Towards Terabit/s Systems: Performance Evaluation of Multi-Rail Systems

Venkatram Vishwanath*, Takashi Shimizu[+], Makoto Takizawa[+], Kazuaki Obana[+], Jason Leigh*

*Electronic Visualization Laboratory (EVL),
University of Illinois at Chicago, Chicago, U.S.A.
venkat@evl.uic.edu

[+]NTT Network Innovation Laboratory,
Yokosuka, Japan
t-shimizu@ieee.org

***Abstract*- We present a novel *multi-rail* approach that is necessary in order for future E-Science applications to effectively exploit Terabit/s networks. The multi-rail approach consists of creating parallel "rails" through every aspect of an end-system: from processing on the multiple cores, generation of multiple application data flows, and streaming over multiple-lanes, multi-wavelength NICs connected via a parallel interconnect. In this paper, we evaluate the end-systems parameters that impact the efficiency of multi-rail systems such as interrupt, memory, thread, and core affinities. These evaluations are tested on the ability of single cluster nodes to achieve TCP and UDP throughput at 10Gbps and 20Gbps rates. We analyze the additive effects of the parameters - a key property for achieving scalable performance towards Terabits/s. Thread and Interrupt affinity together were found to have an additive effect and play a critical role in achieving a throughput of 20Gbps.**

I. Introduction

Interactive exploration of multi-terabyte datasets has been identified as a critical enabler for scientists to glean new insights in a variety of disciplines, such as biomedical imaging, geosciences and high-energy physics [1]. Practically, these large-scale datasets must flow among a Grid of instruments, physical storage devices, visualization displays, and computational clusters. These applications are typically characterized by a myriad parallel flows among the interconnected resources. With the data and data flows growing exponentially, a terabit-LAN interconnecting these resources will be extremely critical. A design for such a Terabit Interconnection has been investigated [2] and the prototype demonstrated with a Multi-party collaboration and data analysis application [3]. Additionally, with the introduction of Teraflop chips from Intel and AMD, with a roadmap for usage in desktop and server environments, one could expect these systems used for such advanced e-science applications in the near future. However, novel techniques are required to enable applications to fully utilize the potential of systems connected by hundreds of gigabits/s to terabit/s of bandwidth.

In this paper, we present multi-rail design as a promising approach towards achieving Terabit/s performance for advanced E-Science applications. To realize a Terabit/s system from Teraflop systems, exploitation of parallelism in every aspect of systems is essential: from processing on the multiple cores, generation of multiple application data flows, and streaming over multiple-lanes, multi-wavelength NICs connected via a Parallel interconnect, which we call a Multi-Rail system design. We investigate the system characteristics that would play a critical role in such future Terabit/s systems. Specifically, we investigate system effects such as Thread Affinity, Interrupt Affinity, Memory Affinity, and Core Affinity on the throughput of network-intensive applications at line rates of 10Gbps and 20Gbps. We show that, at these rates, Thread affinity plays a critical role in achieving line rate performance. We demonstrate that Thread and Interrupt affinity together and Thread, Interrupt and Memory affinity together, have an additive effect on network intensive workload, for both TCP and UDP streams, and enable 20Gbps line rate performance. Memory affinity plays a vital role for large payloads. Identifying the effects of these systems parameters in order to achieve additive performance, will aid in scaling the performance towards Terabit/s.

The remaining part of the paper is organized as follows. In section II, we provide the relevant background information for our work. We present and elucidate the multi-rail design for Terabit/s in section III. Detailed experimental results are presented in section IV and discussed in section V. We describe the related work in section VI. We conclude and discuss our future directions in section VII.

II. Background

In this section, we provide relevant background information on current processor architectures. We briefly discuss bus-based architectures from Intel, e.g. Xeon, and, direct-connect-based architectures from AMD, e.g. Opteron.

A quad processor, bus-based architecture, is shown in Figure 1. In this architecture, a shared front-side bus connects the multiple processors to the North-bridge. The front-side bus can only be used by a single processor at a time. The architecture consists of a single shared memory controller located on the North-Bridge. The various memory banks are connected to the North-bridge by the memory bus and the main memory access latency for every processor is identical. The architecture consists of a shared PCI-express Bridge, located on the North Bridge, to which all the PCI-express devices are connected. This architecture is also referred to as the Symmetric Multiprocessing (SMP) based architecture.

In the quad processor, direct-connect AMD architecture, shown in Figure 2, neighboring processors are directly connected by a point-to-point hyper-transport bus. This architecture features a distributed Memory controller architecture, where each processor has an on-chip integrated memory controller. Each processor has a PCI-express Bridge to which PCI-express devices are connected. Thus, this architecture features a distributed PCI-express Bridge where a PCI-express based IO device is physically bound to a particular processor. Each processor has access a local memory bank connected via a hyper-transport bus. A processor can access the memory banks of the other processors, though with a

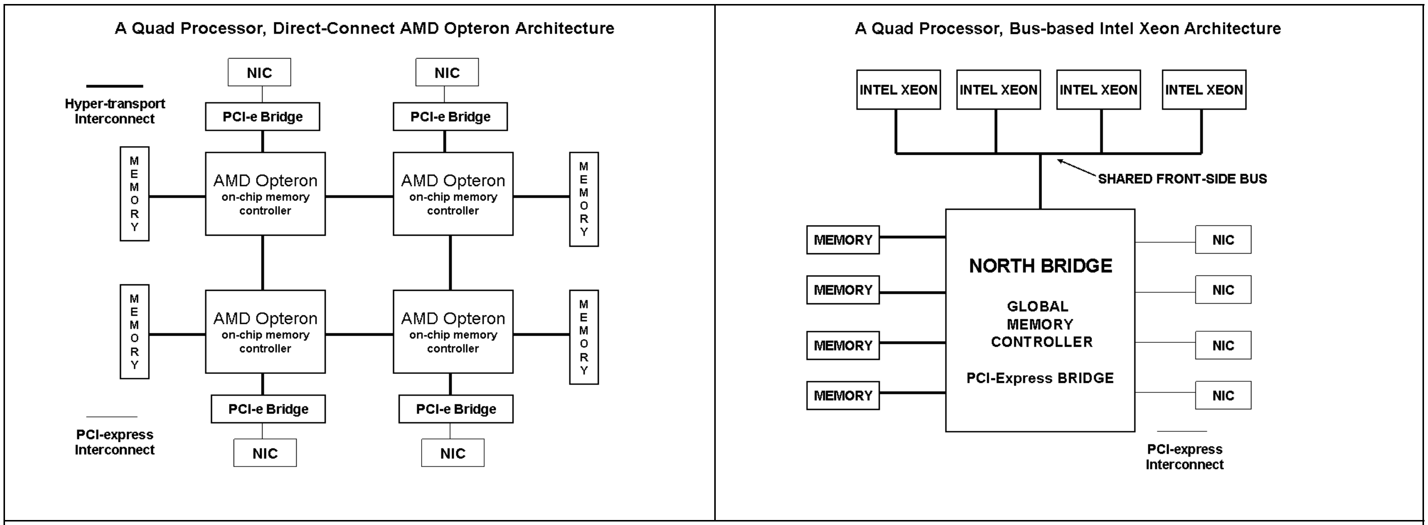
A Quad Processor, Direct-Connect AMD Opteron Architecture
Hyper-transport Interconnect
NIC
PCI-e Bridge
MEMORY
AMD Opteron on-chip memory controller
PCI-express Interconnect
A Quad Processor, Bus-based Intel Xeon Architecture
INTEL XEON
SHARED FRONT-SIDE BUS
MEMORY
NORTH BRIDGE
GLOBAL MEMORY CONTROLLER
PCI-Express BRIDGE
NIC
PCI-express Interconnect

Figure 1: Comparison of a Direct-Connect Based AMD Opteron Architecture and a Bus-Based Intel XEON Architecture

higher access latency as compared to its local memory bank, via the hyper-transport bus. This architecture represents a Non-Uniform Memory Access (NUMA) configuration.

III. Multi-Rail Approach

Teraflop systems from Intel and AMD are characterized by their multi-core parallel architectures. Additionally, multiple IO devices are connected to these Multi-core architectures via a multi-lane parallel interconnect technology such as PCI-express. The 100Gbps Ethernet standard recommends an end-system NIC design to be composed of a 10 x 10Gbps multi-rail NIC, i.e. a parallel NIC. To realize a Terabit/s system from such multi-core Teraflop systems, exploitation of parallelism in every aspect of the system is essential: from processing on the multiple cores, generation of multiple application data flows, and streaming over multiple-lanes, multi-wavelength NICs connected with a Parallel interconnect, which we call a Multi-Rail system design.

We define a *multi-rail* system as one, where, each rail consists of a processor core connected to a lane on a NIC via a dedicated interconnect. The NIC is connected to a Terabit Interconnect such as in [2]. The multi-rail approach can be expanded to exploit parallelism in disk and other subsystems. A multi-rail concept can be easily realized with the help of currently available processor architecture. In case of a typical Multi-core, Multi-lane Nic based systems, as shown in Figure 2, one can partition the above into a four rail system. Each rail consists of a core connected to a NIC for Network intensive workloads.

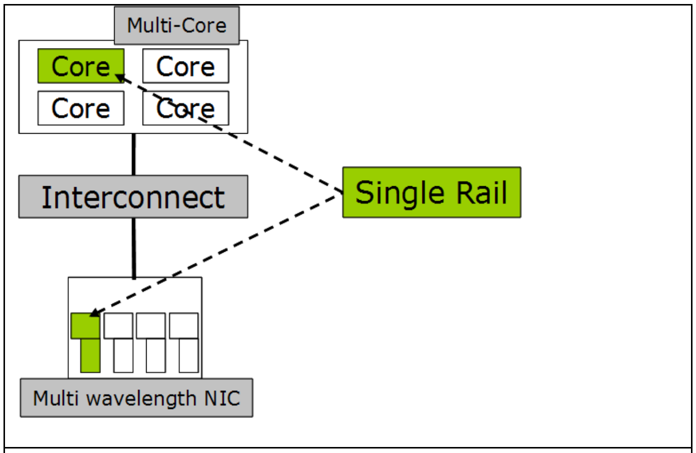

Figure 2: Multi-rail approach for terabit/s system

If **T** is the achievable performance of a single rail, In an **N**-rail system, the expected performance would be:
N x T x ∂, where ∂ is the parallel efficiency.
In an ideal parallel system, $\partial \rightarrow$ **1,** and such a system exhibits additive performance.

We envision future Teraflops chip architecture to have a multi-rail sub-system, as part of it system architecture, for scaling performance towards Terabit/s, for network intensive applications. We investigate the system characteristics that would play a critical role in such future Terabit/s systems. Specifically, we investigate system effects such as Thread Affinity, Interrupt Affinity, Memory Affinity, and Core Affinity on the throughput of network-intensive applications. Identifying the effects of these systems parameters towards achieving additive performance ($\partial \rightarrow$ **1)**, will aid in scaling performance towards Terabit/s.

IV. Experiments and Results

In this section, we first discuss the experimental testbed and methodology and then analyze the results in-depth.

A. *Experimental Setup*

The experimental testbed consisted of the following systems:

- Two Dual-Core, Dual-Processor AMD 2.6 GHz Opteron TYAN 2895 systems with 4GB RAM and 2 PCI-express 16X slots. The two machines were connected back-to-back with two 10G Myrinet NICs.
- Two Dual-Core, Dual-Processor 3.0Ghz Intel Xeon IBM x3500 systems connected back-to-back with two 10G Myrinet NICs.

The Linux kernel version used was 2.6.18 with MSI enabled. “nuttcp” 5.5.4 [9] was used as the Network Intensive Workload. The MTU used for the experiments was 9000bytes.

B. *Experimental Methodology*

We evaluate the effects of the following system parameters on network intensive workloads.

1. Interrupt Affinity (IA)

In Linux, the interrupts are, by default, processed by the most lightly loaded processor. The Interrupt processing by a particular processor or core, for an IO device, can be set by an appropriate mask in the /proc/interrupts configuration file [7]. We consider a system to be Interrupt Affine if the Interrupt processing is done by the processor to which the interrupt is physically bound. In other words, Interrupt affine systems are one where-in the interrupt processing does not generate Inter Processor Interrupt (IPI) Messages. One can set the interrupt affinity for a NUMA-based architecture. In an SMP–based system, the interrupts are physically bound to the North-Bridge. We consider an SMP-based system to be Interrupt affine.

2. Thread Affinity (TA)

Threads are scheduled, in Linux, on the most lightly loaded processor. The Linux scheduler periodically computes the load on each processor and tries to schedule threads in order to balance the load on all the processors. One can override the default scheduling policy and bind a thread to a particular processor via the sched_setaffinity() system call. We consider a system to be Thread affine, if the network application thread, in this case the *nuttcp* thread, is bound to the processor where the Interrupt processing of the network traffic occurs. Thread affinity can be achieved for both NUMA-based and SMP-based systems.

3. Memory Affinity (MA)

We refer to a system as Memory affine system as one where the memory used by an application thread is allocated on the memory bank with the lowest access latency. In case of NUMA-based systems, memory allocation on the local memory bank is considered to be memory affine. In Linux, Memory affinity can be configured via the numactl system call [8]. SMP-based systems have uniform memory latency and are considered to be Memory Affine systems.

We exhaustively study all 8 possible combinations of the above 3 system parameters on network intensive workloads. In the experiments, throughput is the maximum attainable data rate without any packet loss. The goal of the experiments was to study the additive properties of the system parameters and their combinations as we scale the systems by increasing the number of network cards and processors i.e. rails.

C. *Effects of System Parameters on Throughput of a Single UDP Stream*

We evaluate the effects of the system parameters on the achievable throughput of a single UDP stream. In Figure 3, the achievable UDP throughput in Mbps is plotted for various combinations of system parameters. With the default setting of the Linux kernel, one can achieve 90% of the 10Gbps line rate. Affinity helps in improving the throughput. Thread Affinity results in a 10% improvement of the throughput while Interrupt Affinity achieves 8% improvement over the default setting. This indicates that the Interrupt forwarding over the Hyper-transport links, from the processor where the interrupt physically occurs to the processor where the interrupt is handled, scales at 10Gbps. Thread affinity plays a crucial role in achieving close to line rate performance. A combination of the system parameters achieves close to line rate performance. A fully affine (IATAMA) NUMA-based system achieve close to line rate performance of 9.91Gbps. A tuned SMP system with IATAMA, achieves a throughput 230Mbps lower than a tuned NUMA-based system.

D. *Additive Effects of System Parameters on Throughput of a Two Concurrent UDP Stream*

Additive effects of the affinities are desirable as they provide an estimate of the scalability of systems as we strive towards Terabit/s. In Figure 4, twice the throughput achieved by a single UDP stream (Figure 3), in Mbps is shown, via the dashed bars. The achievable UDP throughput in Mbps for two concurrent UDP streams, for various combinations of system parameters, is also plotted. A system parameter or its combination is considered to be additive if the throughput of

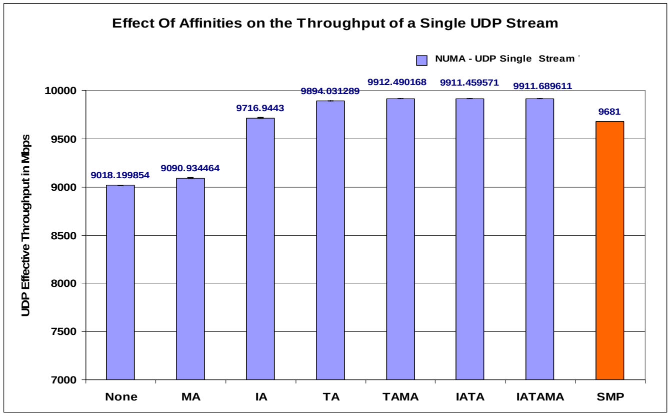

Figure 3: Effect of Affinities on the Throughput of a single UDP Streams

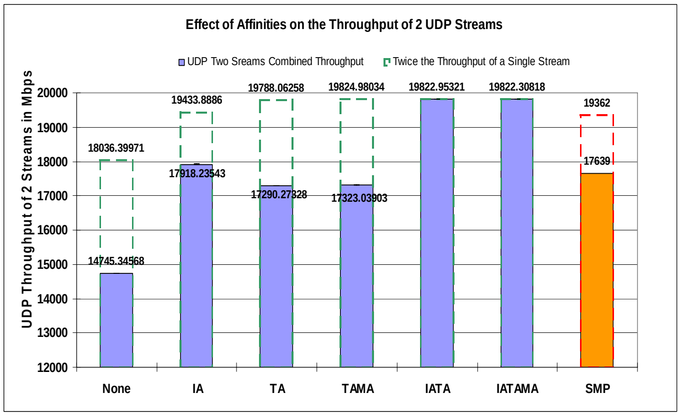

Figure 4: Effect of Affinities on Throughput of two concurrent UDP Streams

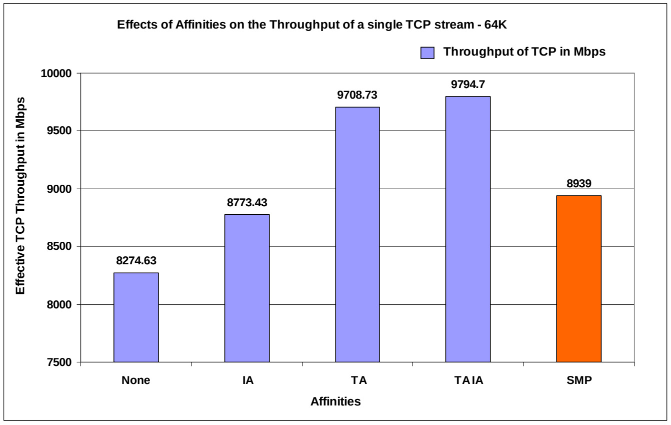

Figure 5: Effect of Affinities on the Throughput of a single TCP Stream

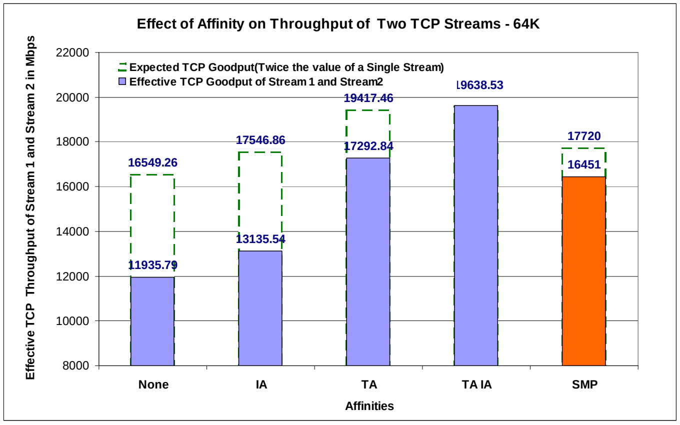

Figure 6: Effect of Affinities on the Throughput of two concurrent TCP Streams

two streams is equal to twice the throughput of a single stream. With the default setting of the Linux kernel, one can achieve 73% of the 20Gbps line rate. Thread Affinity has a lower additive effect than interrupt affinity for two concurrent UDP streams. This is due to the fact that the overhead associated with forwarding the interrupt processing, in case of a Thread affine system, is higher than the protocol processing and cache misses, in case of Interrupt affinity, associated with the network thread for UDP, as UDP has a small protocol processing footprint. Interrupt and Thread affinity (IATA) together achieve a line rate of 20Gbps and thus, demonstrate an additive effect. The combination of Thread, Interrupt and Memory affinity (IATAMA) also demonstrate an additive effect.

A tuned SMP system is unable to achieve additive performance for 2 concurrent UDP streams. It also sustains a throughput 10% lower than a tuned (IATAMA) NUMA-based system. This is primarily due to the shared single Front side-bus and the shared North-Bridge design on the SMP-based system.

E. Effects of System Parameters on Throughput of a Single TCP Stream

We evaluate the effects of the system parameters on the achievable throughput of a single TCP stream. In Figure 5, the achievable TCP throughput in Mbps is plotted for various combinations of system parameters. With the default setting of the Linux kernel, one can achieve 82% of the 10Gbps line rate. Affinity helps in improving the throughput. Thread Affinity results helps in achieving 17.3% improvement of the throughput over the default case. Interrupt Affinity achieves a 6% improvement over the default setting. Thus at 10Gbps, Thread affinity plays a crucial role in achieving close to line rate performance. This is mainly due to the higher protocol overhead and lower cache misses associated with TCP. A combination of the system parameters achieves close to line rate performance. A fully affine (IATAMA) NUMA-based system achieve close to line rate performance of 9.79Gbps. A tuned NUMA based system sustains 850Mbps more throughput than a tuned SMP-based system.

F. Additive Effects of System Parameters on Throughput of a Two Concurrent TCP Stream

Additive effects of the affinities are desirable, as they provide an estimate of the scalability of systems, as we strive towards Terabit/s. In Figure 6, twice the throughput achieved by a single TCP stream (Figure 5), in Mbps is shown, via the dashed bars. The achievable TCP throughput in Mbps for two concurrent TCP streams, for the different system parameters combinations is also plotted. A system parameter or its combination is considered to be additive if the throughput of two streams is equal to twice the throughput of a single stream. With the default setting of the Linux kernel, one can achieve 60% of the 20Gbps line rate. Thread Affinity has a higher additive effect than interrupt affinity for two concurrent TCP streams. This is due to the fact that the overhead associated with TCP protocol processing and cache misses, in the Interrupt Affinity case, is higher than forwarding the interrupt processing, in the Thread Affinity case. Interrupt and Thread affinity (IATA) together achieve a line rate of 20Gbps and thus, demonstrate an additive effect. The combination of Thread, Interrupt and Memory affinity (IATAMA) also demonstrate an additive effect.

A tuned SMP system is unable to achieve additive performance for 2 concurrent TCP streams. It also sustains 3200Mbps (17%) lower throughput than a tuned (IATAMA) NUMA-based system. This is primarily due to the shared

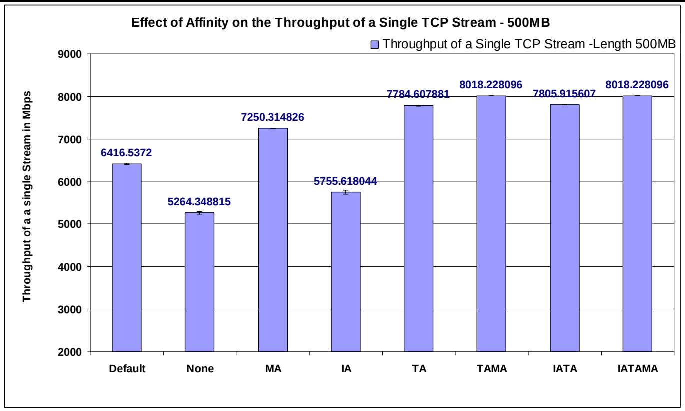

Figure 7: Effect of Affinities on the Throughput of a single TCP Stream for a payload of 500MB

single Front side-bus and the shared North-Bridge design on the SMP-based system.

G. *Effects of System Parameters on Memory Intensive Data Transfer*

We evaluate the effects of the system parameters on the achievable throughput of a single TCP stream for a payload of 500MB. In Figure 7, the achievable TCP throughput in Mbps is plotted for various combinations of system parameters. Memory affinity plays a very critical role for large payloads. However, Thread affinity has more significant effects on the achievable throughput than Memory affinity. Thread and Memory affinity together achieve the maximum throughput for a payload of 500MB.

V. Discussion

NUMA-based systems, with Thread and Interrupt affinity, sustain a line rate of at 20Gbps due to their multi-rail design, wherein each rail operates independent, and in parallel, to the other rails due to the point-to-point dedicated hyper-transport bus. In comparison, SMP-based systems are not true multi-rail systems due to the shared front-side bus and North-Bridge. However, in Intel's roadmap, the shared architecture is being replaced by point-to-point component system interface (CSI) based architecture. We also found that the current 10G Myrinet Network driver design does not take advantage of multi-core technology. The receive and transmit interrupt processing of the NIC, occurs on the same core. This limits the performance at higher data rates and bi-directional traffic. For scaling performance towards terabit/s, the NIC device driver's design needs to be multi-core aware and distribute receive and transmit processing among the different cores of a processor.

VI. Related Work

Effects of affinities on compute intensive workloads, for multi-core SMP and NUMA-based processors have been studied in [5]. In [6], effective use of multiple Nics for efficient transfer of data has been studied. In our multi-rail approach, we optimize the end-system performance for network intensive workloads by considering both multi-core and multi-NIC technologies.

In [10], the effects of Interrupt affinity and Thread affinity on the throughput of a TCP stream at 1Gbps, was investigated. We build upon this work and extend the analysis to 10Gbps for both TCP and UDP streams. Additionally, we also consider the effects of Memory affinity in addition to the Thread and Interrupt affinity considered in [10].

VII. Conclusion and Future Work

In this paper, we presented a multi-rail approach for future Terabit/s E-Science applications. System effects such as interrupt, memory, thread and core affinities play a critical role on the achievable throughput for additive performance - a key property for scalable performance. Thread affinity plays a critical role on the achievable throughput of Network Intensive workloads. The combined effects of Thread and Interrupt affinity is additive and helps in achieving line rate performance as we scale the performance to 20Gbps. NUMA based systems were able to achieve additive performance and line rate performance as compared to SMP-based systems.

These results can help guide the development of future intelligent middleware that will automatically optimize the performance based on system parameters and conditions, thus making it easier for applications developers, who are often not systems experts, to make full use of the system capabilities. We are currently prototyping a multi-wavelength, multi-lane Network card and optimizing our end-system and network aware transport protocol Celeritas [4][11] for multi-rail based architectures to enable future e-Science applications achieve terabit/s performance.

Acknowledgment

We would like to thank Alan Verlo, Lance Long, and Patrick Hallihan for their help in configuration of the testbed. The Electronic Visualization Laboratory (EVL) at the University of Illinois at Chicago specializes in the design and development of high-resolution visualization and virtual-reality display systems, collaboration software for use on multi-gigabit networks, and advanced networking infrastructure. This material is based upon work supported by the National Science Foundation (NSF), awards CNS-0224306, CNS-0420477, OCI-0229642 and OCI-0441094, as well as the NSF Information Technology Research (ITR) cooperative agreement (OCI-0225642) to the University of California, San Diego (UCSD) for "The OptIPuter." EVL also receives funding from the National Institutes of Health, the State of Illinois, the Office of Naval Research on behalf of the Technology Research, Education, and Commercialization Center (TRECC).

References

[1] J. Leigh, L. Renambot et al, “The Global Lambda Visualization Facility: An International Ultra-High-Definition Wide-Area Visualization Collaboratory”, Journal of Future Generation Computer Systems (FGCS) Vol 22 (2006),

[2] M. Tomizawa, J. Yamawaku, Y. Takigawa, M. Koga, Y. Miyamoto, T. Morioka, K. Hagimoto, "Terabit-LAN with optical virtual concatenation for grid applications with super computers," in: Proceedings of OFC2005, paper OthG6, 2005.

[3] A. Hirano, L. Renambot et al., "The first functional demonstration of optical virtual concatenation as a technique for achieving Terabit networking," Future Generation Computer Systems Vol 22 (2006) pp. 876-883.

[4] X. Wang, V. Vishwanath, B. Jeong, R. Jagodic, E. He, L. Renambot, A. Johnson, J. Leigh, LambdaBridge: A Scalable Architecture for Future Generation Terabit Applications. Broadnets 2006 - San Jose, CA, 10/01/2006 - 10/05/2006.

[5] L. Chai, Q. Gao and D. K. Panda, Understanding the Impact of Multi-Core Architecture in Cluster Computing: A Case Study with Intel Dual-Core System, Int'l Symposium on Cluster Computing and the Grid (CCGrid), Rio de Janeiro - Brazil, May 2007

[6] J. Liu, A. Vishnu and D. K. Panda, Building Multirail InfiniBand Clusters: MPI-Level Design and Performance Evaluation. SuperComputing Conference, Nov 6-12, 2004, Pittsburgh, Pennsylvania.

[7] Linux Kernel Documentation on IRQ affinity.

[8] Linux Numactl manual pages.

[9] Nuttcp : http://www.lcp.nrl.navy.mil/nuttcp/

[10] A. Foong, J. Fung, D. Newell, S. Abraham, P. Irelan, A. Lopez-Estrada, Architectural Characterization of Processor Affinity in Network Processing Performance Analysis of Systems and Software, 2005. ISPASS 2005. IEEE International Symposium on March 20-22, 2005 Page(s):207 – 218.

[11] V. Vishwanath, J. Leigh, E. He, M. D. Brown, L. Long, L. Renambot, A. Verlo, X. Wang, T. A. DeFanti. Wide-Area experiments with LambdaStream over dedicated high-bandwidth networks. Workshop on High Speed Networking, *IEEE INFOCOM 2006*.