ROUND ROBIN SCHEDULING TECHNIQUE

A stream data processing system handles unbounded streams of data tuples which may last for a long time. Hence, the average data processing time is considered as the significant performance metric to get better performance for scheduling of big stream data. In [1], accurately predicting data processing time in a stream data processing system is very difficult and also not yet well solved problem. The accurate prediction of data processing time on processing unit or virtual machine helps to make better decision during task scheduling and also enhances scheduling performance. Therefore, Generalized Lasso Prediction Based Lyapunov Weighted Round Robin Seamless Scheduling (GLP-LWRRSS) Technique is introduced. The GLP-LWRRSS Technique is proposed with application of Generalized LASSO Predictive Model and Lyapunov Weighted Round Robin Seamless Scheduling (LWRRSS) algorithm for efficient task scheduling of big stream data. The architecture diagram of GLP-LWRRSS Technique is demonstrated in below Figure 1.

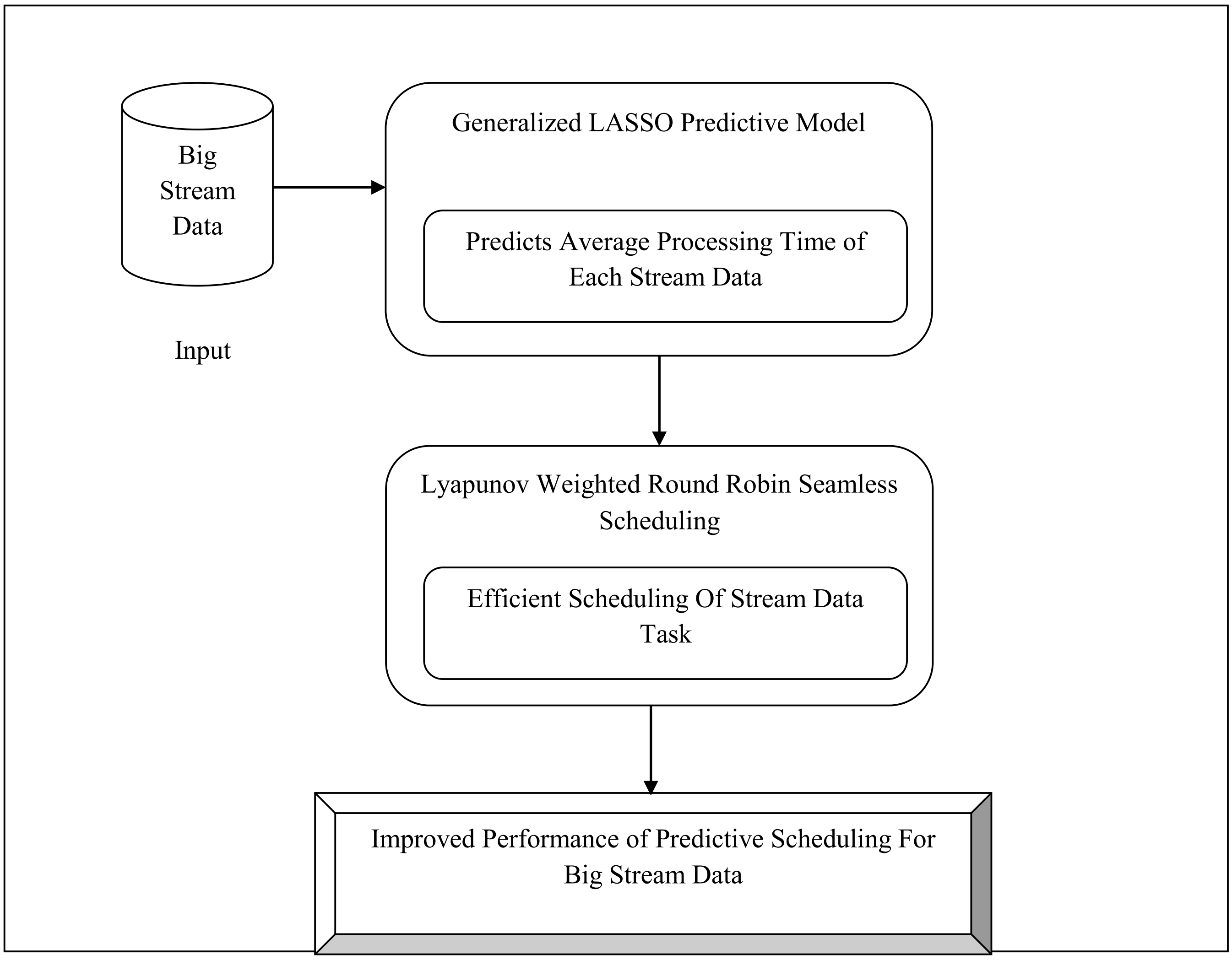
Big Stream Data
Input
Generalized LASSO Predictive Model
Predicts Average Processing Time of Each Stream Data
Lyapunov Weighted Round Robin Seamless Scheduling
Efficient Scheduling Of Stream Data Task
Improved Performance of Predictive Scheduling For Big Stream Data

Figure 1. Architecture Diagram of GLP-LWRRSS Technique for Predictive Scheduling of Big Stream Data

Figure 1 demonstrates the overall flow process of GLP-LWRRSS Technique for predictive scheduling of big stream data processing with higher efficiency. As shown in Figure 1, GLP-LWRRSS Technique at first takes Big Stream Data (i.e. Apache Storm dataset) as input. Then, GLP-LWRRSS Technique used Generalized LASSO Predictive Model with aim of determining average processing time for each stream data with higher accuracy through feature selection. After that, GLP-LWRRSS Technique applied Lyapunov Weighted Round Robin Seamless Scheduling algorithm in order to effectively schedule stream data tasks to appropriate PU with lower amount of time utilization. Thus, GLP-LWRRSS Technique obtains enhanced performance of predictive scheduling for big stream data when compared to existing works. The detailed process of GLP-LWRRSS Technique is shown in below sub sections.

Generalized LASSO Predictive Model

The Generalized LASSO Predictive model is a regression analysis that is used in GLP-LWRRSS Technique in order to enhance the prediction accuracy of stream data processing time. The Generalized LASSO Predictive model is employed to efficiently predict the average data processing time of each PU to schedule task when streaming data (i.e. continuous data tasks) considered as input. On the contrary to different techniques, Generalized LASSO Predictive model is applied in GLP-LWRRSS Technique because it contains good computational properties for both parameter estimation (i.e. prediction of average data processing time) and feature selection. Furthermore, Generalized LASSO Predictive model is generally utilized, when more number of features is considered as it automatically performs feature selection and thereby optimizes features for increasing prediction performance with minimal false positive rate. Therefore, GLP-LWRRSS Technique used Generalized LASSO Predictive model to efficiently compute average data processing time during the task scheduling of big stream data.

As in [1], GLP-LWRRSS Technique considers set of features for predicting the average data processing time of PU. The following are some of features considered in [1] for effective scheduling of stream data such as CPU time, Memory, Thread-Workload, Machine-Workload and Bandwidth etc. The proposed GLP-LWRRSS Technique also considers above said features of processing unit in order to estimate average data processing. Considering all the features for predicting average data processing time on PU takes more amount of time. To optimize the number of features for prediction, Generalized LASSO Predictive model is applied in GLP-LWRRSS Technique. The Generalized LASSO Predictive model performs feature selection based on dependency between input features and output values.

The Generalized Lasso optimization problem is mathematically formulated as,

$$\min_{f \in R^d} \frac{1}{2} \|y - X^T f\|_2^2 + \lambda \|f\|_1 \quad (1)$$

From equation (1), '$f = [f_1, f_2, \dots, f_k]^T$' represent a regression coefficient vector (i.e. Features considered) and f_k denotes regression coefficient of the 'k^{th}'feature. Here, '$\|\cdot\|_1$' and '$\|\cdot\|_2$' are the l_1- and l_2- norms whereas '$\lambda > 0$' is the regularization parameter. The l_1-regurlarization Lasso tends to produce a sparse solution which means that the regression coefficients for irrelevant features become zero. By using the above equation, Generalized LASSO Predictive model selects the CPU time, memory and bandwidth as more relevant features for predicting the average processing time of stream data.

After selecting features, Generalized LASSO Predictive model carry out regression analysis to evaluate average data processing time. With helps of the regression analysis, Generalized LASSO Predictive model measures the relationship between a dependent variable and one or more independent variables. Here, independent variables are the selected features namely CPU time, memory and bandwidth. The dependent variable represents the output i.e. average data processing time of processing unit. By using these independent variables, Generalized LASSO Predictive model exactly determines the average data processing time for all PUs. Based on the predicted results, then scheduling task is efficiently carried out. The block diagram of Generalized LASSO Predictive model is illustrated in Figure 2.

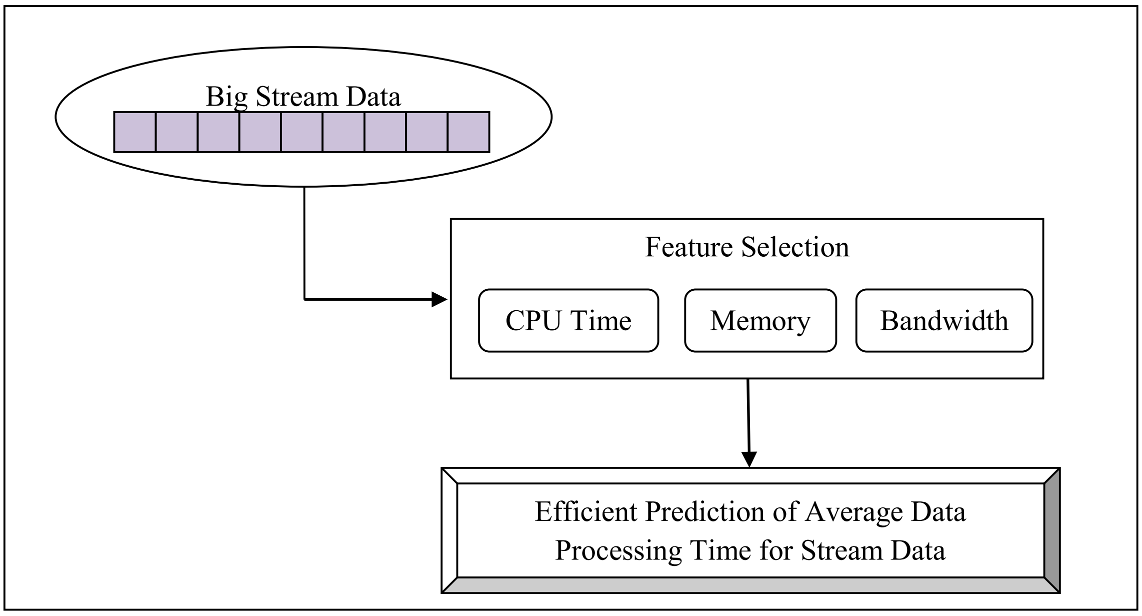
Big Stream Data
Feature Selection
CPU Time
Memory
Bandwidth
Efficient Prediction of Average Data Processing Time for Stream Data

Figure 2. Process of Generalized LASSO Predictive Model for Analytics of Big Stream Data

Figure 2 shows the flow processes of Generalized LASSO predictive model to get better performance for computing average processing time of big stream data. As presented in figure, Generalized LASSO Predictive model at first perform feature selection process. Here the features that is more significant for accurately estimating average processing time of stream data on PU. This helps for Generalized LASSO Predictive model to determine average processing time of stream data with higher prediction accuracy and lower false positive rate.

Let us consider a stream data represented as '$d_i = d_1, d_2, .., d_n$' and number of processing unit denoted as '$PU_i = PU_1, PU_2, .., PU_n$'. The Generalized LASSO Predictive model used three features namely CPU time, memory, and bandwidth of processing unit for efficient prediction. The CPU time is measured as the amount of time needed by a PU to process the data task. The CPU time of PU to perform task of stream data 'd_i' is mathematically measured as,

$$CPU\ time_{PU_i} = T(E(d_i)) \quad (2)$$

From equation (2), '$CPU\ time_{PU_i}$' needed by 'PU_i' whereas '$T(E(d_i))$' represents the time taken by PU to complete data task. Then, bandwidth utilization is estimated as average rate of data transfer to complete data task on PU which is computed as,

$$BU_{PU_i} = [B_I - BC_{d_i}] \quad (3)$$

From equation (3), 'B_I' denotes the initially available bandwidth of PU and 'BC_{d_i}' represents an amount of bandwidth consumed by PU to execute data task.

Followed by, memory is determined as a total amount of storage space needed by PU to perform data task. The memory consumption of PU 'M_{PU_i}' is estimated in terms of megabytes (MB). The amount of memory space utilized by a PU to complete data task is mathematically formulated as,

$$M_{PU_i} = [M_I - M_U] \quad (4)$$

From equation (4), memory is measured as distinctions between the initial memory size 'M_I' and unused memory space 'M_U'. With helps of above equation (2), (3), (4), Generalized LASSO Predictive model evaluates CPU time, bandwidth and memory required by PU to carry out stream data task. Thus, generalized LASSO Predictive model predicts the average data processing time on 'PU_i' using below mathematical expression,

$$APT_{PU_i} = \{CPU\ time_{PU_i},\ BU_{PU_i},\ M_{PU_i}\} \quad (5)$$

From equation (5), the average data processing time is evaluated through regression analysis. By using regression analysis, the Generalized LASSO Predictive model determines stream data with minimum CPU time and minimum task size (i.e. memory) needs lower average data processing time and higher CPU time and large task size requires higher average data processing time. The Generalized LASSO Predictive model applied in GLP-LWRRSS Technique improves performance of predicting tuple processing time in a stream data processing through feature selection. The algorithmic processes of Generalized LASSO Predictive model is explained in below.

```
// Generalized LASSO Predictive Algorithm
Input: Big Stream Data '$d_i = d_1, d_2, .., d_n$'
Output: Increased Predictive Accuracy and reduced false positive rate
Step 1: Begin
Step 2:     Perform feature selection using (1)
Step 3:     For Stream of Data '$d_i$'
Step 4:         For processing unit '$PU_i$'
Step 5:             Evaluate CPU time needed for processing data task '$d_i$' on PU using (2)
Step 6:             Compute memory space '$M_{PU_i}$' using (4)
Step 7:             Measure Bandwidth rate '$BU_{PU_i}$' using (3)
Step 8:             Measure average processing time of each processing unit '$APT_{PU_i}$' using (5)
Step 9:         End For
Step 10:    End For
Step 11:End
```

Algorithm 1 Generalized LASSO Predictive Model

Algorithm 1 demonstrates the step by step processes of Generalized LASSO Predictive Model to obtain enhanced prediction performance for computing average data processing time in a big stream data processing. With help of above algorithmic processes, Generalized LASSO Predictive Model significantly predicts the average data processing time of big stream data with higher accuracy. Hence, GLP-LWRRSS Technique enhances the prediction accuracy and lessens false positive rate of stream data process as compared to existing works.

Lyapunov Weighted Round Robin Seamless Scheduling

The Lyapunov Weighted Round Robin Seamless Scheduling (LWRRSS) algorithm is designed in GLP-LWRRSS Technique to handle PUs with different processing capacities under a big stream data. Each PU is assigned a weight based on determined average processing time of each stream data. The weight is an integer value that indicates the processing capacity. The PU which provides minimal average processing time for given stream data is assigned with higher weight among all other processing units. From that, PU with higher weight is first scheduled to analyze stream data than those with less weight. The LWRRSS algorithm allocates the each stream data to the most suitable PUs based on weight value. The

LWRRSS algorithm uses average processing time of data to select appropriate PU for scheduling.

In order to efficiently schedule stream data without any fault, Lyapunov function is used in LWRRSS algorithm on the contrary to existing techniques. This Lyapunov function helps for LWRRSS algorithm to optimally manage stream data for seamless task scheduling. In addition to that, Lyapunov functions are used extensively to ensure scheduling stability of LWRRSS algorithm for big stream data processing. As a result, GLP-LWRRSS Technique attains better scheduling performance for big stream data as compared to state-of-the-art works. The following diagram shows the process of LWRRSS algorithm for task scheduling of big stream data processing.

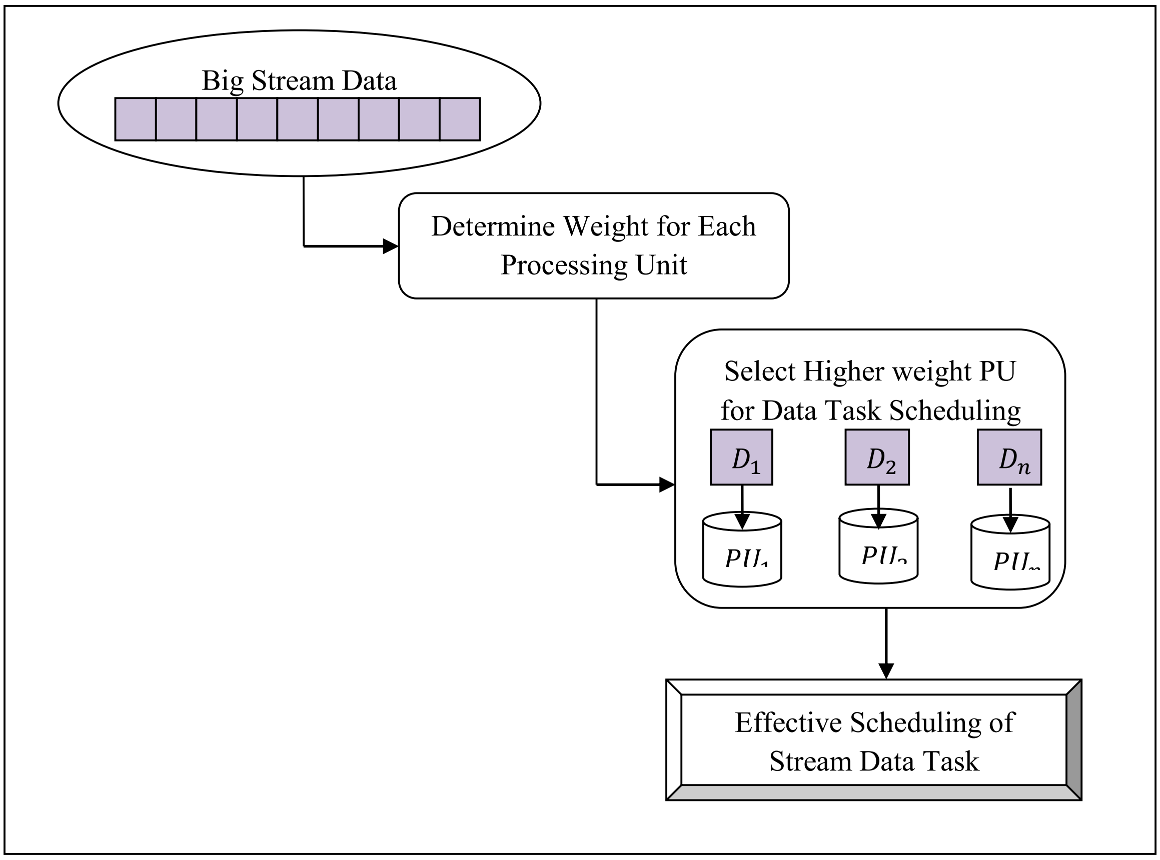
Big Stream Data
Determine Weight for Each Processing Unit
Select Higher weight PU for Data Task Scheduling
D_1
D_2
D_n
PU_1
PU_2
PU_n
Effective Scheduling of Stream Data Task

Figure 3. Flow processes of Lyapunov Weighted Round Robin Seamless Scheduling for Big Stream Data Analytics

Figure 3 depicts the flow process of LWRRSS algorithm for efficient scheduling of big stream data. As demonstrated in figure, LWRRSS algorithm measures weight for each PU based on average processing time of data. Followed by, LWRRSS algorithm picks higher weight PU among others for data task scheduling with higher accuracy and minimal time.

Let us consider 'n' number of processing units (PUs) denoted as '$PU_i = PU_1, PU_2, .., PU_n$'. The stream data '$d_i = d_1, d_2, .., d_n$' is schedule to available PUs for storage or pass it to some other units for further processing. To increase the scheduling performance of big stream data, Lyapunov function is employed in LWRRSS algorithm. The Lyapunov function controls flow of stream data at time 't' for stable task scheduling using below mathematically formula,

$$L(t) = \frac{1}{2}\sum_{i=1}^{N} d_i\,(t)^2 \qquad (6)$$

From equation (6), 'N' denotes number of stream data task for each time slot 't'. Then, LWRRSS schedule each stream data to appropriate PU by considering weight values of all PUs. The weight value of each PU 'W_{PU_i}' is estimated using below mathematical formulation,

$$W_{PU_i} \rightarrow \sum_{i=1}^{n} APT_{PU_i} \qquad (7)$$

From equation (7), 'APT_{PU_i}' represent average processing time of 'PU_i'to process data task. By using the above equation (7), the weight value is evaluated for each PU to make a wise decision on task scheduling for big stream data. Then, the LWRRSS algorithm picks PU with higher weight to schedule data which is mathematically expressed as,

$$d_i \rightarrow arg\ \max W_{PU_i} \qquad (8)$$

From equation (8), '$arg\ \max W_{PU_i}$' refers maximum weight value of PU. With the help of above equation (8), LWRRSS algorithm finds the PU with higher weight value among multiple processing units to process data. This helps for LWRRSS algorithm to efficiently schedule stream data to appropriate PU with higher accuracy and minimal computational complexity. The algorithmic processes of Lyapunov Weighted Round Robin Seamless Scheduling is demonstrated in below,

```
// Lyapunov Weighted Round Robin Seamless Scheduling Algorithm
Input: Stream Data '$d_i = d_1, d_2, .., d_n$', Processing Unit '$PU_i = PU_1, PU_2, .., PU_n$'
Output: Improved Scheduling Efficiency and Reduced Scheduling Time
Step 1: Begin
Step 2:     For stream of data '$d_i$'
Step 3:         For processing unit '$PU_i$'
Step 4:             Assign weight to each processing unit '$PU_i$' based on estimated average
                    processing time using (7)
Step 5:             Select higher weight PU among all other PUs for scheduling data using (8)
Step 6:         End For
Step 7:     End For
Step 8: End
```

Algorithm 2 Weighted Round-Robin Scheduling Algorithm

Algorithm 2 explains the step by step process of LWRRSS algorithm to get enhanced scheduling performance for big stream data. As demonstrated in algorithm 2, the LWRRSS algorithm determines weights for each PU according to average processing time. Subsequently, LWRRSS algorithm selects the higher weight PU among all other PUs to schedule tasks for each stream of data. This supports for LWRRSS algorithm to achieve improved scheduling efficiency for big stream data with minimal time complexity.

EXPERIMENTAL SETTINGS

To determine the performance of proposed, GLP-LWRRSS Technique is implemented in Java Language using Amazon EC2 dataset. This Amazon EC2 dataset manage streaming data solution in the cloud. The GLP-LWRRSS Technique considers different number of stream data tasks from Amazon EC2 dataset for experimental evaluation to achieve higher scheduling efficiency. The efficiency of proposed GLP-LWRRSS Technique is measured in terms of prediction accuracy, false positive rate, scheduling efficiency and scheduling time. The experimental work of GLP-LWRRSS Technique is performed for many instances with respect to various numbers of stream data task in order to analyze the performances. The effectiveness of GLP-LWRRSS Technique is compared with Predictive Scheduling [1] and dynamic assignment scheduling (DAS) algorithm [2] respectively.

RESULT AND DISCUSSIONS

Results and discussion of GLP-LWRRSS technique is explained with different parameters such as prediction accuracy, false positive rate, scheduling efficiency and scheduling time. The results of GLP-LWRRSS technique is compared with Predictive Scheduling [1] and dynamic assignment scheduling (DAS) algorithm [2]. The performance of GLP-LWRRSS technique is discussed with help of tables and graph values.

Impact of Prediction Accuracy

In GLP-LWRRSS Technique, Prediction accuracy 'PA' measures the ratio of number of data tasks (i.e. average processing time of data tasks) that are correctly predicted to the total number of stream data task considered as input. The prediction accuracy is estimated in terms of percentage (%) and mathematically obtained as,

$$PA = \frac{NP}{n} * 100 \tag{9}$$

From equation (9), 'n' denotes a total number of stream data task considered as input for performing experiential process whereas 'NP' indicates the number of data tasks correctly predicted. With help of equation (9), prediction accuracy of big stream data processing is measured with respect to a varied number of data task. While prediction accuracy is higher, the mechanism is said to be more effectual.

Sample calculation:

1. **Existing Predictive Scheduling:** average processing time of data tasks correctly predicted is 6 and the total number of stream data task is 10. Then the prediction accuracy is determined as follows,

$$PA = \frac{6}{10} * 100 = 60\%$$

2. **Existing DAS:** correctly predicted average processing time of data task is 7 and the total number of stream data task is 10. Then the prediction accuracy is calculated as follows, $PA = \frac{7}{10} * 100 = 70\%$

3. **Proposed GLP-LWRRSS**: average processing time of data tasks that correctly predicted is 8, and total number of stream data task is 10. Then the prediction accuracy is evaluated as,

$$PA = \frac{8}{10} * 100 = 80\%$$

To evaluate prediction accuracy of big stream data processing, GLP-LWRRSS Technique is implemented in Java Languages by considering a diverse number of data task in the range of 10-100. The experimental result of prediction accuracy using GLP-LWRRSS Technique is compared against with existing Predictive Scheduling [1] and DAS algorithm [2]. When employing 40 number of stream data task to conduct experimental process, GLP-LWRRSS Technique obtains 87 % prediction accuracy whereas existing Predictive Scheduling [1] and DAS algorithm [2] acquires 73 % and 78 % respectively. From that, it is expressive that the prediction accuracy using proposed prediction accuracy is higher as compared to other existing methods [1], [2]. The comparative result analysis of prediction accuracy is presented in below.

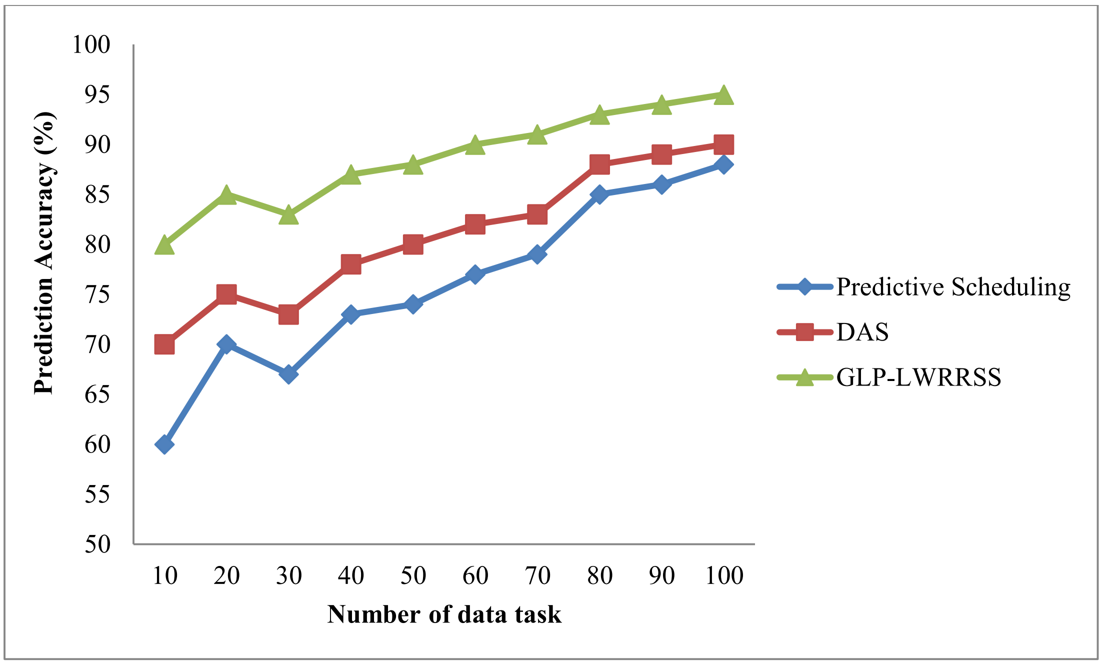

Figure 4. Experimental Result of Prediction Accuracy versus Number of Data Tasks

Figure 4 demonstrates the performance result analysis of prediction accuracy along with various numbers of data tasks in the range of 10-100 using three methods namely Predictive Scheduling [1] and DAS [2] and GLP-LWRRSS technique. As predicted in the figure 4, proposed GLP-LWRRSS technique provides higher prediction accuracy for big stream data processing when compared to existing works namely Predictive Scheduling [1] and DAS [2]. In addition to that, while increasing the number of data tasks for conducting experimental process, the prediction accuracy is also improved using all the three techniques. But comparatively, prediction accuracy using GLP-LWRRSS technique is higher than other existing works. This is owing to the application of generalized LASSO prediction process in proposed technique.

The generalized LASSO is applied in GLP-LWRRSS technique to lessen the number of features needed for efficient prediction through selecting more relevant features of PU. With help of optimized features, then generalized LASSO model significantly predicts average data processing time for each PU with minimal false positive rate. This helps for GLP-LWRRSS technique to attain enhanced prediction accuracy for big stream data processing. As a result, proposed GLP-LWRRSS technique increases the prediction accuracy by 18 % as compared to existing Predictive Scheduling [1] and 10 % as compared to existing DAS [2] respectively.

Impact of False Positive Rate

In GLP-LWRRSS Technique, False Positive Rate 'FPR' evaluates the ratio of a number of data tasks (i.e. average processing time of data task) that are incorrectly predicted to the total number of stream data task taken as input. The false positive rate is estimated in terms of percentage (%) and formulated as,

$$FPR = \frac{NIP}{n} * 100 \qquad (10)$$

From equation (10), 'n' refers number of stream data tasks where 'NIP' indicates a number of incorrectly predicted average processing time of data task. With help of above mathematical expression (10), the false positive rate of GLP-LWRRSS Technique is measured with respect to a different number of data task. While false positive rate of predictive scheduling is lower, the GLP-LWRRSS Technique is said to be more effective.

Sample calculation:

1. **Existing Predictive Scheduling:** average processing time of data tasks incorrectly predicted is 4 and the total number of stream data task is 10. Then the false positive rate is determined as follows,

$$\boldsymbol{FPR = \frac{4}{10} * 100 = 40\%}$$

2. **Existing DAS:** correctly predicted average processing time of data task is 3 and the total number of stream data task is 10. Then the false positive rate is calculated as follows,

$$\boldsymbol{FPR = \frac{3}{10} * 100 = 30\%}$$

3. **Proposed GLP-LWRRSS**: average processing time of data tasks that correctly predicted is 2, and total number of stream data task is 10. Then the false positive rate is evaluated as,

$$\boldsymbol{FPR = \frac{2}{10} * 100 = 20\%}$$

To measure false positive rate involved during prediction of average data processing time, GLP-LWRRSS Technique is implemented in Java Languages using a diverse number of data task in the range of 10-100. The experimental result of false positive rate using GLP-LWRRSS Technique is compared against with existing Predictive Scheduling [1] and DAS algorithm [2]. When considering 70 stream data tasks to accomplish experimental work, GLP-LWRRSS Technique attains 9 % false positive rate for prediction whereas existing Predictive Scheduling [1] and DAS algorithm [2] gets 21 % and 17 % respectively. From these results, it is significant that the false positive rate using GLP-LWRRSS Technique is lower as compared to other existing methods [1], [2]. The experimental result of false positive rate for predictive scheduling is illustrated in below Table 1.

Table 1. Performance Result of False Positive Rate

Number of data task	**False Positive Rate (%)**		
	Predictive Scheduling	**DAS**	**GLP-LWRRSS**
10	40	30	20
20	30	25	15
30	33	27	17
40	28	23	13
50	26	20	12
60	23	18	10
70	21	17	9
80	15	13	6
90	14	11	5
100	12	10	5

Table 1 reveals the comparative result analysis of false positive rate versus dissimilar numbers of data tasks in the range of 10-100 using three methods namely Predictive Scheduling [1] and DAS [2] and GLP-LWRRSS technique. As presented in the Table 1, proposed GLP-LWRRSS technique provides lower false positive rate for predicting average processing time of big stream data processing as compared to existing works namely Predictive Scheduling [1] and DAS [2]. Besides, while increasing the number of data tasks for experimental evaluation, the false positive rate is reduced using all the three techniques. This is due to the usage of generalized LASSO prediction process in GLP-LWRRSS technique.

With the application of generalized LASSO model, GLP-LWRRSS technique optimizes the number of features required for prediction by means of picking only more relevant features of PU. After features selection, the generalized LASSO model effectively predicts average data processing time for each PU with higher accuracy. This assists for GLP-LWRRSS technique to get lower false positive rate for big stream data processing. Thus, proposed GLP-LWRRSS technique decreases the false positive rate of prediction by 55 % as compared to existing Predictive Scheduling [1] and 44 % as compared to existing DAS [2] respectively.

Impact of Scheduling Time

In GLP-LWRRSS Technique, Scheduling Time (ST) determines amount of time taken for scheduling the data tasks to an appropriate PU. The scheduling time is measured in terms of milliseconds (ms) and mathematically obtained as,

$$ST = n * T(SD) \qquad (11)$$

From equation (11), scheduling time of stream data is measured with respect to diverse number of big stream data task processing. Here, 'n' point outs a number of stream data tasks considered for experimental evaluation and '$T(SD)$' indicates the time taken for scheduling one data task to a PU. When scheduling time of big stream data processing is lower, the technique is said to be more effectual.

Sample calculation:

1. **Existing Predictive Scheduling:** Number of stream data task considered is 10, and time for scheduling one data task is 0.22ms. Then the scheduling time is estimated as follows,

$$ST = 10 * 2.2 = 22ms$$

2. **Existing DAS:** Total number of data task is 10 and time for scheduling one data task is 2.9. Then the scheduling time is calculated as,

$$ST = 10 * 2.9 = 29ms$$

3. **Proposed GLP-LWRRSS:** Total number of data task is 10 and time for scheduling one data task is 3.3. Then the scheduling time is determined as,

$$ST = 0.33 * 100 = 33ms$$

To measure amount of time required for scheduling big stream data processing, GLP-LWRRSS Technique is implemented in Java Language using various numbers of data tasks in the range of 10-100. The experimental result of scheduling time using GLP-LWRRSS Technique is compared against with existing Predictive Scheduling [1] and DAS algorithm [2]. When considering 50 number of stream data tasks for the experimental work, GLP-LWRRSS Technique takes 60 ms scheduling time whereas existing Predictive Scheduling [1] and DAS algorithm [2] obtains 75 ms and 70 ms respectively. Thus, it is clear that the scheduling time using proposed GLP-LWRRSS Technique is lower when compared to other existing methods [1], [2]. The performance result analysis of scheduling time is demonstrated in below.

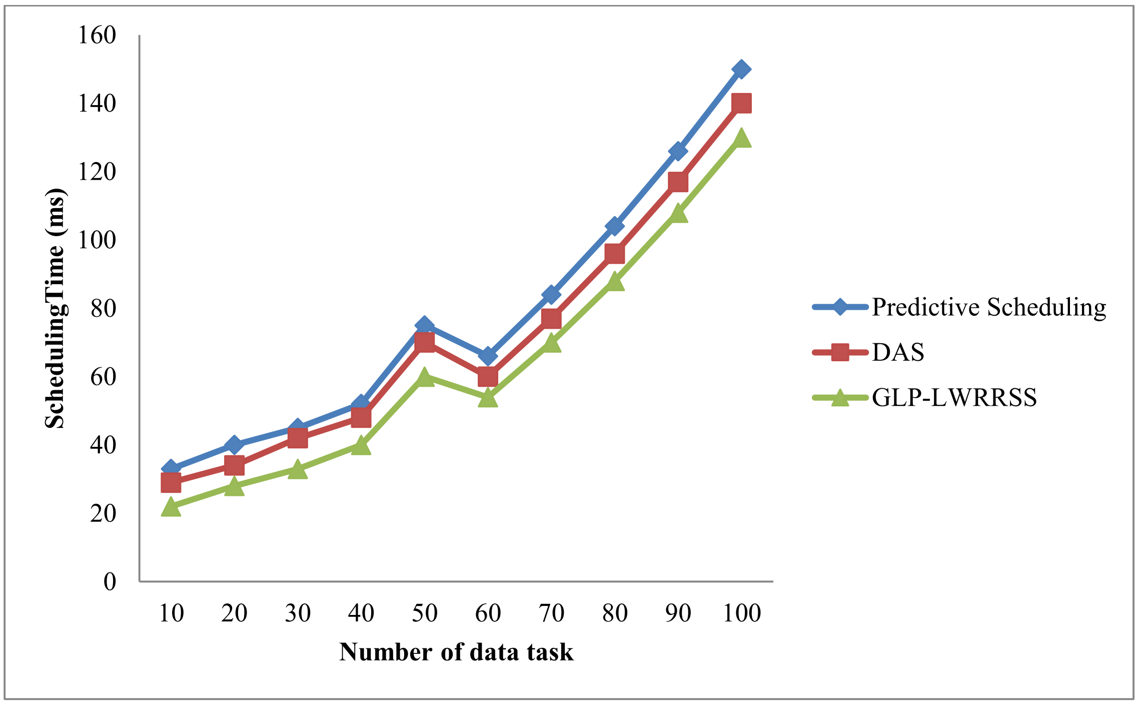

Figure 5. Experimental Result of Scheduling Time versus Number of Data Tasks

Figure 5 shows the comparative result of scheduling time versus dissimilar numbers of data tasks in the range of 10-100 using three methods namely Predictive Scheduling [1] and DAS [2] and GLP-LWRRSS technique. As illustrated in the figure 5, proposed GLP-LWRRSS technique provides lower scheduling time for big stream data processing as compared to existing works namely Predictive Scheduling [1] and DAS [2]. Furthermore, while increasing the number of data tasks for experimental process, the scheduling time is also improved using all the three techniques. But comparatively, scheduling time using GLP-LWRRSS technique is lower than other existing works. This is due to the processes of LWRRSS in proposed technique.

By using processes of LWRRSS algorithm, GLP-LWRRSS technique at first assign weight for all PU based on prediction results of average processing time for streaming of each big data processing. Then, GLP-LWRRSS technique finds higher weight processing unit in order to schedule data task with minimal amount of time utilization. This helps for GLP-LWRRSS technique to minimize the time complexity involved during process of big stream data scheduling. Hence, proposed GLP-LWRRSS technique reduces the scheduling time by 21 %

as compared to existing Predictive Scheduling [1] and 14 % as compared to existing DAS [2] respectively.

Impact of Scheduling Efficiency

In GLP-LWRRSS Technique, Scheduling Efficiency '(SE)' is measured as the ratio of number of data tasks that are correctly scheduled to appropriate PUs to the total number of stream data tasks as input. The scheduling efficiency is evaluated in terms of percentages (%) and determined as,

$$SE = \frac{NDS_{PUs}}{n} * 100 \quad (12)$$

From equation (12), scheduling efficiency of big stream data processing is estimated with respect to various number of stream data. Here, 'n' indicates a number of stream data task whereas 'NDS_{PUs}' denotes the number of data tasks that are correctly scheduled to appropriate PUs. When scheduling efficiency of big stream data processing is higher, the technique is said to be more effective.

Sample calculation:

1. **Existing Predictive Scheduling:** Number of data task that are correctly scheduled is 7, and the total number of stream data task is 10. Then the scheduling efficiency is computed as follows,

$$\boldsymbol{SE = \frac{7}{10} * 100 = 70\%}$$

2. **Existing DAS:** Number of data tasks correctly scheduled is 8, and the total number of stream data tasks is 10. Then the scheduling efficiency is estimated as follows,

$$\boldsymbol{SE = \frac{8}{10} * 100 = 80\%}$$

3. **Proposed GLP-LWRRSS**: Number of data task correctly scheduled is 9, and the total number of data tasks is 10. Then the scheduling efficiency is determined as,

$$\boldsymbol{SE = \frac{9}{10} * 100 = 90\%}$$

The GLP-LWRRSS Technique is implemented in Java Languages by assuming varied numbers of data tasks in the range of 10-100 to determine scheduling efficiency of big stream data processing. The result of scheduling efficiency using GLP-LWRRSS Technique is compared against with existing Predictive Scheduling [1] and DAS algorithm [2]. While considering 60 stream data tasks for experimental work, GLP-LWRRSS Technique gets 93 % scheduling efficiency whereas existing Predictive Scheduling [1] and DAS algorithm [2] acquires 73 % and 83 % respectively. From these results, it is descriptive that the scheduling efficiency of big stream data processing using proposed GLP-LWRRSS Technique is higher when compared to other traditional works [1], [2]. The comparative result of scheduling efficiency using three methods with respect to a varied number of stream data task is depicted in below Table 2.

Table 2. Performance Result of Scheduling Efficiency

Number of data task	**Scheduling Efficiency (%)**		
	Predictive Scheduling	**DAS**	**GLP-LWRRSS**
10	70	80	90
20	71	81	91
30	73	83	93
40	72.5	82.5	92.5
50	74	84	94
60	73	83	93
70	74	84	94
80	75	85	95
90	77	87	97
100	78	88	98

Table 2 portrays the impact of scheduling efficiency versus different numbers of data tasks in the range of 10-100 using three methods namely Predictive Scheduling [1] and DAS [2] and GLP-LWRRSS technique. As exposed in the table 2, proposed GLP-LWRRSS technique provides higher scheduling efficiency for big stream data processing as compared to existing works namely Predictive Scheduling [1] and DAS [2]. Moreover, while increasing the number of data tasks for carried outing experimental work, the scheduling efficiency is also improved using all the three techniques. But comparatively, scheduling efficiency using GLP-LWRRSS technique is higher than other existing works. This is because of the usage of LWRRSS in proposed technique.

With the algorithmic processes of LWRRSS, GLP-LWRRSS technique determines weight for all PU according to prediction results of average processing time for each stream data processing. Consequently, GLP-LWRRSS technique chooses higher weight processing unit for scheduling data task with higher accuracy. The selected higher weight PU takes minimum amount of average data processing time as compared to other PUs. This helps for GLP-LWRRSS technique to quickly carry out big stream data processing as compared to existing works. This supports for GLP-LWRRSS technique to achieve better scheduling efficiency for big stream data. As a result, proposed GLP-LWRRSS technique increases the scheduling efficiency by 27 % as compared to existing Predictive Scheduling [1] and 12 % as compared to existing DAS [2] respectively.

CONCLUSION

An effective GLP-LWRRSS technique is developed with key goal of enhancing the performance of predictive scheduling for big stream data processing with higher efficiency and lower time. The goal of GLP-LWRRSS technique is attained by application of generalized LASSO prediction model and LWRRSS algorithmic processes. The generalized LASSO prediction model applied in GLP-LWRRSS technique helps GLP-LWRRSS technique to get improved performance for predicting average data processing time of big stream data processing. Thus, GLP-LWRRSS technique gets improved prediction accuracy and lower false positive rate for efficiently predictive scheduling process as compared to conventional techniques. Furthermore, LWRRSS algorithm designed in GLP-LWRRSS technique seamlessly schedule each stream data to PU with lower data processing time to complete stream data task (i.e. appropriate PU) with minimum time. Therefore, GLP-LWRRSS technique provides better performance in terms of scheduling efficiency and scheduling time for big stream data processing as compared to state-of-the-art works. The performance of GLP-LWRRSS technique is tested with the metrics such as prediction accuracy, false positive rate, scheduling efficiency and scheduling time and compared with state-of-the-art works. With the experimental conducted for GLP-LWRRSS technique, it is significant that the scheduling efficiency affords more accurate results for big data stream processing as compared to existing methods.

REFERENCES

[1] Teng Li, Jian Tang and Jielong Xu, "Performance Modeling and Predictive Scheduling for Distributed Stream Data Processing", IEEE Transactions on Big Data, Volume 2, Issue 4, Pages 353 – 364, December 2016

[2] Yan Liu, Kun Wang, Yue Yu, Jin Qi, Yanfei Sun, "A dynamic assignment scheduling algorithm for big data stream processing in mobile Internet services", Personal and Ubiquitous Computing, Springer, Volume 20, Pages 373–383, 2016

[3] Teng Li, Jian Tang and Jielong Xu, "A Predictive Scheduling Framework for Fast and Distributed Stream Data Processing", IEEE International Conference on Big Data, Pages 1-6, 2015,

[4] Karim Kanoun, Cem Tekin, David Atienza, and Mihaela van der Schaar, "Big-Data Streaming Applications Scheduling based on Staged Multi-armed Bandits", IEEE Transactions on Computers, Volume 65, Issue 12, Pages 3591 – 3605, December 2016

[5] Dawei Sun, Guangyan Zhang, Songlin Yang, Weimin Zheng, Samee U. Khan, Keqin Li, "Re-Stream: Real-time and energy-efficient resource scheduling in big data stream computing environments", Information Sciences, Elsevier, Volume 319, Pages 92-112, 2015

[6] Adnan Akbar, Abdullah Khan, Francois Carrez, Klaus Moessner, "Predictive Analytics for Complex IoT Data Streams", IEEE Internet of Things Journal, Volume 4, Issue 5, Pages 1571 – 1582, 2017

[7] Dawei Sun, Rui Huang, "A Stable Online Scheduling Strategy for Real-Time Stream Computing Over Fluctuating Big Data Streams", IEEE Access, Volume 4, Pages 8593 – 8607, 2016

[8] Qingchen Zhang, Zhikui Chen and Laurence T. Yang, "A nodes scheduling model based on Markov chain prediction for big streaming data analysis", International Journal of Communication Systems, Pages 1-10, 2014

[9] Ranjitha P, "Streaming Analytics over Real-Time Big Data", Global Journal of Computer Science and Technology: C Software & Data Engineering, Volume 15, Issue 5, Pages 1-7, 2015

[10] Xue-QiangZeng, Guo-ZhengLi, "Incremental partial least squares analysis of big streaming data", Pattern Recognition, Elsevier, Volume 47, Issue 11, Pages 3726-3735, November 2014

[11] Pavel Smirnov, Mikhail Melnik and Denis Nasonov, "Performance-aware scheduling of streaming applications using genetic algorithm", Procedia Computer Science, Elsevier, Volume 108C, Pages 2240–2249, 2017

[12] Navroop Kaur, Sandeep K. Sood, "Dynamic resource allocation for big data streams based on data characteristics (5Vs)", International Journal of Network Management, Volume 27, Issue 4, Pages 1-16, 2017

[13] Yuan Liu, Xuanhua Shi and Hai Jin, "Runtime-aware adaptive scheduling in stream processing", Concurrency and Computation: Practice and Experience, Wiley Online Library , Pages 1-14, 2015

[14] Gulisano, Vincenzo; Jimenez-Peris, Ricardo; Patino-Martinez, Marta; Soriente, Claudio; Valduriez, Patrick, "StreamCloud: An Elastic and Scalable Data Streaming System", IEEE Transactions on Parallel and Distributed Systems, Volume 23, Issue 12, Pages 2351-2365, 2012

[15] Karim Kanoun, Mihaela van der Schaar, "Big-data streaming applications scheduling with online learning and concept drift detection", Design, Automation & Test in Europe Conference & Exhibition (DATE), Pages 1547-1550, 2015

[16] Bolla Saikiran, Kolla Morarjee, "An Efficient Algorithm for Update Scheduling in Streaming Data Warehouses", International Journal of Computer Science and Information Technologies, Volume 5, Issue 2, Pages 1082-1085, 2014

[17] Marcos Dias de Assunçao, Alexandre da Silva Veith, Rajkumar Buyya, "Distributed data stream processing and edge computing: A survey on resource elasticity

and future directions", Journal of Network and Computer Applications, Elsevier, Volume 103, Pages 1–17, 2018

[18] Gabriele Mencagli, "A Game-Theoretic Approach for Elastic Distributed Data Stream Processing", ACM Transactions on Autonomous and Adaptive Systems, Volume 9, Issue 4, Pages 1-34, 2015

[19] Gedik, Bugra, Schneider, Scott, Hirzel, Martin, Wu, Kun-Lung, "Elastic Scaling for Data Stream Processing", IEEE Transactions on Parallel and Distributed Systems, Volume 25 Issue 6, Pages 1447-1463, 2014

[20] Pavel A.Smirnov, Denis Nasonov, "Quality-based Workload Scaling for Real-time Streaming Systems", Procedia Computer Science, Elsevier, Volume 101, Pages 323-332, 2016

[21] Apache Storm: http://storm.apache.org/