百度对Elasticsearch的优化改进

百度大数据部 高攀

2016年12月10日

功能改进

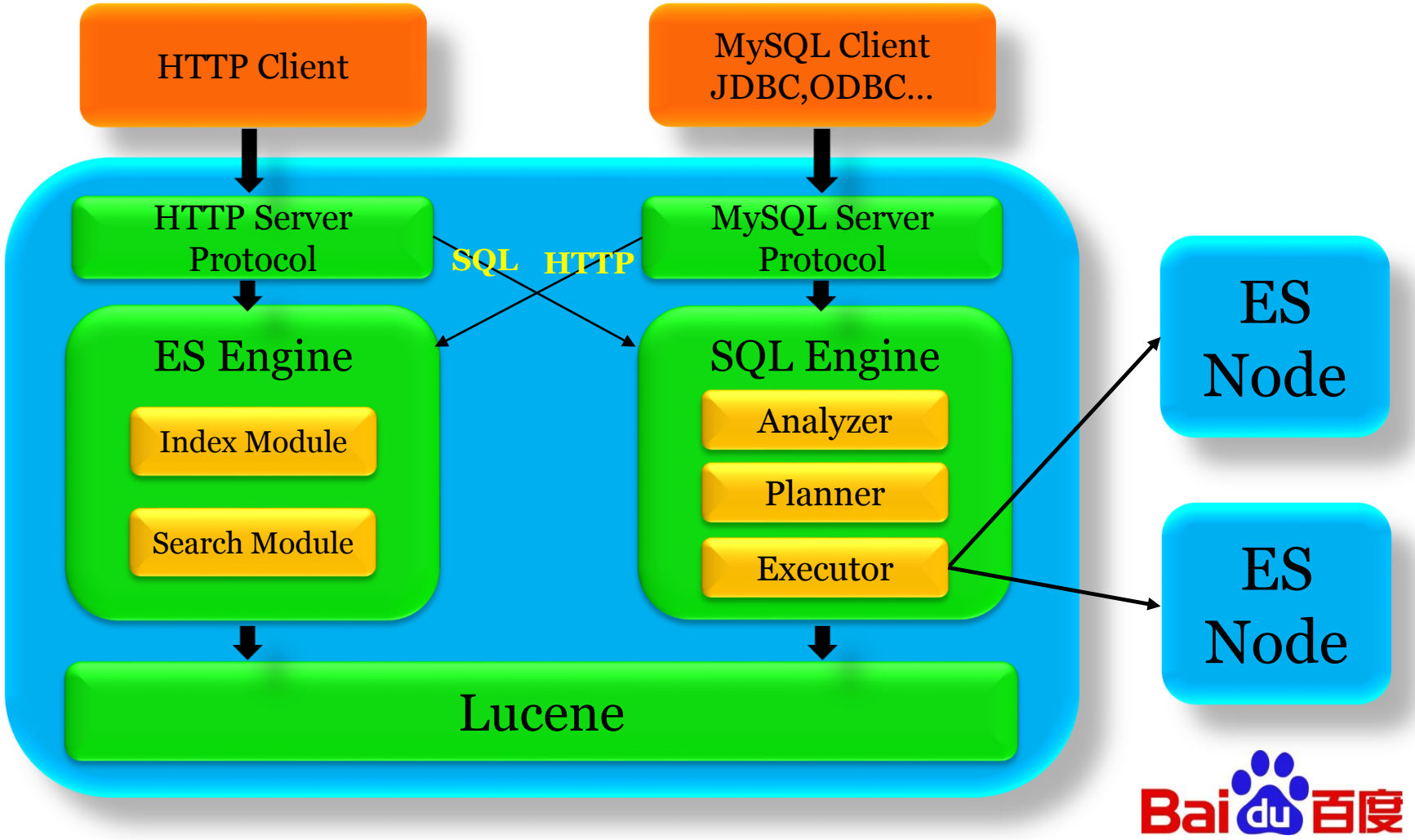
- 分布式SQL查询层
- 权限管理
- Online schema change
- DistributedLog 数据一致性
- 多集群数据同步
- 多租户资源隔离

Baidu 百度

分布式SQL查询层

- 提供标准SQL接口，方便使用，降低学习成本
  - 支持复合数据类型，如 Array、Map等
  - 支持通过SQL全文检索
  - 支持Join
  - 分布式SQL聚合
- 兼容MySQL协议，原MySQL、DDBS业务无缝迁移
  - MySQL Client，JDBC，ODBC …
- 兼容原始HTTP协议
  - 保留原始HTTP接口
  - 支持MySQL发送HTTP请求
  - 支持HTTP发送SQL请求

分布式SQL查询层


HTTP Client
MySQL Client
JDBC,ODBC...
HTTP Server
Protocol
MySQL Server
Protocol
SQL
HTTP
ES Engine
Index Module
Search Module
SQL Engine
Analyzer
Planner
Executor
Lucene
ES
Node
ES
Node

分布式SQL查询层

```
mysql>  create table db1.news(
    ->  id integer primary key,
    ->  title string,
    ->  label array(string),
    ->  author object as(
    ->  name string,
    ->  age integer)
    ->  );
Query OK, 1 row affected (0.17 sec)

mysql> insert into db1.news(id, title, label, author) values
    -> (1,'baidu',['bigdata','search'],{name='gaopan',age=26});
Query OK, 1 row affected (0.08 sec)

mysql> select * from db1.news;
+---------------------------+------+----------------------+-------+
| author                    | id   | label                | title |
+---------------------------+------+----------------------+-------+
| {"name":"gaopan","age":26} |    1 | ["bigdata","search"] | baidu |
+---------------------------+------+----------------------+-------+
1 row in set (0.31 sec)

mysql> update db1.news set author['age'] = 24 where author['name'] = 'gaopan';
Query OK, 1 row affected (0.14 sec)

mysql> select * from db1.news;
+---------------------------+------+----------------------+-------+
| author                    | id   | label                | title |
+---------------------------+------+----------------------+-------+
| {"name":"gaopan","age":24} |    1 | ["bigdata","search"] | baidu |
+---------------------------+------+----------------------+-------+
1 row in set (0.01 sec)
```

分布式SQL查询层

```
mysql> select * from db1.news order by id;
+-----------------------------+------+-----------------------+-------+
| author                      | id   | label                 | title |
+-----------------------------+------+-----------------------+-------+
| {"name":"gaopan","age":26} |    1 | ["bigdata","search"] | baidu |
| {"name":"bbb","age":24}    |    2 | ["bigdata","aaa"]    | baidu |
| {"name":"fff","age":30}    |    3 | ["bigdata","ccc"]    | sina  |
| {"name":"ggg","age":32}    |    4 | ["ddd","eee"]        | baidu |
+-----------------------------+------+-----------------------+-------+
4 rows in set (0.00 sec)

mysql> select * from db1.author order by id;
+------+--------+----------+
| id   | name   | phone    |
+------+--------+----------+
|    1 | gaopan | 12345678 |
|    2 | bbb    | 87654321 |
+------+--------+----------+
2 rows in set (0.01 sec)

mysql> select title,author['name'],author['age'],phone
    -> from db1.news, db1.author
    -> where db1.news.author['name'] = db1.author.name;
+-------+----------------+---------------+----------+
| title | author['name'] | author['age'] | phone    |
+-------+----------------+---------------+----------+
| baidu | gaopan         |            26 | 12345678 |
| baidu | bbb            |            24 | 87654321 |
+-------+----------------+---------------+----------+
2 rows in set (0.01 sec)
```

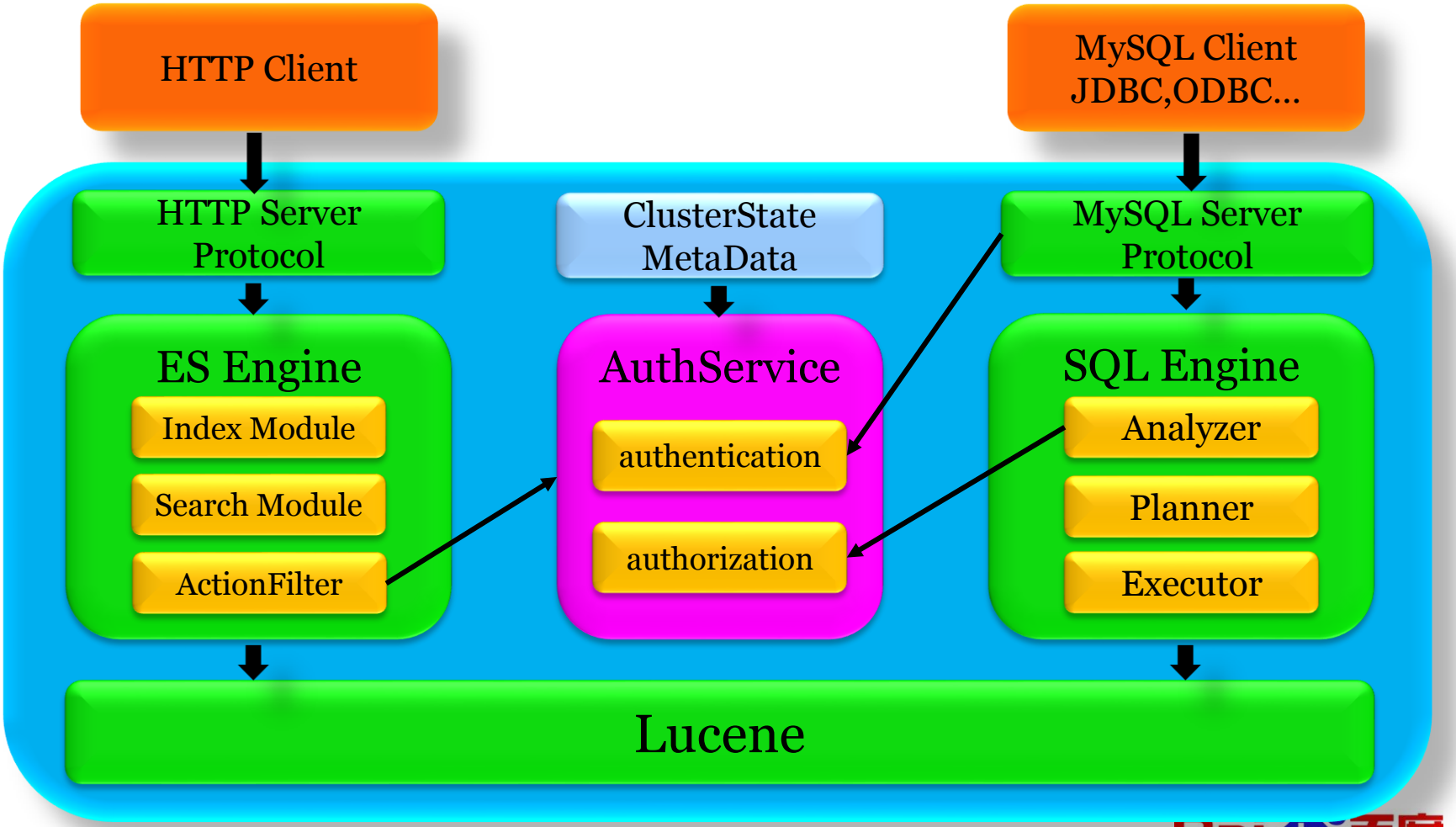
百度

权限管理系统

- 需求背景
  - 业务方对数据安全性敏感，不同角色只能访问自己的表
  - 要求支持读写权限分离，支持动态修改权限，各节点自动同步更新
  - 要求支持IP白名单 & BNS白名单，BNS自动更新
  - HTTP、MySQL 两种访问接口都需要进行权限控制
- 设计实现
  - 兼容ES 和 MySQL，增加database逻辑层，db.table
  - 权限级别：db，table
  - 用户级别：root，superuser，user
    - superuser可以创建用户并分配权限但不能修改集群配置，用于用户自己管理
  - 权限类型：read_only，read_write
  - 白名单：IP（通配符），hostname (BNS)

Baidu 百度

权限管理系统


HTTP Client
MySQL Client
JDBC,ODBC...
HTTP Server
Protocol
ClusterState
MetaData
MySQL Server
Protocol
ES Engine
Index Module
Search Module
ActionFilter
AuthService
authentication
authorization
SQL Engine
Analyzer
Planner
Executor
Lucene

权限管理系统

```
mysql> create user gaopan identified by "gaopan123";
Query OK, 1 row affected (0.01 sec)

mysql> alter user gaopan whitelist "s[redacted]t01";
Query OK, 1 row affected (0.01 sec)

mysql> grant read_only on db1.news to gaopan;
Query OK, 1 row affected (0.01 sec)

mysql> exit
Bye
[search@s[redacted].baidu.com ~]$ ./mysql-client -ugaopan -pgaopan123
 -hs[redacted]t01 -P8306
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 35
Server version: 5.1.0 Baidu Elasticsearch Edition

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select * from db1.news limit 1;
+-------------------------+------+---------------+-------+
| author                  | id   | label         | title |
+-------------------------+------+---------------+-------+
| {"name":"ggg","age":32} |    4 | ["ddd","eee"] | baidu |
+-------------------------+------+---------------+-------+
1 row in set (0.02 sec)

mysql> insert into db1.news(id,title) values (5,'test');
ERROR 1064 (HY000): SQLActionException: UNAUTHORIZED 4401 NoPermissionException: gaopan
have no permission to READ_WRITE on table: db1.news
```

Online schema change

- **需求背景**
  - 业务需要在线动态增加字段、增加索引、删除索引、修改分词器等
  - 要求对业务透明，不能影响在线服务
- **社区 reindex**
  - 新创建一个index，scroll then bulk
  - 无法对在线业务透明，必须双写两个index，不能update
  - 大量额外的网络IO和磁盘空间

```
POST /_reindex
{
  "source": {
    "index": "twitter"
  },
  "dest": {
    "index": "new_twitter"
  }
}
```

Baidu 百度

Online schema change

- **Baidu-ES reindex**
- 每个分片内部本地reindex
- 增加mapping version，插入时使用最新mapping，并使用internal version，防止覆盖
- 后台线程定期检测，Mapping version增加，自动reindex
- 动态调整reindex速度
- 节点宕机、分片迁移、reindex自动恢复
- 无需新建index，无需额外网络IO和磁盘空间
- 支持在线服务，支持update
- 查看每个分片reindex进度

```
PUT twitter/_mapping/tweet?reindex=true
{
    "properties": {
        "message1": {          //增加索引
            "type": "string",
            "index": "not_analyzed",
            "doc_values": true
        },
        "message2": {          //修改分词器
            "type": "string",
            "analyzer":"ik"
        },
        "message3": {          //增加字段
            "type": "text"
        },
        "message4": {          //删除索引
            "index": "no"
        }
    }
}
```

DistributedLog 数据一致性

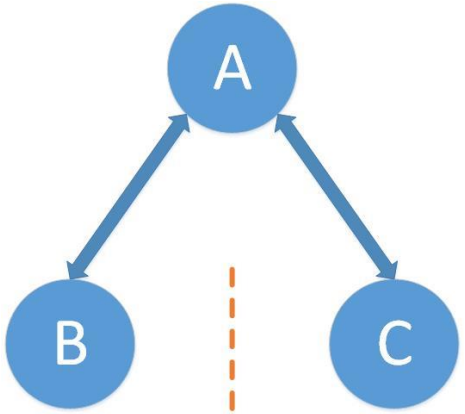
- **需求背景**
  - 糯米、钱包等业务对ES服务可靠性要求很高
  - 不能容忍脑裂、数据不一致、丢数据等情况的发生
- **需解决的问题**
  - 元数据一致性（脑裂）
  - 强一致写
  - 强一致读
- **解决方案：DistributedLog**
  - Twitter 开源分布式日志服务
  - 高可用，高性能，强一致，可扩展

Baidu 百度

DistributedLog 数据一致性（元数据）

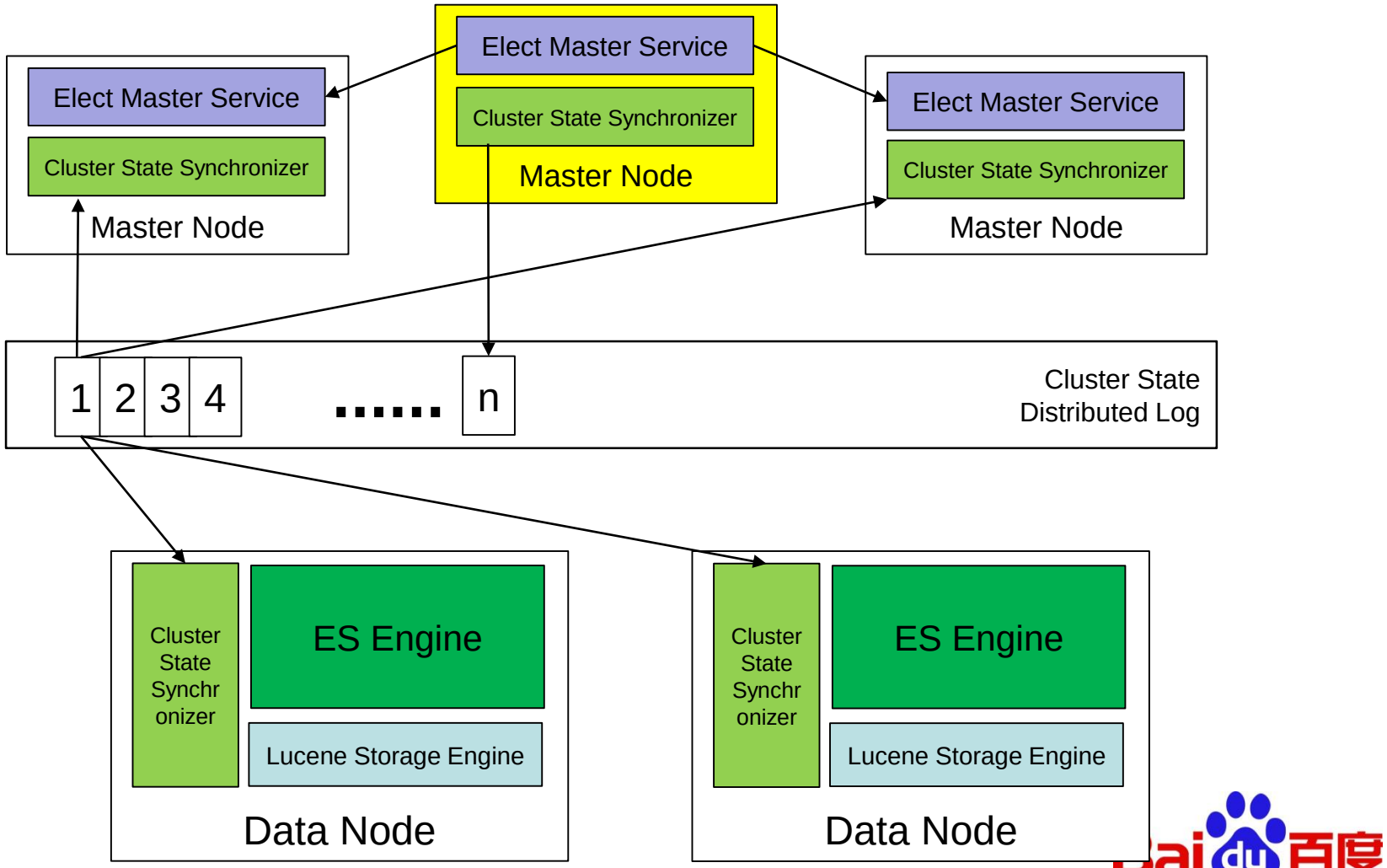
元数据

- 当前问题：排序选主，可能脑裂，元数据不一致
- Elect Master Service
  - 实现简单的类Raft选主协议
    - 本地的cluster state必须是最新的
    - 每个节点一段时间内只能投票一次
    - 各Master Node竞争式的抢占选举，获取Quorum 票数当选



- Cluster State Synchronizer
  - 从DL中Pull元数据变更，向本地Cluster State Apply
- 总体流程
  - Master Leader向DL中写入Cluster State 变更
  - 其余的Master节点和所有的DataNode节点从DL中获取变更并向本地Apply

DistributedLog 数据一致性（元数据）



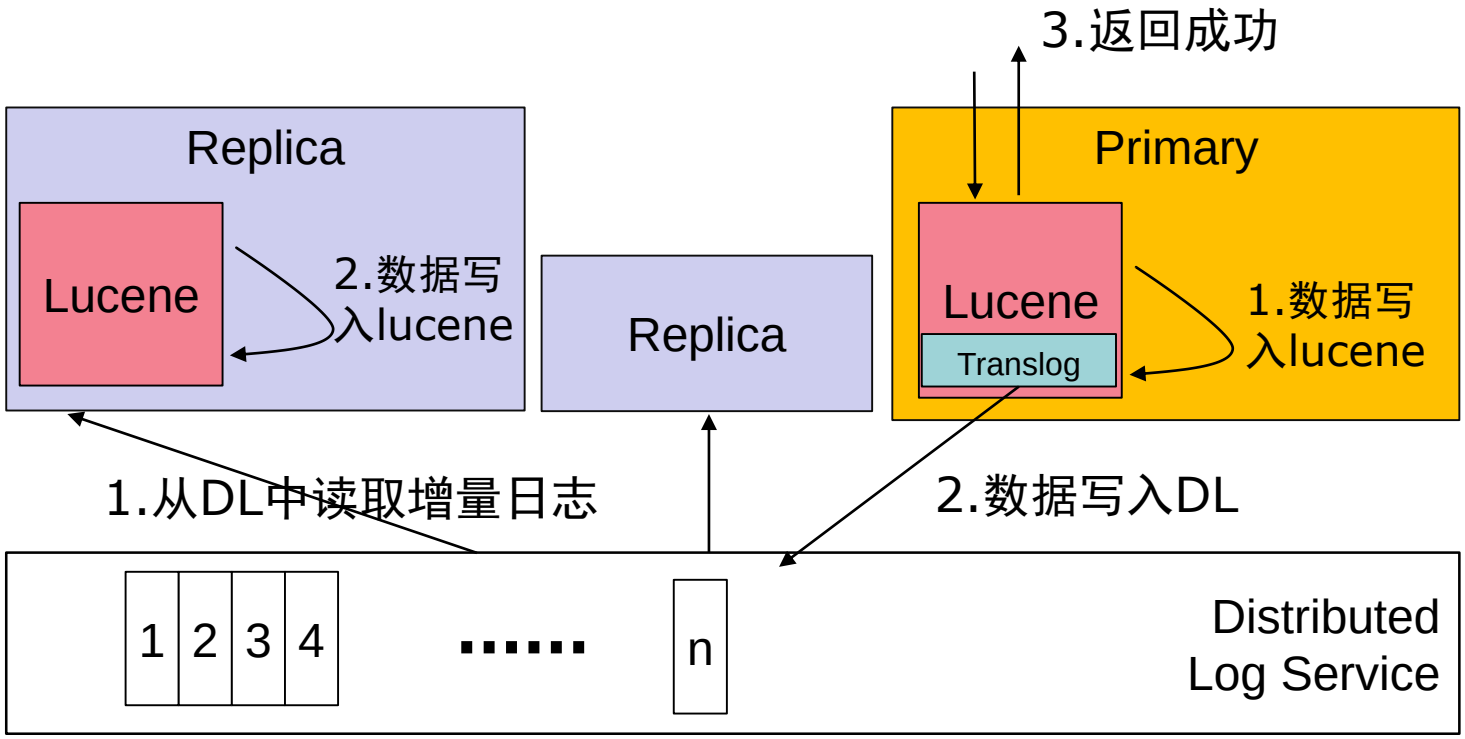
DistributedLog 数据一致性（一致写）

强一致写

- 当前问题
  - 没有较好的一致性算法
  - Primary分发到多个Replica，各replica数据可能不一致(已写成功的无法回退)
  - Primary挂掉，随机选择replica成为primary，数据可能丢失
- 基本思路
  - Master指定Primary
  - Primary将日志写入DL
  - Replica从DL中读取日志并回放
  - Translog必须在Lucene引擎内部实现为原子操作
    - 如果先写log，Lucene可能写入不成功
    - 如果先写Lucene，log有可能写入不成功

DistributedLog 数据一致性（一致写）

强一致写



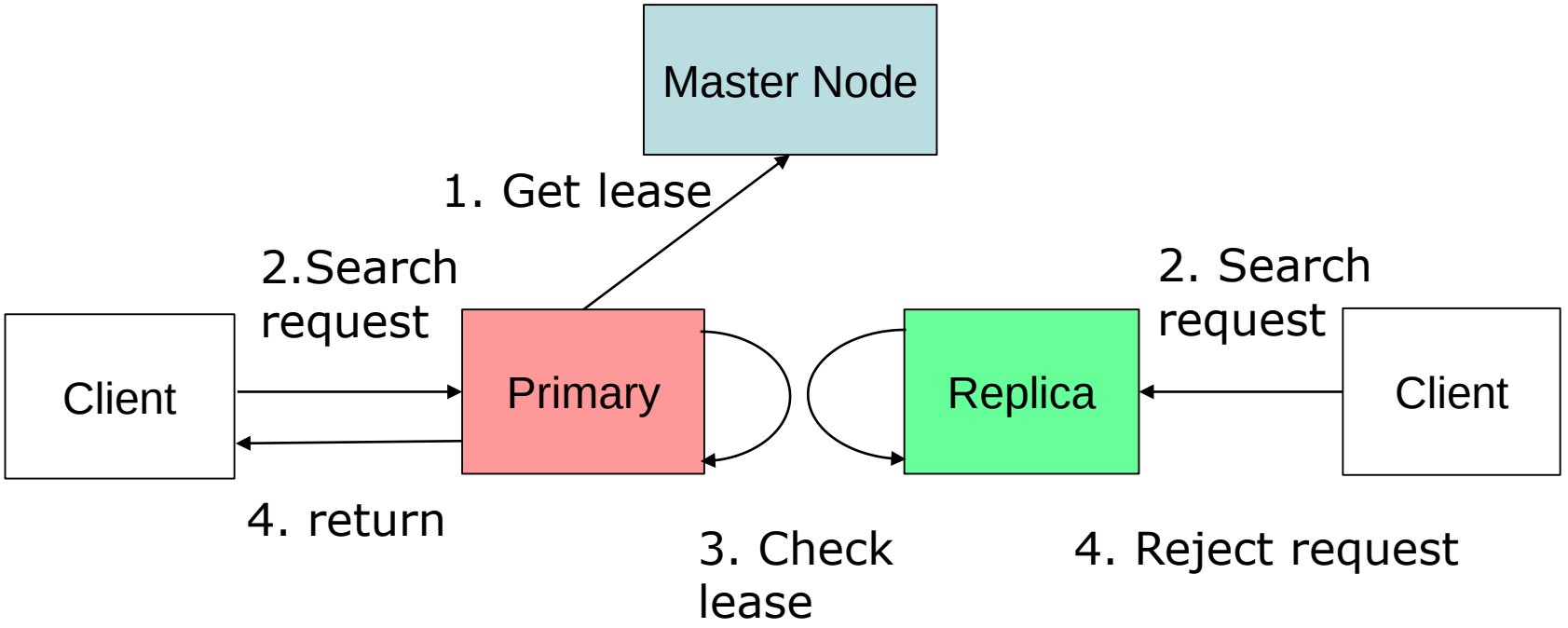
DistributedLog 数据一致性（一致读）

强一致读

- 当前问题
  - Master检测到原primary异常，选出新primary
  - 元数据未及时同步到client，查询被转发到旧primary
  - 旧primary同样未意识到自己已经不是primary，执行查询
- Lease机制
  - Primary需要定时从Master获取Lease
  - 读取时首先检查Lease，当Lease Expire时不再提供读取服务
  - 查询只查primary

DistributedLog 数据一致性（一致读）

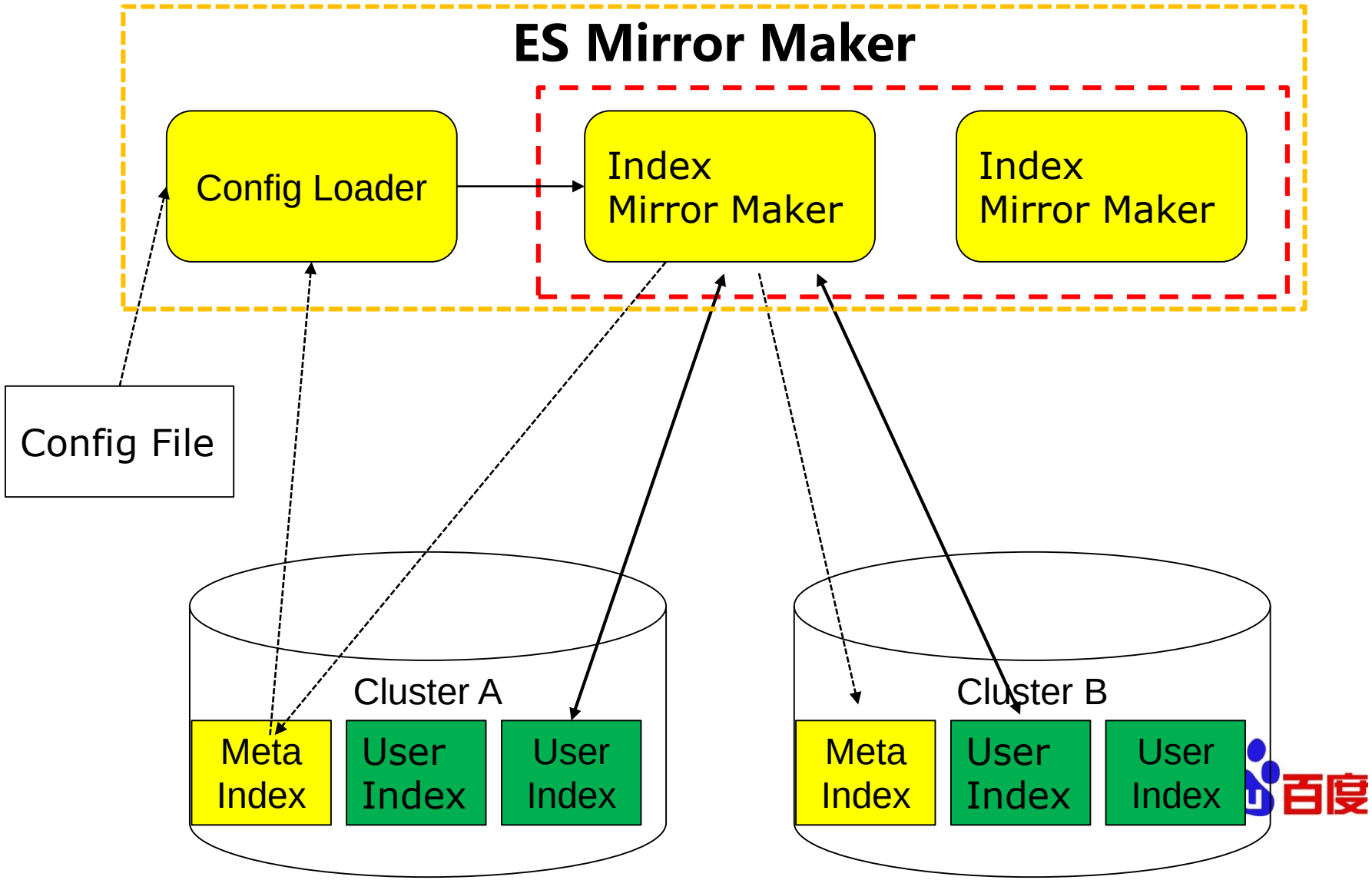
强一致读---Lease机制



多集群数据同步

- 需求背景
  - 业务要求高可用，两地三中心部署
  - 北京、上海、广州，多个主备集群需实时同步增量数据
  - 主备切换后的冲突处理
- 设计实现
  - 每个Doc都有修改的timestmap和version信息
  - Mirror Maker根据timestamp获取index修改的增量信息
  - Mirror Maker将增量的更新发往目标集群的Index中
  - 冲突时根据version来判断是否覆盖目标集群里的Doc
  - 默认version使用timestmap
- 实现方案二：DL

多集群数据同步


ES Mirror Maker
Config Loader
Index
Mirror Maker
Index
Mirror Maker
Config File
Cluster A
Meta
Index
User
Index
User
Index
Cluster B
Meta
Index
User
Index
User
Index

多租户资源隔离

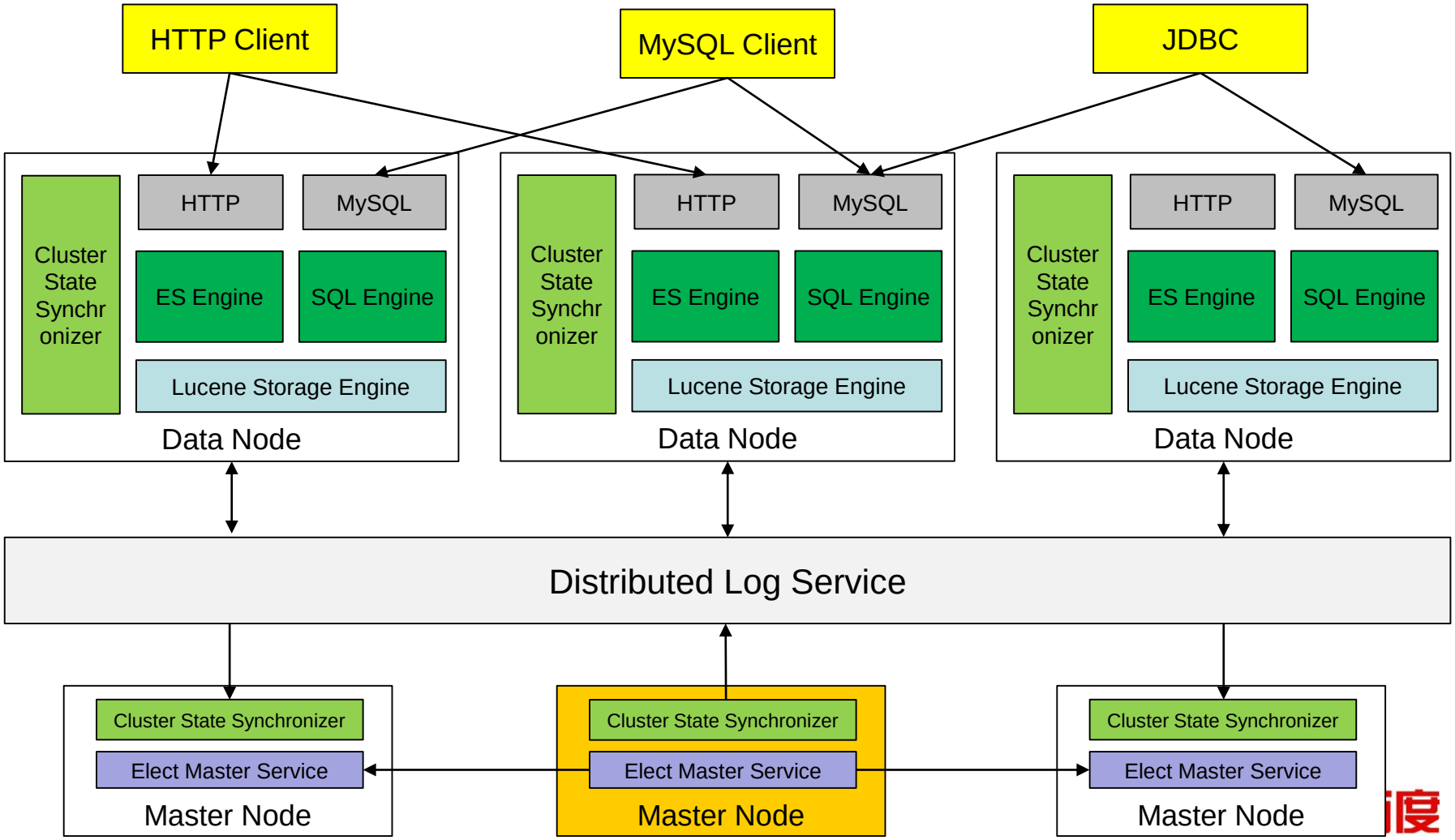
- 需求背景
  - 多个业务使用公共集群，CPU、内存、IO、JVM等相互影响
  - 云化部署
- 设计实现
  - 每台物理机启动多个ES进程组成大集群
  - cgroup 对每个ES进程进行CPU、内存、IO等隔离
  - 引入tenement 概念，分配不同的ES节点为每个租户创建自己的虚拟集群
  - Allocation filter 限制租户的index只能创建在自己的节点上
  - 每个租户分配不同DB，隔离访问权限
  - username@tenement，租户命名空间隔离租户信息
  - 根据租户ID隔离settings，templates、nodes等

Baidu 百度

多租户资源隔离


cgroup
ES node
cgroup
ES node
cgroup
ES node
cgroup
ES node
物理机
cgroup
ES node
cgroup
ES node
cgroup
ES node
cgroup
ES node
物理机
cgroup
ES node
cgroup
ES node
cgroup
ES node
cgroup
ES node
物理机
cgroup
ES node
cgroup
ES node
cgroup
ES node
cgroup
ES node
物理机

Baidu-ES 整体架构



Baidu-ES 规模

- 覆盖百度内部50+业务线
  - 包括凤巢、糯米、金融、钱包、网盟、手百、地图、贴吧、移动、风控、预测、云分析、用户画像等
- 共部署500+机器，启动800+ES节点
- 单集群最大100台机器，启动200个ES节点
- 单集群最大每天导入60TB+，总共每天100TB+
- ES云服务多个付费用户，如大秦铁路、光明网等

Baidu 百度

Baidu-ES 云服务

- https://cloud.baidu.com/product/bes.html
- 需求合作，请联系团队邮箱：palo-rd@baidu.com

Baidu 百度

结束

谢谢！

Baidu 百度