Inquiry: The University of Arkansas Undergraduate Research Journal

Volume 1 Article 16

Fall 2000

Design, Implementation, and Evaluation of Virtual Interface Architecture for Power PC Machines

Ben McKenzie
University of Arkansas, Fayetteville

DESIGN, IMPLEMENTATION, AND EVALUATIONOF VIRTUAL INTERFACE ARCHITECTUREFOR POWER PC MACHINES

by Ben McKenzie
Department of Computer Science and Computer Engineering

Faculty Mentor: Amy W. Apon
Department of Computer Science and Computer Engineering

Ben McKenzie and Amy Apon

Abstract.

The Virtual Interface Architecture (VIA) standard is a low-latency protocol that was designed for use in high-performance networks. VIA improves performance by reducing overhead in messaging. This research has two components. The first part of this research project is the development of a new tool for measuring the performance of a VIA implementation and comparing it to the more traditional high-overhead protocols used on the Internet. The development of the tool represents a significant contribution in and of itself, since the tool has been put into the public domain and will likely become useful by Linux users, both for measuring VIA networks, and as one of the first example codes available for learning how to write programs that use VIA. The new tool has exposed some interesting performance issues of VIA as the number of messages increases that are currently being examined. The second component of this research is the definition and development of appropriate interfaces from the network to the lowest level services in Linux on the Power PC platform, and the testing and evaluation of these functions. The research approach was to port a freely available implementation of VIA that runs on a Pentium platform to the Power PC platform. The architectural differences of the two platforms have raised a number of design and configuration issues that have been investigated and solved.

Introduction

The Virtual Interface Architecture (VIA) standard is a low-latency protocol designed for use in high-performance networks. VIA improves performance by reducing overhead in messaging. [1] Several tools exist for measuring performance but are unreliable and do not provide accurate evaluation of communication times. [3]

This research investigates and improves the performance of cluster computing. There are two major goals to this project. The first goal is to define, develop, test, and evaluate appropriate VIA interfaces from the network to the lowest level services in Linux on the PowerPC platform. The second goal of this research project is the development, testing, and evaluation of a new tool for measuring the performance of a VIA implementation, and comparison to more traditional high-overhead protocols used on the Internet. Finally, the research presents ideas of future work which will come from this project.

Background

The background of this project deals with two areas of research. These areas are the history and origins of VIA and techniques of network benchmarking.

VIA is a protocol for system area networks designed by a consortium of industry and researchers including Intel, Compaq, and Microsoft. VIA specifies a low latency zero-copy messaging system designed for high speed, reliable network architectures on system area networks. System area networks, or SAN's are networks in which the nodes, or individual computers, are physically close to each other and are physically and virtually isolated. Typical SAN's are devoted to a single application and have little to no interfering traffic. SAN's tend to be much more reliable and faster than a local area network or LAN. LAN's are commonly used for larger networks such as a campus or building networks. A SAN is the most common type of network for cluster computing. Cluster computing is using a group of computers collectively to work togetherand solve a problem.

VIA is designed to take advantage of sophisticated hardware platforms that run on a SAN. The native implementation of VIA supports Direct Memory Access (DMA) engines that exist on network cards. A DMA engine allows data to be transferred from the memory of the computer to the network card and then to the network. A DMA engine also allows incoming data to be transferred directly to the computer, or host's memory.

With the network card doing the memory transfers, the host computer is not required to perform costly memory to memory copies.

Locking memory is an operation that interfaces to hardware at a very low level and is not usually performed by user applications. For DMA operations in VIA to work correctly, memory must be "locked". [5] This means that the data will not move to a different physical location during the life of the program. This results in more memory being used during the execution of a program than would otherwise be necessary. Optimally, some of the extra memory could be used for computation. The reason that the memory must be locked is because a physical address of memory is passed to the DMA engine. The DMA engine looks at that physical memory and retrieves the correct data from that physical location.

VIA is designed as a connection-oriented point to point protocol. In this protocol, each Virtual Interface (VI) must connect to another VI and be completely connected before sending any messages. To send to more than one VI from a particular process, multiple connections must be set up within that process.

The industry support of VIA is helpful in assuring that the protocol won't fade into obscurity. Many researchers and industry representatives believe that VIA represents the future of cluster computing. Oracle and Informix both have versions of their database products that rely heavily on VIA. Many high end networking hardware vendors are starting to incorporate support for VIA in their products. There are also several research institutes working on VIA, creating products such as M-VIA and Berkeley VIA. There are also versions of the Message Passing Interface that utilize VIA for their messaging, such as MPI-PRO, [4] MVICH, [1] and MARK. [2] Message Passing Interface is one of the most commonly used clustering packages.

Existing tools for measuring VIA and TCP are TTCP and vpingpong. TTCP is a benchmarking tool for UDP and TCP performance between two computers [3]. VPingPong is a tool included with M-VIA that is also used to measure the round trip time between two VIA enabled computers [1].

Network Benchmarking

Both VPingPong and TTCP use system calls to access the current time. These calls result in context switches which affect the measurements. They are also two separate programs compiled with different switches. Further accuracy is affected by these programs having different program sizes, different memory allocation approaches, and having different numbers of instructions.

TTCP and VPingPong were used as a baseline for a new testing tool created for this project known as TTCP-VIA. This tool allows "ping pong" testing between two different machines to get a round trip time for a single message. A ping pong test is a good measure of real network speed as it uses a send and receive on both sides. In a ping pong test, host 1 sends to host 2, host 2 receives, host 2 sends to host 1, and finally host 1 receives. The TTCP-VIA tool uses the processor's cycle counter to obtain measurements with microsecond accuracy, a minimum of overhead, and no context switch.

Porting VIA to Power PC

The second goal of the project was to port a version of VIA originally developed for the Intel architecture, to the PowerPC architecture. The baseline chosen for our version of VIA is M-VIA, a modular implementation of VIA available for Intel Linux machines made by the National Energy Research Scientific Computing Center. [1] This implementation was chosen as it allowed for ordinary Fast Ethernet cards to transmit and receive information over VIA.

The problem of not running successfully is greatly heightened by the fact that this code is running in kernel space with direct access to memory and has a very high priority. So, great care must be taken to write the code correctly, or it can and will crash the operating system. The code must also be tested extensively.

There are many issues with porting the M-VIA software to PowerPC machines. One of the critical issues faced was none of the Fast Ethernet drivers included with M-VIA work with PowerPC machines. Second, there is assembly code included in the M-VIA code included to optimize performance. Assembly code is inherently not portable since all assembly code is machine specific. The final issue confronted is that the Pentium is a little endian machine and the PowerPC is a big endian machine. In a big endian machine the leftmost bytes are the most significant. In a little endian architecture, the rightmost bytes are the most significant. [6]

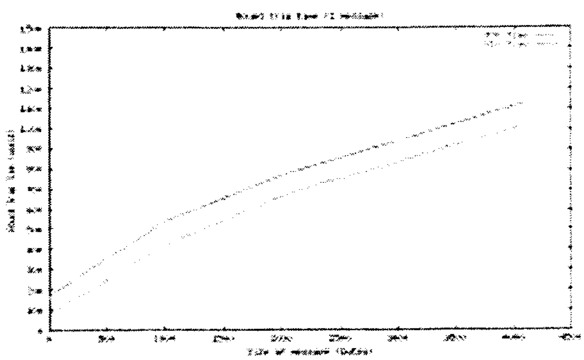
The assembler code contained in the M-VIA code was not necessary for our implementation on the PowerPC. This code is used for what are known to the M-VIA authors as fast operations. Basically, this just allows optimization of the various VIA operations. This code can be bypassed to allow a working but not necessarily entirely optimal version of M-VIA for PowerPC's.

The way that the endian problem was fixed was to typecast the constant values. At compile time, the strings are converted from constant strings to real values stored in memory. Real data values automatically change to the correct architecture form.

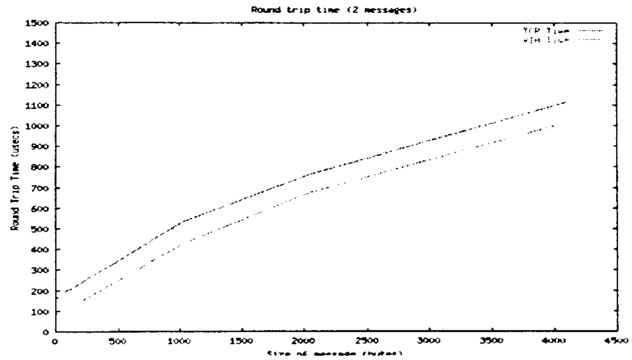
The problem of not having an Ethernet driver is made more simple given the nature of the Linux operating system and the fact that the M-VIA project is an open source project. There are several examples of similar network cards available and these were used to create a new driver. In addition, an existing open source Ethernet driver for this particular card was used as a starting point.

Measurement Results

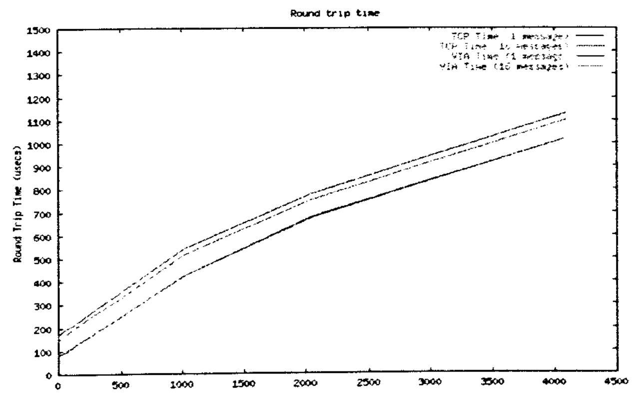
Using the TTCP-VIA tool, we have been able to obtain preliminary performance results on our existing cluster with Intel Linux based machines and have compared the performance of emulated M-VIA to TCP.



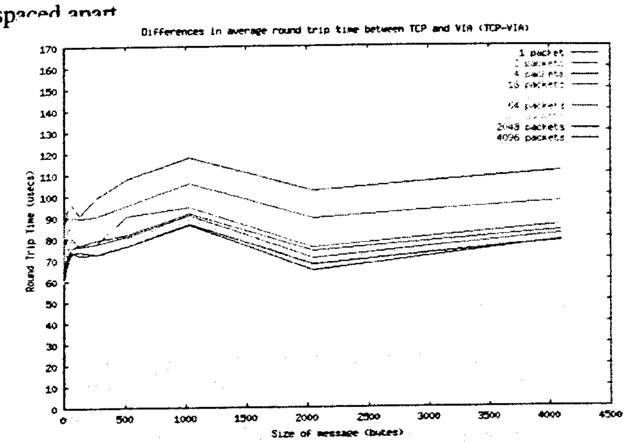
When looking at the data from graph one, the round trip time for a 1 byte message for VIA is 87 microseconds, and for TCP it is 178 microseconds. So, the time to send a 1 byte message is a little less than half the time to send a TCP message for very small messages. This remains true until about 512 byte messages, where the VIA time is 253 microseconds and the TCP time is 263 microseconds. At the largest measurement of 8192 bytes, the VIA time is 1707 microseconds and the TCP time is 2117 microseconds.



Graph two shows very similar results to graph one. All graphs represent average times obtained by dividing the results by the number of messages, so the numbers are more similar to each other. The difference in times for a 4 byte message is 87.25 microseconds/byte for VIA and 156 microseconds / byte for TCP. The difference between the two is a little bit lower than the times for a 1 byte messages, but not much. As you can see the VIA messages are still about twice as fast as TCP messages for very small sizes and for large sizes are still better.



In graph three, TCP's performance inconsistency is shown. The graph shows that the VIA times are very close together, appearing to be one line. The TCP lines, on the other hand, are spaced apart.



Graph four shows the differences in average round trip times between TCP and VIA times. This graph subtracts the TCP time from the VIA time. Notice that the performance difference is greater when sending one message than when sending 4096 messages. This once again shows that TCP performance is inconsistent, and actually changes when sending large numbers. The maximum in the graph at about 100 bytes is due to the change in TCP from using one internal message buffer to using two. The change at around 1200 is due to the Ethernet frame size (1500 bytes).

Conclusion and Future Work

As is shown in the experimental results, there is a lot to learn about network performance measurements. The technique of repeateatedly sending messages and dividing by the number of messages sent can result in biased results. Techniques this thesis presented can help to fully evaluate network performance.

With a version of M-VIA created for PowerPC, many new avenues are open for using PowerPC's for clustering. There are some types of hardware which will only work on PowerPC machines. Plus, this will broaden the influence of VIA, and expand its usefulness.

The results of this thesis show many areas for future work. First, a version of VIA will be developed to run on AmpNet, a gigabit speed Fibre Channel based network created by Belobox Networks. [7] Also, areas of VIA have been identified which could be extended to achieve further functionality such as multicasting and broadcasting of messages. Another future goal is to implement a version of MPI that runs over VIA [2].

Bibliography

[1] Berkely, M-VIA Frequently Asked Questions 1999

[2] Hsiang Ann Chen. Master thesis proposal: On the Optimization of the Implementation of MPI

[3] Chesapeake Online, Chesapeake Computer Consultants, TTCP for Timing TCP. http://www.ccci.com/tools/ttcp/index.html

[4] Rossen Dimitrov and Anthony Skjelum. An Efficient MPI Implementation for Virtual Interface (VI) Architecture-Enabled Cluster Computing. Available from Giganet, http://www.giganet.com, 1999

[5] Intel. Virtual Interface Architecture *Specification. Intel, September 1998.*

[6] *Andrew S. Tanenbaum.* Structural Computer Organization. *Prentice Hall, 1990.*

[7] Larry Wilbur. Advance Multiprocessor Network: A New Networking Technology

Faculty Comments:

Mr. McKenzie's faculty mentor, **Dr. Amy Apon,** says about his project:

> Ben's research examines how to increase the performance of applications in a cluster by optimizing the messaging software. A cluster is a computing system that consists of a collection of interconnected whole computers. The messaging software that Ben is working with conforms to a new standard, the Virtual Interface Architecture (VIA). The VIA standard was published in December, 1997. VIA reduces overhead by interfacing with the operating system at a very low level.
>
> Ben's research project is divided into two components. For the first part of his research project, Ben has developed a new tool for measuring the performance of a VIA implementation and comparing it to the more traditional TCP/IP protocol. Prior to Ben's work, a tool did not exist that fairly compared VIA and TCP/IP. The development of the tool represents a significant contribution in and of itself, since the tool has been put into the public domain and will likely become useful by Linux users, both for measuring VIA networks, and as one of the first example codes available for learning how to write programs that use VIA. Ben's tool, ttcp-via, has exposed some interesting performance issues of VIA as the number of messages increases that we are now currently examining as an extension of his thesis. The second component of Ben's work is to define and develop appropriate interfaces from the network to the lowest level services in Linux, and to test and evaluate these functions. Specifically, Ben is taking a freely available implementation of VIA that runs on a Pentium platform, M-VIA and porting it to a PowerPC platform. The architectural differences of the two platforms have raised a number of design and configuration issues that Ben is having to investigate and solve.

Dr. David Andrews, Head of the Computer Science and Computer engineering Department, comments:

> Mr. McKenzie is one of the Computer Science and Computer Engineering departments brightest students who is at the beginning of a very promising professional career as a computer scientist. This effort is ideal in fostering Mr. McKenzie's growth by providing him the opportunity to enhance his traditional academic classroom requirements with in depth self directed research in support of communications protocols and implementation for cluster computing. Over the last decade, clusters, or networked collections of computers, have come to be used for many high-performance parallel applications, and have helped to redefine the concept of supercomputing. Cluster computing is particularly relevant for high performance applications used by industries in Arkansas. For example, Acxiom Corporation is investigating the use of clusters for their high performance database queries. Wal-Mart Corporation is one of the largest users of data mining in the world, and cluster computing is an ideal approach for speeding up the processing of the enormous amount of data that is collected by Wal-Mart each day.

In addition to allowing applications to execute faster, cluster computing is also being used to build highly available systems. In a highly available system, when a component fails, the workload is redistributed to other components allowing all applications to run without interruption. In the particular case of running a database on a cluster, the system will not lose or damage data when a device fails. For these reasons, virtually all database vendors now target a clusterbased product for systems that require high availability.

Mr. McKenzie has established an impressive academic record within the department, achieving a 3.8 GPA. Mr. McKenzie has the necessary background for this work having enrolled in Dr. Apon's Concurrent Computing class and has traveled to California to learn and work with the company providing the experimental hardware and software that is used in this project. Mr. McKenzie's capabilities can also be seen by the company's' current interest in recruiting him to work after he has completed his education here at the University of Arkansas. capabilities.