Chapter 37
Multi-Core Supported
Deep Packet Inspection

Yang Xiang
Central Queensland University, Australia

Daxin Tian
Tianjin University, China

ABSTRACT
Network security applications such as intrusion detection systems (IDSs), firewalls, anti-virus/spyware systems, anti-spam systems, and security visualisation applications are all computing-intensive applications. These applications all heavily rely on deep packet inspection, which is to examine the content of each network packet’s payload. Today these security applications cannot cope with the speed of broadband Internet that has already been deployed, that is, the processor power is much slower than the bandwidth power. Recently the development of multi-core processors brings more processing power. 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. The purpose of this chapter is to discuss the research on using multi-core technologies to parallelize deep packet inspection algorithms, and how such an approach will improve the performance of deep packet inspection applications. This will eventually provide a security system the capability of real-time packet inspection thus significantly improve the overall status of security on current Internet infrastructure.

1. INTRODUCTION
Current Internet is facing many serious attacks such as financial frauds, viruses and worms, distributed denial of service attacks, spyware, 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 (Intel, 2007). 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 (Johnson & Welser, 2005). 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. However, there are still significant software design challenges that must be overcome. Today the difficulty is not in building multi-core hardware, but programming it in a way that lets applications benefit from the continued growth in CPU performance (Sutter & Larus, 2005). 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.

Taking advantage of the full power of multi-core processor requires an in-depth approach to realize the speedups by parallelizing the traditional deep packet inspection applications. In this chapter we discuss the research direction of using multi-core processors to support real-time deep packet inspection applications. Section 2 introduces the related work in the parallel approaches to enhance the performance of deep packet inspection applications. Section 3 presents our new system architecture of using multi-core to support deep packet inspection applications. Section 4 presents the basic packet-level parallelization and flow-level parallelization. Section 5 presents a new parallel string matching algorithm. Benefits of using multi-core are discussed in Section 6. Section 7 concludes this chapter.

2. RELATED WORK

2.1 Development of Multi-Core Processors

In 1965, Gordon Moore observed an exponential growth in the number of transistors per integrated circuit and predicted that this trend would continue - a prediction today known as Moore’s Law (Moore, 1965). In reality, the doubling of transistors every couple of years has been maintained for almost 40 years. However, scaling up the processor’s frequency has been more difficult because of several constraints revealed recently. First, memory speeds are not increasing as quickly as processor’s logic speeds. Now the processor takes more clock cycles to access memory than before. The wasted clock cycles can nul-
Related Content

Making Scientific Applications on the Grid Reliable Through Flexibility Approaches Borrowed from Service Compositions
www.igi-global.com/chapter/making-scientific-applications-grid-reliable/40821?camid=4v1a

Interoperable PKI Data Distribution in Computational Grids
www.igi-global.com/article/interoperable-pki-data-distribution-computational/3966?camid=4v1a

Scheduling Strategies for Business Process Applications in Cloud Environments
www.igi-global.com/article/scheduling-strategies-for-business-process-applications-in-cloud-environments/102757?camid=4v1a

Managing Inconsistencies in Data Grid Environments: A Practical Approach
www.igi-global.com/article/managing-inconsistencies-data-grid-environments/47211?camid=4v1a