Data Transfer Node for Intercontinental Big Data Transfer

Purit Phan-udom[1], Vasaka Visoottiviseth[1], Ryousei Takano[2]

[1]Faculty of Information and Communication Technology, Mahidol University, Nakhon Pathom, Thailand

[2]Information Technology Research Institute, National Institute of Advanced Industrial Science and Technology, Tsukuba, Japan

ABSTRACT

Our objective is to deliver a set of light-weight yet powerful software stack for ABCI supercomputer's data transfer node to carry out massive data transfer, monitor real-time data traffic under a variety of conditions, and provide network measurements for the user.

In our approach, we simulated the actual working system and hardware architecture of ABCI's in a contained experimental environment, using a 3-node mini-cluster. The use case was to exchange data between ABCI and San Diego Supercomputer Center in California. We used FDT (Fast Data Transfer), a Java-based application for efficient high-speed data transfer, as the core program to direct all of the data transfers between ABCI and its computing center partners. Pairing with the open-source OpenTSDB monitoring server, the statistical values generated by the transfer activities are then projected and visualized in the form of real-time graph. Next we implemented perfSONAR as for the network measurement testing and benchmarking tasks.

In a real environment, we were able to host a stable transmission and achieve average throughput of 6.4 Gbps using a kernel buffer memory tuning approach. The FDT instances successfully sent the time series data to the running OpenTSDB server which plotted the received data points in realtime. Lastly, with perfSONAR testpoint, a scheduled tests of throughput was set and could be visualized in time-domain graph.

TOOLS AND SOFTWARE

FDT 0.26

FDT can transfer a large set of files across the network without the TCP session restarting between each file, suitable for big data application comprising of multiple-file payloads. The basic client-server mode has two instances of the program running as a client and a server on each end device. FDT uses the capability of Java NIO libraries and other TCP/IP classes for its functionality.

OpenTSDB is a time series database server that reads the data points from the FDT instances of multiple metric series. Each point represents the smallest unit of data as an object consisting of metric's properties and the data value it carries. It uses Apache HBase to store the points and retrieve them for visualization based on the user's query as graph.

perfSONAR is a test and measurement tool designed to monitor and maintain the level of network performance. Its ability to look at the performance in multiple domains allows for accurate tests in different use cases, making use of multiple network command-line tools — psclock, owping, iperf3, traceroute, dnspy — for its testing routine.

SYSTEM ARCHITECTURE

Mini-cluster System

Node0 and Node1 were dedicated as the transfer nodes for data payload, and Node2 for OpenTSDB monitoring and perfSONAR test scheduler. The direct LAN connection between Node0 and Node1 allows for 10-gigabit bandwidth capacity. To simulate the RTT delay between ABCI and SDSC on this system, netem was used.

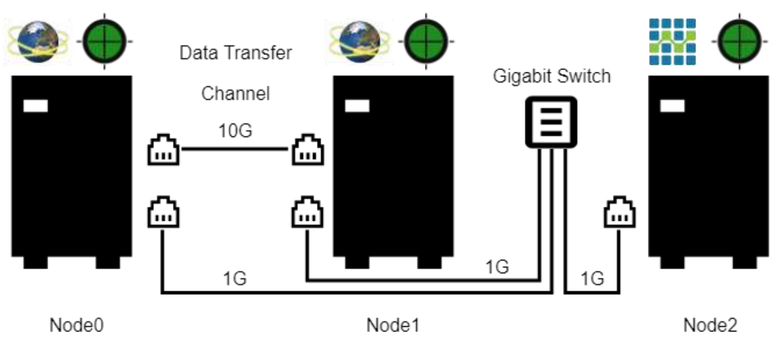

Fig. 1 Mini-cluster experimental environment system

ABCI-SDSC System

The placement of FDT, perfSONAR, and OpenTSDB on ABCI are all located in ABCI data transfer node. The node is connected to the storage — a DDN GRID Scaler (GFPS) — where the transfer payload is kept. The RTT delay between ABCI and SDSC was taken into account for system tuning of buffers.

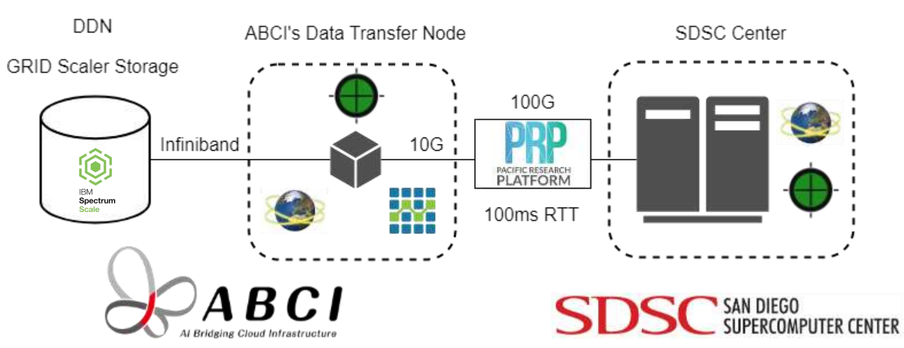

Fig. 2 ABCI - SDSC system implementation

IMPLIMENTATION AND RESULT

Mini-cluster Experimentation

By system tuning of the kernel buffer memory (sysctl) in the sender and receiver nodes, the transfer reach of over 4.4 Gbps for the average throughput over the virtual network delay (netem) put in effect, with the OpenTSDB able to plot the time-series graph from FDT instances in real time.

	Node0 (client)	Node1 (server)
net.core.wmem_max	100000000	100000000
net.core.rmem_max	100000000	100000000
net.ipv4.tcp_wmem	[4096 16384 10^9]	[4096 16384 10^9]
net.ipv4.tcp_rmem	[4096 87380 10^9]	[4096 87380 10^9]

Fig. 3 TCP buffer tuning parameter setting (unit in Byte)

Fig. 4 Graph plotted by OpenTSDB from the user query

ABCI-SDSC System Implementation

The actual transfer result between ABCI and SDSC, accountable for the higher hardware capability, achieved throughput utilization beyond that of the experimental outcome despite the identical tuning approach of the kernel memory, at an average of 6.5 Gbps over 100ms RTT delay. The payload for transfer used was a set of zebrafish brain scan images.

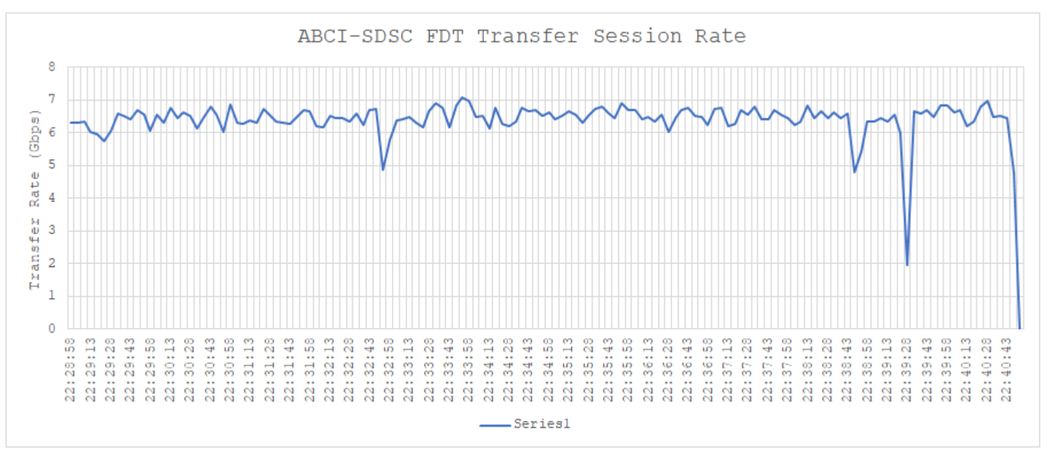

Fig. 5 FDT transfer rate between ABCI and SDSC

ACKNOWLEDGEMENT

This work was supported by the AIST ICT International Collaboration Fund.