Architectural Considerations for Network Processor Design

Literature Survey

The University of Texas at Austin

Department of Electrical and Computer Engineering

David N. Armstrong

25 March 2002

ABSTRACT

In current computer networks, the transmission links between network nodes are generally not the performance bottleneck in the system. Instead, the bottleneck tends to lie at the network nodes, specifically, the processing that goes on at the network nodes, such as packet forwarding. Network processors are processors tailored towards the computer network space and generally perform packet processing operations. Network processors address the performance bottleneck in computer network design while maintaining the flexibility to adapt to future network protocols. This literature survey examines the design characteristics of network processor architectures. This survey goes on to describe the project objective and implementation issues, which have to do with network processor designs that attain high performance and flexibility.

I. INTRODUCTION

Network processors are processors tailored towards the computer network space, specifically packet processing. Network processors, which are high-performance and programmable solutions, offer an alternative to packet processing ASICs, which are high-performance solutions but are not programmable, and also to general purpose processors, which are often lower performing devices, but are programmable. Network Processor design is becoming an important research topic as the use of network processor solutions is taking hold in the marketplace.

This paper is organized as follows. Section II gives an overview of network processors in today's computer networks. Section III describes previous work that has been conducted regarding network processor design. Section IV gives the objectives of this project and discusses the implementation issues involved. Section V concludes.

II. OVERVIEW OF NETWORK PROCESSORS

In current computer networks, the transmission links between network nodes are generally not the performance bottleneck in the system. This is especially the case now as high bandwidth links such as fiber optics are deployed extensively, [11]. Instead, the bottleneck tends to lie at the network nodes, specifically, the processing that goes on at the network nodes, such as packet forwarding, [10, 3, 2].

In the past, a response to this performance bottleneck has been a trend towards using dedicated hardware, or ASICs, to perform the processing at the network nodes instead of software running on general purpose processors. The dedicated hardware, by virtue of it being intended for only a limited set of operations, might be optimized to perform those operations and thereby attain higher performance. However, the ASIC

based solution carries with it a number of disadvantages. The development cycle for ASICs tends to be significant, in terms of both time and cost. The time it takes to develop an ASIC, say, 18 months, [3], introduces the problem that a solution may be ready to be deployed in the market only after the solution has become obsolete. This is becoming a more likely problem especially for ASICs deployed at the edges of computer networks, where network protocols change frequently. It is also the case that it is desirable now for network nodes to perform an increasingly diverse set of operations, [2]. Finally, ASIC solutions tend to be quite expensive [3].

The disadvantages of ASIC based solutions, their expensive design cycle and the fact that they are not programmable, have motivated the development of network processors. Network processors are intended to straddle the performance vs. flexibility trade-off between ASIC based solutions and software running on general purpose processors. Network processors are able to run software that can perform the same operations as the ASIC and general purpose processor solutions. This gives them the flexibility to be re-programmed to accommodate changes in network protocols. Network processors are able to attain high performance by virtue of their designs, which are meant to take advantage of the parallel workload of network data, [2, 3, 1, 4, 6, 11, 15, 16].

The workload of a network processor is composed of packet processing operations on the network traffic (packets) flowing through the node. These packets usually originate from different flows, or source addresses. As such, the packets tend to be independent of one another and therefore provide significant parallelism for the network processor to exploit. The microarchitecture of network processors is designed to take advantage of this parallel workload.

Network processors address the design point of flexibility as well as offer a solution to the steep development cost of ASICs, by way of being programmable. That is, network processors are able to adapt to changes in network protocols, or to perform additional functions that were not necessarily foreseen when the network processor was originally deployed. The software running on the network processor can be quickly an inexpensively updated or reprogrammed to accommodate necessary changes.

In sum, the advantage of the network processor solution is that the processor offers both comparable performance to the special purpose hardware yet also provides desirable flexibility by way of being programmable.

III. Previous Work

The relevant previous studies include the following. Partridge, et al. have examined the design of high performance routers, and offer an example of a network processor solution, [10]. Wolf, et al. studied the performance trade-offs of the allocation of transistors between on chip caches, the number of I/O channels and the number of processor cores, [16]. Crowley, et al. examined the optimal characteristics of network processor architectures with a focus on optimal support for multithreaded architectures, [2]. Finally, Halfill sheds insight into Intel's solution to a high performance network processor, [3]. The proposed project most closely tracks [2] and therefore this literature survey provides the most detailed discussion of [2].

A. The Context of Network Processors: High Performance Router Design

The importance of [10] is that Partridge, et al. establish the case for high performance routers in current computer networks by noting the demand for network performance and observing that packet routing is a bottleneck. The paper goes on to

explore the design of high performance routers and to identify some of the design issues. Their observations are important in order to realistically model network processors and their workload, as discussed in the next section.

B. Network Processor Architecture

In [2], Crowley, et al. study the performance differences between various computer architectures running simulated network processor workloads. The workloads include packet forwarding routines as well as data encryption and authentication routines. The computer architecture models examined in [2] include superscalar processors, fine grain multithreaded processors, single chip multiprocessors and simultaneous multithreaded processors. The superscalar processors studied in Crowley issue multiple instructions per cycle and execute instructions out of order. Fine grain multithreaded processors are similar to superscalar processors and add hardware support for multiple threads: each cycle the processor can fetch instructions from different threads. Chip multi-processors are single chips partitioned into multiple separate independent processors where each processor can operate on a different thread, [8]. Simultaneous multithreaded processors extend the fine grained multithreaded processors further by enabling the processor to fetch instructions from different threads within the same cycle, [14]. Crowley, et al. determine which of the proposed computer architectures are best able to “discover” and exploit the parallel nature of the network processor workload. The superscalar processors attain the least amount of performance due to the limitation that they are bound to discover parallelism only on a per thread basis since they operate on only one thread at a time. The fine grain multithreaded processors perform slightly better than the superscalar processor but are still bound to discover parallelism on a per thread

basis. The chip multiprocessor and simultaneous multithreaded processors achieve the greatest performance as a result of their ability to take the largest view in discovering parallelism both within single threads and also across multiple threads. Crowley et al. determine that single chip multiprocessors and simultaneous multithreaded processors are best suited for network processor workloads when ignoring the affects of operating system overhead.

C. Case Study: Intel's IXP1200

Halfill's work explores Intel's solution to network processor design in their IXP1200 chip, [3]. The article describes Intel's chip multiprocessing solution where a single StrongArm core controls 6 identical microengines, or RISC cores. The microengines are independent cores each with hardware support for executing up to 4 different threads. The microengine model in the IXP1200 corresponds most closely to Crowley, et al.'s description of chip multiprocessors. The design of Intel's IXP1200 demonstrates the desire for network processors to take advantage of parallel workloads, as Crowley, et al. showed in [2]. An additional aspect taken into consideration by Halfill is the instruction set of the microengines on the IXP1200, which the designers have developed to specifically support packet processing operations.

IV. THE PROJECT

A. Objectives

The project corresponds to the work of Crowley et al. in [2], and explores the computer architectures best suited for network processor designs. Crowley et al. determined the optimal design of network processor architectures under workloads that represented IP forwarding, and data encryption and authentication. This project will

explore the architectures best suited to achieve high performance and also to tolerate changes in the network processor workload. It follows from the network processor's design point of flexibility that network processors must achieve sufficient performance on workloads other than those for which the network processor was originally deployed. For example, network processors might be called upon to perform quality of service guarantees for computer networks in the near future. This project will address the flexibility design point and use abstract models for the different network processor architectures: superscalar processors, fine grain multithreaded processors, chip multiprocessors and simultaneous multithreaded processors. The workload used to evaluate these different computer architectures will vary over the amount of processing required for each packet as well as the number of dependencies that exist between the packets flowing through the processors.

B. Implementation

Various implementation methodologies are studied for the processor and workload models. Publicly available software or custom simulators could be used to implement the processors and workloads. A computational model is used to simulate the different processor configurations and the data flowing through the network processor. At this point, a dataflow process network [9] implemented in C, is used to model the processor configurations and workload. This model supports the dynamic nature of the packets flowing through the processor and also provides the necessary flexibility in modeling the network traffic workload. The model is implemented in the following manner. The nodes of the computational model represent the processing elements of the different network processors, e.g., the functional units of a superscalar processor. The

edges of the computational model in turn represent the interconnections between the processing elements as well as the connections between the processing elements and the input network packets. The topology of the edges and network nodes determines the architecture of the network processor, e.g., multiple connections to input ports represent multithreaded processors. The tokens of the computational model represent both interprocessor communication as well as network packets. The schedule is a function of the workload used to evaluate the processor configurations and is randomized. Performance is measured by the number of packets processed per second. The flexibility of the system is evaluated based on how quickly the performance deteriorates as the workload is varied. The algorithms performed by the network processors are abstracted away such that they consist of only the latency at the network nodes and the amount of inter-processor communication between the processor cores.

V. Summary and Conclusion

Network processors offer an alternative to ASIC or general purpose processors to perform packet processing at the nodes of computer networks. Network processors offer a compromise between the high performance of an ASIC based solution and the flexibility of general purpose processors. This survey reviews literature which makes the case for network processors in today's computer networks, explores the design alternatives in the computer architecture of network processors and reviews a current network processor, the Intel IXP1200. The project continues to explore high performance architectures for network processors. The project further examines the degree of flexibility offered by the different architectures in an effort to provide for tomorrow's computer networks.

REFERENCES

[1] B. CHEN, R. MORRIS. "Flexible Control of Parallelism in a Multiprocessor PC Router." *Proc. of the USENIX 2001 Annual Technical Conference*, Boston, MA, June 2001.

[2] P. CROWLEY, M. E. FIUCZYNSKI, J. L. BAER, B. N. BERSHAD. "Characterizing Processor Architectures for Programmable Network Interfaces." Proc. of the 2000 International Conference on Supercomputing, Santa Fe, NM, May, 2000.

[3] T. R. HALFHILL. "Intel Network Processor Targets Routers." Microprocessor Report, 13(12), September 13, 1999.

[4] INTEL CORPORATION. *Next Generation Network Processor Technologies*, 2001.

[5] S. KESHAV, R. SHARMA. "Issues and Trends in Router Design." *IEEE Communications Magazine*, May 1998, pp. 144-151.

[6] LEVEL ONE COMMUNICATIONS, INC. Applications of the Level One IXP1200 Network Processor, 1999.

[7] G. MEMIK, W. H. MANGIONE-SMITH, W. HU. "NetBench: A Benchmarking Suite for Network Processors." *Proc. of IEEE/ACM International Conference on Computer Aided Design (ICCAD)*, San Jose, CA, Nov. 2001, pp. 39-42.

[8] B. A. NAYHFEH, L. HAMMOND, K. OLUKOTUN. "Evaluation of Design Alternatives for a Multiprocessor Microprocessor." *Proc. of the 23rd International Symposium on Computer Architecture*, Philadelphia, PA, May 1996, pp. 67-77.

[9] J. PARKS. *Bounded Scheduling of Process Networks*, PhD Dissertation, Technical Report UCB/ERL-95105, PhD Dissertation, EECS Department, University of California. Berkeley, CA 94720, December 1995.

[10] C. PARTRIDGE, P. P. CARVEY, E. BURGESS, I. CASTINEYRA, T. CLARKE, L. GRAHAM, M. HATHAWAY, P. HERMAN, A. KING, S. KOHALMI, T. MA, J. MCALLEN, T. MENDEZ, W. C. MILLIKEN, R. PETTYJOHN, J. ROKOSZ, J. SEEGER, M. SOLLINS, S. STORCH, B. TOBER, G. D. TROXEL, D. WAITZMAN, S. WINTERBLE. "A 50-Gb/s IP Router." IEEE/ACM Transactions on Networking, 6(3), June 1998.

[11] P. G. PAULIN, F. KARIM, P. BROMLEY. "Network Processors: A Perspective on Market Requirements, Processor Architectures and Embedded S/W Tools." *Proc. of the Design Automation and Test in Europe*, November 2000, pp. 420-427.

[12] T. SPALINK, S. KARLIN, L. PETERSON. "Evaluating Network Processors in IP Forwarding." *Princeton University Technical Report TR-626-00*, November 2000.

[13] T. SPALINK, S. KARLIN, L. PETERSON, Y. GOTTLIEB, "Building a Robust Software-Based Router Using Network Processors." *Proc. of the 18th ACM Symposium on Operating Systems Principles (SOSP'01)*, Chateau Lake Louise, Banff, Alberta, Canada, October 2001, pp. 216-229.

[14] D. TULLSEN, S. EGGERS, H. LEVY, "Simultaneous Multithreading: Maximizing On-Chip Parallelism." *Proc. of the 22nd Annual Symposium on Computer Architecture*, Santa Margherita Ligure, Italy, June 1995, pp. 392-403.

[15] J. WILLIAMS. "Architectures for Network Processing." *Proc. of International Symposium on VLSI Technology, Systems and Applications*, 2001, pp. 61-64.

[16] T. WOLF, M. A. FRANKLIN, E. W. SPITZNAGEL. "Design Tradeoffs for Embedded Network Processors." To appear in *Proc. of International Conference on Architecture of Computing Systems (ARCS)*, Karlsruhe, Germany, April 2002.