

图计算

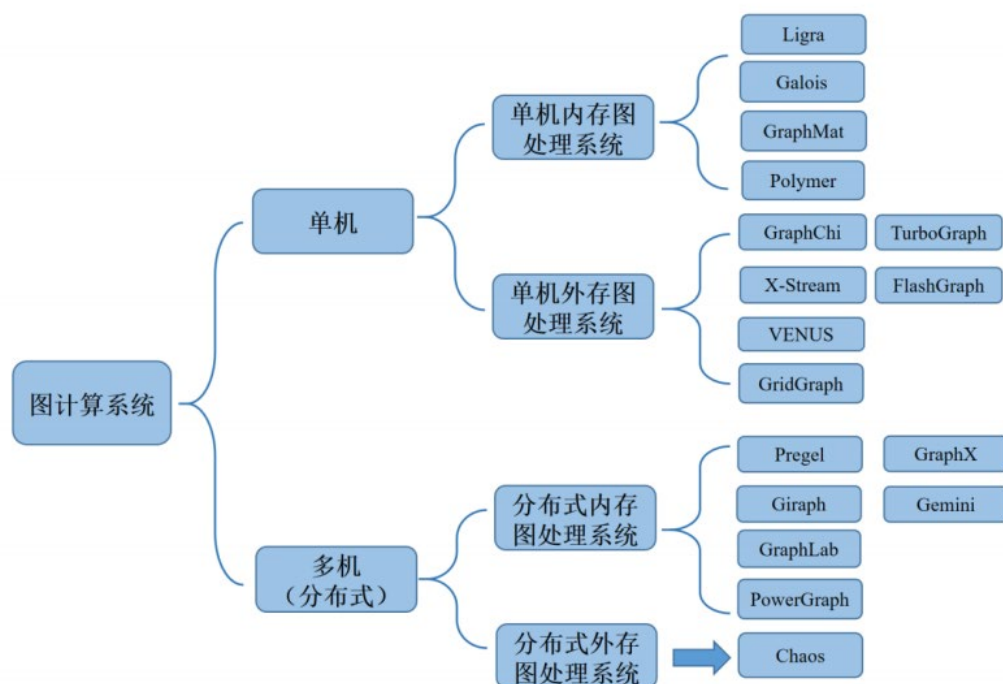


图 4 计算系统框架分类

图计算特点：常常需要进行迭代操作，例如各种遍历算法。

由于 MapReduce 中的中间信息无法缓存，每一步的结果都要进行存入磁盘，在迭代操作中效率较低。

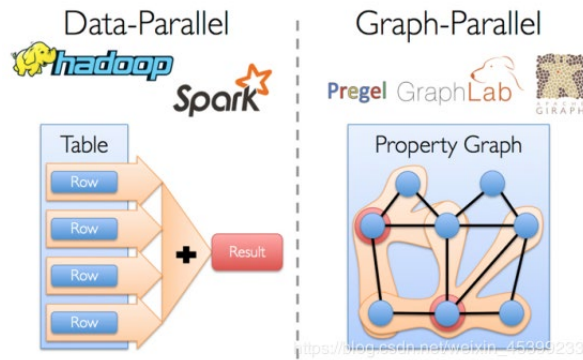
spark 提供了将中间结果缓存的功能，因而相对适合用于图计算。但是由于 spark 本身的特点，计算中其 RDD 的结构始终不变，在过程中可能会存储一些不必要的信息，灵活性较弱。

Pregel

Google 开发了 Pregel。Pregel 以顶点为中心进行计算，在每一轮计算中，活跃节点完成自身状态的更新，并将相关信息发送给其他节点，通常是相邻节点。之后一轮中，这些节点会利用自身原有信息和收到的信息更新自身的信息。在每一轮计算完成后会利用算法保证一致性。该计算系统适合分布式处理。其设计大幅减少了中间结果存储到磁盘的开销。

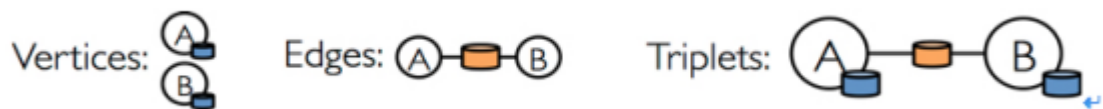
Spark GraphX

基于 Spark 设计的图计算系统



通常进行处理时, 要完成第一部分的结构化处理和第二部分的图计算需要分在两个不同的系统下进行, 造成了大量不必要的开销。由于 GraphX 直接基于 spark, 其中的数据在只存储一份的情况下既可以作为图, 也可以作为普通集合。

存储方式为点分割存储, 即边只存储一份, 点可以存储多份。相对应的是边分割存储, 即点只存储一份, 边存储多份。两者相比, 点分割存储可以提高运算效率, 但是占用更多空间。存储结构:



有较多不同的分块算法 (随机/邻近分块/均衡性..)

Gemini

由于分布式系统中面临大量数据传输/一致性等造成的负担, 其效率有时并不那么理想。在进行某些特定任务时, 有时 100+核的并行计算效率甚至不如经过优化的单核串行计算。因此, Gemini 以计算为核心, 通过尽量避免分布式引入的开销, 在每个节点负责的块的划分等内容以及算法上进行优化, 减少了通信开销。既可以多机运行, 又可以在单机运行下拥有和其他单机计算系统相匹配的效率。