

**课程报告**

**课程名称：并行程序设计**

**专业班级： 计算机硕2101班**

**学 号： M202173535**

**姓 名： 蒋子涵**

**指导教师： 石宣化**

**报告日期： 2021.12.29**

**计算机科学与技术学院**

# 大规模图计算系统的并行优化技术综述

## 引言

图 (graph) 是一种包含了一组顶点以及连接他们的边的数据结构，图在结构和语义等方面具有很强的表达能力[1]，能够很好的描述事物之间的关联性。在实际生活中，诸多场景都采用图来描述关联关系。例如社交网络将 用户作为图顶点，而用户好友关系等作为边来构成图；城市道路系统将路口作为 图顶点，道路作为边来构成图等。

图计算 (graph processing 或 graph computing) 是研究客观世界当中事物与事物之间的关系并对其进行完整地刻画、计算和分析的一门处理图的技术, 是大数据关联属性的最佳表达方式。由于图计算能够用于分析数据之间的关联关系，因此其在互联网应用、科学计算、社会计算、商业计算等诸多领域得到应用广泛，已经成为大数据处理的主流模式之一。当前，随着互联网的快速普及，社会的数字化变革以及经济的快速发展，表达数据关联性的图数据的规模正在呈爆发式增长。如何有效地分析海量图数据已引起工业界和学术界的广泛关注，成为大数据处理的关键问题。

## 图计算背景

### 2.1图计算简介

图是一种由顶点 (vertices) 和连接顶点的边 (edges) 构成的数据结构, G = (V, E), 其中 V表示顶点集合, E表示边的集合. e = (vi, vj) 表示从顶点vi到vj的一条有向边. 同时, 每一个顶点与每一条边都有属于自己的属性值. 不同的领域属性值可代表不同的含义. 如社交网络中, 顶点的属性值为个人的热度, 而边的属性值则表示为有关联的两人之间的紧密程度。在此计算过程中，顶点和边都会包含计算过程中的状态数据，然后通过对每个顶点和边进行一系列的迭代计算来完成状态数据的更新。图计算的结果为图中所有顶点和边的最终状态的聚合。

图计算能够挖掘事物之间潜在不易洞察的行为和联系，而这些关联性很难使用传统数据库来描述。这使得图计算在经济建设、国防安全和社会生活等的方方面面都能够发挥重要的作用。

### 2.2图计算相关算法

对于常用的图算法进行总结与归纳, 可以将图算法分为以下两类, 如图2.1所示。

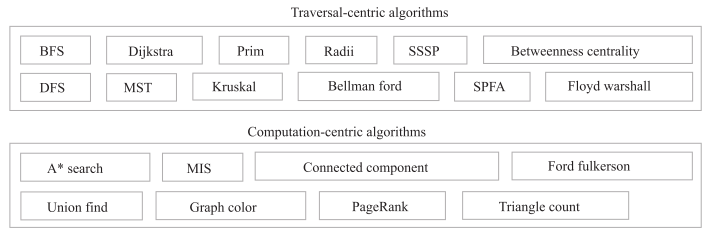


图2.1 图算法分类

1. 遍历为中心的图算法. 该类算法通常需要以特定方式从特定顶点遍历图, 它们的计算内存访问率相对较低, 存在大量随机访问, 大大浪费了存储器带宽。比较经典的算法如广度优先遍历 (breadthfirst traversal, BFS)、单源最短路径 (single source shortest path, SSSP)、中介中心度 (betweenness centrality, BC) 等都属于这种算法。
2. 计算为中心的图算法. 该类算法在一个迭代周期中有大量的运算进行, 在每一次迭代中所有的顶点都会参与. 相比于遍历为中心的图算法, 这类算法的数据局部性较好, 但是在运算过程中会出现大量的浮点运算, 而且每轮迭代所有顶点的参与会增加算法运算过程中对内存的访问频率. 经典的网页排序 (PageRank)、连通分量 (connected component, CC)、三角形计数 (triangle count, TC) 等都属于这种算法。

## 图计算编程模型

为了简单高效得实现并行图计算，目前主流图计算系统采用四种编程模型：以点为中心的编程模型、以边为中心的编程模型和以子图为中心的编程模型。采用合适的编程模型，能够提高计算过程中的并行度，从而可以采用多线程或分布式的方式执行图计算程序。

### 3.1以点为中心的编程模型

由于图数据结构内部存在复杂的依赖关系，图计算系统存在不易扩展和难以并行化的问题。为了解决这些问题，需要提出一种适合图计算的编程模型作为系统设计的基础。Google 在2010年发表的Pregel [2]计算框架所采用的TLAV（Think Like a Vertex）思想，正是针对图计算的特点所采用的一种新式编程模型。该模型把计算范围限定在图数据中的单个点，即从单个点的角度考虑图算法的执行过程，包括每个点上的计算以及相邻点之间的消息传递。这样各个点可以实现相互独立的计算，从而进行细粒度的并行。

如图3.1所示，最早的TLAV是建立在BSP（Bulk Synchronous Parallel）模型之上,即把计算过程分割为多个超级步，超级步之间通过屏障（Barrier）来保证 信息被同时传送和接受，从而避免了数据竞争和死锁。但以点为中心的模型并不局限于上述的同步执行模型，还支持异步执行模型。与同步执行模型不同的是，异步执行模型不需要屏障来限定超级步之间的信息传输，即每个点的更新值可以在当前的超级步中立即可见，从而加速算法的收敛速度。但是，异步执行模型需要考虑数据竞争、一致性等问题。

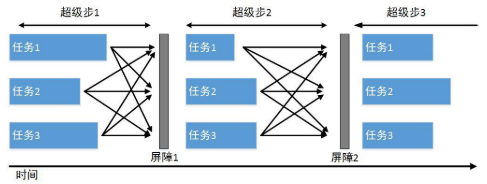


图3.1 BSP模型

以顶点为中心的计算模型可以很方便地表示各种图算法并且可以实现顶点的高并行, 所以这种编程模型已被广泛用于许多图加速器 [2∼5]。然而, 以该模型实现的图计算系统需要对顶点数据和边数据进行大量的随机访存操作，这将导致较大的时间开销。由于现代计算机体系结构的特点，随机访存往往比顺序访存的性能低。因此如何采用顺序访存进行图计算的问题受到业界广大的关注。同时，以点为中心的编程模型需要对原始的图数据进行排序等预处理操作，这也增大总的时间开销。

### 3.2以边为中心的编程模型

如前所述，以点为中心的编程模型存在大量的随机访存和排序预处理等问题，为了解决这些问题，以边为中心的模型被提出[6]。

X-Stream[6]是第一个使用以边为中心的模型，即GAS模型来处理每条边, 主要通过3个步骤迭代执行：(1)收集其源顶点的信息 (gather); (2)更新 (apply); (3)将更新后的值发送到各目标顶点 (scatter)。该编程模型可以实现对边的顺序访存。现有的图加速器通常使用以边为中心的模型来提高其有限内存带宽的利用率 [7, 8]。但与以顶点为中心的编程模型相比, 以边为中心的模型缺乏灵活的调度能力, 同时该模型还可能导致对顶点的大量随机访问。因此, 通常需要额外的优化, 例如细粒度分区和定制的顶点更新策略[7, 9]。