Flink在多中心/边缘计算上的实践

陈仕明 · 虎牙 / 数据平台负责人

Kafka X Flink Meetup 深圳 - 2019年08月31日

Contents

目录 >>

01 实时计算在虎牙的发展

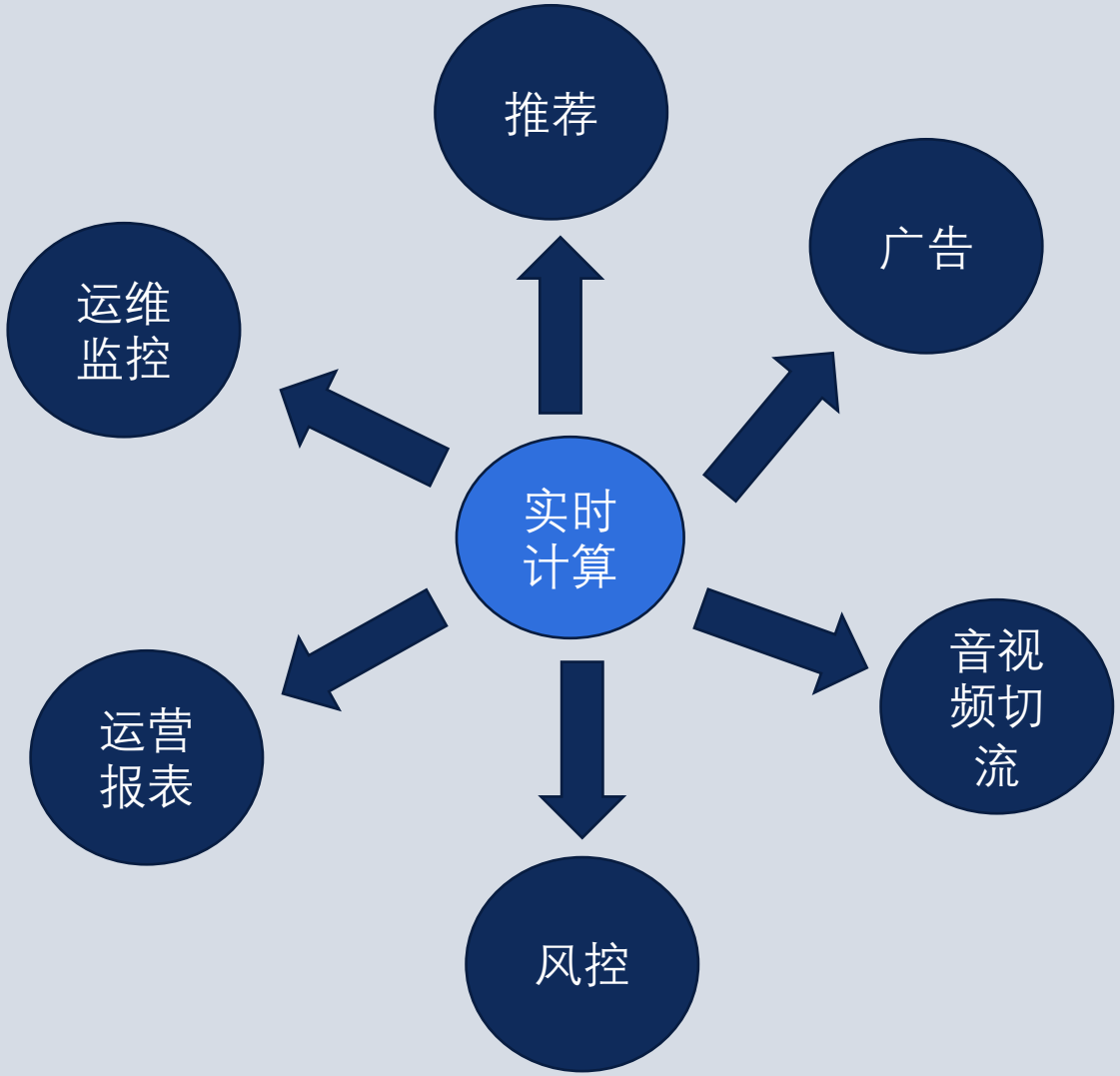
02 当前面临的挑战

03 解决的方式及效果

04 未来展望

虎牙实时计算平台的发展

实时应用场景



之前平台已解决的问题

之前平台已解决的问题

延迟

- 分钟级
- 亚秒级

易用

- 发布SQL
- 发布jar
- 发布scala 代码片段

生态

- source
- sink

运维

- 监控报警
- 日志排障

当前虎牙业务对平台新的要求

多中心

- 容灾：运维侧监控要求大数据计算网络容灾，如基础监控，业务日志监控
- 政策风险：海外数据安全政策不允许数据回国

边缘

- 成本：音视频上报数据所消耗的大量带宽成本
- 节点自控：边缘节点和主控中心节点断网之后依然可以服务

新场景要求带来的挑战

效率

环境差异

网络稳定性

成本

实时计算程序部署

- 一套程序，很多处部署
- 数据他人生成的，程序部署在哪儿
- 不同实例处理不同的数据，配置繁琐，容易出错
- 边缘节点多且变动性大

机房环境资源

- 不同产商交付的资源差异巨大
- 边缘节点多，大数据部署成本高

网络

- 边缘机房弱网环境
- 跨网传输带宽上限

化解挑战我们的解法思路

什么场景适合大数据多中心/边缘(region)计算？

数据可在region收敛

- 计算逻辑允许：只需要region上的数据参与运算
- 明细数据无需进入数据湖归档：可以在region进行重度降维
- 安全政策要求，不能出region

延迟要求高

- 时延主要消耗在网络传输环节的业务场景

可靠性

- 无需长久存储数据的无状态服务
- Region网络异常时需要自治

化解挑战我们的解法思路

需要解决的关键问题？

自适应的发布平台

- 根据数据自动选择region发布
- 根据发布的region自动转发所需其他region的数据

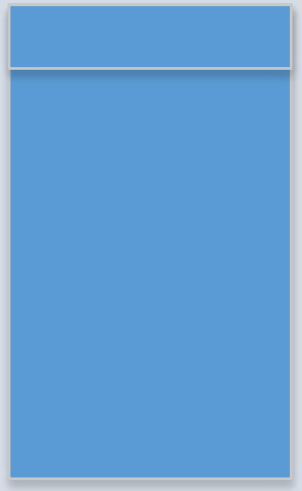
透明的跨region数据实时传输

- 将数据按需自动分发调度到所需的region
- 充分考虑数据跨网传输时的容灾、限流和降级

和业务服务资源混部

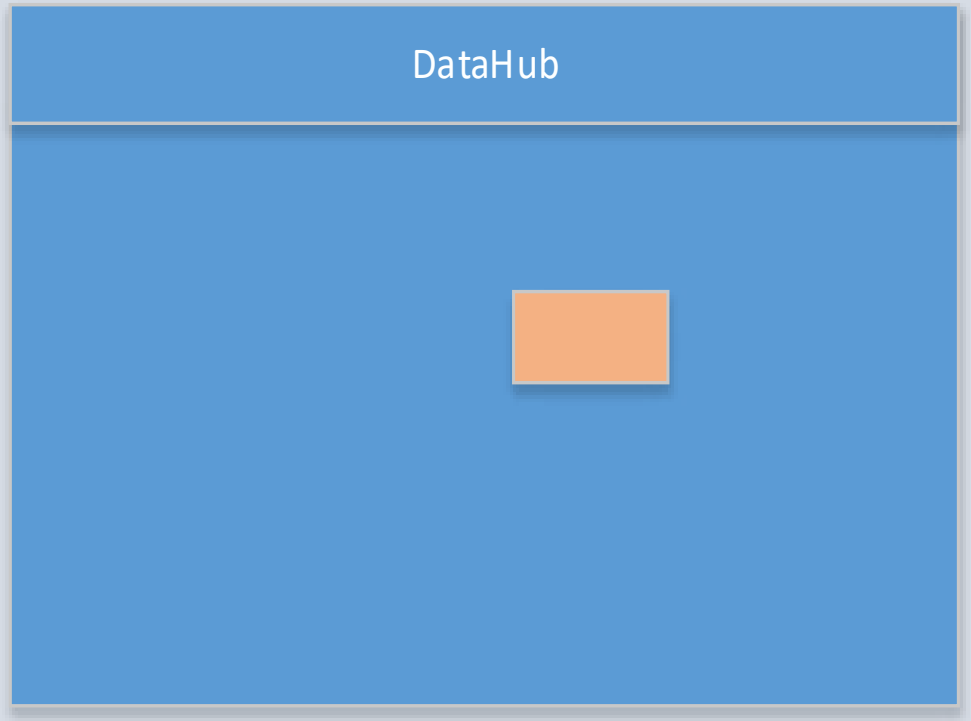
- 资源隔离、高密度的和业务混用资源
- 容器及服务，快速部署

面向多中心/边缘节点的实时架构



- 多中心/边缘部署，独立采集，存储和计算
- 虚拟集中实时数仓
- 集中管控

核心关键点---数据分发网络



核心关键点---自适应发布平台

目标：一次发布，多实例运行

数据需求

app,module,type：需要什么数据

部署模式

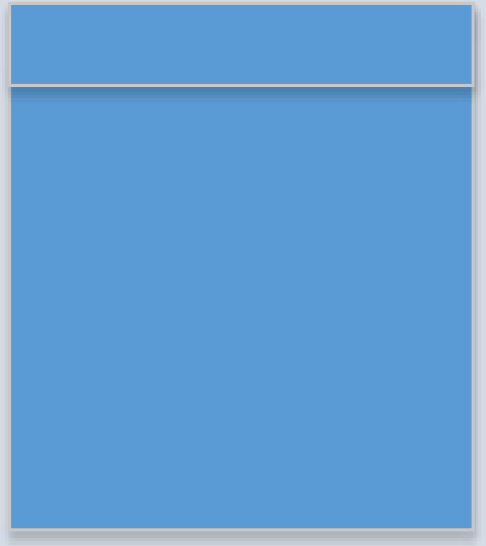
region=total:所有region数据汇总计算，在大数据中心机房计算，所有region上的数据分发调度到该机房
region=each:每个region的数据单独计算，在每个region机房使用自身数据计算
region=A,B:只需要特定region的数据，在大数据中心机房计算，特定region上的数据分发调度到该机房

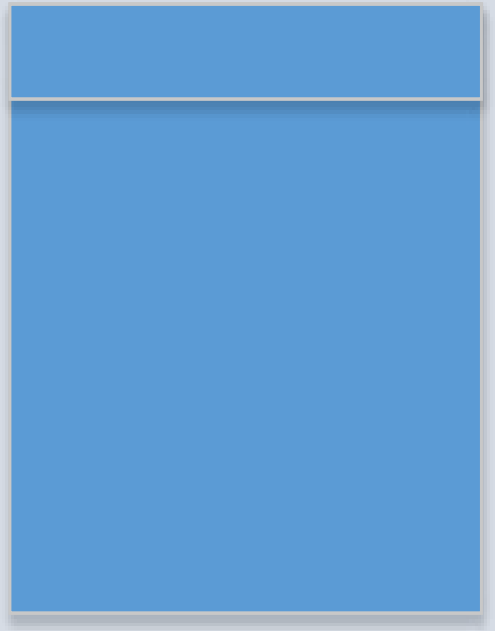
目前实际应用效果

Apache Flink

场景业务问题

因为政策原因，海外用户数据不能回国，所以需要在海外计算





目前实际应用效果

下一步计划

Apache Flink

- 解决DataHub中数据隔离性问题
- 弱网环境导致的数据延迟
- 边缘容器化部署

THANKS

Kafka X Flink Meetup

SHENZHEN

随时随地超大存储

1. **低成本：**冷温热降级、异构存储、容器混部资源、高效硬件、低冗余(副本冗余，实时离线存储一体)、存储格式(语义压缩)
2. **高吞吐/高效率：**并行传输、就近计算、热点调度，存储格式(索引/实时写入更新)，网络架构
3. **多中心：**(异地容灾，资源密度，网络成本) 异地副本容灾、跨网传输、本地Cache、就近计算
4. **多场景：**离线存储(HDFS)/实时存储(Kafka/pulsar)，大文件存储/小文件存储，高并发/高吞吐
5. **精准治理：**基于数据价值的性价比

按需极限高效算力

1. **高效计算引擎：**DAG计算(Spark)、流批一体毫秒级延时计算(Flink)
2. **高性能HTAP/OLAP存储引擎：**实时引擎(clickhouse/druid)，离线引擎(presto/impala)，HOLAP(kylin)，HTAP(tidb)==》尽量融合
3. **弹性算力：**容器化资源混部，按需弹性扩缩容，数据价值资源调度，公有云流量调度
4. **存储计算分离**

异构数据源便捷接入

1. **异构多样：**日志、app、db、nosql
2. **实时：**