

ACyclops: 非对等自适应异步图计算

管仁阳

摘要—本项目分析了同步与异步图计算模型的区别，将 Cyclops 改造为异步模型，并提出了针对幂律图分布的优化机制。

关键词：异步图计算，“边中心”划分，共享内存，消息传递

I. 背景

分布式图计算模型可以分为**同步**与**异步**两类，同步与异步的区别在于每个节点是否知晓其他节点的计算状态。同步的分布式运算与单机结果一致，而异步运算则不然，但是后者能够帮助图机器学习算法快速收敛且提高效果。考虑到经典图算法的数量有限而机器学习算法不断推陈出新，图计算框架有必要针对机器学习算法进行进一步优化。无论同步或是异步，最重要的性能瓶颈都是分布式主机之间的信息交互。

Cyclops

Cyclops [1] 是一个同步的分布式图计算框架。Cyclops 使用“点中心”的划分方法进行子任务划分，即每个主机上的节点数量相当。在主机间交互方面，Cyclops 使用共享内存的方法，即对于那些跨越主机的边在两台主机上都保存边的两个节点，一个实际执行计算，另一个是为了减少跨主机数据传输所进行的只读备份。

使用这种共享内存机制，重要问题是一个节点的多个备份的数据同步。Cyclops 是同步模型，每一个节点都需要等待其他所有节点都计算完成以后才可进入下一轮计算，而本轮计算只需要读取相邻节点在上一轮计算的结果。因此 Cyclops 摒弃了跨设备读取的分布式锁，而让实际执行计算的节点发送全局消息去更新其他备份。

Cyclops 避免了分布式锁的竞态等待拖慢整体速度的问题，然而却有两个缺点：1. 计算较快节点等待其他节点的时延较大 2. 真实图具有倾斜的幂律分布，他们成

为了 Cyclops 执行机器学习算法的性能瓶颈，ACyclops 正是对 Cyclops 的上述缺点进行优化。

II. 方法

ACyclops 的优化总结如下

- 异步计算
- “边中心”的划分方法
- 共享内存与消息传递的自适应选择

异步计算：ACyclops 使用异步分布式模型，即一个节点在完成自己的计算以后直接更新它的所有副本值而不要管其他节点处在何种状态。为此副本的写锁是有必要的，但是此写锁是局部的，其代价与同步模型中的分布式读写锁相比可以忽略。

“边中心”的划分方法：划分子任务时维持每个主机上边的数量相当。这虽然能够有效地解决图幂律分布造成的负载失衡，但会使得高阶 (High-degree) 节点产生大量副本，造成过大的内存开销和沟通成本。

共享内存与消息传递的自适应选择：为解决上述问题，让高阶节点 (99% 上分位数) 使用消息传递机制、低阶节点使用共享内存机制，便可以在时间开销与内存占用之间取得均衡。当然，需要分配给高阶节点所在主机更大带宽。

III. 结论

本文将 Cyclops 扩展为更适合机器学习算法的异步图计算框架，针对幂律图分布使用“边中心”划分方式与共享内存/消息传递自适应选择方法，并对其进行相应的理论分析。

参考文献

- [1] Chen R, Ding X, Wang P, et al. Computation and communication efficient graph processing with distributed immutable view[C]//Proceedings of the 23rd international symposium on High-performance parallel and distributed computing. 2014: 215-226.