Implementing Intrusion Detection System for Multicore Processor

G. Rajeswari[1], B. Nithya[2,]

[1] *Surya School of Engg. & Tech. / CSE , Villupuram Dist Tamilnadu, India.*
rajilaxman_80@yahoo.co.in
[2] *Surya School of Engg. & Tech. / CSE , Villupuram Dist Tamilnadu, India.*
mano_nithi_78@yahoo.com

***Abstract* - Multi-core processors represent a major evolution in computing hardware technology. Multi-core provides a network security application with more processing power from the hardware perspective. However, there are still significant software design challenges that must be overcome. In this paper, we present new architecture for multi-core supported Intrusion Detection System, which aims at providing network security processing without causing performance penalty to normal network operations. While hardware-based parallelisms have shown their advantage on throughput performance, parallelisms based multi-core provides more flexible, high performance, comprehensive, intelligent, and scalable solutions to network security applications. The Intrusion Detection System that we Presented in this paper also protect the multi core systems from Real Time attacks and Packet Filtrations with high performance without any penalty.**

Keynote – Intrusion, Detection, Multicore processor, Real Time, Network Security

1. INTRODUCTION

Current Internet is facing many serious attacks such as financial frauds, viruses and worms, distributed denial of service attacks, spy ware, and Spam. Although many network security applications such as intrusion detection systems (IDS), anti-virus/Spam systems, and firewalls have been proposed to control the attacks, securing distributed systems and networks is still extremely challenging. There are unknown threats and zero day attacks (exploits released before the vendor patch is released to the public) appearing everyday, which place an impractical burden on network security systems. The key question here is can we have real time solutions to identify and eliminate attacks without excessive security and management overhead overburdening the networks and computer systems? To deal with the rapidly evolving threats today and more intelligent and automatic threats in the future, we urgently need new methods that support network security applications, at all times and in real time, without causing performance penalty to normal network and system operations. A multi-core processor combines two or more independent cores into a single package composed of a single integrated circuit (called a die), or more dies packaged together [1]. Multi-core processors represent a major evolution in computing hardware technology. While two years ago most network processors and personal computer microprocessors had single core configuration, the majority of the current microprocessors contain dual or quad cores and the number of cores on die is expected to grow exponentially over time [2]. As the price of multi-core processors keeps falling, multi-core will eventually provide affordable processing power to support the real-time requirement of network security applications. Multi-core provides a network security application with more processing power from the hardware perspective. [3]. From the server or router side, if the network security software is not fast enough, it can be very difficult to process every incoming packet then it would slow down the traffic. From the client side, it can also be very difficult to run network security applications without any interruption to normal applications because those computing-intensive applications significantly slow down other simultaneously running applications.

II. RELATED WORK

As the Internet traffic volumes and rates continue to race forward, it has become difficult for network security applications to process network packets in real-time. Many network security applications nowadays can process network packets at Mbps level. However, most network backbones and many local network interfaces operate at Gbps level. To improve the performance of the network security applications, most previous research focus on parallelism with the hardware approaches such as ASICs and FPGAs [4-8]. They require highly deliberate and customized programming, which is directly at odds with the pressing need to perform diverse, increasingly sophisticated forms of analysis. In [9] the authors argued that it is time to fundamentally rethink the nature of using hardware to support network security applications. Previously, efforts in multi-core software design has been primarily on simultaneous multi-threading (SMT) [10, 11] at a low level, which permits multiple independent threads of execution to better utilize the resources provided by microprocessor architectures. Most of current research is still focused on automatically mapping general-purpose applications onto multi-core systems with instruction, data, or thread level parallelization techniques [12-16] or relying on virtualization technologies such as VMware [17]. Most of them are essentially extensions of utilizing shared-memory multiprocessors and can only execute coarse-grained threads. Network security applications have their own unique behavioral characteristics such as frequent memory or disk access, complex data structures, and high bandwidth and high speed requirements. There is a distinct mismatch between current multi-core hardware development and high performance demand from network security applications. There has been very little preliminary research done in this area [18, 19].

III. DEVELOPING MULTI-CORE SUPPORTED NETWORK SECURITY APPLICATIONS

3.1. System Architecture

The idea of using multi-core processors to enhance the performance of network security applications is promising. However, the research in this area is just emerging and thus requires intensive exploration. It faces many challenges such as □□How can we actually use multi-core to continue running the network security applications while keeping the overall system performance?

- ❖ How can we efficiently partition and distribute the workload of network security applications between the different cores in the multi-core processor?
- ❖ How can we split network data and solve the data dependency problem?
- ❖ As multi-core uses shared off-chip memory, how can we smartly utilize the memory then it will bring less memory access latencies?
- ❖ How can we synchronize and coordinate different threads of the applications when it is parallelized on multi-core?

The essential ability of this architecture is that it can process network packets in parallel and thus meet the real-time requirement. The multiprocessing scheduler coordinates and distributes the workload to different cores. The information from packets, events, flows, and messages are processed in the multi-core processor in parallel. The processor has spare cores to run other applications. To fully utilize the potential of multi-core, this system architecture will use different level of parallelization such as instruction-level parallelization, memory parallelization, loop-level parallelization, and fine-grained thread-level parallelization. High performance can be achieved through interaction between algorithms, strategies, and architectural design, from high-level decisions on data allocation and task partitioning to low-level micro architectural decisions on instruction selection and scheduling. For each network security application, we also need to identify what the potential bottlenecks are and how to possibly avoid them. The potential bottlenecks could be packet processing, data normalization, data correlation, pattern generation, and pattern matching in such a parallel computing environment.

3.2. Benefits of Using Multi-core Supported System Architecture

We summarize the benefits of using multi-core supported system architecture in network security applications as high performance, comprehensive, intelligent, and scalable.

Firstly, traditional network security applications are based on serial or very limited parallel execution of packet processing which includes shallow packet inspection and deep packet inspection [5]. Shallow packet inspection is a form of computer network packet filtering that examines only the header part of a packet. Deep packet inspection examines both the header and payload of a packet as it passes an inspection point, searches for non-protocol compliance, viruses, spam, intrusions or predefined criteria to decide if the packet can pass or if it needs to be routed to a different destination. For example, traditional single-threaded network-based intrusion detection systems log activities that it finds to a safeguarded database and detects if the events match any malicious event recorded in the knowledge base. It must read packet level information and process it on the processor in serial. Secondly, multi-core supported network security applications can provide comprehensive protection against different threats. Currently, if a router performs deep packet inspection, for example, to check a certain virus signature in the packet's payload, its forwarding capability will be significantly affected. Most network providers cannot afford to slow down traffic to perform such security operations. Another fact is that current computer systems can only separately run a single network security application at a time because these computing-intensive network security applications exclusively occupy CPU time. With the support from multi-core and application level parallelization, these computing tasks can be divided into many threads and distributed to different cores for processing. Thus if we have enough cores, the network security applications will then be virtually invisible to users because other applications still have free cores to perform their tasks. This enables comprehensive protection as it can integrate as many modules (such as intrusion detection module, anti-virus module, and anti-spam module) as necessary.

Thirdly, with the support from multi-core, network security applications will have greatly improved intelligence compared to traditional applications because we can employ many computing-intensive methods to perform packet inspection and classification and anomaly detection. In [11] we have tested the performance of using neural network to detect attack packets with the aid of packet marking schemes. It has advantages such as high detection rate and low false positive rate because it relies on more intelligent method rather than signature matching. However, it also has the limitation of long training time, thus it cannot provide real-time protection.

In this paper we improve the performance of the intrusion detection system by utilizing the power of multi-core. Lastly, multi-core supported network security applications are scalable. They can be used on not only network level devices but also end host level devices. As we know, pure network-based security applications cannot fully capture the profile of each end host. Therefore in order to achieve the best protection, security checks must be performed on both network processing devices and end hosts. Many tasks that must be done on infrastructure level computing nodes before can now be moved to the far end of personal computers, which not only alleviate the load of the information infrastructure but also make the security check more meaningful. On the other side, many tasks that must be done on end host level before can now be performed at the infrastructure level, such as checking virus signatures, which can effectively prevent the propagation of malicious codes from reaching the end hosts. They are also customizable according to their scalability for different requirements because switching different parallel applications on or off becomes easy with the support from multi-core. This feature makes our approach distinct from other traditional network security applications.

VI. A MULTI-CORE SUPPORTED INTRUSION DETECTION SYSTEM

4.1. The Design of Multi-core Supported IDS

After given the architecture in the previous section, we present a concrete instance of such a system in this section. We incorporate the system architecture into an intrusion detection system. Intrusion detection techniques can be classified into two categories: signature matching and anomaly detection. Signature matching uses patterns of known attacks or weak spots of the system to match and identify known intrusions. Signature matching can detect known attacks, though it usually cannot accommodate unknown attacks. Anomaly detection models a user's behaviors, and any significant deviation from the normal behaviors is considered the result of an attack. Anomaly detection techniques can be effective against unknown or novel attacks since no a prior knowledge about specific intrusions is required. However, anomaly detection system requires intensive computation power. In our previous research [9], we have used a 3- layer back-propagation neural network based anomaly detection system to detect distributed denial of service (DDoS) attacks. It finds the network anomalies by using neural network, deploy the system at distributed routers, identify the attack packets, and then filter them. Number of source IP address, number of destination IP address, source port, destination port, total length of packets, number of wrong checksum, number of TCP SYN flag, number of TCP FIN flag, number of TCP ACK flag, and concentration of the packets with same digest bits of Flexible Deterministic Packet Marking (FDPM) [10] are used as the features as the input of the neural network for training and test. We then parallelize it as multi-thread intrusion detection system. The neurons belonging to the same layer can be run in parallel. For example, any neuron of the same needs the outputs of the first hidden layer but not from other neurons within its own layer. Therefore the computation can be pipelined by the multiprocessing scheduler through different cores in the multi-core processor.

V. CONCLUSION

Leveraging the power of multi-core processors can be the answer to many yet-unsolved but crucial challenges in network security applications such as isolated security environment, real-time attack detection and attack packets filtering, real-time visualization of network monitoring and real-time verifying of critical flaws and exploits. It enables sophisticated and stateful network processing rich in semantics and context as a routine capability provided by a network's routers. The use of multi-core will support flexible recompilation of security software but rather than redesign of hardware. It will provide significant benefits to the security of future distributed networks and systems.

VI. REFERENCES

[1] Intel, Intel® Multi-Core: An Overview, http://www.intel.com/multi-ore/overview.htm, visited December 2007.

[2] C. Johnson and J. Welser, "Future Processors: Flexible and Modular", Proceedings of 3rd IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, pp. 4-6, 2005.

[3] H. Sutter and J. Larus, "Software and the Concurrency Revolution", ACM Queue, vol. 3, no. 7, pp. 54-62, 2005.

[4] O. Villa, D. P. Scarpazza and F. Petrini, "Accelerating Real-Time String Searching with Multicore Processors", IEEE Computer, vol. 41, no. 4, pp. 42-50, 2008.

[5] S. Dharmapurikar, P. Krishnamurthy, T. S. Sproull, J. W. Lockwood, "Deep Packet Inspection Using Parallel Bloom Filters", IEEE Micro, vol. 24, no. 1, pp. 52-61, 2004.

[6] H. Liu, K. Zheng, B. Liu, X. Zhang and Y. Liu, “A Memory-Efficient Parallel String Matching Architecture for High-Speed Intrusion Detection”, IEEE Journal on Selected Areas in Communications, vol. 24, no. 10, pp. 1793-1804, 2006.

[7] C. L. Hayes and Y. Luo, “DPICO: A High Speed Deep Packet Inspection Engine Using Compact Finite Automata”, Proceedings of ACM/IEEE ANCS’07, pp. 195-203, 2007.

[8] P. Piyachon and Y. Luo, “Efficient Memory Utilization on Network Processors for Deep Packet Inspection”, Proceedings of ACM/IEEE ANCS’06, pp. 71-80, 2006.

[9] V. Paxson, K. Asanovi_, S. Dharmapurikar, J. Lockwood,R. Pang, R. Sommer and N. Weaver, “Rethinking Hardware Support for Network Analysis and Intrusion Prevention”, Proceedings of the 1st conference on USENIX Workshop on Hot Topics in Security, 2006.

[10] S. Eggers, J. Emer, H. Levy, J. Lo, R. Stamm and D. Tullsen, "Simultaneous Multithreading: A Platform for Nextgeneration Processors", IEEE Micro, vol. 17, no. 5, pp. 12-19, 1997.

[11] D. Tullsen, J. Lo, S. Eggers and H. Levy, "Supporting Fine-Grain Synchronization on a Simultaneous Multithreaded Processor", Proceedings of the 5th International Symposium on High Performance Computer Architecture, pp. 54, 1999.