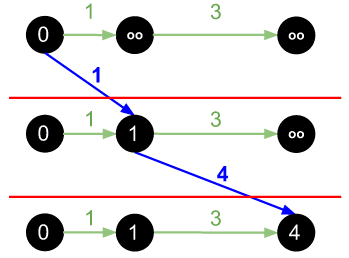
Apache Giraph是一个反复迭代图形处理框架，内置在Apache Hadoop上。

Giraph的计算输入是一个由定点和边组成的图形，如图1. 比如说，图的顶点可以代表人，而边代表朋友请求。每个顶点和每条边都存储一个值。因此，输入的图形不仅决定了图形的拓扑序列，同时也确定了顶点和边的初始值。

举个例子，假设一个计算要求找出从一个预先确定的初始人源S和在社交网络中的任意一个人间的距离，图中每条边E的值是浮点数，代表和相邻人之间的距离。顶点V的值也是浮点数，代表从源顶点S到顶点V的最短距离的上界。预先确定的源顶点S的初始值是0，其余顶点的初始值是。

**顶点值**

**边值**

**信息**

**Superstep 分界**

图**错误!文档中没有指定样式的文字。**‑1 图中是在Giraph中执行的单源点最短路径算法。输入是链式图，有三个节点（黑色的）和两条边（绿色的）；两条边的值分别是1和3.这个算法从最左边的顶点开始计算。顶点的初始值分别是0，（第一行）。距离上线作为消息传递，这样就更新紧接着的下一行顶点值的更新。这个执行过程进行的三步superstep（由红线分割）。

以一些列迭代方式进行的步骤在BSP中称为supersteps。最初，每个顶点都是活跃的。在每步superstep中，每个活跃节点都调用由用户提供的计算方法。这个算法将在输入的图形上执行。直观的看，在设计Giraph算法时应面向顶点考虑。一个面向图形方法将会在后面提到。

计算方法：

从前一步的superstep接受信息，并传递到顶点上；

计算用到的消息，顶点值和传出的边值都可能导致值的变化；

并且可能将消息传递到其他顶点上；

这种计算方法并没有直接和其他顶点值和他们的发出的边值有直接的联系。顶点之间的联系主要是靠发送消息。

在单源点最短路径的示例中，一个计算方法将会：（1）找到任何信息的最小值到达；（2）如果值比当前顶点的值要小，则（3）最小的值想回作为新的顶点值，并且（4）该值加上边值的和将被送往每条发出的边。参考图2的简单的代码。

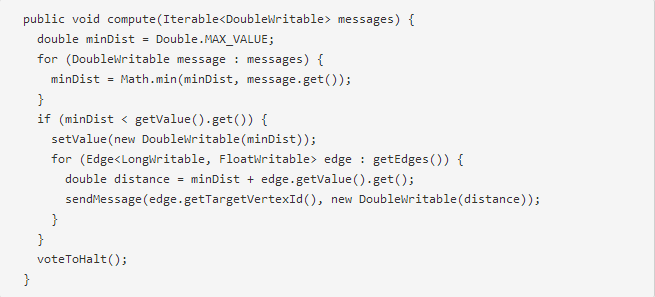


图2 这是单源最短路径计算方法。图中的每个顶点在每个superstep都执行一遍这个方法。该方法计算到达的消息的最小距离，并且可以沿着每条边传送距离。

在相邻的superstep间都有一个分界，这表明：（1）在任何当前superstep传送的消息只能传送到下个superstep中的顶点；（2）顶点将在每个顶点完成计算当前superstep后再开始计算下一个superstep。

当增加或删除顶点或者边时，图会有所改变。我们在例子中的最短路径算法并没有改变图。

在跨过分界线值是保留的，也就是说当图的拓扑序列没有被改变时，在最开始的superstep中任何顶点或者边的值和最后一个superstep中的值是相同的。例如，当一个顶点已设定的距离上界为D，则在接下来的开始的superstep的上界也是D。当然顶点可以在任何superstep中修改顶点或是发出边的值。

任何顶点可以在任何superstep后停止计算。也就是说这个顶点就不再活跃。然而任何接受到的消息都将该顶点变为活跃。

当所有顶点都停止并且没有任何消息在传递中时，这个计算就停止。每个顶点都输出本地信息，通常就是最终的顶点值。

这种