## ZooKeeper 提供的通用服务如下

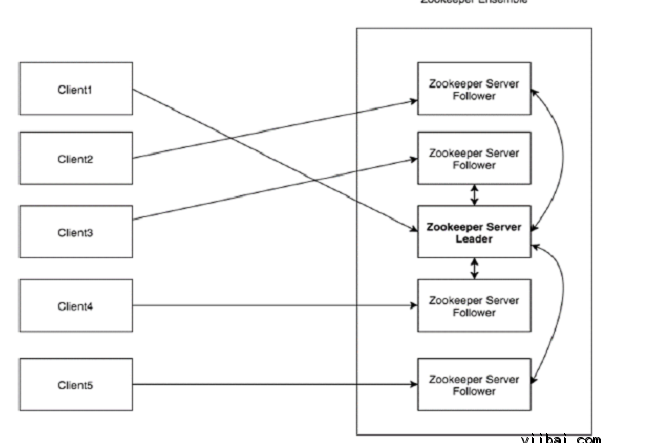
* 命名服务 − 确定在一个集群中的节点的名字。它类似于DNS，只不是过节点。
* 配置管理 − 系统最近加入节点和向上最新配置信息。
* 集群管理 − 加入/节点的群集和节点状态实时离开。
* 节点领导者选举 − 选举一个节点作为领导者协调的目的。
* 锁定和同步服务 − 锁定数据，同时修改它。这种机制可以帮助自动故障恢复，同时连接其它的分布式应用程序。如Apache HBase。
* 高可靠的数据注册表 − 一个或几个节点的可用性的数据向下。

## ZooKeeper的优点

* 简单的分布式协调过程
* 同步 − 互斥和服务器进程之间的合作。这个过程有助于Apache HBase 的配置管理。
* 有序消息
* 序列化− 根据特定的规则进行编码数据。确保应用程序不断地运行。这种方法可以用来在MapReduce的协调队列以执行正在运行的线程。
* 可靠性
* 原子性 − 数据传输成功或完全失败，但没有事务处理部分。

## ZooKeeper的体系结构

看看下面的图。它描绘ZooKeeper 的“客户端 - 服务器架构”。



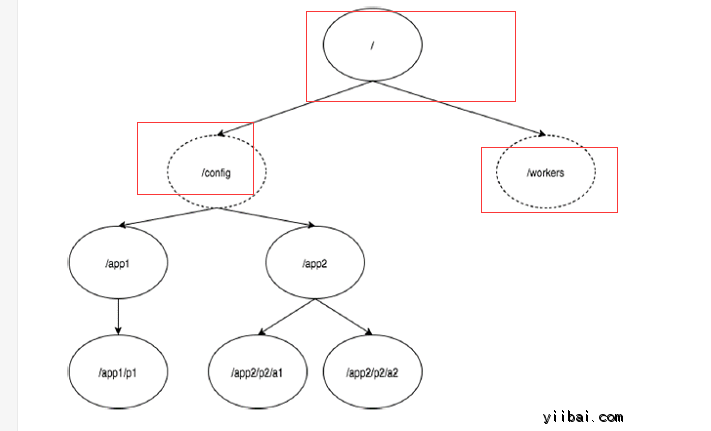
ZooKeeper 架构的一部分组件如下表中所解释。

|  |  |
| --- | --- |
| **部分** | **描述** |
| Client | 客户端，在我们的分布式应用集群的一个节点，从服务器获取信息。对于一个特定的时间间隔，每个客户端将消息发送到服务器，以让服务器都知道客户机是活的。  同样，服务器会发送一个确认当客户端连接。如果没有从所连接的服务器的响应，客户端自动重定向消息到另一个服务器 |
| Server | 服务器，ZooKeeper集成的一个节点，提供所有的服务提供给客户。给出应答客户，告知该服务器还活着 |
| 合组 | ZooKeeper 服务器组。节点所需要形成的合奏的最小数目为3 |
| Leader | 它执行自动恢复，如果任何连接的节点的故障的服务器节点。领导者服务启动 |
| Follower | 遵循领导指示服务器节点 |

## 分层命名空间

下图显示了用于内存中表示 ZooKeeper 文件系统的树形结构。 ZooKeeper节点被称为znode。每个znode由一个名称识别，并通过路径（/）序列隔开。

* 在图中，首先有一个根znode，它由“/”分隔。在根下，有两个逻辑命名空间 config 和 workers。
* 在config命名空间用于集中配置管理以及 workers 命名空间用于命名。
* 在 config 命名空间下，每个znode可以存储高达 1MB 的数据。这类似于UNIX文件系统，不同的是父 znode 也可以存储数据。这种结构的主要目的是存储同步数据以及描述znode的元数据。这种结构被称为 ZooKeeper数据模型



在 ZooKeeper 数据模型中每个 znode 维护一个 stat 结构。 一个统计（stat ）只是提供了一个 znode 元数据。 它由版本号，动作控制列表（ACL），时间戳和数据长度组成。

* **版本号** − 每个znode都有一个版本号，这意味着每个相关的时间使用节点改变数据，其相应的版本号也将增加。使用版本号是重要的，在多个 zookeeper 的客户端正在努力通过相同znode执行操作。
* **动作控制列表（ACL）** −ACL是基本的身份验证机制，用于访问znode。它管理所有的znode读写操作。
* **时间戳**− 时间戳表示过去时间，从znode创建和修改起算。它通常以毫秒表示。ZooKeeper 确定每次从“事务ID”（zxid）更改znodes。Zxid是独特的，为每个事务处理维持时间，使您可以轻松地识别从一个请求到另一个请求经过的时间。
* **数据长度** − 存储在 znode 数据的合计量是数据长度。可以存储的最大数据容量为1MB。

## Znodes 类型

Znodes 被归类为持久性，顺序和短暂。

* **持久性znode** − 持久性 znode 处于活动状态，即使客户端，它创造了特定的 znode。默认情况下，所有的 znodes 是持久的，除非另有说明。
* **短暂znode** − 短暂znodes活跃，直到客户端还活着。当客户端被从 ZooKeeper 集合断开连接，然后znodes自动删除。由于这个原因，只有短暂znodes不允许再有一个子。如果短周期znode被删除，那么下一个合适的节点，将填补其位置。短暂znodes 发挥在领导选举中起重要作用。
* **连续znode**− 连续znodes可以是持久或短暂的。当一个新的znode作为连续znode创建的，则 ZooKeeper 通过将10位的序列号为原始名称设置znode的路径。例如，如果使用路径 /myapp 来创建一个znode作为连续znode，ZooKeeper将改变路径 /myapp0000000001并设置一个序列号为0000000002。如果两个连续znodes同时被创建，ZooKeeper从来不使用相同数量在每个znode上。连续znodes在锁定和同步中起到重要作用。