[](http://image.baidu.com/i?ct=503316480&z=0&tn=baiduimagedetail&word=%D6%D0%B9%FA%D2%C6%B6%AF+logo&in=8288&cl=2&cm=1&sc=0&lm=-1&pn=4&rn=1&di=508018540&ln=140)

**中国移动专利申请**

**技术交底书**

|  |  |
| --- | --- |
| **公司编号** |  |
| **发明名称** | 一种通用的分布式节点管理框架 |
| **申报单位** | 中国移动（深圳）有限公司 |
| **申报类型** | 发明 |
| **发明人** | 陈广胜 |
| **技术联系人** | 陈广胜 [chengs@chinamobilesz.com](mailto:chengs@chinamobilesz.com) 13823799806 |

|  |
| --- |
| **注意事项**  1．技术联系人应为深入了解本申请提案技术方案的技术人员，如交底书撰写人，负责向专利审核人员和代理人解释技术细节、修改交底书、审核申请文件等工作, 请务必填全技术联系人的姓名、E-mail、手机。  2．请按照集团公司提供的本技术交底书模板逐项填写，除交底书第八部分为可选项外，其他均为必须填写的内容。填写不全的专利申请提案，集团公司不予立案。  3．专利申请不要求已具体实现或实施，形成完整的技术方案即可提交申请，特别是需要向合作方公开、向标准提案或以其他方式公开的重要技术构思应在公开前尽早申请。  4．技术交底书文件命名要求：发明名称＋短横线（半角）＋交底书＋版本号，例：一种短消息群发方法-交底书v1.doc |

**中国移动通信集团公司**

一、发明名称

一种通用的分布式节点管理框架。

**注：发明名称反映了本技术方案的核心内容和类型。**

二、技术领域

本发明涉及云计算领域，更具体地说，是一种通用的分布式节点管理框架，不仅可以管理物理节点，还可以管理虚拟节点。

**注：在列出的技术领域中选择了与本技术方案最为相关的领域。**

**。**

三、现有技术的技术方案

要实现分布式节点的管理，例如网元节点的管理，通常是在各节点上部署代理程序，通过代理程序将节点接入到中心管理节点，中心管理节点通过发送命令到代理程序上执行，从而达到管理分布式节点的目的。所以，分布式节点的管理主要包含以下两个部分：

**中心节点**有以下特点

1. 管理节点要保存被管节点的接入信息，以便和被管节点通信；
2. 管理节点作为中心，要做主备来保持高可用，如果做集群的话，需要自己保证集群内部数据和事物的一致性；
3. 管理节点和被管节点不保持长连接，在需要下发命令的时候，发起连接；
4. 管理节点监听特定端口，接受被管节点发来的告警信息和数据；

**代理程序**有以下特点

1. 代理程序主动向管理节点接入；
2. 代理程序接收管理节点发过来的消息，调用对应接口执行任务，新增任务的话，需要新开发对应接口；
3. 代理程序定时采集节点性能数据上报给管理节点；
4. 代理程序定时上报心跳给管理节点，报告节点的状态；
5. 代理程序多采用Java/c++语言编写；

**注：描述了现有技术的具体实现，尤其是说明了在现有技术中的实现中的使用，使人能够符合逻辑地得出第四部分中现有技术存在的缺点。**

四、现有技术的缺点及本申请提案要解决的技术问题

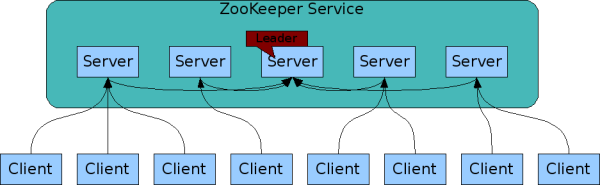
根据上面三种描述特点不难看出，现有技术有以下几个问题

1. 管理端“中心化”，高可用性难保证；
2. 管理端和代理程序耦合紧密；
3. 容易达到管理端性能瓶颈；
4. 代理程序要增加功能的话，需要新开发接口，重新部署；
5. 由于不是保持实时连接，节点发生故障的话，不能及时感知；

**注：逐一说明了各项现有技术存在的缺点，与第三部分现有技术的方案的描述相呼应；并对应于现有技术的缺点，提出本发明要解决的技术问题。**

五、本申请提案的技术方案的详细阐述

ZooKeeper是Hadoop的正式子项目，它是一个针对大型分布式系统的可靠协调系统，提供的功能包括：配置维护、名字服务、分布式同步、组服务等。ZooKeeper的目标就是封装好复杂易出错的关键服务，将简单易用的接口和性能高效、功能稳定的系统提供给用户。其吞吐量达到大约每秒10000基于写操作的工作量，对于读操作的工作量还要高几倍。



在Zookeeper中,znode是一个跟Unix文件系统路径相似的节点,可以往这个节点存储或获取数据.如果在创建znode时Flag设置 为EPHEMERAL,那么当这个创建这个znode的节点和Zookeeper失去连接后,这个znode将不再存在于Zookeeper里.Zookeeper使用Watcher察觉事件信息,当客户端接收到事件信息,比如连接超时,节点数据改变,子节点改变,可以调用相应的行为来处理数。

根据上述描述的zookeeper的特点，本案的设计思路如下图所示，当有一个agent接入管理器的时候，在zookeeper中就会生成一个对应的node目录，该目录下有配置数据conf data，并且，zookeeper跟agent之间保有临时连接，agent通过该连接监听该conf data的变化，当conf data中的cmd对应的值被修改为doInstall的时候，agent1立刻感知到该变化，并找到对应的install.py脚本执行，从而达到任务下发的效果。当agent1执行完脚本成功后，修改conf data中的isSuccess值为1，表明任务执行成功。具体示意图如下：



根据以上思路，我们抽象出一种通用的分布式节点管理框架如下图，其中的**北向接口**：用于供第三方平台调用，用于控制下发命令；

**管理节点**：由zookeeper集群构成，用于分布式协调管理Agent；

**Agent：**部署在节点上的代理程序，用于管理节点；

由于python语言的可移植性，只要在任何节点上部署代理程序，即可将该节点纳入管理节点进行管理，所以它是一种通用的框架，可以管理包括物理设备、虚拟设备、应用容器、hadoop节点等。



下面是该框架的运行视图，其中各个组件的功能介绍如下：

**连接初始化**：用于和zookeeper建立临时连接；

**控制台**：命令下发入口，供上层调用；

**数据采集**：用于采集代理上报的数据，供上层调用；

**节点管理目录**：用于节点管理的目录，每个节点对应一个，里面存放配置文件；

**状态同步目录**：用于节点状态同步的目录；

**数据采集目录**：用于节点数据采集的目录；

**命令处理**：用于接受命令并执行；

**状态同步**：用于节点的状态同步；

**数据上报**：用于节点的数据上报；



一个命令下发的流程图如下：



六、本申请提案的关键点和欲保护点

1.利用zookeeper集群作为分布式协调管理器；

2.利用python做代理程序实现对节点的管理；

**注：提炼出了本方案中作出创新性改进的重要技术点，并按照重要性顺序逐一列出。**

七、与第三条中最接近的现有技术相比，本申请提案有何技术优点

从以上描述可以看出，这个框架相对于传统框架具有以下特点：

1. 使用zookeeper做分布式协调，具有高可用和高性能，并且去“中心化”；
2. 使用zookeeper，使的控制端和代理端解耦；
3. 由于zookeeper里面的znode可以动态增加和减少，使得被管理节点可以动态增加和减少；
4. 由于agent监听着zookeeper里面的znode，并且使用watcher观察事件的，使得zookeeper能够实时感知被管理节点的变化；
5. Agent采用python开发，能够在业务不中断的前提下，实现新功能的上线；
6. 通过zookeeper，能够实现代理的自动升级自管理；
7. 通过zookeeper，可以实现对节点的集群管理；

**注：对应于第四部分中本发明要解决的技术问题，逐一写明了在解决技术问题之后取得的技术效果。**

八、其他有助于理解本申请提案的技术资料

Zookeeper官网：

<http://zookeeper.apache.org/>

**注：该标准与本发明密切相关，且对理解本发明有帮助。**

**提醒：本技术交底书示例所述技术方案尚未公开，请注意保密！**