Scalable OIJ on Modern Multicore Processors in OpenMLDB

动机

- 该研究基于OpenMLDB平台，旨在提高现代多核服务器上进行OIJ的延迟和吞吐量。
- Ooline Interval Join（OIJ）是一种常见的基于时间数据库的区间连接方法，用于在数据流中连接具有重叠时间区间的元组。然而，现有的OIJ解决方案（key-OIJ）在处理大规模数据时存在性能瓶颈，因此需要进行优化。
- 本研究通过实验分析现有解决方案的设计空间和关键问题，并提出了一种新的解决方案——Scale-OIJ，以提高OIJ的性能。本研究的重要性在于，它提供了一种有效的方法来优化OIJ，从而提高时间数据库的处理效率，为机器学习等领域的实时数据处理提供更好的支持。
- Dataset：Both real-world and synthetic
- Indicator: Throughput(mean value); Lateness(CDF); Unbalancedness(standard deviation of workloads); LCC miss(last-level cach miss)

简单对比

- 在文本中，作者提出了一种名为Scale-OIJ的新方法，用于实现可扩展的在线区间连接。与现有的OIJ解决方案Key-OIJ相比，Scale-OIJ具有以下区别：

- Scale-OIJ：
- - 使用SWMR(Single-Writer-Multiple-Reade) Time-Travel数据结构，可以在不锁定数据的情况下进行并发读写操作。
- - 使用动态平衡调度，可以在多个线程之间动态分配工作负载，以实现更好的负载均衡。
- - 使用增量窗口聚合，可以避免重复计算重叠窗口，从而提高计算效率。

- Key-OIJ：
- - 使用基于键的分区并行化策略，可以将输入元组并发地与相同键的缓冲区连接。
- - 为了处理潜在的乱序流到达，元组只能在一段时间后才能被删除。

传统方法的缺点---key-OIJ

- 1. 处理无序数据的成本高昂：Key-OIJ算法在处理无序数据时表现不佳。特别是在大延迟的情况下，需要进行昂贵的无序数据操作。在Flink中，每个连接操作都需要进行完整的数据扫描，导致性能下降。
- 2. 负载不平衡：Key-OIJ算法采用基于键的分区策略，这可能导致负载不平衡的问题。特别是在键数较少的情况下，工作负载可能无法均匀地分布在多个处理器上，从而影响性能。
- 3. 冗余计算：Key-OIJ算法无法利用在重叠窗口中已处理的数据，导致存在显着的冗余计算。这意味着在处理大窗口时，会进行大量不必要的计算操作，降低了性能。
  （当窗口大小较大时，匹配时间变得更加占主导地位，因为需要更多时间进行窗口聚合。相反，当窗口大小适中但延迟较大时，查找时间远远超过匹配时间。这是由于无序元组到达更多，Key-OIJ需要为每个间隔连接访问更多的元组。）

新方法----scale-OIJ

- 整体上，该设计遵循基于键的分区模型（Key-OIJ），但允许根据工作负载分布和相同分区的共享处理来动态重新分区数据。通过精心设计的数据结构和并发模型，提出了一种轻量级的动态调度算法，以实现高度平衡的适应性，而无需数据复制或迁移。
- Time-Travel Data Structure
  为了实现高效的窗口数据检索，文章设计了一种基于double-layered skip-list的数据结构。该数据结构的第一层是用于存储<key，second-layer skip-list>，而第二层是用于存储<timestamp，Tuple>。搜索时：首先在第一层 skip-list中搜索以key为关键字的第二层 skip-list，然后在第二层 skip-list中搜索以时间戳为关键字的元组（tuple）
  时间复杂度：O(logNkey) + O(logNts)（在实验中，随着lateness的增加，吞吐量上新方法优势巨大）
  SWMR Concurrency Property：具有相同键的元组可以由多个连接器（joiners）共同处理。

新方法----scale-OIJ

- Dynamic Schedule
  shared Processin：通过共享处理框架和虚拟团队的设计，连接器可以共享具有相同键的元组，并且可以在不影响正确性的情况下随机分配给virtual team的任何成员进行处理，而不会影响正确性。

  Dynamic Schedule：
  共享处理框架允许在不迁移数据的情况下动态重新分区数据。通过在运行时收集数据分布统计信息，可以推导出所有连接器的工作负载分布，并基于此定期重新调度分区分配（即键分区调度）。重新调度的目标是减少连接器之间工作负载调度的不平衡性。（ALgorithm3）

新方法----scale-OIJ

- Incremental Online Interval Join
  重叠窗口问题：在进行区间连接时，特别是当窗口较大时，存在邻近窗口可能重叠的高概率，这会导致重复的数据访问和计算。
  为了解决重叠窗口问题，文章提出了Incremental Online Interval Join。该方法通过使用增量计算和数据结构维护，避免了重复的数据访问和计算。