关于用RPC和分布式存储改造Lotus的方案

# 背景

Lotus目前将PoRep分成了四个阶段，分别是precommit1/ precommit2/ commit1/ commit2。这四个阶段，有些阶段只能使用单核，有些阶段可以使用多核，而在不同的阶段之间，需要使用不同的数据传输方式。如果将不同的阶段配置在不同的机器上，这些复杂的方案将导致集群的配置困难，同时也会对网络有巨量的要求。

友商给我们提供了一个非常有借鉴意义的方案：在一台机器上运行这四个阶段，通过大量的worker使这机器的CPU跑满；然后通过分布式存储系统把各台机器连接起来，形成一个大的矿池系统。

因方案就是根据此原理的方案实现说明。

# 设计说明

我们知道，Lotus将节点分成两大类和多个小类，两大类是指miner和worker，多个小类是指worker中可以分别执行四个阶段。

本方案目标是实现

1. 通过配置让每个worker的cpu跑满
2. miner和worker之间通过RPC进行通信
   1. 当要生成区块时，由miner向worker发送指令，由worker进行所有的区块证明和生成工作
   2. 所有的区块证明有生成工作都在同一个worker上进行，miner管理哪个worker对应了哪些区块的工作，或者说miner在调用precommit1/ precommit2 /commit1 /commit2的时候，总是会根据自身的记录调用到对应的worker提供的服务。
   3. 当要后续证明的区块的时候，有两种方式可以实现
      1. 仍旧使用RPC，通过RPC让相应的worker给出区块证明，返回给miner
      2. 使用分布式存储系统，把所有的worker和miner的存储合并到一起统一访问，这样miner在需要证明的时候，直接通过分布式存储系统获得数据进行证明。

* 在本方案中，使用分布式系统方案，最主要的原因是后续的证明需要一定的时间完成，仍旧RPC的话，可能证明的时间会过长。

为了对原来代码进可能少的改动，我们引入了两个agent角色，这样系统中的角色就从图1变化成了图2

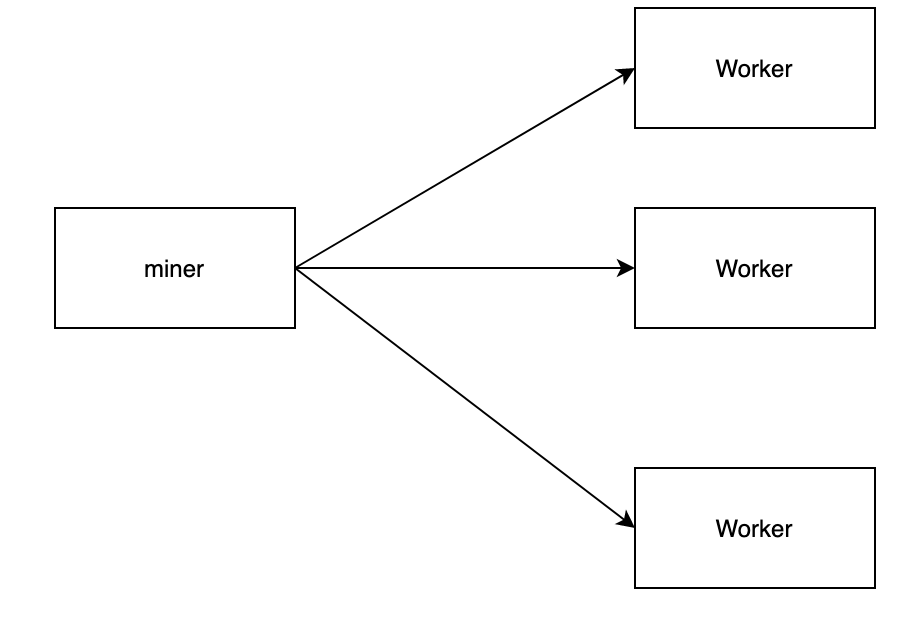


图1 原始的工作方式

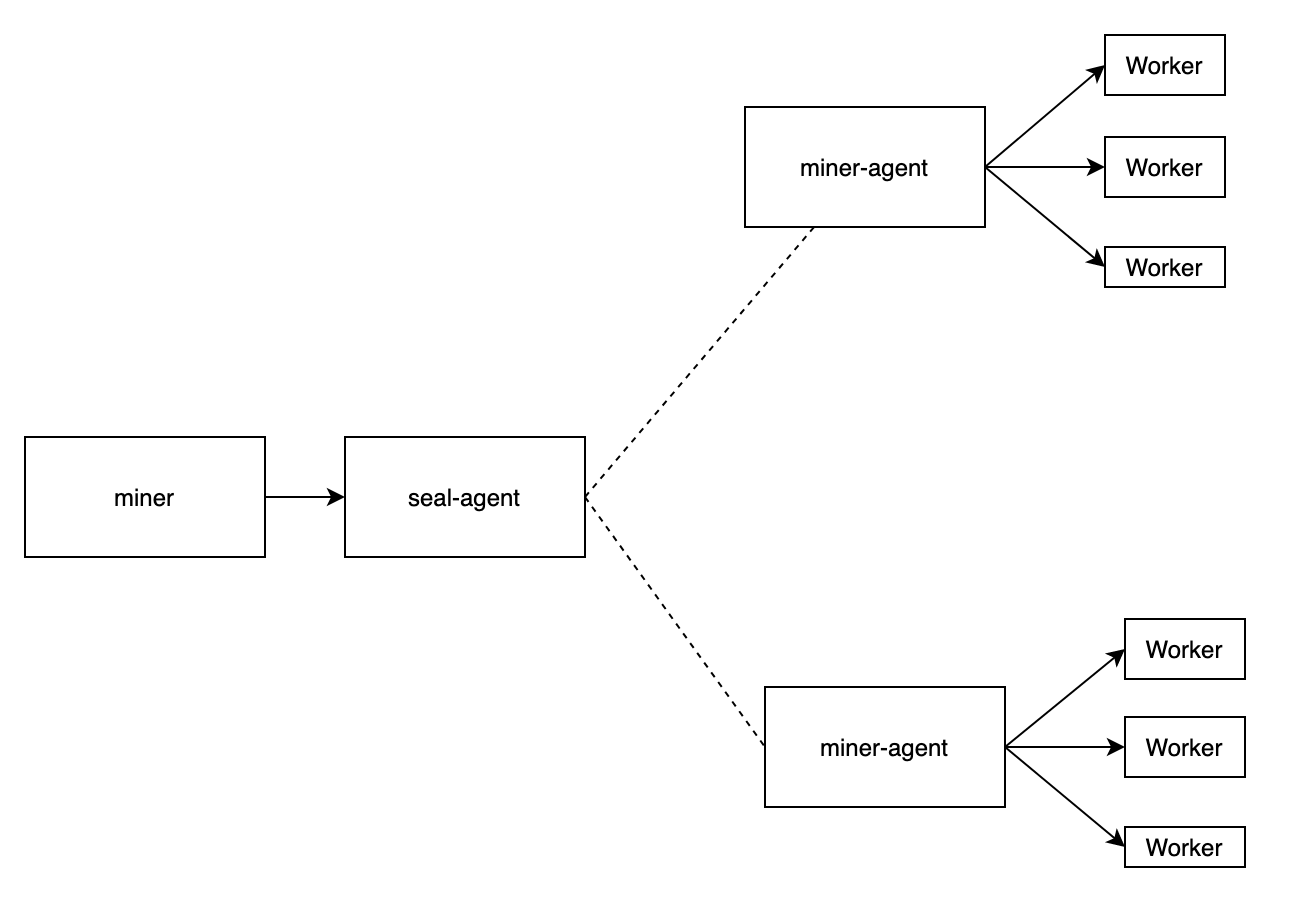


图2：RPC工作方式

注意，在原始的工作方式中，worker是一个集群中的任何一台机器，而在图2的工作方式中，每个miner-agent都是一台机器，所有的worker都在本机器下面。因此在新的方式下，并不需要大量的网络数据传输。证明所需的在分布式存储上的数据是每次8M字节。

## RPC过程描述

为了使代码尽可能地少改动，插入了seal-agent和miner-agent两个对象，miner和seal-agent的功能在一起，将原有的需要worker进行的操作在seal-agent上都有一个对应的代理函数。当miner执行某个rpc操作时，由seal-agent寻找相应的miner-agent，并且分发给他。Miner-agent请求worker执行，执行完毕后，把结果返回给seal-agent，seal-agent再返回给miner。

上述的说明展示了整个RPC的流程，因此实现需要按照以下几个步骤：

1. 编写seal-agent和miner-agent
2. 找到需要改造成rpc调用的函数
3. 把seal-agent侧提供与原来格式一致函数接口给miner
4. 在miner-agent做对等的函数，函数调用原来的实现

RPC的实现方案已经在我们的github上实现了一个初始版本的方案。是在以前的lotus的版本上实现的，新的版本需要重新移值。

原有的实现的代码 <https://github.com/xjrwfilecoin/lotus/tree/v202/cmd>