High Performance Networking Hardware Design

Gokhan Memik (PI)

Department of Electrical Engineering and Computer Science, Northwestern University

Abstract

The growth of the Internet is pushing the capabilities of network infrastructures to their limits and beyond. Response times are slowed, storage is overflowing, and existing system design methodologies are insufficient. Its growing popularity for entirely new applications in the field of e-business and entertainment, as well as its emerging use for well-established applications such as telephony, have resulted in an annual growth factor of at least 4 for traffic carried by the Internet. The Internet is able to grow at this enormous pace because of the exponential growth of optical transmission bandwidth made possible by wavelength division multiplexing (WDM) and commensurate progress in the packet-forwarding capabilities of network nodes. To be able to further increase the capabilities of the Internet as well as other testbeds (e.g., DoE UltraScience Net [5]) and to keep up with the increasing number of users in such networks, packet-forwarding capabilities of network nodes should be able to scale.

Traditionally, the packet-forwarding performance of network nodes is increased by a) increasing the parallelism, and b) by utilizing better manufacturing/architectural technologies and hence increasing the performance of a single processing node. Both of these approaches have inherent limitations. In this work, we will explore other novel techniques to achieve this goal. Particularly, our main approach is to propagate application-specific information to the circuit-level design phase of a processor. We call this approach *vertical integration*. Certain integration efforts already exist in various domains. Our main contribution in this work is to develop the ability of the applications (and even packets) to control the properties of hardware operations. In other words, the ability to integrate the user experience with the hardware behavior. One intriguing example the project particularly targets is the integration of correctness information into the hardware design process. The traditional approach in processor design has been: "regardless of the properties of a processor/system, it must be 100% correct". When networking hardware is considered, this assumption is indeed not absolutely necessary. If the hardware makes a mistake, the higher levels (e.g., TCP) will still maintain the system stability. In this work, we question this assumption and aim to show that by relaxing the correctness constraint, significant performance improvements can be achieved. Particularly, our preliminary results indicate that the performance of a processor can be increased 2 to 4 times without seeing a significant change (less than 1 error in 100,000 iterations) in the behavior of a set of applications. To engineer such processors, we need to answer several question. First, we need to be able to measure the severity of an error. Second, we need to associate a particular hardware error with the corresponding application behavior. Third, we need to engineer the optimal operation point for the hardware and finally, we need to evaluate the effects of such optimization on the overall system. These activities are described in the following section.

Major Research Activities

The research in this project is divided into four major tasks: understanding/quantifying application errors, development and analysis of fault throttling techniques (i.e., tools for error vs. performance trade-off), the engineering of clumsy packet processors, and reliability trade-offs for network processors. Our activities in the last six months and the future activities planned on these topics are described in the following sections.

Understanding/Quantifying Application Errors

Independent of this project, being able to understand the severity of application errors in the networking domain is an intriguing problem. Our previous work in this domain has shown that for a set of applications, one can efficiently identify data structures that signify application errors [2, 3]. Our goal in this task is to build on this work to identify other applications and analyze their error behavior.

Fault Throttling Techniques and Models

The first stage is to investigate techniques that are available to a designer for controlling the fault behavior of a processor or processor component. There is already a set of techniques in the literature that vary the "reliability" of a component. For example, reducing the supply voltage (i.e., voltage scaling) of a component increases its delay and thereby increases the probability that the timing constraint will not be met (i.e., a hardware fault will occur). In the future, we will analyze these tools and investigate their applicability in clumsy execution. For example, in our previous work, we have analyzed the applicability of overclocking for clumsy execution [4].

Clumsy Packet Processors

The main approach in clumsy packet processors is to push the architecture properties (i.e., fault throttling) to a point where the processor components start making mistakes. Doing this increases the performance and/or reduces the energy consumption according to the particular fault throttling technique. In network processors, many applications are known to be immune to errors in higher levels. However, we still need to investigate clumsy packet processors for different applications in order to compare different configurations. Although our previous work has introduced the notion of clumsy packet processors [1], we still have important research questions to tackle. First, we will work on analyzing existing architectures and measure the impact of certain hardware faults on the application behavior. We have already started working on a simulation framework that models representative processing elements. We will continue our work on this framework and include different architectural models. In addition, we will select a set of applications (for the selection of the applications, the PI is in the process of identifying DoE scientists, who work in related areas). These applications will be studied in detail for existing processing models and the impact of hardware faults on them will be measured.

Measuring the Impact on Network Infrastructure

Among the research tasks, this is the least studied and we will mostly concentrate on this aspect for the next year. Particularly, the goal in the short term is to modify existing simulators such as NS-2 to evaluate the impacts of clumsy routers. The PI will also identify DoE researchers who might be interested in solving the problems associated with the measurement of the impact.

DOE Collaborators and Impact to DOE Applications

Currently, the PI is not directly collaborating with DoE scientists. However, he has contacted Dr. Nageswara S. Rao at Oak Ridge National Laboratories for collaboration possibilities. Dr. Rao is one of the investigators of the DoE UltraScience Net project. The Experimental Ultra-Scale Network Research Testbed (UltraScience Net) aims to develop networks with unprecedented capabilities to support distributed large-scale science applications. Therefore, the goals of this project matches directly with the goals of the PI. Another possible collaborator is Steve Hurd from Center for Cyber Defenders (CCD) at Sandia National Labs (David Nguyen, one of the PI's advisees, is doing an internship under his supervision). In the coming months, the PI will continue identifying other DoE scientists, whose research goals overlap with the target of this project.

References

[1] A. Mallik and Memik, G. A Case for Clumsy Packet Processors. In Proc. of *International Symposium on Microarchitecture*. Portland, OR, Dec. 2004.

[2] A. Mallik, Wildrick, M.C., and Memik, G., *Application-Level Error Measurements for Network Processors.* To appear in IEICE Transactions on Information and Systems.

[3] A. Mallik, Wildrick, M.C., and Memik, G. Measuring Application Error Rates for Network Processors. In Proc. of *IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. Hiroshima, Japan, July 2004.

[4] G. Memik, Chowdhury, M., Mallik, A., and Ismail, Y. Engineering Over-Clocking: Reliability-Performance Trade-Offs for High-Performance Register Files. In Proc. of *IEEE/ACM International Conference on Dependable Systems and Networks (DSN)*. Yokohama, Japan, June - July 2005.

[5] ORNL. DOE UltraScience Net, Jan 2004, http://www.csm.ornl.gov/ultranet/.