负载均衡模块针对主要实现两个功能，首先是读取容器的状态数据，然后是通过对容器的状态计算得到下一个任务访问的Geth客户端；针对任务，该模块从Rocket中读取任务组后判断任务的类型，然后根据第三章的任务分发算法决定任务的执行顺序。主要函数核心代码如下所示：

|  |
| --- |
| 代码：负载均衡 |
| 输入：  服务节点的状态数组：*gethStats*；  任务数组：*tasks*；  输出：  下一个任务：*targetTask*；  目标服务节点：*targetGeth*；  核心代码：  *LoadTaskToGeth(GethStat[] gethStats, Task[] tasks )*   1. *int lbh = 0; //初始化健康参数lbh* 2. *List<GethStat> gethPool; //初始化可用服务节点池* 3. *for gethStat in gethStats* 4. *//x1代表geth的内存占用率，x2代表cpu占用率* 5. *//Disave的定义见3.2节* 6. *lbh =lbh + Disave – Math.sqrt(gethStat.x1^2 + gethStat.x2^2)* 7. *if Disave – Math.sqrt(gethStat.x1^2 + gethStat.x2^2) < 0* 8. *then gethPool.add(gethStat)* 9. *if lbh < 0 then goto 3 //如果健康指数小于0持续回收服务节点* 10. *List<Task> taskQueue //初始化任务执行序列* 11. *for task in tasks* 12. *if task.type == w* 13. *then taskQueue.add(task)* 14. *else* 15. *taskQueue.addFirst(task)* 16. *return gethPool.get(0), taskQueue.get(0)* |