Gossip协议

时间 2016-11-20 13:41:00 Edward Desire (/sites/RfQb6f)

原文
http://www.edwardesire.com/2016/11/20/the-intro-of-gossip-protocol/
(http://www.edwardesire.com/2016/11/20/the-intro-of-gossip-protocol/?utm_source=tuicool&utm_medium=referral)
主题 分布式系统 (/topics/11000146)
Gossip是一种去中心化、容错并保证最终一致性的协议。

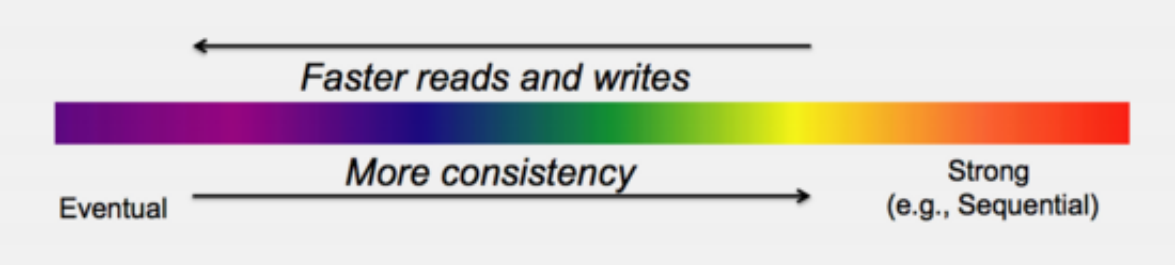
Background：分布式环境

Gossip是为了解决分布式遇到的问题而设计的。由于服务和数据分布在不同的机器上，节点之间的每次交互都伴随着网络延迟、网络故障等的性能问题。可见，分布式系统会比单机系统遇到更多的难题。

如CAP理论 所描述的，CAP三个因素在分布式的条件下只能满足两个。对于分布式系统来说，分区容忍性是其的基本要求。因为分布式系统的设计初衷就是利用集群多集的能力去处理单机无法解决的问题。分区容忍性（可扩展性）通过通过scale up和scale out实现的，也就是通过升级硬件或者增加机器来提升分布式系统的性能。这么说，可扩展性和可用性是相关联的。可扩展性好的系统，其可用性一般会比较高。所以分布式系统的所有问题基本都是在一致性和可用性之间进行协调和平衡。在工程实践中的经验如下：

一般来说，交易系统类的业务对一致性的要求比较高，一般会采用ACID模型来保证数据的强一致性，所以其可用性和扩展性就比较差。而其他大多数业务系统一般不需要保证强一致性，只要最终一致就可以了，它们一般采用BASE模型，用最终一致性的思想来设计分布式系统，从而使得系统可以达到很高的可用性和扩展性。

一致性可以通过信息在分布式环境下分发来保证，而分发的方式和速度则决定一致性的程度。从客户端的角度来讲：一致性包含三种状态：强一致性、弱一致性、最终一致性（弱一致性的特例）。在下图的一致性光谱中我们可以看出，弱一致性性是异步冗余，读写操作的响应更加快；而强一致性一般都是同步冗余的，所以伴随着性能的下降。



而最终一致性还有其他变种：因果一致性（有逻辑关系的操作能读到更新值）、读你所写一致性（Read-your-writes Consistency，A用户操作只保证自己的后续操作能读到更新值）、会话一致性（保证整个会话期间的读写一致性）、单调一致性（单用户的操作顺序一致）。

SWIM：最终一致性

前面提到Gossip解决的问题就是在分布式环境下信息高效分发的问题，这个问题的解决决定着系统的一致性程度。而Gossip协议是基于一种叫做SWIM的协议（ **S** calable **W** eakly-consistent **I** nfection-style Process Group **M** embership Protocol）。SWIM是一种无中心的分布式协议，各个节点之间通过p2p实现信息交流同步各节点状态的方法。看名字也知道这是一种弱一致性的实现。

SWIM协议给每个进程组成员在本地维护一个成员表，记录该组存活的进程。该协议通过失效检测器（Failure Detector）和传播组件（Dissemination Component）来完成工作。

SWIM的失效检测器会检测失效的节点并将失效节点的更新信息发送给传播组件。SWIM的传播组件通过多播（multicast）的形式将失效信息传播给组内的其他成员。

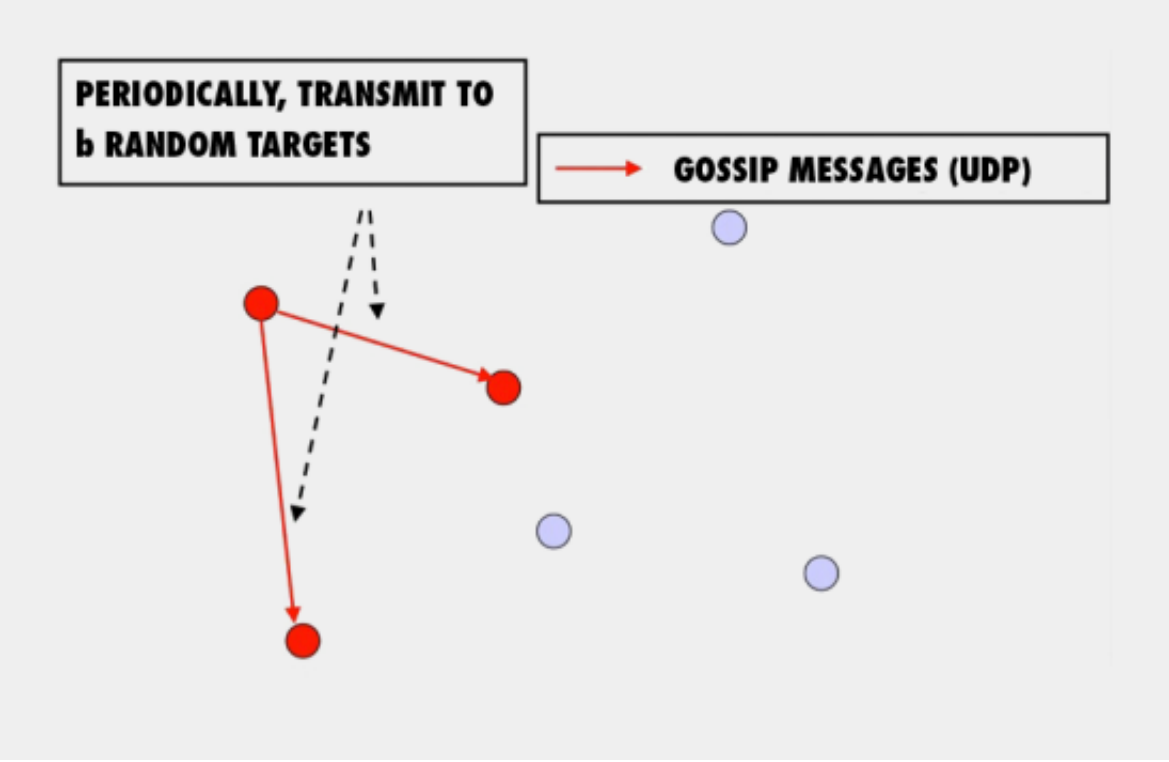
协议的可扩展性体现在：新成员的加入和退出也以同样的方式进行多播通信。而在基本的时间周期内进行失效检测能够保证在限定的时间范围内完成完备性检查，即每个失效的进程都能最终被检测到（最终一致性）。通过多播方式传输协议消的问题在于效率不好也不可靠，通过在ping和ack消息中捎带成员更新信息能够降低丢包率和减少传输时延。这种传播方式被称为可传导的方式（Infection-style）。

Gossip：办公室八卦

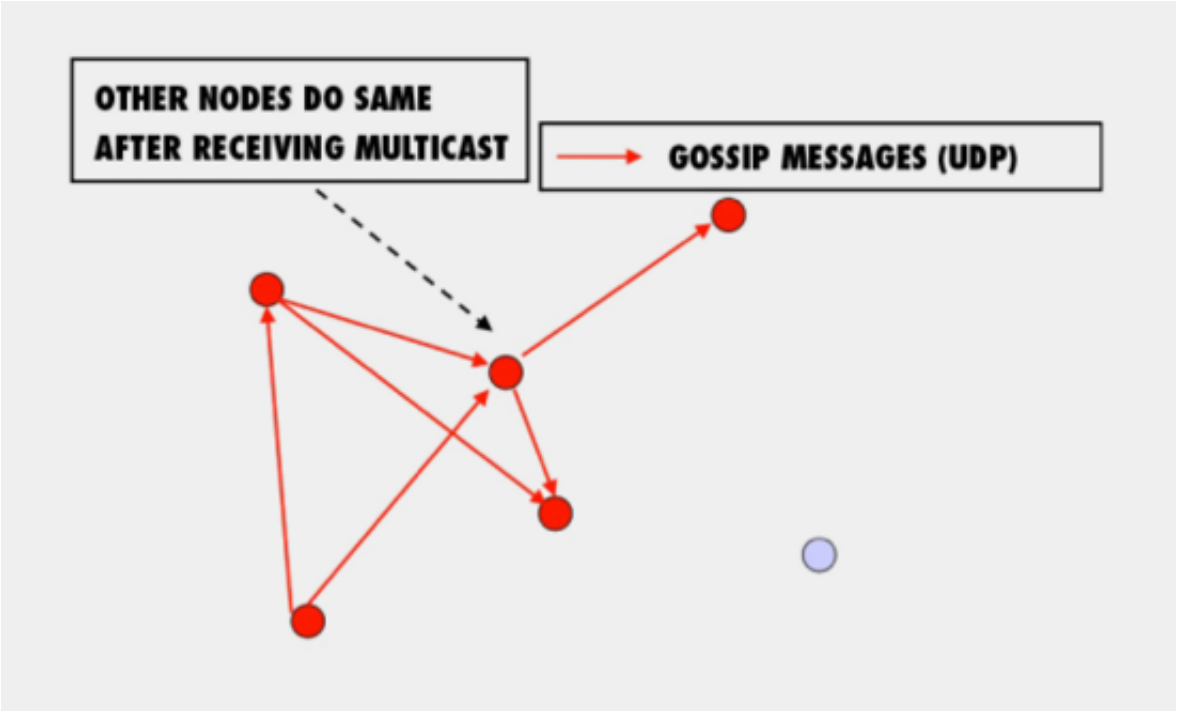
我们的办公室八卦一般都是从一次交谈开始，只要一个人八卦一下，在有限的时间内办公室的的人都会知道该八卦的信息，这种方式也与病毒传播类似。因此 Gossip也有“病毒感染算法”、“谣言传播算法”之称。

Gossip来源于流行病学的研究（括号里就是Gossip协议）：

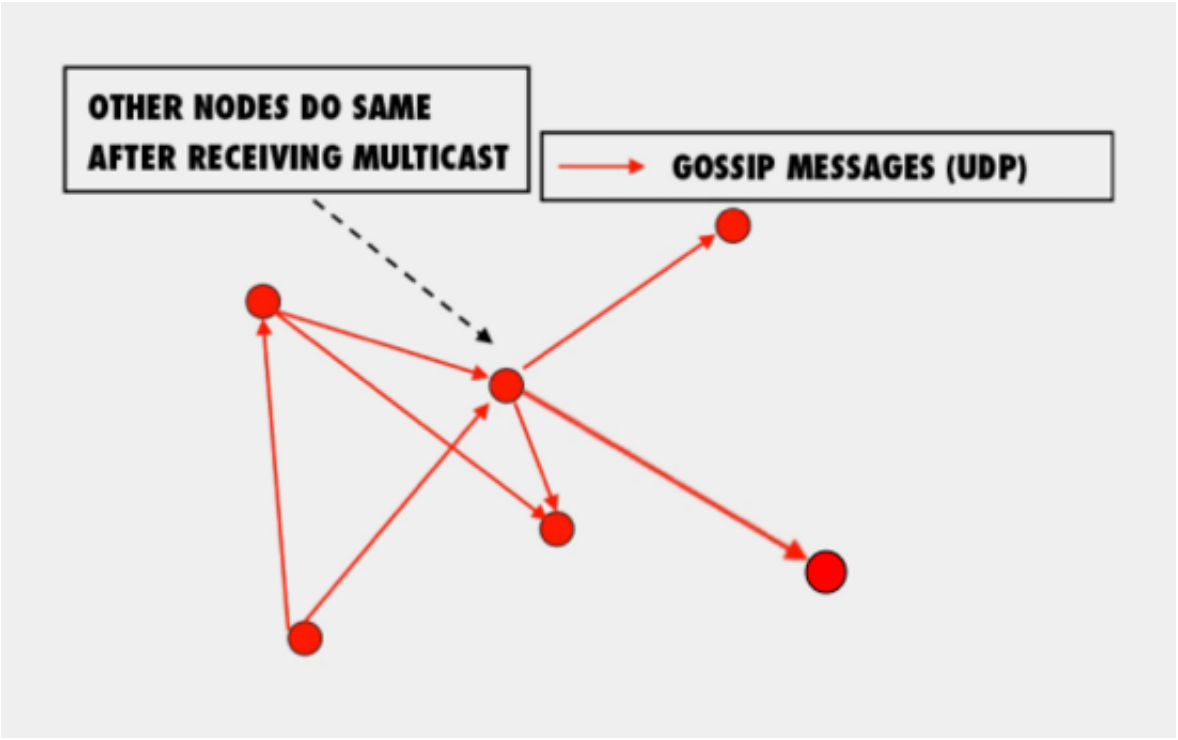
1. 在总数为n+1的人群中，被感染（infected）的人数初始化为1，并向周围传播。（一个节点状态发生变化，并向临近节点发送更新信息）



2. 在每个周期内总有未被感染（uninfected）的人转变成被感染的人，方式委每个被感染的人随机感染b个人。（对于节点状态变化的信息随机发送给b个节点，图例中的b值为2）



3. 经过足够的时间，所有的人都会被感染。（随着时间推移，信息能够传达到所有的节点，下一节会进行简单的证明）



可以看到，协议的核心内容就是节点通过将信息随机发送到b个节点来完成本次信息的传播，其涉及到周期性、配对、交互模式。Gossip的交互模式分为两种：Anti-entropy和Rumor mongering。

- Anti-entropy：每个节点周期性地随机选择其他节点，然后通过相互交换自己的所有数据来消除两者之间的差异。
- Rumor mongering：当一个节点有来新信息后，该节点变成活跃状态，并周期性地联系其他节点向其发送新信息。

每个节点维护一个自己的信息表 `<key, (value, version)>` ，即属性的值以及版本号；和一个记录其他节点的信息表 `<node, <key, (value, version)>>` 。每个节点和系统中的某个节点相互配对成为peer。而节点的信息交换方式主要有3种。

- Push：拥有状态新信息的节点随机选择联系节点并想起发送自己得到信息。

- Pull：发起信息交换的节点随机选择联系节点并从对方获取信息。
- Push-Pull混合模式：发起信息交换的节点向选择的节点发送信息。

上述Gossip为什么能够完成状态的同步呢？我们对其做一个简单的分析。

Analysis：收敛性证明

我们以上一节的Push模式Gossip协议进行分析。

在n+1个节点的系统中，每个节点每次随机向其他b个节点进行信息通信，即传播速率：$\beta = \frac{b}{n}$。未获得更新信息的数量为x（初始为n），获得更新信息的节点数为y（初始为1）。

在连续时间过程中，x的变化速率$\frac{dx}{dt} = -\beta xy$，即传播速率 * 乘以 * 两种类型节点之间可能传播的次数。可以推导出火的更新信息的节点数$y = \frac{n+1}{1+ne^{-\beta(n+1)t}}$

而总时间为 $t = c\log(n)$，即log(n)轮传播乘以一个常数时间。被感染的数量 $y \approx (n+1) - \frac{1}{n^{cb-2}}$

那么当c和b都是独立于n的很小的数值时。Gossip协议能够保证：

- 低延迟：在clog(n)内完成一次信息的更新。虽然不是常数级别的，但是气对数级别增长率在程序世界里是实践上可取的。
- 可靠性：$n+1-\frac{1}{n^{cb-2}}$会收到新信息。
- 轻量级：每个节点不会发送超过cblog(n)条信息。

这样我们不仅证明了Gossip的可靠性，并可以保证其在分布式系统应用的高可用性。注意的是，即使有的节点因宕机而重启或者有新节点加入，但经过一段时间后，这些节点的状态也会与其他节点达成一致。也就是说，Gossip天然具有分布式容错的优点。

Application：应用

除了改善SWIM协议中的多播方式，Gossip还在很多地方有应用：

- 数据库复制：基于Gossip实现分布数据管理的一般思路是：灾一个节点实现数据更新，通过Gossip算法将更新传播导其他节点。
- 聚合计算：在无中心的系统中，没有中心节点存储全局信息。通过Gossip应用导分布环境下的聚合计算中来保证系统的发送消息的容错。

总之，Gossip简单、高效，同时具有很好的可扩展性和鲁棒性，非常适合大规模、动态、资源受限的网络环境。

References：

1. 分布式系统常用思想和技术总结 **(http://blog.arganzheng.me/posts/thinking-in-distributed-systems.html)**
2. **SWIM**协议 **(http://prakhar.me/articles/swim/)**
3. **Gossip** 协议简介 **(http://kaiyuan.me/2015/07/08/Gossip/)**
4. 分布环境下的Gossip算法综述，《计算机科学》。

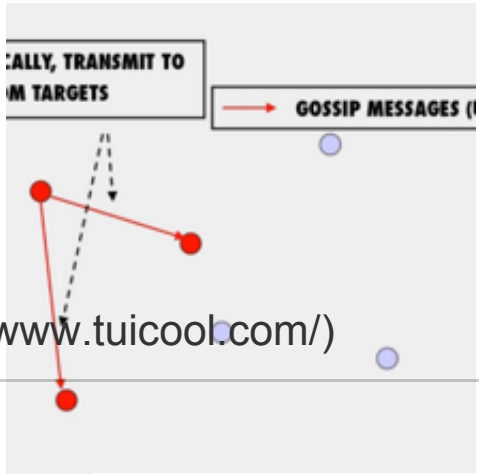
分享

☆ 收藏 ⚠ 纠错

推荐文章

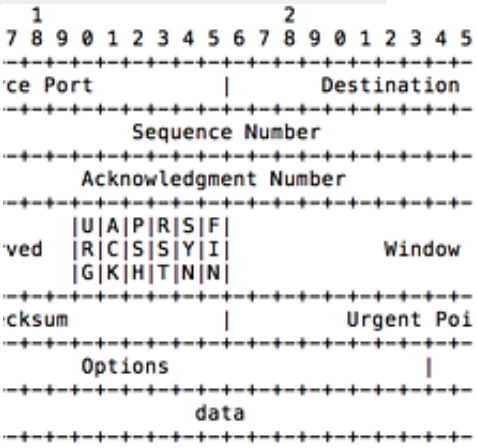
- 1. AI会是提升Twitter的王牌吗？不管如何，Jack Dorsey已经准备这么做.. (/articles/IzeMnaq)
- 2. 社群经济：一种以用户为中心的商业模式 (/articles/qqQNna3)
- 3. S7全球总决赛落户中国，对腾讯来说这是一场充满不确定性的豪赌 (/articles/RBjiUvr)
- 4. 金融大鳄索罗斯清仓NVIDIA股票，AI龙头真的见顶了吗？ (/articles/BzIl7jn)
- 5. 浅析BAT产品会员等级制度 (/articles/julzy2f)
- 6. 中国游客赴美签证出幺蛾子了？美国：你们需要提供社交媒体账号，唔.. (/articles/va6NzeY)

相关推刊



推酷 (http://www.tuicool.com/)

- by txie (/kans/3722883943) 《分布式》 (/kans/3722883943) *1*



- by 西瓜甜惨了 (/kans/2390983857) 《web》 (/kans/2390983857) *210*

- by 10Tbps  (/kans/2065589851) 《啊啊啊》 (/kans/2065589851) *8099*

我来评几句

请输入评论内容...

登录后评论

已发表评论数(0)

相关站点

Edward Desire (/sites/RfQb6f)

+ 订阅

热门文章

- 1. 炙手可热，那些优秀的 Go 存储开源项目和库 (/articles/vERnEfi)
- 2. SSD基本原理 (/articles/eY3MVzY)
- 3. 全球分布式数据库：Google Spanner（论文翻译） (/articles/mimaAfv)
- 4. 玩转 Ceph 的正确姿势 (/articles/FbQZRz3)