Exploration of adaptive network transfer for 100 Gbps networks

Climate100: Scaling the Earth System Grid to 100Gbps Network

October 1, 2012

Project period of
April 1, 2011 through September 30, 2012

Principal Investigators
Alex Sim[1], Dean N. Williams[2]

Project member: Mehmet Balman[1]

Project Website: http://sdm.lbl.gov/climate100

1 Lawrence Berkeley National Laboratory

2 Lawrence Livermore National Laboratory

Table of Contents

1 Overview

Large scientific experiments and simulations running at precision levels provide better insight into scientific phenomena. However, the amount of data from these experiments and large simulations is in the multi-petabyte range, and expected to grow exponentially. This data explosion exist in nearly all fields of science, including astrophysics and cosmology, high energy physics, material science, climate modeling, fusion, and biology, to name a few. In addition, these scientific data need to be shared by increasing numbers of geographically distributed collaborators connected by high performance networks.
The Climate100 project would study massive datasets, high bandwidth 100 Gbps networks, and data transport and management technologies to enable at-scale experimentation, in particular with climate data management and transport. The result of the project would improve the understanding and use of network technologies and transition the science community to a 100 Gbps network and beyond for production and research. The goals of the project are summarized as:

- Study the effective use of emerging 100Gbps networks for massive datasets in the application of climate change simulation.
- Study data transport and management technologies to enable at-scale experimentation.
- Improve the understanding and use of network technologies in order to transition the science community to a 100 Gbps network and beyond.
- Focus on climate data management and transport for requirements of the Earth System Grid.

1.1 The Earth System Grid Federation and Expectations from 100-Gbps System

The ESGF is a successful international collaboration that is recognized as the leading infrastructure for the data management and access of large distributed-data volumes in climate-change research. ESGF, which originated in the Earth System Grid, has evolved to encompass tens of data centers worldwide, collectively holding tens of petabytes of data, and serving tens of thousands of users through ESG P2P Gateways and Data Nodes.
For the IPCC AR5 CMIP-5 data archive is over 30 distributed data archives totaling over 10 PB. The CMIP-5 Replica Centralized Archive (RCA), which the two-dozen major international modeling groups from Japan, U.K., Germany, China, Australia, Canada and elsewhere have replicated to LLNL to form, is estimated to exceed 2 PB of data set volume. Not all data would be replicated to CMIP-5 RCA at LLNL, but the majority of the 10 PB of data will be accessible to users from the ESGF P2P Gateways. Figure 1 shows the envisioned topology of the ESGF based on 100Gbps ESnet network connections to provide a network of geographically distributed Gateways, Data Nodes, and computing facilities in a globally federated, built-to-share scientific discovery infrastructure.
It is projected that by 2020, climate data will exceed hundreds of exabytes (XB). While the projected distributed growth rate of climate data sets around the world is certain, how to move and analysis ultra-scale data efficiently is less understood. The DOE resources at ALCF, LLNL, NERSC and OLCF over 100Gbps are of interest to ESGF for climate analysis.

This document gives a brief overview on the technical progress and achievements in Climate100 project for the project period from January 1, 2012 to September 30, 2012, in addition to the previously released mid-project report from April 1, 2011 to December 31, 2011, which can be found on https://sdm.lbl.gov/climate100/docs/Climate100-MidReport-20120201.pdf.

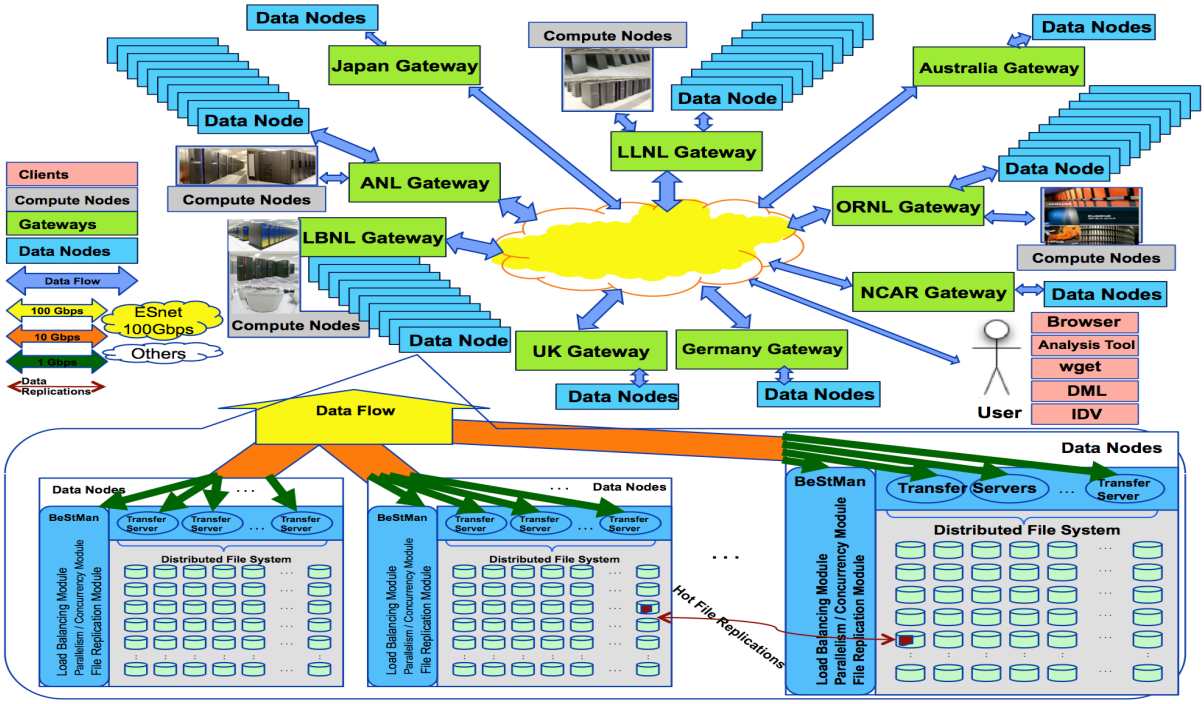


Figure 1: The envisioned topology of the ESGF based on 100Gbps ESnet network connections.

2 Project Tasks

2.1 Summary of the ARRA-supported Climate100 from Oct. 1, 2009 to Mar. 31, 2011

During the period of Oct. 1, 2009 through Mar. 31, 2011, the project was funded through ARRA program, and the following tasks have been accomplished.

- We have established research tasks and integration challenges: Climate data movement use cases were identified, and assessment on requirements and goals were established.
    - Study of the data movement protocols over 100 Gbps network and integration with client applications.
    - Study of the data transfer test cases over 100-Gbps networks and contribute to the system enhancements.
    - Simulation of Round-Trip Time (RTT)/network delays on 100-Gbps testbed. It is understood that large files in size will be transferred for testing on the testbed to avoid complexity of dataset characteristics in large variance of file sizes and file system and storage I/O.
    - Backend storage input/output (I/O) performance for climate data sets. Climate data sets consists of a mix of large and small sized files, and generally much smaller than High Energy Physics (HEP) data files. Parallel/Distributed file system performance on small files is generally very poor. The typical file size distribution in climate dataset in Intergovernmental Panel on Climate Change (IPCC) Coupled Model Intercomparison Project, phase 3 (CMIP-3) indicates that most of the data files have less than 200MB of file size (~60-70% of all files), and among those smaller files, file sizes less than 20MB have the biggest portion (~30% of all files).
- We have studied efficient data transfer mechanism for climate dataset.

    - We have studied the climate dataset characteristics and concurrency vs. parallelism,
    - We developed dynamic adaptation algorithm and model-based concurrency selection algorithm for efficient data movement,
    - We conducted GridFTP transfer experiments over 10Gbps networks, and
    - We developed a visualization tool for data transfer management.
- We have studied network transfer protocols.
    - We have conducted experiments in Remove Direct Memory Access (RDMA) based data movements over wide-area networks.
- We have experimented a large-scale climate simulation analysis on Cloud computing from the data movement perspective.
- We have participated in the ESGF community, by supporting ESG Gateway and Data Node deployment at NERSC.
- We have published 6 conference papers and 3 technical reports during this period, in addition to 1 award from ACM SRC in SC'10, 1 news article and 2 presentations.

The details of the project tasks can be found on final ARRA project report, https://sdm.lbl.gov/climate100/docs/Climate100-Report-2011-Final.pdf.

2.2 Support for Data movement and performance optimization for ESGF

We have engaged in IPCC CMIP-5 dataset replications among international ESGF data centers, and collaborated in network data transfer performance optimization, through supporting the Bulk Data Mover (BDM), which is the data transfer management tool in the ESGF community, as well as the Globus Online usage.
We also have collaborated with NERSC to support faster network data transfers by having NERSC as a data hop, which NERSC provided 100TB of disk allocation to accommodate climate data replication and NERSC ESGF P2P efforts. Current replication size is about 90TB. We have collaborated with ESnet for Trans-Atlantic network issues and all other network performance issues. We have collaborated with other climate data modeling centers for the efficient replication techniques.

2.2.1 Summary

- Help CMIP-5 data replication and network transfer optimization, in collaboration with ESnet and NERSC,
    - from BADC, UK to LLNL and NERSC,
    - from NERSC to LLNL and NCI, Australia,
    - from NCI, Australia to NERSC,
    - from Japan to NERSC, and
    - from DKRZ, Germany to NERSC and LLNL.
- NERSC disk space allocations for 100TB to support CMIP-5 data replication and NERSC ESGF P2P efforts.
- Collaboration with ESnet and international data modeling centers for CMIP-5 dataset replications and network performance.

2.3 Development of Memory-mapped Zero-copy Network channel Technology

2.3.1 Moving Climate Datasets Efficiently

An important challenge in handling climate data movement is the lots-of-small-files problem. Each climate dataset includes many files, relatively small in size. In addition to network optimization, data

transfers require appropriate middleware for managing and transferring a large number of small files efficiently. The standard file transfer protocol FTP establishes two network channels. The control channel is used for authentication, authorization, and sending control messages such as what file is to be transferred. The data channel is used for streaming the data to the remote site. The data channel stays idle while waiting for the next transfer command to be issued. In addition, establishing a new data channel for each file increases the latency between each file transfer. The latency between transfers adds up, and as a result, the overall transfer time increases and total throughput decreases. This problem becomes more drastic for long distance connections where round-trip-time is high. Most of the end-to-end data transfer tools perform best with large data files, and require managing each file movement separately. As a result, dealing with many files imposes bookkeeping overhead for each file. The Globus Project also recognized the performance issues with small files, and added a number of features such as concurrent transfers, pipelining, and connection caching to their GridFTP tool to address this issue. When there are many files, multiple files are transferred concurrently in order to overlap waiting times in data channels and minimize effects of the bookkeeping overhead. However, many concurrent transfers impose extra cost in terms of system and network resources, and may result in poor performance.

We have developed a new approach, in which data movement is not file-centric, but is block-based. The idea is to combine files into a single stream on the fly. We have developed a new block-based data movement technology based on this approach called Memory-mapped Zero-copy Network channel (MemzNet), implemented a new tool, and compared it with GridFTP on the ANI 100Gbps testbed. In this technology, data files are aggregated and divided into simple data blocks. Blocks are tagged and streamed over the network. Each data block's tag includes information about the content inside. For example, regular file transfers can be accomplished by adding the file name and index in the tag header. Since there is no need to keep a separate control channel, it does not get affected by file sizes and small data requests. While testing, we have achieved the same performance regardless of the file sizes, without compromising on the optimum usage of the available network bandwidth. During the Supercomputing 2011 (SC'11) 100Gbps demo, we showed our prototype tool, using real data from the climate simulation. Since the dataset includes many files, and the files sizes are relatively small compared to available bandwidth (e.g., 100MB for 10Gbps), we needed more than 16 concurrent transfers in GridFTP to reduce the effect of having small file sizes. We have observed better performance and efficiency with our new technology in transferring large datasets especially with many files.

Design of this technology was based on decoupling disk and network I/O operations; so, read/write threads can work independently. It allows us to have different parallelism levels in file system access and in network transfer operations. In order to do that, we have designed a memory cache managements system that is accessed in blocks. Data is read directly into these memory blocks. These memory blocks are logically mapped to the memory cache that resides in the remote site. It is the responsibility of the backend threads to transmit blocks over the network. The memory cache is a simple circular buffer, and its synchronization is accomplished based on the tag header that also includes a transaction id. Main concept in this design is that application processes interact with the memory blocks, and they do not need to deal with the network layer. The memory-mapped network access also provides the necessary architecture for zero-copy network operations. In principle, this method can easily support zero-copy because it can avoid copying from application memory to kernel memory.

The following figures show the performance of our new technology. Figure 1 shows test results with GridFTP, and with the first implementation of our technology. The first part is with GridFTP (using concurrent transfers); where the throughput value fluctuates over time. The second part in the graph is with our implementation of the new technology, where it gives steady performance. We used our tool in the SC'11 demo, and were able to achieve 83Gbps total throughput in the ANI 100Gbps connection from NERSC to ALCF. The demo configuration consisted of multiple hosts at each end, and each host has only one 10Gbps NIC connected to the network. The maximum achievable throughput in this environment

(with memory to memory iperf transfers) was also 83Gbps. In our demo, data files were read from a GPFS system, which could provide 120Gbps read performance.

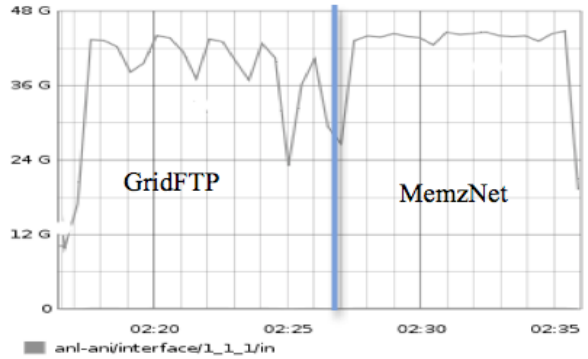


Figure 1: Performance from GridFTP and MemzNet

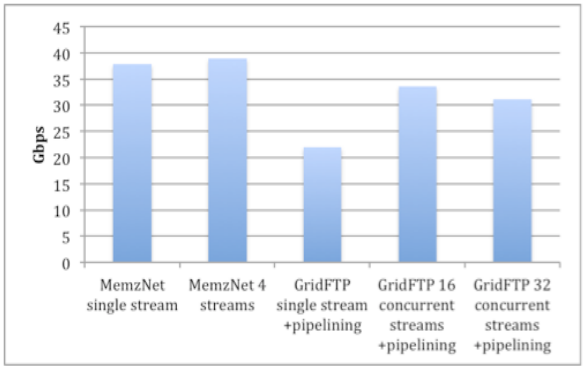


Figure 2: Throughput Performance from GridFTP and MemzNet

After the SC'11 demo, we have continued our evaluation in the ANI 100Gbps testbed. In Figure2, two hosts, one at NERSC and one at ALCF, were used, and each host is connected with four 10Gbps NICs. Total throughput was 40Gbps between two hosts. We did not have access to a high performance file system on the test-bed, and we simulated the effect of file size characteristics by creating a memory file system (tmpfs with a size of 20G). We created files with various sizes (i.e., 10M, 100M, 1G) and transferred those files continuously while measuring the performance. In both MemzNet and GridFTP experiments, TCP buffer size is set to 50MB in order to get the best throughput. The pipelining feature was enabled in GridFTP. A long list of files is given as input. Figure 2 shows the performance results with 10MB files. We initiated four server applications at ALCF node (each running on a separate NIC), and four client applications at NERSC node. In the GridFTP tests, we tried both 16 and 32 concurrent streams, with -cc option. MemzNet-based tool was able to achieve 37Gbps of throughput, while GridFTP was not able achieve more than 33Gbps.

2.3.2 Design of Memory-mapped Zero-copy Network Channel Technology

The architecture of Memory-mapped Zero-copy Network channel (MemzNet) consists of two layers, as shown in Figure 3: a front-end and a back-end. Each layer works independently so that each layer can be tuned separately. Transmitting data over the network is logically separated from the reading/writing of data blocks. Having separate front-end and back-end components has other benefits, giving an ability to have different parallelism levels in each layer. Those layers are tied to each other with a block-based pre-allocated memory cache, implemented as a set of shared memory blocks. In the server, the front-end is responsible for the preparation of data, and the back-end is responsible for sending of data over the network. On the client side, the back-end component receives data blocks and feeds the memory cache so that the corresponding front-end can get and process data blocks. These memory caches are logically mapped between client and server.

MemzNet introduces dynamic data channel management and asynchronous movement of data blocks. In file transfers, the front-end component requests a contiguous set of memory blocks. Once they are filled with data read from the file system, those blocks are released, so that the back-end component can retrieve and transmit the blocks over the network. Data blocks include content information, i.e. file ID, offset and size. Therefore, there is no need for further communication between client and server in order to initiate file transfers. The memory management layer helps to send the data in large chunks when it is ready. A single stream is sufficient to fill up the network pipe. This is analogous in concept to on-the-fly 'tar' approach, bundling and sending many files together.

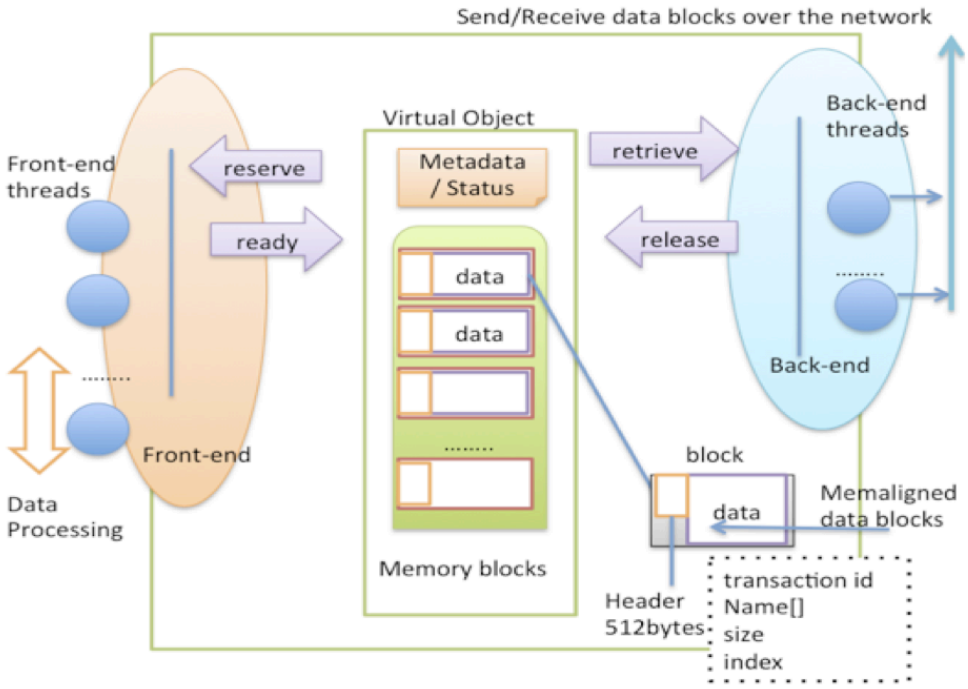


Figure 3: Architecture design of Memory-mapped Zero-copy Network channel (MemzNet)

Moreover, the data blocks can be received and sent out-of-order and asynchronously. Since we do not use a control channel for bookkeeping, all communication is mainly over a single data channel, through a fixed single port. Bookkeeping information is embedded inside each block. This has some benefits for ease of firewall traversal over wide-area. Besides, we can increase/decrease the number of parallel streams dynamically, if necessary, without the need to setup a connection channel, since each block includes its bookkeeping information.

In the MemzNet technology, every block includes its own metadata, and connection management is dynamic. Different from other file transfer protocols, there is almost no cost for changing the number of parallel streams; hence, it perfectly matches adaptive parameter tuning.

Although we prototyped and tested MemzNet only for file transfers, the concept can also be extended to build efficient end-to-end communication channel for general network applications, as well as a library that can be used as a plug-in or a driver by external tools and applications. These libraries can be used to deliver streaming over the network for remote data analysis.

2.3.3 Summary

- We have studied data transfer protocols and management techniques for efficient data movements with the climate datasets.
- We have developed Memory-mapped Zero-copy Network channel (MemzNet) technology that shows promising results in efficient block-based data movement.
- We have further improved our prototype implementation of the MemzNet technology to achieve 100Gbps maximum throughput on ANI 100Gbps testbed, as shown in Figure 4.

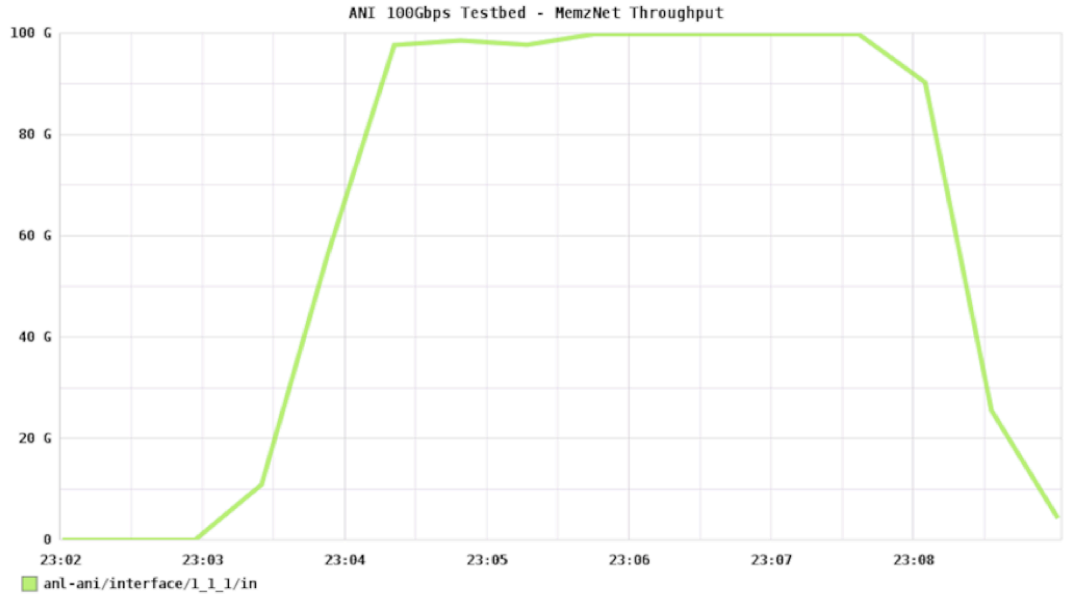


Figure 4: MemzNet data transfer performance over 100Gbps network connection, showing the movement of total 3TB in 5 minutes.. 100Gbps bandwidth is fully utilized.

2.4 System Performance Study over 100Gbps Network Transfers

During the experiments on ANI 100Gbps testbed, we have realized that one of the major obstacles in use of high-bandwidth networks is in the limitation of host system resources. 100Gbps is beyond the capacity of most of today's commodity machines, since we need substantial amount of processing power and involvement of multiple cores to fill a 40Gbps or 100Gbps network. As a result, host system performance plays an important role in the use of high-bandwidth networks. When the ANI 100Gbps testbed became ready, we conducted a large number of experiments with our new block-based MemzNet tool and with current available file-based data movement tools. We found that, in general, the block-based method works better, mostly because it uses very few streams. However, the explanations of the experimental results need to be better understood and validated in order to understand in terms of the parameter tuning and optimization to achieve 100Gbps performance, such as how host performance affects the overall bandwidth usage, how resources are consumed in an end-to-end network transfer, and how the use of multiple streams affect efficiency and throughput performance.

During the SC'11 demo, we have developed a simple performance module embedded in our block-based MemzNet tool to interactively monitor the amount of data sent/received over the network. In the demo configuration, each host is connected with a 10Gbps NIC. An important problem we have observed was that some hosts were able to have almost 10Gbps throughput, but some hosts were only able to achieve less than half of what's available, and their instant throughput usage fluctuated significantly. In a normal case, total bandwidth should be distributed equally among each connection. This unusual observation became more apparent when the number of network streams and the number of hosts connecting were increased. With UDP, we were able to saturate the network, but still seeing many packages lost. There was a big variance in bandwidth usage among different hosts in TCP. We also observed an increase in the number SACK recovery in the hosts in which throughput usage fluctuated a lot. This problem has been resolved when the 100Gbps aggregator switch, a Juniper 4500, has been removed from the configuration, and the problem was in balancing the load across multiple 10G circuits. We think that the aggregator switch was delaying some packages and that situation was causing problems when the host network stack is trying to adjust TCP data flow. Later, we did not have a chance to make further experiments with the

same SC'11 demo systems, but this first hands-on experience with a 100Gbps network led us to investigate further and analyze end-to-end data movements in more details. Since SC'11, there have been several updates/fixes in the network configuration based on this experience. We have continued our experiments in the ANI 100Gbps testbed where we could finally achieve 100Gbps throughput. This prompted us to look into network performance issues and the effects of the resource utilization in the host systems.

In 10Gbps transition from 1Gbps networks many years ago, we required intense fine-tuning both in network and application layers to take advantage of the higher network capacity. Different from that, we now reached to the limits of internal bus speeds in commodity hardware, and we necessitate substantial amount of processing power and involvement of multiple cores to fill up a 40Gbps or 100Gbps network pipe. Today's computer architecture faces the limited memory performance issues, compared to the increasing processing speed. Besides, the network latency stays the same as we increase the bandwidth. 100Gbps networks demand high performance in CPU and memory usage compared to 10Gbps networks, and use of multiple NIC cards and multiple cores need complex tools that are optimized for data movement operations. The performance of the end-systems will play an important role in use of next-generation networks.

2.4.1 Experimental setup

In the current ANI 100Gbps testbed environment, we have three hosts at each end, and each host has four 10Gbps network cards. In order to achieve100Gbps throughput, we needed fine host tuning. We have observed that the host system performance can easily be the bottleneck. This led us to look into the host systems resource utilization.

First, we have studied application profiling including memory usage, number of context switches, time spent waiting for I/O completion, user and system time, call graph of system calls, time spent in each user operation. After several experiments, we concluded that collecting performance metrics directly from the host system gives better and more useful results. We have collected monitoring data from Linux subsystems (CPU, memory and IO) in real time, when data movement operations are performed. Our evaluation shows that multiple NICs and concurrent transfers increase the context switch time, system CPU usage, number of interrupts and system load, both in the server and client sides for an end-to-end data movement. In our experiments, we started the data transfer operation with different parameters and let it run for 5 minutes. While transfer was active, we collected system wide performance metrics and compared those results with the total throughput achieved. Performance metrics were collected both in the client and server sides. We tested the effect of parallel streams for memory-to-memory transfers. In concurrent transfers, we started multiple transfer applications at the same time with a single stream for each application. Also, we set the TCP buffer manually to see how it affects throughput and system wide performance metrics, as we observed that the best performance resulted from 50M buffer size on the ANI testbed environment. Furthermore, we simulated the effect of moving many files by creating a memory file system (tmpfs with a size of 20G) in each host. We created files in various sizes (i.e., 10M, 100M, 1G) and transferred those files continuously while measuring the performance.

2.4.2 Test results and conclusion

We present several graphs from our test results from the ANI 100Gbps testbed.

Figure 5 shows the total throughput when three hosts and a total of 10 NIC pairs were used. The maximum available bandwidth was 100Gbps. The maximum throughput was achieved with four streams. However, even if TCP buffer was set properly, we observed that performance dropped after 4 parallel streams, as shown in Figure 6.

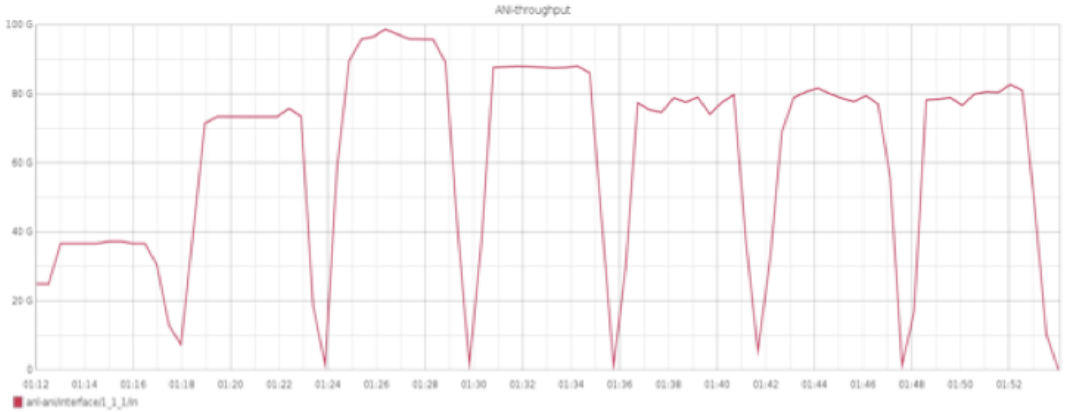


Figure 5: ANI testbed 100Gbps (10x10NICs, three hosts): Throughput vs the number of parallel streams [1, 2, 4, 8, 16, 32, 64 streams - 5min intervals], TCP buffer size is default.

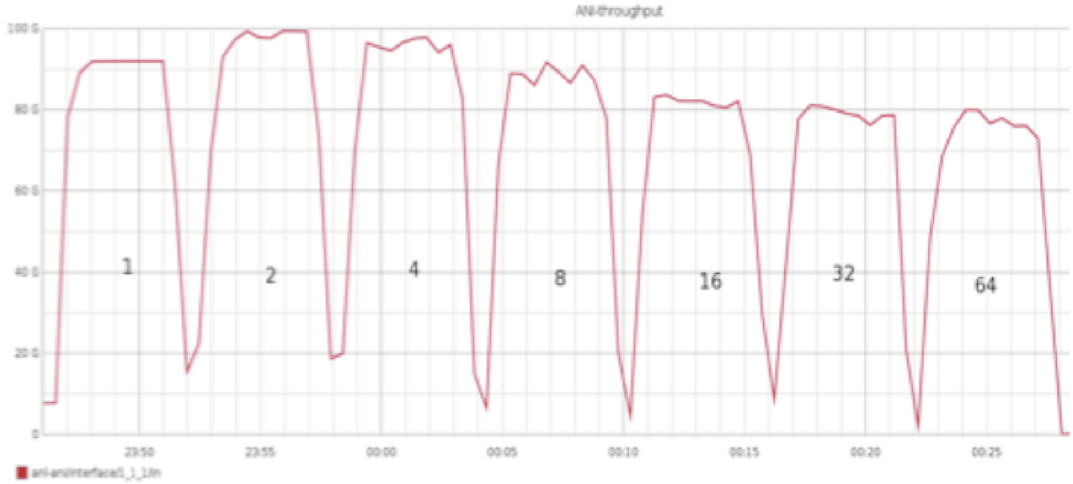


Figure 6: ANI testbed 100Gbps (10x10NICs, three hosts): Throughput vs the number of parallel streams [1, 2, 4, 8, 16, 32, 64 streams - 5min intervals], TCP buffer size is 50M.

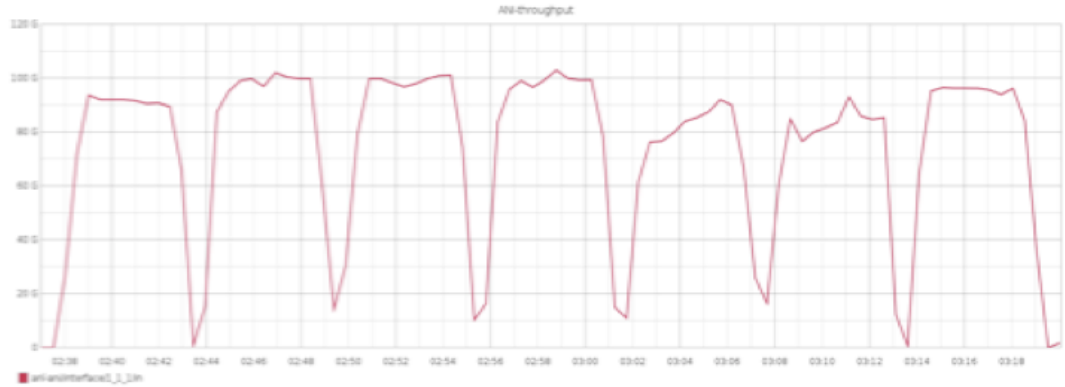
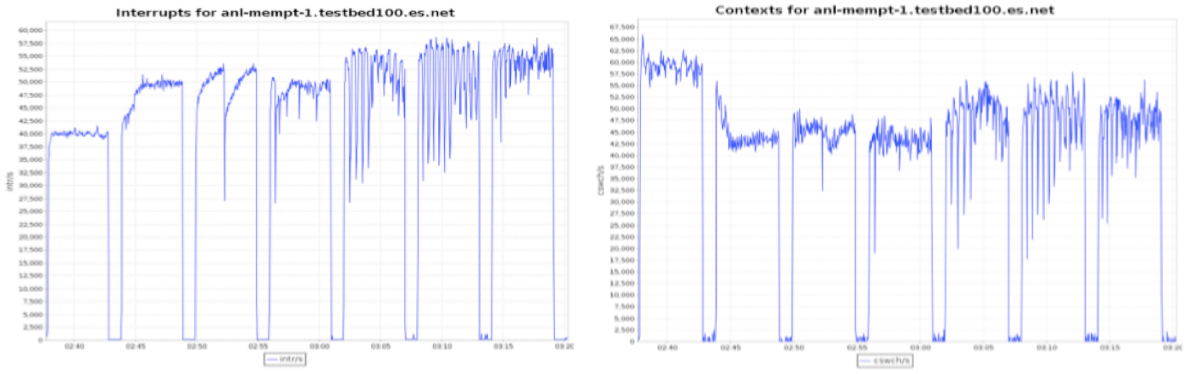


Figure 7: ANI testbed 100Gbps (10x10NICs, three hosts): Throughput vs the number of concurrent transfers [1, 2, 4, 8, 16, 32, 64 concurrent jobs - 5min intervals], TCP buffer size is 50M.

Figure 8: ANI testbed 100Gbps (10x10NICs, three hosts): CPU utilization vs the number of concurrent transfers [1, 2, 4, 8, 16, 32, 64 concurrent jobs - 5min intervals], TCP buffer size is 50M

(a) (b)

Figure 9: (a) ANI testbed 40Gbps (4x10NICs, single host): interface traffic vs the number of parallel streams [1, 2, 4, 8, 16, 32, 64 streams - 5min intervals], TCP buffer size is 50M, (b) ANI testbed 100Gbps (10x10NICs, three hosts): interface traffic vs the number of concurrent transfers [1, 2, 4, 8, 16, 32, 64 streams - 5min intervals], TCP buffer size is 50M.



(a) (b)

Figure 10: (a) ANI testbed 100Gbps (10x10NICs, three hosts): Interrupts vs the number of concurrent transfers [1, 2, 4, 8, 16, 32, 64 concurrent jobs - 5min intervals], TCP buffer size is 50M, (b) ANI testbed 100Gbps (10x10NICs, three hosts): Context switches vs the number of concurrent transfers [1, 2, 4, 8, 16, 32, 64 concurrent jobs - 5min intervals], TCP buffer size is 50M.

Figure 8 shows the CPU utilization for the memory-to-memory concurrent transfers. We initiated multiple data movement jobs (each job using a single stream) at the same time. Each peak in these graphs represents a different test with different number of concurrent operations. As shown in Figure 9-b on concurrent transfers, the interface traffic shows different characteristics as compared to Figure 9-a on parallel streams. Figure 10-a shows the interrupts per second, and Figure 10-b shows the number of context switches per second.

We concluded that parallel streams and concurrent transfers have different effects if more than one NIC is involved. The total throughput drops as we increase the number of parallel streams. However, we do not see the same effect with concurrent transfers, although logically the number of sockets used is the same (parallel streams and concurrent transfers use multiple TCP sockets). Even though the CPU utilization is similar, concurrent streams perform better consistently in most cases, as shown in Figure 7. However, context switches increases as concurrency level increases.
When we collect traffic information from the interface, we can see the effect of multiple streams clearly; with a single stream and TCP buffer set, we see a steady performance peak, usually the best possible throughput value. Setting up the TCP buffer precisely also slightly reduces context switching.
When we set the interrupt affinity (each NIC is assigned to separate core, instead of sending all NIC interrupts to a single core), we observe an increased CPU usage and increased context switches.
With these results, it seems evident that using 100Gpbs efficiently is a feasible goal for the future, but careful evaluation is necessary to avoid host system bottlenecks. Further studies are needed for a good understanding of how host performance affects the overall bandwidth usage.

2.4.3 Summary

- We have experimented and studied system performance bottlenecks during data transfers over 100Gbps network, and observed relationships between the network streams and host system resource utilization.
- We produced preliminary results and further study plans on the study of system resource performance during the network transfers over 100Gbps network.

3 Publications, presentations and other activities

Papers and talks presented during the period of April, 2011 through September, 2012:

3.1 Publications

1) "*Experiences with 100Gbps Network Applications*", Mehmet Balman, et al., The Fifth International Workshop on Data Intensive Distributed Computing (DIDC), 2012
2) "*Earth System Grid Federation: Infrastructure to Support Climate Science Analysis as an International Collaboration. A Data-Driven Activity for Extreme-Scale Climate Science*", D. N. Williams et al., Data Intensive Science Series: Chapman & Hall/CRC Computational Science, 2012.
3) "*Earth System Grid Center for Enabling Technologies (ESG-CET): A Data Infrastructure for Data-Intensive Climate Research*", D. Williams, et al., SciDAC Conference 2011.

3.2 Presentations

1) "*Experience with 100Gbps Network Applications*", M. Balman, LBNL CRD AHM, 2012.
2) "*Data Movement over 100Gbps Network*", M. Balman, A. Sim, ESGF P2P Meeting, LLNL, 2011.
3) "DOE's Climate and Earth System Modeling Town Hall: Climate Model Intercomparison and

Visualization Efforts for Next Generation Needs – Climate100: Scaling Climate Applications to 100Gbps Network", Dean N. Williams, American Geophysical Union (AGU) Town Hall meeting, San Francisco, CA, 2011.

3.3 Demonstration

1) "*Scaling the Earth System Grid to 100Gbps Networks*", M. Balman, A. Sim, IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC'11), Seattle, WA, 2011. https://sdm.lbl.gov/climate100/docs/SC11-demo.pdf.

3.4 Open source

1) "*Climate100 Toolkit*", including MemzNet prototype, A. Sim, M. Balman, LBNL CR-3098, open source under a BSD license with a grant back provision. Available on https://codeforge.lbl.gov/projects/clim100/.

3.5 Workshop

1) International Workshop on Network-Aware Data Management Workshop (NDM2012), in conjunction with the IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC'12). Mehmet Balman as General Chair of the workshop.
2) International Workshop on Network-Aware Data Management Workshop (NDM2011), in conjunction with the IEEE/ACM International Conference for High Performance Computing, Networking, Storage and Analysis (SC'11). Mehmet Balman as General Chair of the workshop.