This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2022-11083C

Sandia National Laboratories

Exceptional service in the national interest

Panel: Computing at Extreme Scales

Ron Brightwell, R&D Manager

Scalable System Software Department

ModSim

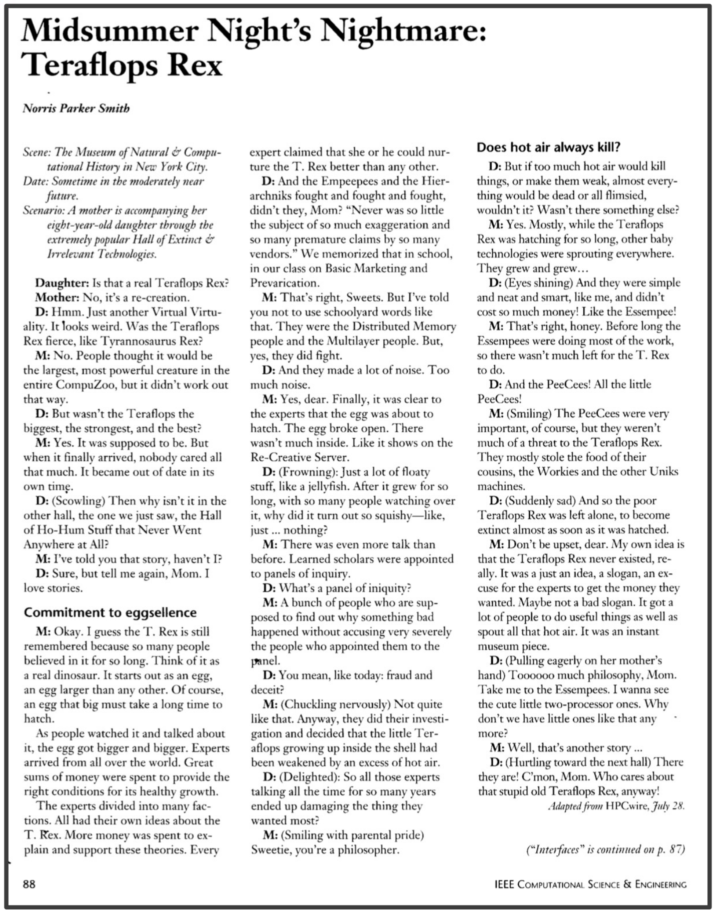
August 10-12, 2022

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

Is it real? Does it provide 1000X application performance of a Petasacle, and at the expense of how much power?

Is it real?



Midsummer Night's Nightmare: Teraflops Rex

Norris Parker Smith

Scene: The Museum of Natural & Computational History in New York City.
Date: Sometime in the moderately near future.
Scenario: A mother is accompanying her eight-year-old daughter through the extremely popular Hall of Extinct & Irrelevant Technologies.

Daughter: Is that a real Teraflops Rex?

Mother: No, it's a re-creation.

D: Hmm. Just another Virtual Virtuality. It looks weird. Was the Teraflops Rex fierce, like Tyrannosaurus Rex?

M: No. People thought it would be the largest, most powerful creature in the entire CompuZoo, but it didn't work out that way.

D: But wasn't the Teraflops the biggest, the strongest, and the best?

M: Yes. It was supposed to be. But when it finally arrived, nobody cared all that much. It became out of date in its own time.

D: (Scowling) Then why isn't it in the other hall, the one we just saw, the Hall of Ho-Hum Stuff that Never Went Anywhere at All?

M: I've told you that story, haven't I?

D: Sure, but tell me again, Mom. I love stories.

Commitment to eggsellence

M: Okay. I guess the T. Rex is still remembered because so many people believed in it for so long. Think of it as a real dinosaur. It starts out as an egg, an egg larger than any other. Of course, an egg that big must take a long time to hatch.

As people watched it and talked about it, the egg got bigger and bigger. Experts arrived from all over the world. Great sums of money were spent to provide the right conditions for its healthy growth.

The experts divided into many factions. All had their own ideas about the T. Rex. More money was spent to explain and support these theories. Every expert claimed that she or he could nurture the T. Rex better than any other.

D: And the Empeepees and the Hierarchniks fought and fought and fought, didn't they, Mom? "Never was so little the subject of so much exaggeration and so many premature claims by so many vendors." We memorized that in school, in our class on Basic Marketing and Prevarication.

M: That's right, Sweets. But I've told you not to use schoolyard words like that. They were the Distributed Memory people and the Multilayer people. But, yes, they did fight.

D: And they made a lot of noise. Too much noise.

M: Yes, dear. Finally, it was clear to the experts that the egg was about to hatch. The egg broke open. There wasn't much inside. Like it shows on the Re-Creative Server.

D: (Frowning): Just a lot of floaty stuff, like a jellyfish. After it grew for so long, with so many people watching over it, why did it turn out so squishy—like, just ... nothing?

M: There was even more talk than before. Learned scholars were appointed to panels of inquiry.

D: What's a panel of iniquity?

M: A bunch of people who are supposed to find out why something bad happened without accusing very severely the people who appointed them to the panel.

D: You mean, like today: fraud and deceit?

M: (Chuckling nervously) Not quite like that. Anyway, they did their investigation and decided that the little Teraflops growing up inside the shell had been weakened by an excess of hot air.

D: (Delighted): So all those experts talking all the time for so many years ended up damaging the thing they wanted most?

M: (Smiling with parental pride) Sweetie, you're a philosopher.

Does hot air always kill?

D: But if too much hot air would kill things, or make them weak, almost everything would be dead or all flimsied, wouldn't it? Wasn't there something else?

M: Yes. Mostly, while the Teraflops Rex was hatching for so long, other baby technologies were sprouting everywhere. They grew and grew…

D: (Eyes shining) And they were simple and neat and smart, like me, and didn't cost so much money! Like the Essempee!

M: That's right, honey. Before long the Essempees were doing most of the work, so there wasn't much left for the T. Rex to do.

D: And the PeeCees! All the little PeeCees!

M: (Smiling) The PeeCees were very important, of course, but they weren't much of a threat to the Teraflops Rex. They mostly stole the food of their cousins, the Workies and the other Uniks machines.

D: (Suddenly sad) And so the poor Teraflops Rex was left alone, to become extinct almost as soon as it was hatched.

M: Don't be upset, dear. My own idea is that the Teraflops Rex never existed, really. It was a just an idea, a slogan, an excuse for the experts to get the money they wanted. Maybe not a bad slogan. It got a lot of people to do useful things as well as spout all that hot air. It was an instant museum piece.

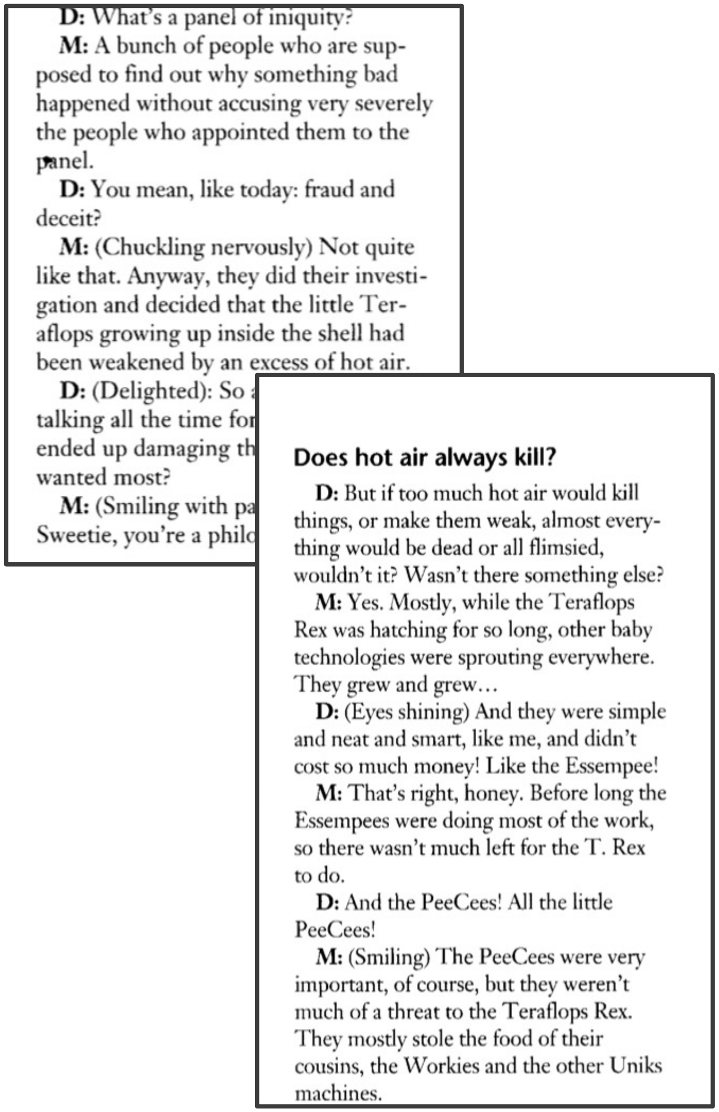
D: (Pulling eagerly on her mother's hand) Toooooo much philosophy, Mom. Take me to the Essempees. I wanna see the cute little two-processor ones. Why don't we have little ones like that any more?

M: Well, that's another story ...

D: (Hurtling toward the next hall) There they are! C'mon, Mom. Who cares about that stupid old Teraflops Rex, anyway!

Adapted from HPCwire, *July 28.*

("Interfaces" is continued on p. 87)



D: What's a panel of iniquity?

M: A bunch of people who are supposed to find out why something bad happened without accusing very severely the people who appointed them to the panel.

D: You mean, like today: fraud and deceit?

M: (Chuckling nervously) Not quite like that. Anyway, they did their investigation and decided that the little Teraflops growing up inside the shell had been weakened by an excess of hot air.

D: (Delighted): So a
talking all the time for
ended up damaging th
wanted most?

M: (Smiling with pa
Sweetie, you're a philo

Does hot air always kill?

D: But if too much hot air would kill things, or make them weak, almost everything would be dead or all flimsied, wouldn't it? Wasn't there something else?

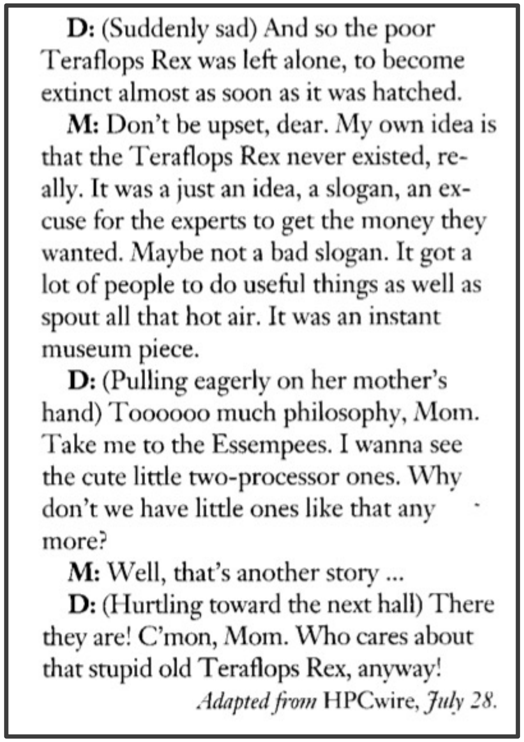
M: Yes. Mostly, while the Teraflops Rex was hatching for so long, other baby technologies were sprouting everywhere. They grew and grew…

D: (Eyes shining) And they were simple and neat and smart, like me, and didn't cost so much money! Like the Essempee!

M: That's right, honey. Before long the Essempees were doing most of the work, so there wasn't much left for the T. Rex to do.

D: And the PeeCees! All the little PeeCees!

M: (Smiling) The PeeCees were very important, of course, but they weren't much of a threat to the Teraflops Rex. They mostly stole the food of their cousins, the Workies and the other Uniks machines.



D: (Suddenly sad) And so the poor Teraflops Rex was left alone, to become extinct almost as soon as it was hatched.

M: Don't be upset, dear. My own idea is that the Teraflops Rex never existed, really. It was a just an idea, a slogan, an excuse for the experts to get the money they wanted. Maybe not a bad slogan. It got a lot of people to do useful things as well as spout all that hot air. It was an instant museum piece.

D: (Pulling eagerly on her mother's hand) Toooooo much philosophy, Mom. Take me to the Essempees. I wanna see the cute little two-processor ones. Why don't we have little ones like that any more?

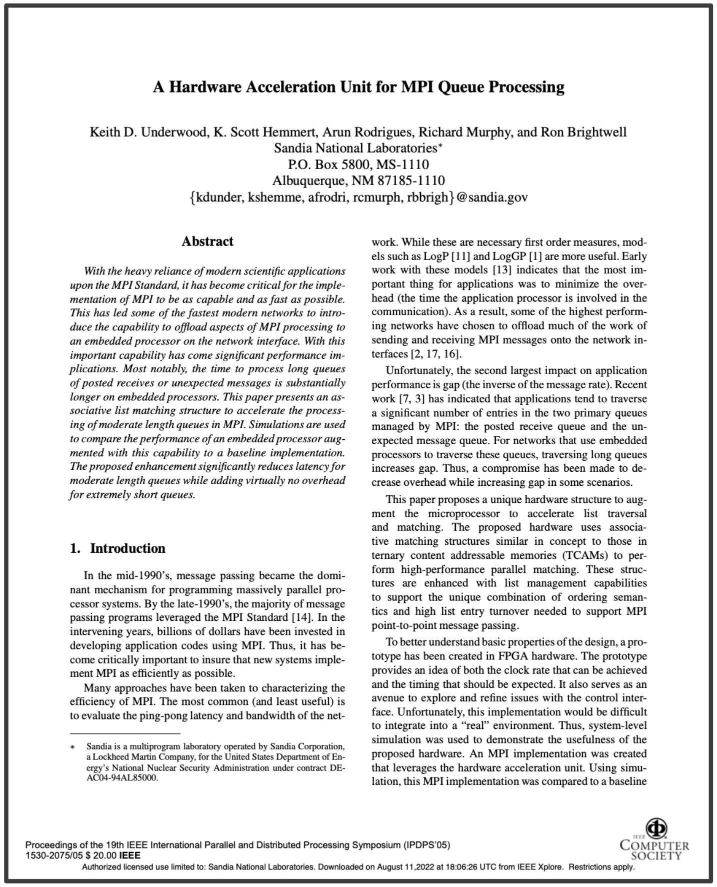
M: Well, that's another story ...

D: (Hurtling toward the next hall) There they are! C'mon, Mom. Who cares about that stupid old Teraflops Rex, anyway!

Adapted from HPCwire, *July 28.*

What role did modeling and simulation technologies play in successful deployment, did they identify the shortcomings, and were they accurate?

What role did modeling and simulation technologies play in successful deployment, did they identify the shortcomings, and were they accurate?



A Hardware Acceleration Unit for MPI Queue Processing

Keith D. Underwood, K. Scott Hemmert, Arun Rodrigues, Richard Murphy, and Ron Brightwell
Sandia National Laboratories*
P.O. Box 5800, MS-1110
Albuquerque, NM 87185-1110
{kdunder, kshemme, afrodri, rcmurph, rbbrigh}@sandia.gov

Abstract

With the heavy reliance of modern scientific applications upon the MPI Standard, it has become critical for the implementation of MPI to be as capable and as fast as possible. This has led some of the fastest modern networks to introduce the capability to offload aspects of MPI processing to an embedded processor on the network interface. With this important capability has come significant performance implications. Most notably, the time to process long queues of posted receives or unexpected messages is substantially longer on embedded processors. This paper presents an associative list matching structure to accelerate the processing of moderate length queues in MPI. Simulations are used to compare the performance of an embedded processor augmented with this capability to a baseline implementation. The proposed enhancement significantly reduces latency for moderate length queues while adding virtually no overhead for extremely short queues.

1. Introduction

In the mid-1990's, message passing became the dominant mechanism for programming massively parallel processor systems. By the late-1990's, the majority of message passing programs leveraged the MPI Standard [14]. In the intervening years, billions of dollars have been invested in developing application codes using MPI. Thus, it has become critically important to insure that new systems implement MPI as efficiently as possible.

Many approaches have been taken to characterizing the efficiency of MPI. The most common (and least useful) is to evaluate the ping-pong latency and bandwidth of the network. While these are necessary first order measures, models such as LogP [11] and LogGP [1] are more useful. Early work with these models [13] indicates that the most important thing for applications was to minimize the overhead (the time the application processor is involved in the communication). As a result, some of the highest performing networks have chosen to offload much of the work of sending and receiving MPI messages onto the network interfaces [2, 17, 16].

Unfortunately, the second largest impact on application performance is gap (the inverse of the message rate). Recent work [7, 3] has indicated that applications tend to traverse a significant number of entries in the two primary queues managed by MPI: the posted receive queue and the unexpected message queue. For networks that use embedded processors to traverse these queues, traversing long queues increases gap. Thus, a compromise has been made to decrease overhead while increasing gap in some scenarios.

This paper proposes a unique hardware structure to augment the microprocessor to accelerate list traversal and matching. The proposed hardware uses associative matching structures similar in concept to those in ternary content addressable memories (TCAMs) to perform high-performance parallel matching. These structures are enhanced with list management capabilities to support the unique combination of ordering semantics and high list entry turnover needed to support MPI point-to-point message passing.

To better understand basic properties of the design, a prototype has been created in FPGA hardware. The prototype provides an idea of both the clock rate that can be achieved and the timing that should be expected. It also serves as an avenue to explore and refine issues with the control interface. Unfortunately, this implementation would be difficult to integrate into a "real" environment. Thus, system-level simulation was used to demonstrate the usefulness of the proposed hardware. An MPI implementation was created that leverages the hardware acceleration unit. Using simulation, this MPI implementation was compared to a baseline

[*] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS'05)
1530-2075/05 $ 20.00 **IEEE**

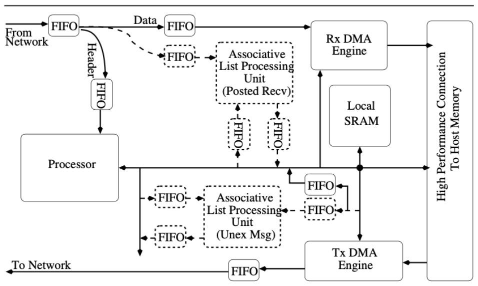

Data
FIFO
From Network
FIFO
Header
FIFO
FIFO
Associative List Processing Unit (Posted Recv)
FIFO
FIFO
Rx DMA Engine
Local SRAM
High Performance Connection To Host Memory
Processor
FIFO
FIFO
Associative List Processing Unit (Unex Msg)
FIFO
FIFO
Tx DMA Engine
To Network
FIFO

Figure 1. The proposed NIC architecture (new features shown in dashed lines)

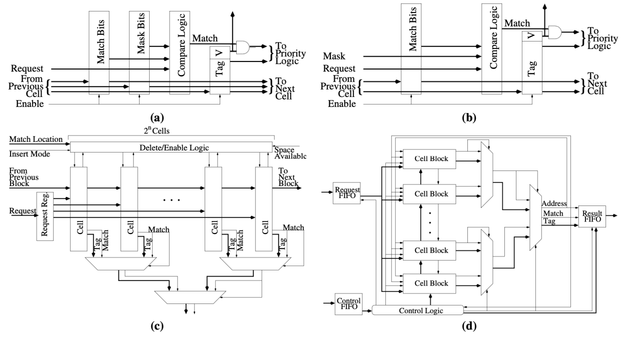
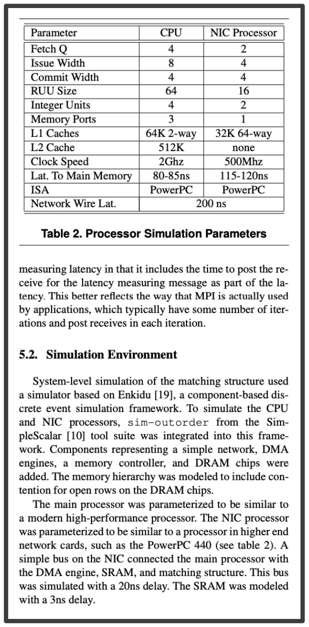


Figure 2. (a) A cell with a single match unit for the posted receive queue; (b) The corresponding cell a for the unexpected message queue; (c) A block of cells; (d) The associative match engine



Parameter	CPU	NIC Processor
Fetch Q	4	2
Issue Width	8	4
Commit Width	4	4
RUU Size	64	16
Integer Units	4	2
Memory Ports	3	1
L1 Caches	64K 2-way	32K 64-way
L2 Cache	512K	none
Clock Speed	2Ghz	500Mhz
Lat. To Main Memory	80-85ns	115-120ns
ISA	PowerPC	PowerPC
Network Wire Lat.	200 ns	

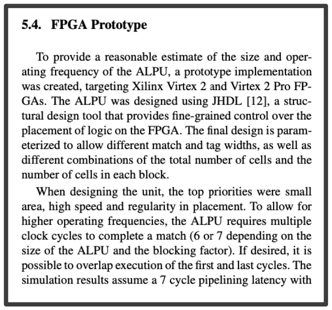
Table 2. Processor Simulation Parameters

measuring latency in that it includes the time to post the receive for the latency measuring message as part of the latency. This better reflects the way that MPI is actually used by applications, which typically have some number of iterations and post receives in each iteration.

5.2. Simulation Environment

System-level simulation of the matching structure used a simulator based on Enkidu [19], a component-based discrete event simulation framework. To simulate the CPU and NIC processors, `sim-outorder` from the SimpleScalar [10] tool suite was integrated into this framework. Components representing a simple network, DMA engines, a memory controller, and DRAM chips were added. The memory hierarchy was modeled to include contention for open rows on the DRAM chips.

The main processor was parameterized to be similar to a modern high-performance processor. The NIC processor was parameterized to be similar to a processor in higher end network cards, such as the PowerPC 440 (see table 2). A simple bus on the NIC connected the main processor with the DMA engine, SRAM, and matching structure. This bus was simulated with a 20ns delay. The SRAM was modeled with a 3ns delay.



5.4. FPGA Prototype

To provide a reasonable estimate of the size and operating frequency of the ALPU, a prototype implementation was created, targeting Xilinx Virtex 2 and Virtex 2 Pro FPGAs. The ALPU was designed using JHDL [12], a structural design tool that provides fine-grained control over the placement of logic on the FPGA. The final design is parameterized to allow different match and tag widths, as well as different combinations of the total number of cells and the number of cells in each block.

When designing the unit, the top priorities were small area, high speed and regularity in placement. To allow for higher operating frequencies, the ALPU requires multiple clock cycles to complete a match (6 or 7 depending on the size of the ALPU and the blocking factor). If desired, it is possible to overlap execution of the first and last cycles. The simulation results assume a 7 cycle pipelining latency with