1. 什么情况下会发生负载倾斜？

点与点之间的连接紧密度存在明显差异时会导致负载倾斜。比如在社交网络中一些名人的关注度非常高，这些点的连接了相当多的点，而一些普通人之间的关系相对简单、局限，因而这些点的计算量也就很小。在Web链接数据中也存在同样的问题，事实上这样的倾斜在大部分数据中都存在。

1. 负载迁移是如何进行的？

搜索过程中，每一个子图状态之间都是独立可计算的，因而在前一步的计算过程中因超时或者超大的子图先临时写到本地磁盘，在下一步的作业迭代中将这些子图随机发散到不同的计算节点上均衡负载。

1. Binary算法与BK算法各自的优缺点是什么？

BK在存在数据点整体上连接紧密时可以很快切分子图，Binary可以将数据图依次分割搜索空间小。

1. 并行化算法的目标？

并行化算法的目标是通过将任务划分为可独立计算的单元同时充分利用各个计算节点的计算能力将计算量有效发散到各个节点上，使得可以通过简单的增加机器数目获得计算效率的提升。

1. 切分之后不同的计算节点之间如何通信？

切分之后各个子图之间都是可以独立计算的单元，一个节点上的数据计算不依赖于其他节点，因而不存在数据计算时数据交互。在上一轮作业完成后，进行下一轮作业迭代时需要将上一轮切分出来的子图通过MapReduce的框架重新发散，这里各个节点间存在一次数据交互。

1. 最多有多少个节点，加速比如何？

目前的实验中用了10个计算节点1个主节点的并行，加速比多数情况下较好。个别情况由于数据中只存在个别难切分点或者本身计算量较小导致加速比不佳。