Received October 24, 2021, accepted November 12, 2021, date of publication November 22, 2021, date of current version December 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3129885

Adaptive Trust Management and Data Process Time Optimization for Real-Time Spark Big Data Systems

SEUNGWOO SEO AND JONG-MOON CHUNG, (Senior Member, IEEE)
School of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Republic of Korea

Corresponding author: Jong-Moon Chung (jmc@yonsei.ac.kr)

This work was supported by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) support Program through the Institute for Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2021-2018-0-01799.

ABSTRACT Applications supporting businesses, smart systems, social networks, and advanced video applications such as eXtended Reality (XR) require large amounts of data processing to be provided in real-time. Therefore, the processing speed of big data systems is more important than ever. On the other hand, protecting a big data system is not easy, as various types of nodes and clusters are supported by various wired and wireless networks. Commonly security procedures slow down the response time of big data networks, and therefore, enhanced security and performance speed techniques need to be co-designed into the system. In this paper, a trusted streaming adaptive failure-compensation (TSAF) scheme is proposed that uses a trust management scheme to identify malicious nodes in Spark big data systems, exclude them from job/task processing, and calculate the number of nodes that can satisfy the process's object completion time. The TSAF scheme shows an improved processing performance when there are attacks on the big data system compared to other existing real-time big data processing schemes. For the case of no security attack, the results show that the processing time of TSAF is faster by about $1 \sim 2\%$ compared to the existing big data processing schemes when the process completion object time is set to 0.5 s. Even when the ratio of malicious nodes performing security attacks on worker nodes reaches 0.5, the results show that TSAF can satisfy over 75% of the tasks within the object time, which is significantly higher compared to the existing big data processing schemes.

INDEX TERMS Trust management, big data, time bound optimization, real-time processing, security.

I. INTRODUCTION

Considering the significant increase of internet of things (IoT) devices, drones, autonomous driving vehicles, and various interactive video services and systems, such as eXtended Reality (XR), the amount of data that needs to be processed in real-time is rapidly increasing. Because big data systems need to collect massive amounts of structured and unstructured data, big data systems are interconnected with many diverse networks, servers, blockchain databases, and clouds. As there are many interfaces to a big data system network, the possibility of a security attack is very high.

The associate editor coordinating the review of this manuscript and approving it for publication was Tai-hoon Kim.

To protect a system in which various types of nodes and wired/wireless networks are combined, various security methods have been proposed. Among them, trust management technology has been very effective. Trust management can help maintain the security level by granting trust to members based on previous activities. Trust management technology has shown good protection results in systems where heterogeneous nodes and networks were used.

Several methods have been proposed to utilize trust management in big data systems. In [1]–[4], the security of big data systems are maintained by calculating the trust for data collected from the IoT sensors, unmanned aerial vehicles (UAVs), autonomous driving vehicles, and IIoT network. In [5]–[7], algorithms to manage user or data trust levels using blockchains and authentication were proposed.

However, in these studies, trust of the worker nodes for big data processing is not considered. In [8], when big data is processed on the cloud platform, a trust level is assigned to the data, task, and big data processing resources. The scheme proposed in [8] maintains the security level of the data by allowing tasks to be processed differently according to the assigned trust level. In [9], when multiple clouds are connected to a big data system, trust of the corresponding clouds are determined, and an example of parallel big data processing is suggested. However, in this study, the cloud to which worker nodes are connected is not adapted in reference to the security situation, so the scheme may be vulnerable to varying attacks and have issues in computational complexity.

In this paper, a trusted streaming adaptive failure-compensation (TSAF) scheme is proposed that uses trust management to identify malicious nodes in Spark big data systems, exclude them from job/task processing assignments, and calculate the number of nodes that can satisfy the processing object time.

The contributions of TSAF are as follows.

- TSAF is a novel trust-based security technique for Apache Spark big data processing systems that can satisfy predefined object times even when the ratio of malicious nodes performing security attacks on worker nodes is very high.
- TSAF reduces the recovery time caused by malicious nodes being assigned as worker nodes along with errors that occur during the communication and execution process considering past records.
- TSAF minimizes the computational complexity of determining the trust level of worker nodes through clustering. This approach focuses on the concept that parallel clustering based on divided big data nodes can enable faster big data processing.

In section II, the big data processing system model for TSAF is described. In section III, the TSAF scheme is proposed, where details on the trust management process and data processing time estimation are provided along with the trust management scheme's computation complexity analysis. The operational flowchart of TSAF is described in section IV. Furthermore, the performance of TSAF and other real-time big data processing schemes are compared through simulation in section V and the conclusion of this paper is provided in section VI.

II. SYSTEM MODEL

The big data system and network model investigated in this paper is described in Fig. 1. It is assumed that the worker nodes supporting the big data process are connected to a cloud. To process real-time streaming big data, the big data system has one master node and several worker nodes. Worker nodes are divided into several clusters and managed, and each cluster has one cluster head (CH). The master node communicates directly with the user and distributes the data coming into the system to each cluster. The CH performs the role of the master node in the cluster and also the role of the worker node at the same time. The system processes the streaming data in parallel using multiple clusters to satisfy the object time T_{object}, and the result is reported to the user through the CH and master node [10].

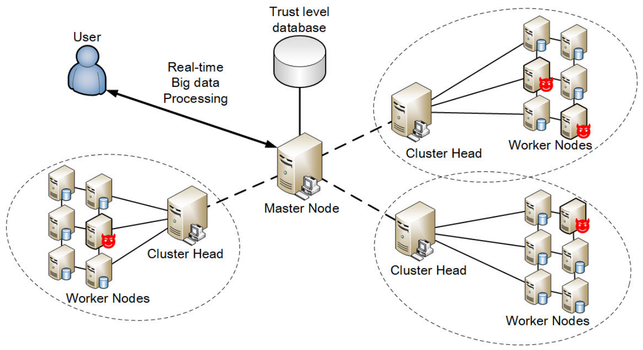


FIGURE 1. Real-time big data processing system model of TSAF, which includes one master node that manages several worker nodes and is assisted by a trust level database. Worker nodes can be divided into multiple clusters and each cluster is managed by a cluster head.

Malicious nodes also exist among the worker nodes of the cloud system. Malicious nodes intentionally degrade the system performance inside the cloud (e.g., data export, no communication response, reduction in data processing speed, etc.).

The master node maintains a separate trust level database to manage the trust evaluation of all worker nodes. The trust level database stores the trust-related information of the existing worker nodes and verifies the trust information of the worker nodes reported by the CH based on synthesizing the trust level data (in the database) and the trust information reported by the CHs. In addition, when it is necessary to adjust the clusters (due to a changes in the number of worker nodes), the trust level database is used to designate nodes with the highest trust level to serve as CHs.

III. TRUSTED STREAMING ADAPTIVE FAILURE-COMPENSATION

The TSAF scheme is divided into two parts. In part 1, calculation of the optimal number of clusters is conducted while considering the trust calculation of the worker nodes in the system and the computational complexity. In part 2, the scheme calculates the number of worker nodes required to satisfy the object time based on the cluster and honest worker nodes derived from the trust management process.

The trust management process (using the optimal cluster size) and the real-time big data process (that satisfies the object time) are performed simultaneously while the system is running.

A. TRUST MANAGEMENT PROCESS

GlobalTrust [11], [12] was used to calculate the trust level of the worker nodes, where all worker nodes report a local trust opinion (LTO) to the cluster head (CH) for trust evaluation. The reported behavior is observed based on the big data

TABLE 1. Variables and expressions for the trust management process.

Variables	Expressions
LTO	Local trust opinion
p	Number of positive events
n	Number of negative events
SR	Subjective reputation
S_u	Group of worker nodes that submit LTOs of node u
HR	Hierarchical rank of nodes
$sim(w, j)$	Cosine similarity of node w and node j
D	Trusted quorum
BR	Behavior reputation
CR	Credibility reputation
GR	Global reputation
ρ	Coefficient of GR
N	Total number of worker nodes in the data processing system
k	Number of clusters

processing time for a certain period. TABLE 1 summarizes the variables and expressions used in the trust management process. $LTO_{\mathrm{j,k}}$ is the LTO measured by worker node j for worker node k, which is calculated as

$$LTO_{j,k} = \frac{p_{j,k}}{p_{j,k} + n_{j,k}} \tag{1}$$

where p is the number of times that data is normally processed as a positive event, n counts negative events, which increases whenever an abnormal behavior (e.g., exceeding the object time required for processing or an error occurs) in the data communication process occurs.

Based on the collected LTOs, the CH calculates the trust of the nodes in the cluster. First, the subjective reputation (SR) is calculated from

$$SR_{w,u} = \sum_{j \in S_u} LTO_{j,u} \frac{HR_j sim(w, j)}{\sum_{j \in S_u} HR_j sim(w, j)} \tag{2}$$

where $sim(w, j)$ is the cosine similarity of the LTO reported by worker nodes w and j. HR_j represents the hierarchical rank of node j in which non-CH worker nodes are 1, and CH nodes have a higher value.

Based on the created SR, the CH creates a trusted quorum (D). Using the SR information of the nodes belonging to D, the behavior reputation (BR) for each node is calculated using (3).

$$BR_{\mathrm{CH}_i,u} = \frac{\sum_{w \in D} SR_{w,u}}{|D_{\mathrm{CH}_i}|} \tag{3}$$

Using the generated BR, the credibility reputation (CR) can be obtained using (4). CR measures the similarity between the LTO reported by a specific node u and the calculated BR, where the more similar the opinion is, a higher CR value is assigned

$$CR_{\mathrm{CH}_i,u} = 1 - \sqrt{\frac{\sum_{j \in \mathbb{N}} (LTO_{u,j} - BR_{CH_i,j})^2}{|j \in \mathbb{N}|}} \tag{4}$$

where $\mathbb{N}$ is a group of nodes that $LTO_{u,j} \neq null$.

The global reputation (GR) can be obtained from the BR and CR using

$$GR_{\mathrm{CH}_i,u} = \rho BR_{\mathrm{CH}_i,u} + (1 - \rho) CR_{\mathrm{CH}_i,u} \tag{5}$$

where ρ is a value between [0,1].

The trust information of the clusters is reported to the master node. The master node verifies the trust information of each cluster and performs its role such as excluding potentially malicious nodes from the data processing.

B. TRUST MANAGEMENT SCHEME ANALYSIS

When performing trust management, TSAF calculates the number of clusters that minimizes the time it takes to calculate the trust level. *Lemma 1* calculates the computational complexity of the trust management process, and *Lemma 2* calculates the number of clusters k_{opt} that minimizes the calculated computational complexity, which will effectively minimize the time consumed in the trust calculation process.

Lemma 1: The computational complexity of calculating the trust of the entire big data system is $O(\lceil N/k \rceil^3 + k^3)$. □

Proof: When calculating the trust of worker nodes, the master node divides all N worker nodes into k clusters. At this time, there will be a maximum of $\lceil N/k \rceil$ worker nodes in at least one cluster (where $\lceil x \rceil$ refers to an integer greater than or equal to x).

The computational complexity inside the cluster is derived as follows. The trust scheme used in this paper creates a trust quorum to judge the trust levels [11], [12]. When all clusters simultaneously compute trust values, the computational complexity of the cluster with the largest number of worker nodes will be $O(\lceil N/k \rceil^2)$. Since the accuracy of all trust values inside the cluster is determined through the created trusted quorum, the computational complexity inside the cluster will be $O(\lceil N/k \rceil^2 \times \lceil N/k \rceil) = O(\lceil N/k \rceil^3)$

The master node performs the accuracy judgment on the k trust values reported from the CH once more as if it was performed inside the cluster, and the computational complexity at this time is $O(k^3)$. Therefore, the overall computational complexity is $O(\lceil N/k \rceil^3 + k^3)$. ■

Lemma 2: The number of optimal clusters k_{opt} that minimizes the computational complexity is $nint(\sqrt{N})$. □

Proof: According to *Lemma 1*, the trust computational complexity of TSAF is $O(\lceil N/k \rceil^3 + k^3)$. The value k' that minimizes computational complexity can be obtained as follows.

$$\begin{aligned} k' &= \arg_k \left[\frac{\partial}{\partial k} O\left(\left\lceil \frac{N}{K} \right\rceil^3 + k^3 \right) = 0 \right] \\ &= \arg_k \left[-\frac{3N^3}{k^4} + 3k^2 = 0 \right] \end{aligned} \tag{6}$$

The k' values that satisfies (6) is $\pm\sqrt{N}$. Since $N > 0$, k must be greater than 0, so $k' = \sqrt{N}$ minimizes the computational complexity.

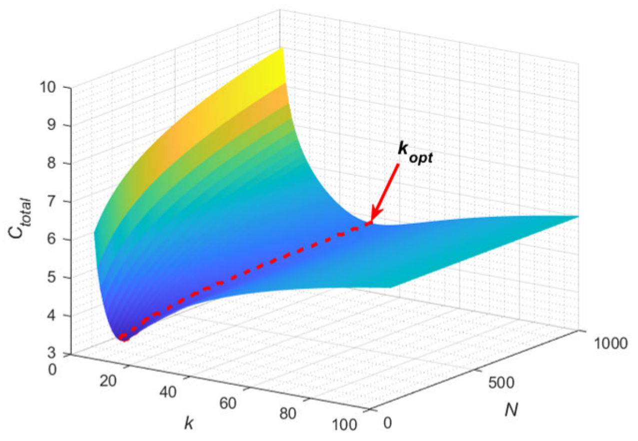


FIGURE 2. Computational complexity (C_{total}) graph based on number of worker nodes (N) and clusters (k). For a given N value, C_{total} decreases from $k = 1$ to $k = k_{\text{opt}}$ and increases steadily after $k = k_{\text{opt}}$.

The second derivative of the computation complexity equation is presented in (7).

$$\frac{\partial^2}{\partial k^2} O\left(\left\lceil \frac{N}{k} \right\rceil^3 + k^3\right) = \frac{12N^3}{k^5} + 6k \tag{7}$$

Since $N > 0$ and $k > 0$, the second derivative of the computational complexity is always greater than 0, so it is a convex function.

Since the number of clusters k is a positive integer, the number of clusters that minimizes the computational complexity is $k_{\text{opt}} = nint(\sqrt{N})$ (where $nint(x)$ is the nearest integer to x). ■

Fig. 2 is a graph showing the change of computational complexity according to the change of k and N. At the same k, the computational complexity increases as N increases. At the same N, the computational complexity continues to decrease until k reaches k_{opt}, and has a minimum value at k_{opt}. After passing k_{opt}, the computational complexity continues to increase.

C. DATA PROCESSING TIME

Based on the previously calculated number of optimal clusters k_{opt}, the master node divides the worker nodes into k_{opt} clusters to process the data. The big data processing time (T_{P}) can be divided into two parts. The first is the time consumed by the master node to transmit data to the CHs (T_{m}), and the second is the time (T_{c}^k) to process the data inside the k clusters. TABLE 2 summarizes the variable and expressions used in the data processing time analysis.

$$T_{\text{P}} = T_{\text{m}} + T_{\text{c}}^k \tag{8}$$

The master node transmits the real-time data to be processed to each CH. The transmission time (T_{m}) is

$$\begin{aligned} T_{\text{m}} &= T_{\text{vs}}^{\text{m}} + T_{\text{comm}}^{\text{m}} + T_{\text{add}}^{\text{m}} \\ &= kiA + \frac{Bs}{k} + P_{e_{\text{comm}}} \frac{Bs}{k} \end{aligned} \tag{9}$$

TABLE 2. Variables used in the data processing time analysis.

Variables	Expressions
k	Number of clusters
i	Number of iterations
s	Input data size
n	Number of worker nodes
n_k	Number of worker nodes in cluster k
N^{H}	Number of honest worker nodes
N^{M}	Number of malicious worker nodes
T_{p}	Big data processing time
T_{m}	Transmission time from master node to CHs
T_{vs}^{m}	Variable sharing time from master node to CHs
$T_{\text{comm}}^{\text{m}}$	Data communication time from master node to CHs
$T_{\text{add}}^{\text{m}}$	Additional time in transmission caused by an error
$T_{\text{vs}}^{\text{baseline}}$	Baseline time of variable sharing
$T_{\text{comm}}^{\text{baseline}}$	Baseline time of communication
θ_{vs}	Coefficient of the baseline time in variable sharing
θ_{comm}	Coefficient of the baseline time in communication
T_{c}^{k}	Data processing time in cluster k
T_{vs}^{k}	Variable sharing time from CH to worker nodes
$T_{\text{comm}}^{\text{k}}$	Data communication time from CH to worker nodes
$T_{\text{exec}}^{\text{k}}$	Execution time on worker nodes
$T_{\text{add}}^{\text{k}}$	Additional time in data processing on cluster k caused by an error
$P_{e_{\text{comm}}}$	Error rate of the communication phase
$P_{e_{\text{exec}}}$	Error rate of the execution phase
T_{object}	Object time to process the data
M_a	Unit size data processing time (based on the application characteristics)

where $A = \theta_{\text{vs}} T_{\text{vs}}^{\text{baseline}}$, $B = \theta_{\text{comm}} T_{\text{comm}}^{\text{baseline}}$, T_{vs}^{m} is the variable sharing time, $T_{\text{comm}}^{\text{m}}$ is the data communication time, and T^{baseline} is the time it takes to transmit unit-sized variables and data. In the variable sharing process, variables necessary for data processing are transmitted to k CHs. In the data communication process, data is transmitted as much as the ratio of honest worker nodes that belong to each cluster among all honest worker nodes. $T_{\text{add}}^{\text{m}}$ is the time it takes to recover a communication error that occurs when data is transmitted to the cluster and is restored by retransmitting the data.

The CH distributes the received data to honest worker nodes. The time it takes to process data in cluster k (T_{c}^k) is

$$T_{\text{c}}^k = T_{\text{vs}}^k + T_{\text{comm}}^k + T_{\text{exec}}^k + T_{\text{add}}^k \tag{10}$$

where T_{vs}^k is the variable sharing time, T_{comm}^k is the data communication time, T_{exec}^k is the execution time, and T_{add}^k is the recovery time of errors that occurred during data processing. Errors can occur in the communication process and data processing inside the cluster. The data processing time of cluster k (T_{c}^k) with n_k worker nodes, that also considers the recovery time caused by an error, can be expressed as follows.

$$\begin{aligned} T_{\text{c}}^k &= T_{\text{vs}}^k + T_{\text{comm}}^k + T_{\text{exec}}^k + T_{\text{add}}^k \\ &= n_k iA + \frac{Bs\frac{n_k}{n}}{n_k} + \frac{iM_a\frac{n_k}{n}}{n_k} + P_{e_{\text{comm}}} \frac{Bs\frac{n_k}{n}}{n_k} \\ &\quad + P_{e_{\text{exec}}} \left(n_k iA + \frac{Bs\frac{n_k}{n}}{n_k n_k} + \frac{iM_a\frac{n_k}{n}}{n_k n_k} \right) \end{aligned}$$

$$= n_k iA + \frac{Bs}{n} + \frac{iM_a}{n} + P_{e_{\text{comm}}}\frac{Bs}{n} + P_{e_{\text{exec}}}\left(n_k iA + \frac{Bs}{nn_k} + \frac{iM_a}{nn_k}\right) \quad (11)$$

The total big data processing time (T_{P}) is as follows.

$$\begin{aligned} T_{\text{P}} &= T_{\text{m}} + T_{\text{c}}^k \\ &= \left(T_{\text{vs}}^{\text{m}} + T_{\text{comm}}^{\text{m}} + T_{\text{add}}^{\text{m}}\right) + \left(T_{\text{vs}}^k + T_{\text{comm}}^k + T_{\text{exec}}^k + T_{\text{add}}^k\right) \\ &= kiA + \frac{Bs}{k} + P_{e_{\text{comm}}}\frac{Bs}{k} + n_k iA + \frac{Bs}{n} + \frac{iM_a}{n} \\ &\quad + P_{e_{\text{comm}}}\frac{Bs}{n} + P_{e_{\text{exec}}}\left(n_k iA + \frac{Bs}{nn_k} + \frac{iM_a}{nn_k}\right) \end{aligned} \quad (12)$$

Lemma 3: the optimal number of worker nodes N_{opt} is $N_{\text{opt}} = \{\lceil \min(n_i) \rceil \,|n_i > 0, n_i \in \mathbb{R}, i = 1, 2, 3\}$. □

Proof: The number of worker nodes to process the data within the object time can be obtained by deriving the n that satisfies (13).

$$T_{\text{P}} = T_{\text{m}} + T_{\text{c}}^k \leq T_{\text{object}} \quad (13)$$

$$\begin{aligned} T_{\text{P}} &= \left(T_{\text{vs}}^{\text{m}} + T_{\text{comm}}^{\text{m}} + T_{\text{add}}^{\text{m}}\right) + \left(T_{\text{vs}}^k + T_{\text{comm}}^k + T_{\text{exec}}^k + T_{\text{add}}^k\right) \\ &= kiA + \frac{Bs}{k} + P_{e_{\text{comm}}}\frac{Bs}{k} + n_k iA + \frac{Bs}{n} + \frac{iM_a}{n} + P_{e_{\text{comm}}}\frac{Bs}{n} \\ &\quad + P_{e_{\text{exec}}}\left(n_k iA + \frac{Bs}{nn_k} + \frac{iM_a}{nn_k}\right) \leq T_{\text{object}} \end{aligned} \quad (14)$$

Using $n_k \simeq n/k$ and rearrangeing for n, the following can be obtained

$$\begin{aligned} &n\left(\frac{1}{k}\left(1 + P_{e_{\text{exec}}}\right)iA\right) + \left(kiA + \frac{Bs}{k} + P_{e_{\text{comm}}}\frac{Bs}{k} - T_{\text{object}}\right) \\ &\quad + \frac{1}{n}\left(Bs + iM_a + P_{e_{\text{comm}}}Bs\right) \\ &\quad + \frac{1}{n^2}\left(kP_{e_{\text{exec}}}\left(Bs + iM_a\right)\right) \leq 0 \end{aligned} \quad (15)$$

where n is the number of worker nodes, so $n > 0$. In addition,

$$\begin{aligned} &n^3\left(\frac{1}{k}\left(1 + P_{e_{\text{exec}}}\right)iA\right) + n^2\left(kiA + \frac{Bs}{k} + P_{e_{\text{comm}}}\frac{Bs}{k} - T_{\text{object}}\right) \\ &\quad + n\left(Bs + iM_a + P_{e_{\text{comm}}}Bs\right) + kP_{e_{\text{exec}}}\left(Bs + iM_a\right) \\ &= xn^3 + yn^2 + zn + w \leq 0 \end{aligned} \quad (16)$$

where $x = \frac{1}{k}\left(1 + P_{e_{\text{exec}}}\right)iA$, $y = kiA + \frac{Bs}{k} + P_{e_{\text{comm}}}\frac{Bs}{k} - T_{\text{object}}$, $z = Bs + iM_a + P_{e_{\text{comm}}}Bs$, $w = kP_{e_{\text{exec}}}\left(Bs + iM_a\right)$. The solutions of the above cubic equation are n_1, n_2, and n_3, which are given below

$$\begin{aligned} n_1 &= -\frac{y}{3x} - \frac{1}{3x}\left[\frac{\Psi + \left\{\Psi^2 - 4(y^2 - 3xz)^3\right\}^{\frac{1}{2}}}{2}\right]^{\frac{1}{3}} \\ &\quad - \frac{1}{3x}\left[\frac{\Psi - \left\{\Psi^2 - 4(y^2 - 3xz)^3\right\}^{\frac{1}{2}}}{2}\right]^{\frac{1}{3}} \end{aligned} \quad (17)$$

$$\begin{aligned} n_2 &= -\frac{y}{3x} + \frac{1 + i\sqrt{3}}{6x}\left[\frac{\Psi + \left\{\Psi^2 - 4(y^2 - 3xz)^3\right\}^{\frac{1}{2}}}{2}\right]^{\frac{1}{3}} \\ &\quad + \frac{1 - i\sqrt{3}}{6x}\left[\frac{\Psi - \left\{\Psi^2 - 4(y^2 - 3xz)^3\right\}^{\frac{1}{2}}}{2}\right]^{\frac{1}{3}} \end{aligned} \quad (18)$$

$$\begin{aligned} n_3 &= -\frac{y}{3x} + \frac{1 - i\sqrt{3}}{6x}\left[\frac{\Psi + \left\{\Psi^2 - 4(y^2 - 3xz)^3\right\}^{\frac{1}{2}}}{2}\right]^{\frac{1}{3}} \\ &\quad + \frac{1 + i\sqrt{3}}{6x}\left[\frac{\Psi - \left\{\Psi^2 - 4(y^2 - 3xz)^3\right\}^{\frac{1}{2}}}{2}\right]^{\frac{1}{3}} \end{aligned} \quad (19)$$

where $\Psi = 2y^3 - 9xyz + 27x^2w$.

The optimal number of worker nodes N_{opt} is the smallest real number among n_1, n_2, and n_3, therefore,

$$N_{\text{opt}} = \{\lceil \min(n_i) \rceil \,|n_i > 0, n_i \in \mathbb{R}, i = 1, 2, 3\} \quad (20)$$

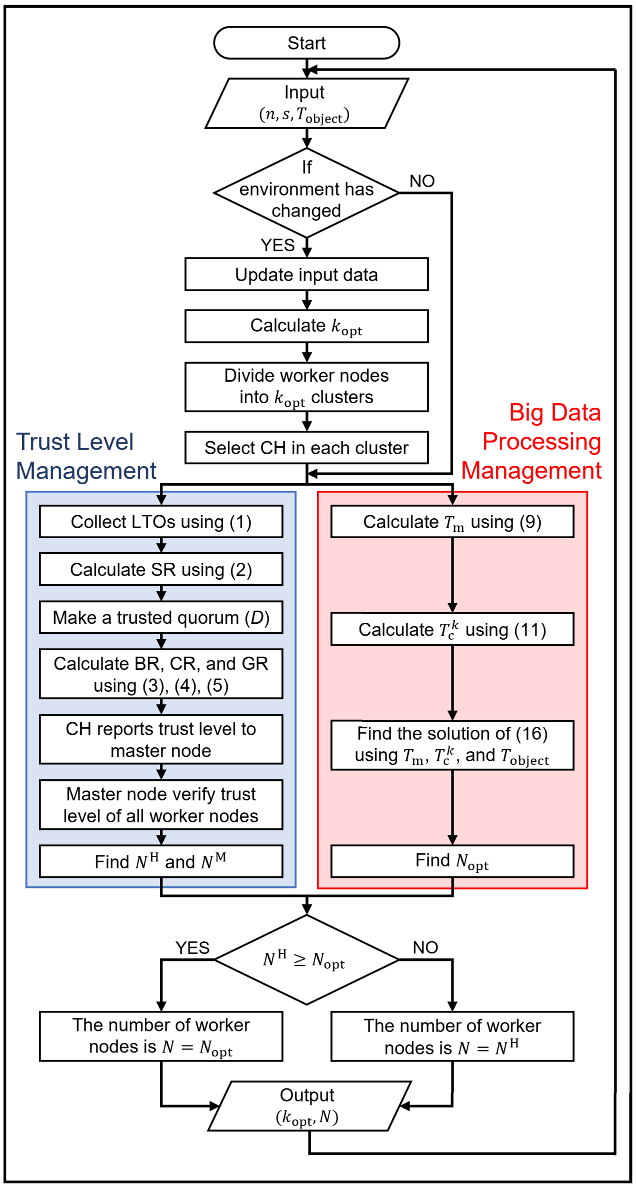
where $\min(x)$ is the minimum number of x. ■

IV. OPERATIONAL FLOWCHART OF TSAF

The operational flowchart of the TSAF scheme, which finds the optimal number of clusters (k_{opt}) needed to manage the trust level, as well as the number of worker nodes (N) to satisfy real-time data processing, is presented in Fig. 3. In Fig. 3, the input includes the total number of worker nodes (n), the target time to process the received data (T_{object}), and the input data segment size (s) that is used as the unit level of data processing in Spark DStream. The parameters n, s, and T_{object} can be changed by the administrator at the beginning or during the DStream data processing.

When TSAF starts, the big data scheme first checks whether the parameters have changed by comparing it with the previous parameters. If the parameters have changed, the value of k_{opt} is recalculated. Next the master node distributes the task and data to the worker nodes of the k_{opt} clusters. The master node selects the worker nodes with the highest trust level to serve as CHs. Then the trust level management process and the big data processing are performed simultaneously. In the big data processing, the total number of required worker nodes N_{opt} is calculated using T_{m} (the time for the master node to distribute data to each cluster), T_c^k (the time to process data inside the cluster), and T_{object}. In the trust level management process, as described in Section III-1, the CH calculates the trust level of the worker nodes in the cluster based on the events that occurred during recent big data processing. The calculated trust level of worker nodes is reported to the master node. The master node performs the process of verifying the reported trust level. Afterwards, the worker nodes are divided into honest nodes and malicious nodes, and the number of honest nodes in the system (N^{H}) and the number of malicious nodes (N^{M}) are identified.

The obtained N_{opt} is compared with N^{H}, which is the estimated number of honest worker nodes that can be assigned to data processing. If $N^{\text{H}} \geq N_{\text{opt}}$, the data is processed by N_{opt}



Start
Input $(n, s, T_{\text{object}})$
If environment has changed
NO
YES
Update input data
Calculate k_{opt}
Divide worker nodes into k_{opt} clusters
Select CH in each cluster
Trust Level Management
Big Data Processing Management
Collect LTOs using (1)
Calculate SR using (2)
Make a trusted quorum (D)
Calculate BR, CR, and GR using (3), (4), (5)
CH reports trust level to master node
Master node verify trust level of all worker nodes
Find N^H and N^M
Calculate T_m using (9)
Calculate T_c^k using (11)
Find the solution of (16) using T_m, T_c^k, and T_{object}
Find N_{opt}
$N^H \geq N_{\text{opt}}$
YES
NO
The number of worker nodes is $N = N_{\text{opt}}$
The number of worker nodes is $N = N^H$
Output (k_{opt}, N)

FIGURE 3. Operational flowchart of the TSAF scheme.

nodes. However, if $N^H < N_{\text{opt}}$, all N^H nodes will be used to process the data.

The number of N_{opt} can change as the input data size changes. As explained in section II, TSAF considers not only directly connected worker nodes, but also external worker nodes connected through the cloud. Therefore, if the number of honest worker nodes required to satisfy the object time is insufficient, the master node will also use worker nodes in the external cloud. However, when new worker nodes in the cloud are connected, there may be changes in the clusters of the Spark big data processing system due to a change in the total number of worker nodes in the overall system.

V. SIMULATION AND DISCUSSION

In this section, real-time big data processing technologies OptEx [13], Spark adaptive failure-compensation (SAF) scheme [14], Spark real-time streaming adaptive failure-compensation (SRSAF) scheme [15], and TSAF scheme are compared using MATLAB 2020a based simulation. Most of the simulation parameters were performed according to the basic environment of Apache Spark Streaming. The big data targeted object time was set to 0.5 seconds, which is the shortest output sequence time for Apache Spark Streaming systems. The number of worker nodes in the Spark big data system is 75 and the error rate of the communication process and execution process was assumed to be 0.05. The batch size of DStream was chosen to satisfy the object time limit of 0.5 based on an errorless status condition.

Fig. 4 shows the data processing time for each scheme according to the number of DStreams when there are no malicious nodes. In this case, no recovery time due to malicious nodes is needed. However, recovery time due to errors occurring in the communication and data processing is needed. In the case of OptEx, it was found that the object time could not be satisfied for any case when errors occur. In the case of SAF and SRSAF, a performance close to the object time is obtained, however, SRSAF and SAF each experience an object time violation when the number of DStreams

TABLE 3. Simulation parameters.

Parameters	Value
Object time of data processing	0.5 s
Number of worker nodes	75
CPU	Intel i5-7600
Number of CPU cores	4 cores
RAM size	4 GB
LTO window size to calculate the trust level	30 s
Number of iterations	3
Error rate of the communications process	0.05
Error rate of the execution process	0.05
Misbehavior probability of malicious nodes in CBA	0.5

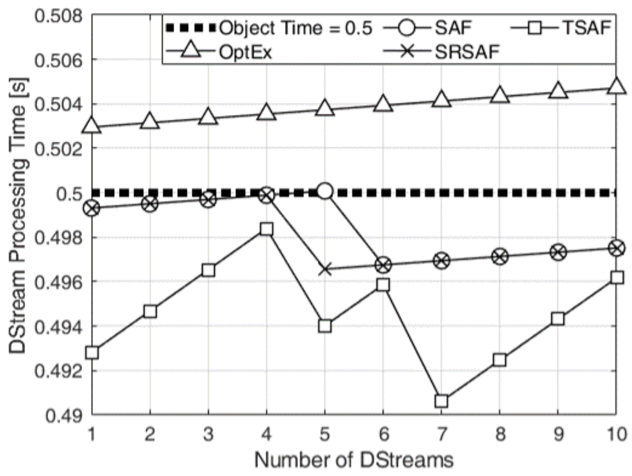


FIGURE 4. Comparison of processing time based on big data processing techniques according to the number of DStreams. The proposed TSAF always shows a lower processing time than the other reference big data models.

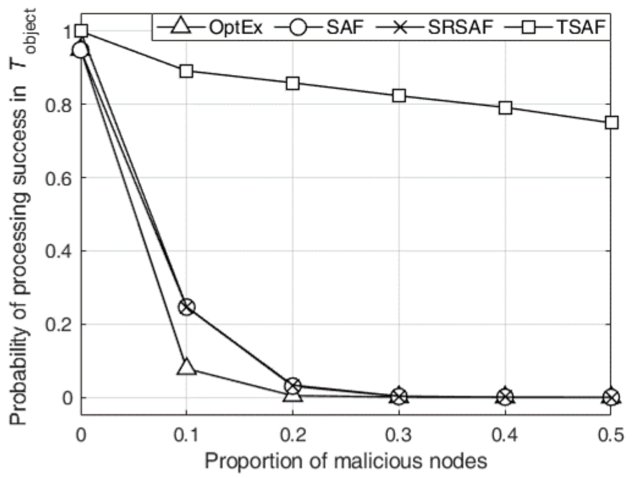


FIGURE 5. Probability of successful processing within T_{object} according to the ratio of malicious nodes among the worker nodes. The proposed TSAF shows a significantly higher probability of satisfying T_{object} by using a trust management scheme that defends against malicious node security attacks.

are 4 and 5, respectively. The results show that the TSAF scheme can satisfy the targeted object time for all cases tested, and the processing time was consistently $1 \sim 2\%$ lower than that of OptEx, SAF, and SRSAF.

In the case of SAF, SRSAF, and TSAF, there is a section where the processing time suddenly decreases, which is due to a change in the number of worker nodes (which are positive integers) used by the schemes. In the case of TSAF, the change occurs more frequently than in the case of SAF or SRSAF, because more error recovery control variables are considered internally when calculating the data processing time based on changes in the number of DStreams.

Fig. 5 shows the probability of processing data within the object time T_{object} when malicious nodes exist. In this simulation, conflicting behavior attacks (CBAs) are applied [11]. In CBA, malicious nodes have a probability of 0.5 to perform wrong behavior on half of the honest nodes and pretend to be correct for the other honest nodes. In addition, the malicious nodes report LTOs as 1 for half of the honest nodes and report LTOs as 0 for the remaining honest nodes.

If there are no malicious nodes, all schemes satisfy the processing object time. However, as the ratio of malicious nodes increases in the OptEx, SAF, and SRSAF big data systems, the ratio that satisfies the object time rapidly decreases. This is because data processing is delayed by the malicious nodes, which takes more time to enter the recovery process and re-process the DStream.

Fig. 5 shows that when the ratio of malicious nodes performing security attacks on worker nodes increases, the probability of satisfying the object time decreases as there is a lack of honest nodes (i.e., the $N^{\text{H}} < N_{\text{opt}}$ case) to complete the data process in time. When the ratio of malicious nodes performing security attacks on worker nodes reaches 0.5, Fig. 5 shows that the TSAF scheme can still satisfy over 75% of the tasks within the target object time, which is significantly higher compared to the other big data processing schemes. TSAF can perform at this level because malicious nodes are identified in advance and only honest nodes are used to process the DStreams.

If the ratio of malicious nodes is larger than 0.5, the processing success probability of TSAF may drop sharply. GlobalTrust was used to make trust decisions in the simulation experiments [11], [12], and when the ratio of malicious nodes exceeds 0.5, the opinions of the malicious nodes become a majority, and when they collude, the efficiency of the trust management scheme drops sharply, making it impossible to distinguish malicious nodes.

VI. CONCLUSION

Network security procedures commonly slow down the response time of big data systems. To overcome this issue, in this paper a scheme to enhance the security and processing speed in networks with errors and malicious nodes is proposed. The proposed TSAF scheme is effective in identifying malicious nodes among the worker nodes and helps to satisfy the targeted object time of real-time big data processing jobs. The TSAF scheme first computes the number of clusters that results in the lowest trust computational complexity, and then the optimal number of nodes is derived and used to process the data within the targeted object time. When compared to OptEx, SAF, and SRSAF, in the case of no malicious nodes, the TSAF scheme shows a superior performance in satisfying the targeted object time. In addition, when the big data network has malicious nodes conducting CBA attacks, the TSAF scheme results in a significantly higher probability of satisfying the job processing object time compared to the other schemes.

When there are changes in the input data size and security attacks in the Spark big data processing system, TSAF can respond by immediately adjusting its system configuration and number of required worker nodes according to the monitored system circumstances.

In this paper, the proposed TSAF system can only make optimized adjustments corresponding to errors occurring during the communication and data processing of the Spark big data system. However, errors occurring during the initial setup time where Spark system configuration changes are made, and other related initial communication adjustments are conducted, were not considered. In future work, a more comprehensive optimized error recovery scheme needs to be developed so that accurate processing times can be predicted during the initial setup time as well.

REFERENCES

[1] H. Fatemidokht, M. K. Rafsanjani, B. B. Gupta, and C.-H. Hsu, "Efficient and secure routing protocol based on artificial intelligence algorithms with UAV-assisted for vehicular ad hoc networks in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 7, pp. 4757–4769, Jul. 2021.

[2] S. Huang, Z. Zeng, K. Ota, M. Dong, T. Wang, and N. N. Xiong, "An intelligent collaboration trust interconnections system for mobile information control in ubiquitous 5G networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 1, pp. 347–365, Jan. 2021.

[3] M. Huang, A. Liu, N. N. Xiong, and J. Wu, "A UAV-assisted ubiquitous trust communication system in 5G and beyond networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 11, pp. 3444–3458, Nov. 2021, doi: 10.1109/JSAC.2021.3088675.

[4] C. Boudagdigue, A. Benslimane, A. Kobbane, and J. Liu, "Trust management in industrial Internet of Things," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3667–3682, 2020.

[5] M. Zhaofeng, W. Lingyun, W. Xiaochang, W. Zhen, and Z. Weizhe, "Blockchain-enabled decentralized trust management and secure usage control of IoT big data," *IEEE Internet Things J.*, vol. 7, no. 5, pp. 4000–4015, May 2020.

[6] S. Atiewi, A. Al-Rahayfeh, M. Almiani, S. Yussof, O. Alfandi, A. Abugabah, and Y. Jararweh, "Scalable and secure big data IoT system based on multifactor authentication and lightweight cryptography," *IEEE Access*, vol. 8, pp. 113498–113511, 2020.

[7] J. Shen, D. Liu, Q. Liu, X. Sun, and Y. Zhang, "Secure authentication in cloud big data with hierarchical attribute authorization structure," *IEEE Trans. Big Data*, vol. 7, no. 4, pp. 668–677, Oct. 2021.

[8] D. T. Dang, D. Hoang, and D. Nguyen, "Trust-based scheduling framework for big data processing with MapReduce," *IEEE Trans. Services Comput.*, early access, Sep. 3, 2019, doi: 10.1109/TSC.2019.2938959.

[9] X. Li, J. Yuan, H. Ma, and W. Yao, "Fast and parallel trust computing scheme based on big data analysis for collaboration cloud service," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 8, pp. 1917–1931, Aug. 2018.

[10] X. Yue and Q. Liu, "Parallel algorithm of improved FunkSVD based on spark," *KSII Trans. Internet Inform. Syst.*, vol. 15, no. 5, pp. 1649–1665, May 2021.

[11] X. Chen, J. H. Cho, and S. Zhu, "GlobalTrust: An attack-resilient reputation system for tactical networks," in *Proc. 11th IEEE SECON*, Singapore, Jun. 2014, pp. 275–283.

[12] S. Seo, J.-W. Kim, J.-D. Kim, and J.-M. Chung, "Reconfiguration time and complexity minimized trust-based clustering scheme for MANETs," *EURASIP J. Wireless Commun. Netw.*, vol. 2017, no. 1, pp. 1–7, Sep. 2017.

[13] S. Sidhanta, W. Golab, and S. Mukhopadhyay, "Deadline-aware cost optimization model for spark," *IEEE Trans. Big Data*, vol. 7, no. 1, pp. 115–127, Mar. 2021.

[14] J. Lee, B. Kim, and J.-M. Chung, "Time estimation and resource minimization scheme for apache spark and Hadoop big data systems with failures," *IEEE Access*, vol. 7, pp. 9658–9666, 2019.

[15] S. Seo, D. Ko, and J. Chung, "Combined time bound optimization of control, communication, and data processing for FSO-based 6G UAV aerial networks," *Electron. Telecommun. Res. Inst. J.*, vol. 42, no. 5, pp. 700–711, Oct. 2020.

SEUNGWOO SEO received the B.S. degree from the School of Electrical and Electronic Engineering, Yonsei University, Seoul, South Korea, in 2014, where he is currently pursuing the combined M.S. and Ph.D. degree in electrical and electronic engineering. He is also a Research Member of the Communications and Networking Laboratory (CNL). His research interests include big data, augmented reality, 6G UAV networks, and trust network security management technology.

JONG-MOON CHUNG (Senior Member, IEEE) received the B.S. and M.S. degrees in electronic engineering from Yonsei University, and the Ph.D. degree in electrical engineering from Pennsylvania State University. From 1997 to 1999, he was an Assistant Professor and an Instructor with Pennsylvania State University. From 2000 to 2005, he was with Oklahoma State University as a Tenured Associate Professor. Since 2005, he has been a Professor with the School of Electrical and Electronic Engineering. He is currently the Associate Dean of the College of Engineering, a Professor with the Department of Emergency Medicine, College of Medicine, and the Director of CNL, all at Yonsei University. He is also an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, an Associate Editor of the IEEE TRANSACTIONS ON CONSUMER ELECTRONICS, the Section Editor of the *ETRI Journal*, (Wiley), the Co-Editor-in-Chief of the *KSII Transactions on Internet and Information Systems*, and the Vice President of the IEEE Consumer Technology Society (CTSoc) and the IEEE Product Safety Engineering Society (PSES).