Big Data Architecture Design

Accurate Advertising

---- Based on Alibaba's taobao.com

Professor: Kambiz Heydari
Student: Xuan Gao
NUID: 001220084

Table of Contents

Overview

Taobao.com is an online shopping website belongs to Alibaba company. Advertisements and recommendations shown on the right moment will attractive people to buy things, and we can earn a lot of extra money through this. Also, the algorithm for accurate advertising values a lot. This is a new profit channel for our company. This document is a design of big data architecture for this project.

Requirements

Firstly, we need to consider the goals and the entities:

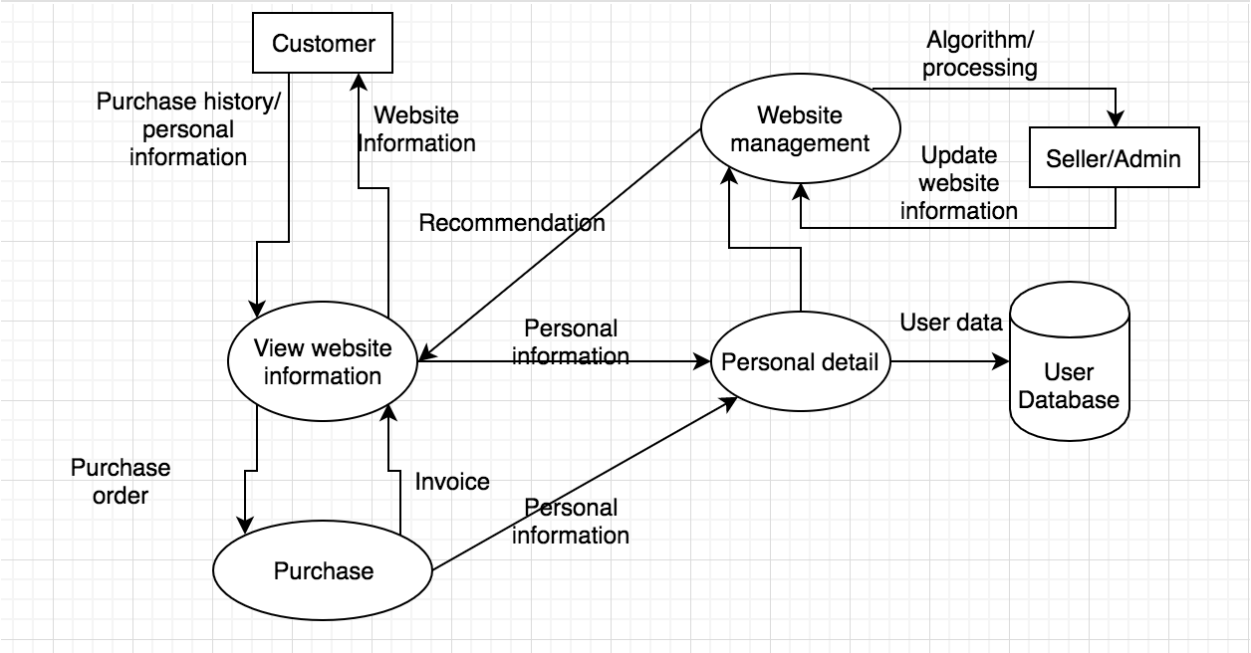
- ✓ Element: customer, platform (taobao.com), seller
- ✓ Connection: search, advertising, algorithm
- ✓ Target: High accuracy recommendation

Secondly, we need to make sure the requirements.

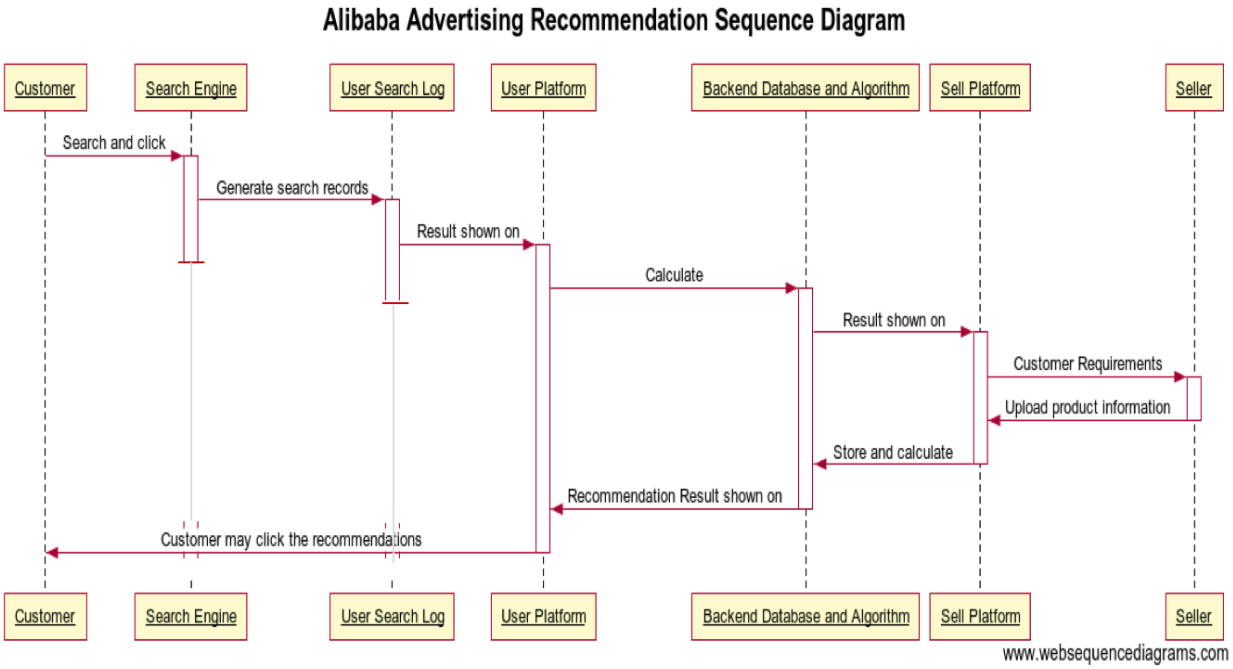
- ✓ So many data. The goal is to process 10 billion messages per day. Assuming that the record size is 2KB, we need 70PB physical capacity to accommodate these data. According to the requirement of query range, it is deduced that the computation time of off-line computation is 24 hours, and online computation is less than 10 minutes.
- ✓ Support. Big data technology is changing rapidly, hoping that the version of components can be updated in time.
- ✓ Cost. According to the SWOT, the external business environment is changing rapidly, hoping that computing resources can be dynamically increased or reduced to save costs.
- ✓ Security. Hope to get professional safety services at a lower cost.
- ✓ Open source
- ✓ Real time calculation and online analysis

All the selection of tools and technologies should be based on the requirements we need, and satisfy the volume, velocity, variety, veracity principles of big data.

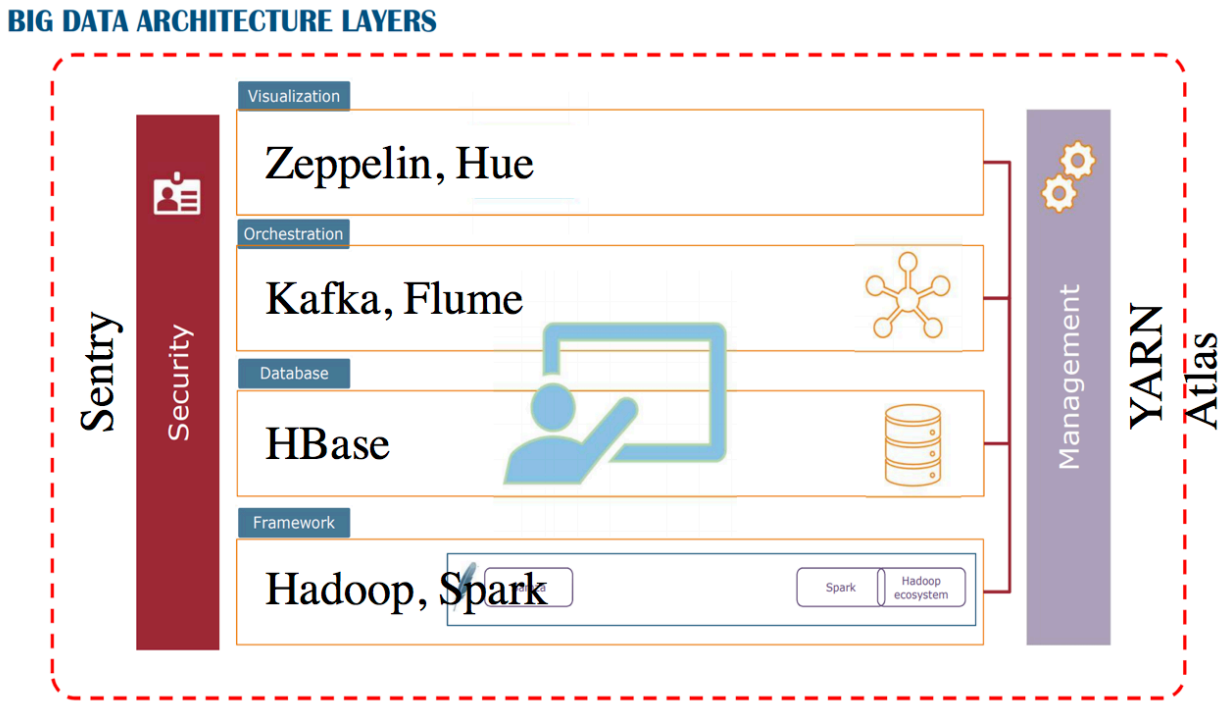
Vision diagram



Sequence diagram



Big data architecture layers



Visualization

Zeppelin

Description

A web-based notebook that brings data ingestion, data exploration, visualization, sharing and collaboration features to Hadoop and Spark. Large scale data analysis workflow includes multiple steps like data acquisition, pre-processing, visualization, etc and may include inter-operation of multiple different tools and technologies. With the widespread of the open source general-purpose data processing systems like Spark there is a lack of open source, modern user-friendly tools that combine strengths of interpreted language for data analysis with new in-browser visualization libraries and collaborative capabilities.

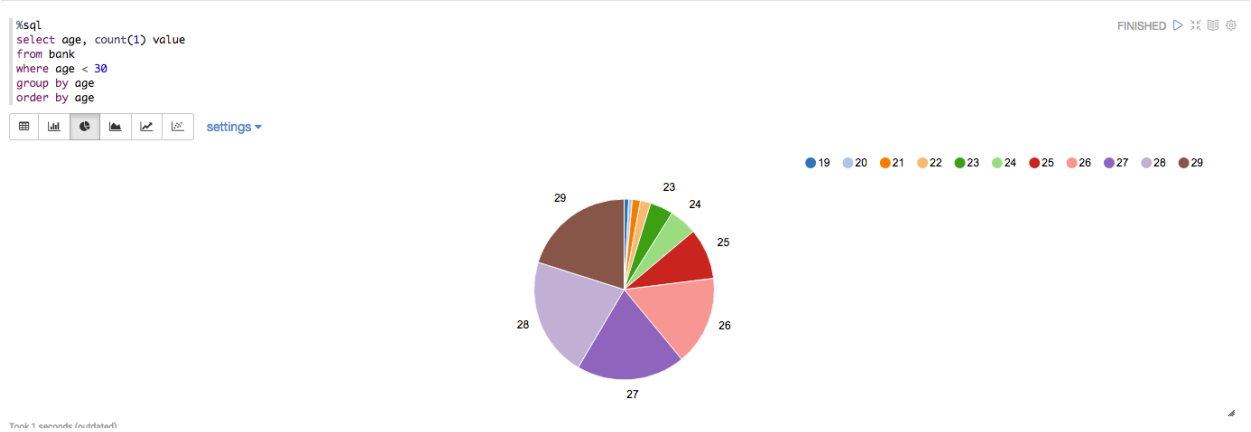
Why use Zeppelin

Open source, Support the frame and orchestration I use (Spark & Hadoop), multiple support, good visualization, multiple language backend.

Through it, business people can use HiveQL, SparkSQL, Phoenix, Presto and other inquiring and interactive queries in the form of Web, without programming and log in SSH, and can save past queries, and can also form simple histogram pie charts. Our DMP engineers no longer need to write code for a statistical number all night, so the business people can do it themselves.

How to use Zeppelin

- ✓ Create a cluster.
- ✓ Get through SSH with no secret login and build a SSH tunnel.
- ✓ Support markdown syntax.
- ✓ Native support Scala. Using Scala for load data.
- ✓ Use spark SQL search and show results.



Hue

Hue is an open source Apache Hadoop UI system, which was first evolved from Cloudera Desktop and contributed by Cloudera to the open source community. It is based on Python Web framework Django. By using Hue, we can interact with the Hadoop cluster on the browser - side Web console to analyze processing data, such as operating data on HDFS, running MapReduce Job and so on.

Language

Apache Hive

An easy way to process large scale data.
Support SQL-based queries.
Provide more user defined interfaces to extend.
Programmability.
Efficient execution plans for performance.
Interoperability with other database tools.

Apache Pig

Apache Pig is an abstraction over MapReduce. It is a tool/platform which is used to analyze larger sets of data representing them as data flows. Pig is generally used with Hadoop; we can perform all the data manipulation operations in Hadoop using Apache Pig.

To write data analysis programs, Pig provides a high-level language known as Pig Latin. This language provides various operators using which programmers can develop their own functions for reading, writing, and processing data.

To analyze data using Apache Pig, programmers need to write scripts using Pig Latin language. All these scripts are internally converted to Map and Reduce tasks. Apache Pig has a component known as Pig Engine that accepts the Pig Latin scripts as input and converts those scripts into MapReduce jobs.

Apache Impala

Impala raises the bar for SQL query performance on Apache Hadoop while retaining a familiar user experience. With Impala, you can query data, whether stored in HDFS or Apache HBase – including SELECT, JOIN, and aggregate functions – in real time. Furthermore, Impala uses the same metadata, SQL syntax (Hive SQL), ODBC driver, and user interface (Hue Beeswax) as Apache Hive, providing a familiar and unified platform for batch-oriented or real-time queries. (For that reason, Hive users can utilize Impala with little setup overhead.)

Orchestration & Framework

Hadoop HDFS

Description

Hadoop is an open source, Java-based programming framework designed to deliver a distributed file system (HDFS) that supports the processing and storage of extremely large data sets in a distributed computing environment. It is part of the Apache project sponsored by the Apache Software Foundation.

Why use Hadoop

Scalable, Cost effective, Flexible, Fast, Resilient to failure.

HDFS distributed file system, storage is the foundation of big data technology. It can process a rich supply of data and simple, convenient data achieve. It can be built on cheap machines.

Apache Hadoop and its MapReduce processing engine provide a set of tested batch models that are most suitable for handling very large datasets with less time requirements. With a very low cost component, a Hadoop cluster of complete functions can be built, making this cheap and efficient processing technology flexible in many cases. With the compatibility and integration capabilities of other frameworks and engines, Hadoop can become the underlying foundation of various workload processing platforms that use different technologies.

How to use Hadoop

- ✓ Install Hadoop
  Configuring SSH without password login, all hosts in the cluster should be able to log in without password SSH.
- ✓ Install Java JDK
  Install JDK on master, and control the JDK on the slaves remotely.
- ✓ Install Hadoop tar and start the test cluster
- ✓ Format HDFS, turn off the firewall and start the cluster

Spark

Description

Apache Spark is an open source cluster computing framework. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance. It is built around speed, ease of use, and sophisticated analytics.

Spark comes with GraphX, a distributed graph system. For Big Data Architecture & Management Class Use Only – Should not copy or duplicate. Spark is ideal for iterative processing, interactive processing and event stream processing. Spark can run on Hadoop alongside other tools in the Hadoop ecosystem including Hive and Pig.
It is very flexible and powerful.

Why use Spark

The main reason for using Spark instead of Hadoop MapReduce is speed. With the help of memory computing strategy and advanced DAG scheduling mechanism, Spark can process the same data set faster.

Another important advantage of Spark is diversity. The product can be deployed as an independent cluster or integrated with existing Hadoop clusters. The product can run batch and stream processing, and run a cluster to handle different types of tasks.

In addition to the ability of the engine itself, an ecosystem of various libraries is built around Spark, which can provide better support for machine learning, interactive query, and other tasks. Compared with MapReduce, Spark task is known as easy to write, so it can greatly improve productivity.

Spark is the best choice for dealing with tasks with diverse workload. Spark batch processing capabilities provide an unmatched speed advantage at the expense of higher memory footprint.

How to use Spark

- ✓ Install Hadoop
- ✓ Install Scala
- ✓ Install Spark package and start the test cluster
- ✓ Format HDFS, turn off the firewall and start the cluster

Kafka

Description

Kafka is a distributed pub/sub and message queueing system that provides at-least once messaging guarantees (i.e. the system guarantees that no messages are lost, but in certain fault scenarios, a consumer might receive the same message more than once), and highly available partitions (i.e. a stream's partitions continue to be available even if a machine goes down).

Why use Kafka

- ✓ The message persistence capability is provided in a time complexity O (1). Even for data above TB level, which guarantees the access performance of constant time complexity.
- ✓ High throughput rate. Even on a very cheap business machine, single machine can support 100K message transmission per second. It is known that Kafka can produce about 250 thousand messages per second (50 MB), processing 550 thousand messages per second (110 MB).
- ✓ It supports message partition between Kafka Server, and ensures the sequential transmission of messages in every Partition.
- ✓ Distributed system, easy to expand. All producer, broker and consumer will be distributed. The machine can be extended without a stop.
- ✓ The state of message processing is maintained on the consumer side, rather than on the server side. It can be automatically balanced when it fails.
- ✓ It also supports off-line data processing and real-time data processing.

How to use Kafka

The overall architecture of Kafka is very simple. There are multiple producers, brokers and consumers. Producer, consumer implement the interface of Kafka registration, data is sent from producer to broker, broker takes the role of intermediate cache and distribution. The broker is distributed to the consumer in the system. The function of broker is similar to that of cache, namely the cache between active data and off-line processing system. The communication between client and server is based on simple and high-performance TCP protocol, which is independent of programming language.

Flume

Description

Apache Flume is a distributed, reliable, and available system for efficiently collecting, aggregating and moving large amounts of log data from many different sources to a centralized data store. The primary use case for Flume is as a logging system that gathers a set of log files on every machine in a cluster and aggregates them to a centralized persistent store such as the Hadoop Distributed File System (HDFS).

Why use flume

Flume is a distributed, highly available and reliable system. It can collect, move, and store different massive data, and store it into a data storage system. Lightweight and simple configuration, it is suitable for all kinds of log collection, and supports Failover and load balancing. It has a very rich component.

Database

HBase

Description

HBase is a data model that is similar to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault tolerance provided by the Hadoop File System (HDFS). It is a part of the Hadoop ecosystem that provides random real-time read/write access to data in the Hadoop File System. One can store the data in HDFS either directly or through HBase. Data consumer reads/accesses the data in HDFS randomly using HBase. HBase sits on top of the Hadoop File System and provides read and write access.

Why use HBase

HBase is a distributed column-oriented database built on top of the Hadoop file system. It is an open-source project and is horizontally scalable. HBase load accesses reach hundreds of GB/ seconds (write) and hundreds of GB/ seconds (read).

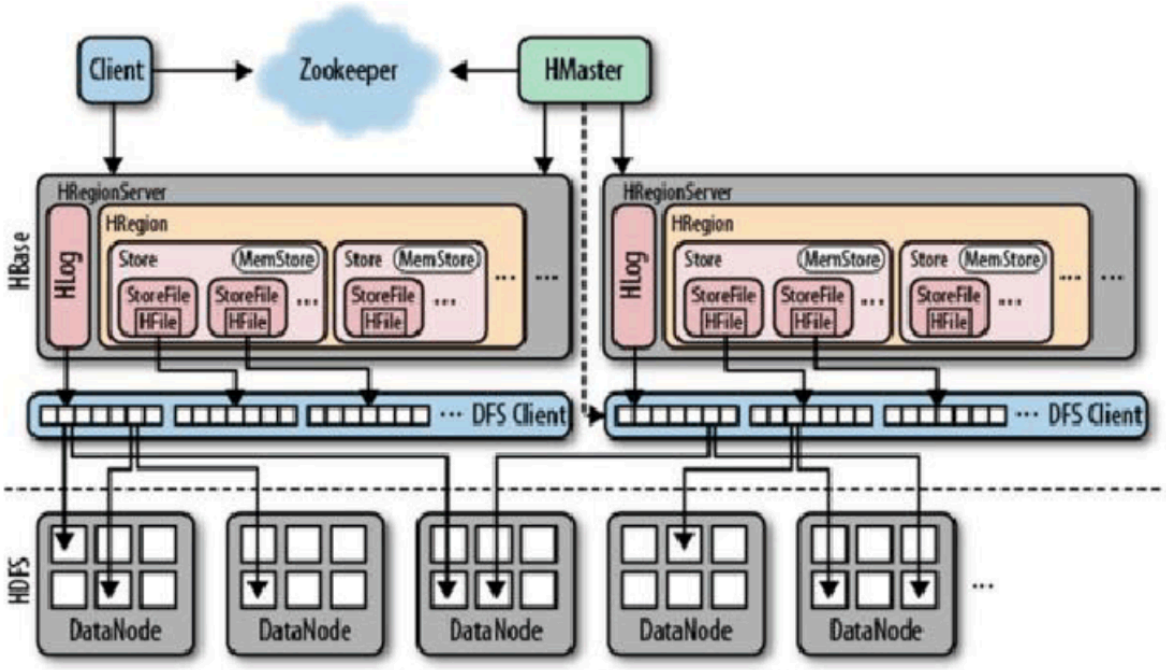
- ✓ High availability construction.
  The continuous availability of services is a significant feature of the Internet system, but because of the uncertainty of the physical environment and the software bug, it is often not easy to achieve the high availability of the system, especially for a state storage system. Today, we use a unified SLA (service level protocol) to measure the availability of a distributed system, such as a system with a SLA of 99.99%, whose unavailable time of the year is less than 52.6 minutes; 99.999% of the system, whose unavailable time is less than 5.25 minutes for the whole year, is generally referred to as a high availability system.

- ✓ Multi-link
  Multi-location and multi-cell deployment is an important feature of Alibaba's technology architecture, which requires basic storage to have flexible data link. HBase will deploy multiple clusters, and data flow between clusters will meet the needs of unit businesses.

- ✓ Consistency of data
  Today, most production systems use asynchronous way to achieve data replication between clusters, because it is more efficient and logical. This means that the data between clusters is a final consistent model. When the traffic is switched from the master to the standby, the complete data can not be accessed from the standby. Because the replication exists lag, and the data will be partially lost when the main cluster is permanently unrecoverable.

- ✓ Less redundancy and cost
  Redundant duplication of data in clusters brings a level of improvement to the availability of the system, but it also means a greater cost.

How to use HBase



Client: use HBase RPC to communicate HMaster and HRegionServer.
Zookeeper: collaborative service management. HMaster can perceive all HRegionServer health status at any time through Zookeepe.
HMaster: manage user to add, delete and modify the table
HRegionServer: the core module in HBase. It is mainly responsible for reading and writing data to the HDFS file system in response to user I/O requests.
HRegion: the smallest unit of distributed storage in Hbase. It can be understood as a Table.

HStore: the core of HBase storage. It is made up of MemStore and StoreFile.
HLog: every time user operation writes to Memstore, it also writes data to HLog file.

Security

Sentry

Description

Apache Sentry, a Hadoop open source component released by Cloudera, is currently an incubator for Apache, which provides a fine-grained, role-based authorization, and multi-tenant management. Sentry can now integrate with Hive/Hcatalog, Apache Solr and Cloudera Impala, and will expand to other Hadoop components, such as HDFS and HBase in the future.

Why use Sentry

- ✓ Security authorization
  Sentry can control data access and provide data access privileges to authenticated users.

- ✓ Fine grained access control
  Sentry supports fine grained Hadoop data and metadata access control. In the initial release of Sentry in Hive and Impala, Sentry provides access control at different privileged levels in the server, database, table, and view range, including lookup, insert, and so on, allowing administrators to use view restrictions to restrict access to rows or columns. Administrators can also shield data in files by using Sentry and view or UDF with selection statements.

- ✓ Role-based management
  Sentry simplifies management through role-based authorization, and you can easily grant multiple groups to different privileges at the same data set.

- ✓ Multi-tenant management
  Sentry allows permission to delegate different data sets to different administrators. In the case of Hive/Impala, Sentry can perform privilege management at the level of database /schema.

- ✓ Unified platform
  Sentry provides a unified platform for ensuring data security, and implements the security authentication using the existing Hadoop Kerberos. At the same time, the same Sentry protocol can be used when accessing data through Hive or Impala. In the future, Sentry protocol will be extended to other components.

How to use Sentry

The goal of Apache Sentry is to achieve authorization management. It is a policy engine, which is used to verify access rights by data processing tools. It is also a highly extended module that can support any data model. Currently, it supports the relational data model of Apache Hive and Cloudera Impala, as well as the inheritance data model in Apache.

Sentry provides a way to define and persist policies to access resources. At present, these policies can be stored in files or database backend stores that can be accessed by RPC services. Data access tools, such as Hive, identify users' requests for accessing data in a certain mode, such as reading a row of data from a table or deleting a table. This tool asks Sentry to verify whether access is reasonable. Sentry builds the mapping of requesting permission for users and determines whether a given request allows access. The request tool will then allow or prohibit user access requests based on Sentry's judgement results.

Management

YARN

YARN (Yet Another Resource Negotiator) is Hadoop's next-generation cluster scheduler. It allows you to allocate a number of containers (processes) in a cluster of machines, and execute arbitrary commands on them.

It has low Latency - if you need a very low latency system that can get you responses in milliseconds or few seconds, then storm is preferred. There is also a good pro for Storm, that it lets you manage your data flow in the system in a visual topology map.

Atlas

This is an Alibaba open source container framework. It has three main characters.

- ✓ High reusability and requires uniform control. For all middleware and all services, one party needs to have unified control.
- ✓ Low intrusion. For developers, they don't need to be aware of the container needs, only need to code, which is low intrusive, that is, developers do not need a lot of black technology. For example, that developers do not need to understand the underlying design of the Android system. And only API that uses Java layer on Android container can accomplish the functions that we want to achieve.
- ✓ High availability ensures that plug-ins can't cause too much loss to the performance of applications.

Conclusion

This is a real big project for Alibaba Company. I considered the data quantity, support, cost, security and selected some tools for my project. Actually, all the technologies have pros and cons and we should let them work together to satisfy our needs. I introduced some primary tools in detail. Through this process, I learned more about big data architecture and found there are too many tools for us to use. There are also lots of tools die out with the development of big data. This architecture design is a good fit for present Alibaba, but decades or several years later, there should be a better design.

Reference

1. “Big Data Architecture HandBook V4”
2. Alibaba Forum: https://yq.aliyun.com/
3. Zeppelin: https://yq.aliyun.com/articles/42282
4. HBase: https://yq.aliyun.com/articles/70467
5. Wiki and Google website