A Network Intrusion Detection System Framework based on Hadoop and GPGPU

Sanraj Bandre[1], Prof. Jyoti Nandimath[2]

[1]M.E. Student, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Affiliated to Savitribai Phule Pune University, Pune, India

[2]Assistant Professor, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Affiliated to Savitribai Phule Pune University, Pune, India

Abstract: ***In IT industry the business data grows exponentially, which results in concern to enhance the security system by implementing effective NIDS (Network Intrusion Detection System).The quick response to detecting intrusion an essential feature of any NIDS system, but due to the huge amount of data obtained from organizations which impacts the performance of NIDS. The reason could be of wide range like network speed, amount of data from servers, and an algorithm which directly or indirectly impact the performance. This paper deals with design consideration of NIDS framework which is based on Hadoop and GPGPU (General Purposed Graphical Processing Unit) approach. The proposed NIDS system handles network traffic through Hadoop Framework and intrusion detection functionality will carry-out by GPGPU. The proposed approach improves the NIDS performance and its capability is to provide quick response to various types of network attacks. We have configured our proposed system with Hadoop Data-platform along with its ecosystems to process large volume of network traffic. We apply NVidia CUDA technology (Compute Unified Device Architecture) the parallel programming model for intrusion detection. In our implementation phase we have analyzed Hadoop framework which is capable to process 1, 2 and 4 Giga bytes of server logs in efficient time of 29.86, 47.09 and 94.96 Seconds. We have further added analytics over intrusion by using PF-ICF (Pattern frequency Inverse cluster frequency) approach.***

Keywords: CUDA, GPGPU, Hadoop, Network Security, NIDS

1. Introduction

In today’s computer industry, information security is the most challenging issue in order to fortify network security. The objective of any NIDS system is to provide quick response from various types of intrusions which detect over a network. The size of business data has been increased on a daily basis, which leads to increase in security issues of the organization. The adoption of NIDS in IT industry to provide an effective countermeasure for mitigating the various type network attacks. The performance of the conventional NIDS systems gets downgraded due to data intensive traffic on it. In the design consideration of proposed work we have addressed implementation details by utilizing Hadoop Data-platform and NVidia CUDA technology. It includes Hadoop data-platform configuration with its eco-system like HBase and Flume to deliver real-time data streaming over a network. The CUDA technology offloads the intrusion detection functionality and coordinates with Hadoop to apply further analytic on intrusion. The computational application of GPU - CUDA is carried in many research field of computer science like data mining, network simulation, cyber forensic and cryptography. The objective of the proposed work is to design high performance NIDS for handling large scale data on a network by effectively utilizing parallel computing CUDA technology. The proposed NIDS is capable to provide a wide scope for statistical analysis over the detected network intrusion and able to process large volume of network traffic.

2. Fundamentals

In today’s computer industry, information security is the most challenging issue in order to fortify network security. The NIDS is a security system which detects and identifies network related attacks and provide monitoring feature to administrator by sending alerts. This system automatically detects an active attempt of the attack or malicious activities that are carried by an intruder over the network [1]. The NIDS work at the network level in order to monitor network traffic and perform packet classification on network traffic. Then network traffic gets aggregated, analyzed and compared with intrusion detection rules called as defined. Intrusion detection systems are broadly classified into two groups NIDS and HIDS. In Network Intrusion Detection Systems (NIDS), the classification is done over network traffic directly, and in Host Intrusion Detection System (HIDS), based on individual hosts (system). HIDS includes a software agent that analyzes an application, operating system logs, system files and local database for identifying intrusion attempt on the network. When we come to the enterprise level security then it consists of a huge traffic flow which requires large processing power. The fundamental working of IDS is based on two approaches: Anomaly detection and Signature detection [2] [3].

1. Anomaly Based Detection: In this approach, IDS tries to figure out anomalies over the network. When any deviation occurs in network activity the irregularity get detected by a system which conclude the type of intrusion on a network. The system uses system variables to figure out events like positive and negative false that is used for predicting attack on the network. In this type of IDS there may be possibilities of many false alerts.

2. Signature Based Detection: In this approach, IDS tries to match predefine patterns on the network traffic. If a pattern is detected in any packet payload then the system triggers an intrusion alert on that network.

A Snort is an open source network intrusion detection and prevention system (IDS/IPS) developed by Sourcefire [4].The

Licensed Under Creative Commons Attribution CC BY

Snort system is a widely deployed system for securing IT industry from potential network threats. It combines the signature, protocol and anomaly mapping methodology for packet inspection in order to determine the network vulnerabilities.

3. Related Work

3.1 Apache Hadoop

An Apache Hadoop is an open source distributed environment which facilities data intensive task over big-data[]. It maintains the properties of volume, velocity and variety and supports the distributed application on large clusters of commodity hardware's. Hadoop has two logical modules: HDFS (Hadoop Distributed File System) as a storage and Map-Reduced as a computational workflow. HDFS provides high availability and fault tolerance by compromising the data redundancy (default=3). The stored data on HDFS, support random file of 64/128 megabytes of blocks in size and the block gets duplicated to default replication factor of 3. Hadoop supports various eco-systems like Ambari, HBase, Zookeepers which provides added functionality.

3.2 Outlines of NVidia GPGPU

The GPGPU abbreviation was first used by Mark Harrison [7]. The GPU is commonly used as a co-processor that work parallel with the CPU, which is occasionally called as Visual Processing Unit (VPU). The GP-GPU (General Proposed - Graphical Processing Unit) was an idea to use GPU hardware as Co-Processor that offloads CPU processing. A GPU is a massively parallel SIMD processor that supports computing power up to TFLOP unit's now-days. GPU architecture is based on a high number of parallel control units (up to 2688 control cores in the latest graphics adapter Tesla K20X by NVidia[8]). The NVidia CUDA (Compute Unified Device Architecture) is a parallel computing architecture that enables programmers to use CPU and GPU to work parallel in order to solve complex computational problems. The figure illustrates a simple GPU architecture. A GPU has been exhibited as a set of Streaming Multi-Processors (SM's), with each include set of Scalar Processors (SP's). An SM can exhibit the work of single Instruction, Multiple Threads (SIMT) in which each SP's of a Multi-Processor execute simultaneously same instruction on multiple data elements.

- **Survey Paper A[9]:**

The objective of this paper [9] is to provide distributed approach in processing snort alerts using the Hadoop framework. This system handles large amounts of network traffic from multiple snort processes which runs on different servers and analysis the logs using Hadoop Map-Reduced computation. The experimental setups exhibit system performance improvement by running on 8 slaves nodes, which is about 4.2 times faster speeds then single computer system. This system is scalable in nature but certainly it lacks in real time property to handle big data. In order to add real-time processing module, the system required to investigate with Hadoop data platform and its ecosystem like Ambari, Storm and Cloudera Impala.

- **Survey Paper B[10]:**

The related paper [10] B is based on the scalable NIDS logs analysis using cloud infrastructure. The large volume of network logs is processed by NIDS by using Hadoop and Cloud Infrastructure. System performance is carried by analyzing Snort logs report of size 4 GBytes. In the results, as the number of nodes increases, the system performance increase as compared to the standalone system. The system performance with 5 nodes is about 2.5 times better than that of standalone system. This system required improvement related to the data analytic with open Hadoop ecosystem like Ambari, HBase with Zookeeper for dynamic scheduling.

- **Survey Paper C[11]:**

The Related work [11], is based on Hybrid Intrusion Detection System based on Snort and Hadoop. In this system intrusion analysis is carried out using Hadoop Hive ecosystem. Hive is used as a data warehousing system, which analysis large volume of network traffic. This system offers optimized log analysis in order to trigger quick intrusion alerts on big network. The system performance could increase by adopting cluster environment which supports streaming access to network traffic.

4. Implementation Details

The proposed work is based on integration of Hadoop and NVidia CUDA technology that work co-operatively to process large network traffic. We have separated out system functionality into two different modules in which, Hadoop would take care of network traffic and GP-GPU is dedicated for intrusion detection and identification. In order to process network traffic we have configured framework with Hadoop Data platform with its eco-systems like Hadoop, HBase, Flume. The Flume support real-time streaming of data on a network which help to pull, collect and apply context-routing on data flow. After traffic ingestion is done over HDFS, the system has to start pre-compilation of server logs and data in it. The server logs include user details and its network access information and packet payload contains actual data along with protocol details. In our proposed framework network traffic compilation is done and dataset has to move forward on GP-GPU on order to detect intrusion in it. The intrusion detection is carried out by Parallel Failure-less Aho-Corasik (PFAC) algorithm approach of pattern matching. The PFAC algorithm which detects a pattern based on a state transition machine. The Hadoop gets intrusion results back from GP-GPU and perform further analytic on those results. The analytic over intrusion are useful for taking further preventing action against major security threats.

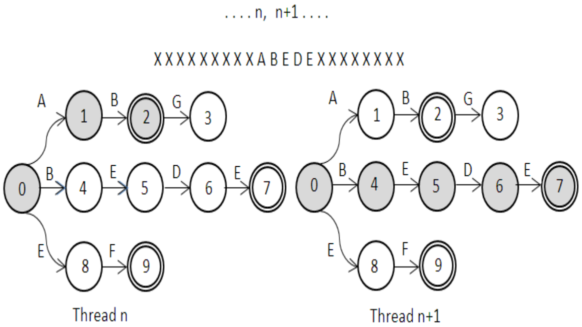
4.1 System Functionality

- **Packet Analysis:** The preliminary packet classification takes place to capture all packets from all available NIC (Network Interface Cards). In this module, we use Python-Scapy special packet crafting library for packet classification. In this process system extracts information like protocol numbers, payloads, and source - destination ip and mac address. Depending upon packet information NIDS tries to figure out malicious activities on the network. The preliminary results of packet processing and classification are depicted in

Licensed Under Creative Commons Attribution CC BY

the (Packets Capture Result).

- **Data Compilation:** The respective system logs would be generated from networks onsite servers and stores into local database. This data will be then pushed to HDFS by using Hadoop agent and its eco-system. Once data are pulled into HDFS then it is processed by Hadoop Data platform and stored in a structured format using HBase. The proposed system compiles logs and associated information from gateway servers and sends this dataset to the next step for GP-GPU to detect intrusion into the network logs.



- **Intrusion Detections:** The compiled logs are moving forward at GP-GPU for intrusion detection by performing pattern matching into it. We perform signature mapping using Aho-Corasick algorithm on GP-GPU in order to provide Multi-Pattern matching using SIMD architecture. The system first performs intrusion detection functionality based on state transition and parallel calculates intrusion on that dataset. The result includes information regarding number of detected intrusion in particular network logs.

- **Intrusion Analysis:** The result from CUDA is dumped back into file-system, which contains the threads count and vulnerability found in respective data sets. The information is utilized by the Hadoop ecosystem to perform analytics over the dataset. This helps the security administrator to configure out network security policies and network settings. In this module Hadoop performs analytic based on PF-ICP scoring values which will see in the next section of the paper.

5. System Modeling and Design

In our proposed work, the intrusion detection module is implemented by using PFAC algorithm approach, in which failure transitions are removed from the state transition machine.

- **Parallel Failure-less Aho-Corasick Approach:**

The intrusions analytics will help network administrators to configure network security policy as per changing methodology of attacks on the network. If we want to know the statically view of network attack, then the analysis is best option to perform checks on network attack with its types and priority. We apply Pattern Frequency Inverse Cluster Frequency (PF-ICF) approach which is based on Term Frequency - Inverse Document Frequency [12]. In this methodology system calculates the pattern score for detecting intrusion on the network. The pattern gets ranked with respect to the cluster frequency in which it gets detected. It considers as pattern, frequency such that how many times the pattern is detected in a particular cluster.

Fig. 1.Working of PFAC Algorithm

Cluster frequency describes the cluster, which gets detected by the intrusion pattern “p” with respect to the crop data. The corpus is a collection of clusters data from entire infrastructure. With this approach, we consider both higher and lower score values in order to determine an intrusion which is found rare and found majorly in a particular cluster. Finally taking the product of PF and ICF frequency we get scoring weight.

- **Traffic Analysis:** We have implemented a NIDS framework with functionality of Traffic Ingestion, Packet Analysis and Hadoop Computation on the sample log of network traffic (packet data).In this implementation of NIDS, we analyze the design consideration by integrating Hadoop and GPGPU environment. We have constructed a cluster of three computer systems of Pentium core i7 processor with 4 physical cores, 16 GB memory for observing preliminary outputs of the system. A Hadoop master node that has a Namenode and a Tasktracker configured with 16 Giga Bytes of RAM. Clusters are installed with Apache Hadoop 2.1.0 on Ubuntu 12.04 LTS operation system. Software’s are Java 1.7 JDK, Eclipse Luna, Hadoop 1.0 plugins, HBase to be installed on a master node.

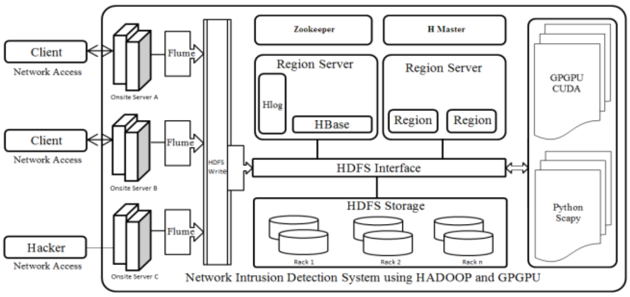


Figure 2: Proposed NIDS Architecture

- The network traffic sniffing is carried out by using Python-Scapy packet crafting library [13]. The packet capturing script is a code in python language which provides fast packet capturing and result manipulation. The initial results of packet capturing on the network are shown as following.

6. System Model

The Network Intrusion Detection System Architecture Based on Hadoop and GPGPU is defined by $\omega = (S, \mathrm{I}, \mathrm{O}, \delta, F)$. Let us assume the NIDS is in its initial secure state S.

1. S : System States including system processes or program.
2. I : The set of input to the system that includes data or file.
3. O : The set of possible output to the system.
4. A : System access attribute that includes read, write, append, executes and control (r, w, e, and c).
5. δ : System Transition Function.
6. F : Final state of the system.
7. T : Time variable refer by system components $T \in t > 0$.

A.System States: Let set S be defined as the initial state of the system. Assuming initial state of the NIDS is secure with

Licensed Under Creative Commons Attribution CC BY

reference to system time $T \in t > 0$.

$$S = \{S \mid S \in (0 \mid 1) \, System\ States\}$$

$$S = \begin{cases} 0 \ for,\ t = 0 \\ (0 \mid 1) \ for\ ,t > 0 \end{cases}$$

B. System Input: Let set I be defined as the input to the system. The inputs $ix,\ iy \in \mathrm{I}$ are two input sources to the system.

$$\mathrm{I} = \{ix, iy\ are\ InputEvents\ of\ system\}$$

$$\mathrm{I} = \begin{cases} ix = packet\ \ data\ \ from\ network \\ iy = accessLogs\ \ from\ network \end{cases}$$

C. System Output: Let set O be defined as the output from the NIDS system. The inputs $ox,\ oy \in O$ are two output sources to the system.

$$O = \{ox,\ oy\ \cup\ S^{+} \cup F\ OutputEvents\ \ with\ \ FinalState\}$$

$$O = \begin{cases} ox = System\ Output \\ oy = Final\ \ State\ \ Output \end{cases}$$

D. Transition Function: Let set δ be defined as a Transition Function of the system. The system transaction depends upon system initial state S and with system functionality represented by δ.

$$\delta = \sum (S^{+} * A * F) \, w.r.t\ \ Timet > 0\ \ of\ \ the\, system.$$

$$\delta = \{I_t * f_{id} * A_t\ \ w.r.t\ \ Time\ \ t > 0$$

$$f_{(1)} = Packet\ \ Analyzer(packet)$$

$$f_{(2)} = TrafficIngesion(dataset)$$

$$f_{(3)} = IntrusionDetection(packet, dataset)$$

$$f_{(4)} = IntrusionAnalysis(dataset)$$

E. Final State: Let set F be defined as the final state of the system.

$$F = \{z \in F \mid F\ is\ \ FinalState\ \ of\ \ system\}$$

$$F = \begin{cases} z = 0\ \ System\ \ exit\ \ Successfully \\ z = 1\ System\ \ exit\ \ Unsuccessfully \end{cases}$$

A. System Model Diagram: The proposed NIDS model is represented and shows input to the system from various networks. The system includes two Hadoop and GPU modules to perform all system functions and generates output from the system. The operational flow of the system is represents in Fig.4.

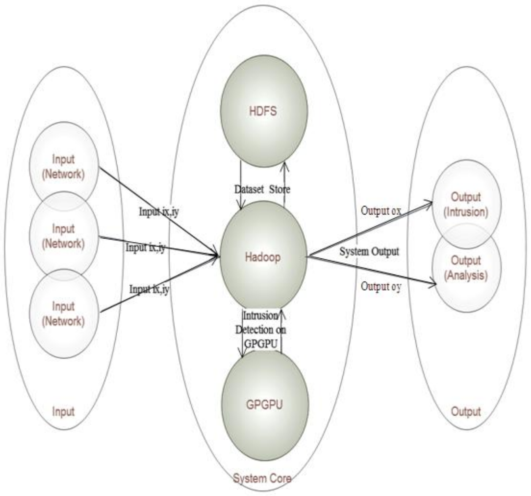


Figure 4: NIDS Proposed Model

7. Analysis

The proof of concept (POC) for the proposed systems is described below. We have implemented small functional module for Traffic Ingestion, Packet Analysis, Hadoop computation on log dataset, and data analytics over dataset. Depending on sample module and mathematical analysis of the following function, it is easy to understand the concept of integrating GPGPU into Hadoop framework. We have constructed a cluster of three computer systems of Pentium core i5 processor for observing preliminary outputs of system. A master node that has a Namenode and Tasktracker configure with 4 GBytes of RAM. Clusters are installed with Apache Hadoop 2.1.0 on Ubuntu 12.04 LTS operation system. Software’s are Java 1.7 JDK, Eclipse Luna, Hadoop 1.0 plugins, HBase and Flume to be installed on a master node. Nodes are connected by 100 MBPS Ethernet cable.

7.1 Packet Analysis:

The packet analysis is carried out by using Python-Scapy packet crafting library [13]. Packet capturing script is written in python which provides fast packet designing, interactive packet and result manipulation.

A. Data Analytic using PF-ICF:

Using Hadoop, we get flexibility to perform wide range of analytics on Big-data. Here in our proposed system (NIDS), we applied PF-ICF to perform analysis on intrusions pattern over entire infrastructure. The data analysis requires some unique scoring values. In this approach we use scoring weight for intrusion pattern over the network. Depending upon pattern frequency over network cluster frequency, we can determine the wide spread of intrusions over the network.

B. Intrusions Types on the network:

a. Node 1 : DOS, DOS, ArpPoi, Tcp_Syn, Tcp_Syn, Tcp_Syn.
b. Node 2 : DOS, DOS, DOS, Tcp_Syn, Tcp_Syn, Tcp_Syn.
c. Node 3 : DOS, DOS, ArpPoi, Tcp_Syn, Tcp_Syn, Tcp_Syn.

TABLE I. PF-ICF Score Weight

Id	**Intrusion Detected on Network**		
	Denial of Services	**Tcp_Syn**	**ArpPoi**
Node 1(PF-ICF):	0.33	0.50	0.1872

Licensed Under Creative Commons Attribution CC BY

Id	Intrusion Detected on Network		
	Denial of Services	Tcp_Syn	ArpPoi
Node 2(PF-ICF):	0.50	0.50	0
Node 3(PF-ICF):	0.33	0.66	0.1872

7.2 INTRUSION Statistics using PF-ICF:

The graph represents the maximum and minimum numbers of the vulnerabilities that are found across infrastructure. The above shown statistics of intrusions on network help a network administrator to configure the network security.

7.3 Observations:

a. If an intrusion appears on few clusters: Low Score.
b. If an intrusion appears on many clusters: High Score.

```
1. Ethernet Header :--> Destination MAC : a4:1f:72:8e:05:56, Source MAC : a4:1f:72:8e:00:92, Protocol : 8
2. IP Header =, IP Version : 4, Header Length : 5, TTL : 64, Protocol : 6, Source Address : 172.25.24.123, Destination Address : 172.25.24.124
3. TCP Header =, Source Port : 1500, Dest Port : 58171, Sequence Number : 279799081, Acknowledgement : 3489936958, TCP header length : 8
4. Data =, t 316:07:20 User: Sent Data with sig:E#^0z1#^! 11515115continue transition on DFA
```

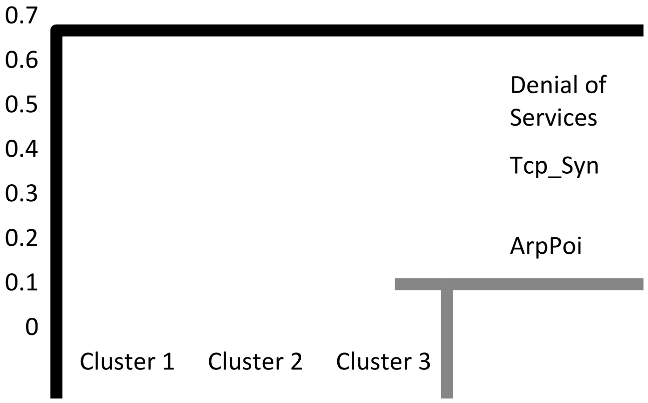


Figure 5: Intrusion Analytic using IP-ICF

8. Conclusion

In this paper, we have analyzed the design consideration of NIDS using Hadoop and GPGPU. The objective is to optimize NIDS performance by offloading intrusion mapping functionality from Hadoop to GP-GPU. In our proposed work, we have configured NIDS with Hadoop data platform in order to process large volumes of network traffic. We found that our design consideration is capable of processing log files of 1, 2, and 4 gigabytes in a very efficient time that of 29.86, 47.09, and 94.96 Seconds. Integration of Hadoop Data platform and NVidia GPGPU technology for offloading of pattern mapping job from Hadoop framework would result in optimizing NIDS performance.

In future work, the proposed NIDS system would capable to perform on high end computation based on Kepler GPGPU architecture in order to accelerate overall performance. The system is also very open to adapt high data analytic computation using Hadoop Eco-systems. The network security always remains a very challenging aspect so we must always keep on updating network security by applying new technology and approach.

References

[1] Axelsson, Stefan, “Intrusion Detection Systems: A Taxonomy and Survey”, Technical Report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Sweden, March 2000.

[2] H.Debar, M. Dacier, and A. Wespi, “Towards a taxonomy of Intrusion Detection Systems”, The International Journal of Computer and Telecommunications Networking - Special issue on computer network security, Volume 31 Issue 9, Pages 805 – 822, April 23, 1999.

[3] Holtz, Marcelo D, Bernardo David, Sousa Jr., R. T., “Building Scalable Distributed Intrusion Detection Systems Based on the MapReduce Framework”, Telecomunicacoes (Santa Rita do Sapucai), v. 13, p. 22-31, 2011.

[4] Snort Official Page, Online 2014, Available: http://www.snort.org.

[5] Suricata Open Source IDS-IPS engine, Online 2014, Available: http://suricata-ids.org /2013/07/26/suricata-1-4-5-released/

[6] Apache Hadoop 2.4.1, Online 2014, Available:http://hadoop.apache.org/

[7] About GPGPU, Online 2014, Available: http://gpgpu.org/about.

[8] TESLA GPU ACCELERATORS FORSERVERS, Online 2014, http ://www.nVidia.com /object/tesla-servers.html.

[9] JeongJin Cheon, ”Distributed Processing of Snort Alert Log using Hadoop”, Department of Computer Engineering, Kumoh National Institute of Technology, Gumi, Gyeongbuk, Korea.

[10] Manish Kumar, ”Scalable Intrusion Detection Systems Log Analysis using Cloud Computing Infrastructure”, M. S. Ramaiah Institute of Technology, Bangalore and Research Scholar, Department of Computer Science and Applications, Bangalore University, Bangalore, INDIA.

[11] Prathibha,”Design of a Hybrid Intrusion Detection System using Snort and Hadoop”, International Journal of Computer Applications (0975 – 8887), Volume 73– No.10, July 2013.

[12] Puneet Goswami, PhD,”The DF-ICF Algorithm- Modified TF-IDF”, International Journal of Computer Applications (0975 – 8887), Volume 93 – No.13, May 2014, Associate Professor Galaxy Global Group of Institutions Dinarpur Ambala, Haryana, India.

[13] About Scapy, Online 2014, http://www.secdev.org/ projects/scapy/.

[14] Herodotos Herodotou,”Hadoop Performance Models”, Technical Report, CS-2011-05, Computer Science Department, Duke University.

[15] Chen-Hsiung Liu, ”PFAC Library GPU-based string matching algorithm”, Online 2014, http://code.google.com/p/pfac/

Licensed Under Creative Commons Attribution CC BY