



Alibaba Developer  
Conference

## 淘宝数据服务平台 产品、架构与实践

---

淘宝网- 数据平台与产品部    泽远

## 淘宝数据服务平台

产品

架构

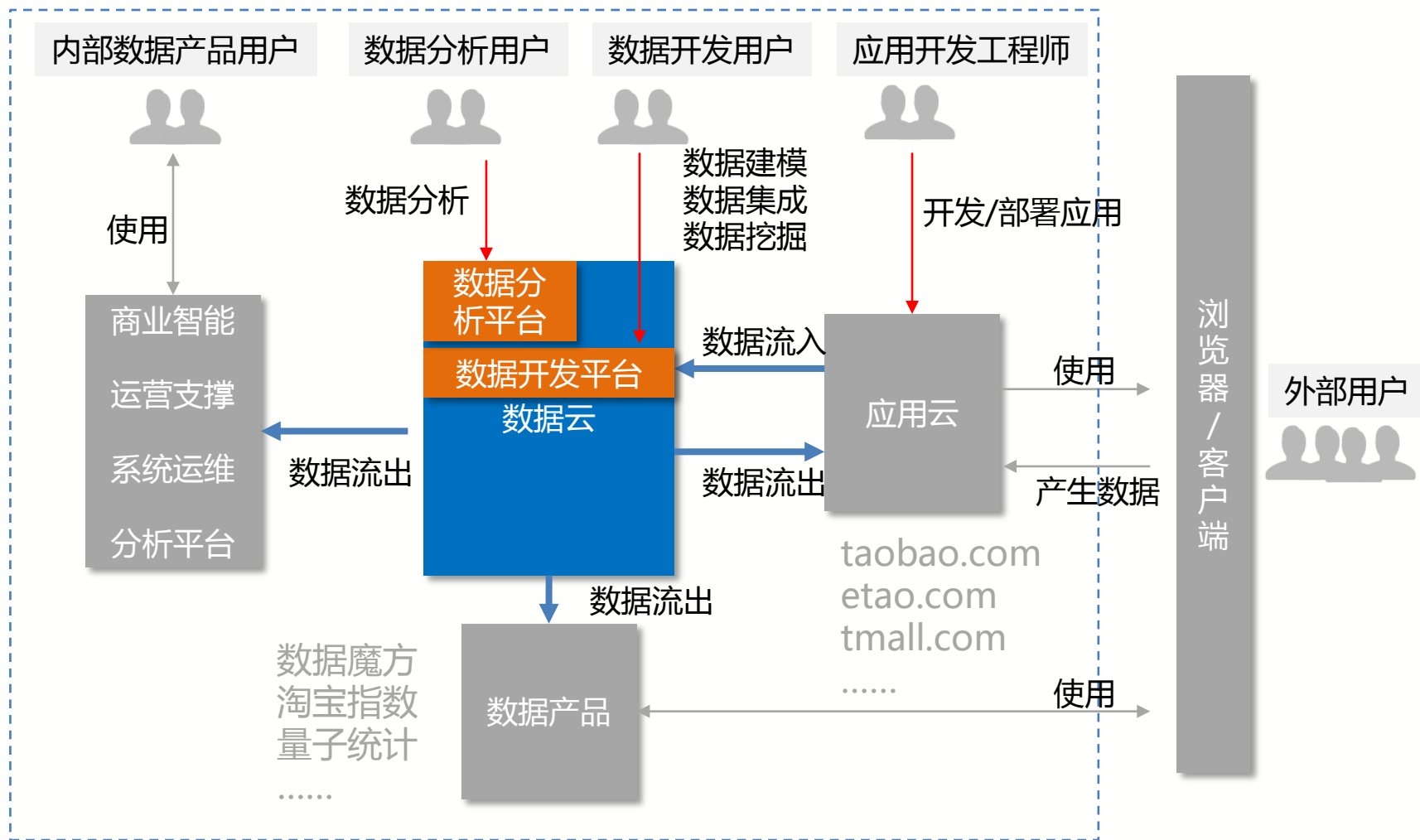
实践

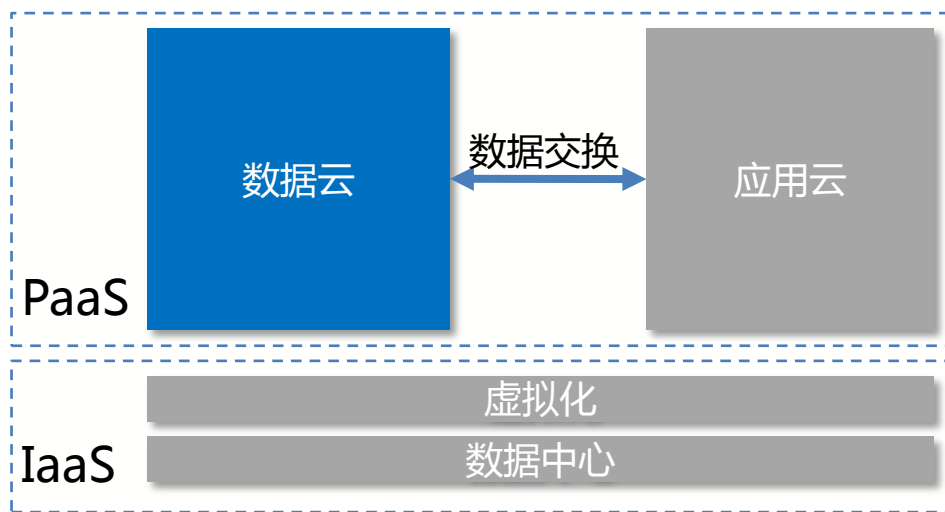
## 淘宝数据服务平台

产品

架构

实践

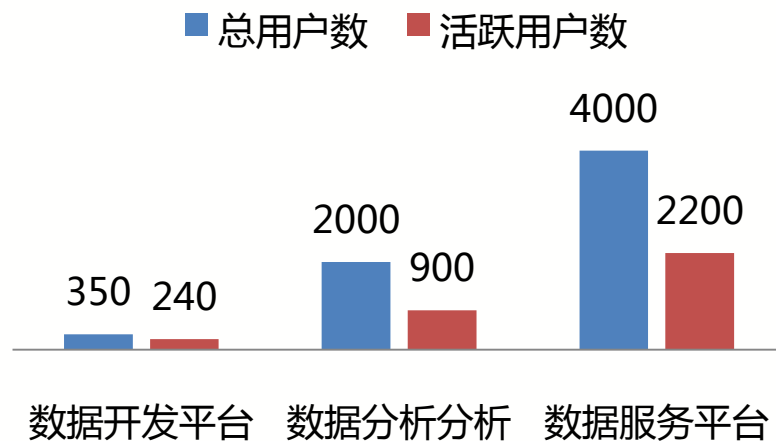
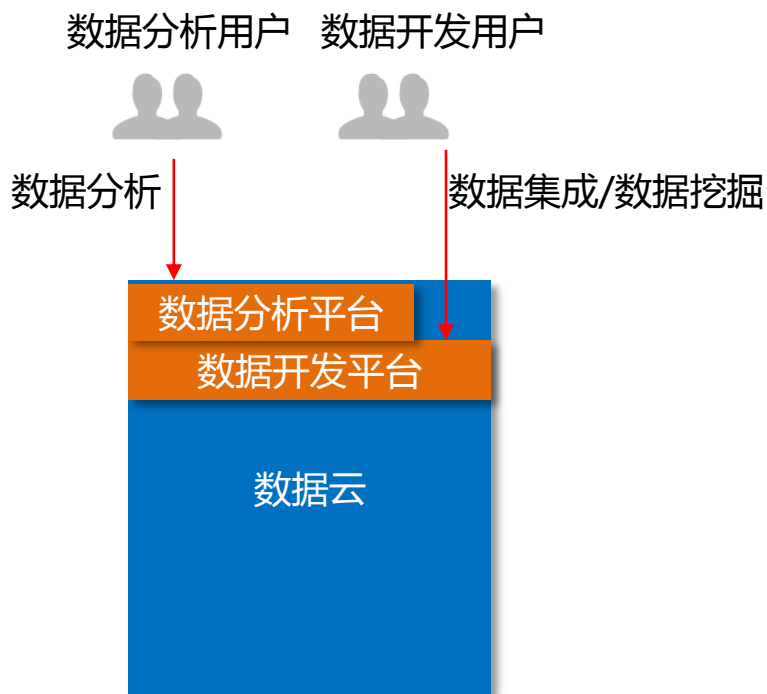




	数据云	应用云
目标	处理海量数据	处理海量HTTP请求
核心组件	海量数据存储 海量数据处理框架 海量数据仓库 元数据管理 数据同步与集成 工作流与调度	应用服务器 分布式缓存 分布式消息队列 分布式文件系统 分布式数据库 分布式简单存储
权衡	成本和吞吐量	SLA

数据化运营：全民挖掘数据价值

互联网精神：数据使用草根化，平台化

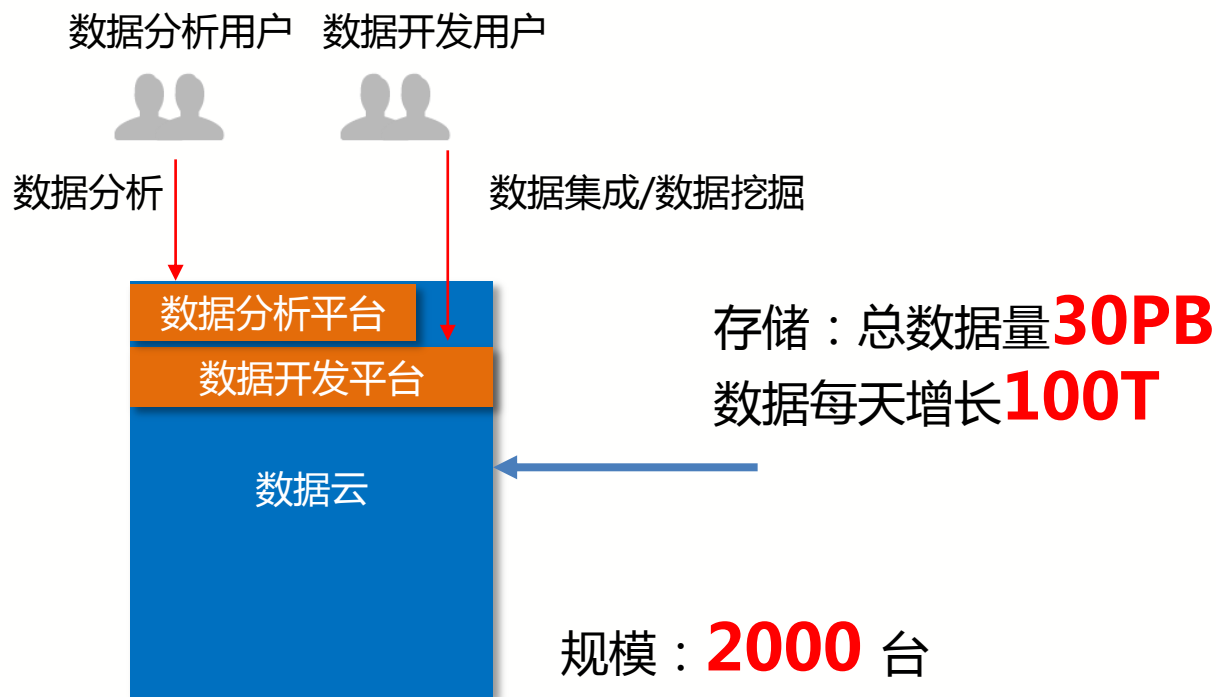


每 **8** 个小二中，就有一个是 **数据分析** 平台的活跃用户

每 **30** 个小二中，就有一个经常使用 **数据开发** 平台的活跃用户。

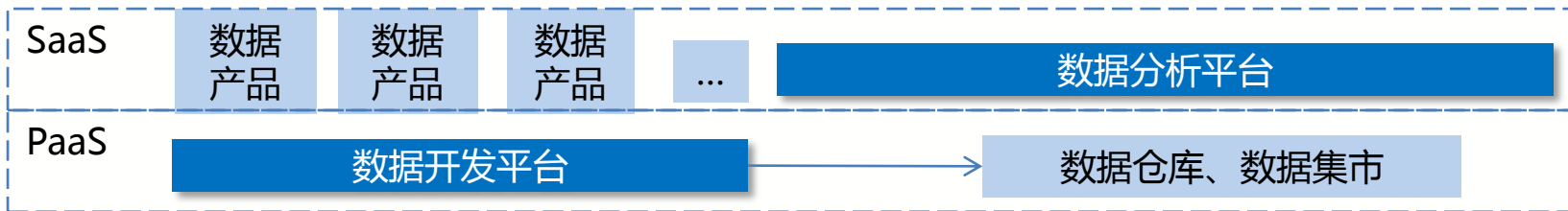
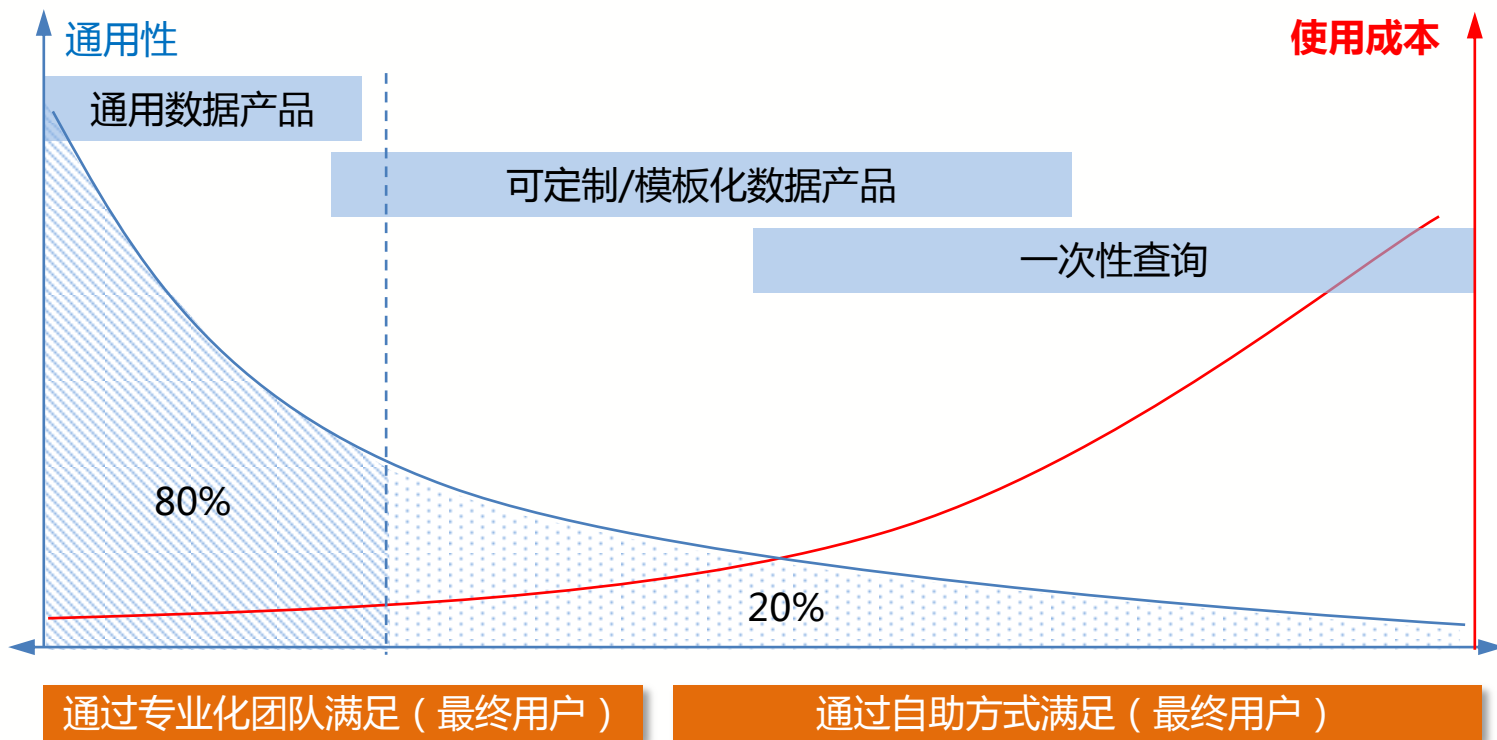
每天提交的分析任务数 **10K**

每天的集成/挖掘任务数 **100K**



计算：读取数据量 **5PB**，写出数据量 **500TB**

每天消耗的总CPU\*S数：**2.4Gs \* 2.4GHz**





### 场景比较

方面	数据开发	数据分析
目标	数据建模，数据集成，数据挖掘	数据分析
需求通用性	高	低
代码规模	大	小
是否需要 workflow	一般需要	一般不需要
团队规模	3-50人	1-3人
开发流程和多环境支持	需要	不需要
用户背景	专业开发团队	可能没有技术背景
SLA要求	高	低
是否需要数据架构	一般需要	一般不需要
服务层次	PaaS	SaaS

### 产品权衡

数据开发	数据分析
方法论驱动，强调标准和规范	伸缩性第一
以项目为中心	以用户为中心
有时为了灵活性牺牲简单	有时为了简单牺牲灵活性
适当暴露技术细节	尽可能屏蔽技术细节



Analyze terabytes of data with just a click of a button

Use Google BigQuery to interactively analyze massive datasets — up to billions of rows.

### Compose Query

```
SELECT timestamp, title, COUNT(*) as cnt
FROM publicdata:samples.wikipedia
WHERE LOWER(title) CONTAINS 'speed' AND wp_namespace = 0
GROUP BY title, timestamp ORDER BY cnt DESC LIMIT 20;
```

**RUN QUERY** Query complete (4.1s elapsed, 11.5 MB)

### Query Results

Row	timestamp	title
1	1196276720	New Hampshire Motor Speed
2	1187028345	Speedway World Team Cup
3	1043861144	Speed of gravity

### What is Google BigQuery?

Google BigQuery is a web service that lets you perform interactive analysis of massive datasets—up to billions of rows. Scalable and easy to use, BigQuery lets you and your business tap into powerful data analytics.

[Learn More](#)

项目 | 文件 | 编辑 | 查看 | 搜索 | 运行 | 工具 | 帮助

当前项目: 李远的个人项目

文件 | 公共表 | 个人表 | 函数 | 计划任务

关键字

- 李远的个人项目
  - abc
    - abc.conf 可编辑 2012-02-21 13:14:10
    - untitled-2012/2/21/13:14:14 李远锁定
    - untitled-2012/2/7/15:11:17 李远锁定
    - untitled-2012/3/31/21:15:40 可编辑
  - abc 可编辑 2012-05-21 10:03:22
  - abc.conf 李远锁定中 2012-04-12 15:43:43
  - abc\_3 可编辑 2012-04-12 15:43:28
  - helloworld 李远锁定中 2012-04-12 15:43:43
  - test 可编辑 2012-04-12 15:43:43
  - untitled...08:41 可编辑 2012-03-07 10:13:50
  - untitled-2012/3/28/12:43:17 李远锁定中
  - untitled-2012/3/8/17:27:27 李远锁定中
  - untitled-2012/4/1/15:19:28 李远锁定中
  - untitled-2012/4/10/18:50:38 李远锁定中
  - untitled-2012/4/13/11:13:50 李远锁定中
  - untitled-2012/4/13/13:10:58 李远锁定中
  - untitled-2012/4/13/13:50:11 李远锁定中
  - untitled-2012/4/13/14:24:08 可编辑 2012-04-13 14:24:08
  - untitled-2012/4/13/14:28:45 可编辑 2012-04-13 14:28:45
  - untitled-2012/4/13/14:41:20 可编辑 2012-04-13 14:41:20
  - untitled-2012/4/13/17:40:51 可编辑 2012-04-13 17:40:51
  - untitled-2012/4/13/21:42:47 可编辑 2012-04-13 21:42:47
  - untitled-2012/4/16/15:33:35 可编辑 2012-04-16 15:33:35
  - untitled-2012/4/18/15:38:03 李远锁定中
  - untitled-2012/4/21/18:02:47 李远锁定中

abc.conf | 计算一年

```
#!/bin/bash
#每天最多计算多少个分区
max_count=10;

#开始日期
begin_date="2012-04-07"
#结束日期
end_date="2012-04-11"

begin_date="2012-04-07"
echo $((($date -d "$begin_date" +%Y)-1990)*365 + $(date -d "$begin_date" +%j))

echo $(date -d "2012-01-01" +%j) +
echo $(date -d "2012-04-07" +%j)

echo $((($date -d "2012-04-11" +%j)-$date -d "2012-04-07" +%j)))

count=1;
hive="/home/hive/hive/bin/hive -u root -p root";

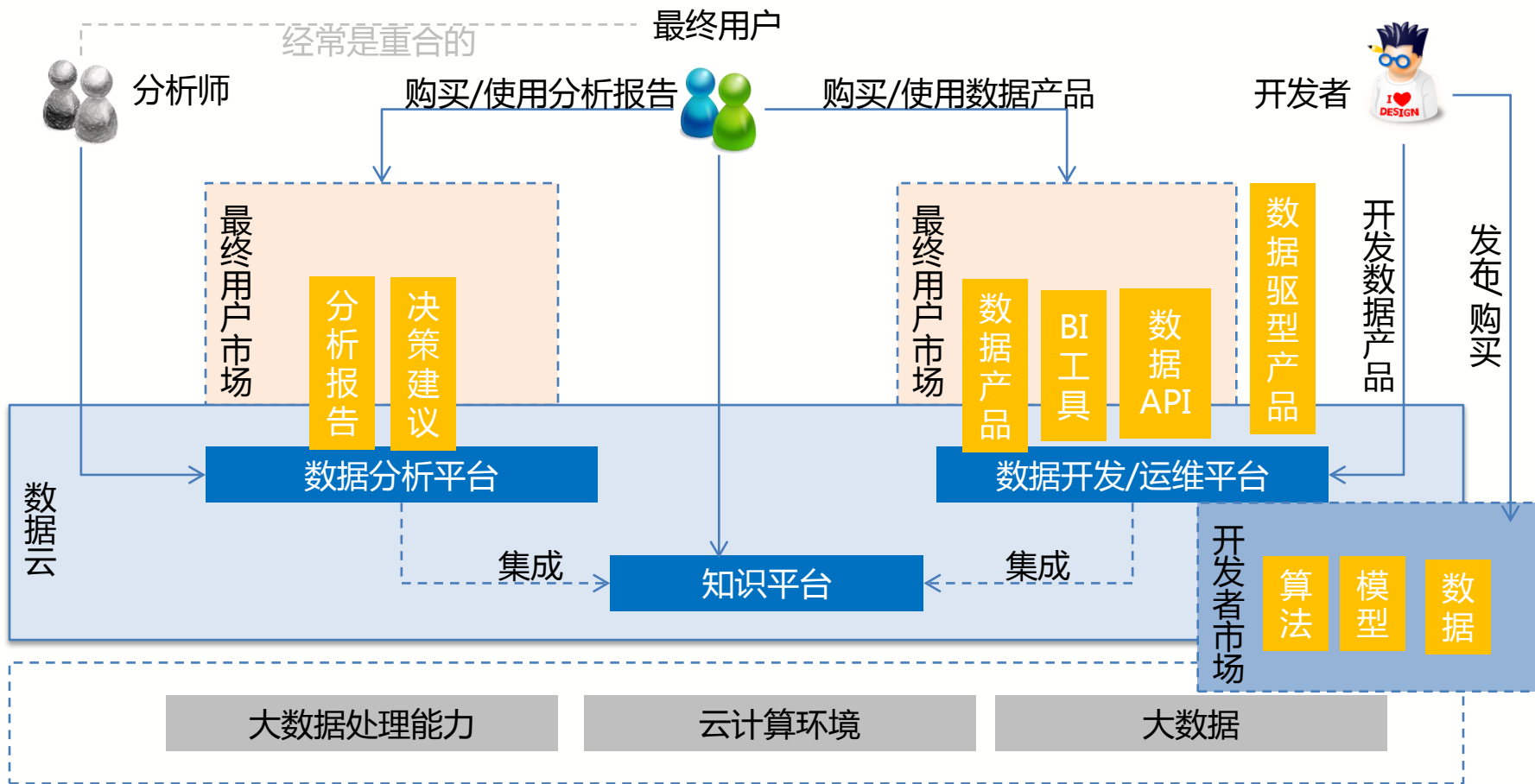
运行日志
```

大纲 | 运行历史记录

SQL

```
#!/bin/bash #每天最多计算多少个分区 max_count=10
#开始日期 begin_date="2012-04-07" #结束日期 end_date="2012-04-11"
begin_date="2012-04-07" echo $((($date -d "$begin_date" +%Y)-1990)
hive="/home/hive/hive/bin/hive -u root -p root"
for (( i = 1
i <= 6
i++ )) do ptdate=`date -d "$date" +%Y-%m-%d`
partition_size=$(hadoop dfs -count /group/analysts/ide_taobao-hz_zeyuan.wf/hive/tmp_user_a... | awk '{print($1)}')
continue
else echo "-----$ptdate的分区不存在，开始计算(这是今天计算的第$count个分区)-----"
fi
hivesql="insert overwrite table
shive -e "hivesql" echo "-----$ptdate的分区计算完成-----"
count=$((count+1)) if (( count >= max_count + 1 ))
exit
fi done
```



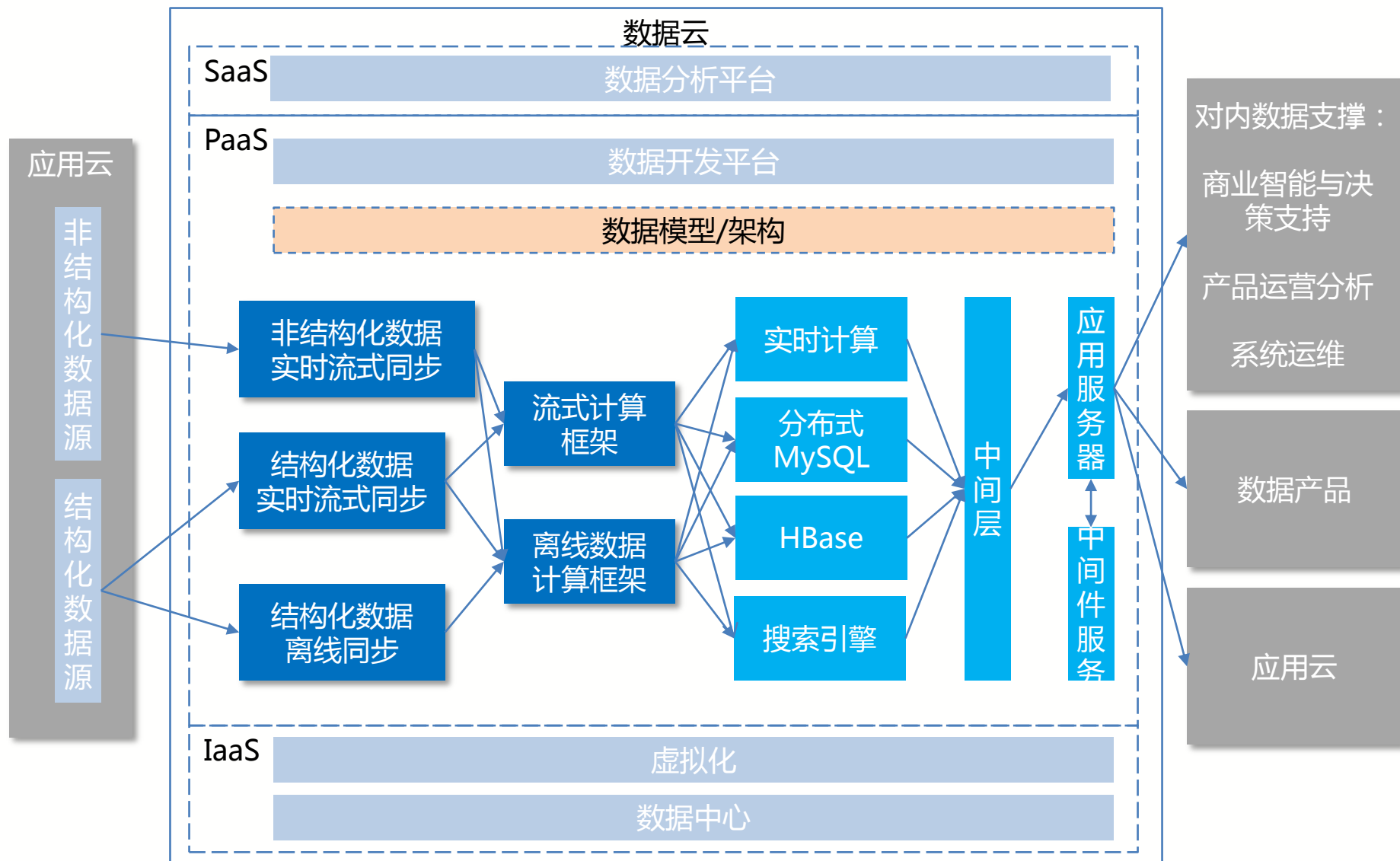


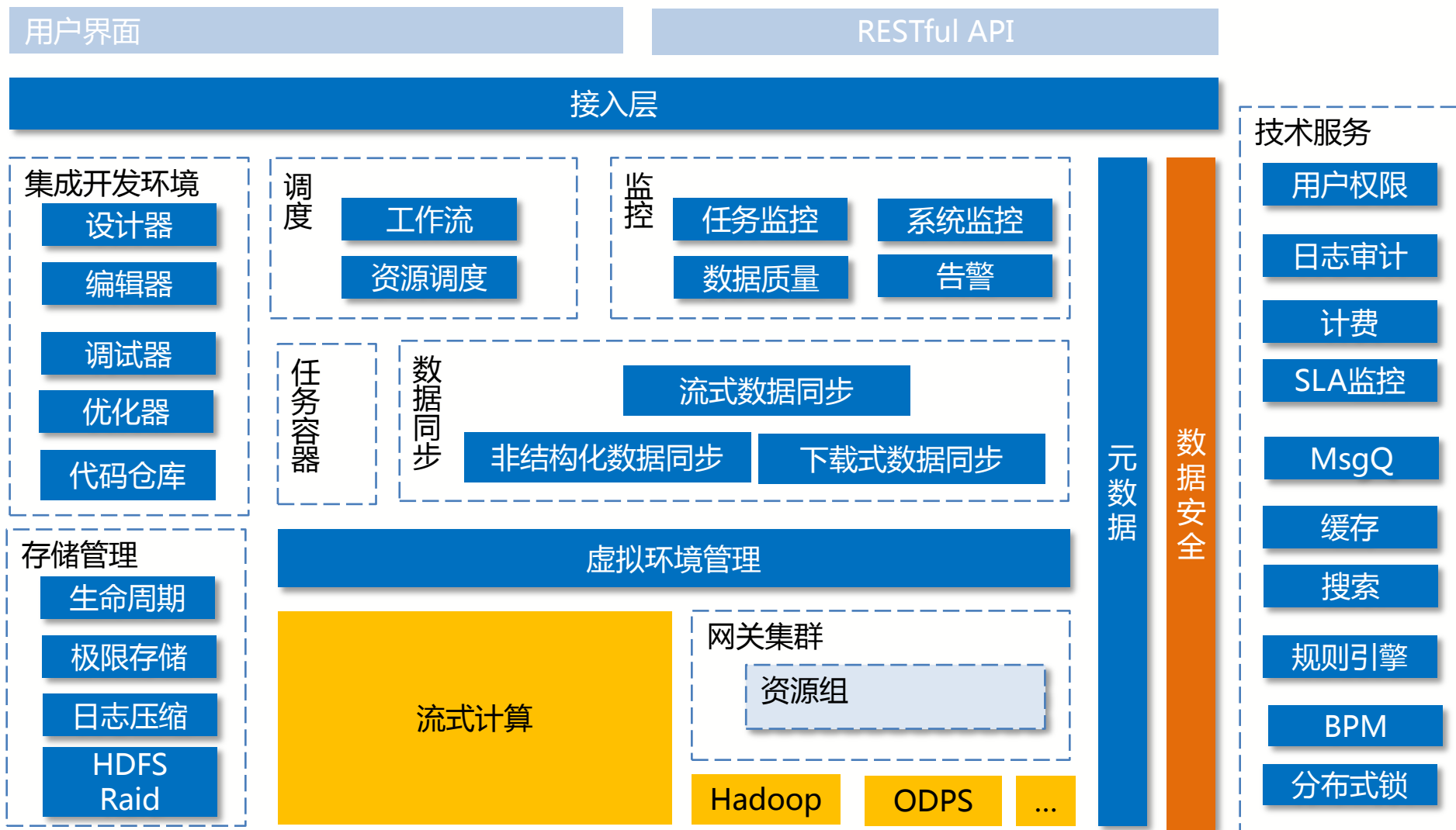
## 淘宝数据服务平台

产品

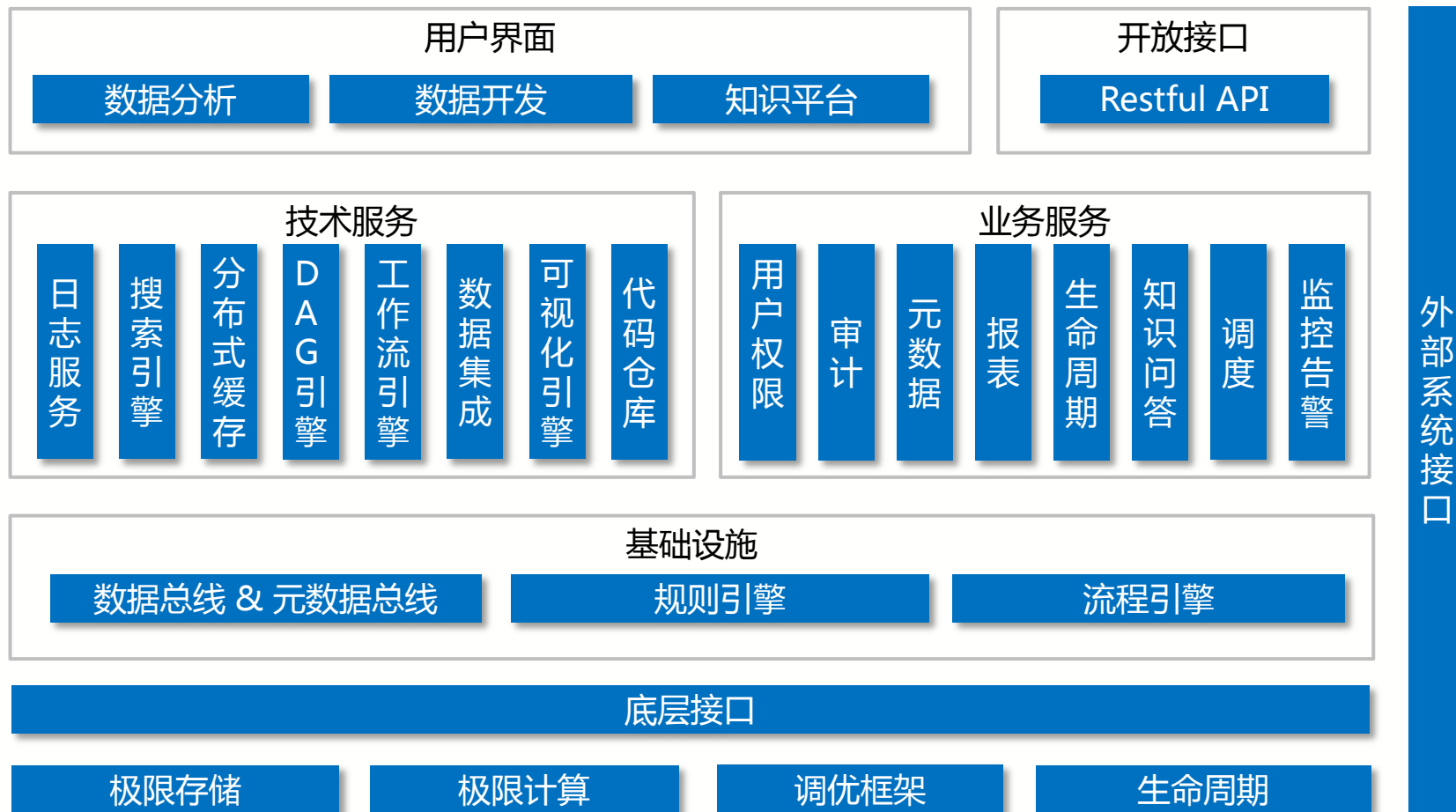
架构

实践









## 淘宝数据服务平台

产品

架构

实践

成本

可用性

伸缩性

可扩展性

性能、安全...

产品

架构

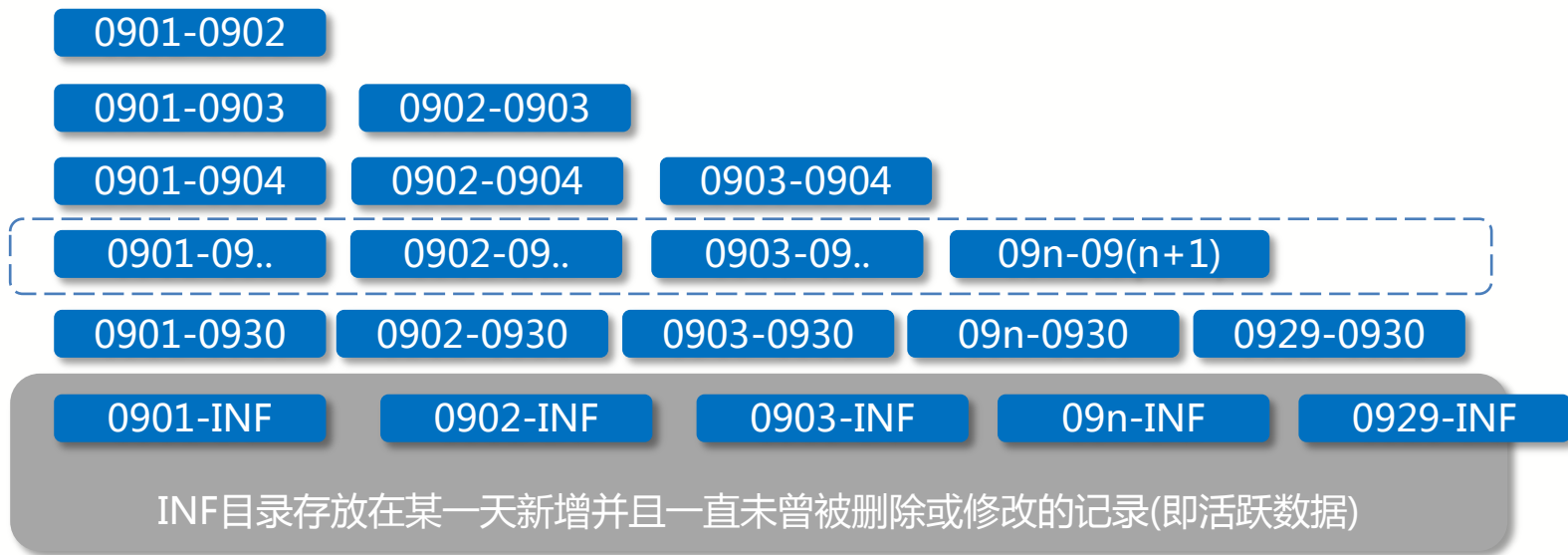
实践

产品

架构

实践

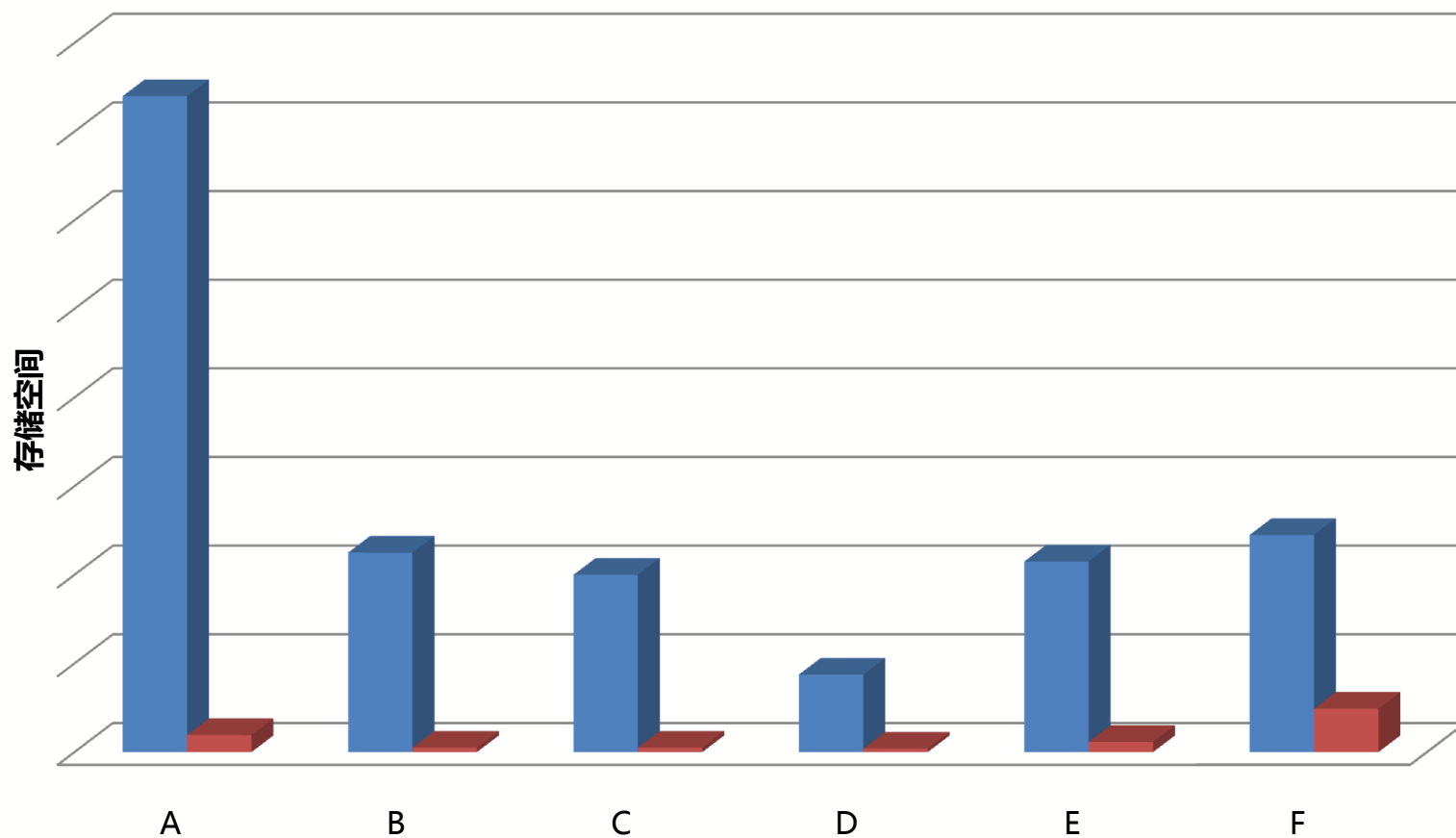


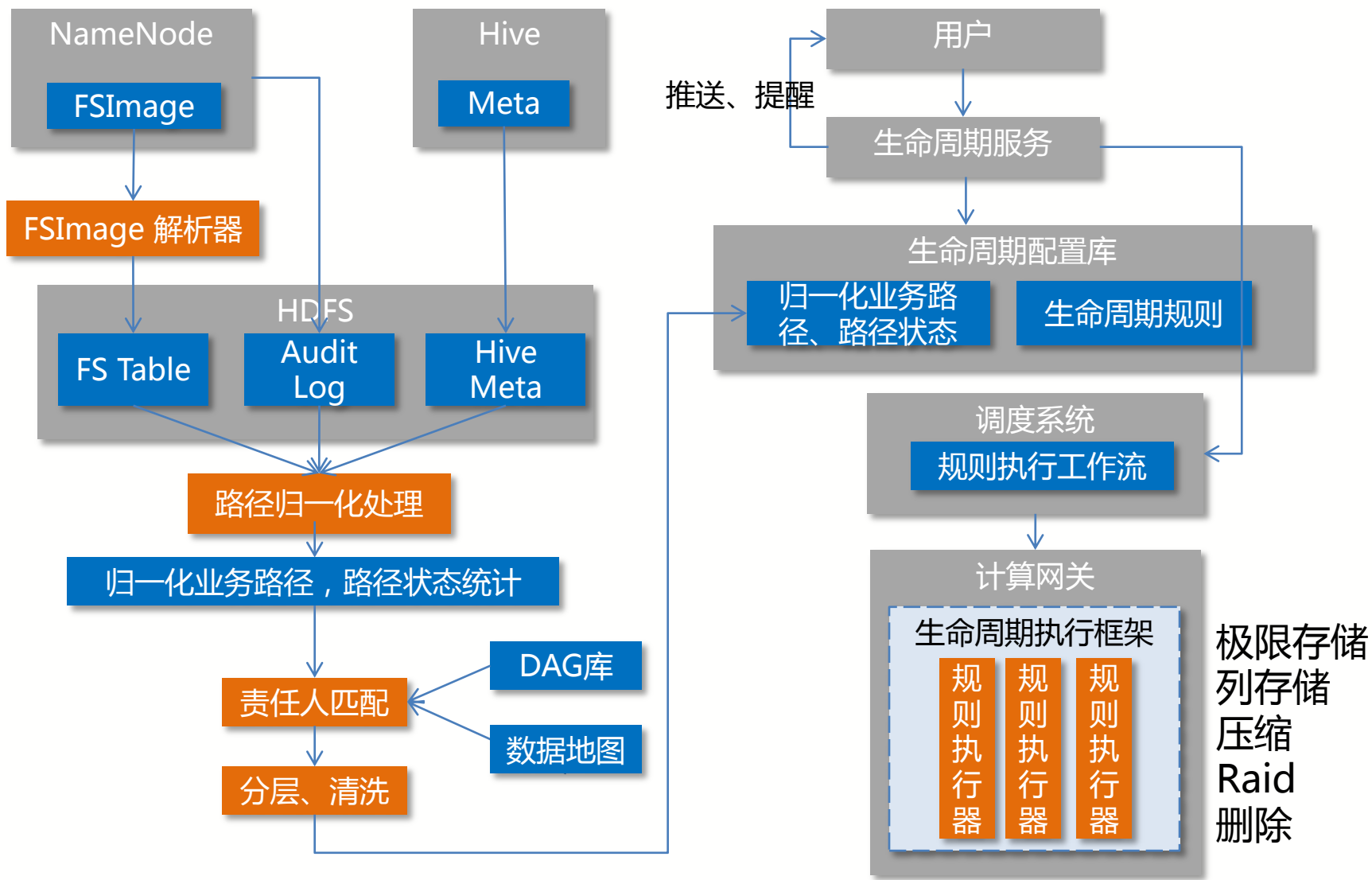


### 三个结论:

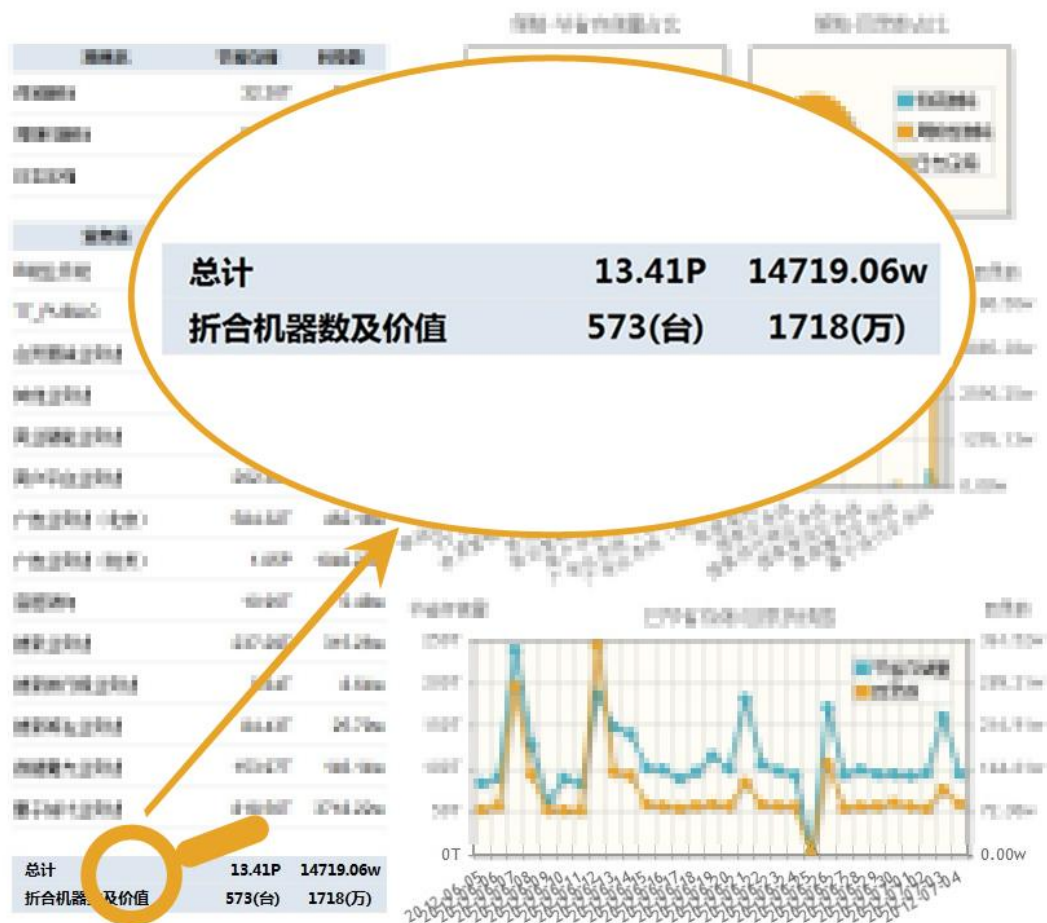
- ✓ 任意一条记录，由于其生命周期确定，必定对应唯一的一个数据标签
- ✓ 一个数据标签对应符合该生命周期的记录集合(该记录集合有为空的可能性)
- ✓ 历史上出现的所有记录，必然可以成功的划分到不同的生命周期数据标签里去

极限存储效果





生命周期管理效果数据：





### 问题

每天新增大量的作业？如何不断进行优化？  
数据架构如何适应业务的不断变化？

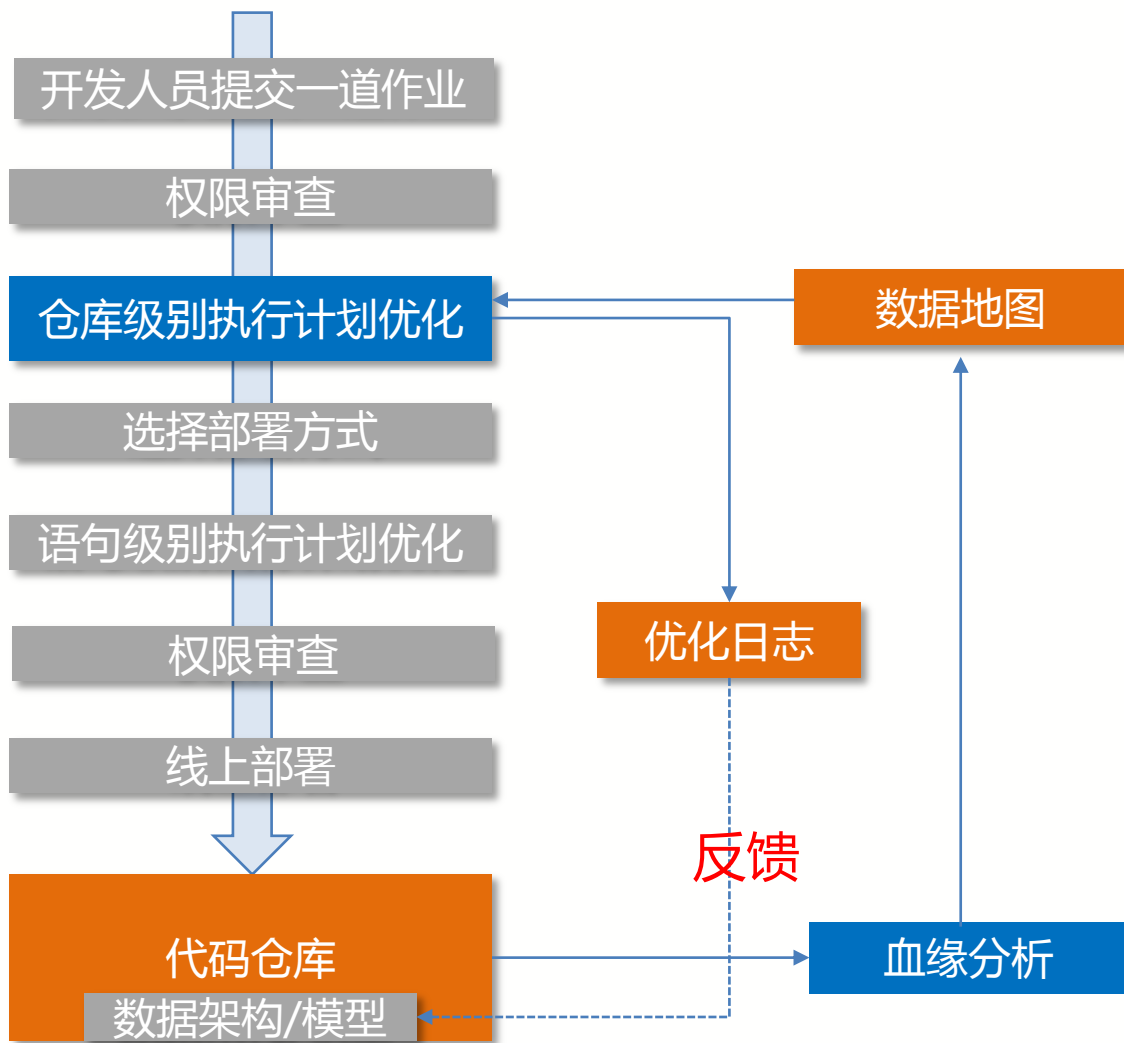
### 定义

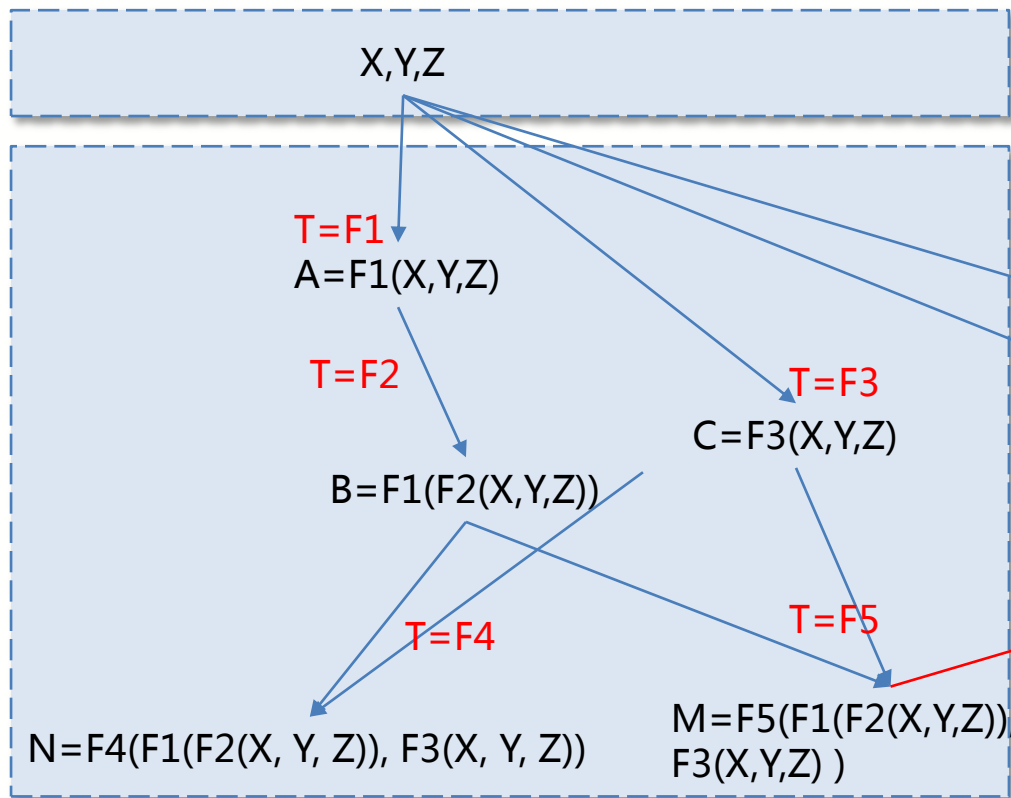
基于整个数据仓库的数据地图，寻找最优计算路径，  
对用提交的计算任务进行整体重写。

执行计划的优化命中情况，可以用来评估数据架构设计的优劣。  
并用来指导数据架构的优化。

	仓库级执行计划优化	语句级查询优化
粒度	工作流级别（包含一条或以上）	优化单条语句
方法	基于数据地图	基于物理存储
依赖	依赖于数据架构/模型	依赖于特定的数据库与存储引擎
效益	高	低

## 优化案例



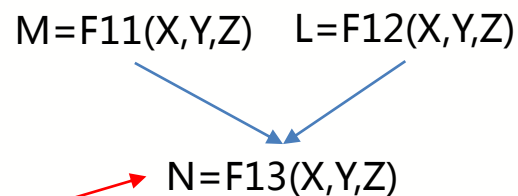


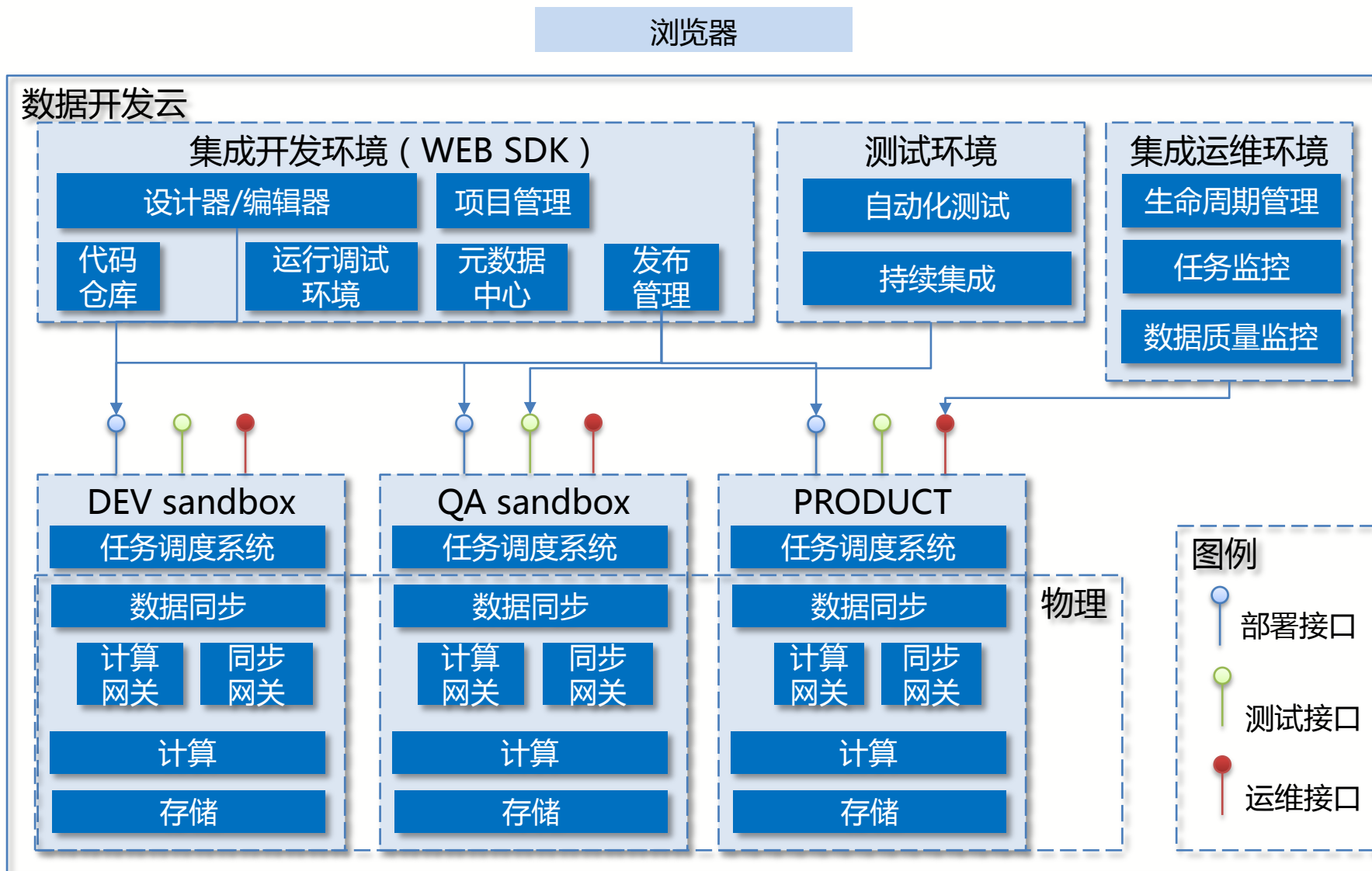
建立数据地图：

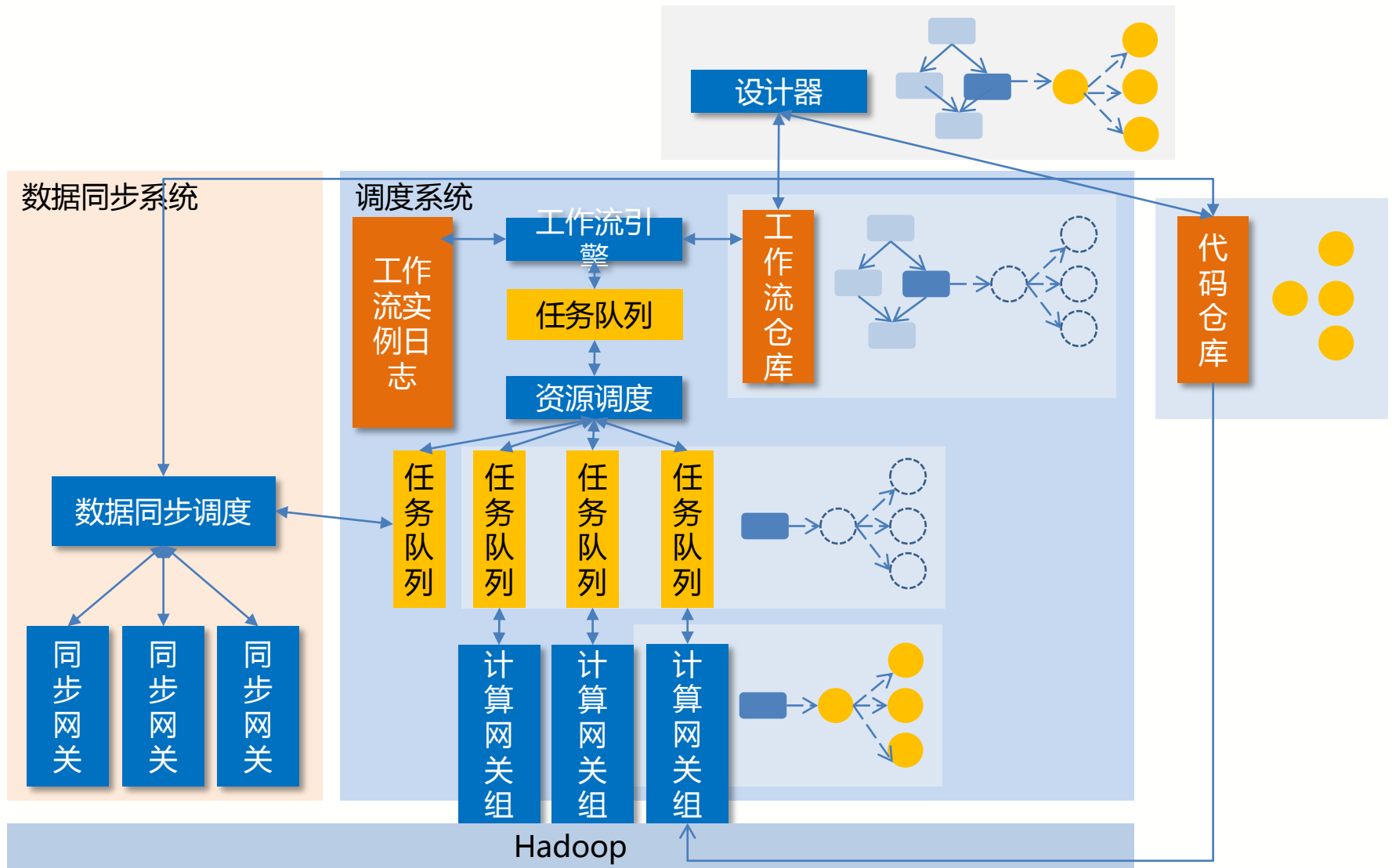
- 1 建立以ODS层字段表示字段级表达式
- 2 简化字段表达式
- 3 归一化字段表达式
- 4 计算字段距离

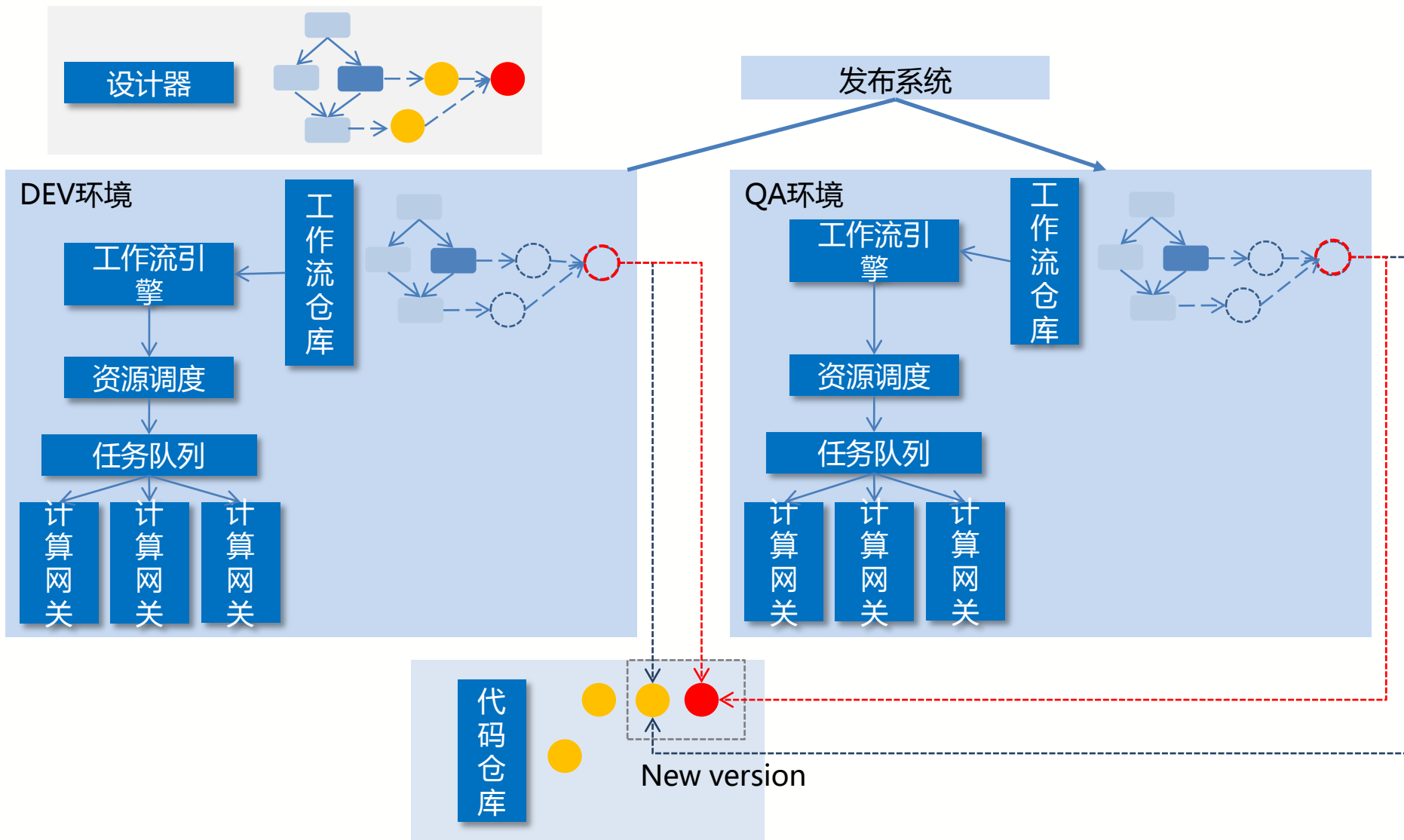
优化：

- 1 将用户作业转化为字段表达式
- 2 将字段定位到数据地图
- 3 重新选择计算路径









1 去除无用的  $N \rightarrow 0$

2 去除重复的  $N \rightarrow 1$



产品

架构

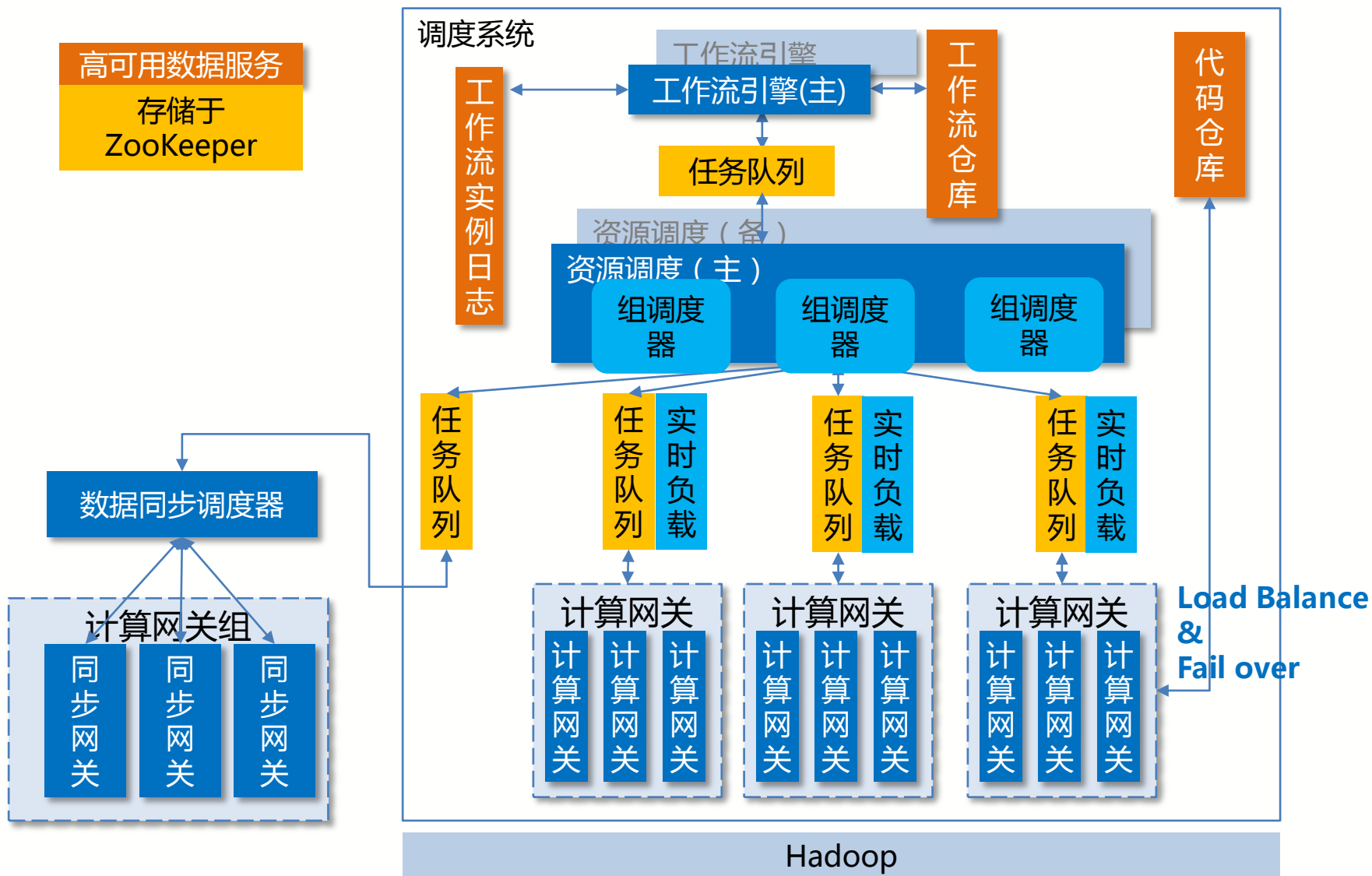
实践

成本

可用性

伸缩性

可扩展性



- 1 无状态，冗余
- 2 模块化，松耦合
- 3 故障隔离，最坏情况假设
- 4 自动化
- 5 完善的监控和告警
- 6 单元测试与持续集成

成本

可用性

伸缩性

可扩展性

产品

架构

实践

这一节被缩掉了....

产品

架构

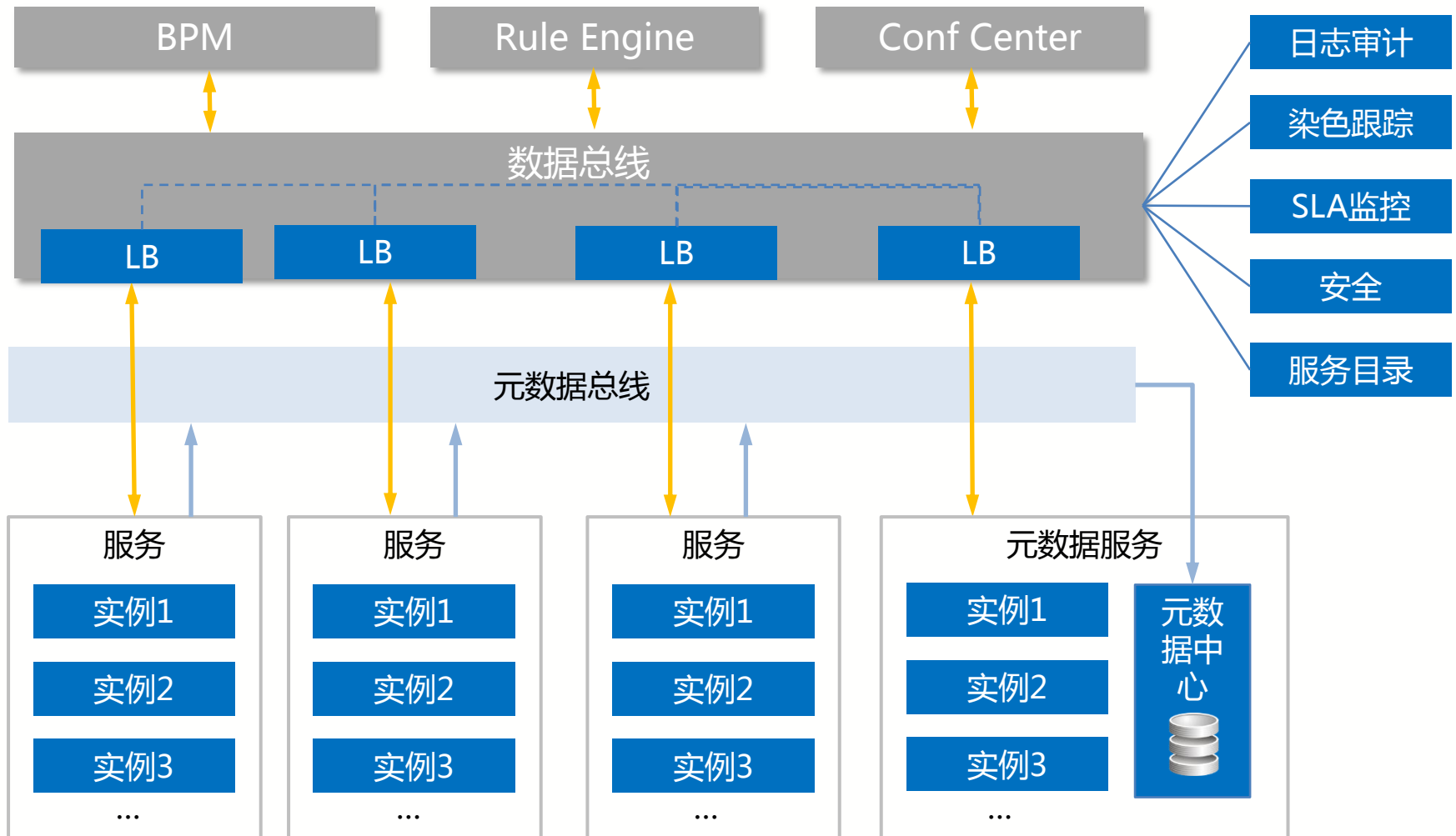
实践

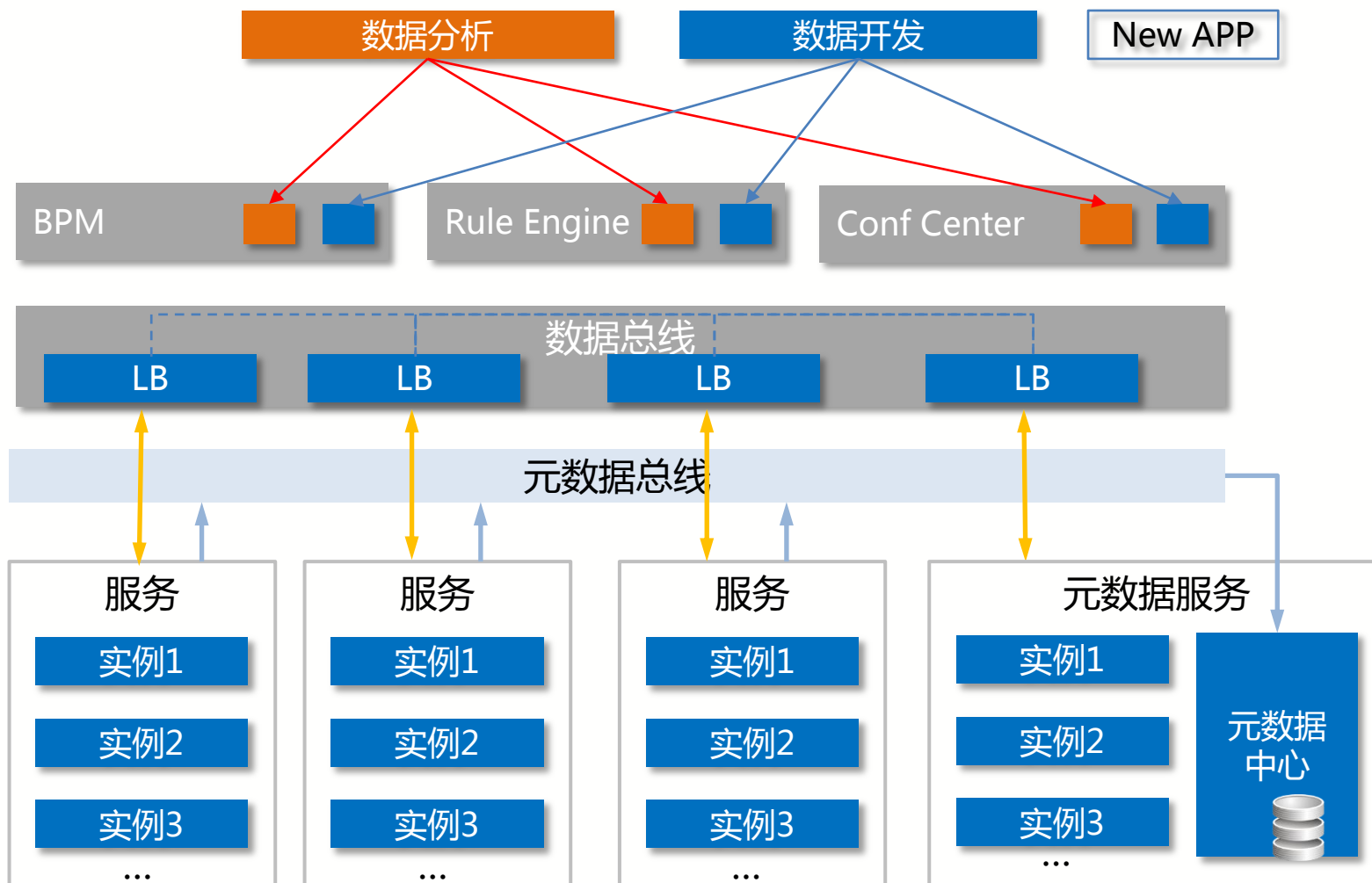
成本

伸缩性

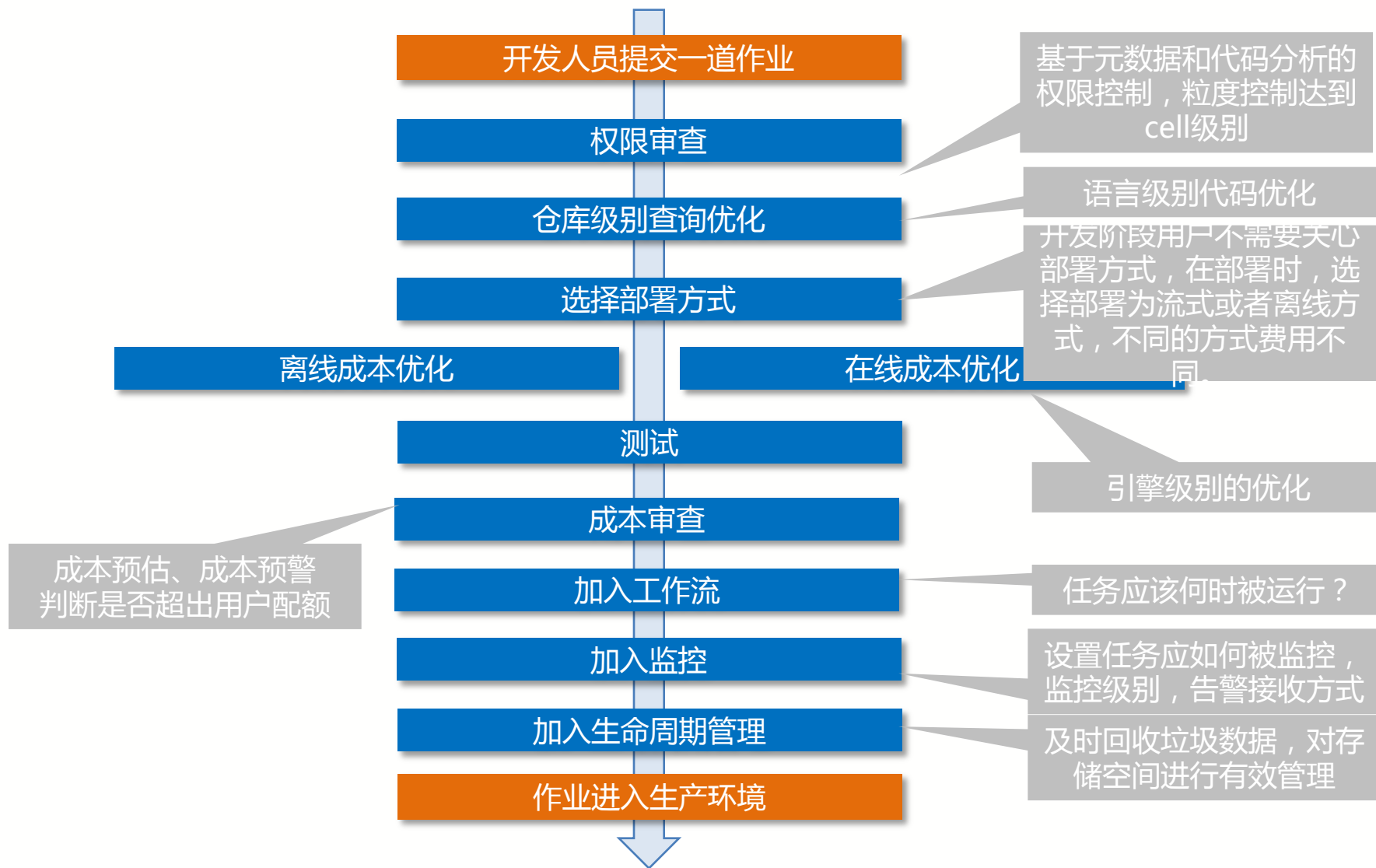
可用性

可扩展性

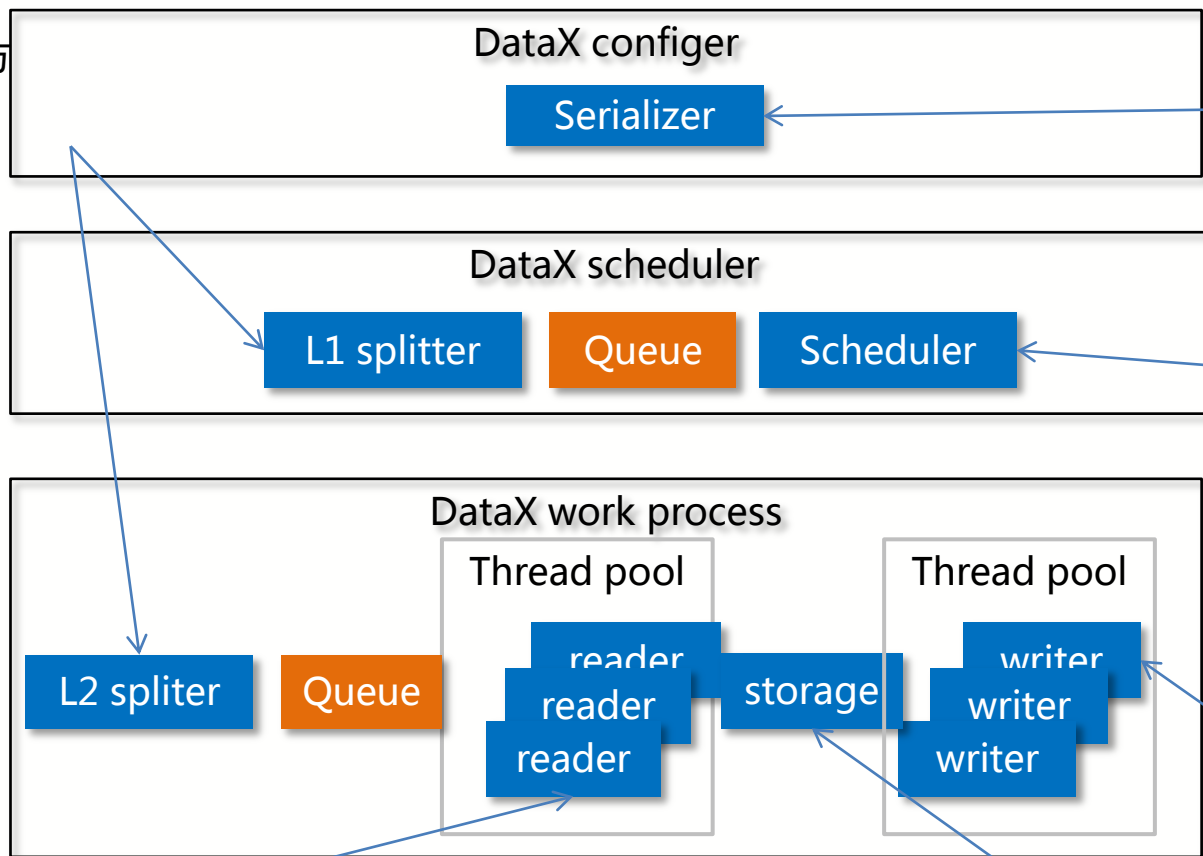








扩展的拆分方式：  
分库分表  
RowID拆分  
Region拆分  
时间拆



扩展不同配置持久化策略：

- 1 DB
- 2 XML
- 3 代码库

扩展不同调度策略：

- 静态负载均衡
- 动态负载均衡
- 网络位置感知

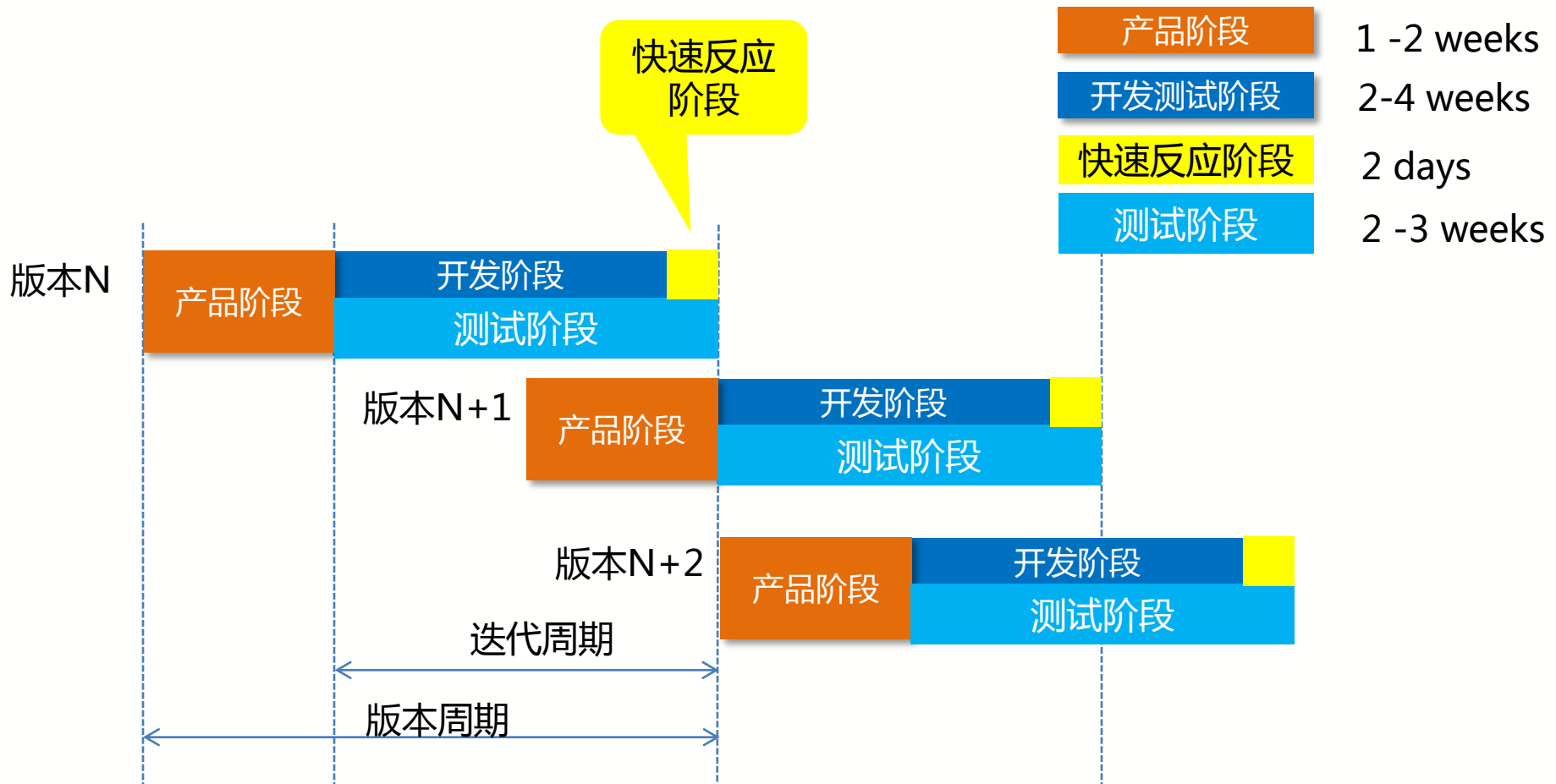
...

扩展不同的数据目标：

- Mysql/Oracle/Hbase/SQL server/Http/Local File...

扩展不同的数据源：  
Mysql/Oracle/Hbase/SQL server/Http/Local File...

扩展不同的中间管道：  
Double Buffered Queue  
FS  
Mysql  
Hbase



- 业务专家&产品专家
- 平衡：有限可扩展
- OOP、OCP、Design Pattern
- 分离易变部分和稳定部分
- 隔离变化：interface & Mediator
- 推迟实例化、动态绑定
- 敏捷开发模式有助于培养可扩展思维

产品

架构

成本

可用性

伸缩性

可扩展性

实践

谢谢

更多细节交流：淘宝泽远@新浪微博

会场用餐自理，以下用餐场所供大家参考

海外海皇冠假日酒店.主楼

名人名家餐厅：二楼（包厢）、三楼（大厅）

西餐厅：四楼

海外海酒店马路对面有各色小吃店

出酒店左拐直走10分钟左右，胜利河美食街

**祝您用餐愉快！**

