```
public class SimpleClusteringMaster extends Master {

  @Override
  public void compute(int nextSuperstepNo) {
     if (nextSuperstepNo == 1) {
         pickKVerticesAndPutIntoGlobalObjects();
         getGlobalObjects().put("comp-stage", new IntGlobalObject(CompStage.CLUSTER_FINDING_1));
     }  else {
        int compStage = getGlobalObject("comp-stage").value();
        switch(compStage) {
        case CompStage.CLUSTER_FINDING_1:
            getGlobalObjects().put("comp-stage", new IntGlobalObject(CompStage.CLUSTER_FINDING_2));
        break;
        case CompStage.CLUSTER_FINDING_2:
              if (numActiveVertices() == 0) {
                 getGlobalObjects().put("comp-stage", new IntGlobalObject(CompStage.EDGE_COUNTING_1));
              }
        break;
        case CompStage.EDGE_COUNTING_1:
            getGlobalObjects().put("comp-stage", new IntGlobalObject(CompStage.EDGE_COUNTING_2));
        break;
         case CompStage.EDGE_COUNTING_2:
             int numEdgesCrossing = getGlobalObject("num-edges-crossing").value();
              if (numEdgesCrossing > threshold) {
                  pickKVerticesAndPutIntoGlobalObjects();
                  getGlobalObjects().put("comp-stage", new IntGlobalObject(CompStage.CLUSTER_FINDING_1));
              } else {
                  terminateComputation();
              }
         }
    }
}
```