面向大数据图计算的在线图划分算法研究

信息科学技术学院 2013 级计算机系　潘成

摘要：

随着大数据时代的到来，基于云环境的大图迭代计算已经成为新的研究热点，其中提高图划分算法的执行效率和降低划分后子图之间的通信边规模是改善计算性能的关键。现在已经有的工作主要分成了离线划分和在线划分两个大类，离线算法有些可以得到比较好的通信边规模，而算法的执行效率达不到实时划分的要求；在线算法在效率方面比较不错，但是如何控制通信边的规模又成了一个难题。如何平衡执行效率和划分效果是当前图划分算法一个研究的热点。

同时，由于传统的如 Hadoop 等通用的云计算平台并不适合迭代式的处理图数据，研究人员基于 BSP 模型提出了新的处理方案，比如 Pregel，Hama，Giraph 等图计算框架。然而，图处理算法需要按照图的拓扑结构频繁交换中间计算结果而导致了巨大的通信开销，这严重影响了基于 BSP 模型的系统的处理性能。本次本科生科研的主要任务就是探求如何在保证效率的情况下，尽可能减少通讯边的开销，并且试图发现不同的图计算需求对划分方案有何影响。

一、项目概述

随着互联网技术的普及和新兴社交网络等应用的快速发展，大规模图数据的高效处理已经成为学术界和工业界的研究热点。真实图数据集尤其是基于互联网的图数据，其规模通常可以达到数十亿定点和上百亿条边，如 WWW 的 web 页面关联图有 5000 亿定点和一万亿条边；全球最大的社交网站 Facebook 有近 80 亿个顶点和 1040 亿条边。显然，传统的单机串行和小规模集群都无法处理如此庞大的数据。因此，如何高效处理大规模图数据成为亟待解决的问题。

一个很直觉的想法就是将一个大规模的图数据划分成尽量均匀的若干个小的子图，将每个子图的计算放在单个的节点上，而整个大图的计算就是由这若干计算节点组成的分布式的环境。当然，划分出的子图之间也会有相联系的边集，不可避免的，这些边必须在图迭代计算的过程中担任传递消息的工作。一个好的划分，会使得这些通信边在全图中占的比例尽可能小，同时保证划分是尽可能均匀的。

二、准备工作

目前图划分算法从大的方向来看，主要可以离线算法和在线算法。离线算法可以针对读入的图数据反复迭代调优，不用特别在意处理的时间，只需要最终的划分效果尽可能好。而在线算法比较在意的是算法处理的时间，任何超过线性的复杂度的算法都是不可接受的。

离线划分算法的一个经典处理方案是通过对原图进行多级”coarsening”操作不断压缩图的规模，然后对压缩过的图采用 Kernighan-Lin 或者 FM 等复杂算法

进行初始划分，之后再执行”uncoarsening”操作，得到最终的划分结果。例如 Chaco, Metis 和 Scotch 等都是面向集中式的多级划分算法库，而因划分后的子图具有较少的切分边而被业界广为青睐。此外，ParMetis 和 PT-Scotch 等并行化的版本可以进一步提高扩展性。然而这类算法在”coarsening”阶段的最大匹配操作引入了极为昂贵的开销，因此，整体算法的执行效率大大受阻。

因此，MLP 采用分布式的基于标签传播的的连通域查找算法替换”coarsening”阶段的最大匹配，以此来降低执行开销。甚至，有一些算法直接采用类似标签传播的方式来执行图划分——每个顶点采取其邻居顶点中具有最多的标签作为自己的标签，而具有相同标签的顶点属于同一个划分后的子图。这个方法采用线性规划来约束每个子图的大小，实现负载均衡。

Rahimian 等人设计了 JA-BE-JA 算法，将顶点的标签成为 color，每个顶点和自己的邻居以及部分随机顶点交换标签，以减少子图之间的交互边。JA-BE-JA 算法采用模拟退火算法来避免算法陷入到局部最优。进一步，为了满足实际应用中的多种需求，Slota 等人提出一种基于标签传播的多目标划分方法 PuLP，这个算法针对不同的约束条件进行调整。然而，基于标签传播的方法都需要每个顶点迭代更新自己所属的类别，所以仍需要扫描多次图数据，制约了执行的效率。

刚刚介绍的多种算法都是针对离线的处理过程的，而在线的划分算法会假设图数据以顶点流或者边流的方式到达，在流处理过程中根据已到达数据的分布信息，通过启发式的规则决定当前图数据的划分位置。业界常用的 Pregel、Giraph 等图计算平台会在执行迭代计算前加载图数据，因此实现图划分的一个很恰当的时机就是在图数据的加载阶段。与 Metis 为代表的离线划分算法相比，在线的划分通过在一定程度上牺牲切分边规模的优化效果来避免”coarsening”阶段的多次数据扫描，极大提高了执行效率。比如集中式的 LDG 和 FENNEL 算法可以准确的实时维护数据的分布信息，保证了启发式计算结果的精度，切分边的规模比较小。同时，Nishimura 等人提出的”restreaming”方法，即相同的图数据被反复加载处理时（例如，首次加载用于计算单源最短路，第二次加载用于计算 PageRank），可以利用上一次流式划分的结果，来改善本次划分的效果。

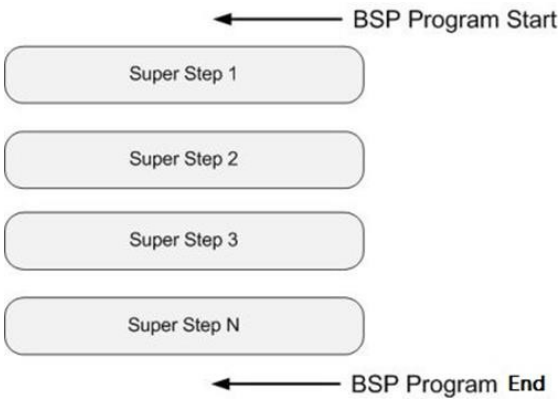
Hash 划分可以看做是一种最简单的分布式流式划分算法，其启发式的规则是：通过顶点标签或者边标签的 HashCode 值决定数据的存放位置。虽然 Hash 划分通常由于难以保留图的局部性而引入大量的通信开销，但是它可以在分布式环境下快速实现图的分割，因此被 Pregel 等系统作为默认的划分方式。

上述的图划分工作都是围绕静态图展开的，而实际应用中，对于 Pregel 等迭代的处理系统，图是动态变化的，因为在在迭代的过程中，值达到稳定的图顶点通常不再参与后续计算，例如单源最短路径计算，所以实际参与计算的图数据是动态变化的，这使得初始的图划分可能产生负载偏斜，或者划分效果不再是最优的。这个问题在后面是试验中是真实存在的，目前已经有 GPS，Mizan 和 Xpregel 等系统支持在迭代过程中重新划分数据。

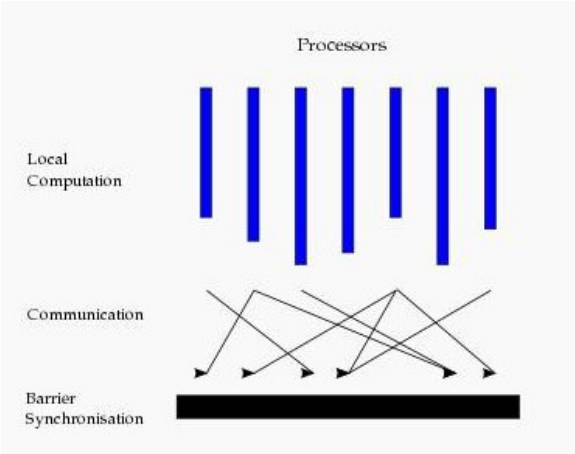
三、模型构建

MapReduce 是 Google 推出的一种简单有效的工作流式处理模式，适用于通用的大数据计算，并且现在 MapReduce 的开源实现 Hadoop 已经能够提供商用计算，支持对大数据的批量计算。但是对于图算法、图挖掘以及机器学习等需要进行多次迭代的计算，MapReduce 模型并不适用。

所以，基于 BSP（Bulk Synchronous Parallel，整体同步并行计算模型）模型的分布式迭代处理系统应运而生。BSP 计算模型不仅是一种体系结构模型，也是设计并行程序的一种方法。BSP 程序设计准则是整体同步(bulk synchrony)，其独特之处在于超步(superstep)概念的引入。一个 BSP 程序同时具有水平和垂直两个方面的结构。从垂直上看,一个 BSP 程序由一系列串行的超步(superstep)组成，如图所示:



这种结构类似于一个串行程序结构。从水平上看，在一个超步中，所有的进程并行执行局部计算。一个超步可分为三个阶段，如图所示:



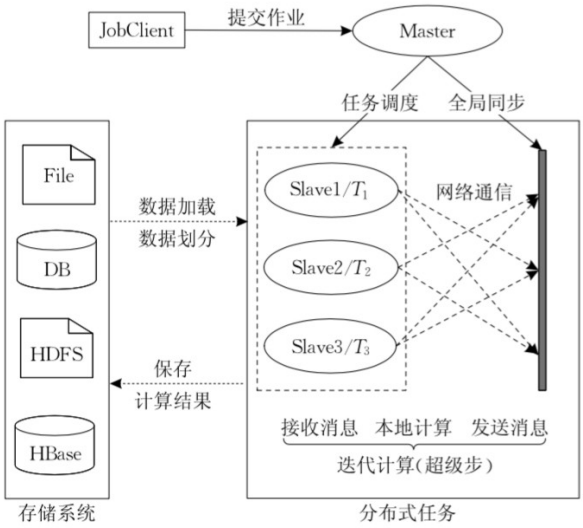
1. 本地计算阶段，每个处理器只对存储本地内存中的数据进行本地计算。
2. 全局通信阶段，对任何非本地数据进行操作。
3. 栅栏同步阶段，等待所有通信行为的结束。

针对这个模型，有各种开源的或者闭源的实现，比如 Google 的 Pregel，Apache 的 Giraph、Hama 等等，总体上，他们的思想是一样的。我们可以用 Pregel 作为例子，简单梳理一遍整个图计算的处理流程：

如下图所示，作为一个以顶点为中心的基于 Master-Slave 架构的系统，Pregel 采用邻接表组织图数据，Master 是分布式系统的控制中心，而 Slave 是工作节点。用户的图处理作业通过 JobClient 向 Master 提交，而 Master 将一个

图处理作业分割为若干任务(Ti)并分配给 Slave 节点执行。

具体的计算流程为：



1） 数据加载/划分，各任务从分布式存储系统并行加载数据到本地，然后进行图划分，每个任务处理一个子图；

2） 迭代处理，每次迭代计算视为一个超步，两个超级步之间通过全局同步来协调各任务处理进度，任务之间通过消息交换中间结果，在每个超级步中，图顶点接受消息，执行本地计算并发送消息，因此计算负载通常由出边数决定；

3） 迭代收敛后，保存计算结果。

致谢

（约为 200 字左右，小四，宋体，单倍行距）×××××××××××。

参考文献（或注）

作者简介（小四，黑体）：文字为 200 字左右（小四，宋体，单倍行距）

×××，男（女），×年×月出生于×地，×年从×中学考入（或由于获×奖励保送进入）北京大学×学院（系），在校期间的全面表现（德、智、体）。

感悟与寄语（小四，黑体）：文字为 200 字左右（小四，宋体，单倍行距）。

指导教师简介（小四，黑体）：文字为 200 字左右（小四，宋体, 单倍行距）

×××，男（女），职称。×年×月出生于×地，主要学术经历及研究方向。

几点说明：

1、“萧政基金/校长基金/毛玉刚基金/教育基金会基金/锺夏校际科研资助基金”论文用英文书写和英文书写均可。

2、论文如果公开发表或被接收在学术刊物上，请按学术刊物的排版格式提交论文。提交 pdf 格式论文的同学，请用 Word 格式提交论文的标题、致谢、作者简介、感悟与寄语、指导教师简介等部分的内容。（若有论文公开发表，请将发表论文与结题报告一起上传至 FTP）

3、论文用 A4 纸单倍行距排版（不要更改 A4 纸页面设置），复杂的图、表格和公式在文中插入时，图号及图名，表号及表名用 5 号字。请注意修改页眉。