

Akka Cluster 深入浅出

(基于2.3.1版本)

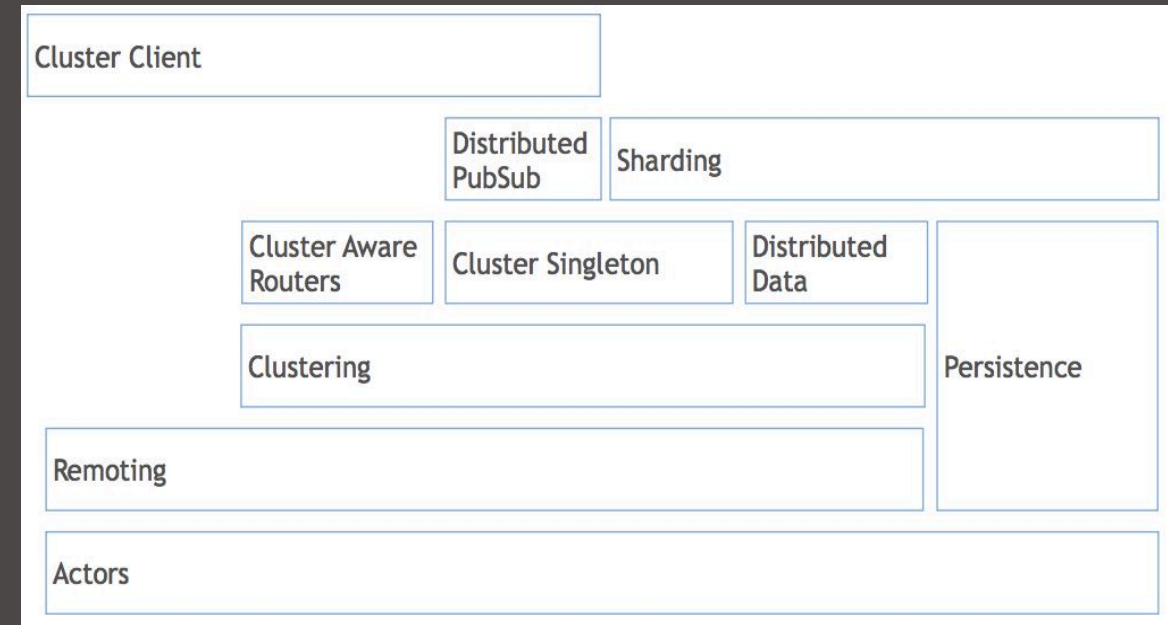
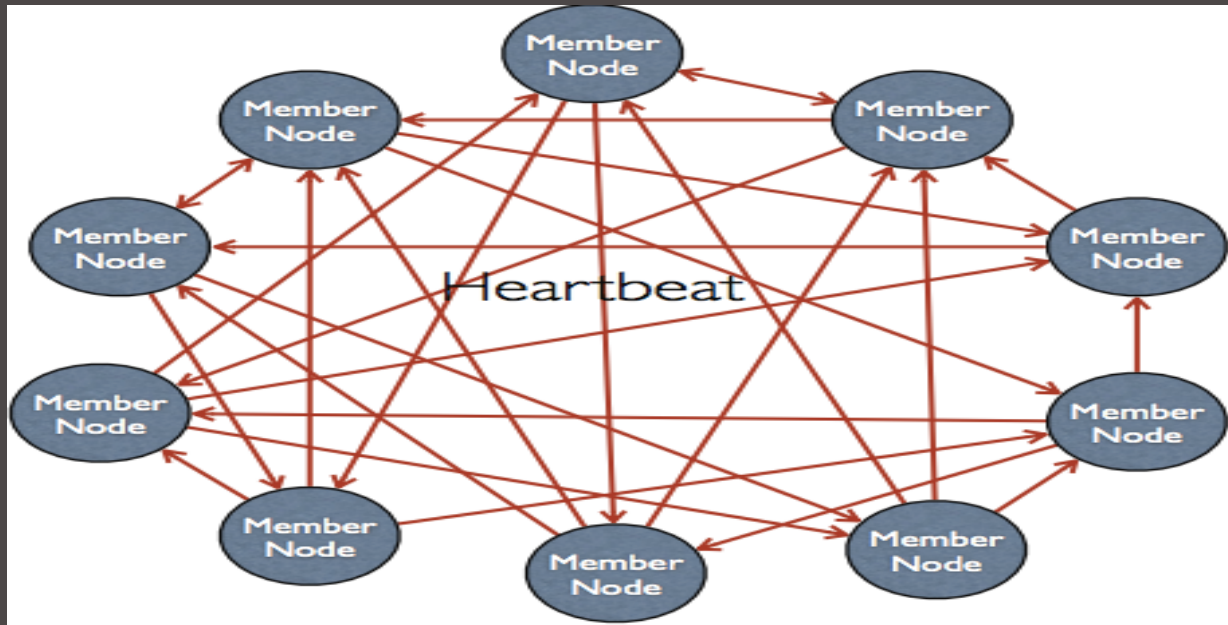
山行 资深软件开发工程师



Agenda

- Akka集群概述
- 集群启动
- Gossip协议的实现

Akka集群概述



- . Gossip-based Cluster Membership Manage
- . Cluster Functions: Sharding/Router ...
- . Failure Detector

...

集群启动_说明

Akka集群的启动首先是要启动一种叫做种子节点(SeedNode) 的节点们，它们启动可以分为以下几种情况：

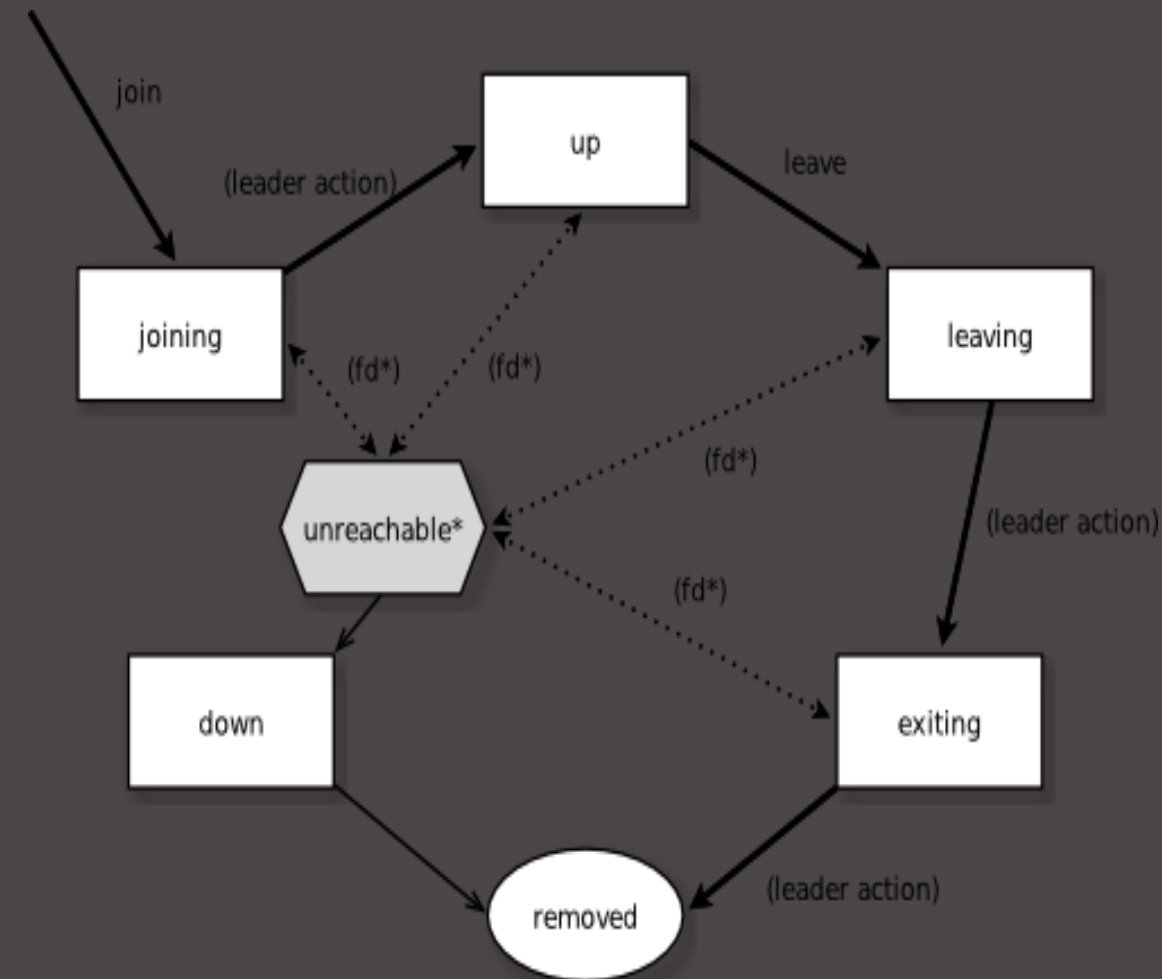
1. 某种子节点启动，它首先判断自己的ip是否在种子节点配置列表中，如果在并且是第一个，则它在一个规定时间内(默认是5秒)，向其他种子节点发送 'InitJoin' 消息，如果有确认消息返回，则加入第一个返回确认的种子节点所在的cluster中，否则，它自己将创建一个新的cluster。
2. 某种子节点启动，它首先判断自己的ip是否在种子节点配置中，但不是第一个，则它向其他种子节点发送消息，如果在一个规定时间内(默认是5秒)没有收到任何确认消息，则它将不断重试，直到有一个种子节点返回正确的确认消息，然后就加入这个种子节点所在的cluster中。

集群启动_实例

下面我们举例说明，有seed节点1、2、3：

1. seed2启动, 但是没有收到seed1 或seed3的确认。
2. seed3启动，没有收到seed1 的确认消息(seed2处在inactive状态)。
3. seed1 启动，创建cluster并加入到自己中。
4. seed2 重试加入过程，收到seed1的确认, 加入到seed1。
5. seed3 重试加入过程，先收到seed2的确认, 加入到seed2。

集群启动_状态变更



下面具体说明状态的变迁：

节点初始状态是joining。

通过gossip该信息被传递到所有节点上。如果gossip能够收敛，那么该节点的状态变成up。

用户可以人工调用leave、down命令，让节点状态发生改变。

由于Akka提供基于心跳的故障检测模块(failure detector)，所以当故障检测模块发现某个节点offline，则会把该节点置为unreachable*(它不是一种节点状态)，当该节点online后，则自动恢复到当前状态(如图所示：unreachable*和其他节点状态之间是用虚线表示，说明其并没有发生状态的变迁)。

从图中可以看出节点的最终状态都是removed。

从上图还可以看出一个节点如果状态变更路径是joining->up->leaving->exiting->removed，则它要经过4次gossip收敛(图中的leader action，图中漏了一个把removed传递给其他节点的gossip收敛)

Gossip协议的实现_两个重要概念

Leader

- 所有节点都可以成为Leader
- 所有节点都会公认同一个Leader
 - 在排序好的membership 环的第一个位置

Convergence

- 由Leader发起
- 当每个节点都'看到'相同的Vector Clock
- 当集群中出现'unreachable'的节点时候，就没法完成本次收敛
- 必须把这些'unreachable'节点置为状态'DOWN'才能继续本次收敛
 - 人工把问题节点下线
 - 集群配置为自动下线

自然界没有完全的自由，任何时候都有个老在罩着你

Gossip协议的实现_详细分析

<http://netcomm.iteye.com/blog/2080440>

广告时间



菜鸟网络
CAINIAO

打造全球顶尖仓内管理系统 ----- 菜鸟WMS

欢迎加入！

联系方式: wenzhi.bwz@alibaba-inc.com



Q/A
Thanks!

Early to bed, Early to rise