Tinkerpop OLAP 以及图计算

金世钰

2019.08.11

1 Tinkerpop 框架以及 gdm

gdm 是基于 Tinkerpop 图计算框架开发的一款数据库产品，关于 Tinkerpop，其官方网站对它自己的定位是：开源的图计算框架。其作用是通过以提供 API 以及相关工具的方式，简化开发者创建图应用（个人觉得包括图数据库及其处理器和其他相关应用）的难度。Tinkerpop 是一个位于不同的图数据库和图处理器上的抽象层，可以视作实现一个图数据库的公约，只要将 Tinkerpop 这个抽象层具体化，就可以开发出自定义的图应用。

Tinkerpop 框架从大的方向来说，包含 OLTP 和 OLAP 两块，分别代表着在线事务处理和在线分析处理，如果要基于 Tinkerpop 实现一个图数据库，那么 OLTP 这一块是必不可少的，而 OLAP 是可选的。对一个数据库来说，CRUD 必须实现，而这一块正是属于 OLTP 的范畴，所以 OLTP 必须实现。OLAP 则提供了除 OLTP 之外的其他功能，比如基于某种算法对整个数据库里面的数据进行分析得出某些想要结论，这对一个图数据库来说不是必须提供的功能，属于可选部分。Tinkerpop 自身的架构（图 1）对外提供了 OLTP 和 OLAP 的接口（Provider API），交由图数据库的开发者去具体实现，当然 Tinkerpop 官方也提供了一个对这些 API 的实现-TinkerGraph。

目前基于 Tinkerpop 实现的图产品（图数据库）包括以下列表中几个比较典型的图数据库，而且在 DB-engines 这个全球图数据库产品排名上，下方列表中的很多产品也是名列前茅。

- Microsoft Azure Cosmos DB
- OrientDB

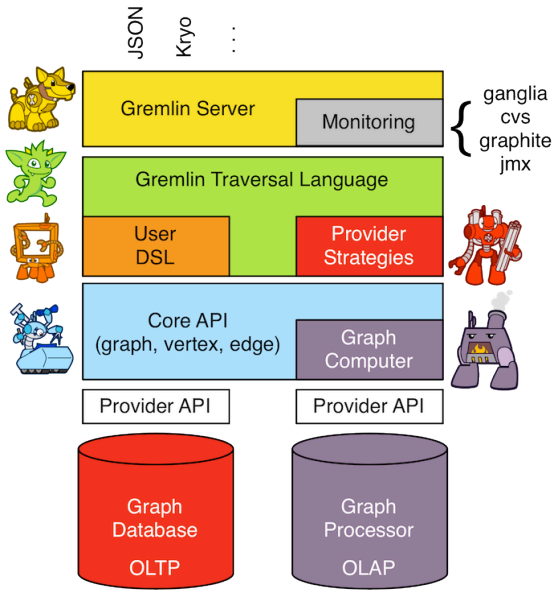

图 1: Tinkerpop 框架组成

- ArangoDB
- JanusGraph
- Amazon Neptune
- Baidu HugeGraph
- Alibaba GDB

2 Tinkerpop OLAP

Tinkerpop 的 OLAP 部分主要用于全图分析（即相关计算必须涉及到图里面所有顶点），使用了批量同步并行计算模型（Bulk Synchronous Parallel，BSP），同时规定了一系列和 BSP 模型中相对应的编程接口，这一部分内容将在下面阐述。

2.1 BSP 与 Tinkerpop OLAP

批量同步并行计算模型（Bulk Synchronous Parallel，BSP），由哈佛大学的 Leslie Valiant 在 20 世纪 80 年代提出，是一个用于设计并行算法的桥接模型。一个 BSP 模型包括以下三个要点

1. 拥有计算能力和内存事务的组件（比如处理器）
2. 一个能在上述组件中按照路由传递信息的网络
3. 一个能同步上述组件的硬件设施

说的直白点，一个 BSP 模型（图 2）就是一系列的处理器，每个处理器都能做计算且配置着高速的本地内存，所有处理器之间可以通过通信网络之间进行信息交流。一个基于 BSP 模型的算法，最依赖的是上述的第三个要点，一个 BSP 算法的执行由很多超步组成，每个超步都包含三部分

1. 并行计算，即每个处理器之间的计算是并行的
2. 通信，每个处理器会接受输入，并执行计算产生输出，输出会成为后续计算的输入，这就涉及到通信
3. 栅栏同步，后续计算的开始必须在满足某个条件的情况下在能继续

Tinkerpop 的 OLAP 部分就采用了 BSP 模型，所以相应图计算算法的设计，必须符合 BSP 的模型要求，接下来说明一下 Tinkerpop 的 OLAP 部分对 BSP 模型的框架实现。请参看图 3 中的 Tinkerpop 对 BSP 实现的示意。

可以发现，BSP 中的处理器对应着 Tinkerpop 图里面的一个个顶点，BSP 传递信息的网络对应着 Tinkerpop 里面对应的消息传递组件，BSP 中同步组件的设施就对应这 Tinkerpop 中的下一轮计算进行前的消息同步，顶点执行的算法就是 VertexProgram，在整个计算过程中，每个顶点接收上一轮发送给自己的消息，作为 VertexProgram 算法执行时的输入，然后执行计算流程，将产生的结果也作为消息传递给下一轮作为某个顶点的输入，每个顶点在每执行完一次 VertexProgram 之后需要判断是否满足算法的结束条件（比如 PageRank 算法中规定的迭代次数是否已经达到或者 PageRank 的值已经趋于收敛），当所有顶点都满足算法的结束条件之后，整个算法流程执行完毕，这就是 BSP 也是 Tinkerpop OLAP 的计算核心流程。这里面

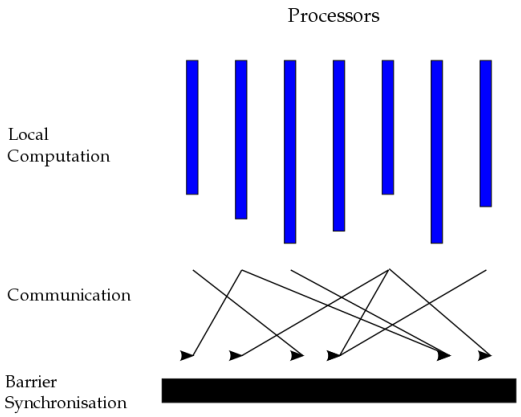

图 2: BSP 模型示意图

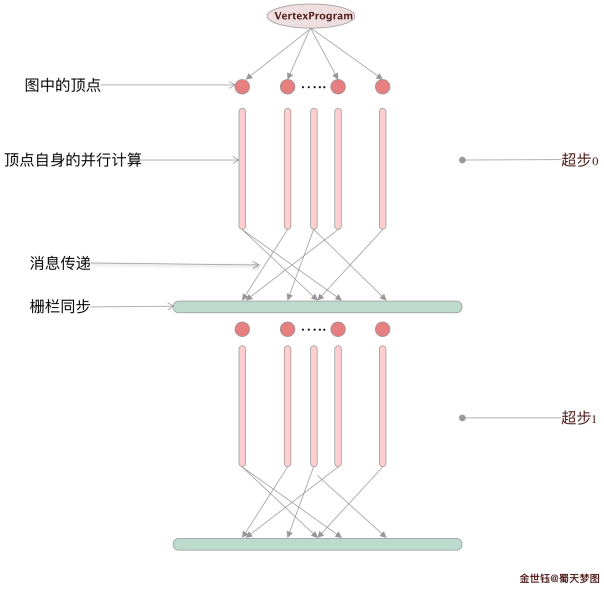

图 3: Tinkerpop 对 BSP 模型的实现

需要注意的有两点，第一是所有要执行的算法必须按照 BSP 模型编写，这可能和我们平常使用的算法有所出入；第二是要确定好消息的类型，确保消息能够正确的表达算法数据的传递，比如在 PageRank 中传递的消息就是数值类型，而在最短路中传递的消息就是包含路径、边和对应数值型数据的三元组信息。

整个 BSP 模型以消息传递为核心，这是 BSP 模型的亮点，同时也限制了该模型，当计算过程中产生大量信息导致系统无法处理的时候，这个基于消息传递的 BSP 算法就会彻底崩溃。

2.2 Tinkerpop 已有的 OLAP 的实现

TinkerGraph 是一个实现了 Tinkerpop 所有规范的内存图数据库，我们在开发 gdm 的 OLAP 时，就参考了 TinkerGraph 的流程。TinkerGraph 作为一个内存图数据库，其 OLAP 的实现也是主要面对内存的，消息传递组件就是内存中的基于顶点和对应消息队列的一个共享对象。

TinkerGraph 的 OLAP 的实现主要如下

- 实现 GraphComputer 接口做图计算任务的提交和执行和结果的处理
- 顶点均分为 N 份，交由 N 个线程轮询自己负责计算的顶点并执行计算
- 使用共享的内存对象做消息接收和传递

在这个实现里面，使用共享的内存对象做消息接收和传递这一条将会在消息数据量巨大的情况下成为 OLAP 部分的瓶颈甚至是直接导致运算时出现 OOME（Out of Memory Error）从而导致相关服务宕机。

2.3 Tinkerpop 分布式 OLAP 架构设想

单机 OLAP 在消息量大情况下的缺陷使得我们不得不想办法解决这个问题，在我 7 月份经历过对基于 Tinkerpop 属性图模型实现的 Titan、HugeGraph 和 JanusGraph 的调研以及对 BSP 模型的进一步了解之后，基于 BSP 模型和 Tinkerpop OLAP 流程，我构想出了图 4 所示的一个分布式 OLAP 架构

其实这个模型就是基于单机 OLAP 改进而来的，它和单机 OLAP 存在下面的对应关系

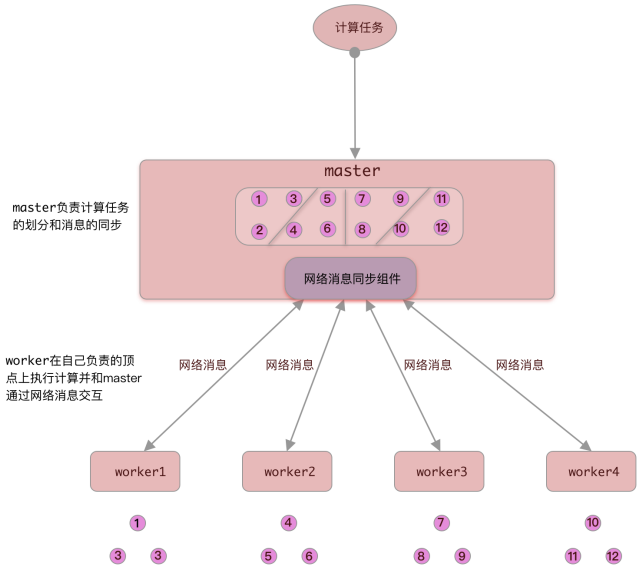

图 4: Tinkerpop 分布式 OLAP 设想

1. 单机的多线程对应分布式的多个 worker 计算节点
2. 单机的共享消息存放组件对应分布式的 master 节点的消息服务组件
3. 单机中的消息对象对应着分布式中在网络中传递的消息

整体来看，这个分布式的思路和单节点的思路是一致的，也比较具有可实施性，但是仔细往下思考可以发现这里面有很多问题需要解决，现在想到的问题主要有消息在超步执行之间的同步和保证消息正常发送和接收，以及计算节点和主节点的意外下线等，这些都是我们在未来需要解决的问题。