Solving a Big-Data Problem with GPU: The Network Traffic Analysis

Mercedes Barrionuevo, Mariela Lopresti, Natalia Miranda, Fabiana Piccoli
UNSL - Universidad Nacional de San Luis, Departamento de Informática, San Luis 5700

***Abstract*—The number of devices connected to the Internet has increased significantly and will grow exponentially in the near future, it is due to the lower costs.**

It is expected that next years, data traffic via Internet increases up to values around zettabyte. As a consequence of this increase, it can be observed that the data traffic is growing faster than the capacity of their processing.

In recent years, the identification of Internet traffic generated by different applications has become one of the major challenges for telecommunications networks. This characterization is based on understanding the composition and dynamics of Internet traffic to improve network performance.

To analyse a huge amount of data generated by networks traffic in real time requires more power and capacity computing. A good option is to apply High Performance Computing techniques in this problem, especifically use Graphics Processing Unit (GPU). Its main characteristics are high computational power, constant development and low cost, besides provides a kit of programming called CUDA. It offers a GPU-CPU interface, thread synchronization, data types, among others.

In this paper we present the causes of increasing data volumes circulating on the network, data analysis and monitoring current techniques, and the feasibility of combining data mining techniques with GPU to solve this problem and speed up turnaround times.

I. Introduction

The popularity of the Internet has brought an intrinsic alteration of the traffic on a network, due to not only to the volume of data transferred, but also to the type of applications used and therefore, the nature of traffic generated by them. For this reason computer networks have became an indispensable component of any computer system. The daily growth of networks, greater requirements and better performance are demanded by them.

The networks growth was increased with the advent of Web 2.0, which has caused a change in the mode of communication, the Internet users have left to be passive recipients of information to become generators of it.

Today, most of the users are part of the social networks, have their own blogs and comment on them, participate in forums, say about news, and so on, causing not only the growth of large volumes of information, but also network traffic.

The growth of both aspects: information volume and communication traffic, occurred exponentially in recent years, everyday examples of these activities are the number of images uploaded daily to social networks (300 million in Facebook, 45 million in Instagram), viewed videos on YouTube per day (2 billion), the amount of text messages sent by a teenager in a month (4762), the monthly number of searches on Twitter, the world's Internet traffic, among others.

This can be applied not only to the daily activities developed directly on the Internet, but also to those related to the natural phenomena as the weather and seismic, environments related to health, safety and of course, the business environment. All these applications have in common the generation of a large amount of data and the using their own or public networks for their communications [1] [2] [3].

One consequence of the exponential growth data-intensive applications is the need of having high performance networks. Analyze traffic in such networks, and networking in general, in-

volves capturing network traffic and inspecting it in detail to determine what is happening on the network and hence make good decisions.

The aim to manage the performance of a network is to collect and analyze traffic flowing through the network to determine its behavior in various ways, either at a particular time (real time) or a time interval. This will allow us to make appropriate decisions according to the behavior found.

To achieve an efficient administration two stages are required, in the first place is the monitoring and data collection, and in the second is the analysis of information [4] [5] [6]. As regard to monitoring stage, the aims is to observe and collect data concerning with the behavior of network traffic. While the analysis stage should allow us to obtain information from the data collected in the previous step, to determine the network behavior and make appropriate decisions, which it could help us to improve the performance.

It is easy to see the computational complexity for solving each of the tasks involved in each previous stages, not only respect to hardware resources, but also the time involved in obtaining each answer.

A good network performance management works with large volumes of data and make decisions as quickly as possible so that they are consistent with the network current behavior and usage. The features found in Internet traffic, such as speed, variety and volume data are part of a new concept called *Big Data*.

This process should take into account the volume of data (terabytes is passed to zetabytes), the growing velocity (data in batch/files is passed to data in ”streaming”) and the variability in the format of the data (structured data to semi-structured or unstructured data).

Working with large volumes of information derived from network traffic is not easy if it is done with traditional tools and methods [7] [8] [9]. For this reason, its is necessary to think about techniques and architectures in High Performance Computing (HPC), they can help to improve performance; so the search and selection of HPC techniques in each of the steps or processes involved, particularly in the network performance management allow to solve effectively their objectives.

In HPC field, GPUs have been widely applied to accelerate scientific purpose computing and engineering in general. This is due to the characteristics of its architecture, a GPU provides more computing power than a CPU with multiple cores [10] [11]. Our proposal aims to use the GPUs and all its programming and execution environment for accelerating the steps involved in the procces of network performance management.

In this work, we analyse all activities involved when we want to improve the network performance through of application of GPGPU techniques in its two main tasks: monitoring and analysis traffic.

This paper is organized as follows: the next section (II) defines and introduces Big Data and Data Mining concepts. The stages of monitoring and analysis network traffic is discussed in Section III; Section IV focuses on programming using general-purpose GPU (GPGPU). Finally, Section V shows the feasibility to use GPU programming environment for monitoring and analysis network traffic.

II. Data Mining and Big Data

Collecting and storing large volumes of data not is enough for many organizations, also they require extract useful information from them and act in consequence, by making good decisions. Finding useful information or pattern from the large volumes of data, where there is apparently nothing useful, is known as Data Mining [12].

Not all information is right, it must satisfy the next characteristics:

- Valid and without errors.
- Innovative, it provides something new.
- Potentially useful, it should conduce to the correct decision making.
- Understandable for the user.

Data mining requires of other disciplines such as statistics, computing, information retrieval systems, among others. All of them are necessary to facilitate the processing of the calculations. Even more when the large amounts of data can be stored in distributed and unstructured way. By this razon, the traditional methods of information processing are not useful, it is necessary to formulate new ones.

This reality has led to a new concept: *Big Data*. Big Data refers to data sets whose size is beyond that capacity of typical database to store, manage and analyze data [9]. This new concept means large amount of data (sizes of terabytes are changed by zetabytes or more), with great variety of formats (the structured data are replaced by semi-structured or unstructured data) and generated a high speed data (the data in batch or files are substituted by "streaming"). To process this new information type is a big challenge [13] [14]. Another important characteristic of Big Data is the information value, its extraction will mark the next decade. We can find the value in different forms, for examples in the improvements in business performance, a new customer segmentation, or automation of tactical decisions, among others. When we speak about Big Data, we make reference to profitable and innovative techniques used to solve business problems whose requirements computational resources exceed the capabilities of traditional computing environments [13].

The need of Big Data plataforms obeys to many causes, some of them are the increment of: volume of captured and stored data, amount of data placed on the network and variation in data type to be analized, furthermore of the rapid data growth and increasing demand for real time analytical results. In a study of IDC [14] shows that the size of digital information reached, in 2012, is 2.8 zettabytes and predicts that by 2020 will reach 40 zettabytes. These data will be generated by individuals and devices. Only 0.5% of information is analized.

In the next section we discuss the monitoring and analysis network traffic stages. For monitoring stage we talk about two techniques and how classify the network traffic. In the other stage we show the categories of analysis and the different traffic analysis techniques.

III. Monitoring and Analysis Network Traffic

One consequence of the exponential growth of data intensive applications is the increase of the traffic in the network and the need for a good administration.

For efficient network management is necessary to analyse the network behaviour. In this direction, first we have to monitor and collect data, and then (the next step) to analyse these collected data to obtain relevant information and make a decision, whose objective is improve network performance. All this process involves several tasks, where each of them has to be performed quickly to achieve a response time of whole process close to real time and, consequently, to provide a good service for network users.

It is important to distinguish between network monitoring (also called network measures) and the analysis of the information or the identification of an application [15]. The first refers to the collection and data counting, while the second deals with the recognition and classification of some traffic characteristics (which vary according to the technique used).

Every high-speed network generates a massive amount of data. In consequence, the monitoring task of Internet traffic becomes computationally expensive, therefore, applying of some sampling technique can be a good solution for a scalable Internet monitoring. These techniques allow lowering the processing cost and hardware requirements like the storage. A drawback of sampling techniques reduce the volume of collected data but can produce information losses.

In general, the sampling of traffic takes some packets circulating in the network, this process makes partial observations of packets and draws a conclusion about the behaviour of the system. Transforming the obtained partial data

into information or knowledge about the overall system behavior is known as the "inversion problem". By all of the expressed above, the straight consequence of this approach (to reduce considerably in the number of data needed to compute the descriptive measures of network behavior) is stronger than its disadvantage (loss of information).

In the next sections, we detail the before mentioned stages: Data Monitoring and Information Analysis, and their main characteristics.

A. Data Monitoring

In most of cases, the goal of network monitoring is to identify performance problems or analyze different aspect of its design and use.

Network monitoring can be defined as a system dedicated to supervise constantly the network to detect anomalies, such as slow or failed components, and notify to its administrator if a problem is detected.

The main input source of an analysis of network traffic is "dataset", a set with all captured network packets (for example it can be all protocol header information). From this dataset, we can extract data (data mining) for each particular analysis.

In the last years, two different kind of effective techniques for network monitoring have been used, they are [16]:

- *Agent-based Monitoring*: In each controlled device of network, there is an "agent", a piece of software. Each agent collects device information (such as connectivity state of its network interface, link performance, throughput, among other interest information), and sends it (by the network) to management platform. SNMP (Simple Network Management Protocol [17]) is a clear example of this kind of protocol for management and monitoring of a network.
- *Agentless Monitoring*: It depends directly of analysis of network traffic. This system supervises the network traffic according to, for example, the connection performance, information routing of TCP packets [14], window state to estimate congestion, host services used (web, ftp, ssh, etc.), among others.

In [15], the authors propose a classification of traffic measurement. It can be divided in active and passive measurements, each one has the following characteristics:

- *Active Measurement*: The data are obtained from injected traffic (as test packets) in the network. The work methodology is: many test packets are sent and measured their response times. This value is mainly used for fault detection, network vulnerabilities or performance proof of an application. However, this measure way is not suitable when the objective is to consider the user influence in the network, because the method sends packets independently user behavior, changing in consequence the network metrics.
- *Passive measurement*: The data are acquired from observation of network traffic, its packets and flows. It shows the dynamic and distribution of traffic. Its main problem is data amount, it scales according to connection capability. The captured data volume can be massive when there are a lot of connection or a very large amount of user. It can be done in two levels, they are:
  - *Packet level*: Each packet that passes across measurement point is evaluated. Some examples of collected information are IP direction of source and destination, port number of source and destination, packet size, protocol numbers and specific data application. There are several open-source tools to capture packets, called sniffers, some of them are Tcpdump [18], Ethereal [19], Wireshark [20].
    This level has many difficulties, for example as the collected data amount to archive is huge, the efficient access to these databases is a critical point. Other problem is to count with sub-

optimal hardware to capture packets (eg, Low-end NICs, CPU low power-computing, etc.) and, in consequence, loss some of them.

– *Flow level*: In this case, the flow is considered. A correspondence of packet in flow has to be established. A flow is a set of packets with the same IP direction, the same TCP port of source and destination, and the same application type.
The collected data include flow number by time unit, transfer rate, size and duration of flow, etc. Some of tools for flow-level passive measurement are NetFlow [21] and JFlow [22].

For all above exposed, before to perform the data monitoring in a network it is necessary to establish its characteristics: agent-based or agentless, active or passive measurement, and if it is passive, which level: packet or flow. The combination of them defines the monitoring policy to be implemented.

B. Information Analysis

Network traffic analysis could be defined as: *the inference of information from observation of the network traffic data flow* [16].

The network traffic analysis can be classified in one of the following three categories: *real-time*, *batched* and *forensic analysis*. The first two categories are not event-driven, because their analysis is continuous. On the other hand, the forensic analysis is made when a particular event occur.

Real-time analysis is performed over the data as it is obtained, or using small batches often called buffers to efficiently analyze data. Its has the problem that need high computational resources. Otherwise, the *batched analysis* is performed periodically, storing data in large data warehouses. *Forensics analysis* in the other hand, is performed when a particular event occurs (triggered analysis). A typical forensics analysis occurs, for example, when an intrusion is detected to a particular host.

At this stage, we can identify two main issues to resolve, they are: where the information is extracted and how to model it. Each of issues has its own techniques, they are:

- *Techniques of Data Inspection (Where)*
The information to analyse comes of the packets circulating in the network, generally it is extracted from header of each packet. Once acquired, the information is processed to obtain the results. There are different techniques of traffic analysis, some of which are:
  – **Packet Decoding (Packet Analyzing)**: In this case, all header fields of packet are decoded and presented in a human-readable way. Network traffic analyzers like *Tcpdump* or *Wireshark* are some examples of packet decoding applications. This technique is generally used for security purposes (intrusion detection, bandwidth abuse), network management and fault detection.
  – **Extracting of Specific Data of Packet**: This kind of techniques considers to process some parts of the extracted data packets. Generally it is performed when is necessary to study particular aspects of the traffic.
- *Techniques of Information Processing (How)*
The collected data set has to be processed to obtain useful information and act in consequence. There are different processing tasks that can be performed on this data set, some of them are:
  – *Graphical Representation of Raw Data* [23][24]: The useful information is represented generally through 2D and 3D graphics based on time, histograms, pie charts or diagrams dispersion. This representation is of interest in many areas, mainly in monitoring, management and security of network.
  – Extraction of Statistical and Pattern: Based on the collected information,

different statistical functions are calculated, among them we can find average, time distribution, and probability distribution functions. All these statistic calculated information allows us establish patterns of network behaviour, an example is NetAnalyzer traffic analysis platform [25].
- Analysis based in rules [26][27][28]: To this kind belong all analyzes of traffic inspection that look for coincidences with a particular rule or signature. The signatures are defined as specific values of determined header fields or a combination of several values of them. These techniques are used mainly with security purposes and particularly in intrusion detection systems (IDS) such as Snort [29].
- Flow-based analysis [15]: These techniques treat to network traffic as a flow because most of the information exchanged in a computer network is oriented to connection and non-oriented to packets. This kind of analysis takes advantage of this feature. A typical example of network flow is a TCP connection, where the data exchanged are governed by the TCP state machine [14].

In the case of information processing for analyzing network traffic, there are several data modeling techniques used by researchers when conducting network traffic analysis. To examine the users behavior from collected data, some of next methods can be used:

- Linear methods : This category includes the logistic models, regression models, and the cluster-based analysis, among others. These methods use complex statistical modeling techniques to examine the user behaviours considering data traffic [27].
- Nonlinear methods are fundamentally based in Artificial Intelligence algorithms like Neural networks, Fuzzy-logic algorithms and K-nearest neighbour algorithms. All of them have also been found effective for helping in network intrusion detection decisions [27].
- Bayesian network [30][31]: They assume that parameters to be studied are random parameters rather than fixed. Before looking at the current data, old information can be used to construct a prior distribution model for these parameters and, therefore, classify new data based on the probability of some values are unknown parameters. Then, the current data are used to revise this starting assessment .
- Data Mining Techniques [32]: These are based on the combination of machine learning, statistical analysis modeling and database technologies to find patterns and relationships among data fields to predict future results.

For everything expressed, the improvement of network performance implies a lot of tasks which demand high computational cost, so it is necessary to apply HPC techniques to increase the speed of all involved tasks. In particular, this work focuses in HPC techniques using GPU or a cluster of them. The next section delineates the main characteristics of GPU and its programming model.

IV. GPGPU

The GPU is a dedicated graphic card for personal computers, workstations or video game consoles. It is an interesting architecture for high performance computing. The GPU was developed with a highly parallel structure, high memory bandwidth and more chip surface dedicated to data processing than to data caching and flow control. It offers, in principle, a speedup to any standard graphics application [10]. Mapping general-purpose computation onto GPU implies the use of the graphics hardware to solve any applications, not

necessarily of graphic nature. This is called GPGPU (General-Purpose GPU), GPU computational power is used to solve general-purpose problems [10][33]. The parallel programming over GPUs has many differences from parallel programming in typical parallel computer, the most relevant are: *The number of processing units*, the *CPU-GPU memory structure* and the *number of parallel threads*.

Every GPGPU program have some basic steps. Firstly the input data should be transferred to the graphics card from the CPU (host). Once the data is in place, a massive amount of threads can be started (with little overhead). Each thread works over its data and, at the end of the computation, the results should be copied back to the host main memory. Not every class of problem can be solved with the GPU architecture, the most suitable problems are those implementable with stream processing and using limited memory size, i.e. applications with abundant data parallelism without cross-talking among processes. The programming model is Single Instruction Multiple Data (SIMD).

The CUDA, supported since the NVIDIA Geforce 8 Series, enables to use GPU as a highly parallel computer for non-graphics applications [10][34]. CUDA provides an essential high-Level development environment with standard C/C++ language. It defines the GPU architecture as a programmable graphic unit which acts as a coprocessor for the CPU. It has multiple streaming multiprocessors (SMs), each of them contains several (eight, thirty-two or forty-eight) scalar processors (SPs).

The CUDA programming model has two main characteristics: the parallel work through concurrent threads and the memory hierarchy. The user supplies a single source program encompassing both host (CPU) and *kernel* (GPU) code. Each CUDA program consists of multiple phases that are executed on either CPU or GPU. All phases that exhibit little or no data parallelism are implemented in CPU. In opposition, if the phases present much data parallelism, they are implemented as *kernel* functions in the GPU. A *kernel* function defines the code to be executed by all the threads launched in a parallel phase. The GPU resources are much more efficiently used if the kernel do not make branching (represented as `if` instructions), in other words, if all the threads follow the same execution path.

GPU computation considers a hierarchy of abstraction layers: $grid$, $blocks$ and $threads$. The $threads$, basic execution unit that executes *kernel* funtion, in the CUDA model are grouped into $blocks$. All threads in a block are executed on one SM and can communicate among them through the *shared memory*. Threads in different blocks can communicate through *global memory*. Besides shared and global memory, the threads have their own local data space for variables. All $Thread - blocks$ form a $grid$. The number of grids, blocks per grid and threads per block are parameters fixed by the programmer, they can be adjusted to improve the performance.

With respect to the memory hierarchy, CUDA threads may access data from multiple memory spaces during their execution. Each thread has private local memory and each block has shared memory visible to all its threads. These memories have the same lifetime than the kernel. All threads have access to the global memory and two additional read-only memory spaces: the constant and texture memory spaces. The constant and texture memory spaces are optimized for different memory usages. The global, constant and texture memory spaces are persistent all the application life time. Each kind of memory has its own access cost, in order of speed it will be local, shared and global memory which is the most expensive to access. Please notice that local and shared memory have higher throughput and smaller latency than standard RAM in the CPU. Please bear that in mind, because this contributes to the very large speedup of our algorithms.

V. Big Data and GPGPU

For all before exposed, it is clear the need of applying HPC techniques, particularly GPGPU, in computing solutions for Big Data.

There are a big variety of real applications that work with massive data, and some researches where propose the applying of GPGPU techiques in their computacional solutions, for example:

- In [35], they develop a tool chain based on database management system and Parallel Data Processing in GPU using CUDA for large and intensive smart meter analytics. The global launch of smart meters opens new business paradigms for utilities with data collection/transaction at such a high volume and velocity. Your aim is to utilize the processing power of GPU to develop an adequate Big Data Analytics platform for an instant, in-depth analysis of massive volumes of smart meter data, which can be used in industry, personalized medicine, among others.
- In [36], they propose a framework for in-memory "Big Text Data" analytics that provides mechanisms for automatic data segmentation, distribution, execution, and result retrieval across multiple cards (CPU, GPU and FPGA) and machines, and a modular design for easy addition of new GPU kernels.
  They have presented a framework for text processing and analytics for GPUs called *MiAccLib* that allows users to exploit the powers of the GPU by providing the ability for efficient work distribution across multiple GPUs with regards to I/O access and load-balancing. Using *MiAccLib* framework, they show a significant improvement by using two GPU cards over single GPU cards, and single GPU cards over multi-core CPUs for text data sorting and matching.
- City-wide GPS recorded taxi trip data contains rich information for traffic and travel analysis to facilitate transportation planning and urban studies. However, traditional data management techniques are largely incapable of processing big taxi trip data at the scale of hundreds of millions. For this reason, in [37] use the GPGPU technologies to speed up processing complex spatial queries on big taxi data on inexpensive commodity GPUs. By using the land use types of tax lot polygons as a proxy for trip purposes at the pickup and drop-off locations, they formulate a taxi trip data analysis problem as a large-scale nearest neighbor spatial query problem based on point-to-polygon distance.
  In their experiments have used 170 million real taxi trip records have shown that our GPU implementations can complete such complex spatial queries in about 50-75 seconds using an inexpensive commodity GPU device. The performance is 10X-20X higher than that of the host machine with an Intel dual-core CPU when all the cores and hardware supported threads are fully used.

Regarding today's computer networks, achieve their good administration involves fast response times in each task of every stage: monitoring and analysis of information. Considering this and the wide range of existing applications generating heterogeneous traffic, the traditional solutions fail to resolve these problem, and new HPC solutions are proposed. Current approaches to solve these problems tend to distribute the tasks of analysis on a number of computers (cluster) or on powerful computers accompanied by co-processors such as GPUs. Some of them are:

- A group of researchers of Fermi National Laboratory [1] develops G-NetMon [38], a prototype able to analyse and monitor of network data. Its only purpose is to detect large movements of data. G-NetMon has a generic architecture for I/O which was developed to carry traffic from the network to the GPU. They implement a CUDA library to capture, monitor and analyze the network traffic through of dozens of CUDA kernels that were combined to perform them.

[1] http://www.fnal.gov/

By means this development, the researches show as the GPUs can accelerate all tasks of two stages: monitoring and analyzing (It can consider more than 11 million packages without lost any and using a single Nvidia M2070). They achieved to significantly increase speedup of packet processing (Compared with a single CPU, the speedup varies between 8.82 and 17.04, while if is considered a 6-core CPU, it oscillates between 1.54 and 3.20).

- GSDT (GPU Stream-Oriented Decision Tree) [39] in another example of using GPU in the field of networks. The algorithm applies stream data mining and classification of traffic based on the traditional construction of decision trees and flow-based techniques. The obtained results show an high accuracy, increase the speedup of classification process, and reduce the use of storage requirements for flow records. Two main steps integrate GSDT, the first is the training process which is executed on CPU. The second is the classification process that runs on GPU.
- In [40], the authors propose to FBTI(Flow-Based Traffic Identification) where combine GPU with techniques to identify network traffic as Deep Packet Inspection (DPI) [4] which depends mainly of Regular Expression (RE) evaluated by Finite Automata (FA). The FBTI target is to increase the performance of DPI systems for commodity platforms and its ability to identify traffic in high-speed networks. It uses a multithreaded to identify traffic and is characterized by strict scope mechanism, it inspects only the first K bytes of each data flow. Combining FA to assess RE and packet sampling to level of flow, FBTI achieves a performance of 60 Gbps on GPUs.

All of these works, general or network specific, show that apply GPGPU techniques in monitoring and analysing stages of network is a good alternative to improve the performance. Our goal is to study in depth and develop different algorithms for architectures such as multi GPUs or GPUs cluster.

VI. Conclusions

Big data processing is receiving significant amount of interest as an important technology to reveal the information behind the data, such as trends, characteristics, etc.. Therefore, new programming techniques are needed as GPGPU.

The GPU appear to be a very suited architecture to accelerate computations when working with voluminous data sets; e.g., for transformations, filtering, aggregation, partitioning and, particularly monitor and analysis of network traffic.

In this work, we analyse all involved issues in a systems to monitoring and analysis network traffic to improve its performance. Further, we consider GPGPU as a valid alternative to achieve it. Our challenge is big if we take account the variety of task involved, the diversity of traffic kinds, the wide range of GPU architectures and the need of obtaining good response times.

The next steps are directed to establish a traffic kind and develop all task involved in monitoring and analysis stages using GPGPU.

References

[1] M.Alvarez-Campana, A.Azcorra, J.Berrocal, J.Domingo, D.Larrabeiti, X.Martínez, J.Moreno, J.Pérez, and J. Solé-Pareta, "Castba: Medidas de tráfico sobre la red académica española de banda ancha," in *Departamento de Ingeniería de Sistemas Telemáticos Universidad Politécnica de Madrid*, 1998.

[2] A.Lakhina, K.Papagiannaki, M.Crovella, C.Diot, E.Kolaczyk, and N.Taft, "Structural analysis of network traffic flows," in *SIGMETRICS Perform. Eval. Rev*, vol. 32, no. 1, June 2004, pp. 61–72.

[3] M. Sullivan, "Tribeca: A stream database manager for network traffic analysis," in *Proceedings of the 22th International Conference on Very Large Data Bases*, ser. VLDB '96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, pp. 594–604.

[4] R.Antonello, S.Fernandes, C.Kamienski, D.Sadok, J.Kelner, I.Gódor, G.Szabó, and T.Westholm, "Deep packet inspection tools and techniques in commodity platforms: Challenges and trends," *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 71 863 – 1878, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii

[5] A. Dainotti, A. Pescape, and K. Claffy, "Issues and future directions in traffic classification," *Network, IEEE*, vol. 26, no. 1, pp. 35–40, January 2012.

[6] W. Didimo, G. Liotta., and S. Romeo, "Graph visualization techniques for conceptual web site traffic analysis," in *Pacific Visualization Symposium (PacificVis), 2010 IEEE*, March 2010, pp. 193–200.

[7] M. Barlow, *Real-Time Big Data Analytics: Emerging Architecture*. O'Reilly, February 2013.

[8] V. Mayer-Schönberger and K. Cukier, *Big Data: A Revolution That Will Transform How We Live, Work, and Think*, H. Mifflin, Ed. Houghton Mifflin Harcourt, march 2013.

[9] J.Manyika, M.Chui, B.Brown, J.Bughin, R.Dobbs, C.Roxburgh, and A.Byers, "Big data: The next frontier for innovation, competition, and productivity," McKinsey Global Institute, Tech. Rep., 2011.

[10] D. Kirk and W. Hwu, *Programming Massively Parallel Processors: A Hands-on Approach*, ser. Applications of GPU Computing Series. Elsevier Science, 2010.

[11] N. Wilt, *The CUDA Handbook: A Comprehensive Guide to GPU Programming*. Pearson Education, 2013. [Online]. Available: http://books.google.com.ar

[12] H.Suárez, "Minería de datos, big data y seguridad." in *Instituto Nacional de TecnologÃas de la Comunicaciones (INTECO)*, 2013.

[13] D.Loshin, *Big Data Analytics From Strategic Planning to Enterprise Integration with Tools, Techniques, NoSQL, and Graph*, M. Kaufmann, Ed. Elsevier Science & Technology Books, 2013.

[14] I. S. Institute, *Transmission Control Protocol: DARPA Internet Program Protocol Specification*. Defense Advanced Research Projects Agency, Information Processing Techniques Office, 1981.

[15] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, "A survey on internet traffic identification," *Communications Surveys Tutorials, IEEE*, vol. 11, no. 3, pp. 37–52, August 2009.

[16] M.Clos, "A framework for network traffic analysis using gpus," Master's thesis, Universitat Politécnica de Catalunya (UPC) Escola Técnica Superior dEnginyeria de Telecomunicació de Barcelona (ETSETB), 2010.

[17] "Simple network management protocol," 2013.

[18] "Tcpdump and libpcap." [Online]. Available: www.tcpdump.org

[19] "Cloud services." [Online]. Available: http://www.aos5.com/cloud

[20] [Online]. Available: https://www.wireshark.org/

[21] Cisco. [Online]. Available: http://www.cisco.com/c/en/us/products/index.html

[22] [Online]. Available: http://www.jumper.net/techpubs/software/erx/junose80/swconfig-ip-services/html/ip-jflow-stats-config2.html

[23] L. Nagios Enterprises, "The industry standard in it infrastructure monitoring," `http://www.nagios.org/`.

[24] Z. Inc., "Unified it monitoring and analytics for the modern datacenter," `http://www.zenoss.com/`.

[25] A. Wang, C. Talcott, L. Jia, B. Loo, and A. Scedrov, "Analyzing BGP instances in maude," in *Formal Techniques for Distributed Systems - Joint 13th IFIP-WG, 6.1 International Conference Proceedings*, 2011, pp. 334–348.

[26] W. Chen, *Statistical Methods in computer security*. CRC Press, 2004.

[27] Y. Wang, *Statistical techniques for Network security*. IGI Global, 2008.

[28] R. Bejtlich, *The TAO of network security: Beyond Intrusion Detection*. Addison-Wesley Professional, 2004.

[29] Cisco, "World leading open-source ids/ips snort," `https://www.snort.org/`.

[30] Y. Wang, I. Kim, G. Mbateng, and S. Ho, "A latent class modeling approach to detect network intrusion," *Computer Communications*, vol. 30, no. 1, pp. 93–100, 2006.

[31] D. Barbard, N. Wu, and Jajodia, "Detecting novel network intrusions using bayes estimators." 2001.

[32] S. S. Lee W. and K. Mok, "A data mining framework for building intrusion detection models." in *Proceedings of the IEEE Symposium on Security and Privacy*.

[33] J. Owens, M. Houston, D. Luebke, S. Green, J. Stone, and J. Phillips, "GPU Computing," in *IEEE*, vol. 96, no. 5, 2008, pp. 879 – 899.

[34] NVIDIA, "Nvidia cuda compute unified device architecture, programming guide version 4.2." in *NVIDIA*, 2012.

[35] "Gpu + in-memory data management for big data analytics."

[36] P. K. Chong, E. Karuppiah, and K. K. Yong, "A multi-gpu framework for in-memory text data analytics," in *Advanced Information Networking and Applications Workshops (WAINA), 2013 27th International Conference on*, March 2013, pp. 1411–1416.

[37] J. Zhang, S. You, and L. Gruenwald, "High-performance spatial query processing on big taxi trip data using gpgpus," in *Big Data (BigData Congress), 2014 IEEE International Congress on*, June 2014, pp. 72–79.

[38] W. Wu, P. DeMar, D. J. Holmgren, A. Singh, and R. Pordes, "G-netmon: A gpu-accelerated network performance monitoring system for large scale scientific collaborations," *CoRR*, vol. abs/1108.1785, 2011.

[39] P.Lopes, S.Fernandes, W.Melo, and D.Sadok, "Gpu-oriented stream data mining traffic classification," in *IEEE Symposium on Computers and Communications, ISCC 2014, Funchal, Madeira, Portugal, June 23-26, 2014*, 2014, pp. 1–7.

[40] A.Feitoza, S.Fernandes, P.Gomes-Lopes, D.Sadok, and G. Szabo, "Multi-gigabit traffic identification on gpu," in *Proceedings of the First Edition Workshop on High Performance and Programmable Networking*, ser. HPPN '13. New York, NY, USA: ACM, 2013, pp. 39–44.

Received: June 2014. Accepted: February 2015.