**Stable and Consistent Membership at Scale with Rapid**

（Rapid服务模式下稳定且一致的大规模集群成员的管理）

Lalith Suresh1, Dahlia Malkhi1, Parikshit Gopalan1, Ivan Porto Carreiro3, Zeeshan Lokhandwala2 1VMware Research Group, 2VMware, 3One Concern

<https://www.usenix.org/conference/atc18/presentation/suresh>

1. **发表会议**

论文《Stable and Consistent Membership at Scale with Rapid》发表于ATC18.ATC全称为USENIX Annual Technical Conference,USENIX是一个成立于1975年、由计算机操作系统用户、开发者和研究者所组成的机构，它不仅在创始之初就开始出版一本旨在关注UNIX系统及系统管理的杂志《;login:》，而且每年都会赞助若干个学术会议和工作室会议，其中ATC（USENIX年度技术会议）即是当中最负盛名的会议之一，它们已经成为介绍和讨论有关计算系统各方面发展的最先进信息的主要场所。ATC18于2018年7月11—13日在美国波斯顿举行。

1. **领域进展**

分布式计算技术的形成最早可追溯于1992年由OMG(Open Management Group) 组织提出的CORBA (Common Object Request Broker Architecture)，当时的的分布式应用环境均采用Client/Server架构，CORBA的应用在很大程度上提高了分布式应用软件的开发效率。分布式架构是分布式计算技术的具体应用和工具，目前成熟的[技术](https://baike.baidu.com/item/%E6%8A%80%E6%9C%AF/832247" \t "https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E6%9E%B6%E6%9E%84/_blank)有J2EE, CORBA和.NET(DCOM)，它自初始阶段将应用程序、数据库、文件等所有的资源都在一台服务器上的小型系统逐渐演进到现在使用应用服务器集群的超大型系统，因而对于大型集群中各成员结点的管理也成为进一步发展的焦点。现今的成员管理策略无外乎两种，一是通过辅助服务来管理集群，另一种则是基于流言的完全分布式管理。前者具有代表性的技术有Zookeeper,它是Google的Chubby的一个开源实现，起源于Hadoop且帮助Hadoop集群维护各种组件，它的历史悠久并被广泛应用于Yahoo!、Racksapce、Reddit、Facebook、eBay等公司的项目以及一些诸如Solr的开源企业级搜索系统中。此外还有作为后起之秀的etcd,它是采用Go语言编写的可靠的分布式KV存储产品，该开源项目由CoreOS公司；后者则主要被应用于Akka、Cassandra、Serf等部署系统中，在Amazon、Facebook、Google等企业的项目中也能看到它们的具体应用。

1. **主要挑战**

建立在网络环境中的大规模集群容易遭遇复杂且难以预测的网络故障，例如：非对称网络故障、联通中的flip-flops错误、大量包丢失等，因而在遭遇故障的情况下保证集群成员的稳定性与一致性是很有必要的。对于现有的3种集群成员管理策略而言，主要情况如下：

①逻辑集中式配置服务

原理：由集群中的少数结点掌握并维护成员列表的强一致性，其余结点通过定期查询成员列表来得到成员关系信息。

优点：设计逻辑简明。

缺点：对小集群的依赖降低了系统在遭遇故障后的恢复能力，即稳定性不佳。

②基于流言的成员关系

原理：各结点通过“流言”传播积极信息（即结点正常），当有结点出现故障时，经过一定的延迟后终将把该故障结点移除，改进后的策略将可以传播消极信息（即结点故障）。

优点：遭遇故障后的恢复能力较好，衰退缓慢。

缺点：弱一致性保证，复杂的紧急状况将导致各结点有关成员关系的观点出现分歧。

③组成员管理

原理：基于SMR(state-machine replication)的动态服务调度，仅在采样结点子集上运用观点更改的共识协议，即结构层次上的局部感知。

优点：强一致性保证。

综上所述，现有策略所能提供的稳定性与一致性往往呈现正交关系，即难以同时保证。

1. **改进策略——Rapid服务模式**

Rapid服务模式在组成员管理的思想之上进行了改进，它能同时应用于分布式与逻辑集中式设计中且效能同样良好。其主要过程如Figure 1所示。Rapid服务模式的要点有：

①**Expander-based monitoring edge overlay**

**在Rapid模式下，假定结点之间会不断地进行通信。对于该集群中的每个结点而言，都将有K个观察者与K个检测对象，此结点同时也是其自身检测对象的观察者。观察者通过通信是否联通来判定该结点是否出现异动，若出现异动，则观察者将对该集群中所有正常结点和可能连入的结点广播出去一个信号（若检测对象需要连入，则信号为JOIN；若检测对象出现故障，则信号为REMOVE）。该过程的背后还有稀疏矩阵的数学思想。**

**②Multi-process cut detection**

**根据4.①所述，接收到相应结点信号的结点将通过位图进行“投票”，此处设定一个低阈值L和一个高阈值H，关系是1≤L≤H≤K，≥H时为稳定状态，＜L为可忽略的噪声状态，在L和H之间时为不稳定状态。只有所有异动结点的相关投票都进入稳定状态且没有一个处于不稳定状态，这时该进行投票的结点算是达成了需要JOIN或REMOVE异动结点的认知。即almost-everywhere agreement.**

**③Practical consensus**

**在汇集了4.②过程中各进行投票的结点的结果后，需要在集群中达到普遍共识才能真正将JOIN或REMOVE应用到集群中，即consensus protocol，Rapid模式规定需要占该集群3/4的结点认知相同才进行应用。**

**通过实际测试可得知，相对现有策略，Rapid服务模式下结点数量超过2000个的大规模集群建立速度颇具优势，同时在遭遇网络故障的时候表现出优异的稳定性和一致性，收敛性极佳，端到端的一体化过程也更为容易。**

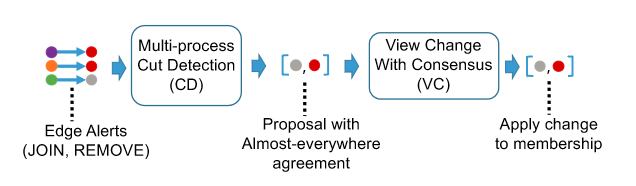


Figure 1: Solution overview, showing the sequence of steps at each process for

a single conﬁguration change.