WHITE PAPER

Accelerating Enterprise Search with Fusion ioMemory™ PCIe Application Accelerators

SanDisk®
a Western Digital brand

Western Digital Technologies, Inc.
951 SanDisk Drive, Milpitas, CA 95035

www.SanDisk.com

Table of Contents

1. Abstract

With enterprise data sets growing larger, the need to provide accessibility to that data is more important than ever. Analytics are important, but they are only a part of your overall data ecosystem. More and more, companies are finding the need to make their content searchable, and they are turning to enterprise search applications, such as Apache Lucene, to make that happen.

However, techniques that work on traditional data sets often fail when applied to today's big data problems. Large indexes can quickly overwhelm traditional search platforms, forcing users to invest in expensive scale-out architectures in order to meet their performance requirements. In this paper we describe how Fusion ioMemory PCIe solutions from SanDisk® can be used to dramatically increase search performance, without a massive investment in infrastructure.

2. Performance Considerations for Apache Lucene

Apache Lucene is an open source Java library, which provides full text search for a variety of different content types. It is managed by the Apache Software Foundation and released under a Creative Commons license. Because of its powerful text search capabilities, Lucene has been used to create a number of enterprise search engines, such as Solr, ElasticSearch, and Blur.

The hardware architectures for these applications have traditionally relied on locally attached hard drives to store their indexes. While this works well for smaller indexes, larger indexes can quickly overwhelm these drives, resulting in degraded performance. To mitigate this problem, search applications store as much of their indexes in memory as possible. This approach is not always sufficient, however, as larger indexes can quickly exceed the amount of available memory in the server, forcing indexes to be read from disk.

Search applications provide scalability by distributing the index across multiple servers in a cloud configuration. This makes it possible to store the entire index in memory, using the combined memory capacity of the cluster. However, the "RAM cloud" approach requires a significant investment in infrastructure and may be cost prohibitive for many organizations.

3. Solving the Scalability Problem

Traditional storage devices rely on disk controllers and RAID controllers for access. This imposes additional latency and overhead, as data is serialized, copied and re-copied through multiple layers of controllers and embedded processors. The Fusion ioMemory PCIe devices from SanDisk use a virtual memory architecture, whereby the CPU accesses the NAND directly, as though it were second tier of server memory. The Virtual Storage Layer (VSL) presents the device to the host operating system as a virtual disk, creating a storage device, which has the performance characteristics of memory. By storing Lucene's indexes on a Fusion ioMemory device rather than traditional memory, it is possible to support much larger data sets, allowing users to expand their search capabilities without a massive investment in infrastructure.

4. Performance Testing

To evaluate the performance characteristics of this approach, a test system was built with the following configuration1:

- Dual Intel® Xeon® E5-2600 Processors
- 64 GB memory
- 6x 1TB 7200 RPM HDDs in RAID 0
- 6x ioScale 1.6 TB PCIe card in RAID 0
- CentOS 6.4
- Apache Solr 5.0

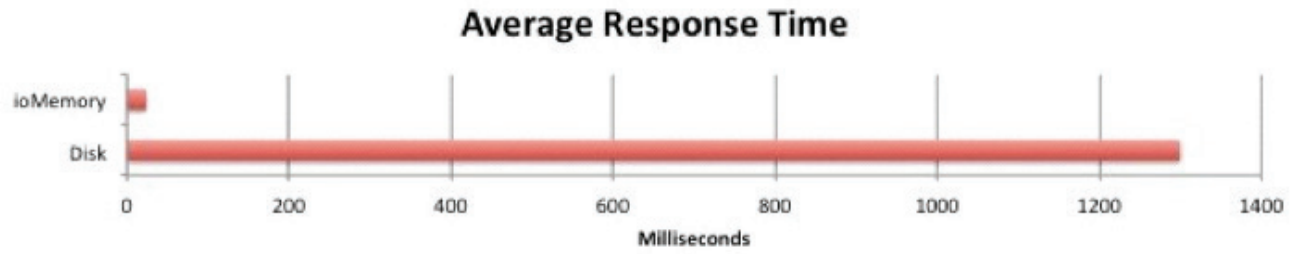
Two instances of Apache Solr were created, using the default configuration. The first was installed on the HDD array, while the second was stored on an array of six Fusion ioMemory devices. An identical index was loaded onto each instance, and several tests were conducted in order to compare their performance.

The data set used for testing was derived from Wikipedia's database, which is available for download under the Creative Commons license. A single database dump contains roughly 50 GB of data. This data set is not large enough for a meaningful test on a system with 64 GB of memory, as the majority of data would be cached in memory. Therefore, the test data set consisted of six Wikipedia dumps from six different months, totaling 246 GB, and creating an index of 202 GB.

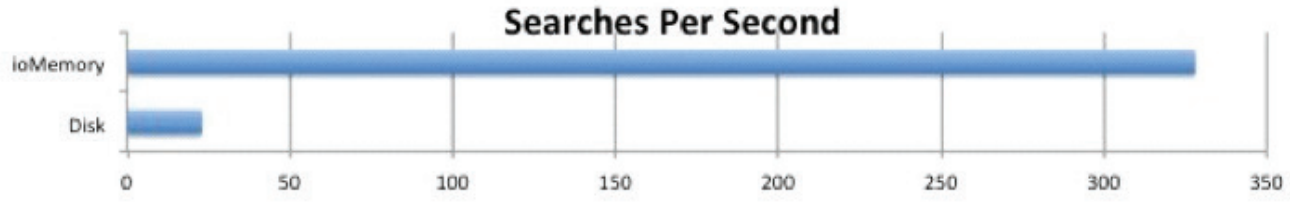
At the time of this writing, there were no benchmarking tools available for Apache Solr 5.0, so a custom benchmarking tool was written. The test tool generated random search terms, consisting of one, two, or three dictionary words. The tool spawned multiple threads, which submitted search requests using Solr's RESTful interface. Before each test, the OS buffer cache was cleared so as not to skew the results. The test was run for a total of 120 minutes, and statistics were collected on the number of searches performed and the latency of each request.

5. Test Results

Comparing the HDD-based solution to the Fusion ioMemory flash-based solution the results were as follows:



The average response time on disk was 1.3 seconds, with many request taking 30 seconds or more. The Fusion ioMemory based instance showed an average response time of 24 milliseconds, with sub-second responses for every request. Overall, the Fusion ioMemory-based instance showed a 54x improvement in response times over the disk-based system.



In the overall performance test, the Fusion ioMemory-based system was able to service 328 searches per second, as opposed to 23 searches per second on disk. The overall performance improvement was more than 14x

6. Conclusion

The Fusion ioMemory-based solution showed a significant performance improvement over disk. It was able to maintain sub-second response times with the larger index, while the disk-based solution experienced degraded performance.

Overall, the Fusion ioMemory-powered system offers a much more cost effective solution to scaling with DRAM, delivering more than ten times the performance of disk within a single server.