

ArchSummit全球架构师峰会深圳站2016

推送背后的大数据系统

袁凯@个推



促进软件开发领域知识与创新的传播



关注InfoQ官方微信
及时获取ArchSummit
大会演讲信息



全球软件开发大会

[上海站] 2016年10月20-22日

咨询热线: 010-64738142



全球架构师峰会 2016

[北京站] 2016年12月2-3日

咨询热线: 010-89880682

主要内容

- 个推推送业务数据背景
- 我们面临的挑战
- 大数据系统演进之路
- 经验与总结
- Q&A

个推推送业务数据背景 (2016.06)

- SDK累计安装覆盖量达90亿 (含海外)
- 接入应用超过43万
- 接入开发者超过22万
- 独立终端覆盖超过10亿 (含海外)
- 日均活跃用户近7亿
- 同时在线超过3.5亿
- 日分发消息29亿条



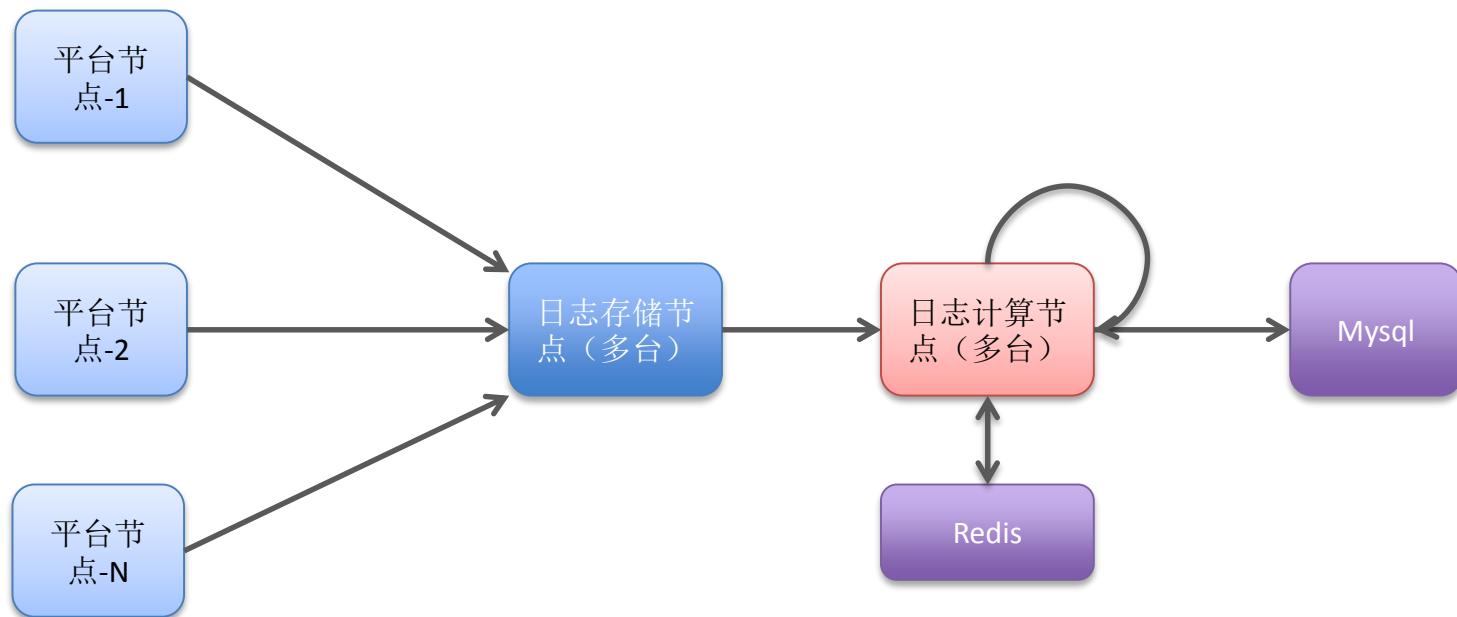
我们面临的挑战

- 数据存储(PB级别)
- 日志传输(10TB+/day、实时)
- 日志分析处理(日常定时执行、交互式)
- 大量任务调度和管理(2K+ jobs/day)
- 数据分析处理服务高可用
- 海量多维度报表(推送、用户等)
- 用户画像(标签系统等)
- 快速响应分析和取数需求(开发、分析、运营人员等许多业务方)

个推大数据系统演进之路



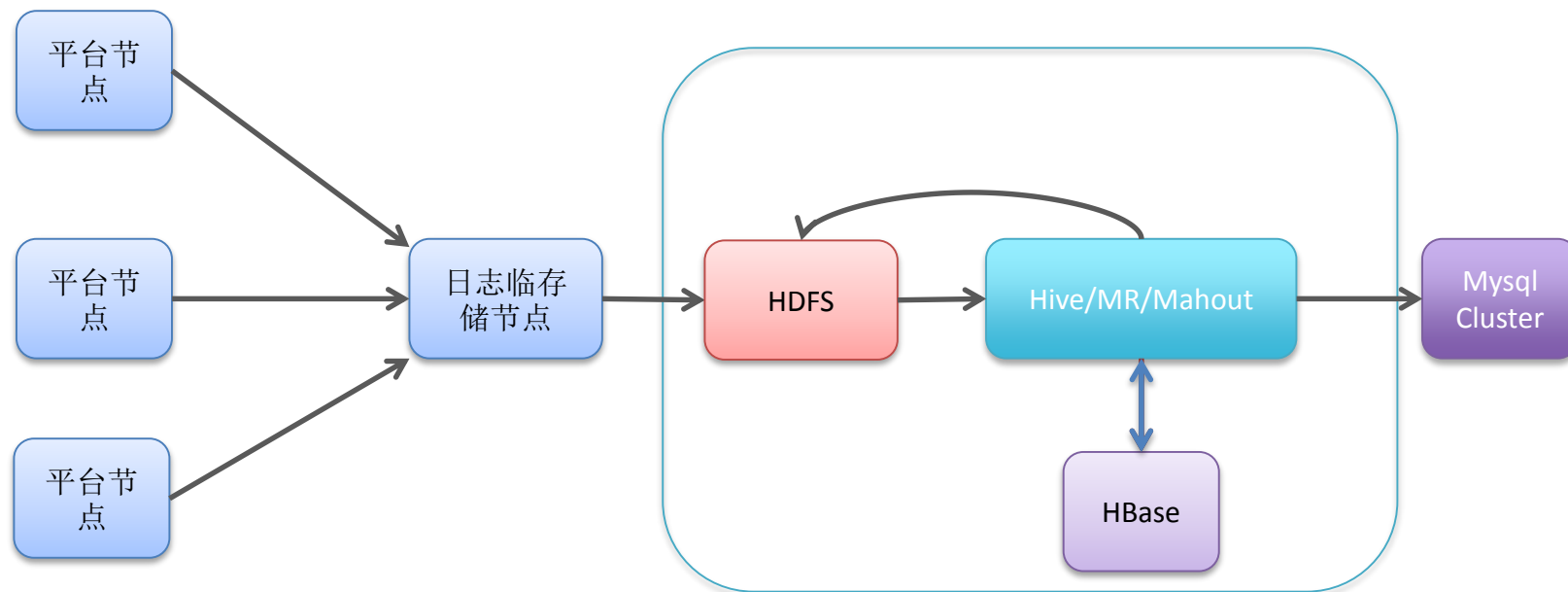
演进之——统计报表计算



业务特点：

- 传输上只需要运维定时脚本传输到指定中间节点
- 用户虽然有亿级别但是日志种类比较单一
- 统计处理需求只需要使用简单shell、php脚本采用多进程模式运行在多台高配置的主机上即可满足。
- 数据只是需要保存短期（结果集长期保存、中间数据和原始数据很短时间）。

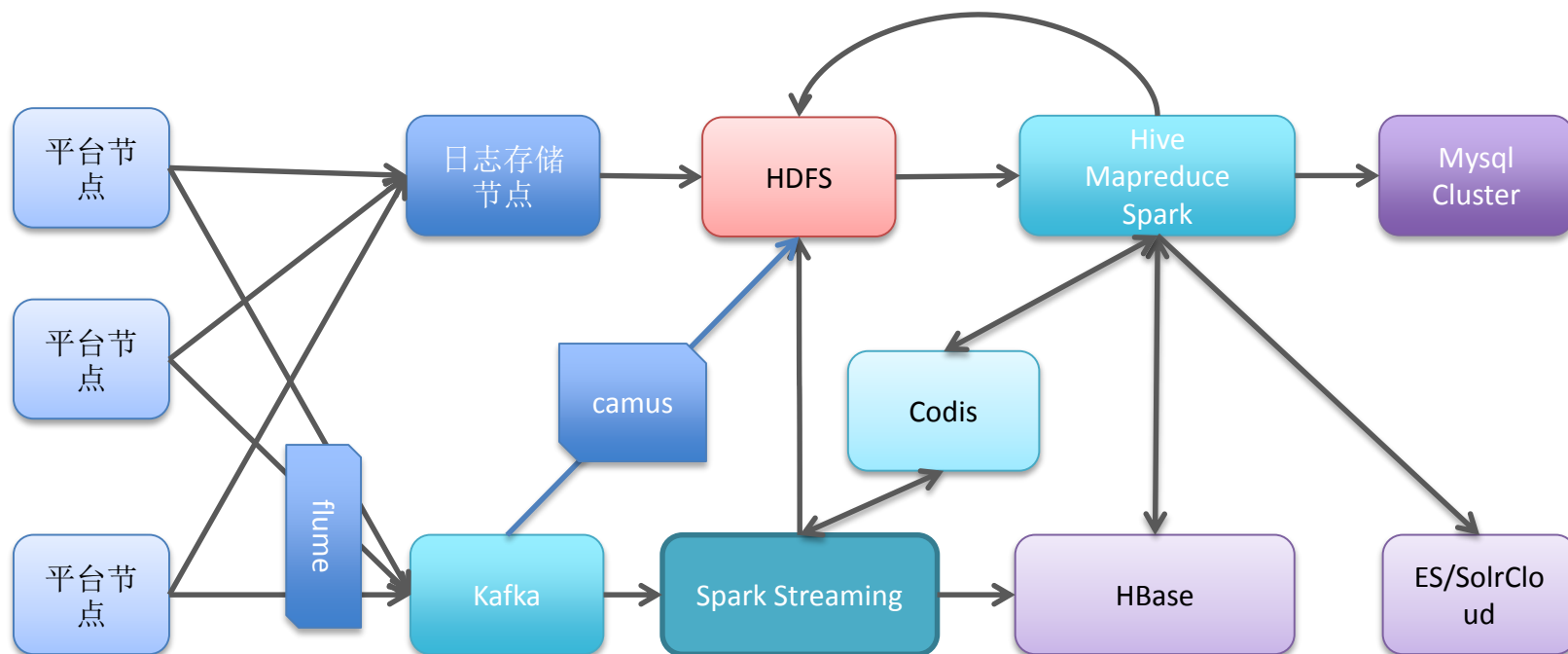
演进之——大数据基础建设：离线批处理系统



业务特点：

- 用户数量暴增，客户报表和内部统计分析需求复杂（海量、多维度）。
- 很多需求需要在T+1时间内满足（用户标签更新、日常取数、报表等）。
- 数据需要保存更长的周期（挖掘、统计）。

演进之——大数据基础建设：实时处理系统



业务特点：

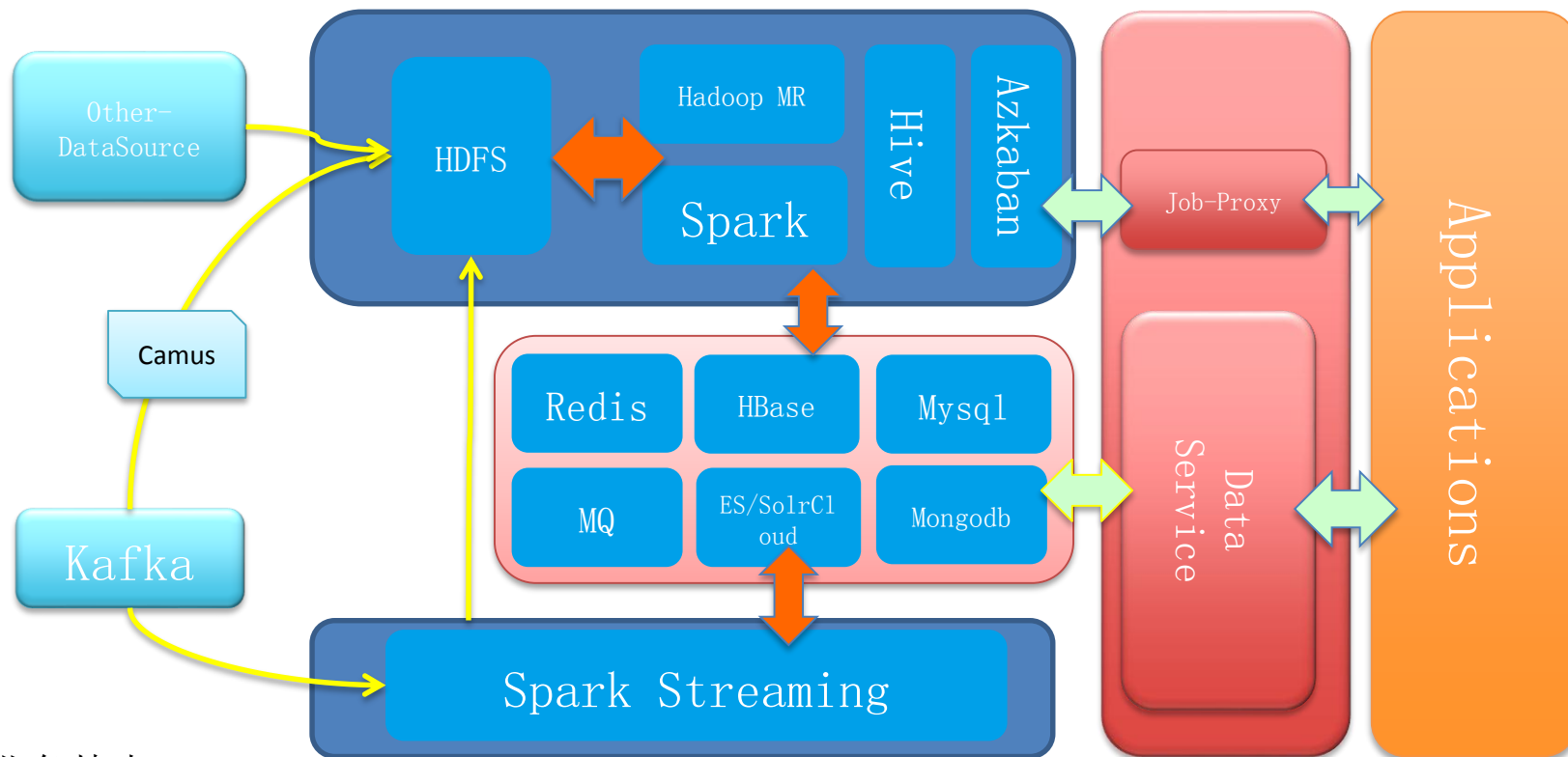
- 部分大业务需要T+0时间内满足（报表、事件分析等）。
- 实时数据产品需求（内部工具、热力图等实时统计分析）。
- 机器学习需求（数据挖掘、画像系统等）。
- 快速查询检索（工具箱等产品）

主要完成了以下基础建设工作：

- 采用Lambda架构（Batch Layer、Speed Layer、ServingLayer）
- 引入Hadoop（Hdfs、Hive/MR、Hbase、Mahout等）。
- 引入Spark技术栈（复杂迭代计算、机器学习、SQL、流式计算）。
- 采用ES、SolrCloud+ HBase方案 实现多维度检索。
- Flume 、Kafka、Camus引入和优化改造日志传输。
- 国产开源的Redis集群方案-Codis 引入和优化。



演进之——工具化+服务化+产品化

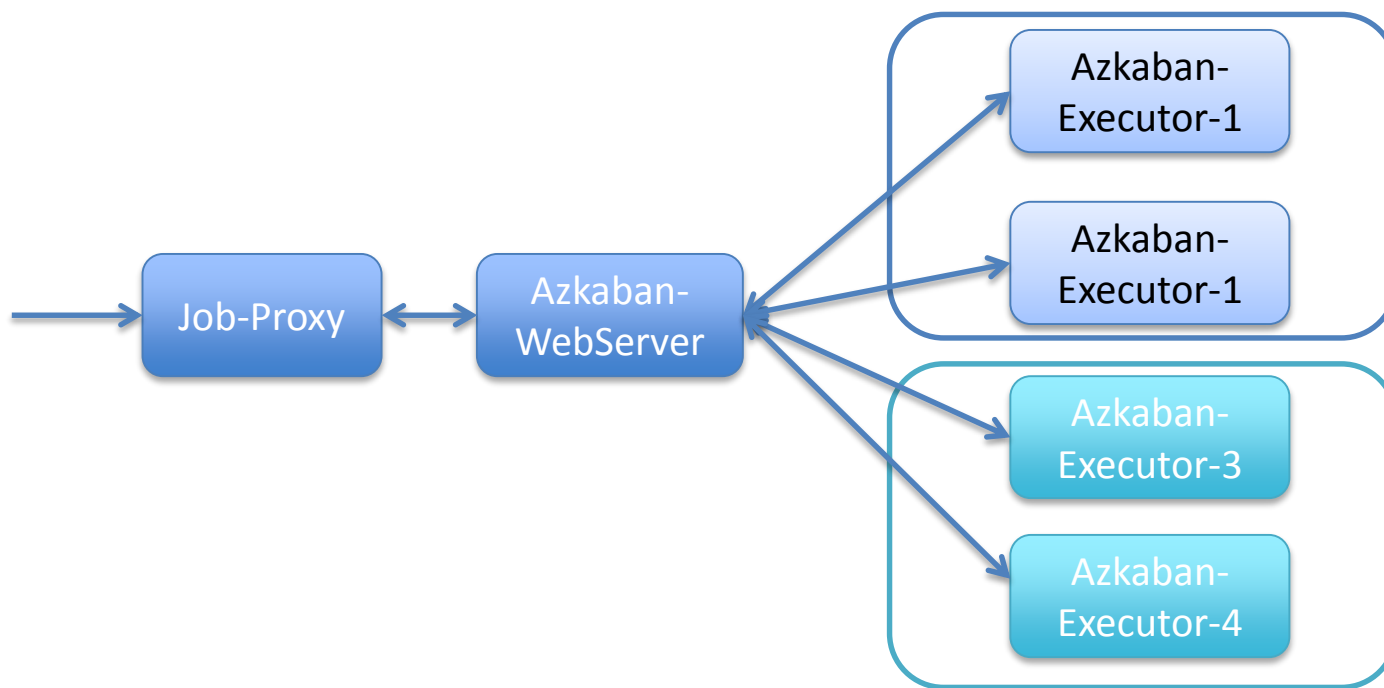


业务特点：

- 服务的业务人员增多（产品运营、运维、分析人员、市场等）。
- 线上对数据系统依赖增多。
- 大量的任务运行在集群上，优先级不一样。
- 相同数据支持多类业务。

主要完成了以下工作：

- 增加Job调度管理：引入Azkaban和进行改造（变量共享、多集群支持等）。
- 增加服务代理层：引入DataService和Job Proxy(开放给更多产品线使用并解耦)。
- 增加应用层：基于服务代理层研发相应的工具和取数产品。



经验与总结

- 探索数据和理解数据是开发前必备工作
 1. 脏数据种类和分布（无效数据和缺省情况的发现等）
 2. 数据各个维度分布（数据倾斜的发现和资源评估等）

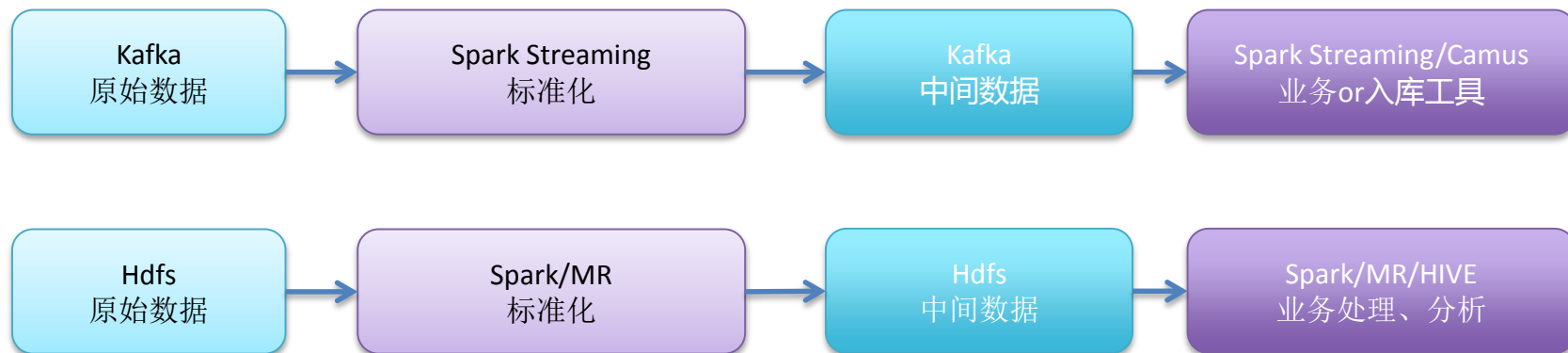
经验与总结

- 存储方案向分析和计算需要靠拢
 1. 数据合理分区，时间维度、地理纬度、冷热温等规则分区
 2. 数据列式存储提高压缩率和减少不必要的IO（Parquet，ORC等）
 3. 考虑使用类似Carbondata等带有索引的文件格式

经验与总结

- 数据标准化是提高后续处理首要手段

绝大部分数据需要标准化后供给后续使用（基本清洗、统一内部ID、增加必备属性）



经验与总结

- 工具化、服务化、产品化提高整体效率
 1. 开发层面将MR、Spark进行进一步API封装并且提供足够的工具包。
 2. 常规操作做成工具（web类或脚本类），将取数规则抽离为取数模板，让业务方在web端填写参数自助取数, 减少数据分析人员的手工提数。
 3. 将业务抽象为统一取数服务，开发服务API给到业务部门进行快速简单的开发，实现调用集群服务。

经验与总结

- 大数据系统监控很重要
 1. 集群侧：内存、CPU、磁盘、带宽
 2. 数据源采集侧：日志生成节点压力、flume状况监控
 3. 数据传输：采集侧和Kafka、HDFS条目数一致性实时监控、Kafka分片均匀情况
 4. 批处理监控：日常任务运行时间监控、是否出现倾斜、结果集每日曲线、异常数据曲线，GC监控
 5. 流式处理监控：原数据波动监控、消费速率监控报警、计算节点delay监控、GC监控
 6. 数据仓库监控：数据各个维度分布监控，数据准确性监控

经验与总结

- 保持技术栈单一性，不要扩展太多
 1. 大数据处理分析技术很多，精力有限，需要考虑团队学习成本
 2. 系统中技术栈太宽泛不易运维
 3. 系统实体太多稳定性不易控制

Q&A

Thanks!

