

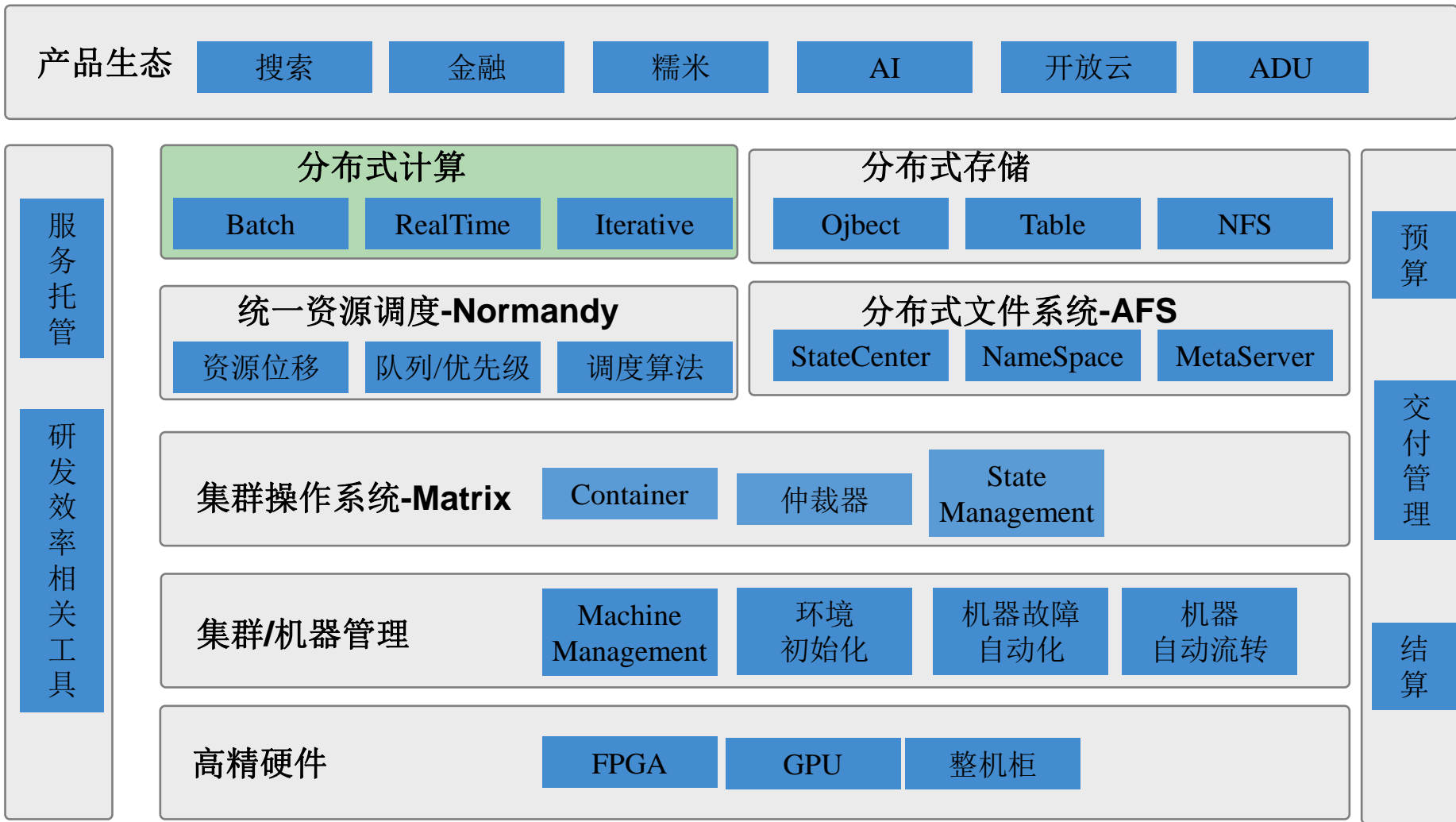


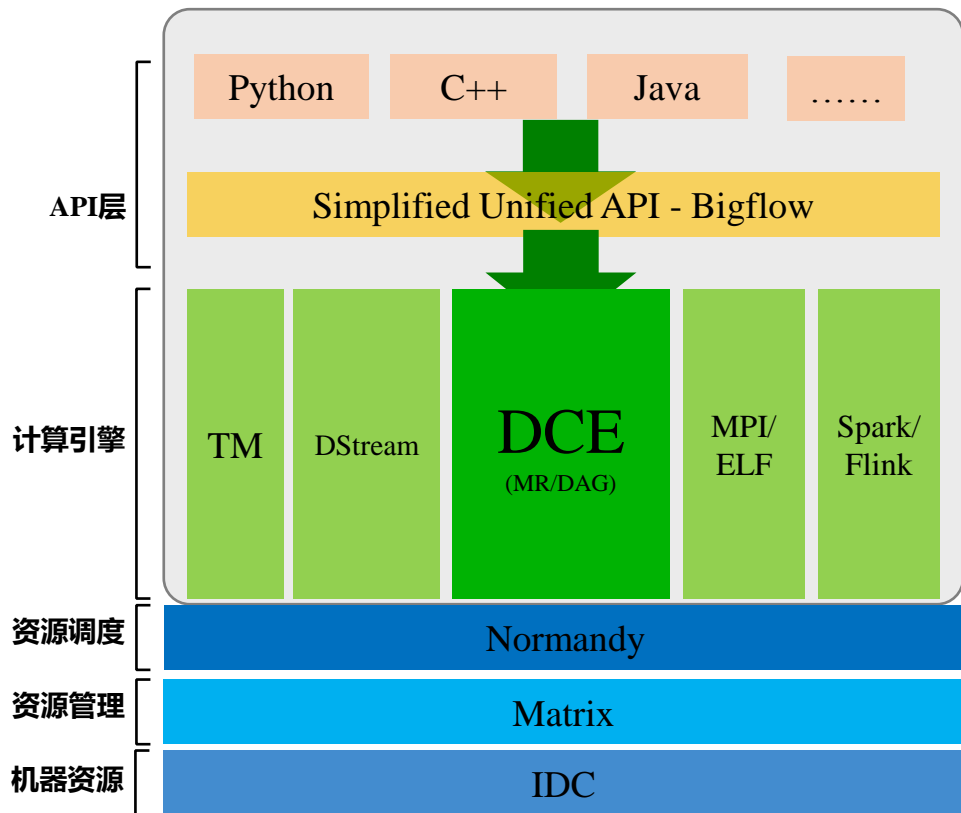
百度大数据离线计算平台发展历程

百度 黄鑫

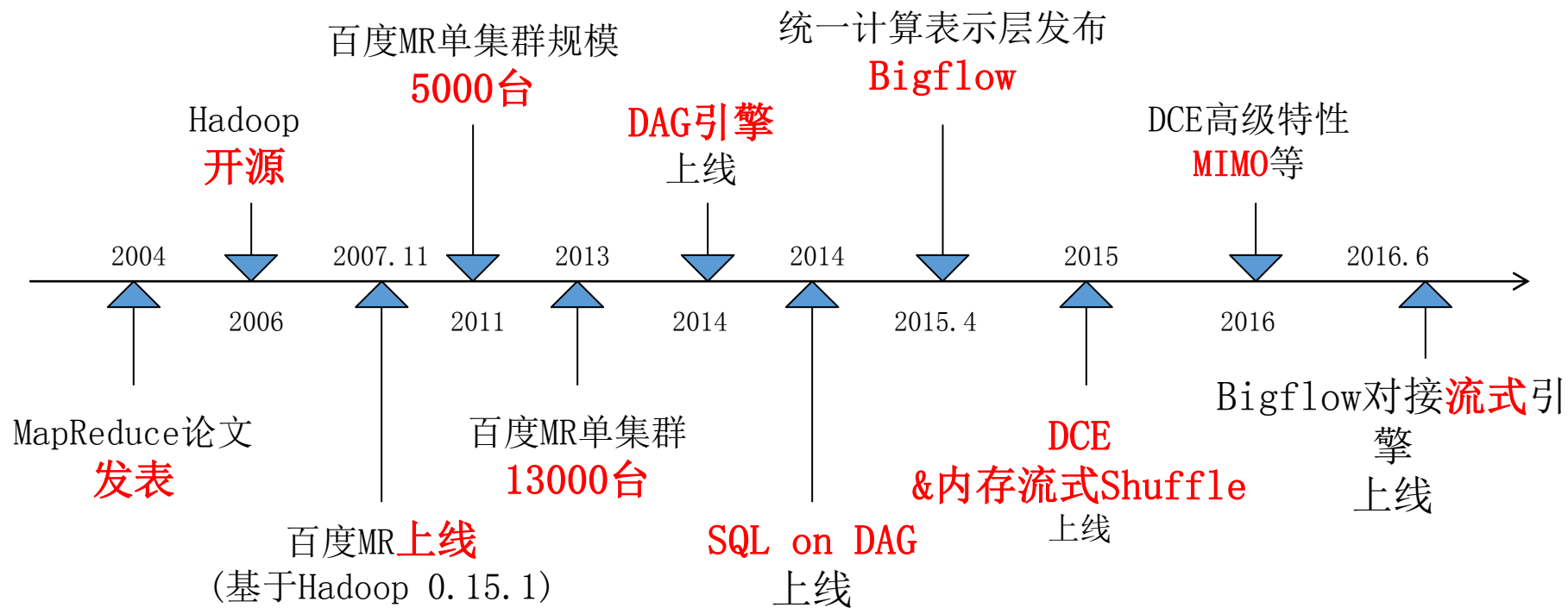
- 百度离线大数据平台发展历程
- 离线计算引擎DCE揭秘
- 统一分布式计算API-Bigflow

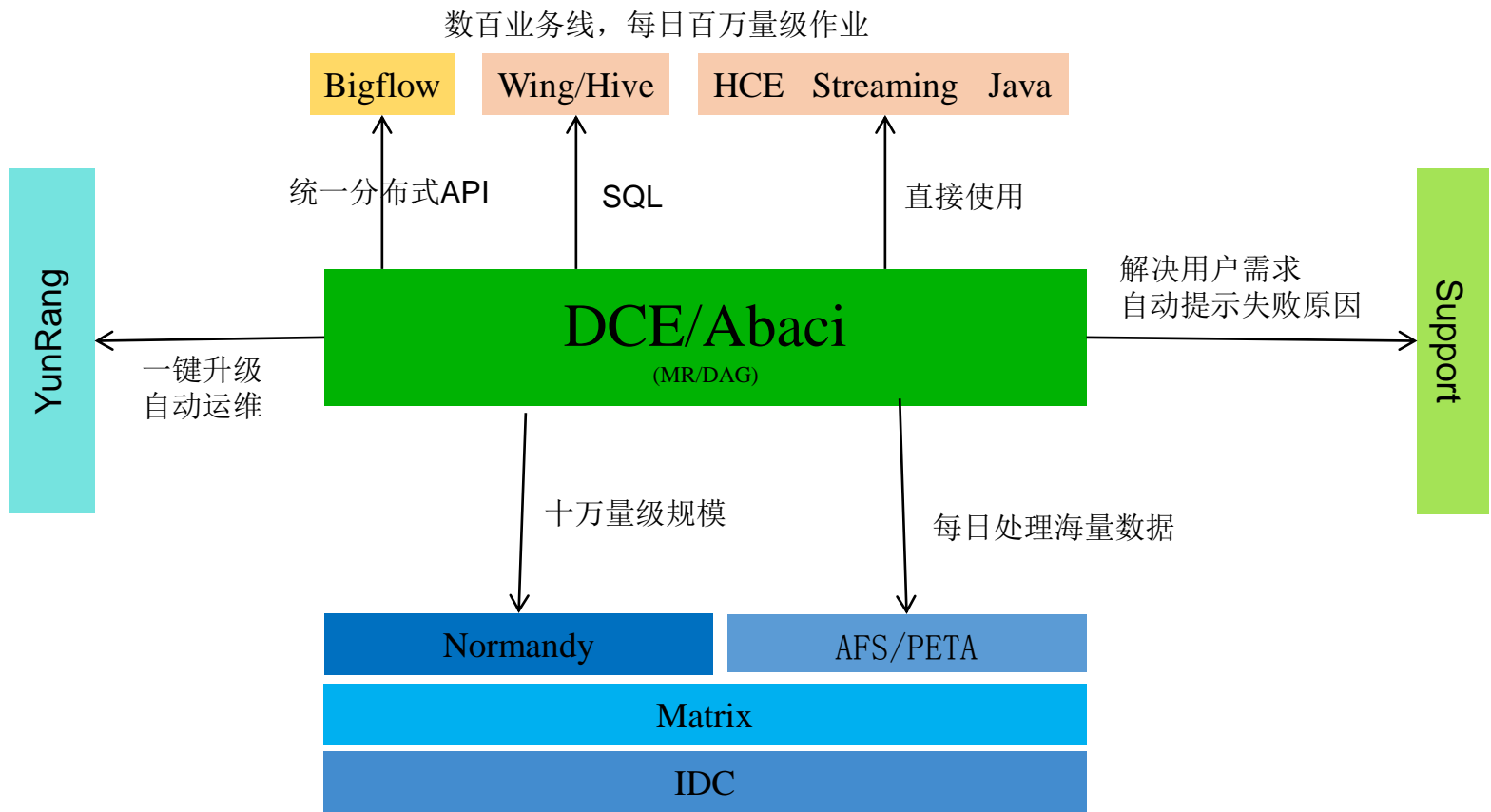
- 百度离线大数据平台发展历程
- 离线计算引擎DCE揭秘
- 统一分布式计算API-Bigflow



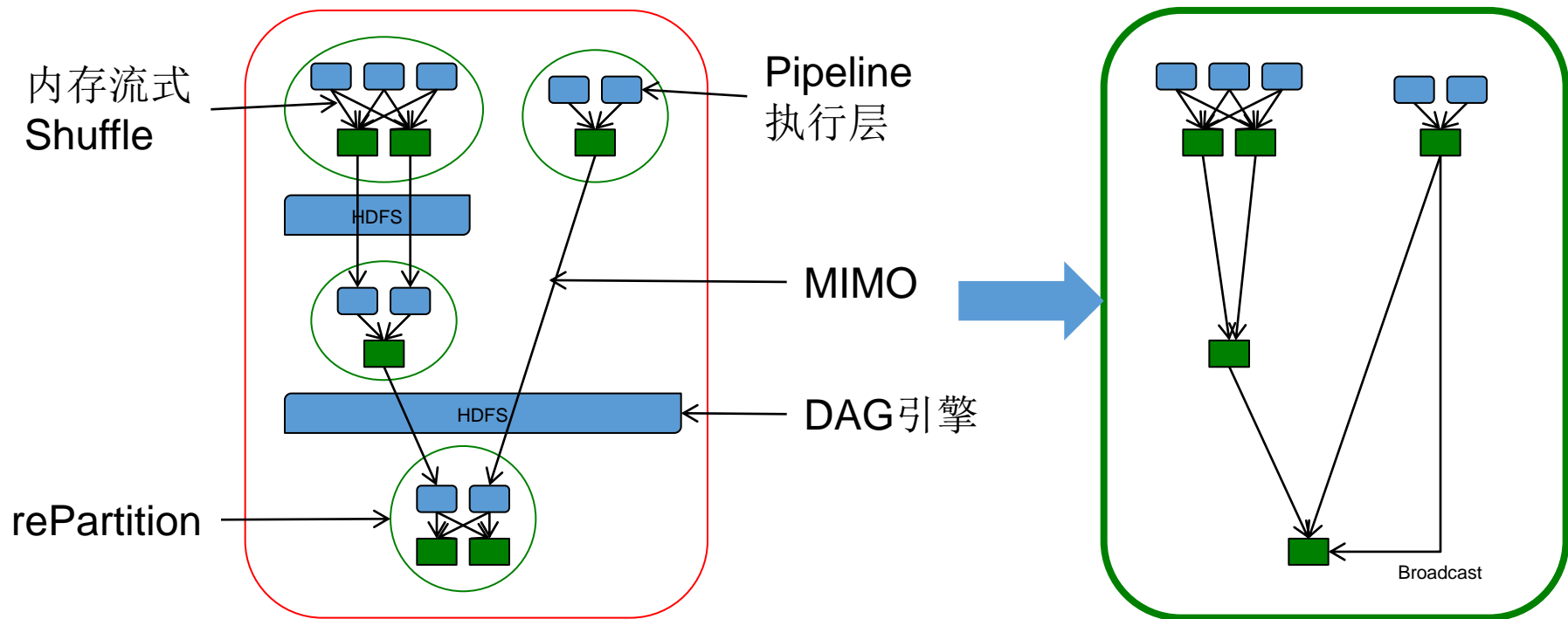


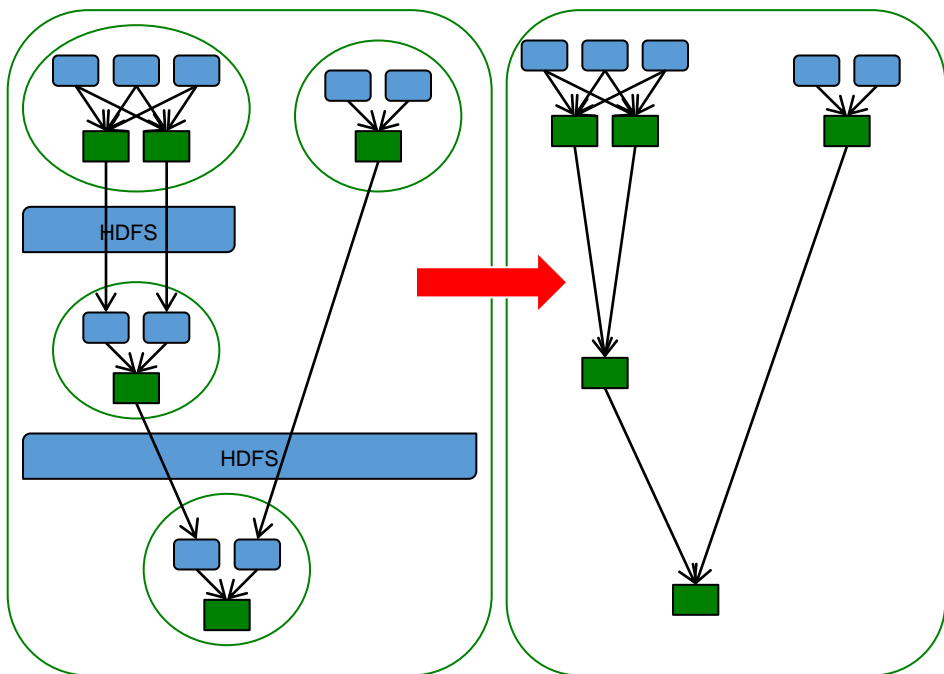
百度大数据离线计算平台发展历程





- 百度离线大数据平台发展历程
- 离线计算引擎DCE揭秘
- 统一分布式计算引擎API

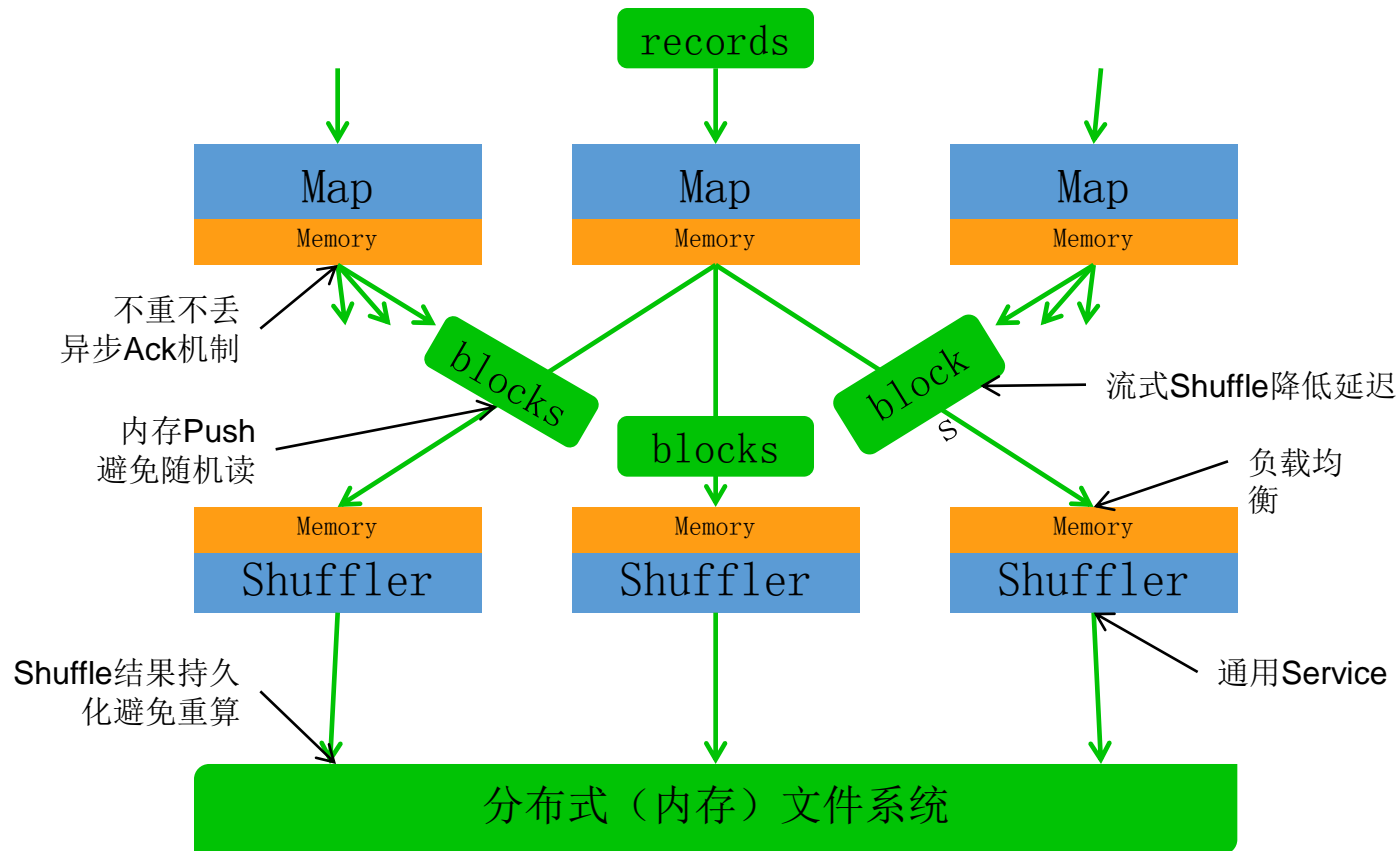




优化计算模型
避免MR作业间IO读写。

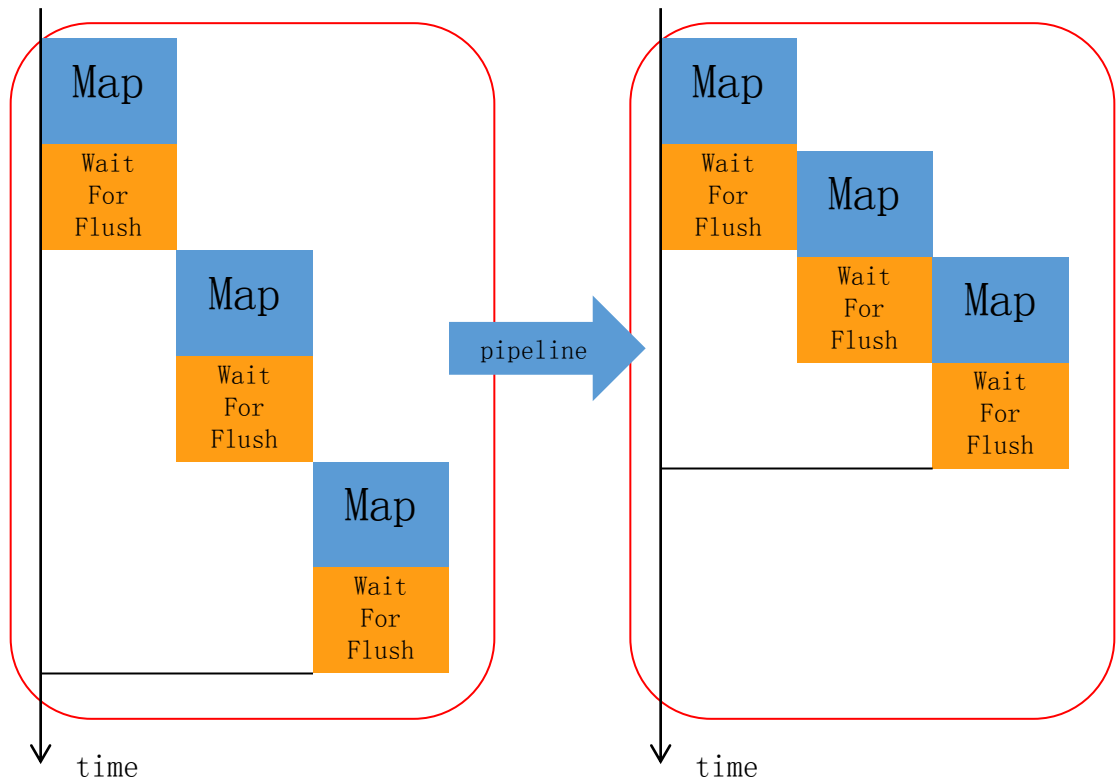


基于MR引擎，需要翻译成 **25个** MR Job
基于DAG引擎，只需要翻译成 **1个** DAG Job
运行时间从 **5小时** 缩减到 **1小时**



优势：
减少IO
减少随机读
避免重算

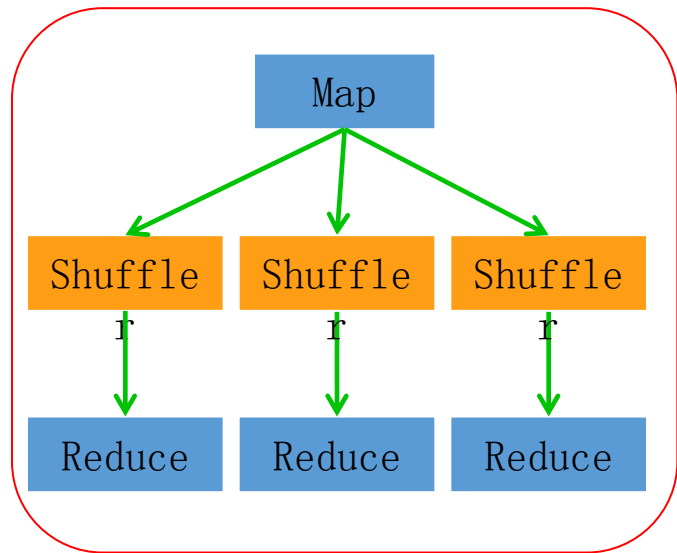
Map Pipeline



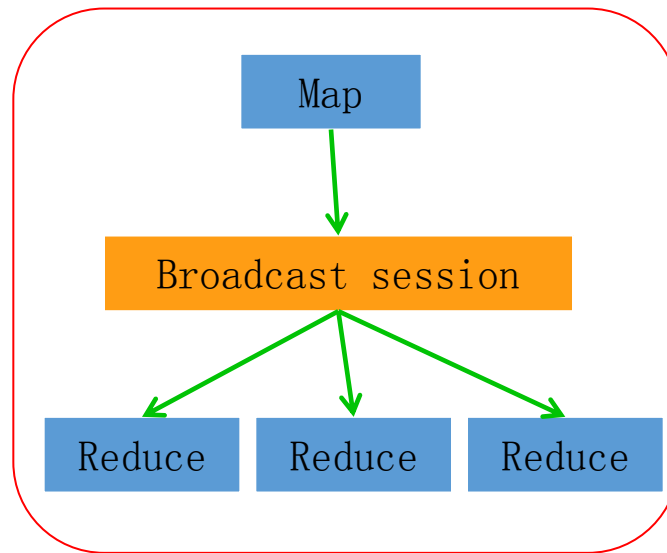
借鉴CPU流水线
资源使用更平稳
运行时间缩短

MIMO (Multi-Input Multi-Output)

Normal Shuffle

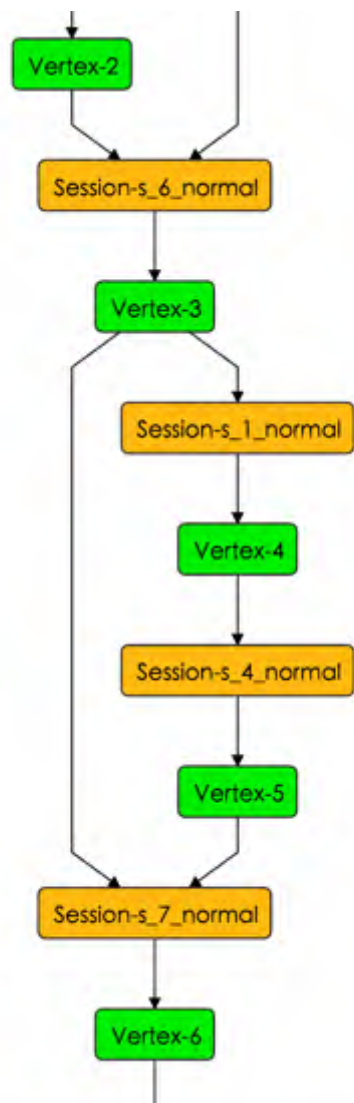


Broadcast



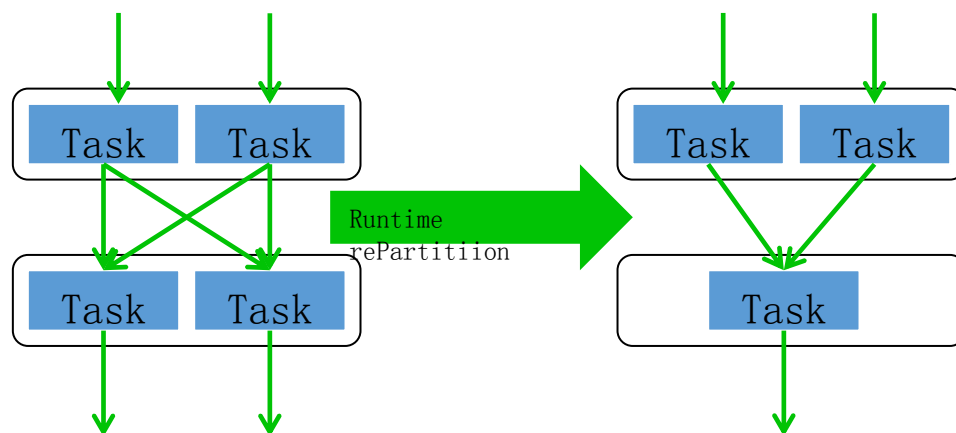
多种数据传输模式

多下游共用数据



数据传输方式可以任意定制
高效、通用的DAG引擎

Runtime rePartition



运行期间，
动态调整Parittion规则

收益

流式shuffle:

减少旧shuffle map merge、
reduce pull时间消耗

内存Push:

map端不落盘

Shuffler内存聚合:

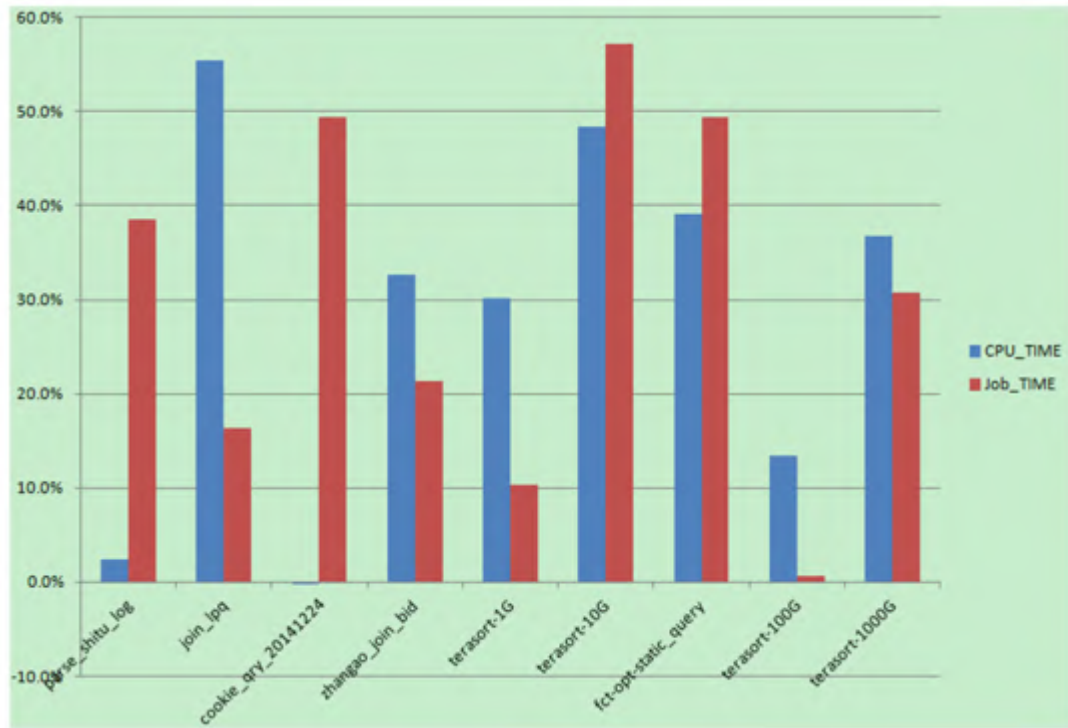
聚合度高, 减少map端seek, 减少
reduce端merge路数, 减少IO

Pipeline:

大大提高中小作业map端运行速

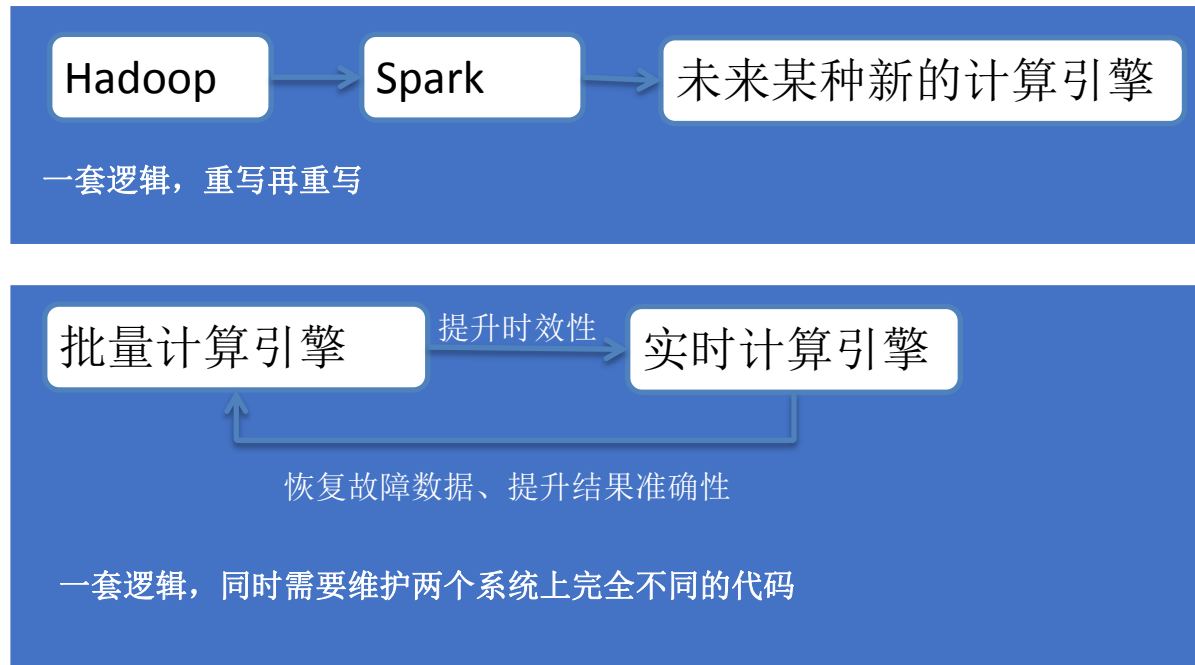
中间数据持久化:

避免重算 (对dag作业尤为重要)



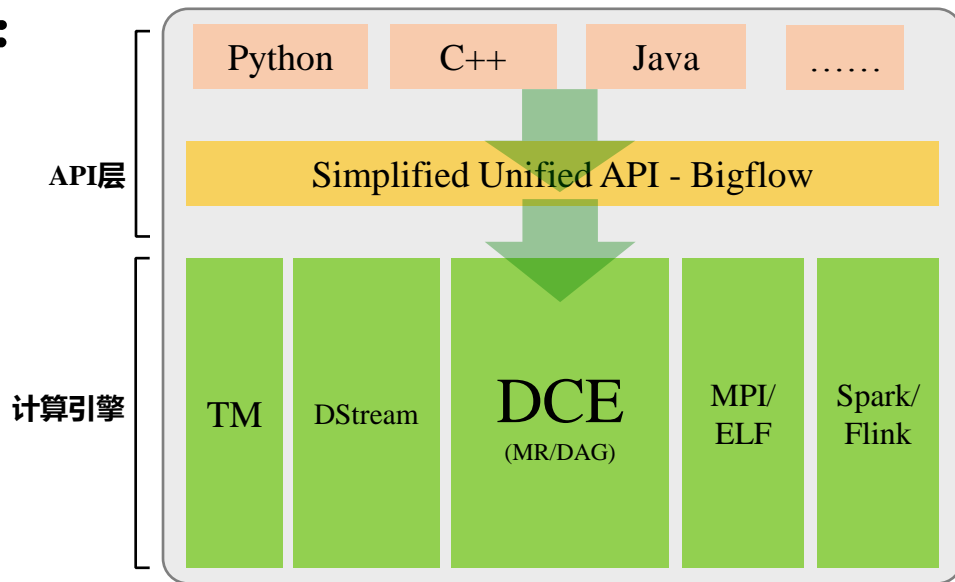
- 百度离线大数据平台发展历程
- 离线计算引擎DCE揭秘
- 统一分布式计算引擎API-Bigflow

- **学习成本:**
 - 学习使用、学习优化
- **迁移成本:**
 - 单机作业迁移分布式、流式与批量迁移、新引擎迁移
- **维护成本:**
 - 用户作业维护、引擎演化兼容维护



- 统一分布式计算API-Bigflow:

- 统一流式和批处理计算模型
- 自动优化用户代码
- 针对引擎特性，进一步优化执行
- 简单易学，高层抽象API



从点击衍生数据 (<地区, 用户>) 对每个地区的用户求UV

Bigflow示例:

全白盒

```
def uv(x): #x : PCollection<str>
    return x.distinct().count()
grouped_uv = data.group_by_key()\
    .apply_values(uv)\
```

Spark示例:

黑盒

```
def uv(x): #x : list<str>
    return len(set(x))
grouped_uv = data.groupByKey()\
    .mapValues(lambda x:uv(x))\
```

1. 提出了分布式可嵌套数据集 (NDD) 模型, 相比于业界同类系统抽象程度更高。
2. 对接了多种计算引擎, 包括批量引擎、迭代引擎、流式引擎, 方便用户切换执行引擎。
3. 完成了许多优化策略, 使得Bigflow可以高效运行。
4. 我们在线上大规模验证了Bigflow的可嵌套数据集模型确实可以起到统一多平台的目标

成功对接多种批量、迭代、流式引擎

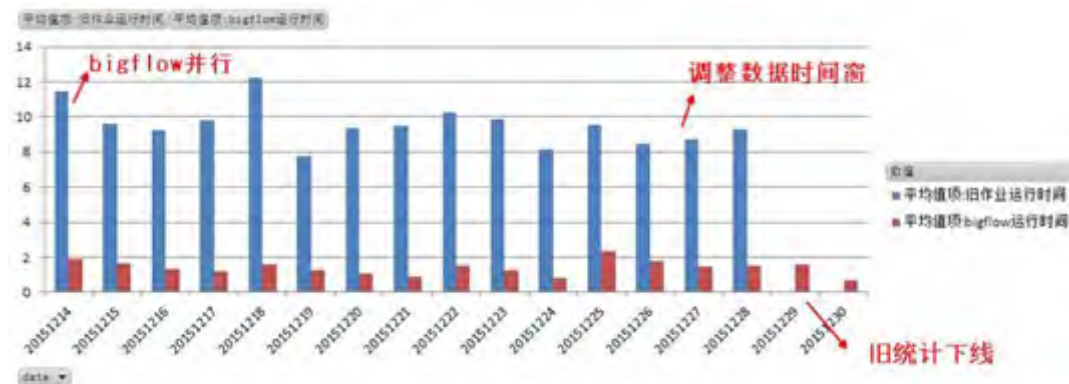
比直接使用底层引擎接口性能平均高100%+

用户代码平均减少60%

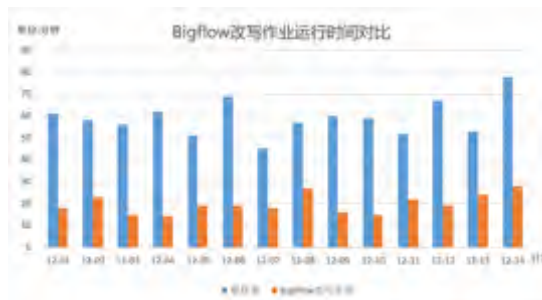
近百产品线

数百活跃用户

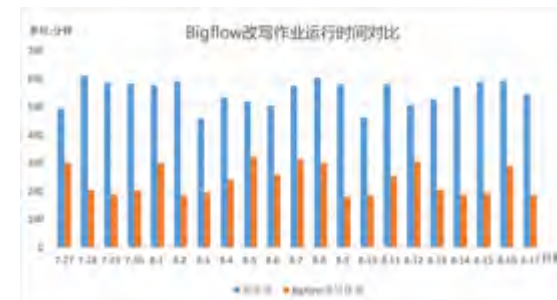
日处理数据量2.5P



凤巢某作业运行时间对比图



凤巢另一项目作业运行时间对比图



复合搜索某项目作业运行时间对比图

THANK YOU

cloud.baidu.com