数据库系统概论新技术篇

分布式图处理系统

卢卫

中国人民大学信息学院

2017年6月

提纲

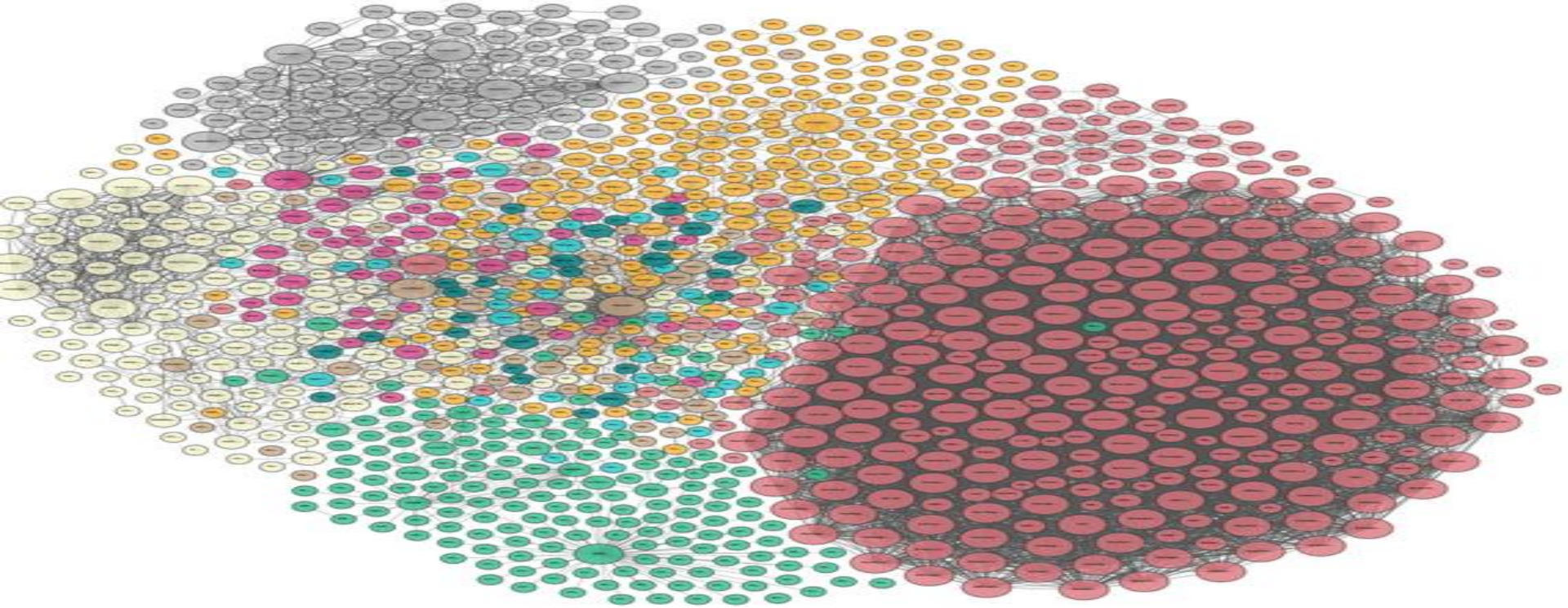
- 分布式图处理系统的必要性
- 谷歌的分布式图处理系统Pregel
- 分布式图算法的实现举例

提纲

- 分布式图处理系统的必要性
- 谷歌的分布式图处理系统Pregel
- 分布式图算法的实现举例

分布式图处理系统的必要性

- 图的数据规模庞大



分布式图处理系统的必要性

- 图的数据规模庞大
  - 社交网络
    - Facebook用户：22亿用户，千亿规模的联系
    - 新浪微博：4亿多个用户，百亿规模的联系
    - 微信和WeChat合并每后个月活跃用户数达8.89亿
  - 互联网
    - 截至2016年12月，中国网页数量为2360亿个

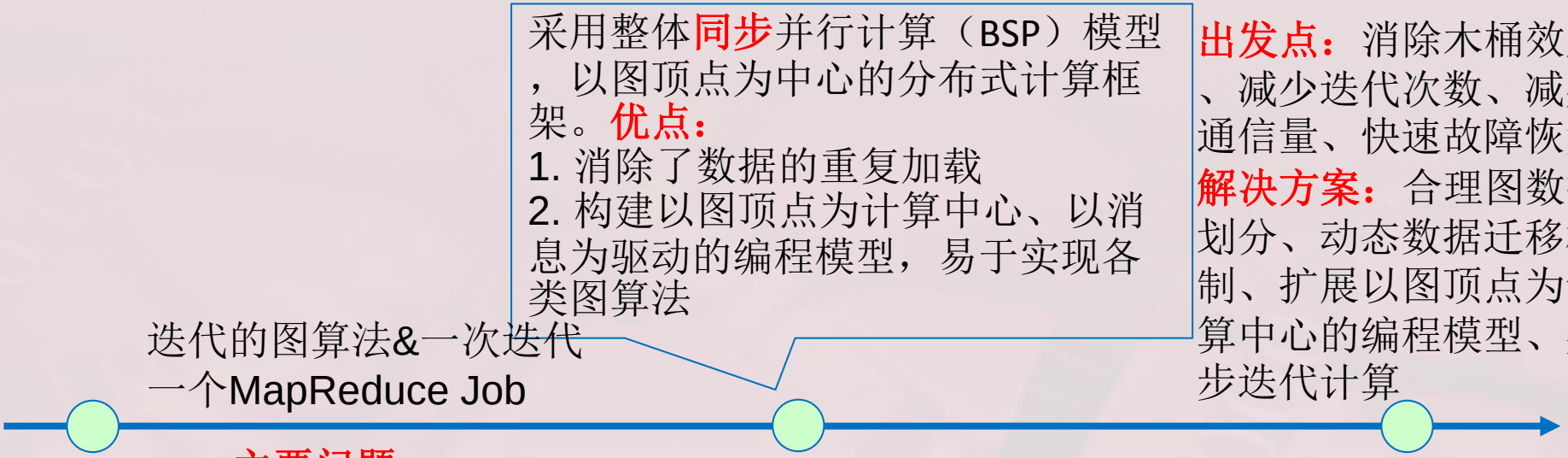
集中式下图数据库的问题

- 存不了：单机的存储能力有限
- 算不出：即使是相对较为简单的最短路径发现操作，其计算时间复杂度是O(n^2)或O(m + nlogn)，其中n是图中顶点的个数，m是边的数目,图数据规模增长导致算法代价迅速增加。更糟糕的是，很多图算法的计算时间复杂度与顶点或边的个数呈超线性关系

分布式图处理系统的提出

- 新的硬件和计算环境提供了可能性
  - CPU、GPU、大内存技术的发展
  - 云计算、大数据技术的出现

分布式图处理系统的兴起

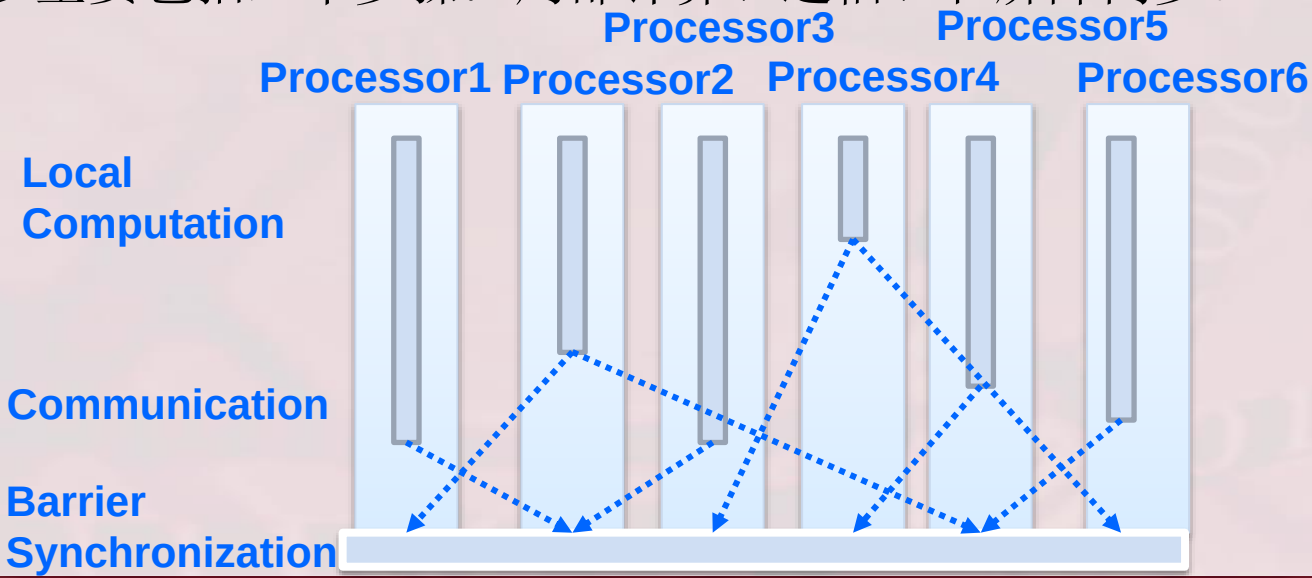


提纲

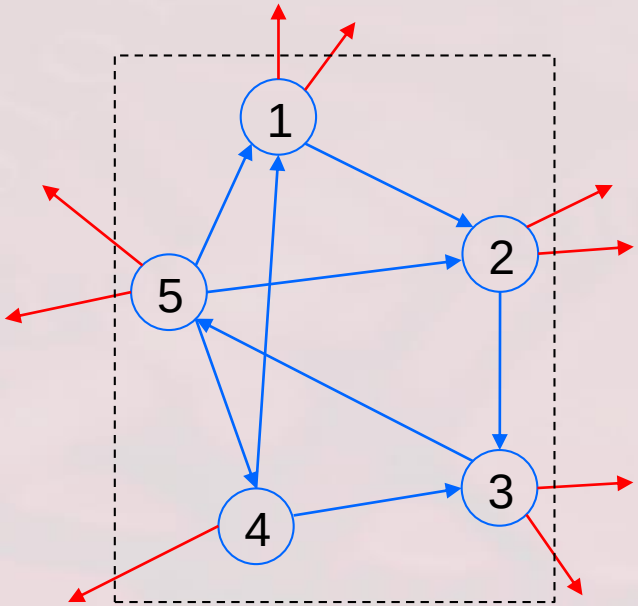
- 分布式图处理系统的必要性
- 谷歌的分布式图处理系统Pregel
- 分布式图算法的实现举例

Pregel计算模型

- Pregel是基于BSP（Bulk Synchronous Parallel，整体同步并行计算）模型实现的分布式图处理系统
- BSP模型的主要思想：一次BSP计算过程包含一个全局超步（Superstep），每一超步主要包括三个步骤：局部计算、通信、和屏障同步。



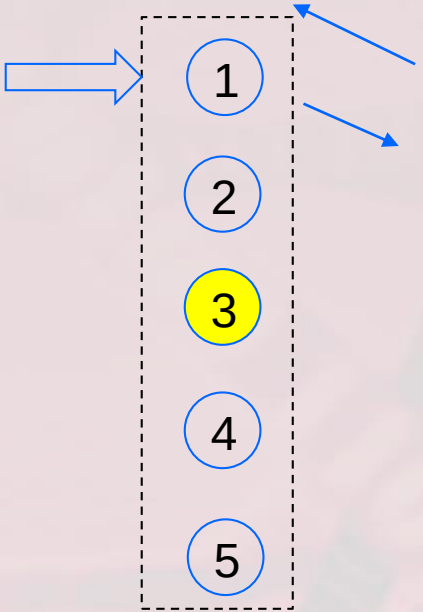
局部计算（Local computation）



Processor1

输入

执行Compute()函数



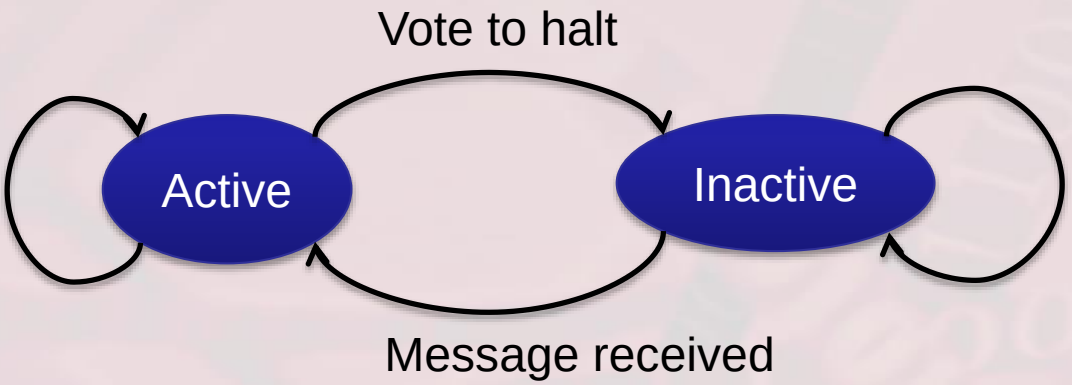
顶点值、上一个迭代收到的消息

更新顶点值、发送其他顶点消息

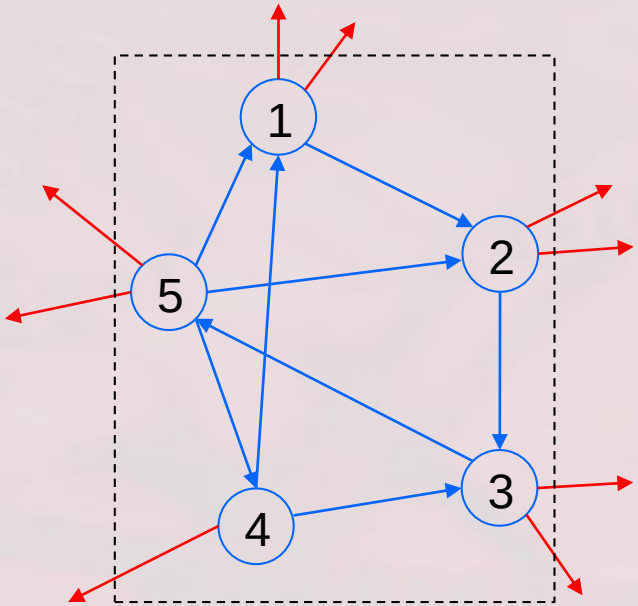
Processor1

❖ 顶点的状态

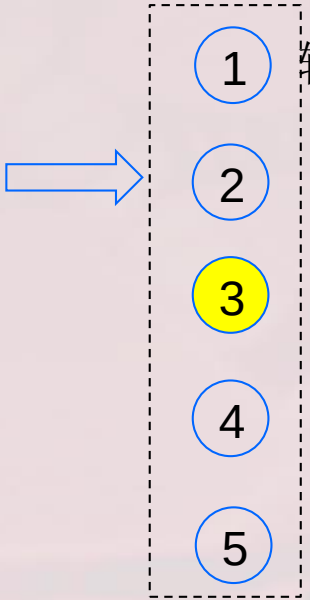
- ■ 活跃与非活跃



局部计算（Local computation）



Processor1



输入

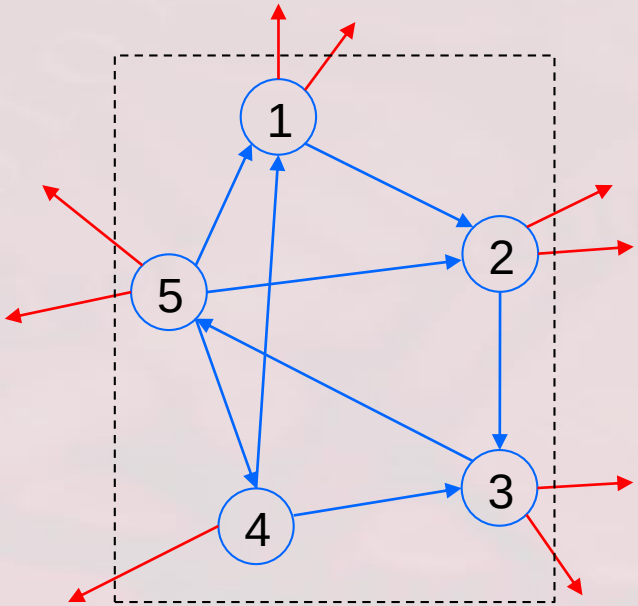
执行Compute()函数

顶点值、上一个迭代收到的消息

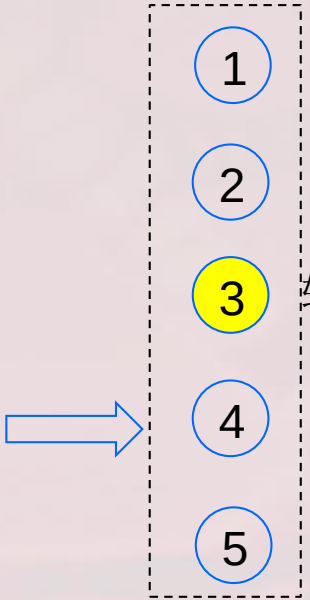
更新顶点值、发送其他顶点消息

Processor1

局部计算（Local computation）



Processor1



输入

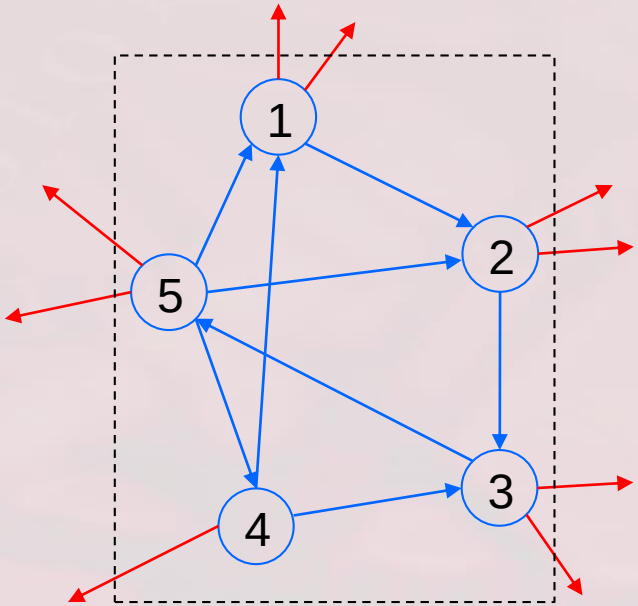
执行Compute()函数

顶点值、上一个迭代收到的消息

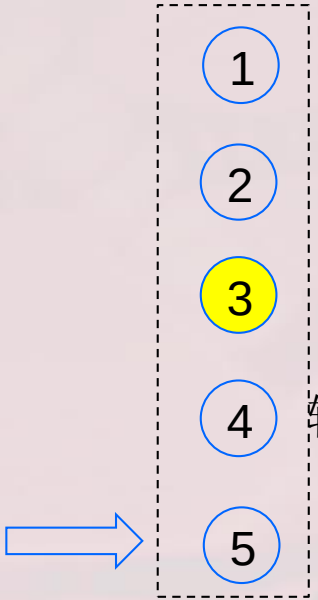
更新顶点值、发送其他顶点消息

Processor1

局部计算（Local computation）



Processor1



输入

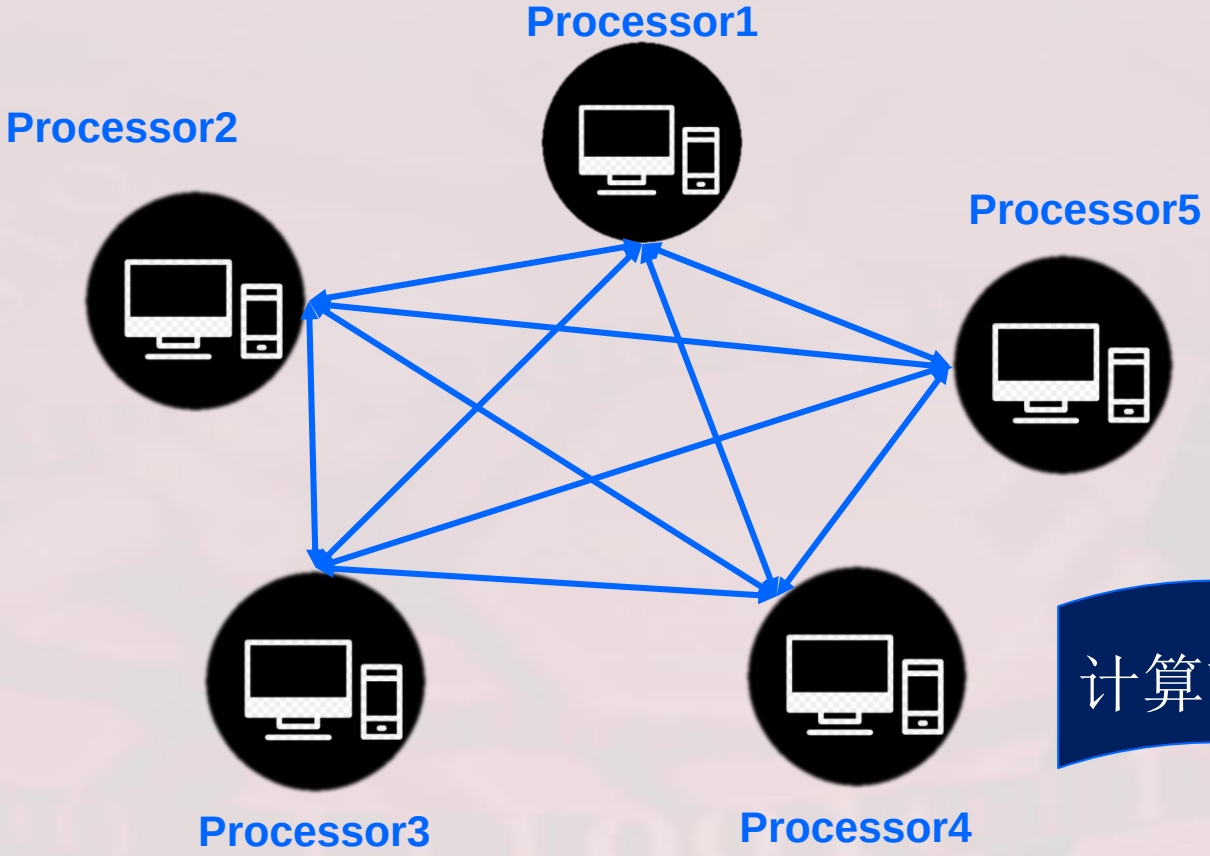
执行Compute()函数

顶点值、上一个迭代收到的消息

更新顶点值、发送其他顶点消息

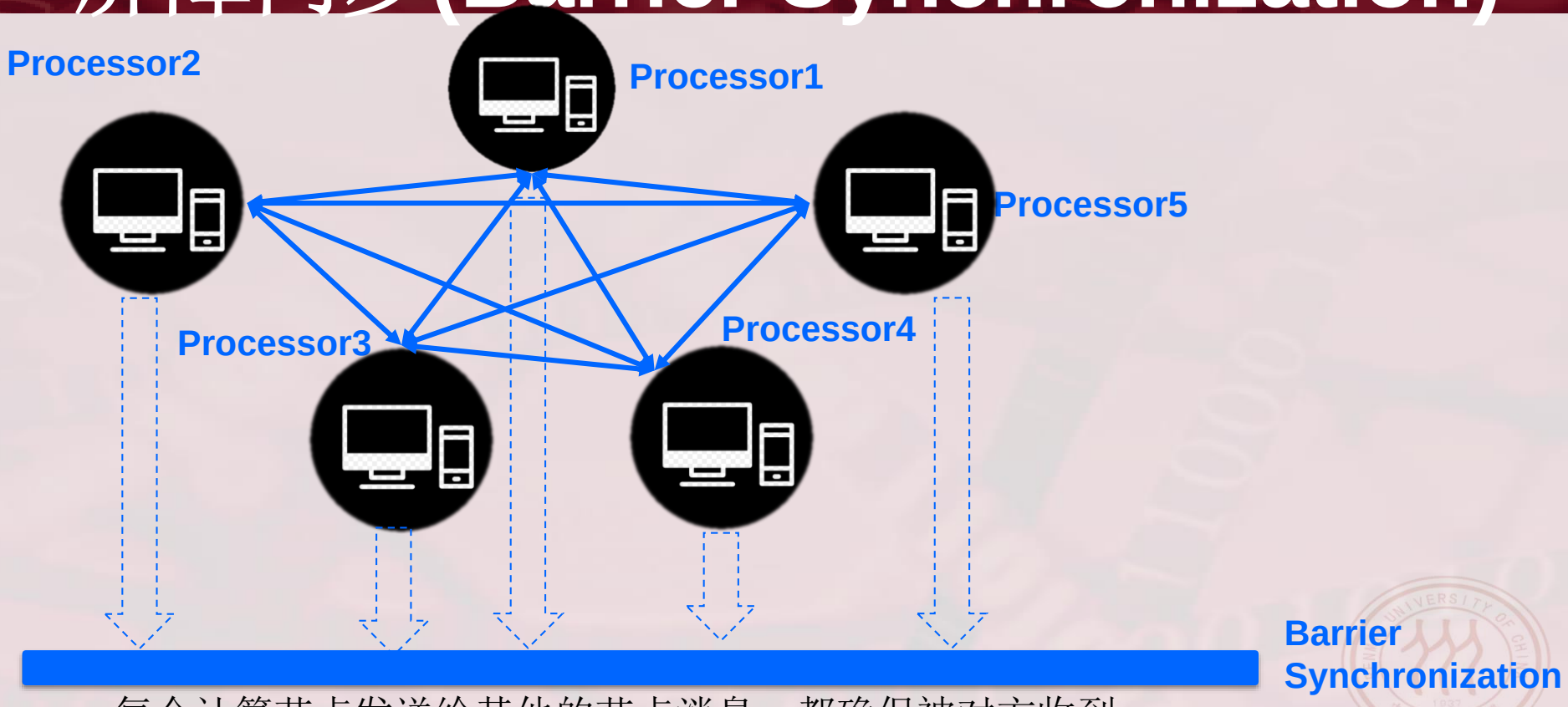
Processor1

通信（Communication）



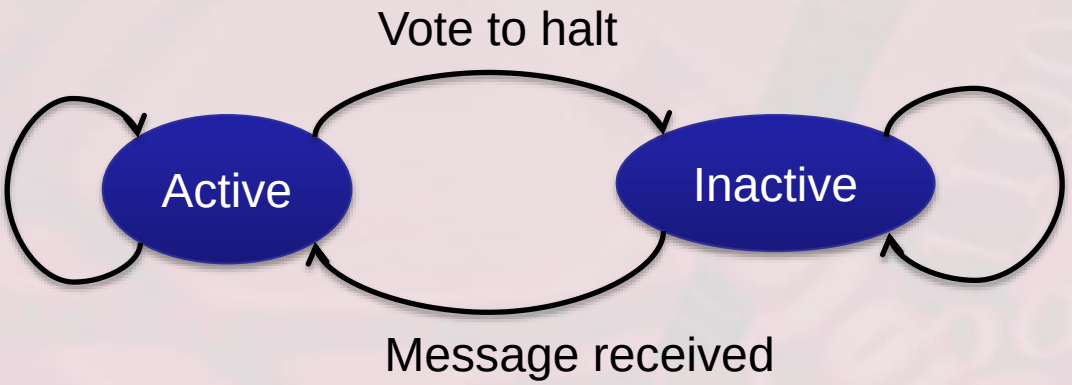
计算节点之间的消息通信

屏障同步(Barrier Synchronization)



每个计算节点发送给其他的节点消息，都确保被对方收到

- 在**BSP**计算过程中超步迭代执行，直到图中所有顶点都被标识为“非活跃（**inactive**）”状态且没有消息在传递。

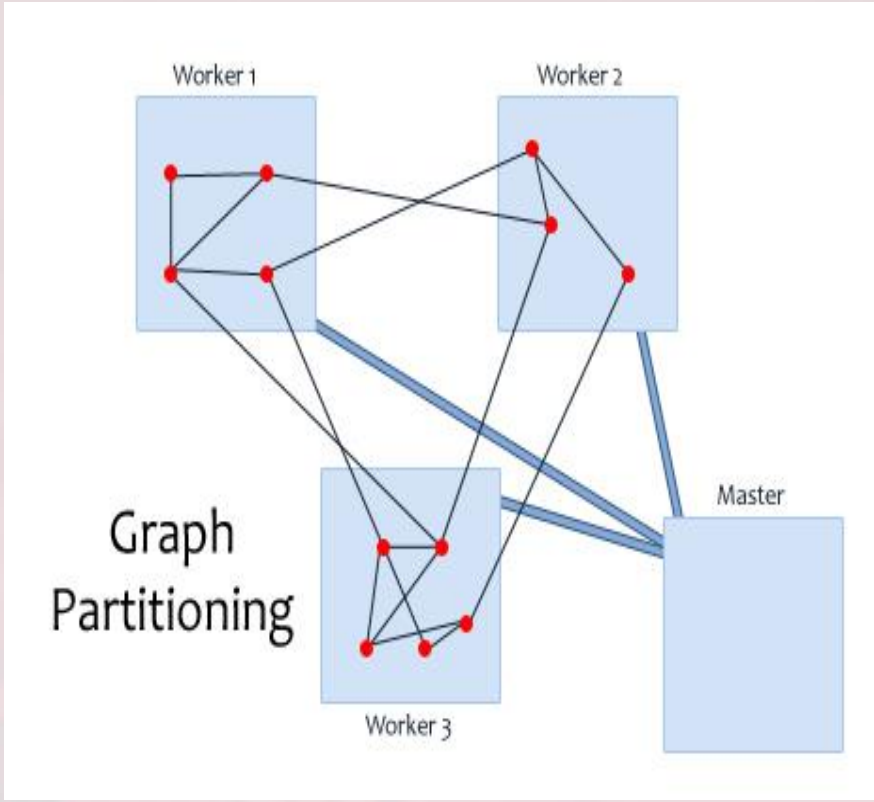


Pregel系统结构

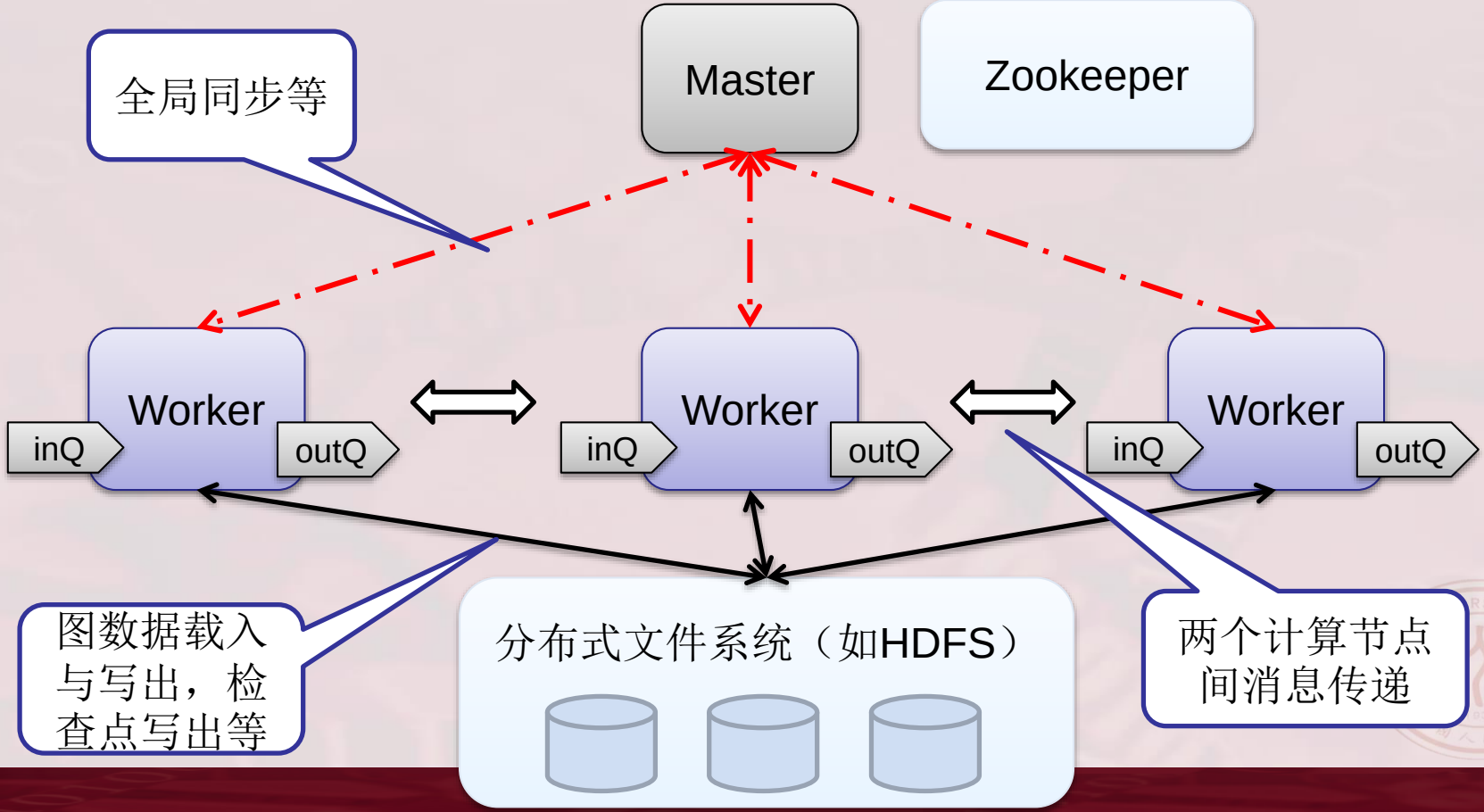
- Pregel采用主从式结构（Master/Worker）
  - Master
    - 任务分配
    - 协调从节点（Worker）的计算和同步
    - 从节点的故障恢复
  - Worker
    - 任务计算
    - 节点间通信
- 数据存储在分布式文件系统中（例如HDFS）
- 临时数据存放在每个从节点的本地磁盘

Pregel系统结构（续）

- Master进行图划分，每个worker读取相应分片
- Master协调worker的计算
  - 每个worker遍历其分片中包含的顶点，执行compute函数
  - 各work之间消息通信；当前superstep的同步
  - 重复执行以上superstep，直到图中所有顶点都被标识为“非活跃（inactive）”状态且没有消息在传递



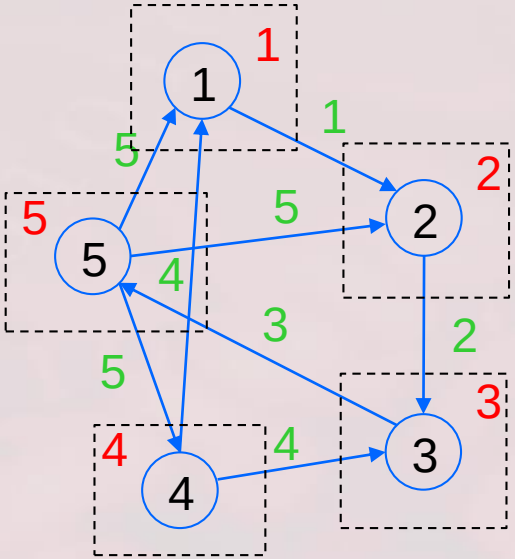
Pregel系统结构（续）



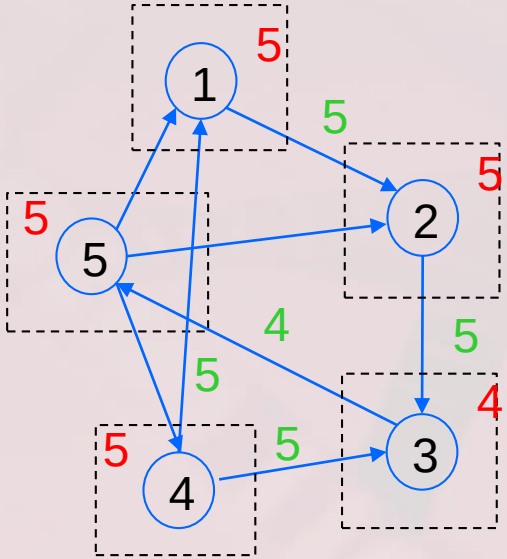
提纲

- 分布式图处理系统的必要性
- 谷歌的分布式图处理系统Pregel
- 分布式图算法的实现举例

在连通图中求最大值

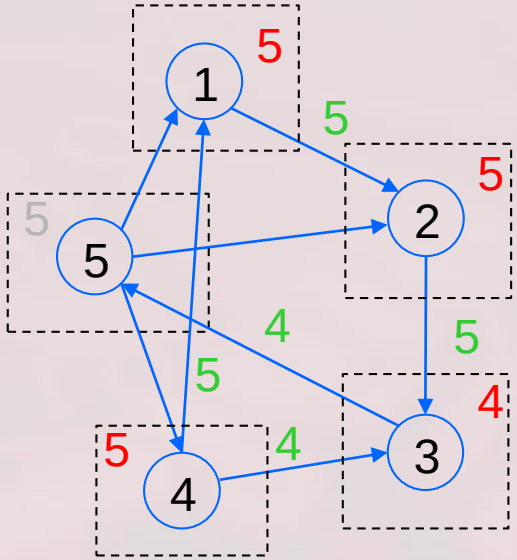


Superstep 0

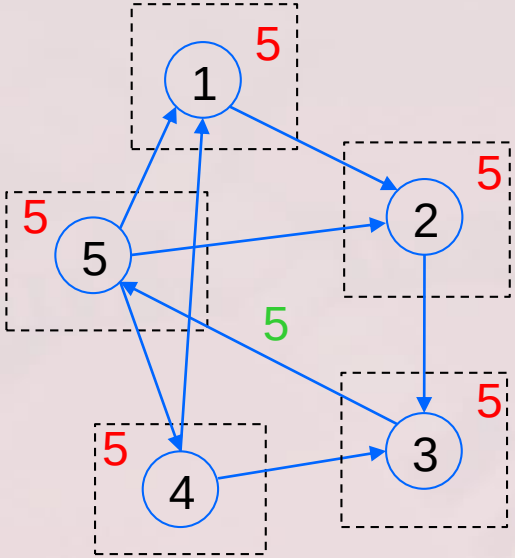


Superstep 1

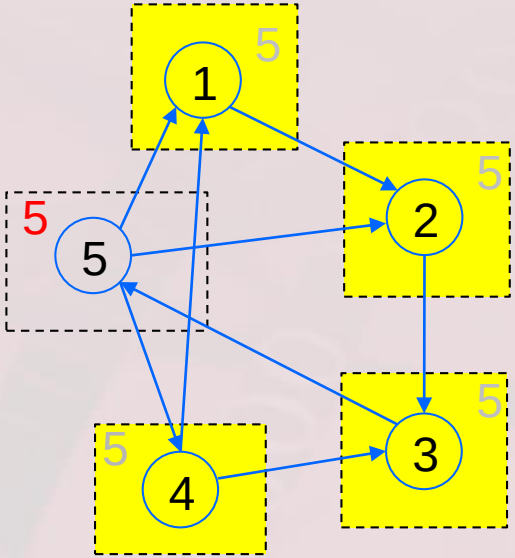
在连通图中求最大值



Superstep 1



Superstep 2



Superstep 3

Pregel的其他特性

- **Combiners :**发送消息，尤其是当目标顶点在另外一台机器时，会产生一些通信开销。某些情况可以在用户的协助下降低这种开销。比方说，假如**Compute()** 收到许多的**int** 值消息，而它仅仅关心的是这些值的和，而不是每一个**int**的值，这种情况下，系统可以将发往同一个顶点的多个消息合并成一个消息，该消息中仅包含它们的和值，这样就可以减少传输和缓存的开销
- **Aggregators：** 是一种提供全局通信，监控和数据查看的机制。在一个超级步**S**中，每一个顶点都可以向一个**aggregator**提供一个数据，系统会使用一种**reduce**操作来负责聚合这些值，而产生的值将会对所有的顶点在超级步**S+1**中可见。