

YigFS

A distributed file system

zouy26@chinaunicom.cn

Background

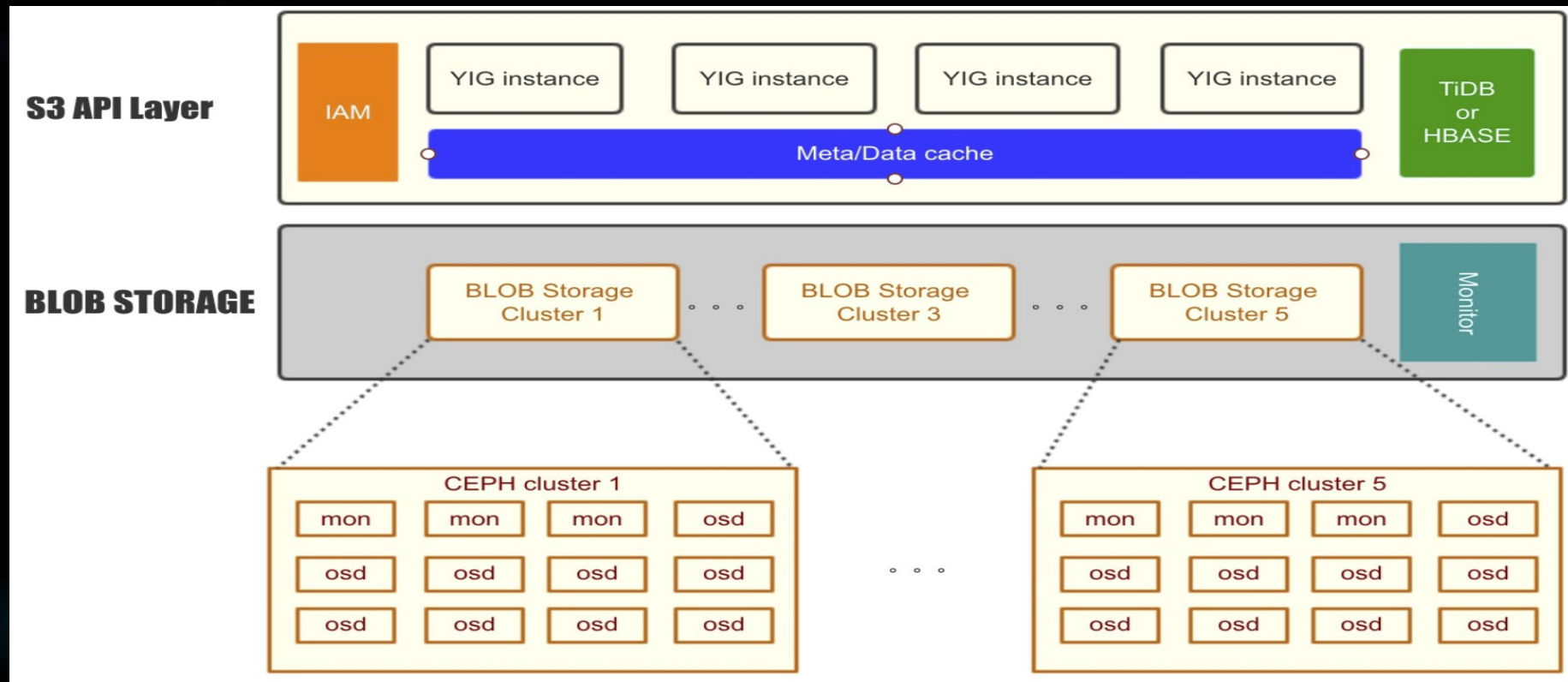
We created Yig in 2014 as the object storage which provides the s3 protocol. But many clients which want to use the object storage must develop the logic to communicate with Yig by using s3 sdk. This will introduce some works for the clients. Today, there are many IOT devices which have great requirement to upload their data to the object storage. And data lake is the next-gen cloud basic infrastructure, So, we need a uniform mechanism to access the Yig object storage. In the end, we decide to develop YigFS, which is a distributed file system which supports POSIX api and is on top of Yig object storage.

Yig Introduction

Yig is an object storage service which can be accessed through s3 protocol. Yig extends Minio backend storage to allow more than one ceph cluster to work together and form a super large storage resource pool, users could easily enlarge the pool's capacity to EB level by adding a new ceph cluster to this pool. Benefits are avoiding data movement and IO drop down caused by adding new host or disks to old ceph cluster as usual way. To accomplish this goal, Yig needs a distributed database to store meta information. Now already supports TiDB, MySQL.

Yig Introduction

Below is the architecture of Yig



YigFS introduction

YigFS is a distributed file system which provides POSIX API and stores the data into Yig object storage.

YigFS consists of two parts: YigFS client(fs client for short) and meta service.

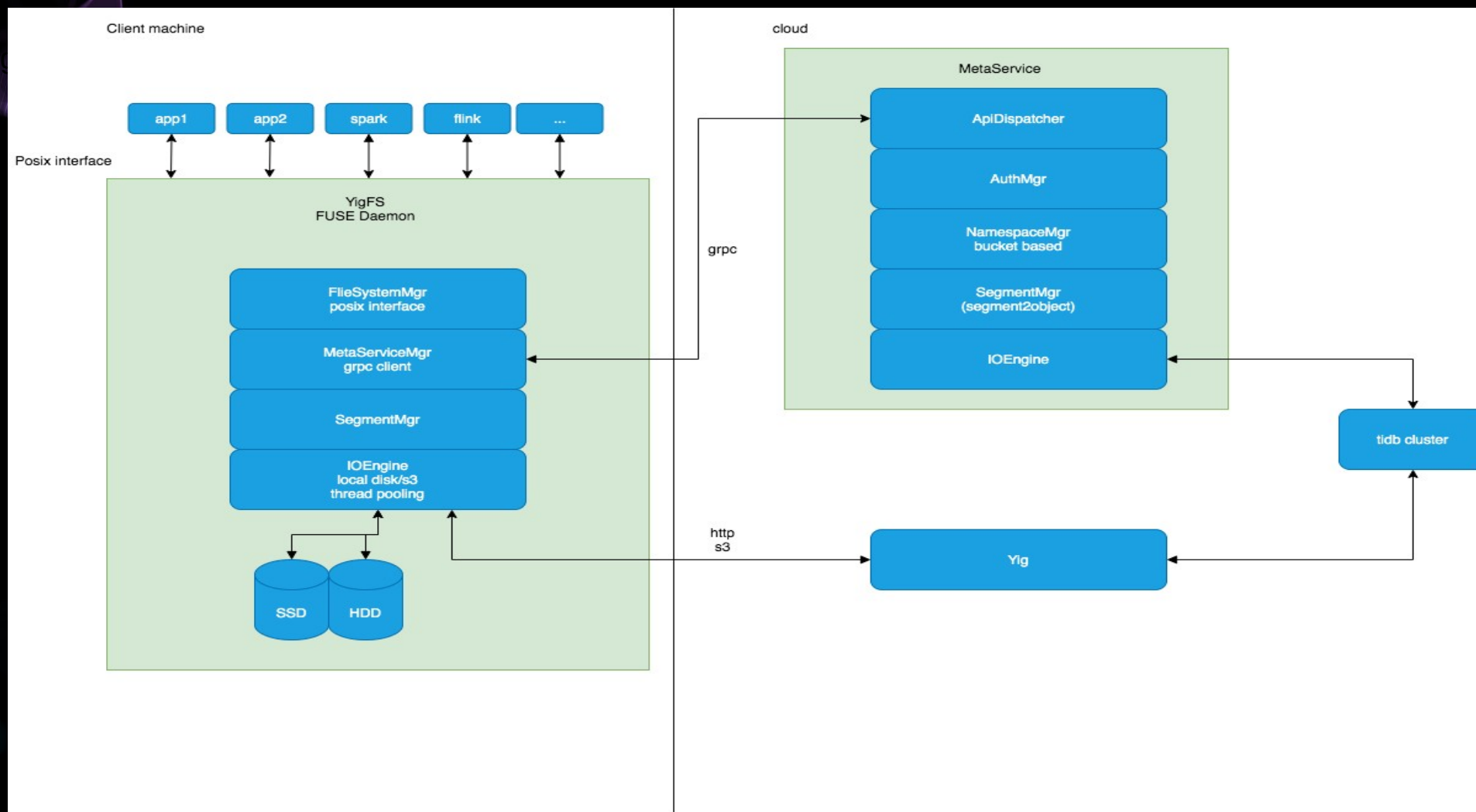
YigFS client is based on FUSE provided by linux platform. Actually it is a FUSE daemon, and it communicates with meta service for file meta and communicates with Yig for data.

YigFS meta service is a distributed service which provides the metadata management, such as file attributes, etc.

By providing the POSIX api, YigFS have below benefits:

1. Easy to use, no SDK integration
2. Support broad applications, such as Spark, etc.
3. Have high availability
4. Have high capacity

YigFS introduction



YigFS introduction

YigFS client:

In the left of the above architecture, it is the YigFS client. It is a FUSE daemon and implements the POSIX api according to FUSE specification. We use FUSE because it is already a part of Linux, and the YigFS client daemon runs in user space, so it is safe and flexibility.

YigFS client consists of FileSystemMgr, MetaServiceMgr, SegmentMgr and IOEngine.

FileSystemMgr:

It is responsible for the POSIX API.

MetaServiceMgr:

It is responsible for communication with Meta Service.

SegmentMgr:

It is responsible for cache data in local disks.

IOEngine:

It is responsible for cache IO in local disks and Yig object store.

YigFS introduction

YigFS Meta Service:

In the right of the above architecture, it is the YigFS Meta Service. It is a distributed system which provides file system meta data management. And it saves all the meta data into Tidb currently.

YigFS introduction

Communication:

Currently, YigFS client uses restful http to communicate with the Meta Service. But we will switch to grpc in the future.

YigFS client will save the data into Yig by using s3 protocol.



Thank You