Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 31 (2014) 852 – 859

Procedia Computer Science

Information Technology and Quantitative Management (ITQM 2014)

A Real-time Updatable FPGA-based Architecture for Fast Regular Expression Matching

Qiu Tang[a]*, Lei Jiang[a], Xin-xing Liu[a], Qiong Dai[a]

[a]Institute of Information Engineering，Chinese Academy of Sciences， No 89, Mingzhuang RD.,Haidian District, Beijing,100089, China

Abstract

In recent years, regular expression has been widely used in many network fields, but more and more applications require real-time update FSM and a space reduced DFA due to limited memory capacity. In this paper, we firstly propose a new architecture on FPGA supporting real-time update FSM, and design a special protocol for this update. Secondly, in order to support large-scale and complex semantic regular expression rule sets, we design an improved run-length encoding (iRLE) algorithm based on FPGA to reduce the DFA's storage space. The proposed algorithm gains a good compression ratio and requires only 2 clock cycles per a state transition. The experimental results also show that the new algorithm has both advantages of compressing ratio and speed, and the maximum throughput of the automaton can reach 10.7Gbps.

Keywords: Regular Expression Matching, Finite State Machine (FSM), Field Programmable Gate Array (FPGA) , Run Length Encoding (RLE)

1. Introduction

As we all know, regular expressions (regex) with a power of description, in recent years, have been widely used in many fields of networking, such as traffic control and optimization, content audit, service billing, intrusion detection system (IDS), user behavior analysis, and information filter. The premise of these applications can be achieved is that can perform line-speed deep packet inspection (DPI) using regular expression matching technique, which means that comparing every bytes of the real-time income stream with a great number of regular expression rules determine whether a particular string described by regular expressions occurs in input stream or not. However, both bandwidth of network and scale of regular expression rules set are

* Corresponding author. Tel.: +0086-010-82858269-8012.
E-mail address: tangqiu@iie.ac.cn

doi:10.1016/j.procs.2014.05.336

growing. For large scale regular expression rule sets, performing deep packet inspection incurs performance bottleneck in terms of line-speed content matching and state explosion of automata due to complex rule semantics. Therefore, facing the gigabit per second real-time network flow, the traditional software-based solution for deep packet inspection is hard to meet the requirement of real-time matching input data, because the capacity of memory and the compute power of processor are limited, while the solution based on reconfigurable hardware, like FPGA, can perform regular expression matching in line-speed on the ground of that it can concurrently compare each byte of importing stream with each rule of regular expression sets. In recent years, FPGA-based regular expression matching technology has become a new research hotspot. This paper focuses on using FPGA to design a fast and efficient regular expression matching system.

Traditionally, finite state machine (FSM) is used to implement regular expression matching. There are two main types of FMS: the non-deterministic finite automata (NFA) and deterministic finite automata (DFA). The space complexity of NFA is O(n) and its searching time complexity for per input character is O(n), while DFA's storage complexity is $O(2^n)$ in the worst case and its searching time complexity is O(1), where *n* represents the length of regular expression. In FPGA platform, the NFA's searching time can reduce to O(1), such as [1], but it is hard to update the FSM real-timely because its implement is based on circuit. To a certain extent, this problem has hindered the application of NFA in real-time monitor system, such as IDS. Because DPI need to real-timely respond to the income packets and does not allow a long time to update the on-chip automaton when the rule sets changed. However, DFA approaches based on FPGA matching one input character is still O(1) like software implementation, but it consumes a great quantity of memories due to $O(2^n)$ of storage space in the worst case. And a DFA based on FPGA can real-timely update the automata because the DFA is a memory-based FSM and its memory updating, compared with circuit-based FSM, is relatively easy to implement.

The rest of the paper is organized as follows. Section 2 briefly mentions related work, while section 3 illustrates the details of DFA real-time update structure and implementation. In section 4 provides the principle and the implementation of iRLE DFA compression algorithm. Section 5 describes the experimental evaluation, and finally, section 6 concludes our work and suggests next possible ways of our research.

2. Related Work

In recent years, many researchers have proposed a lot of effective regular expression matching algorithms. In 2001, R. Sidhu and V. K. Prasanna [1] firstly implement regular expression matching on FPGA using non-deterministic finite automata. Their design special algorithm to compile regular expressions into VHDL described circuit. Since then, it has aroused many related research work such as [2-8], which concern about how to take full advantage of FPGA resources to speed up regular expression matching. Hayes C. L. with Luo Y. [2] propose a named DPICO FSM, the DPICO merge transitions that are the maximum number of transitions from one state to an identical state into a default transition. The FSM will select the default transition to transfer when it fails to meet the right transition. Their experimental results show that the DPICO can reduce more than 90% storage space and its compression rate is inverse ratio to transition factor of number of transitions to number of states. In [3] Yi-Hua et al. propose a method can directly map NFA into a circuit, the method can efficiently match single character constrained repetitions by using parallel shift-register lookup tables and performance complex character classes matched using a block memory based classifier shared among regular expressions. Their experimental results show that their FSM's throughput can reach 11Gbps on Xilinx Virtex 5 chip with Snort regular expression set.

In order to reduce memory redundancy of DFA, many works study how to reduce the space redundancy of a DFA transitions matrix in recent years, such as D^2FA [9] and δ FA [10]. The main ideas of these algorithms are to eliminate the redundancy of DFA by introducing a new compressed DFA structure, and many DFA compression algorithms make a tradeoff between matching speed and storage space. Kumar et al. [9] observe