UDC 004.77:519.2

K.S. Deev,
Yu.V. Boyko, PhD, Assoc.Prof.
Taras Shevchenko National University of Kyiv, 60 Volodymyrska Str., 01033 Kyiv, Ukraine; e-mail: kostic2006@ukr.net

DETERMINE POINT-TO-POINT NETWORKING INTERACTIONS USING REGULAR EXPRESSIONS

К.С. Дєєв, Ю.В. Бойко. **Визначення мережевої взаємодії типу точка-точка за допомогою регулярних виразів.** З ростом популярності мережі Інтернет і загальної кількості інформаційних потоків, що супроводжується підвищенням вимог до пропускної спроможності, операторам послуг і корпоративним замовникам необхідна можливість ідентифікації потоків даних точка-точка. Оптимальним є застосування спеціальних програмно-апаратних реалізацій, які розподіляють навантаження вже в середині комплексу, використовуючи принципи і підходи, що, зокрема, описано в даній роботі. В даній роботі розглянуто принципи реалізації системи, яка проводить пошук збігу за регулярними виразами, використовуючи обчислення на графічному адаптері серверної станції. Суттєва обчислювальна потужність і можливості до паралельного виконання на сучасних графічних процесорах дозволяють проводити перевірку великої кількості даних через набори правил. Використання вказаної особливості дозволяє досягати підвищення обчислювальної потужності в 30…40 разів у порівнянні з аналогічними реалізаціями на центральному процесорі. Потенціал підвищення полоси спроможності може бути використаний в системах аналізу пакетів, захисних екранах і детекторах мережевих аномалій.
Ключові слова: мережева взаємодія типу точка-точка, регулярний вираз, графічний процесор.

K.S. Deev, Yu.V. Boyko. **Determine point-to-point networking interactions using regular expressions.** As Internet growth and becoming more popular, the number of concurrent data flows start to increasing, which makes sense in bandwidth requested. Providers and corporate customers need ability to identify point-to-point interactions. The best is to use special software and hardware implementations that distribute the load in the internals of the complex, using the principles and approaches, in particular, described in this paper. This paper represents the principles of building system, which searches for a regular expression match using computing on graphics adapter in server station. A significant computing power and capability to parallel execution on modern graphic processor allows inspection of large amounts of data through sets of rules. Using the specified characteristics can lead to increased computing power in 30…40 times compared to the same setups on the central processing unit. The potential increase in bandwidth capacity could be used in systems that provide packet analysis, firewalls and network anomaly detectors.
Keywords: point-to-point networking interaction, regular expression, graphics processor.

Introduction. Passive monitoring is used in networking in many situations for the purpose of early detection of deviations from the standard behavior of links load or statistical parameters estimation of increasing capacity for planning further development and optimization of network architecture.

Literature review. Most modern analysis systems are based on deep packet inspection (DPI) and provide checking of packet to legitimate data stream or determination of malformed packet to attack the network. Traditionally, the packet payload testing is performed by searching the appropriate sequences of bytes in packet, whose analysis is based on pre-assembled sets of signatures. One or more matches can be collected in a separate rule sets that characterize the flow and all streams that are of interesting for further analysis.

Using a sequence of binary data that is derive interaction of point-to-point peers, it is possible to identify some of these information flows and place them into a privileged class or limit their bandwidth for those. However, we need to consider false positives [1], so creating rules for identification is one of the most important task. Moreover, the presence of conflicting rules results in the inability of an unambiguous classification, increasing the number of iterations in computing time count and data structures dimension in memory of system that is used for analysis. Therefore, to describe attacks on

DOI 10.15276/opu.2.46.2015.21

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

the network usually using rules records that contain large amounts of binary checks and calculation of individual fields offset due to presence of key sequences for increasing accuracy. On the other hand, the use of regular expressions is more flexible in terms of keeping the rules up to date, as part of a software, application can often change, including networking related communication. A single representation a large number of parameters in the form of a regular expression allows make full analysis per one iteration, which positively reflected the final performance.

The presence of a large base of regular expressions has a significant effect on the characteristics of the packet analysis, including the final processing time of network packets. Optimization of the described process is a priority part that is considered in the work. Intrusion detection system Snort [2] and Bro [3] contains a large number of regular expressions to improve accuracy of network threats by signature method. Using this approach is quite costly in terms of computing power. Most of the time each byte of intercepted packet should be analyzed for the coincidence of large sets of regular expressions, it is identifying the main part of the process. Another variant of solution to this problem is to use a specialized hardware platform, which checks packets. These devices are ASIC and FPGA based, which makes inspection of many threads simultaneously. Both are very effective and cope with the task. Their main drawback is the inability to modify program that runs on them — it is impossible to be reprogrammed in real-time. Flexibility of these systems is limited, as it is usually closely associated with a particular implementation. Let's consider a hardware part of a graphics processor (GP), including its benefits and performance indicators in parallel calculations the efficiency of its application for batch processing confirmed by many documents [4…6]. Previous work related to creating methodology that can be used for testing purposes, detailed description of metric counters could be checked in [4].

Aim of the Research is the creation of automatic pattern for interested traffic types, which need separated processing due company policies or bandwidth constraints. Automated method of such calculation guarantees faster reaction to changes in peer-to-peer communication, furthermore it minimize false positive during matching.

Main Body. Modern GP specializing in cost-effective computing and parallel calculations mainly calculating graphical representation of information. Their transistors are designed mostly for data processing than for use as a cache and control flow that occurs in the CPU. Ability of modern GP to process a big number of calculations on single input is well known and widely used in nowadays computing. Using graphic card as off loader for pattern analysis starting from special hardware design of that elements. Piping data over all processors simultaneously grants performance to increase as every single pattern is checked in parallel.

This paper reviews principles, application and comparison packet analysis of systems based on interaction in graphics subsystem of the server. Architecture of proposed solution similar to the open source system Gnort [5], a separate library of which allows to transfer calculation of determining matches of the pattern based on regular expressions to GP. If we compare the throughput of the system to the Snort IDS [2], then there is a deterioration of almost in an order.

The proposed solution is a software implementation of Compute Unified Device Architecture (CUDA) framework [7] on the NVIDIA GP G9x series. In the process of studying documentation, it was reported that GP is unable to directly access the intercepted packets coming from the network card because packets copied by CPU. It used for pre-compilation previous rule sets in a format compatible for the implementation on GP. An important indicator is the speed of data transfer on the internal bus of the computer and the GP. Based on this we use blocked mode of access to memory pages that significantly benefit in performance, since it uses Direct Memory Access (DMA). The limitation of this approach is the fact that the locked memory cannot released if not used. However, this is irrelevant in this case because the system has a significant amount of RAM (64GB). Highlighted some areas in locked memory for packets and using it as a buffer, each time the packet is identified as the corresponding to match by regular expression, it is copied to that place and marked in addition that it was captured by particular rule. This double buffer scheme allows spread over time calculation processes on GP and communication between the CPU. When the first packet transmitted to the GP via direct memory access, the next intercepted packet copied to the first buffer, and so on. Special ridges should

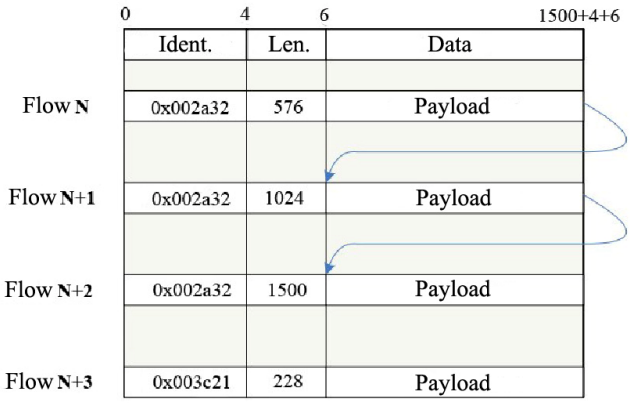

Fig. 1. Single session packet processing

be given to the processing of data streams over TCP protocol, this approach is the same as in implementing in Snort IDS [2] — aggregated formed packets that includes several relevant packets. This is achieved through the preservation identity and reflecting each active TCP session carried out by inspection under the finite state machine protocol. Thus, the maximum amount of the packet could be 65535 bytes. Nevertheless, copy of these structures takes a lot of time, resulting in degradation of total bandwidth. For minimizing this problem, the structure divided into several consecutive rows. Each line processed by different flow. In order not to lose the match of offset, signatures that have more than one line is processing lines consistently until found the first difference that is discrepancy packet criteria. The implementation of this mechanism is shown on Fig. 1.

Results. One of the factors affecting the performance of analysis complex packet is copying packets from memory to GP memory register. The capacity of this exchange depends on the size of most packets or the size of the structure in which they presented and is used memory lock or not. It is advisable to determine the limit to carry out several tests using different graphics cards. Since the graphics card is connecting via the PCIe-x16 bus keep in mind that work is possible in several modes (v1.1 and / or v2.0). Copying packets in blocking memory mode, as expected, showed higher results, as access is asynchronously via DMA. However, deviation from the theoretically calculated values of bandwidth of 4GB/s was quite substantial — using packet buffer capacity of 4MB maximum throughput limited to 2GB/s, i.e. 50 %. Some deviation from the theoretical value can be explained by encoding 8b/10b at the physical level of PCIe bus. Other limiting performance factors cannot be determined at the moment.

Complete assessment was done to determine which speed thresholds can match packets on regular expressions the while checking. The important factor is to study the performance of different types of memory — global and texture. Depending on the area in which the table is stored, performance conditions may also differ. In practical way, result shown that in the case of using GP as network packet analyzer the best is to use global memory. This test issue the average value of performance using GP as computing accelerator. One of the features of CUDA SDK is a neediness to create multiple threads for execution on multiple GP. However, implementation of OpenDPI library, through which we make analysis, involves classifying using a single stream. Making search for the regular expression often occurs by combining several expressions in one rule. Combining achieved by using the logical && operator. However, a combination of several expressions at one might significantly (exponentially) increase the number of finite automaton. To minimize this phenomenon, in practice we use a representation of complex rules in the form of several groups of less complex [8].

Further speed optimization concerning detection via search character coincidence in headers or payload of packets using wildcard Perl Compatible Regular Expressions (PCRE) [9] expressions performed in GP. Depending on the traffic volume (number of independent data streams) performance may differ significantly, so to improve the assessment test accuracy we must use identical data sets. Using different network stack configuration on server and tuned parameters of the network adapter and memory graphics accelerator we received a graph of how bandwidth depends from the size of the buffer element, it reflected in the general increase of productivity represented on Fig. 2.

This illustration requires additional explanation, with increasing number of packets in the buffer productivity of the system growth in the rapid nature, further increasing of buffer amount is no longer provide so meaningful result. Outlined feature depends on the access mode of GP components to

ISSN 2076-2429 (print)
ISSN 2223-3814 (online)

memory where packet stored. As we uses DMA page size is crucial. Setting page size depends on the average length of the packet and byte multiplicity of 2^6.

Conclusions. The paper presents a flexible approach to match network packet via search engine using regular expressions. The specified tasks on GPU results to improvement of system performance as a whole as 30…40 times. Using this mechanism, we created software and hardware that might be used as a detector of anomalies in the network. Test environment showed the maximum throughput at 12Gbit/s. Comparing the characteristics of the same set of hardware components and software, bandwidth growth was 32 times, while identified network traffic have similarities in both cases. The inclusion of this functionality to the open software package OpenDPI [10] gave system performance enhance in general up to 50…55 %. The result is not so high, but we must consider the fact that the implemented realization does not allow parallel analysis in multiple threads. Further study is provided toward the organization possibilities of using application in multiple threads and creating management system. We planned to adapt the implementation of OpenDPI library on multiple GP. Extension functional analyzer is well covered in [4…6]. Creating such system makes available analysis of network packets at speeds that were previously available only to specialized hardware [11].

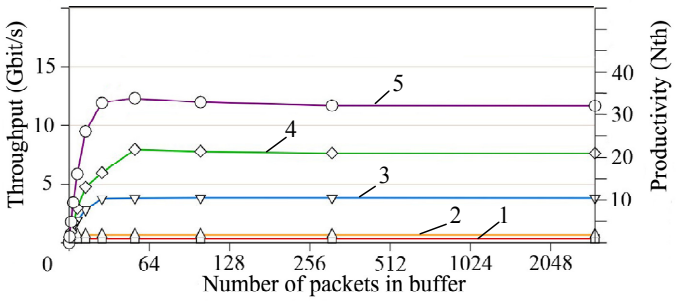

Fig. 2. Comparing throughput of analysis systems: 1 — Central Processor (CPU, one core); 2 — Central Processor (CPU, two core); 3 — Graphic card 7600GS (GP, 12 cores); 4 — Graphic card 9600GT (GP, 96 cores); 5 — Graphic card GTX260 (GP, 216 cores)

Література

1. Sommer, R. Enhancing byte-level network intrusion detection signatures with context / R. Sommer, V. Paxson // Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03), October 27–31, 2003, Washington, DC, USA. — New York: ACM Press, 2003. — PP. 262 — 271.
2. Roesch, M. Snort – Lightweight intrusion detection for networks / M. Roesch // Proceedings of the 13th USENIX Conference on System Administration (LISA'99), November 7–12, 1999, Seattle, WA, USA. — Berkeley, CA: USENIX Assoc., 1999. — PP. 229 — 238.
3. Paxson, V. Bro: a system for detecting network intruders in real-time / V. Paxson // Computer Networks. — 1999. — Vol. 31, Issues 23–24. — PP. 2435 — 2463.
4. Бойко, Ю.В. Методи покращення ефективності для систем високошвидкісної класифікації пакетів / Ю.В. Бойко, К.С. Дєєв // Вісн. ХНУ. Сер.: Математичне моделювання. Інформаційні технології. Автоматизовані системи управління. — 2014. — Вип. 25. — С. 5 — 12.
5. Gnort: High performance network intrusion detection using graphics processors / G. Vasiliadis, S. Antonatos, M. Polychronakis *et al.* // Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID'08), September 15 – 17, 2008, Cambridge, MA, USA. — Heidelberg: Springer-Verlag, 2008. — PP. 116 — 134.
6. Fast and memory-efficient regular expression matching for deep packet inspection / F. Yu, Z. Chen, Y. Diao *et al.* // Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems (ANCS 2006), December 3 – 5, 2006, San Jose, California, USA. — New York: ACM, 2006. — PP. 93—102.
7. NVIDIA CUDA Compute Unified Device Architecture Programming Guide, ver. 1.1 [Електронний ресурс] / NVIDIA. — 2007. — Режим доступу: http://sbel.wisc.edu/Courses/ME964/2008/Documents/Programming_Guide_1.1.pdf (Дата звернення: 03.03.2015).
8. Berry, G. From regular expressions to deterministic automata / G. Berry, R. Sethi // Theoretical Computer Science. — 1986. — Vol. 48. — PP. 117 — 126.
9. PCRE — Perl Compatible Regular Expressions [Електронний ресурс] / P. Hazel. — Режим доступу: http://www.pcre.org (Дата звернення: 03.03.2015).

10. OpenDPI [Електронний ресурс] / T. Bhatia. — 2012. — Режим доступу: https://github.com/thomasbhatia/OpenDPI (Дата звернення: 03.03.2015).
11. Clark, C.R. Efficient reconfigurable logic circuits for matching complex network intrusion detection patterns / C.R. Clark, D.E. Schimmel // Proceedings of 13th International Workshop on Field Programmable Logic and Applications (FPL 2003), September 1 – 3, 2003, Lisbon, Portugal. — Berlin; New York: Springer, 2003. — PP. 956 — 959.

References

1. Sommer, R. and Paxson, V. (2003). Enhancing byte-level network intrusion detection signatures with context. In V. Atluri, P. Liu (Eds.), *Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS'03)* (pp. 262-271). New York: ACM Press.
2. Roesch, M. (1999). Snort – Lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX Conference on System Administration (LISA'99)* (pp. 229-238). Berkeley, CA: USENIX Assoc.
3. Paxson, V. (1999). Bro: a system for detecting network intruders in real-time. *Computer Networks*, 31(2324), 2435-2463.
4. Boyko, Yu.V. and Deev, K.S. (2014). Methods of improvement effectiveness for high-speed packet classifying. *Bulletin of V. Karazin Kharkiv National University: Mathematical modeling. Information technology. Automated control syste*ms, 25, 5-12
5. Vasiliadis, G., Antonatos, S., Polychronakis, M., Markatos, E.P. and Ioannidis, S. (2008). Gnort: High performance network intrusion detection using graphics processors. In R. Lippmann, E. Kirda, A. Trachtenberg (Eds.), *Proceedings of the 11th International Symposium on Recent Advances in Intrusion Detection (RAID'08)* (pp. 116-134). Heidelberg: Springer-Verlag.
6. Yu, F., Chen, Z., Diao, Y., Lakshman, T.V. and Katz, R.H. (2006). Fast and memory-efficient regular expression matching for deep packet inspection. In *Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems (ANCS 2006)* (pp. 93-102). New York: ACM.
7. NVIDIA. (2007). *NVIDIA CUDA Compute Unified Device Architecture Programming Guide, ver. 1.1.* Retrieved from http://sbel.wisc.edu/Courses/ME964/2008/Documents/Programming_Guide_1.1.pdf
8. Berry, G. and Sethi, R. (1986). From regular expressions to deterministic automata. *Theoretical Computer Science*, 48, 117-126.
9. *PCRE — Perl Compatible Regular Expressions*. Retrieved from http://www.pcre.org
10. *OpenDPI*. Retrieved from https://github.com/thomasbhatia/OpenDPI
11. Clark, C.R. and Schimmel, D.E. (2003). Efficient reconfigurable logic circuits for matching complex network intrusion detection patterns. In P.Y.K. Cheung, G.A. Constantinides, J.T. de Sousa (Eds.), *Proceedings of 13th International Workshop on Field Programmable Logic and Applications (FPL 2003)* (pp. 956-959). Berlin; New York: Springer.

Received May 25, 2015