CASE STUDY

Baidu Queries Data 30 Times Faster with Alluxio

As the largest Chinese language Internet search provider, Baidu is very experienced with stressing their production data serving systems. In this case study, Shaoshan Liu -- senior architect at Baidu -- shares his experiences with Alluxio in production, and how the technology has led to dramatic performance gains. With Alluxio, batch queries are transformed into interactive queries. This enables Baidu to discover insights interactively leading to increases in productivity by 10 fold and improvements in customer experience.

THE BUSINESS CHALLENGE

Baidu is the biggest search engine in China, put simply — we have a lot of data. How to manage the scale of this data, and quickly extract meaningful information, has always been a challenge.

For example, due to the sheer volume of data, queries would take tens of minutes, to hours, just to finish — leaving product managers waiting hours before they could enter the next query. Even more frustrating was that modifying a query would require running the whole process all over again. About a year ago, we realized the need for an ad-hoc query engine. To get started, we came up with a high-level specification: the query engine would need to manage petabytes of data and finish 95% of queries within 30 seconds.

We switched to Spark SQL as our query engine (many use cases have demonstrated its superiority over Hadoop Map Reduce in terms of latency). We were excited and expected Spark SQL to drop the average query time to within a few minutes. However, it did not achieve the query response times we were hoping for. While Spark SQL did help us achieve a 4 time speed up of our average query, queries still took around 10 minutes to complete.

So, we took a second look and dug into more details. It turned out that the issue was not CPU — rather, the queries were stressing the network. Since the data was distributed over multiple data centers, it was highly likely that a query would need to transfer data from a remote data center to the compute data center — this is what caused the biggest delay when a user ran a query. Since the storage data center nodes and the compute data center nodes had different optimal hardware specifications, the answer was not as simple as moving the computation to the storage data center. We needed an in-memory storage system that would store the frequently used “hot” data, and would be local to the compute nodes.

WHY ALLUXIO

We needed an in-memory storage system that could provide high performance and reliability, and manage petabytes of data. We developed a query system that used Spark SQL as its compute engine, and decided to use Alluxio as the local in-memory storage solution, and we stress tested for a month. For our testing, we used a standard query within Baidu, which pulled 6TB of data from a remote data center, and then ran additional analysis on top of the data.

Alluxio enabled extraordinary performance. With Spark SQL alone, the query took 100-150 seconds to complete; with Alluxio, the query took 10-15 seconds. Additionally, if all of the data was stored in Alluxio local nodes, it took about 5 seconds, flat — 30 times faster than Spark SQL alone. Based on these results, and the system’s reliability, we built a full system around Alluxio and Spark SQL.

Our system consists of the following components:

- Operation Manager: A persistent Spark application that wraps Spark SQL. It accepts queries from query UI, and performs query parsing and optimization.
- View Manager: Manages cache metadata and handles query requests from the operation manager.
- Alluxio: Serves as a compute-local in-memory storage system that stores the frequently used data.
- Data Warehouse: The remote data center that stores the data in HDFS-based systems.

Now, let’s discuss the physiology of the system:

1. A query gets submitted. The operation manager analyzes the query and asks the view manager if the data is already in Alluxio.
2. If the data is already in Alluxio, the operation manager grabs the data from Alluxio and performs the analysis on it.
3. If data is not in Alluxio, then it is a cache miss, and the operation manager requests data directly from the data warehouse. Meanwhile, the view manager initiates another job to request the same data from the data warehouse and stores the data in Alluxio. This way, the next time the same query gets submitted, the data is already in Alluxio.

REALIZED BENEFITS

With the system deployed, we measured its performance using a typical Baidu query. Using the original Hive system, it took more than 1,000 seconds to finish a typical query. With the Spark SQL-only system, it took 150 seconds, and with Alluxio, it took about 20 seconds. The query ran 50 times faster and met the interactive query requirements we set out for the project. Therefore, by using Alluxio, a batch query lasting over 15 minutes was transformed into an interactive query taking less than 30 seconds.

In the past year, the system has been deployed in a cluster with more than 100 nodes, providing more than two petabytes of Alluxio-managed storage, using an advanced feature (tiered storage) in Alluxio. This feature allows us to use memory as the top-level tier, SSD as the second-level tier, and HDD as the last-level tier; with all of these storage mediums combined, we are able to provide over two petabytes of storage space.

Besides performance improvement, what is more important to us is reliability. Over the past year, Alluxio has been running stably within our data infrastructure and we have rarely seen problems with it. This gave us a lot of confidence. Therefore, we are preparing for larger scale deployment of Alluxio. To start, we verified the scalability of Alluxio by deploying a cluster with 1000 Alluxio workers. In the past month, this 1000-worker Alluxio cluster has been running stably, providing over 50 TB of RAM space. As far as we know, this is currently one of the largest Alluxio clusters in the world.

CONCLUSIONS

We have verified that Alluxio greatly improves performance, it is reliable, and it is scalable. For the next steps, we are gradually migrating different Baidu workloads onto our Alluxio clusters. For example, to improve the performance of online image serving and offline image analysis, we are working closely with the community on developing a high-performance Key-Value store on top of Alluxio. This way, only one storage system, in this case, Alluxio, is needed: the Key-Value store can perform efficient online serving; and for offline analysis we could directly access Alluxio for image data. This greatly reduces our development and operation costs.

As an early adopter of Alluxio, I can testify that it lives up to its description as “a memory-centric distributed storage system, enabling reliable data sharing at memory-speed, across cluster frameworks.” Besides being reliable and having memory-speed, Alluxio also provides a means to expand beyond memory to provide enough storage capacity.

ABOUT BAIDU

Baidu, Inc. is the leading Chinese language Internet search provider. As a technology-based company, Baidu aims to provide the best and most equitable way for people to find what they're looking for. In addition to serving individual Internet search users, Baidu provides an effective platform for businesses to reach potential customers.

Baidu USA is home to Baidu USDC and Baidu Research. Baidu USDC develops Internet-related business opportunities and advanced technologies in areas such as advertising, security, big data and cloud computing. Baidu Research focuses on fundamental technologies in areas such as image recognition and image-based search, voice recognition, natural language processing and semantic intelligence.