PROPOSAL

Proposed Session Title:

High Performance Computing Networks for Apache Hadoop

Abstract (2000 word limit):

For the last 10 years, data magnitude has increased beyond traditional storage, tools and systems capabilities. Big Data frameworks enable unstructured, large magnitude data processing and analytics. To benefit from Big Data applications, there is a need for a scale-out cluster deployment connected via a network fabric. Deploying such clusters over low bandwidth, high latency and CPU exhausting fabrics impede efficient cluster construction. Utilizing High-Performance Computing (HPC) cluster technologies, customers are able to increase performance by two folds or more. With the expected 4,300% increase in data magnitudes by 2020 and customers storing 1ExaByte of data this year, building Big Data clusters are imperative for the success of businesses and academic research. The following article provides a glance into networking technologies and approaches to help current users of the Hadoop framework, the dominant Big Data application, to deploy Big Data clusters.

In the article we will describe:

1. The integration process of RDMA API into the Java based Apache Hadoop
2. Ways to better utilize server memory with RDMA and Apache Hadoop
3. Application performance comparison between RDMA based and socket based Apache Hadoop
4. Call for participation in an RDMA based open source HDFS™
5. Scaling Hadoop clusters with InfiniBand
6. Building flat networks for rack-aware applications
7. Improving SSDs benefits with low latency networks

The RDMA Programing Model, Highest Utilization of Network and CPU.

Hadoop framework and High Performance Computing (HPC) share similar requirements for scalability and linear performance growth. The RDMA programing model is widely used in the HPC market for efficient data transfer. In this presentation we explain the usage of RDMA API within the Apache Hadoop framework and the benefits of using it.

RDMA stack provides a fast API for applications on a scale-out system. Hadoop applications are deployed today on massive number of state-of-the-art servers and fast network gear, to benefits from this investment the application needs a contemporary networking stack. The RDMA based stack bypass the less efficient socket stack to and enable a low latency communication model for a distributed system application.

In addition, RDMA based networks minimize the usage of CPU cycles for data transfers, providing increased number of CPU cycles for the application maximize servers' throughput and utilization.

When an Application Hampers CPU Utilization

The Hadoop MapReduce framework is an exhausting CPU application, using HPC technologies such as InfiniBand and RoCE enable higher CPU utilization. The proof is shown via the November 2011 TOP500 supercomputer list which highlights that the most efficient clusters use an RDMA-based interconnect. RDMA-based clusters excel in their ability to move large amounts of data with minimal CPU intervention, freeing resources for Map and Reduce tasks. The RDMA-based interconnect enables more Hadoop Mappers and Reducers on the same machines than previously using sockets-based interconnects.

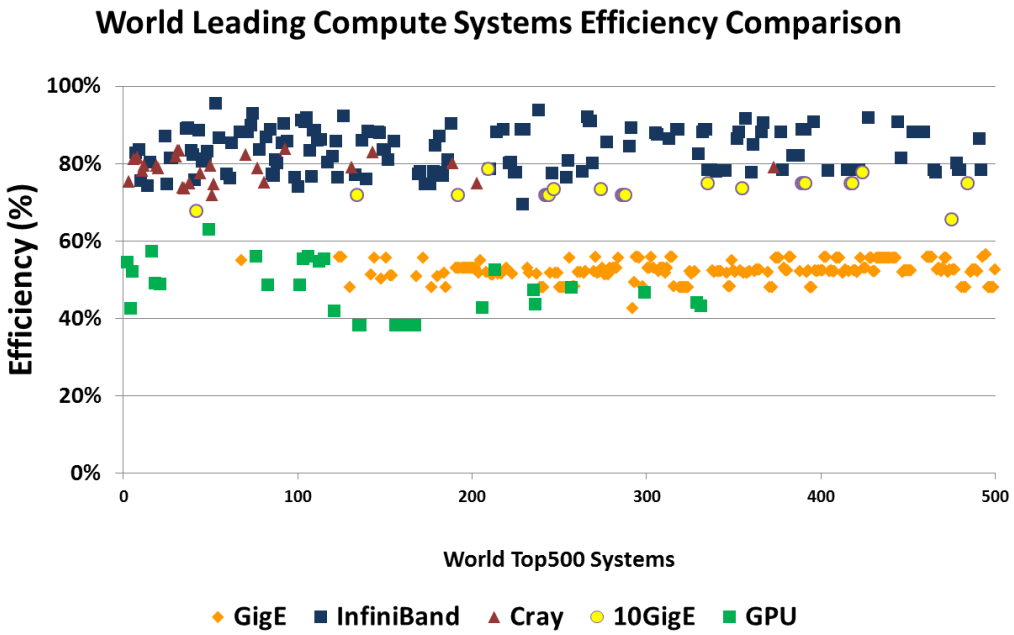

Figure 1 TOP500 System Efficiency

During the network intensive shuffle and sort section, Mapped data is sorted and shared among multiple nodes of the cluster, feeding data to the Reduce section. In a sockets-based architecture, the CPU is constantly occupied by data transfer during the shuffle task at the same time other Map tasks are still active and reducers are kicking-in additional loads. This intense section results in incomplete Map tasks and delayed Reducers, causing longer execution times and low CPU utilization. However, with RDMA-based programing, all data transfers during the shuffle section are handled by the fabric, freeing the CPUs to complete more Map and Reduce tasks without overruns.

Scalability and Performance

Customers add more compute nodes, expecting the new nodes to contribute the same computational power as their older counterparts. This method proved to be wrong when performed without intensive examination of load and data traffic balancing within the enlarged cluster.

HPC clusters, when scaled to thousands of nodes provide nearly linear scalability in performance. Scaling Big Data clusters should take into account similar parameters which are: oversubscription in the fabric, switch latency, ease of future expansion and management constraints.

It is economically tempting to heavily oversubscribe Top-of-Rack (ToR) switch uplinks, however, the result is congested switches. While some applications consider this a non-issue, Hadoop

Distributed File System (HDFS™) clogging can result in loss of data or significant performance drawbacks. Many Big Data applications are based on CAP theorem, where availability and partitioning are guaranteed. Clogged switches can result in misinterpretation of data when quorum is not satisfied in a timely manner.

Latency, The Short Way to Unlock SSD Performance

Just-A–Bunch-Of-Disks (JBOD) is commonly used in Big Data applications, JBODS are typically connected to the data node, providing fast data access for the local CPU. Still, to meet the latency requirements, multiple disks need to be connected in parallel, creating physical and power constraints. Although SSDs seems the right solution, customers claim that they can't justify the ROI and can't experience the performance promised by SSDs. The reason for that claim is the high latency inherent with legacy clusters.

Using low latency fabrics, with latency as low as 1.3us (application-to-application), SSD arrays are exposed as a single, fast storage entity. The SSD's fast data access and no seek delay time combined with the low latency fabric provides remote nodes performance similar to a local disk one. The HDFS™ replication process benefits from faster access and restoring capabilities in the likely case of node or rack failure, causing little to no penalty to application performance. Moreover, SSDs use significantly lower power, with 1TB SSD consuming ¼ of a similar size HDD.