International Journal of Soft Computing 13 (5): 139-148, 2018
ISSN: 1816-9503

High Performance Complex Event Processing for Nuclear Reactor Simulation

Nehal Safwat, Sherihan Abuelenin and Ahmed Tolba
Faculty of Computers and Information, Mansoura University, Mansoura, Egypt

Abstract: Many application fields require detecting real-time composite events and reacting to them immediately considering their sensitive condition. This raises the need for efficient, situational aware Complex Event Processing (CEP) engines to handle massive amount of streamed data. To satisfy this need an optimization method to enhance the engine's efficiency and reliability is proposed. It adopts a combination of enhanced buffer allocation method with pattern matching process under the supervision of domain expert. An error detection and recovery to handle any failure case is proposed with the system to assure high reliability and accuracy in delivering and detecting the events. The proposed system is evaluated in terms of performance over time, throughput, error rate and pattern detection accuracy. A comparative evaluation with a state-of-art engine is done and the proposed system outperformed it under challenging workloads.

Key words: Complex event processing, buffer allocation, pattern matching, big data, comparative evaluation, outperformed

INTRODUCTION

Big data is widely recognized as an incremental innovation that organizations are now using to handle new technologies and production processes (Lugmayr *et al*., 2017). A new generation of technologies called Big Data Analytics (BDA) is the main engine to discover the true meaning behind large data volumes.

Many projects are established on conducting researches on big data with areas such as data mining, machine learning, computational intelligence (Jung, 2017), information fusion (Gao *et al*., 2017), semantic Web and social networks (Storey and Song, 2017). Also, it helps to implement number of application areas, such as in personal location data (Grewal *et al*., 2016), governance services (Wang *et al*., 2018; Williams *et al*., 2017), fraud detection (Terzi *et al*., 2017) and national security (Chan and Moses, 2017).

There're limitations that faced big data such as, scalability and storage issues, timeliness of analysis, representation and dealing with heterogeneous data, data errors, privacy and security (Bhadani and Jothimani, 2016). These challenges justify the need for revolutionary step forward technologies that matches the new needs like Complex Event Processing (CEP) and Data Stream Management Systems (DSMS).

DSMS is concerned about processing unbounded flowing information by running continuous queries, then producing the output to end-users with low possible latency. It requires ensuring reliability, scalability and abstraction in handling large-scale and data-intensive volumes on cloud computing systems (Jing *et al*., 2017; Guo *et al*., 2017).

Complex Event Processing (CEP) systems come with a different method, aiming to process notifications of events observed by heterogeneous sources in real-time for situation-aware applications that requires near immediate response. This task is done by a central processing component called CEP engine which is a part of the CEP middleware that also, contains the set of rules and the events sources. The CEP engine is the main responsible module which filters the relevant generic events then derives the correlations between them based on query matching process. CEP is a technology which allows finding real-time relationships between different events using elements such as timing, causality and membership in a stream of data in order to extract relevant information (Flouris *et al*., 2017). CEP relies on a number of techniques including: event-pattern detection, event abstraction, event filtering, event aggregation and transformation, modeling event hierarchies, detecting relationships between events and abstracting event driven processes (Dayarathna and Perera, 2018).

Many problems could arise while running the event processing engine either in applying the rules, making the rightful decision or even in updating the system with most recent rules.

A new technology is proposed to solve such problems called the digital twin. It is an integrated multi-physics, multi-scale and probabilistic simulation of any complex product. It uses all available sources either

Corresponding Author: Nehal Safwat, Faculty of Computers and Information, Mansoura University, Mansoura, Egypt

streamed or physical data to mimic the behavior of its corresponding twin (Tao *et al*., 2018; Bayer *et al*., 2018). This means there're two models from the same object one in the physical shape and the other is virtual entity placed in the cloud. It replicates everything from the machine's history, performance and the deviations from expected operation to analyze the real machine performance and help optimizing various activities in the entire machine lifecycle.

Literature review: In the last few years the research community focused on developing CEP systems for both academia and industry purposes. The enhancement goes through two major ways, either by providing expressive query language for efficient temporal pattern matching or optimizing the engine's performance by improving the event driven architecture and processing algorithm. Few researches have merged both aims and built their own new systems with specially designed event processing language. This research falls in the field of handling critical complex events with consideration to the sensitivity of the discussed environment. Many research works adopted this viewpoint trying to redirect the research community orientation to study optimizing the CEP engines considering the situational awareness concept.

To achieve such goals by Sandha *et al.* (2017) developed real-time analysis framework based on Kafka and Spark platforms. The former helps users to produce and consume events in a fault tolerant manner and the latter provides both in-memory and real-time data processing guided by predefined rules. The goals of implemented use cases are to detect and predict heart failure situation and stress measurement, proving that general platforms can process any behavioral patterns. They simulated the users by sending real-time data using Spark streaming to the Kafka cluster and evaluated their work by how accurate the heart failure diagnosis data returns.

Processing all incoming events is the main concept about CEP but it may lead to longer execution time especially if we 're handling sensitive systems. Therefore, (Chandrathilake *et al*., 2016) merged both CEP and Machine Learning (ML) to process only hazardous events. Using lightweight data filtering algorithm in the CEP engine the filtered events are fed to the machine learner to be more refined as a preprocessing steps. The system is applied on weather monitoring case associated by the Weather Research and Forecasting (WRF) Model for temperatureprediction and pattern detection. The result shows 75% less execution time using CEP and ML together than using WRF alone.

Proving that ML isn't a useful solution for every case (Viegas *et al*., 2017) developed intrusion detection resilient system for real-time network traffic. First the collected events from monitored sources are carried through a broker (message middleware) to a message consumer to the proposed system. Using stream learning decision trees, they classify the network flow for either normal or attack upon specified rules. Their evaluation shows higher advantage for the approach over the machine learning with higher throughput.

But Cugola *et al.* (2015) had different point of view from the aforementioned works, believing that not all rules applied by the domain expert will be 100% related to the environment's behavior. They addressed the problem of verifying the rule's relevance to the system's requirements.

Taking into consideration the critical environment and the possibility of writing wrong rules, they developed a scalable tool named CAVE for testing rule's reliability and compared them to the application's assumptions.

In CAVE rules are translated into set of configurations and the verification is checked by solving constraints method to assure if the rules will satisfy the environment's behavior. They evaluated the performance under different constraints types and parameters.

The research of Korber *et al.* (2018) proposed TPStream, an event processing operator focuses on detecting complex temporal relations among situations. The problem is having only simple expressions to define the temporal relations between events which are not enough for all situations. Some situations need more than before, after and during temporal patterns for events that take periods of time.

TP Stream works on two levels by deriving the interested situation from the streams then detecting temporal patterns among them. A continuous process is performed with low latency in analysis and matching to keep deriving the situations and relating them to each other. To derive situations the system uses aggregation and predicate evaluation on continuous stream and Allen's interval algebra is adopted to define the appropriate temporal relations. The evaluation is measured in terms of processing time, latency and throughput in comparison with Esper, SASE+ and ISEQ. The system shows a good performance considering the tested scenarios.

As many CEP applications grow with high complexity levels, it becomes difficult to understand and analyze them properly. A formal framework using rewriting logic and Maude (Clavel *et al*., 2007) to prove any CEP engine's properties is proposed by Burgueno *et al.* (2018). Maude ecodes the concepts and mechanisms of CEP concept in terms of CEP patterns, time model, executing the system and structural aspects. By mapping the CEP patterns into Maude rules it becomes easier to make static and dynamic analysis to discover any potential errors and understand the CEP applications behavior. To validate the framework two case studies are applied, a temperature monitoring

system and air quality monitoring system. This shows efficiency and correctness of the framework in simulating both systems and analyzing them statically and dynamically. Both systems are mapped from Esper EPL using Esper parser into Maude specifications to cover any required feature.

Researchers by Itria *et al.* (2017) proposed an accurate, flexible crisis management system to detect any critical situation with eliminating false information to create trusted events. The way to improve this issue is through correlating events based on temporal patterns according to specified categories. Then it passes to Event Trust Analysis (ETA) component to classify them either anomaly or normal. This classification is done by event filtering, mean value and k-medoids algorithms in specific order.

The proposed system is developed in the event extraction and integration level of the secure framework. Experimental validation is performed to show the functionality behavior, performance and scalability of the system with and without ETA. It shows adequate performance for the purpose of crisis detection and management and promising effectiveness in other related fields.

A new concept named "Digital Twin" isintroduced in this field to give a realistic feedback with increased representativeness for any experiment with reduced cost. By Zhuang *et al.* (2018) emerged this technology for complex product assembly shop-floor in four main stages. A real-time information acquisition to collaborate all the intended data sources either streamed or physical in an organized behavior. Build a virtual digital twin to map the physical assembly shop-floor with assuring its effectiveness and fidelity. Provide a large-scale process optimization and decision-making by combining the digital twin with a big-data driven predictive management. This helps giving active instead of passive prediction for any possible upcoming failures. Map between both twins, physical and virtual, to reflect the conditions and exchange the real-time analytics. The applied case study showsdetailed implementation for digital twin-based smart production management and will continue to improve this framework for each stage.

MATERIALS AND METHODS

Overview: Since, each event can probably be an indicator of temperature anomaly, it is necessary to identify every possible occurrence of incoming events and map them to the relevant parameters in safe and reliable system.

Figure 1 shows the proposed system consists of three major parts applied in a simulated nuclear reactor. In temperature simulation part the stream data is first fed from the hypothetical nuclear simulation part into the Complex Event Processing (CEP) engine part. Second in

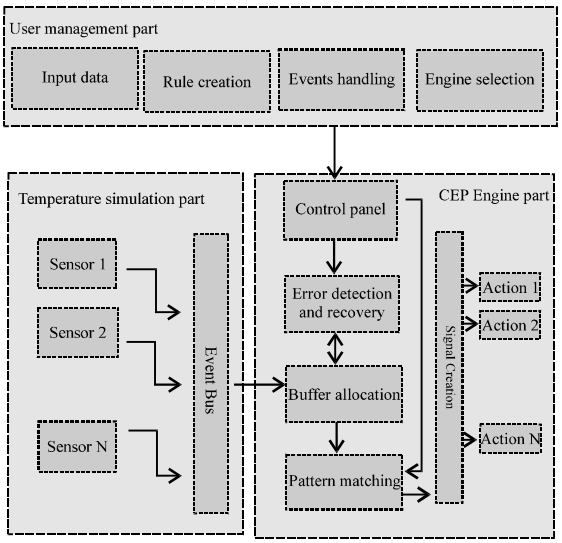

Fig. 1: Structural level of the proposed model

the user management part the expert engineer set the rules to be matched to the event notifications after being stored in buffers. Finally in the CEP engine, the processed primitive events form complex events and initiate a signal that fires the suitable action. The next subsections discuss in details the system's components.

The first part is a user management panel where the domain expert defines the primitive rules and limitations of the system's behavior. Different tools are used to ensure covering all aspects like: input data, rules creation, event handling and engine selection which will be stored in the rule database. With the combination of different constructors that each tool provides, rule manager will have full control to describe various rules abstractions. They don't only define the queries that'll be applied to the streams but also, the appropriate action for the detected events. Instead of firing actions over single event, these event abstraction rules are released by event patterns which promote simultaneous events detection.

Considering the sensitivity of the system there must be decision anyway during the process. So, it is flexible to add new rules or new event sources if required. These rules will be customized and fed to the control panel waiting to be applied on the streams as they occur.

Through this panel the expert sets the rules for four types of events: monitor sends the average temperature degree during a specific temporal pattern that could change if required; critical alarms the system if there're two consecutive degrees above certain threshold (500°C), e.g., 550, 600°C; dangerous warns the system if a rapid increasing in temperature degrees happens where the first one is above the dangerous event threshold and each

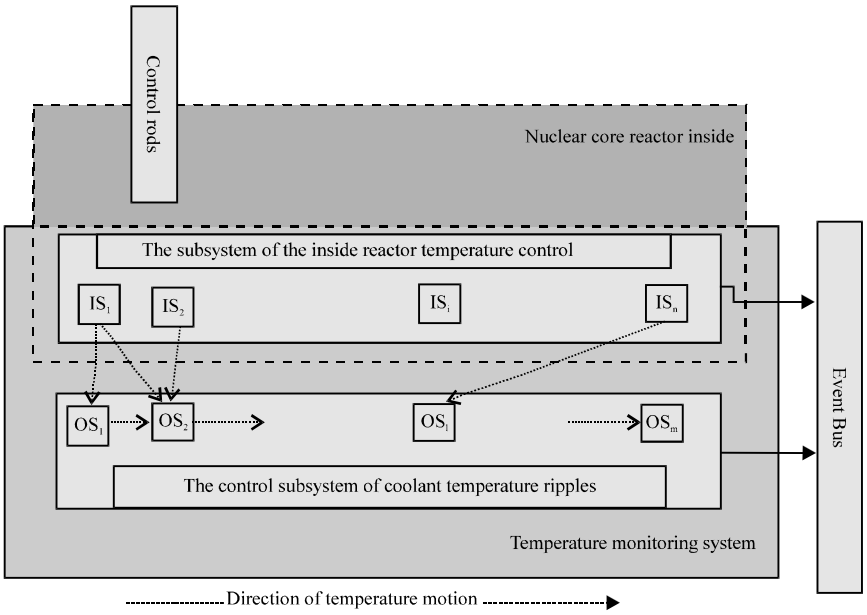

Fig. 2: Temperature monitoring system of nuclear core reactor

subsequent event is greater than its former and the fourth event is greater than the first with specific event multiplier (1.9) e.g., 550, 720, 960, 104°C , finally, vision alerts the system if there're two consecutive events critical then dangerous to send the most warning alert. Each event is subject to modification based on the expert's knowledge.

Temperature simulation part: Designing a comprehensive model to mimic the nuclear reactor temperature subsystem needed wide investigations to correctly place the sensors. Figure 2 represents the system's input and the temperature sensors. They are divided into two groups: Inner Sensors (Is_i) with i = 1, ... , n where n is quantity of inner sensors and Outer Sensors (OS_j) with j = 1, ... , m where m is the quantity of outer sensors.

OS_1 is the sensor which makes fast reaction to the temperature changes, IS_n makes the slowest reaction amidst all inner sensors. There're dependencies between the two groups using mathematical functions set by the expert D (IS_i, OS_j). It shows an image of temperature dissemination in the nuclear core reactor. Since, the signals of action are based on signals from sensors, verification is made to assure the authenticity of the temperature degree released by each sensor.

Complex event processing engine: This section explains the approach to extract the meaningful patterns from primitive events which are streamed by the temperature system using the customizations from user management panel. After data acquisition, the knowledge used to make the decision will go through two consecutiveapproaches: "Buffer allocation" and "Pattern matching". When the pattern matching finds a decision it will inform the "action part" where it transforms the signal to execution.

Algorithm 1: Pseudo code of proposed system

```
Inputs:                 ISi, OSj
        D(ISi, OSj)
        Btn,//Set the required amount of buffers
        Buffers rules,//prioritize event signals
        PM rules//in pattern matching panel
Output:                 Flow f such that f is a set of actions
function                SIMULATE STREAM OF RANDOM DEGREES
BASED ON D(ISi, OSj)
for each event in the stream
{
        Send to BTn through Event Bus in FIFO
                        if (BTn fail || BTn overflow) then
                        {
                                Divide BTn events on BTn-1
                        }
                        Check in PM rules to detect complex events
}
return f
```

The implemented rules in user management panel will show the dependencies between inside and outside sensors. Additionally, buffers rules to clarify the direction of event transmission from temperature sensors to the temperature buffers. The importance of these rules is to prioritize event's signals (Fig. 3). For example, assuming the buffer BT_i receives all events from the most sensuous and essential sensors IS_1 and IS_2 which are closest to the core. The less important temperature events from other sensors will be directed to other temperature buffers,

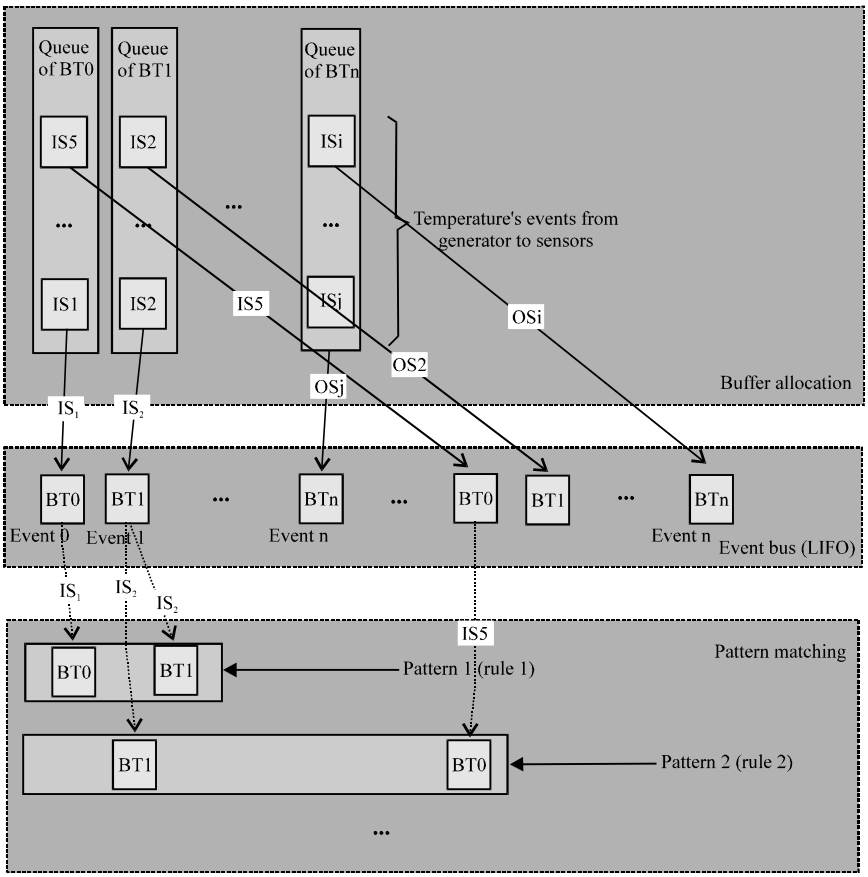

Fig. 3: Signal transmission through the system

subsequently there'll be a mapping from all sensors to all buffers. This mapping of activities will help optimizing the work of pattern matching algorithm which as well enhances the reliability in the whole system by transmitting the highest priority events first.

The buffer allocation algorithm is endorsed as an efficient algorithm in a continuous flow production line with unreliable machines where the problem is formulated as a non-linear programming with linear constraints. It is defined as a data transfer conveyor between two related objects which gives an opportunity to ensure smooth synchronization of data transmission.

Figure 3 shows this layer is showing the signal transmission from temperature events in the buffers to the pattern matching part through event bus. Instead of applying the queries over each event upon arrival the event buffering will be more effective by performing the queries on set of buffered events during precise sliding window.

To optimize the buffer allocation problem for data streams, the temperature events are distributed amidst the Buffers Temperature (BT) in First in First Out (FIFO) behavior using multithreading concept. The FIFO architecture is more efficient because it will save the order of the temperature events from the temperature sensors in the buffers.

The quantity of buffers, capacity of queues for each BT and relations between sensors and the BT are identified earlier in the user management panel. These objective features may be considered with maximizing throughput, minimizing average work-in-process or minimizing the total number of buffer slots subject in each case to appropriate constraints.

The proposed system takes as input the primitive events and forms the complex, meaningful ones which exemplify the real-world environment. These detected indications will either fire the suitable actions or be processed again with other primitive events to constitute a new complex one. The system needs to find relationships between streamed events as they occur immediately by analyzing them over predefined patterns described by the domain expert. For example, consecutive increasing in temperature degrees above certain threshold during short time intervals or intermittent periodic increasing through wide time range. Both situations are

real scenarios which are streamed from the same sensors but will be treated differently in terms of the pattern rules and the action released. In addition, the system includes a method to detect buffers failure to support the optimization with immediate solution if such situation occurred during real-time processing.

Figure 3 shows the pattern matching algorithm is grasping the actual temperature events from the event bus and compares them with the rules from knowledge base.

The event bus works in FIFO mode to assure including all events in consecutive manner. Pattern matching in such sensitive and critical systems shouldn't miss any event because the action is fired base on multiple events rather than single one. If the CEP engine had to deal with many rules and pattern definitions it'll spend more time to analyze them, so, they are designed to be short, efficient and comprehensive to describe more situations. Also, concurrent and fast analysis is considered to maintain the actuality of the results.

Considering the system's sensitivity there must be an action or alert for every streamed data without neglecting any. An optimized method is provided to diagnose and handle failures by a continuous performance of the error detection and recovery task which scans the buffers and sensor's work. Each storage buffer has a separate thread assigned with a queue of required tasks to execute them. An ID is identified for each thread to dynamically keep tracking the executed tasks and to alert if there's any buffer failure. For the purpose of tracking the tasks, the system take the thread's ID and returns the list of remain tasks to be executed and gives a comprehensive survey about the performance in a HashMap form.

In case of any buffer failure or even if it took longer time than expected, the system first sends an alert and increases error's number. Then, it divides the assigned tasks of the failed buffer for the remaining ones considering the working load and capacity of the other buffers. This process preserves the system's reliability for the whole running time to assure the delivery of all streamed events into the pattern matching without dropping any.

RESULTS AND DISCUSSION

Experimental setting: The experiment setup to evaluate the CEP engine used simulated cases to demonstrate the actual generation of nuclear core reactor temperature degrees with control over required number of vents. All the experiments discussed below are erformed using Windows 7 ultimate operating system with Intel(R) Core(TM) i5-2450 and 4 GB f RAM on 64-bit Java JDK V1.7.

Performance evaluation: One of the proposed system goals is to accelerate and enhance, the detecting procedure under any sudden failure situation in order to handle the sensitive and crucial real-time events. Therefore, the performance is evaluated under different circumstances to ascertain if the system can manage the situational awareness necessity and to show the effectiveness of the optimization. In the proposed optimization, the CEP module acts as real-time engine to filter the temperature anomalies from stream of nuclear reactor temperature degrees with alerting the incidents on monitoring screen to the expert engineer. The evaluation will go through two ways:

- Examining the system's performance under different scenarios
- Comparing the proposed system's functionality with another CEP engine that considered the same case and could handle same rules

The search for a similar system to compare with required finding widely used open-source Java-based CEP software to facilitate execution and testing. Esper solution is chosen as it is an open source java based engine and contains expressive event processing language with efficient analysis procedure. The evaluation of both systems is done with same amount of events, number of working sensors and same rules. They are executed five times for each test and the average values are mentioned here.

The evaluation is designed to test the performance from three different aspects to gather the major required points: achievement in detecting events over specified time, performance under increasing number of buffers and rules and the accuracy of detecting events.

Firing the accurate action in different situations requires precise and rapid filtering process without neglecting any event. Many obstacles will be a challenge to do so, like the ability of the system to remain running steadily the whole processing time without dropping any of the continuous data stream.

The performance of the system is evaluated to check at which situation the system can keep working well with a steady state. The evaluation is presented in three scenarios, each scenario represents the system with the same number of sensors, events and rules but in different amount of buffers.

- Scenario one is the system's performance with a ratio 1:1 between numbers of sensors to buffers
- Scenario two is the system's performance with a ratio 1:2 between numbers of sensors to buffers
- Scenario three is the system's performance with a ratio 1:3 between numbers of sensors to buffers

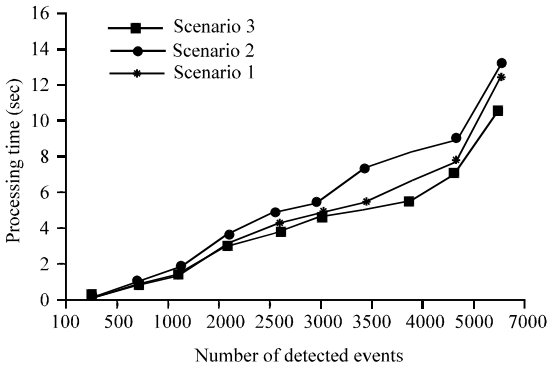

Fig. 4: System's performance in detecting events over time

The system does an initializing step as a preparatory work that doesn't influence the processing time. Afterwards, the system handles events depending on the amount of dedicated buffers. Figure 4 shows the number of detected events over the increment of processing time in three different scenarios.

Figure 4 shows from the beginning until the 1000 events, the three scenarios are very close at processing time. Then scenario 1 kept running steadily and smoothly in processing events without rapidly increment in processing time until it reached 5000 events, unlike scenario 2 that had noticeable increase in processing time after 3000 events. Scenario 3 is close to results of scenario 1 but with minor rise in processing time after 3500 events.

The increase in running time is not considered as critical zone because the last amount of events is twice the aforementioned increase. Number of rules isn't considered as a factor in affecting the system's overall time because simply in complex event processing the unsatisfied rules by the data streams will be immediately skipped.

Figure 5 shows examining the system's throughput while increasing number of buffers under the same scenarios as previously mentioned, to show how it affects the system performance.

Low latency and high throughput are the aim to achieve in different situations. At the beginning the throughput increases steadily in the whole scenarios but with different amount depending on number of dedicated buffers. Scenario 3 has a drop around the second 340 and then it continues to increase but lower than scenario 1 and 2. The maximum capability of the system's throughput is around 387 k events at 600 sec and then either it remains at the same amount or will have a minor drop. The result allows the expert engineer to design more rules to filter the stream events and initialize the appropriate actions. Accurate detection of the composite

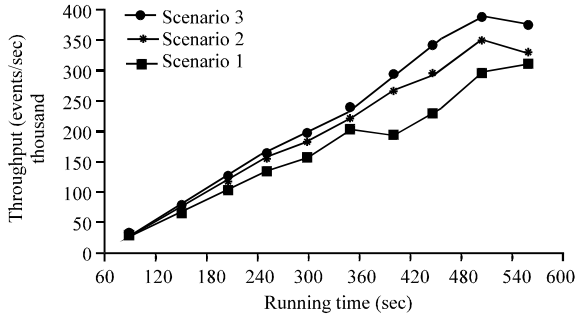

Fig. 5: System throughput with different buffer's number

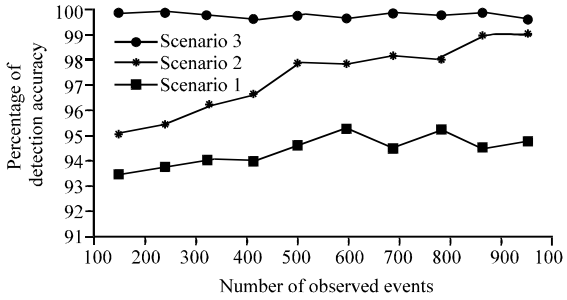

Fig. 6: Events detection accuracy in the system

events is a necessity in the CEP field to assure considering every situation. Figure 6 shows how well the system performs in detecting composite events to examine the buffers affection on the query matching process.

In scenario 3 with the obvious distinction between numbers of sensors to buffers, the accuracy started from 93.4% and increased. Conversely, the accuracy is higher in scenario 2 and kept increasing until it reached 98.9%. While in scenario 1 where the ratio is 1:1, the accuracy is the highest among all scenarios with above 99% accuracy for the whole events. This explains the best average ratio between sensors to buffers is in scenario 1 to avoid missing any significant event that may be processed in different buffer, followed by scenario 2.

The system's duty is to detect, alert and handle any error incident because the domain expert won't be aware that an error occurred which may lead to making decisions based on incomplete events. The CEP requirements demand an appropriate handling to any incorrect or missing event. Few factors affect the severity of error's incident as the amount and frequency of happening. Figure 7 shows the performance of the system in the three scenarios in handling the errors. Both scenario 2 and 3 didn't act well in the beginning but soon as number of events increased scenario 2 managed the errors well and the percentage decreased. Unlike scenario 1 which kept performing well in detecting and handling any error

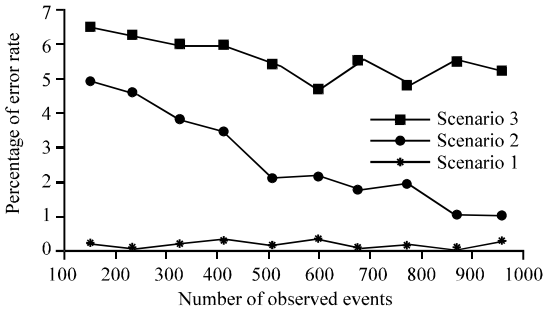

Fig. 7: System's error rate

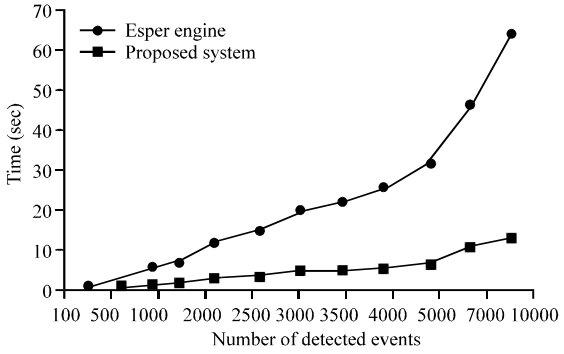

Fig. 8: Comparison between proposed system and Esper: Performance over time

with percentage of errors under 0.5%. It is important to study the behavior of the system under different situations and to compare the behavior with another system. This comparison is made with scenario 1 from the proposed system to consider evenness in sensor's number and to evaluate the affection of query processing and buffer allocation. Figure 8 the direct comparison with Esper shows the effectiveness of the proposed system in terms of performance over time.

At the very beginning they are both detecting same amount of events at same running time but soon Esper's processing time increases rapidly while the proposed system remains processing events steadily in <10 seconds until 7000 events. This test is made with the same amount of rules and sensors.

Figure 9 shows the obtained results for both systems in terms of throughput. Generally, both systems performed well with a throughput of tens of thousands of detected events per sec. But the proposed system outperforms Esper with capability to process more streamed events before dropping them from the input queue, even the limitation of the proposed system is higher than the compared engine. These experiments show two substantial factors of a CEP system: how well

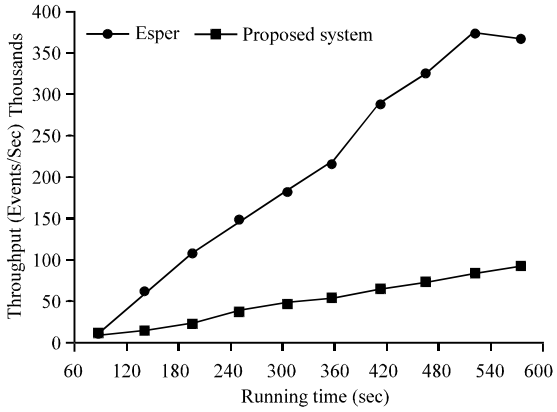

Fig. 9: Comparison between proposed system and Esper: performance under workloads

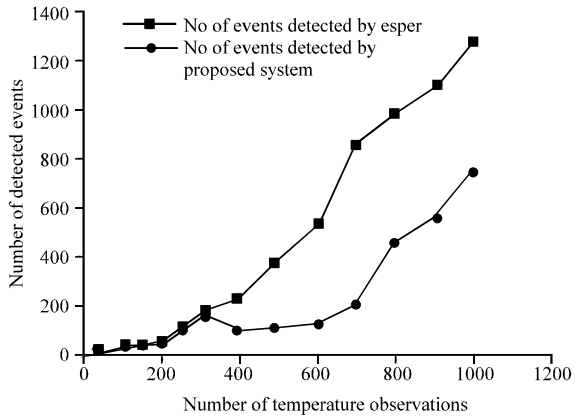

Fig. 10: Comparison between proposed system and Esper: number of detected events

the system performs without delays and under workloads for filtering input events and detecting patterns of events. The performed test in Fig. 10 is done using different synthetic workloads to demonstrate how well both systems perform in pattern detection. To perform this test, the same amount of temperature degrees is fired with the same rules on both systems to detect numerous types of composite events.

Both systems perform well as equal in detecting events from 400 observations but then the proposed system outperforms Esper. In particular, the proposed system is capable of processing more primitive events before starting to clear them from the input queue.

CONCLUSION

This study presents an optimized CEP system that is concerned with situational awareness concept. It

combines an optimized CEP engine capable of processing large volume of events efficiently, a simulation panel to stream events and a user interface panel to customize the rulesbased on an expert domain. The performance evalu ation of the proposed system shows that it isfeasible and effective in terms of throughput, processing time, error handling and accuracy. The results show three different scenarios for the system to achieve the best running case and good performance. The comparison with Esper shows promising optimization performance.

REFERENCES

Bayer, V., S. Kunath, R. Niemeier and J. Horwege, 2018. Signal-Based metamodels for predictive reliability analysis and virtual testing. Adv. Sci. Technol. Eng. Syst. J., 3: 342-347.

Bhadani, A.K. and D. Jothimani, 2016. Big Data: Challenges, Opportunities and Realities. In: Effective Big Data Management and Opportunities for Implementation, Kumar, S.M. and K.G. Dileep (Eds.). IGI Global, Pennsylvania, USA., ISBN:9781522501824, pp: 1-24.

Burgueno, L., J. Boubeta-Puig and A. Vallecillo, 2018. Formalizing complex event processing systems in Maude. IEEE. Access, 6: 23222-23241.

Chan, J. and L.B. Moses, 2017. Making sense of big data for security. Br. J. Criminology, 57: 299-319.

Chandrathilake, H.M.C., H.T.S. Hewawitharana, R.S. Jayawardana, A.D.D. Viduranga and H.D. Bandara *et al*., 2016. Reducing computational time of closed-loop weather monitoring: A complex event processing and machine learning based approach. Proceedings of the 2016 International Conference on Moratuwa Engineering Research Conference (MERCon), April 5-6, 2016, IEEE, Moratuwa, Sri Lanka, ISBN:978-1-5090-0644-1, pp: 78-83.

Clavel, M., F. Duran, S. Eker, P. Lincoln and N. Marti-Oliet *et al*., 2007. All about Maude-A High-Performance Logical Framework: How to Specify Program and Verify Systems in Rewriting Logic. Springer, Berlin, Germany, ISBN:978-3-540-71999-1, Pages: 802.

Cugola, G., A. Margara, M. Pezze and M. Pradella, 2015. Efficient analysis of event processing applications. Proceedings of the 9th ACM International Conference on Distributed Event-Based Systems, June 29-July 03, 2015, ACM, New York, USA., ISBN:978-1-4503-3286-6, pp: 10-21.

Dayarathna, M. and S. Perera, 2018. Recent advancements in event processing. ACM. Comput. Surv. CSUR., 51: 1-36.

Flouris, I., N. Giatrakos, A. Deligiannakis, M. Garofalakis and M. Kamp *et al*., 2017. Issues in complex event processing: Status and prospects in the big data era. J. Syst. Software, 127: 217-236.

Gao, X., W. Li, M. Loomes and L. Wang, 2017. A fused deep learning architecture for viewpoint classification of echocardiography. Inf. Fusion, 36: 103-113.

Grewal, D., Y. Bart, M. Spann and P.P. Zubcsek, 2016. Mobile advertising: A framework and research agenda. J. Interact. Marketing, 34: 3-14.

Guo, Y., J. Rao, D. Cheng and X. Zhou, 2017. Ishuffle: Improving hadoop performance with shuffle-on-write. IEEE. Trans. Parallel Distrib. Syst., 28: 1649-1662.

Itria, M.L., M. Kocsis-Magyar, A. Ceccarelli, P. Lollini and G. Giunta *et al*., 2017. Identification of critical situations via. event processing and event trust analysis. Knowl. Inf. Syst., 52: 147-178.

Jing, W., D. Tong, Y. Wang, J. Wang and Y. Liu et al., 2017. MaMR: High-Performance MapReduce programming model for material cloud applications. Comput. Phys. Commun., 211: 79-87.

Jung, J.J., 2017. Computational collective intelligence with big data: Challenges and opportunities. Future Gener. Comp. Syst., 66: 87-88.

Korber, M., N. Glombiewski and B. Seeger, 2018. TPStream: Low-Latency temporal pattern matching on event streams. Open Proc., 1: 313-324.

Lugmayr, A., B. Stockleben, C. Scheib and M.A. Mailaparampil, 2017. Cognitive big data: Survey and review on big data research and its implications: What is really new in big data?. J. Knowl. Manage., 21: 197-212.

Sandha, S.S., M. Kachuee and S. Darabi, 2017. Complex event processing of health data in real-time to predict heart failure risk and stress. Comput. Soc., 1707: 1-5.

Storey, V.C. and I.Y. Song, 2017. Big data technologies and management: What conceptual modeling can do. Data Knowl. Eng., 108: 50-67.

Tao, F., J. Cheng, Q. Qi, M. Zhang and H. Zhang *et al*., 2018. Digital twin-driven product design manufacturing and service with big data. Intl. J. Adv. Manuf. Technol., 94: 3563-3576.

Terzi, D.S., R. Terzi and S. Sagiroglu, 2017. Big data analytics for network anomaly detection from netflow data. Proceedings of the 2017 International Conference on Computer Science and Engineering (UBMK), October 5-8, 2017, IEEE, Antalya, Turkey, ISBN:978-1-5386-0931-6, pp: 592-597.

Viegas, E., A. Santin, N. Neves, A. Bessani and V. Abreu, 2017. A resilient stream learning intrusion detection mechanism for real-time analysis of network traffic. Proceedings of the 2017 IEEE International Conference on Global Communications GLOBECOM 2017, December 4-8, 2017, IEEE, Singapore, Singapore, ISBN:978-1-5090-5020-8, pp: 1-6.

Wang, Y., L. Kung and T.A. Byrd, 2018. Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. Technol. Forecasting Soc. Change, 126: 3-13.

Williams, M.L., P. Burnap and L. Sloan, 2017. Crime sensing with big data: The affordances and limitations of using open-Source communications to estimate crime patterns. Br. J. Criminology, 57: 320-340.

Zhuang, C., J. Liu and H. Xiong, 2018. Digital twin-based smart production management and control framework for the complex product assembly shop-floor. Intl. J. Adv. Manuf. Technol., 96: 1149-1163.