Fast Classification of XML with Network Processors

Philip Gross
Dept. of Computer Science
Columbia University, New York, NY
phil@cs.columbia.edu

Abstract

As XML documents become steadily more prevalent for data storage and communication, there is a growing need for routing XML messages in a network. Ideally one would be able to route XML packets at line speed.

An important technology for complex packet routing are so-called Network Processors, which have a set of small "microengines" classifying incoming packets.

This paper describes an attempt to implement XML packet classification on the Intel IXA network architecture.

1 Introduction

Extensible Markup Language (XML) is becoming a ubiquitous data format for information storage and interchange. As the standard spreads, XML documents are increasingly being sent directly across networks. Routing of these documents is frequently based on the contents of the document itself. For routing purposes, a document will belong some equivalence class that determines the set of recipients. An important goal, therefore, is high-speed *classification* of XML documents as they are sent across the network.

Meanwhile, the exponential growth in size and complexity of the Internet has created a major classification problem of a different sort. Internet Protocol (IP) packets are being broadcast onto Ethernets at gigabit speeds, and need to be routed (i.e. classified) according to complex policies and protocols to their destination. In addition to the custom, proprietary chips that have been developed to solve this problem, a new class of programmable *Network Processors* has appeared. These typically contain a master CPU core that manages routing policies and tables, and a number of *microengines* that classify and route packets based on the current contents of the routing tables. For example, the IXA family from Intel have an ARM-based CPU sharing the die with six to sixteen microengines.

The goal of this experiment is to implement an XML packet classifier on Network Processor hardware.

2 XML Packet Classification

There are a number of difficulties with high–speed XML classification. The first is the complexity of the data and filters. XML is a tree–based data structure, and the most common form of filters, known as XPath Expressions (XPEs), are basically subtrees with wildcards. Given a set of XPEs, the goal is to determine what subset matches a particular document.

When one begins looking at Network Processors (NPs), additional problems appear. Although programmable, the design of NPs is overwhelmingly oriented towards IP (and TCP) packet classification. Classifying a TCP packet is almost always based purely on five fields in the packet header: protocol, source address, source port, destination address, and destination port. NPs are optimized for extracting this information, hashing it, and doing a lookup. More complex algorithms are possible, but challenging.

NPs have additional complexity. The memory hierarchy is exposed, requiring the programmer to manage the use of the 4kb scratchpad memory, the 8MB of fast static RAM, and the hundreds of megabytes of slow SDRAM. Retrieving data from SRAM can take as much time as a dozen microengine instructions, while retrievals from SDRAM may take over a hundred.

Finally, although the many microengines of an IXA NP look like a rich computational resource, they have a mere 1–2kb of program memory each. Additionally, data is presented to them not as complete Ethernet frames but rather as 64byte MAC packets, so two microengines must be dedicated to simply assembling and disassembling incoming and outgoing packets.

3 Related Work

Intel [2] describes programming their IXP1200 processor at the assembler level (Intel calls it "microcode").

Fast filtering of XML documents has been attracting increasing attention in recent years. The XTrie-based approach of Chan et al. [1] appears to be the most promising at the moment, due both to its excellent performance, as well as its specific goal of reducing memory accesses.

Given the incredibly constrained computational resources available in a Network Processor, actually parsing ASCII-formatted XML will probably be impossible. The XML will need to be reduced to a binary format prior to transmission. Intelligent compression of XML has been attempted in a number of projects, such as Liefke and Suciu's XMill [4] and Girardot and Sundaresan's Millau [3].

References

[1] Chee Yong Chan, Pascal Felber, Minos N. Garofalakis, and Rajeev Rastogi. Efficient filtering of XML documents with XPath expressions. In *ICDE*, 2002.

[2] Intel Corporation. *Intel IXP1200 Network Processor Family*. Intel Corporation, 2002.

[3] Marc Girardot and Neel Sundaresan. Millau: an encoding format for efficient representation and exchange of xml over the web. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam, the Netherlands, May 15–19 2000.

[4] Hartmut Liefke and Dan Suciu. XMill: an efficient compressor for XML data. pages 153–164, 2000.