**ALLUXIO**

**PROJECT ANALYSIS**

**Introduction:**

Alluxio is the world's first open source data orchestration technology for analytics and AI for the cloud. It bridges the gap between data driven applications and storage systems, bringing data from the storage tier closer to the data driven applications and makes it easily accessible enabling applications to connect to numerous storage systems through a common interface. Alluxio’s memory-first tiered architecture enables data access at speeds orders of magnitude faster than existing solutions.

**Project Summary**

| **Website** | https://www.alluxio.io/ |
| --- | --- |
| **Organization/Foundation Name** | Alluxio |
| **Licence** | Apache licence 2.0 |
| **Open/Proprietary** | open |
| **Source Path(if open source)** |  |
| **Brief Description** | The Alluxio project originated from the BCberkeley AMPLab (see papers) as the data access layer of the Berkeley Data Analytics Stack (BDAS). It is open source under Apache License 2.0. Alluxio is one of the fastest growing open source projects that has attracted more than 1000 contributors from over 300 institutions including Alibaba, Alluxio, Baidu, CMU, Google, IBM, Intel, NJU, Red Hat, Tencent, UC Berkeley, and Yahoo. |

## **PROJECT DETAILS:**

## **Key Features:**

### **1.Compute-Focused**

* **Support for hyperscale workloads-**support of files and thousands of workers and client, all with high availability
* **Flexible APIs-**Integrates your compute frameworks like Spark, Presto, Tensorflow, Hive and more out-of-the-box using the HDFS, S3, Java, RESTful, or POSIX-based APIs.
* **Intelligent data caching and tiering-** Automatically utilises near-compute storage media for optimal data placement based on data topology and workload.

**2.Storage-Focused**

* **Built-in data policies-** Provides highly customizable data policies for persistence, cross storage data migration, and distributed load.
* **Plug and play under stores-** Integrates your under store systems like HDFS, S3, Azure Blob Store, Google Cloud Store and more through a range of interfaces.
* **Transparent unified namespace for file system and object stores-** Mounts multiple storage systems into a single consolidated namespace for both read and write workloads.

**3.Enterprise-Ready**

* **Security-** Provides data protection on the wire and in the cloud with built-in auditing, role-based access control, LDAP, active directory, and encrypted communications.
* **Monitoring and Management-** Provides a user-friendly web interface and command line tools, allowing users to monitor and manage their cluster.
* **Enterprise high availability with tiered locality-** Includes adaptive replication across regions and zones to maximise performance and availability.

## **Architecture Diagram**

Alluxio serves as a new data access layer in the big data and machine learning ecosystem. It resides between any persistent storage systems such as Amazon S3, Microsoft Azure Object Store, Apache HDFS, or OpenStack Swift, and computation frameworks such as Apache Spark, Presto, or Hadoop MapReduce. Note that Alluxio itself is not a persistent storage system. Using Alluxio as the data access layer provides multiple benefits:

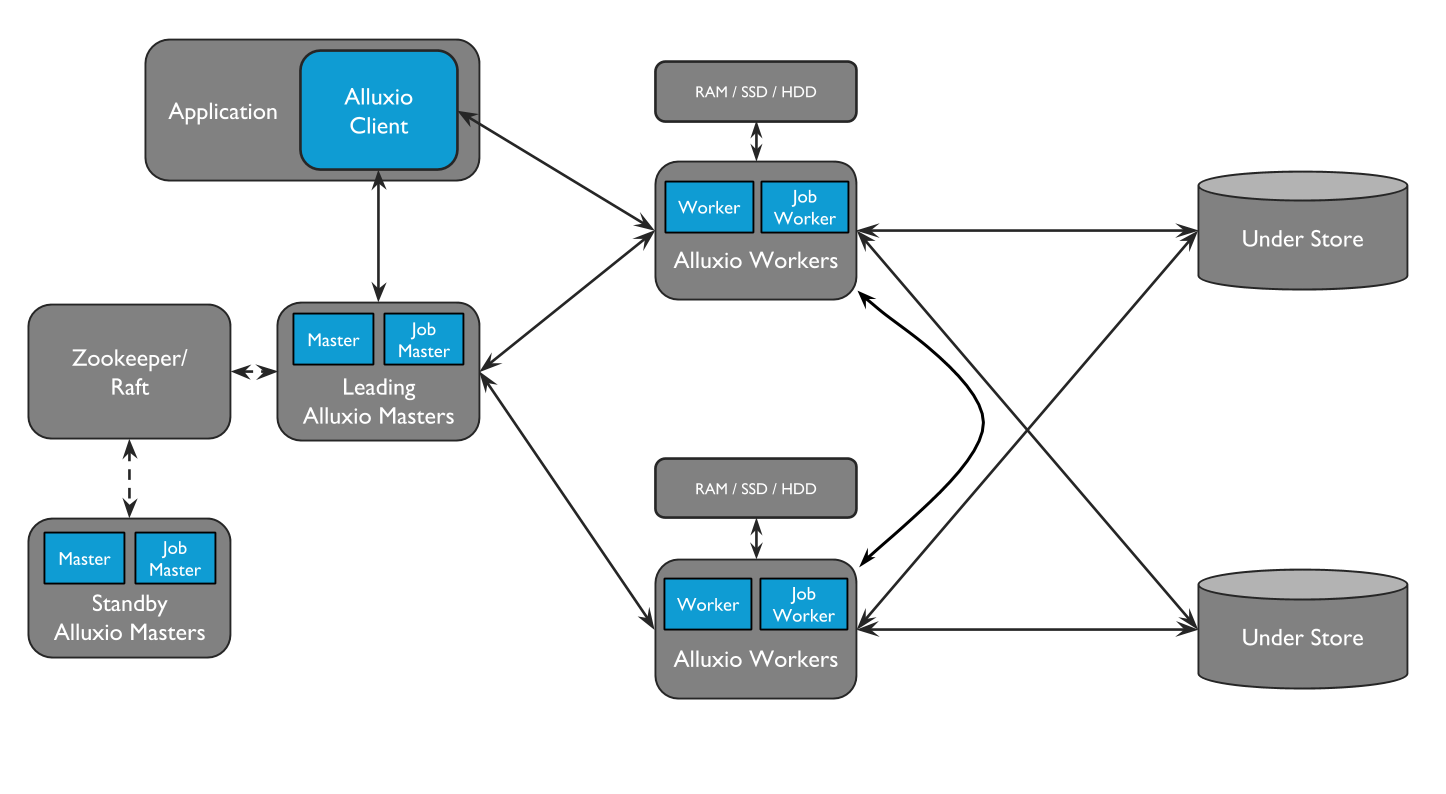
* For user applications and computation frameworks, Alluxio provides fast storage, and facilitates data sharing and locality between applications, regardless of the computation engine used. As a result, Alluxio can serve data at memory speed when it is local or at the speed of the computation cluster network when data is in Alluxio. Data is only read once from the under storage system when accessed for the first time. Data access is significantly accelerated when access to the under storage is slow. To achieve the best performance, Alluxio is recommended to be deployed alongside a cluster’s computation framework.
* For under storage systems, Alluxio bridges the gap between big data applications and traditional storage systems, expanding the set of workloads available to utilize the data. Since Alluxio hides the integration of under storage systems from applications, any under storage can support any of the applications and frameworks running on top of Alluxio. When mounting multiple under storage systems simultaneously, Alluxio serves as a unifying layer for any number of varied data sources.

Alluxio can be divided into three components: masters, workers, and clients. A typical cluster consists of a single leading master, standby masters, a job master, standby job masters, workers, and job workers. The master and worker processes constitute the Alluxio servers, which are the components a system administrator would maintain. Clients are used to communicate with the Alluxio servers by applications such as Spark or MapReduce jobs, the Alluxio CLI, or the Alluxio FUSE layer.

Alluxio Job Masters and Job Workers can be separated into a separate function which is termed the Job Service. The Alluxio Job Service is a lightweight task scheduling framework responsible for assigning a number of different types of operations to Job Workers. Those tasks include

* Loading data into Alluxio from a UFS
* Persisting data to a UFS
* Replicating files within Alluxio
* Moving or copying data between UFS or Alluxio locations

The job service is designed so that all job related processes don’t necessarily need to be located with the rest of the Alluxio cluster. However, we do recommend that job workers are co-located with a corresponding Alluxio worker as it provides lower latency for RPCs and data transfer.



# **CURRENT USAGE:**

# SEE ALL ALLUXIO PARTNERS ACROSS THE ECOSYSTEM

Alluxio partners with leading technology companies and systems integrators. Find a partner that meets your needs.

* Kyligence Cloud
* H2O.ai
* Hitachi
* Huawei
* Dell EMC
* Cloudera
* Microsoft

Etc…….

# **TECHNICAL DETAILS:**

Alluxio brings three key areas of innovation together to provide a unique set of capabilities.

1. **Global Namespace:** Alluxio serves as a single point of access to multiple independent storage systems regardless of physical location. This provides a unified view of all data sources and a standard interface for applications. See Namespace Management for more details.
2. **Intelligent Multi-tiering Caching:** Alluxio clusters act as a read and write cache for data in connected storage systems. Configurable policies automatically optimize data placement for performance and reliability across both memory and disk (SSD/HDD). Caching is transparent to the user and uses buffering to maintain consistency with persistent storage. See Alluxio Storage Management for more details.
3. **Server-Side API Translation:** Alluxio supports industry common APIs, such as HDFS API, S3 API, FUSE API, REST API. It transparently converts from a standard client-side interface to any storage interface. Alluxio manages communication between applications and file or object storage, eliminating the need for complex system configuration and management. File data can look like object data and vice versa.

END