ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

A Systemfor High Throughput Performanceand Reduce Low Memoryusing Pattern Matching with Hash Key

[I]Viswanathan.V, [II]Suresh.P
[I]PG Student, [II]Associate Professor
[I,II]Dept. of CSE, Sethu Institute of Technology, Affiliated Anna University Kariapatti

Abstract

Pattern matching is one of the most critical elements because it allows for the system to make decisions based not just on the headers, but the actual content flowing through the network. Network Intrusion detection and Prevention systems have emerged as one of the most effective ways of providing security to those connected to the network, and at the heart of almost every modern intrusion detection system is a pattern matching algorithm. I have developed an approach that relies on a special purpose architecture that executes novel pattern matching algorithms specially optimized for implementation in our design.

Keywords

Pattern Matching, Hash Key, Resource allocation, Auction, Utility Computing

I. Introduction

In the recent decades, network security is very important to avoid the unauthorized access. Network security consists of the policies and provision adopted by administrator to prevent the unauthorized access, monitoring the network by maintaining traffic protecting the data and network resources.Pattern matching techniques have been recently applied to network security application such an intrusion detection virus production and spam filter. In a wider sense the techniques has been using for information retrieval, text editing by applying signature matching and to help detect malicious attackers against network. Pattern matching used in data mining for information retrieval.But are not suitable for matching multiple.In traditional symmetric-key encryption schemes, a user encrypts a message so that only the owner of the corresponding secret key can decrypt it. Decryption is all-or-nothing"; that is, with the key one can decrypt the message completely, and without the key one learns nothing about the message. Pattern matching is the act of checking a perceived sequence of tokens for the presence of the constituents of some pattern. In contrast to pattern recognition, the match usually has to be exact. The patterns generally have the form of either sequences or tree structures. Uses of pattern matching include outputting the locations (if any) of a pattern within a token sequence, to output some component of the matched pattern, and to substitute the matching pattern with some other token sequence (i.e., search and replace).Using a traditional encryption scheme, any global query on the data would require retrieving and decrypting the entire original data, negating many of the advantages of cloud storage. Then, using the secret key, the client can issue a query q by executing an interactive protocol with the server. For example, for pattern matching queries, a query q is a pattern string, the message M is a string, and F (M; q) returns the set of indices of all occurrences of q as a substring of M.For security, i think of the server as an adversary trying to learn information about the message and the queries. Ideally, I would like that an adversary that is given a cipher text and that engages in query protocols for several queries learns nothing about the message or the queries. Security that specifies explicitly what information is leaked, and guarantee that an adversary learns nothing more than the speed leakage. In the malicious model, the adversary tries to learn information, possibly by not following the protocol honestly. Pattern matching encryption. Constructing a pattern matching encryption scheme, that is, a query able encryption scheme that support pattern matching queries {given a string s and a pattern string p, return all occurrences of p as a substring of s. In the genomic data application, for example, researchers may wish to query the database to determine whether a particular cancer marker sequence appears in any of the data.For efficiency, our goal is for space and computation complexity to be comparable to that of evaluating pattern matching queries in the unencrypted setting. By focusing on the specific functionality of pattern matching queries and achieves a scheme with much better efficiency. To construct a pattern matching encryption scheme, use suffix trees, a data structure used to efficiently perform pattern matching on unencrypted data. Combine basic symmetric-key primitives to develop a method that allows traversal of select edges in a suffix tree in order to efficiently perform pattern matching on encrypted data, without revealing significant information about the string or the queries.

To obtain High throughputs (stateful and parallel processing) to reduce Storage of Aho-Corasick algorithm Stateful nature of Aho-Corasick algorithm reduce repeated pattern matching. It stores the previous results. Parallel processing provides High speed. The existing algorithms are efficient for single-pattern matching, but are not suitable for matching multiple patterns. The AC algorithm pre-processes the patterns builds a finite automation that can match multiple pattern simultaneously, but may require a huge amount of memory space to do so. However, even with the linear-time worst- case performance guarantee, the throughput of the AC algorithm cannot keep up with transmission speeds in high-speed networks. The proposed verification engine, which is modification of the ac automation, checks all candidate patterns simultaneously rather than sequentially. Proposed stateful pre-filter is suitable for pattern of moderate or large lengths. The stateful pre-filter concept and present an AC-based verification engine that can check all candidate patterns simultaneously.

II. Related Work

The existing algorithms are efficient for single-pattern matching, but are not suitable for matching multiple patterns. The AC algorithm pre-processes the patterns builds a finite automation that can match multiple pattern simultaneously, but may require a huge amount of memory space to do so.however, even with the linear-time worst- case performance guarantee, the throughput of the AC algorithm cannot keep up with transmission speeds in high-speed networks.

A. Literature Review

Bremler-Barr,Y.Harchol,etal(2011) has presented "Space-time tradeoffs in software-based Deep packet inspection". In this paper the state elimination procedure is proposed state elimination procedure is following if:1)it is a single-child state.2)there is no incoming failure transition to it.Another effective compression technique called leaf compression, eliminates leaf states with two modification;1)Pushing the indication of the match to the penultimate state.2) Copying the failure transitions of leaf states to the corresponding penultimate states as their new goto transitions. This compression technique use to reduces the transition table section and achieving 60% comparable throughput.

Y.E.Yang,H.Le et al(2010)has presented "High performance dictionary-based string matching for deep packet inspection". This paper describes about throughput of pattern matching.AC algorithm cannot keep up with transmission speeds in high speed networks in WM(Hash-AV Clam-AV)algorithm pre-filter was implemented as a shift table, and the verification engine checks all candidate pattern sequentially when a potential starting position is identified. The proposed architecture consists of pre-filter and verification engine. Once a suspicious starting position is found, the verification engine conform true pattern occurrence.

N.S.Artan and H.I.Chao et al(2008)has presented"Boundary Hash for Memory-Efficient Deep Packet Inspection Network Intrusion Detection and Prevention Systems(NIDPS)are critical for network security". Deep Packet Inspection (DPI) is at the heart of these NIDPSs.DPI is the detection of malicious packets by comparing the packet payloads against excerpts from known intrusion packet. I refer this paper for signature verification process.

K.Sinkar,J.Patel,et al(2007)has presented "Aggregated bloom filter for intrusion detection and prevention hardware".Aggregated Bloom Filter(ABFs) to increase the throughput and scalability of BFs. The proposed ABF has two methods to improve average speed and scalability. First, to remove redundancy, the hash functions for each query are calculated sequentially. As soon as I have a no match in any of the hash results, the query is immediately abandoned. I refer this paper for our proposed Pre-filter construction.

III. Proposed System

The proposed verification engine, which is modification of the ac automation, checks all candidate patterns simultaneously rather than sequentially. Proposed stateful pre-filter is suitable for pattern of moderate or large lengths. The stateful pre-filter concept and present an AC-based verification engine that can check all candidate patterns simultaneously.

A. Resource Allocation

In the efficient and responsive economic resource allocation in high-performance computing environments. While economic resource allocation provides a well-studied and efficient means of scalable decentralized allocation it has been stereotyped as a low performance solution due to the resource commitment overhead and latency in the allocation process. The high utilization strategies proposed in this paper are designed to minimize the impact of these factors to increase occupancy and improve system utilization.

B. Utility Computing

Utility computing models have long been the focus of academic research, and with the recent success of commercial cloud providers, computation and storage is finally being realized as the fifth utility. Computational economies are often proposed as an efficient means of resource allocation. However adoption has been limited due to a lack of performance and high overheads. In this paper, we address the performance limitations of existing economic allocation models by defining strategies to reduce the failure and reallocation rate, increase occupancy and thereby increase the obtainable utilization of the system.

C. Auction

Reducing the duration of the auction. The problem with this approach is that there is minimal time for providers to discover the auction and to compute their bids.. Bid as late as possible. The advantage with this approach is that providers can compute their bids with the most up to date resource state and resources are reserved for a shorter time. The primary problem with this approach is time sensitivity, the auction can be missed if the bid is too late or experiences unexpected network delays.

Existing Architecture System

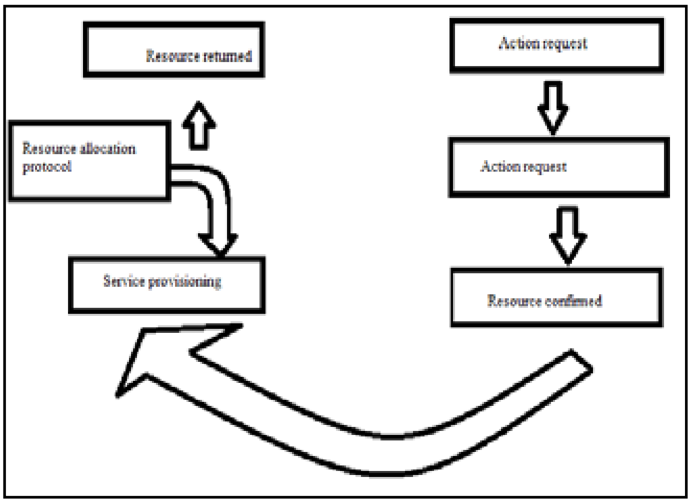


Fig. 3.1 existing architecture

Proposed Architecture System

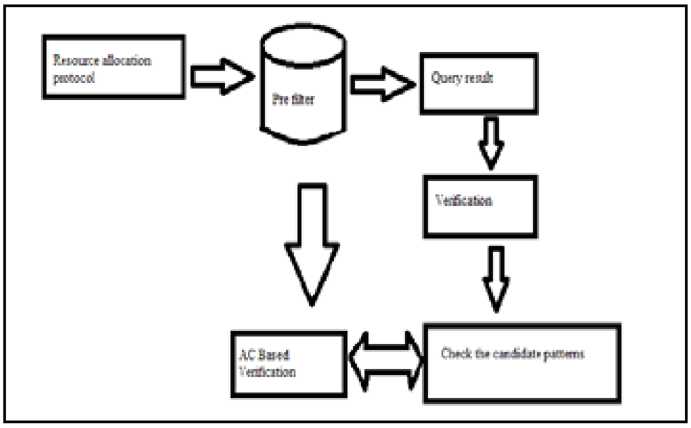


Fig. 3.2 proposed architecture

IV. Performance Evolution

This graph shows the packet delivery ratio. The packet delivery ratio graph consists of time as X axis and delivery ratio as a Y axis. In existing system, when the time is increase the delivery ratio is said to be reduces. In the proposed system, when the time is increase the delivery ratio is said to be increases. So in the proposed system the delivery of packet is said to be faster and higher than the existing system.

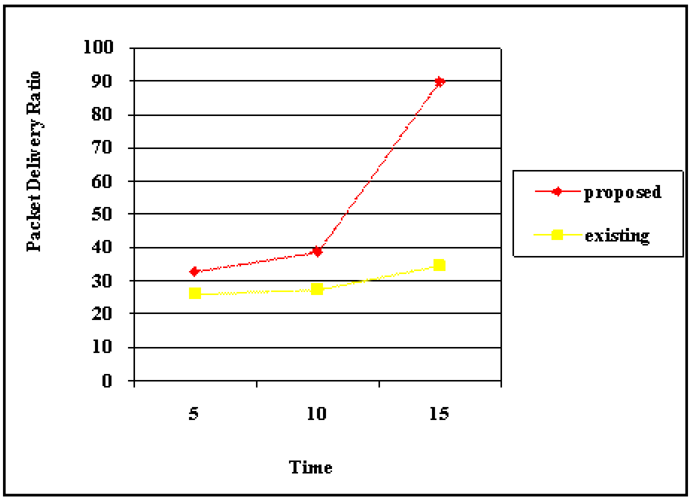


Fig. 4.1 packet delivery ratio

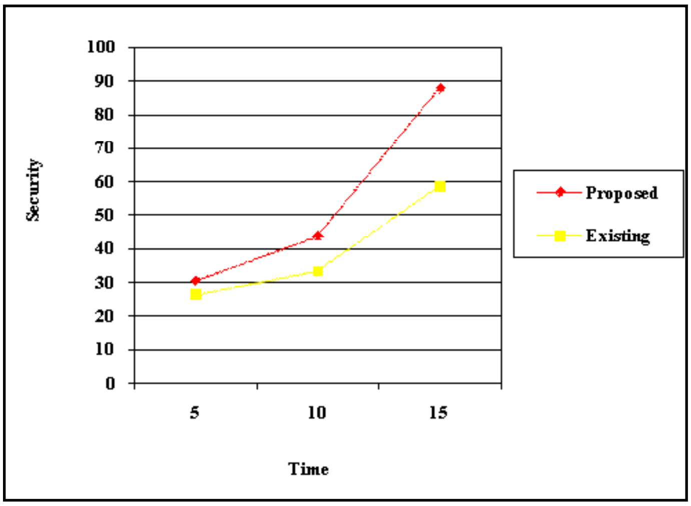


Fig 4.2 security

This graph shows the security performance. The security graph consists of time as X axis and security valve as a Y axis. In existing system, when the time increase the security is said to be reduces. In the proposed system, when the time is increases the security is also said to be increases. So in the proposed system the data is said to be well secured than the existing system.

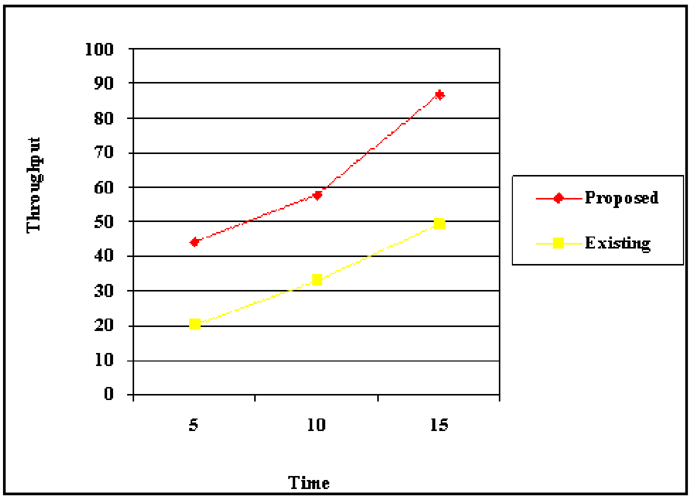
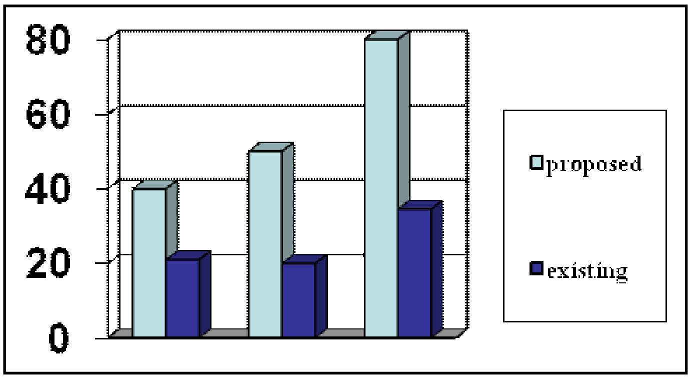


Fig 4.3 Throughput

This graph shows the security throughput performance. The throughput graph consists of time as X axis and throughput valve as a Y axis. In existing system, when the time increase the throughput is said to be reduces. In the proposed system, when the time is increases the throughput is also said to be increases. So in the proposed system the throughput is said to be higher than the existing system.



Packet delivery Security Throughput

Fig. 4.4 Comparison between Proposed and Existing system

Table 1. Comparison between Proposed and Existing System

System	Packet Delivery	Security	Throughput
Existing	22.5	20.4	34.2
Proposed	38.7	52.3	79.8

V. Conclusion

In my paper,I propose a pattern- matching framework in that, requirement of memory is low and throughput of my system is high. In my framework, I introduce the concept named as stateful pre-filter and I introduce an AC-based verification engine, its used to verify all applicant patterns concurrently. The stateful framework performs the finest in both memory requirement and throughput. The Proposed stateful framework is the favoriteresult for both memory requirements and throughput. A well throughput performance is offered by larger search window. The length of the shortest pattern is used to increase the length of the search window.

VI. Future Enhancement

In future the throughput is said to increase and even the memory is low the message is easily transferred by using the hash key algorithm. Hash key used for encrypt and decrypt. In this attackers are not able to easily access the data. The individual strategy, and the combination of the different strategies, was shown to dramatically improve occupancy and utilization in a high performance situation. The increase in allocation rate was shown to be up to 286 percent for the dynamic workloads and 109 percent for the batch model.

VII. Acknowledgement

References

[1] A.Bremler-Barr,Y.Harchol,"Space-time tradeoffs in software-based deep packet inspection" in Proc.IEEE

HPSR,2011,pp.1-8.

[2] Y.Yang,H.Le"Highperformance dictionary-based string matching for deep packet inspection,"inProc.IEEE INFOCOM,2010,pp.1-5.

[3] N.S.Artan,M.Bando,"Boundary hash for memory efficient deep packet inspection,"in Proc. IEEE ICC,May 2008,pp.1732-1737.

[4] M.Bando,H.J.Chao,"Highlymemory-efficient loglog hash for deep packet inspection,"inProc.IEEE GLOBECOM,2008,pp.1-6.

[5] Y.Haowei and H.J.Chao,"A dynamic load balanced hashing scheme for networking applications,"inProc.IEEE GLOBECOM,2008,pp.1-6.

[6] K.Sinkar,J.Patel,"Aggregated bloom filters for intrusion detection and prevention hardware,"inProc.IEEE GLOBECOM,Nov.2007,pp.349-354.

[7] N.S.Artan and H.J.Chao,"Trie bitmap content analyzer for high-speed network intrusion detection,"inProc. IEEEINTOCOM,May 2007,pp125-133.

[8] S.Dharmapurikar and J.Lockwood,"fast and scalable pattern matching for network intrusion detection systems,"IEEE Oct.2006,pp1781-1792.

[9] L.Tan and T. Sherwood"A high throughput string matching architecture for intrusion detection and prevention,"in Proc. IEEE,2005,pp.112-122.

[10] R.Ghosh and G.Yanchuan,"A 10-Gbps high speed single-chip network intrusion detection and prevention system,"inProc. IEEE GLOBECOM,Nov.2007,pp.343-348.

[11] N.S.Artan and H,J.Chao,"Multi-packet signature detection using prefix bloom filters,"inProc.IEEE GLOBECOM,2005,pp.1811-1816.

Author's Profile and Image

V.Viswanathan received his Bachelor of Technology degree in Information Technology from Anna University, Chennai. He is currently pursuing his Master degree in Computer Science and Engineering from Sethu Institute of Technology, Under the control of Anna university Chennai. My area of interest includes networking.

P.Sureshreceived his Master degree in Computer Science and Engineering from Anna University. He is currently pursuing his PhD degree. He currently is working as an Associate Professor in Sethu Institute of Technology. His area of interest includes data structures and programming languages.