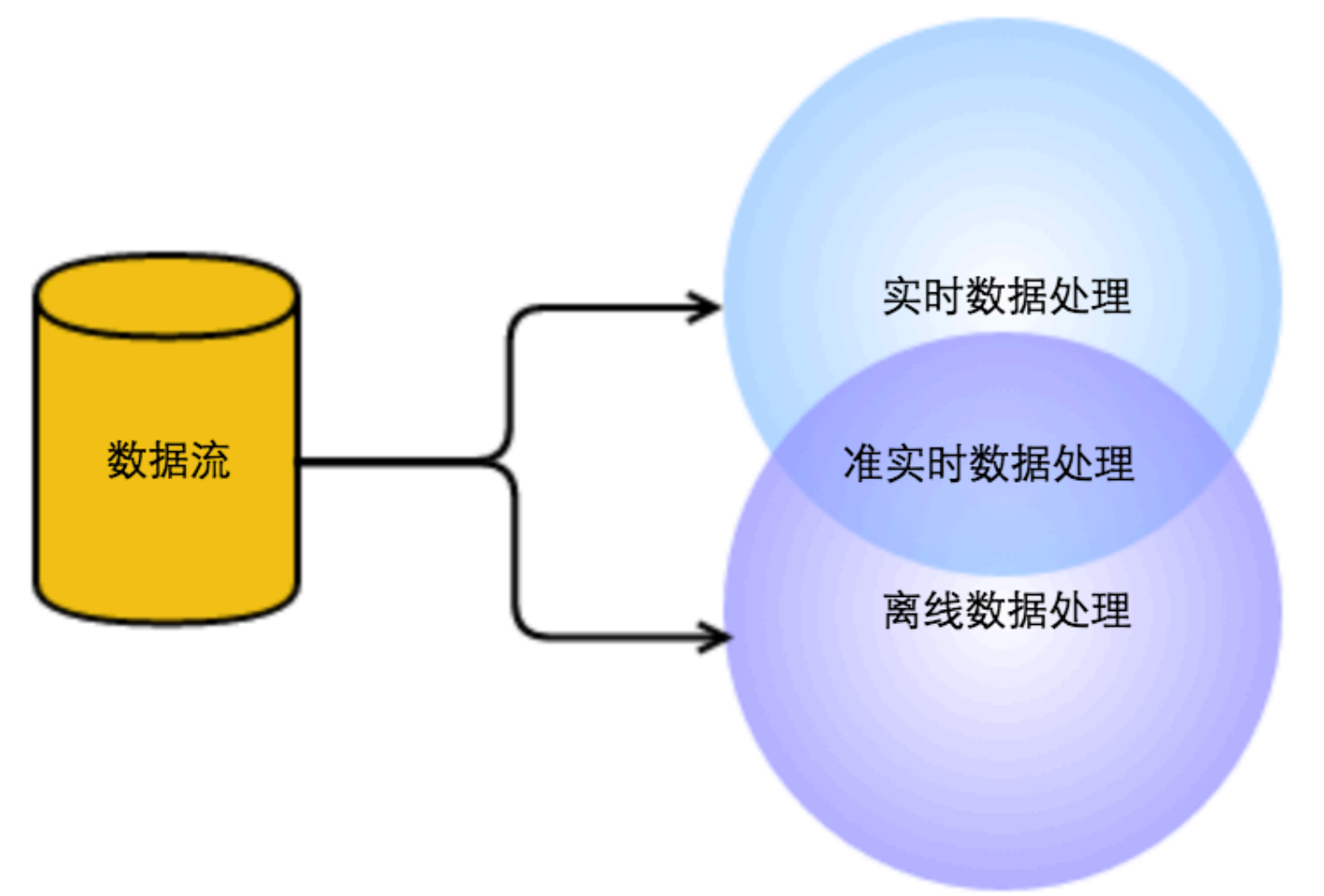


计算框架选型原则

划分依据

根据数据进入平台后，需要对数据进行处理并对外提供服务的时间周期，分别分为实时数据处理、准实时数据处理和离线数据处理。

数据处理类型划分



- 实时数据处理
在数据进入平台后，需要在秒级别对数据进行处理，并响应服务请求，这种场景定义为对实时数据的处理。根据行业经验进行调研，目前在1min内对数据完成处理并响应服务的过程，定义为实时数据处理。
- 准实时数据处理
数据进入平台后，需要在数分钟内处理完成并响应服务请求，定义为准实时数据处理。根据行业经验进行调研，定义在1min~30min内的数据处理定义为准实时数据处理。
- 离线数据处理
对于采集的数据，可能会长时间不会被消费使用，或者是处理的时间不会有太多限制。将超过30min对

外提供服务的数据处理定义为离线数据处理。

不同数据处理计算框架

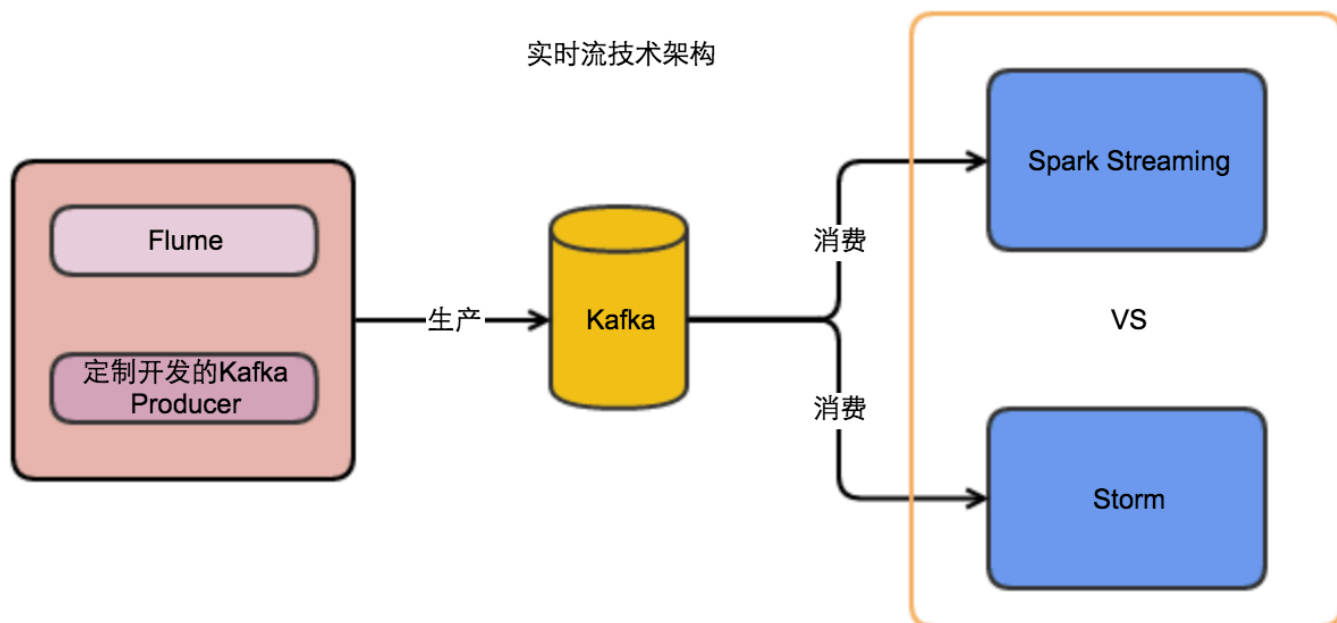
实时数据处理

在处理实时数据时，需要考虑的内容包括有：

1. 可扩展性
2. 容错性
3. 延时程度

业内在处理实时数据上，目前在可扩展性、延时、监控和容错角度都比较好的计算框架主要为：

- 通过Flume或者定制化的Kafka生产者将数据导入到Kafka中
- 并通过Spark Streaming或者是Storm，对Kafka中的数据进行消费



但何时采用Kafka + Spark Streaming，或者Kafka + Storm，下面将针对该两种框架进行说明。

Kafka + Spark Streaming框架

- 框架说明

在实时流数据处理中，业内比较流行的方案为本方案。

Kafka既可以作为消息的接收方，也可以作为提供方，而且Kafka是一个分布式的，高吞吐量，易于扩展地基于主题发布/订阅的消息系统，对实时 / 非实时的数据流都能够完全容纳，同时Kafka对数据做了容灾处理，能够很好的支撑业务需求。

Spark Streaming 模块是对于 Spark Core 的一个扩展，目的是为了以高吞吐量，并且容错的方式处理持续性的数据流。且Spark Streaming可以很好的对接Kafka、flume数据源。

- 使用原则

- 使用java、Scala、python调用Spark的API接口
- 支持秒级别的实时数据处理
- API接口较为丰富，可以实现更为复杂的需求
- 可以依赖Yarn做统一资源管理
- 使用建议
 - 如果平台需要做离线处理或批处理，或者需要做图算法处理、机器学习，建议使用该框架。主要原因是，Spark Streaming依附的Spark框架，对上述场景有着更佳的支持和处理。
 - 如果平台已经搭建了Spark，建议采用本方案，可以达到统一技术站的效果，平台只需提供一套计算框架，管理维护比较方便。

Kafka + Storm框架

- 框架说明

如同第一点的框架说明，Kafka既可以作为消息的接收方，也可以作为提供方，作为统一的数据源处理系统，比较完善。

Storm基于对实时流数据按照一条条的方式进行处理，有着更高的实时性。

- 使用原则

- 使用java、Clojure、Scala调用Storm的API接口
- 支持毫秒或更短到秒级别的实时处理
- 使用Mesos做资源管理
- 按单条数据进行处理

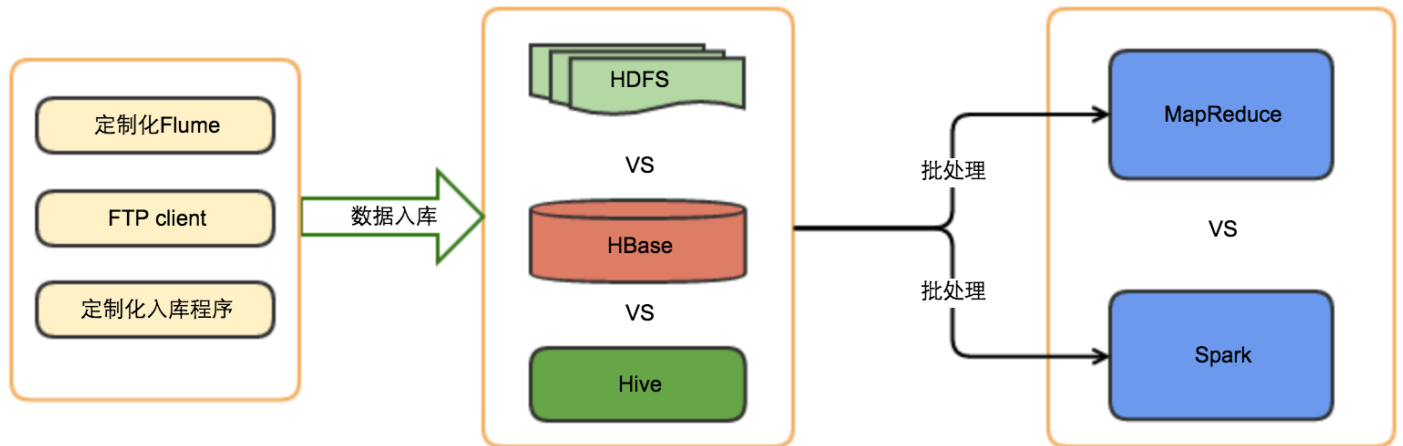
- 使用建议

- 如果对平台性能不是很关注，资源比较充足，且实时性要求很高，可以采用该方案。但由于针对离线数据处理存在较高的性能浪费，针对离线数据，可以考虑下述离线数据处理方案。
- 如果平台已经有了其他的离线处理工具（非spark），可以采用本方案，主要考虑到Storm更加轻便，且更好的实现业务分离，核心处理实时情况。

离线数据处理

考虑到离线数据处理的流程，目前从数据存储以及数据计算处理两个角度进行分析。

离线处理技术架构



- 数据入库时，可以根据需要存储在HDFS、HBase或者是Hive中
- 当需要对数据进行处理时，可以考虑采用MapReduce或者是Spark 天进行处理

但数据入库时，需要选择何种存储架构，以及后续对数据处理，采用何种计算框架，下面将针对性的进行原则性说明。

数据存储

针对离线的大量数据，根据数据的类型和需求不同，划分为三种主流存储框架。

- HDFS

分布式存储架构，有良好的吞吐能力，主要存储非结构化的数据。且自动做数据备份，有较好的容灾能力。

- 虽然支持对任何数据的存储，但建议存储数据原始数据和处理的中间结果，对于结构化、半结构化数据，建议保存在如下两种存储架构中

- HBase

基于列存储的分布式数据库。可以存储按Rowkey检索的数据，基于HDFS做底层存储，有着高可靠的容灾能力

- 需要保存历史变更纪录的场景。由于HBase自带有timestamp的概念，可以根据不同版本获取结果
- 需要支撑TB / PB数据较快的查询、导出和更新。由于HBase对大量的数据操作有着比较完善的处理机制，保证数据的完整性

- Hive

基于Hadoop的一个数据仓库工具，可以将结构化的数据文件映射为一张数据库表，并提供完整的SQL查询功能，可以将SQL语句转换为MapReduce任务进行运行

- 需要保存大量的结构化数据，且希望通过类似于通用的SQL语言进行数据的分析处理
- 需要做数据的大量分析工作和标签工作

数据计算

针对大量的数据进行分析 and 处理，目前比较通用的两种计算框架为MR和Spark。

- MR

比较传统的分布式计算框架，通过实现Mapper和Reducer来实现分而治之，将大量的任务进行拆解和合并。

- 需要采用业内比较稳定的计算框架时，首选该计算框架
- 需要简单平台环境。MR直接依托原生Hadoop平台，无需搭建更加复杂的计算框架

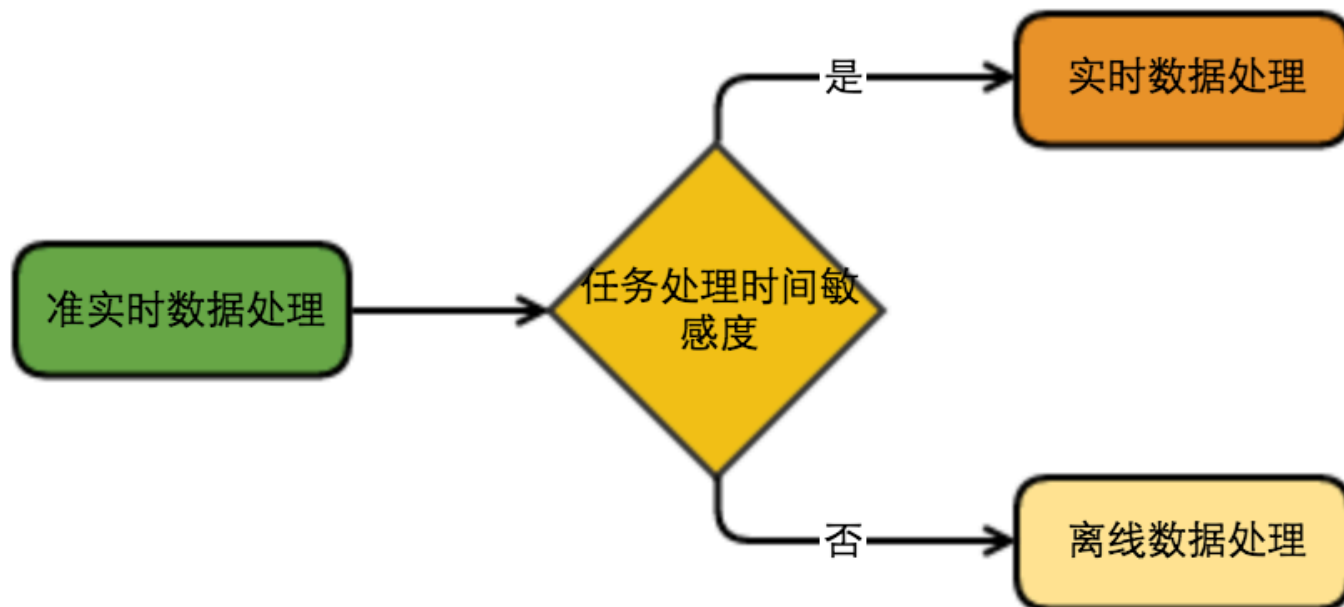
- Spark

Spark 是一种与 Hadoop 相似的开源集群计算环境，Spark使用了内存内运算技术，能在数据尚未写入硬盘时即在内存内分析运算，而相对于传统的大数据解决方案Hadoop的MapReduce会在运行完工作后将中介数据存放到磁盘中，Spark在内存内运行程序的运算速度能做到比Hadoop MapReduce的运算速度快上100倍

- 需要较短的开发时间。Spark采用Scala函数式语言，对比传统的MR可以用较短的代码行数实现比较复杂的任务，且维护更为简单容易
- 对计算时间比较重视。Spark基于内存计算，比传统的MR计算框架在速度上快100倍。

准实时数据处理

准实时是指对于进入平台的数据在实时数据与离线数据处理之间的一种场景进行处理。给出的推荐理由是考虑到未来业务会更加偏向于实时数据还是偏向于离线，因此给出如下建议。



- 选择实时技术框架

- 如果在业务发展过程中，需要更高的实时性，则可以采用对应的实时数据处理框架对数据进行处理
- 对任务处理时间比较敏感，且无法把控执行时间

- 选择离线技术框架
 - 如果在业务发展过程中，如果对数据处理敏感性降低，可以采用离线数据的处理框架对数据进行批处理

阅读资料

实时数据

[一个用于大数据实时处理的开源分布式系统druid（用于统一的数据库数据分析和监控，留着用）](#)

[阿里druid的github访问](#)

[实时计算,流数据处理系统简介与简单分析](#)

[经典大数据架构案例：酷狗音乐的大数据平台重构](#)

[使用 Kafka 和 Spark Streaming 构建实时数据处理系统](#)

[Kafka+Storm+HDFS整合实践](#)

[Storm与Spark streaming的比较](#)

[Apache storm vs. Spark Streaming](#)

[Yahoo compares Storm and Spark](#)

离线数据

[Hive, Impala, and Spark, Oh My: SQL-on-Hadoop in Cloudera 5.5](#)

[Spark SQL - Hive Tables](#)

[Hive on Spark](#)