International Journal of Recent Engineering Science (IJRES) Volume 7 Issue 2, 1-7, Mar-Apr 2020
ISSN: 2349 – 7157 / doi: 10.1445/23497157/IJRES-V7I2P101

Design and Development of Honey Bee Behavior Excited Modern Quality Constructed Entry Controller using Bell-Lapadula Paradigm in Cloud Computing Methodology

Dr. S. Ravichandran[1], R. Benjohnson[2], K. Ramanathan[3]
[1](Computer Science Department, AnnaiFathima College of Arts & Science, Madurai, India)
[2](Computer ApplicationsDepartment,Coimbatore Institute of Management & Tech.,Coimbatore,India)
3(Computer Science Department, AnnaiFathima College of Arts & Science, Madurai, India)

Abstract
Distributed computing is one of the rising advances that is being utilized broadly nowadays. It utilizes the registering assets, for example, equipment and programming that is conveyed over the web and offers remote types of assistance with client's information, programming, and calculation. There has been a developing pattern to utilize the cloud for an enormous scope of information stockpiling. This has raised the significant security issue of how to control and forestall unapproved access to information put away in the cloud. There are different access control methods in cloud conditions, for example, IBAC, RBAC, ABAC, MAC, and DAC. Among these systems, Attribute-Based Access Control (ABAC) is increasing more significance. Here access is allowed dependent on characteristics. Our essential goal is to condense all the entrance control strategies in cloud condition. Our principal objective is to thought of a Novel Attribute-Based Access Control for cloud security utilizing Enhanced Bell-Lapadula Model propelled from Honey Bee behavior. The Honey Bee keeps the interlopers from going into their hives. This is like the entrance control system in cloud condition. It distinguishes the honey bee with a place with a similar hive by the ownership of the little points on the stings. So also, we are attempting to limit the clients dependent on the ownership of the right arrangement of traits by utilizing the ABAC strategy

Keywords–*Attribute-Based Access Control, Bell-Lapadula Model, and Cloud security*

I. INTRODUCTION

The never-ending network improvements on a large scale have dragged up many users online in the past few years. Many network applications have popped up, and users prefer to use several well-known websites to download such applications and other desired content for educational as well as for entertainment purposes. This paves a new way for malicious software to create false contents and enter into the user system and causes malfunctioning of the host system by unauthorized access and stealing of user data. It may cripple the host system and block its normal functions, causing inconvenience to the user.

Hence several software-based algorithms have been proposed and currently used in day to day applications to block such malicious software activities from entering the host system. So the Network Intrusion Detection System was introduced, which prevent the intrusion of malicious software inside the network. These so-called NIDS used in the commercial world have served their purpose well, but in turn, have consumed much power and occupied a vast memory space in the host system. So these computer network-oriented applications have shown similar inefficiencies in usual implementation methods. So the proposed project is focused on improving the efficiency of the NIDS system through effectively optimizing the performance parameters.

The proposed method finds its application in the network intrusion detection system. The system is also used in antivirus software programs to detect and avoid intrusion of malicious programs into the user host system. The proposed method finds significant application in pattern matching of data sequences in the network path. The multi-pattern matching algorithm also uses an increased memory space, but constant power usage and improved performance are visible by using these algorithms. The deterministic finite automata algorithm is mostly used in the pattern matching process, while the non-deterministic finite automata algorithm is solely used for network intrusion detection systems. Some algorithms are used for memory reduction, while some are used to reduce processing time.

In network security, most of the multi-pattern matching solutions today are based on the Aho-Corasick (AC) algorithm. It performs multi-pattern matching in linear time based on constructing a finite state machine to do so.[1] A finite state machine can be efficiently implemented using the Ternary Content Addressable Memory (TCAM) based architecture, where the number of TCAM entries is equal to the number of transitions in the state machine. A TCAM has the "do not care" matching feature, which may reduce the amount of TCAM entries. Gould et al. proposed a way to scale back the amount of TCAM entries by employing a logic minimization method. The proposed paper uses an encoding scheme called a Covered State Encoding (CSE), which takes advantage of the “do not care” feature of TCAMs in the TCAM-based implementation of the AC algorithm. Since the received information of failure transitions in the Non-Deterministic Finite Automata (NFA) of the AC algorithm is implicitly captured in the covered state encoding, the failure transition entries are often eliminated within the proposed scheme, and one can further substantially reduce the memory requirement. The pattern matching can be performed using either the NFA or the Deterministic Finite Automata (DFA). To match a string, one starts from the initial state (usually 0). If a go-to transition or a cross transition is matched with input within the current state, the present state is moved along the matched transition.

II. RELATED WORK

Identification motors fit for examining bundle payloads for application-layer organize data are desperately required. The most significant innovation for quick payload investigation is an effective multi-design coordinating calculation, which performs exact string coordinating among parcels and an enormous arrangement of predefined designs. The paper proposed a novel Enhanced Hierarchical Multipattern Matching Algorithm (EHMA) for parcel review. In light of the recurrence of grams, a little arrangement of the most continuous grams is found and utilized in the EHMA.

EHMA is a two-level and bunches quick coordinating calculation, which fundamentally decreases the measure of external memory, gets to, and the limit of memory [2]. Utilizing a skippable sweep procedure, EHMA accelerates the examining procedure. Moreover, autonomous of equal and extraordinary capacities, EHMA is straightforward and along these lines commonsense for both programming and equipment usage. Reenactment results uncover that EHMA essentially improves the coordinating presentation. The speed of EHMA is about O.89-multiple times quicker than that of current coordinating calculations. Significantly under simple exceptional assault, EHMA still performs well.

System interruption location framework is utilized to review bundle substance against many predefined vindictive or suspicious examples. Since conventional programming alone example coordinating methodologies can never again meet the high throughput of the present systems administration, numerous equipment approaches are proposed to quicken design coordinating. As equipment draws near, the memory-based design has pulled in a ton of consideration in light of its simple configurability and scalability [3]. To oblige the expanding number of assault examples and meet the throughput necessity of systems, a useful system interruption identification framework must have a memory-proficient example coordinating calculation and equipment plan. The paper proposed a memory-effective example coordinating calculation which can fundamentally lessen the memory prerequisite. For Snort rule sets, the new calculation accomplishes 21% of memory decrease contrasted and the standard Aho-Corasick calculation. What is more, an addition of 24% in memory decrease is accomplished by this way to deal with the bit-split calculation, which is the cutting edge memory-based methodology.

Interruption Detection frameworks (IDS) were created to recognize and report assaults in the late 1990s, as programmer assaults and system worms started to influence the web. Conventional IDS advances distinguish unfriendly traffic and send cautions yet do nothing to stop the assaults. System Intrusion Prevention Systems (NIPS) are sent following the system portion being ensured. As the traffic goes through the NIPS, it is examined for the nearness of an assault. Like infections, most interloper exercises have a type of mark. In this manner, a design coordinating calculation dwells at the core of the NIPS. At the point when an assault is recognized, the NIPS obstruct the culpable information. There is a supposed exchange off between the exactness of location and algorithmic productivity. Both are principal in guaranteeing that authentic traffic is not deferred or upset as it moves through the gadget. Consequently, the example coordinating calculation must have the option to work at wire speed while distinguishing the principle heft of interruptions. With systems administration speeds multiplying each year, it is getting progressively challenging for programming based answers to stay aware of the line rates. [4]The paper proposed a novel example of coordinating calculation. The calculation utilizes a Ternary Content Addressable Memory (TCAM) and is equipped for coordinating different examples in a solitary activity. The calculation accomplished line-rate speed of a few sets of greatness quicker than current works while achieving comparable identification accuracy. Besides, the framework is entirely perfect with Snort’s guidelines punctuation, which is the real standard for interruption anticipation frameworks. Various strings coordinating that can look at several string designs simultaneously is a primary segment of these frameworks and is a very much examined problem [5]. A large portion of the string coordinating arrangements today depend on the exemplary Aho-Corasick calculation, which has an inborn restriction; they can process just one information character in one cycle. . Furthermore, the proposed novel advancements utilize the properties of TCAMs to lessen the memory necessities of the proposed calculation fundamentally. The last introduction shows broad reproduction consequences of the system based substantial bundle location utilizing Modalism and Xilinx recreation devices to delineate the proposed plan's adequacy.

A. Air conditioning DFA Matching Algorithm

Most multi-pattern coordinating arrangements today depend on the Aho-Corasick (AC) calculation. It performs multi-pattern coordinating indirect time, dependent on building a limited state machine to do as such. A limited state machine can be productively executed utilizing TCAM-based engineering, where the quantity of TCAM sections is equivalent to the number of advances in the state machine[1]. A TCAM has the “could not care less" coordinating element, which can be utilized in diminishing the quantity of TCAM sections. Gould et al. proposed a strategy to decrease the quantity of TCAM sections by utilizing a rationale minimization technique, so a proposed state encoding plan misusing the "could not care less" component of TCAMs, where a few changes are dispensed within the TCAM passages. Their strategies, in any case, have constraints in diminishing the number of sections since they depend on a Deterministic Finite Automaton (DFA) of the AC calculation, which has considerably more changes than the comparing Non-Deterministic Finite machine (NFA). The NFA can be actualized with littler TCAM sections than the DFA even though it makes a more significant number of changes than the DFA during the example coordinating because of disappointment advances. A proposed encoding plan called a secured state encoding, which exploits the "could not care less" component of TCAMs in the TCAM-based execution of the AC calculation. Since the data of disappointment advances in the NFA of the AC calculation is undoubtedly caught in the secured state encoding, the disappointment change sections can be totally wiped out in the proposed plan, and one can assist significantly lessen the memory necessity. The AC calculation first builds a limited state machine from the arrangement of examples and utilizes the limited state machine to process the content string in a solitary pass[1]. The calculation manufactures a non-deterministic limited robot by building the go-to and disappointment changes and afterward changes over it into a deterministic limited machine. A DFA has a more significant number of changes than the relating NFA. The limited state machines depict the usefulness of both the current just as the proposed frameworks. In the current framework, the real example coordinating is advised out as appeared in the figure.

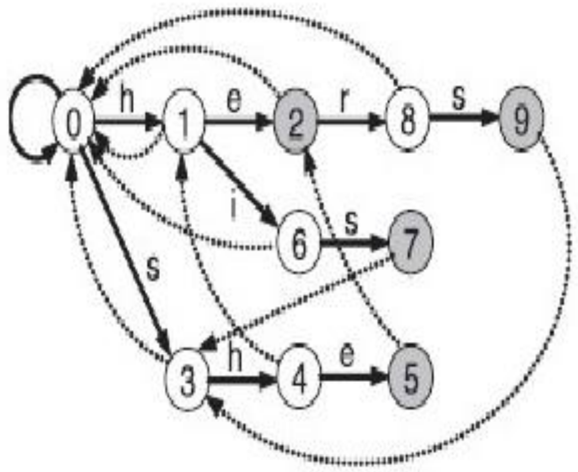

Fig1. DFA AC Design coordinating FSM

In the DFA, the stippled lines speak to changes, called cross advances, which are recently included by taking out disappointment advances. Concealed states speak to the example coordinating states called yield states. The nominal advances going to the underlying state are overlooked in the figure. To coordinate a string, one beginning from the underlying state (generally 0). On the off chance that go-to progress or cross changes are coordinated with a contribution to the present express, the present state is moved along the coordinated change. Something else, for a DFA-based coordinating, the present state returns to the underlying state, and the coordinating procedure rehashes for the following info. The DFA inspects each information just once. Subsequently, the AC calculation has a deterministic execution time[1]. The AC DFA requires much memory in a straightforward RAM-based usage that keeps a table of pointers to the next states for each contribution since the table contains additionally little changes that return to the underlying state. These techniques lessen the memory necessity in the RAM-based execution. The AC DFA can be actualized more proficiently utilizing a TCAM since it needs just nontrivial changes. The TCAM based multi-byte numerous string coordinating calculation has restricted help for trump cards.

B. Air conditioning NFA Algorithm

The proposed framework utilizes AC NFA calculation to improve the exhibition of the system interruption location framework just as comparative courses of action made for improving the presentation of the example coordinating framework as portrayed follows. Air conditioning - NFA: The NFA can be executed with littler TCAM passages than the DFA even though it makes many advances than the DFA during the example coordinating because of disappointing advances. The proposed paper exhibits a state encoding plan called a secured state encoding, which exploits the "could not care less" component of TCAMs in the TCAM-based execution of the AC calculation.

NFA

cs	in	ns	F
8	s	9	0
6	s	7	0
4	e	5	0
3	h	4	0
2	r	8	0
1	e	2	0
1	i	6	0
0	h	1	0
0	s	3	0
9	*	3	1
7	*	3	1
5	*	2	1
4	*	1	1

covered state encoding

	cs	in	ns
0	8	s	9
1	6	s	7
2	4	e	5
3	3, 7, 9	h	4
4	2, 5	r	8
5	1, 4	e	2
6	1, 4	i	6
7	all states	h	1
8	all states	s	3

cs : current state
in : input
ns : next state
F : failure transition

Table1. NFA state transition using covered state encoding

Since the data of disappointment advances in the NFA of the AC calculation is undoubtedly caught in the secured state encoding, the disappointment change passages can be wiped out in the proposed plan, and one can advance significantly to decrease the memory necessity. The AC calculation first develops a limited state machine from the arrangement of examples and afterwards utilizes the limited state machine to process the content string in a solitary pass.

The calculation manufactures a non-deterministic limited machine by building the go-to and disappointment changes and afterwards changes over it into a deterministic limited robot. A DFA has a more significant number of advances than the relating NFA. In the NFA, the strong lines and stippled lines speak to and disappointment advances, individually. For an NFA-based coordinating, the present state is moved along its disappointment change, and the coordinating procedure rehashes for the present information. The DFA looks at each information just once, while the NFA may analyze each information a few times along the disappointment change way.

In coordinating a content string of length n, the DFA makes n state changes, and the NFA makes less than 2n state advances. Consequently, the AC calculation has a deterministic execution time. The AC DFA requires much memory in a straightforward RAM-based usage that keeps a table of pointers to the next states for each contribution since the table contains additionally minor changes that return to the underlying state. Tan et al. and Lunteren proposed techniques diminishing the memory necessity in the RAM-based usage.

C.Engineering of a simple TCAM

The Architecture of a basic TCAM clarifies the usefulness of non-deterministic limited automata calculation from the above state chart. The engineering comprises of a TCAM, SRAM, and a rationale. Each TCAM passage speaks to a query key, which comprises of current state and input, and has relating information, which is the following state, in the SRAM whose address is given by the TCAM yield.

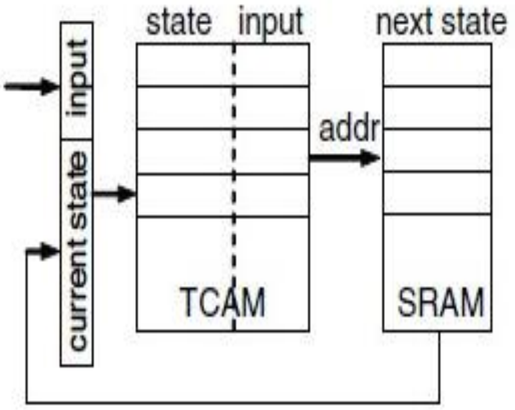

Fig2. The architecture of a simple TCAM

Two registers' current state and information are introduced to the state 0 and the info cushion's beginning information, separately. On the off chance that there is a coordinating section for the state and information esteem in the TCAM, the TCAM yields the coordinating passage file, and afterwards, the SRAM yields the following state information situated in the comparing area. Since a TCAM has "could not care less" bits, numerous sections can be at the same time coordinated, and when this is the situation, the record of the central coordinated passage is yielded. On the off chance that there is no such match in the TCAM, the following state is the underlying state.

At each state's progress, info is progressed to the following information, and the following state esteem is put away into the present state register. Each TCAM section speaks to progress in the state machine. The quantity of TCAM passages is equivalent to the number of advances and autonomous of the number of states.

D. Merge Algorithm

Consolidation calculations are a group of calculations that run successively over different arranged records, regularly creating increasingly arranged records as yield. This is appropriate for machines with tape drives. Use has declined because of enormous irregular access recollections, and numerous uses of consolidation calculations have quicker choices when an arbitrary access memory is accessible

Hence inTCAM based memory application systems, the merging algorithm performs better by improving the processing time and at the same time improving the memory utilization of the overall system for the same power consumption at no additional cost.

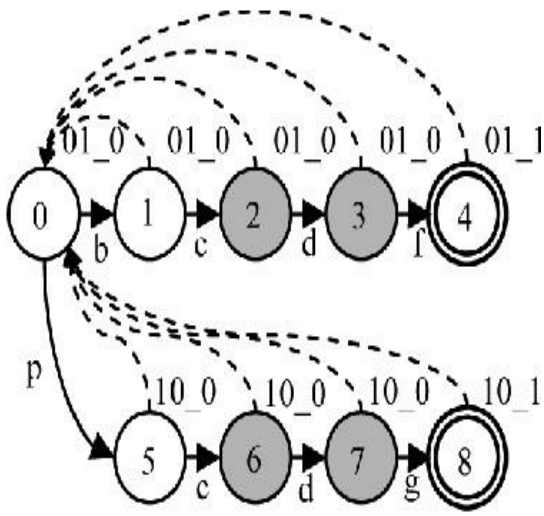

Fig3. Multi-pattern matching state diagram by AC algorithm

The multi-pattern matching algorithm, as per Ahocorasik, is that for each failure in pattern matching, a new transition is established to reduce the data losses that may occur during network delay in data packet transition. Hence the failure transition blocks detect the actual data packets that exist in the port of end-user, thereby reducing the data losses.

Nevertheless, the drawback of the existing system in pattern matching is that the number of state transitions increases with an increase in data packet rate, thereby simultaneously increasing the load on the network and the NIDS system. The number of transitions and the number of states used by the pattern matching sequences are directly proportional to each other.

The merging algorithm concept was introduced to reduce the load on the network by reducing the number of transition paths unnecessary to the network

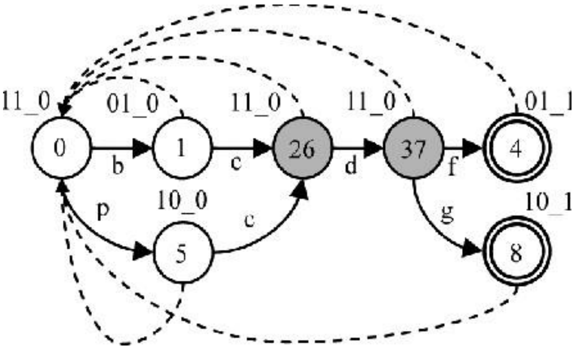

Fig4. Multi-pattern matching state diagram by merging algorithm

So the merging algorithm concept is used to avoid unnecessary state transitions and the number of states required by the NIDS system.

III. IMPLEMENTATION

. The actual implementation of the multi-pattern matching process involves the simulation tools of Xilinx and Modalism software for determining the area, power, and processing time for each input pattern.

The shaded circles denote the intermediate states, while the double circled denote the final states. Here the state transitions are more in number, and their corresponding intermediate states are equally more in number than single pattern matching as explained before. The patterns that go via a similar path for the particular data sequence is taking more transition paths than necessarily required by the state cycle in determining the input data sequences. The resulting paths taken by the pattern matching system is time-consuming and more memory is utilized for determining the data sequences. Hence the Merging algorithm brings similar states in a single state by combining the similar transition paths taken by the state machine as shown below.

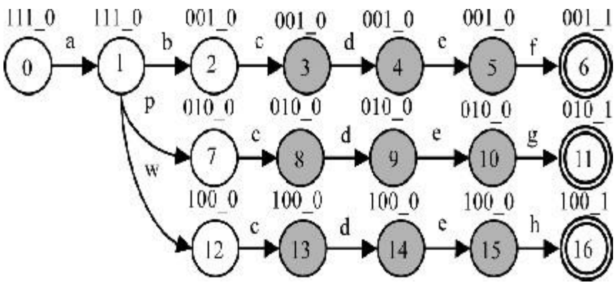

Fig5. Multi-pattern matching state diagram without modification

By giving the respective inputs using the modalism simulation tool, the system checks for data patterns in the database with the incoming input data pattern. If the pattern matches, then the NIDS system displays as a "valid pattern", and other additional systems for packet validation will determine whether to allow it or not based on the Network Intrusion Detection system. If the pattern at the input port does not match with those patterns in the database, then it displays as a pattern not matched, and the input data pattern will be rejected.

The merging algorithm also performs in an way mentioned above, like the traditional AC algorithm, but the purpose of merging algorithm is to effectively utilize the area through reducing the excessive states in the process of multi-pattern matching. This is done by merging the states of the above-shaded portions to a single state, reducing the overall transition states as well as the number of transition paths as shown below.

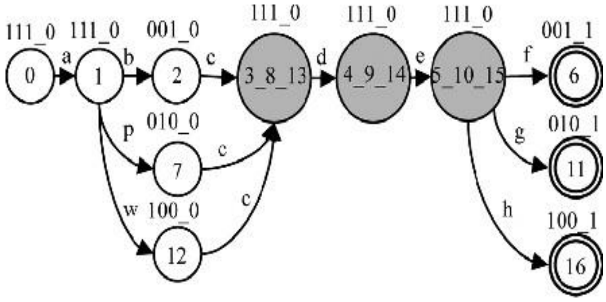

Fig6. Multi-pattern matching state machine diagram with reduced transition states

The above state transition diagram explains the concept behind the Merging algorithm. The states (3,8,13), (4,9,14),(5,10,15) are all combined(merged) into a single state, thereby reducing the overall states from 16 states to 10 states. This improves the overall performance of the system increases as the load in the NIDS system decreases consistently.

IV. SIMULATION AND RESULTS

The simulation is divided into2 categories, one for performing AC modified DFA algorithm, and the other for Merging algorithm. By comparing the performance parameters of both the methods, the needs of performance and memory management can be varied as per user's circuit design needs.

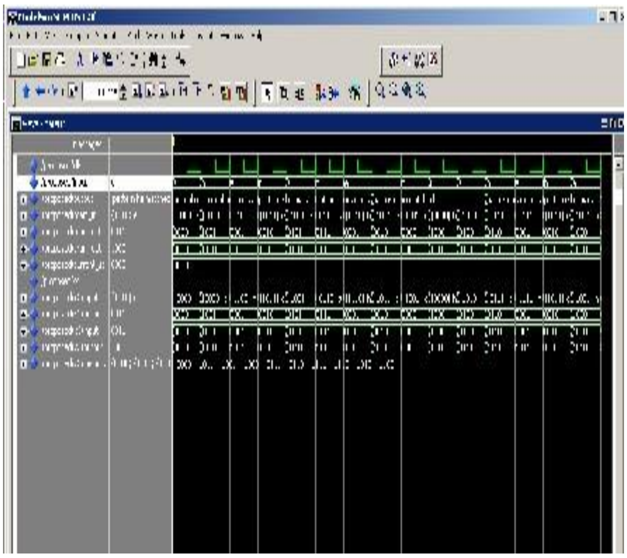

Fig7. Timing graph of pattern matching in CS NFA algorithm

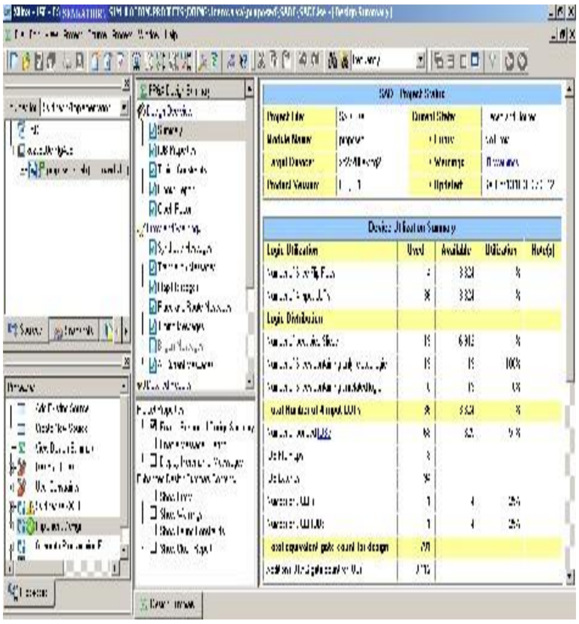

Fig8. Area and memory usage of modified DFA Algorithm

The modified DFA algorithm which influences the trade-off principle has performed the similar tasks, and the same level has maintained the power consumption, and only 0.08mV variation has been observed, while the processing time has been dramatically decreased by400ns. Thus performance has been increased with the same power consumption, with an increase in TCAM memory. Two TCAM memory units are utilized for the modified DFA algorithm simulation. Nevertheless, if the modified DFA algorithm is used, then only791 logic gates are used compared to 899 logic gates in NFA algorithm simulation. So performance and memory are proportional to one another.

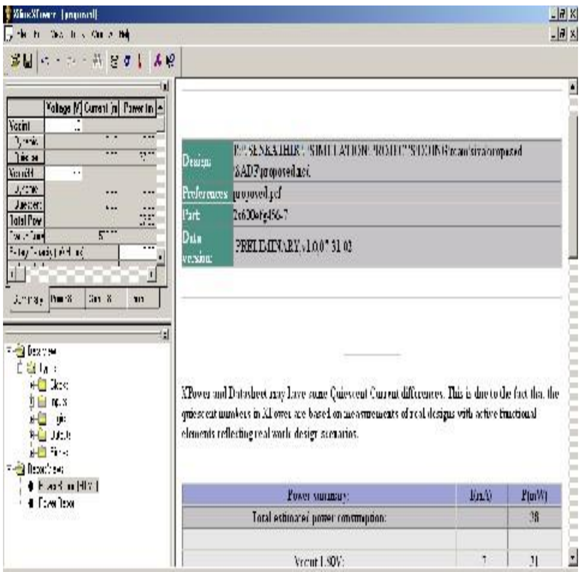

Fig9. Power analysis of modified DFA Algorithm

The power consumption by Xilinx software is 38mW, and the area utilization is 27% with 899gates, but the modified DFA algorithm gives791gates per chipareawith25% area utilization based on the existing method.

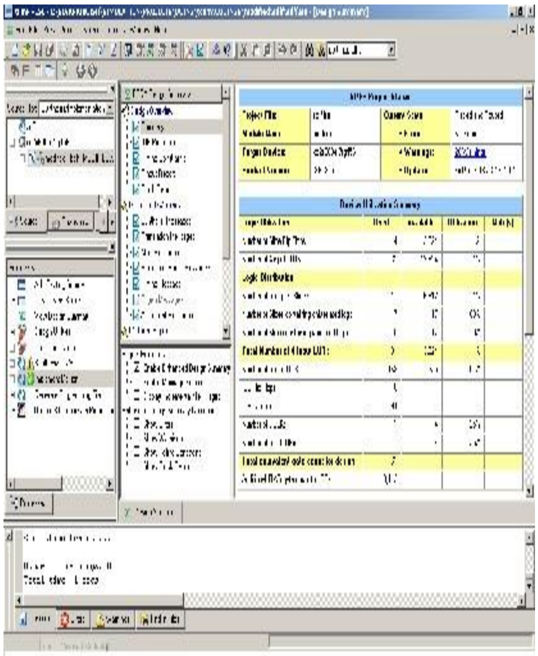

Fig10. Area Analysis of multi-pattern matching by merging algorithm

The area utilization of the merging algorithm is less when compares to that of Modified DFA algorithmbyalmost3-5% lesser than the existing method.

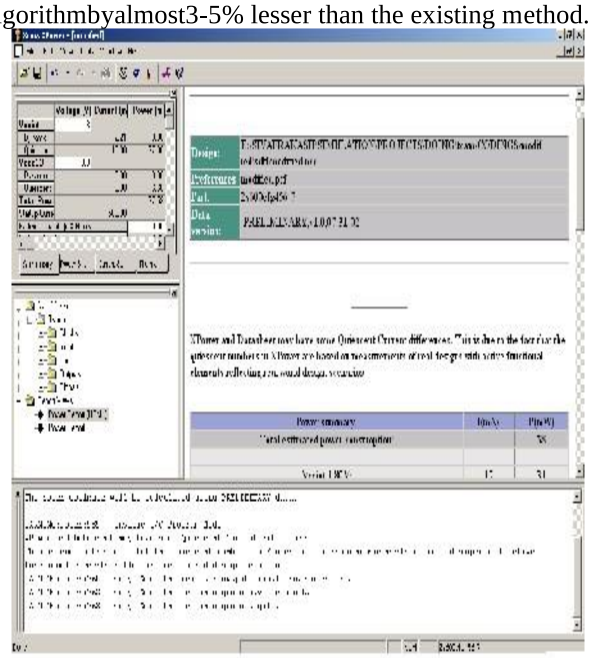

Fig11.Power Analysis of multi-pattern matching by merging algorithm

The power consumption remains the same but with improved processing time.

V. CONCLUSION

In this final proposed paper, the AC modified algorithm proves useful for multi-pattern matching in performance and power management, but area efficiency is not considered. In contrast, the Merging Algorithm improves the overall performance not only by reducing the power consumption but also the processing time by reducing the number of transition states and the corresponding transition paths. This is deemed as the better for multi-pattern matching process in Network Intrusion detection system in improving performance and optimizing the power consumption as compared to conventional and traditional methods of pattern matching.

.

ACKNOWLEDGEMENTS

The authors are thankful to D. Sahithi, Dr Harini, and R.A. Chowdhury for providing the necessary facilities for the preparation of the paper. Also, thanks to the IJRES Journal staff to publish this paper.

REFERENCES

[1] SankyunYun., An Efficient TCAM based Implementation of Multipattern Matching usingCovered StateEncoding, IEEETransactionson Computers. 61(2) (2012) 213.
[2] D.Sahithi Dr.Harini., Enhanced Hierarchical Multipattern Matching Algorithm for deep packet Inspection, International Journal of Computer Science and Information Technology&Security.2(3) (2012) 524.
[3] AnatBremler-Barr and YaronKoral., AcceleratingMulti-Pattern Matching on Compressed HTTP Traffic, International Conference, Blavatnik School of Computer Sciences Tel-Aviv University,Israel. (2011) 1.
[4] ChengHungLin,ShihChiehChan., AnEfficient Pattern Matching Algorithm ForMemory Architecture ,IEEE Transactions on Very LargeScale Integration.19(1) (2011).
[5] Po-Ching Lin, Ying-Dar LinandYuan- ChengLai., A Hybrid Algorithm of Backward Hashing and Automaton Tracking for virus scanning, IEEE international Conference, Taiwan University, Computer Science. 60(4) (2011) 33.
[6] Raravi, Gurulingesh, Sharma, Neera, Ramamritham, Krithi, and Malewar, Sachitanand., Efficient real-time support for automotive applications: A case study. In Proceedings of the 12th IEEE International Conference on RTCSA. (2006) 335-341.
[7] R. A. Chowdhury, et al., Oblivious algorithms for multicores and network of processors, in Parallel & Distributed Processing (IPDPS), IEEE International Symposium. (2010) 1-12.
[8] G. M. Amdahl., Validity of the single processor approach to achieving large scale computing capabilities, presented at the Proceedings of the April 18-20, spring joint computer conference, Atlantic City, New Jersey. (1967).
[9] Aggarwal and S. V. Jeffrey., The input/output complexity of sorting and related problems, Commun. ACM. 31 (1988) 1116-1127.
[10] Uno, A. Sakaguchi, T. and Tsugawa, S., A merging control algorithm based on inter-vehicle communication, In IEEE International Conference on Intelligent Transportation Systems. (1999) 783-787.
[11] M.Sharada Varalakshmi., An Introduction to Multilevel Security in Cyber Physical System, SSRG International Journal of Computer Science and Engineering 4(7) (2017) 26-29.