

Kafka相关

日志处理已经成为消费者互联网公司数据管道的关键组成部分。我们引入了Kafka，这是一个分布式消息传递系统，我们开发它来收集和传递大量低延迟的日志数据。我们的系统融合了现有日志聚合器和消息传递系统的思想，适合离线和在线消息的使用。我们在Kafka中做了很多非常规但实用的设计选择，以使我们的系统高效和可扩展。我们的实验结果表明，与两种流行的信息系统相比，Kafka具有更好的性能。我们已经在生产中使用Kafka有一段时间了，它每天处理数百GB的新数据。任何大型互联网公司都会产生大量的“日志”数据。这些数据通常包括：(1)对应于登录、页面浏览量、点击、“喜欢”、分享、评论和搜索查询的用户活动事件；操作度量，如服务调用堆栈、调用延迟、错误，以及系统度量，如CPU、内存、网络或每台机器上的磁盘利用率。长期以来，日志数据一直是跟踪用户参与度、系统利用率和其他指标的分析工具。然而，最近互联网应用的趋势已经使活动数据成为直接用于站点特性的生产数据管道的一部分。这些用途包括(1)搜索相关性、(2)建议可能是由项流行或合作发生的活动流、(3)广告目标和报告、和(4)安全防范滥用行为的应用程序如垃圾邮件或未经授权的数据抓取、和(5)订阅功能聚合用户状态更新或行动的“朋友”或“关系”来读。这种日志数据的生产、实时使用为数据系统带来了新的挑战，因为它的体积比“真实”数据大了几个数量级。例如，搜索、推荐和广告经常需要计算粒度的点击率，它不仅会生成用户每次点击的日志记录，还会生成每个页面上数十个未点击的条目的日志记录。中国移动每天收集5-8TB的电话通话记录，Facebook每天收集各类用户活动事件近6TB。处理这类数据的许多早期系统都依赖于从生产服务器上物理地抓取日志文件进行分析。近年来，一些专门的分布式日志聚合器已经建立起来，包括Facebook的Scribe、雅虎的数据高速公路和Cloudera的Flume。这些系统主要用于收集日志数据并将其加载到数据仓库或Hadoop中以供离线使用。在LinkedIn(一个社交网站)，我们发现除了传统的离线分析外，我们还需要支持上面提到的大多数实时应用程序，并且延迟不超过几秒钟。我们已经为日志处理建立了一个名为Kafka的新型消息系统，它结合了传统日志聚合器和消息系统的优点。一方面，Kafka是分布式的，可扩展的，并提供高吞吐量。另一方面，Kafka提供了一个类似于消息传递系统的API，允许应用程序实时使用日志事件。Kafka已经开源，并在LinkedIn的生产中成功使用了6个多月。它极大地简化了我们的基础设施，因为我们可以利用单个软件在线和离线使用所有类型的日志数据。

flink相关

数据流处理(如复杂事件处理系统)和静态(批处理)数据处理(如MPP数据库和Hadoop)传统上被认为是两种截然不同的应用程序。它们使用不同的编程模型和api进行编程，并被不同的系统(例如专用的流系统，如Apache Storm、IBM Infosphere Streams、Microsoft StreamInsight，或Streambase相对于关系数据库或Hadoop的执行引擎，包括Apache Spark和Apache Drill)所切割。传统上，批处理数据分析占据了用例、数据规模和市场的大部分份额，而流数据分析主要服务于专门的应用程序。然而，越来越明显的是，今天大量的大规模数据处理用例处理的数据实际上是随着时间的推移不断产生的。这些连续的数据流例如来自web日志、应用程序日志、传感器，或者来自数据库中应用程序状态的更改(事务日志记录)。今天的设置忽略了数据生产的连续性和及时性，而不是将数据流视为流。相反，数据记录(通常是人为地)被批处理成静态数据集(例如，每小时、每天或每月的数据块)，然后以时间不可知的方式进行处理。数据收集工具、工作流管理器和调度器编排了批处理的创建和处理，这实际上是一个连续的数据处理管道。诸如“lambda架构”这样的架构模式结合了批处理和流处理系统来实现多路径的计算：用于及时近似结果的流快速路径，以及用于后期精确结果的批脱机路径。所有这些方法都有很高的延迟(分批产生的)，高度复杂性(连接和编排多个系统，并实现两次业务逻辑)，以及任意的不准确性，因为时间维度没有由应用程序代码显式地处理。Apache Flink遵循一种范式，将数据流处理作为编程模型和执行引擎中实时分析、连续流和批处理的统一模型。与允许数据流的准任意重放的持久消息队列(如Apache Kafka或Amazon Kinesis)相结合，流处理程序不区分实时处理最新事件、周期性地在窗口中持续聚合数据或处理兆字节的历史数据。相反，这些不同类型的计算只是从持久流中的不同点开始处理，并在计算期间维护不同形式的状态。Flink程序通过一种高度灵活的开窗机制，既可以计算早期和近似，也可以计算延迟和准确，结果在同一个操作中，避免了对两个用例进行不同系统组合的需要。Flink支持不同的时间概念(事件时间、摄入时间、处理时间)，以便为程序员提供定义事件如何关联的高度灵活性。与此同时，Flink承认现在和将来都需要专门的批处理(处理静态数据集)。静态数据上的复杂查询仍然很适合批处理抽

象。此外，批处理对于流用例的传统实现和分析应用程序仍然是需要的，因为目前还没有有效的算法来对流数据进行此类处理。批处理程序是流程序的特殊情况，其中流是有限的，记录的顺序和时间并不重要(所有记录都隐式地属于一个包含所有内容的窗口 - dow)。然而，为了支持具有竞争性的易用性和性能的批处理用例，Flink有一个专门的API来处理静态数据集，为操作符(如join或分组)的批处理版本使用专门的数据结构和算法，并使用专用的调度策略。结果是，Flink在流运行时之上展示了一个成熟而高效的批处理处理器，包括用于图形分析和机器学习的库。Flink起源于Stratosphere项目，是Apache软件基金会的顶级项目，由一个大型且活跃的社区(截至撰写本文时，由超过180名开源贡献者组成)开发和支持，并在几家公司的产品中使用。

ETL相关

数据仓库和相关技术允许业务用户实时的做出以数据为依据的决策。它有效地处理和存储机构的用于数据分析的关键历史数据集。来自交易源系统的最新数据通过提取，转换和加载（ETL）流程存储到数据仓库中。传统上，ETL被安排为夜间批处理过程，以在非工作时间刷新数据。但是，随着科技的进步，24X7全天候业务运营模式，市场竞争以及引入现代数据源已迫使组织采取以下行动：更频繁地刷新其数据仓库，以提供近乎-实时环境。传统的ETL流程难以应对具有近实时环境的要求，因为它没有为此目的进行设计和优化。结果，更先进的ETL版本才能满足近乎实时环境，例如低延迟，高可用性和高扩展性。现阶段可以通过大数据技术，如 kafka, flink, clickhouse 等进行组合，来实现近实时ETL。根据Bill Inmon的定义，数据仓库是“一个主题面向，集成，时变和非易失性集合支持管理层决策的数据”。换句话说，它是组织数据的集中存储库，专为查询和数据分析而设计，可帮助企业领导者做出以数据为依据的决策。它集成自各种源系统的数据，例如交易系统，客户关系管理（CRM）数据库，企业资源计划（ERP）系统等等。一组业务转换应用于源数据，使它更符合报告要求将其存储在数据仓库中。使用提取转换加载刷新数据仓库（ETL）流程。顾名思义，ETL具有三个主要方面阶段，每个阶段都照顾整体的特定部分数据刷新过程。提取阶段从异构数据系统读取数据并将其存储到中间存储中。转换阶段读取提取的数据，开展业务规则和基本数据操作，例如清理，数据类型转换等。加载阶段最终将转换后的数据加载到目标数据仓库。如今，为了节省成本并在竞争中脱颖而出,全球各地的组织都在采取新举措使他们的业务流程更加完善。数据仓库在这些计划中起着关键作用因为它为分析和决策提供了必要的过程。但是，这些计划需要数据仓库更新频率更高，以获取最新数据进行分析。此外，还引入了现代数据源，例如传感器，闭路电视，Web反馈等。比传统的夜间批处理过程更频繁地读取更多的数据，因为数据量太大，无法在单个ETL运行中处理，有时源数据仅保留一小段时间，这意味着直到下一个批处理过程，它才能在源上提供，这使得ETL流程容易出现数据差异问题。同样，大多数大型组织的办公室遍布在世界上的多个国家/地区，这意味着用户正在积极地使用24X7的数据仓库。由于这种范例数据处理的和分析要求的变化，组织将数据仓库更多地视为“活动”或“实时”数据仓库，而不是作为历史数据的系统分析，并期望更频繁地刷新数据，例如每小时或每隔几分钟。自然，传统的ETL还需要更改流程以处理较少的数据量快速接连以使其更接近实时数据仓库环境