Improving Performance of Big Data using BlueDBM

Gisha George
Assistant Professor
Department of Computer Science and Applications
St. Mary's College
Thrissur

***Abstract*—We are in the era of Big Data—a term that refers to the explosion of available information. Such a Big Data movement is driven by the fact that massive amounts of very high-dimensional or unstructured data are continuously produced and stored with much cheaper cost than they used to be. Complex analytics of the vast amount of data collected via social media, cell phones, ubiquitous smart sensors, and satellites is likely to be the biggest economic driver for the IT industry over the next decade. For many "Big Data" applications, the limiting factor in performance is often the transportation of large amount of data from hard disks to where it can be processed, i.e. DRAM. Here present a solution, BlueDBM, a novel Big Data flash storage platform that includes a dedicated low latency sideband storage network between flash controllers, reconfigurable fabric for implementing in-store hardware accelerators and a flash-aware file system.**

Keywords—FPGA; PCIe; DRAM; Flash memory.

I. INTRODUCTION

"Big data" originally meant the volume of data that could not be processed (efficiently) by traditional database methods and tools. The original definition focused on structured data, but most researchers and practitioners have come to realize that most of the world's information resides in massive, unstructured information, largely in the form of text and imagery. We define "big data" as the amount of data just beyond technology's capability to store, manage and process efficiently.

Big Data by definition doesn't fit in personal computers or DRAM of even moderate size clusters. Since the data may be stored on hard disks, latency and throughput of storage access is of primary concern. Here presents BlueDBM, a new system architecture which has flash based storage with in-store processing capability and a low latency high-throughput inter-controller network. BlueDBM presents an attractive point in the cost-performance trade-off for Big Data analytics.

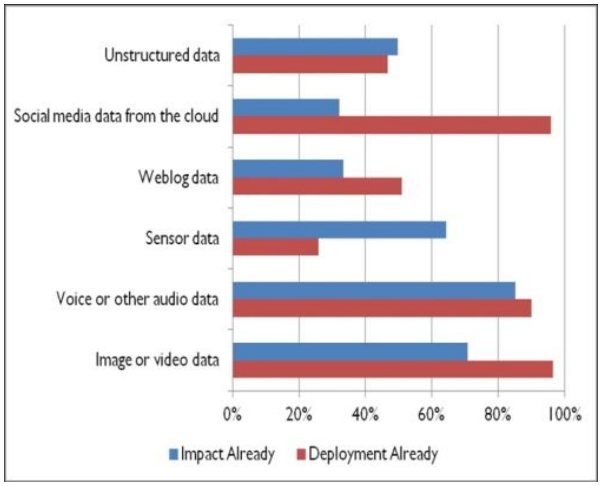

Fig. 1. Data Types Which Deliver the Most Positive Impact in a Big Data Context

There's a big problem with big data: the huge RAM memory required. Data sets in areas such as genomics, geological data, and daily twitter feeds can be as large as 5TB to 20 TB. Complex data queries in such data sets require high-speed random-access memory (RAM). But that would require a huge cluster with up to 100 servers, each with 128GB to 256gbs of DRAM (dynamic random access memory).

Flash memory (used in smart phones and other portable devices) could provide an alternative to conventional RAM for such applications. It's about a tenth as expensive, and it consumes about a tenth as much power. The problem: it's also a tenth as fast.

Flash storage systems perform better at tasks that involve finding random pieces of information from within a large dataset than other technologies. They can typically be randomly accessed in microseconds. This compares to the data "seek time" of hard disks, which is typically four to 12 milliseconds when accessing data from unpredictable locations on demand.

Flash systems also are nonvolatile, meaning they do not lose any of the information they hold if the computer is switched off.

II. INTODUCTION TO BLUEDBM

Many big-data applications require real-time or fast responses. For such applications, BlueDBM or Blue Database Machine is an appealing solution. Relative to some other proposals for streamlining big-data analysis, the main advantage of BlueDBM might be that it can easily scale up to a lot bigger storage system with specialized accelerated supports.

BlueDBM is a system that could enable flash-based servers to perform as well as RAM-based servers, but at less cost and using less energy.

This is not a replacement for dynamic RAM or similar, but there may be many applications that can take advantage of this new style of architecture. The high-level goal of BlueDBM is to provide a high-performance storage system that accelerates the processing of very large datasets.

The BlueDBM design aims to achieve the following goals:

A. Low Latency, High Bandwidth

To increase the performance of response-time sensitive applications, the network should add negligible latency to the overall system while maintaining high Bandwidth.

B. Scalability

Because Big Data problems are constantly increasing in size, the architecture should be scalable to higher capacity and node count.

C. Low-Latency Hardware Acceleration

In order to reduce data transport and alleviate computationally bound problems, the platform should provide very low-latency hardware acceleration.

D. Application Compatibility

As a general storage solution for Big Data, existing applications should run on top of our new storage hardware without any modification.

E. Multi-accessibility

In order to accommodate distributed data processing applications, the system should be capable of handling multiple simultaneous requests from many different users.

III. TECHNICAL DETAILS

In BlueDBM, each flash device is connected to a field-programmable gate array (FPGA) chip to create an individual node. The FPGAs are used not only to control the flash device, but are also capable of performing processing operations on the data itself.

The researchers were able to make a network of flash-based servers competitive with a network of RAM-based servers by moving a little computational power off of the servers and onto the chips that control the flash drives. By preprocessing some of the data on the flash drives before passing it back to the servers, those chips can make distributed computation much more efficient. And since the preprocessing algorithms are wired into the chips, they dispense with the computational overhead associated with running an operating system, maintaining a file system, and the like.

With hardware contributed by some of their sponsors — Quanta, Samsung, and Xilinx — the researchers built a prototype network of 20 servers. Each server was connected to a field-programmable gate array, or FPGA, a kind of chip that can be reprogrammed to mimic different types of electrical circuits. Each FPGA, in turn, was connected to two half-terabyte — or 500-gigabyte — flash chips and to the two FPGAs nearest it in the server rack.

Because the FPGAs were connected to each other, they created a very fast network that allowed any server to retrieve data from any flash drive. They also controlled the flash drives, which is no simple task. The controllers that come with modern commercial flash drives have as many as eight different processors and a gigabyte of working memory.

Finally, the FPGAs also executed the algorithms that preprocessed the data stored on the flash drives. The researchers tested three such algorithms, geared to three popular big-data applications. One is image search, or trying to find matches for a sample image in a huge database. Another is an implementation of Google's PageRank algorithm, which assesses the importance of different Web pages that meet the same search criteria. And the third is an application called Memcached, which big, database-driven websites use to store frequently accessed information.

FPGAs are about one-tenth as fast as purpose-built chips with hardwired circuits, but they're much faster than central processing units using software to perform the same computations. Ordinarily, either they're used to prototype new designs, or they're used in niche products whose sales volumes are too small to warrant the high cost of manufacturing purpose-built chips.

And since FPGAs are reprogrammable, they could be loaded with different accelerators, depending on the application. That could lead to distributed processing systems that lose little versatility while providing major savings in energy and cost.

IV. SYSTEM ARCHITECTURE OF BLUEDBM

The BlueDBM architecture distributes high performance flash storage among computational nodes to provide a scalable, high-performance and cost-effective distributed storage. In order to achieve this, BlueDBM introduces a low-latency and high-speed network directly between the fash controllers. The direct connection between controllers not only reduces access latency by removing the network software stack, but also allows the flash controllers to mask the network latency within flash access latency. This result in sub-microsecond latency per network hop, which is negligible compared to the access latency of the flash chip and software stack. This means that the entire network of flash storage performs as if it were a single, large, uniform-latency storage device. Algorithms that use large heaps as a data structure should benefit from this fast random-access capability. Controller-to-controller latencies in such a

network can be insignifcant compared to flash access latencies, giving us the potential to expose enormous storage capacity and bandwidth with performance characteristics similar to a locally attached PCIe flash drive.To further improve the effectiveness of the storage system, BlueDBM includes a FPGA-based reconfigurable fabric for implementing hardware accelerators near storage.

FPGA chips can be linked together using a high-performance serial network, which has a very low latency, or time delay. So information from any of the nodes can be accessed within a few nanoseconds. If we connect all of our machines using this network, any node can access data from any other node with very little performance degradation. It will feel as if the remote data were sitting here locally. Using multiple nodes allows to get the same bandwidth and performance from their storage network as far more expensive machines.

The BlueDBM architecture is a homogeneous cluster of host servers coupled with a BlueDBM storage device (See Fig. 2). Each BlueDBM storage device is plugged into the host server via a PCIe link, and it consists of flash storage, an in-store processing engine, multiple high-speed network interfaces and on-board DRAM. The host servers are networked together using Ethernet or other general-purpose networking fabric. The host server can access the BlueDBM storage device via a host interface implemented over PCIe. It can either directly communicate with the flash interface, to treat is as a raw storage device, or with the in-store processor to perform computation on the data.

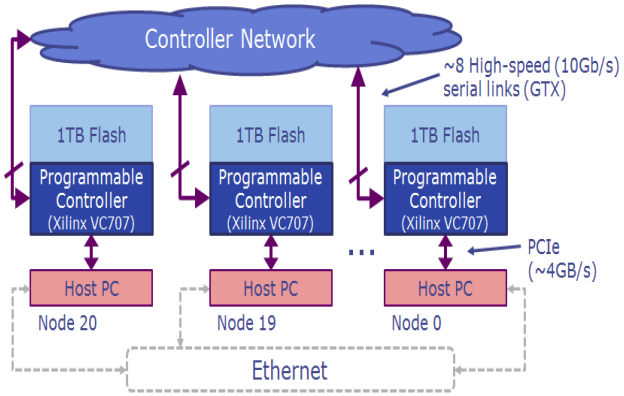

Fig. 2: BlueDBM overall architecture

The in-store processing engine has access to four major services: The flash interface, network interface, host interface and the on-storage DRAM buffer. Fig. 3 shows the four services available to the in-store processor in a node.

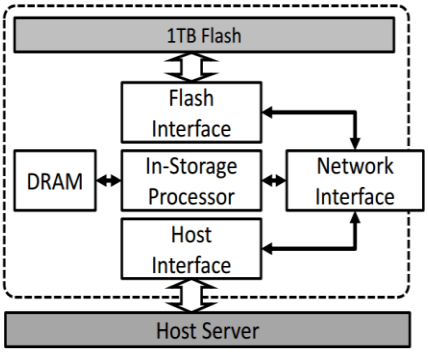

Fig. 3: BlueDBM Node Architecture

Each BlueDBM node consists of flash storage coupled with an FPGA, and is plugged into a host system's PCIe port. Each node is connected to up to 8 other BlueDBM nodes over a high-speed serial link capable of 10 gigabit bandwidth at 0.5 us latency. By default, the FPGA includes platform functions such as flash, network and on-board DRAM management, and exposes a high-level abstraction.

Raw page-access performance characteristic of BlueDBM, measured on our four-node prototype, is shown in the graph below (fig.4).

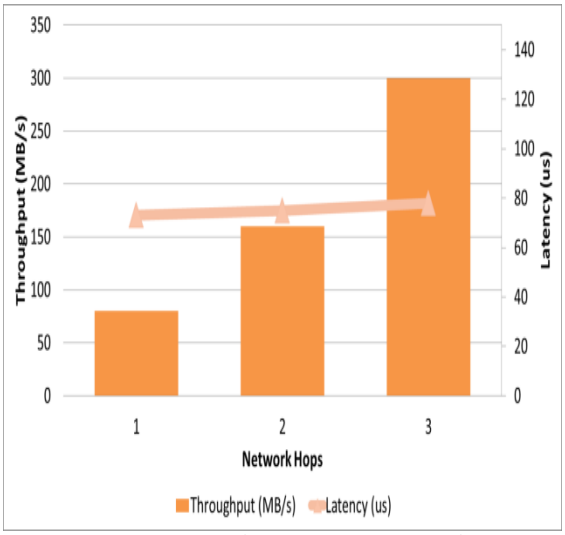

Fig.4. Raw page-access performance characteristic of BlueDBM

V. COMPARISON BETWEEN A POPULAR SOLUTION - RAM CLOUD AND FLASH-BASED SOLUTIONS

In RAM Cloud is having the advantage of Cluster of machines with large DRAM capacity and fast interconnect. But power consumption is more and expensive too. Its Performance will drop when data doesn't fit in DRAM.

Flash – based solutions are faster than Disk and cheaper than DRAM. It is having Low power consumption than both. But it is slower than DRAM.

VI. ILLUSTRATION OF IMAGE QUERY PERFORMANCE WITHOUT SAMPLING

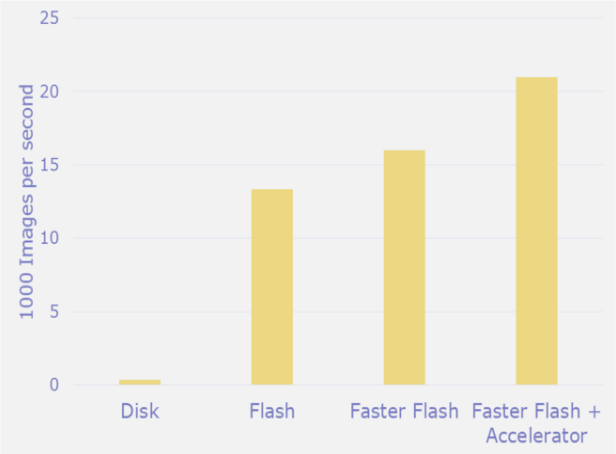

Fig. 5: Faster flash with acceleration can perform at DRAM speed.

VII. POWER CONSUMPTION

Table 1 shows the overall power consumption of the system, which were estimated using values from the datasheet. Each Xeon server includes 24 cores and 50GBs of DRAM. FPGA and flash devices are having low power consumption.

BlueDBM adds less than 20% of power consumption to the system.

TABLE I. POWER CONSUMPTION IS LOW

Storage device power consumption is a very conservative estimate	
Component	Power (Watts)
VC707	30
Flash Board (x2)	10
Storage Device Total	40

Component	Power (Watts)
Storage Device	40
Xeon Server	200+
Node Total	240+
GPU-based accelerator will double the power	

VIII. APPLICATIONS

There are multiple applications that will benefit from this architecture.

- Content-based Image Retrieval.
- Network-accelerated Flash-based Memcached.
- Scientific data analysis by accelerating SQL queries.
- Database for MATLAB (D4M).
- Column-oriented SQL database.
- Flash-based MapReduce platform.

REFERENCES

[1] Jianqing Fan ,Fang Han, and Han Liu , National science review, oxford Journal, "Challenges of Big Data analysis" October 15, 2013
[2] Stephen Kaisler,i_SW Corporation ,Frank Armour ,American
[3] University , J. Alberto Espinosa, American University ,William Money, George Washington University-"Big Data: Issues and Challenges Moving Forward", 2013 46th Hawaii International Conference on System Sciences
[4] Sang-Woo Jun, Ming Liu, Kermin Elliott Flemingy, Arvind, Department of Electrical Engineering and Computer Science , Massachusetts Institute of Technology, Cambridge, MA 02139 "Scalable Multi-Access Flash Store for Big Data "
[5] Prof. Arvind, Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology "BlueDBM: A Multi-access, Distributed Flash Store for Big Data Analytics",
[6] Larry Hardesty, MIT News Office- "Cutting cost and power consumption for big data", July 10, 2015
[7] Sang-Woo Jun,Ming Liu,Sungjin Lee,Jamey Hicks,John Ankcorn ,Myron King,Shuotao Xu,Arvind, "BlueDBM: An Appliance for Big Data Analytics" 2015
[8] Helen Knight –"Storage system for 'big data' dramatically speeds access to information" , February 1, 2014