分布式系统中的 SWIM 成员协议 (/2015/02/765880.html)

07net01.com 发布于 2015-02-26 21:42:32 分类：IT技术 (/itjishu/) 阅读(181) 评论

我们假设你要建立一个类似于Cassandra的分布式数据库 (http://www.07net01.com/tags-数据库-0.html)。你所使用的存储 (http://www.07net01.com/storage_networking/)系统将在许多商业服务器 (http://www.07net01.com/tags-服务器-0.html)上存储和处理大量的数据。换句话来说，也就是你所使用的存储系统要管理好这些数据需要运行在100多个节点上。

以这样大的规模来运行，节点失效就是常态，而不是意外了。我们即便假设一个节点组成的集群可持续运行1000天（大约是3年时间），那么对于一个由500个节点的集群来说，每两天就会出现一个节点失效。

为了应对这种情形，你需要引进失效检测服务。失效检测服务除了能够检测到失效的节点外，它还能使所有正在运行的且未失效的进程保持同步。我们将在这篇博客 (http://www.07net01.com/tags-博客-0.html)文章 (http://www.07net01.com/2015/07/860262.html)里介绍一种名字为SWIM的失效检测协议，并说明其内部的运行机制。

SWIM

SWIM，或者称为可伸缩可传导的弱一致性进程组成员资格协议。它用来维护分布式系统中进程成员资格的协议。

> 成员资格协议让进程组中的每个进程在本地维护着一个该组未失效进程的列表，即成员列表。

因此，这个协议主要执行两个重要的操作：

- 检测失效节点，即如何识别出已经失效的进程
- 失效信息的广播，即如何通知整个系统中的其他进程哪些进程已经失效了。

那么毫无疑问的是：成员资格协议应该可伸缩型强，可靠性高，而且检测失效节点的速度快。成员资格协议的可伸缩型和执行效率主要由以下几个方面确定：

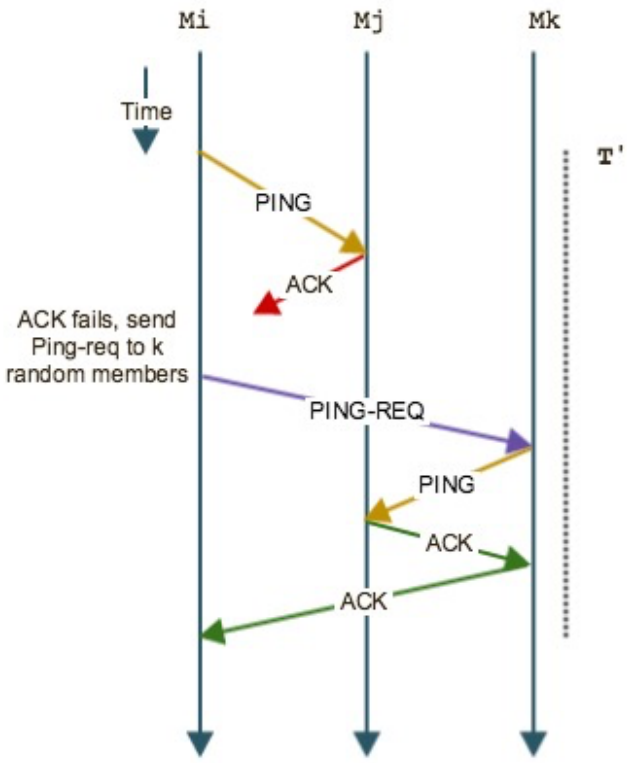
- 完备性：是不是每个失效的进程**最终**都能够检测到？
- 失效节点的检测速度：一个节点失效到它被非失效节点检测到的平均时间间隔是多长？

- 准确性：实际上进程未失效但却被认为是失效的频度（即误判率）是多少？
- 信息量：每个节点生成的网络 (http://www.wredian.com/tags-网络-0.html)通信的信息量有多大，它是否也是分布式的？

理想情况下，我们需要这样的协议：它一定要完全100%准确，这就意味着可以检测到每一个失效进程，而且不存在任何误判。然而，像分布式系统里的其他协议一样，存在这样的事实：在异步网络上保证100%的完备和准确是不可能的。因此许多成员资格协议（包括SWIM)为了完备性就会降低准确性，同时尽最大可能降低误判率。

SWIM失效检测器

SWIM失效检测器使用了两个参数-一个是协议的执行周期 T 和一个是执行失效检测的子进程组的大小，即整数 k 。



SWIM失效检测过程

上图显示了SWIM协议是如何运行的。每隔 T 时长后，进程M_i从自己的成员资格列表里随机选取一个进程，比如M_j,向这个进程发送ping。接着它就等待M_j的确认应答。如果在预先指定的超时时间内没有接收到确认应答，M_i就会随机的选择 k 个目标来间接地探测M_j,通过这 k 个目标发送ping给M_j,接下来，这 k 个目标中的每一个都会以M_i的名义给M_j发送ping，并等待接收告知M_i的确认应答。由于某些原因，如果这些进程都没有收到确认应答，那么M_i就会断定M_j失效，紧接着就会把更新消息发送给（下面即将讨论的）广播组件。

SWIM与其他心跳协议或者gossip协议的最大的不同点在于SWIM使用了其他目标来检测M_i是否在运行，这样可以规避M_i和M_j网络通路上出现拥塞的情形。

SWIM广播组件

广播组件做的工作仅仅是把失效进程的更新信息组播给该组中其余进程。接收到这个消息的所有成员都会从自己本地的成员资格列表中删除M_j。新成员加入或者成员自行离开的信息是以同样的方式进行组播的。

改进

以可传导的方式进行信息广播-在增强版的SWIM里，广播组件不会进行不可靠地且效率不高的信息组播，而是在失效检测协议发送ping和确认应答消息里附加上成员资格更新信息。这种方法被称为以可传导的方式进行广播（因为它与八卦消息或者传染病的传播方式相似），它可降低丢包率，减少传输时延。

猜测机制-虽然SWIM协议可以通过ping k个节点来确保不会在两节点的通信通道上出现拥塞，但是仍然无法避免这样一种情形，即运行非常良好的进程Mj可能（由于高负载而）运行的很慢，或者由于自身网络部分原因而临时不可到达，因此而被SWIM标记为失效。

SWIM在检测到有一个失效的进程时，通过运行一种称为猜测子协议的机制来减少这种情况的发生。在修改后的协议里，当M_i发现M_j无法响应（不管是直接不响应还是间接不响应），它都不会把M_j标记为已失效，而标记为可疑的。然后它通过广播组件（以传导的方式）发送M_j: `suspect` 消息给其他节点。如果接下来任何一个进程发现Mj对其ping做出了应答，那么它就可以去掉可疑标记，然后把M_j: `alive` 消息以可传导的方式发送给整个系统。

在限定的时间范围内进行完备性检查-基本的SWIM协议通常是在固定数量的协议运行周期内进行失效检测的。而要确保最终每个失效进程都能够检测到，就有可能遇到一种很少出现的情形：由于要随机选择目标节点，那么在给失效节点发送ping之前就可以出现非常大的时延。

为了减少这种情形的出现，建议SWIM做一个很小的改进：维护一个已知成员数组，以轮询的方式选择要进行ping的目标。在对这个数组里的成员从头到尾都ping过之后，这时对整个数组内各成员随意打乱次序，然后再进行下一轮处理。这样对同样目标的两次连续选择之间所用的时长就有了上限。

总结

SWIM协议已经用在许多分布式系统里。其中一款非常流行且使用SWIM的开放源代码 (http://www.07net01.com/tags-源代码-0.html)系统是Serf，它是由Hashicorp公司开发的一种集群成员资格的非集中式解决方案 (http://www.wredian.com/tags-解决方案-0.html)。其中文档部分对底层结构有一个非常清晰的说明。Hashicorp公司的友好人士还在Github上开放了源代码。如果你想通过阅读源代码而更加深入的了解的话，请下拉源代码。

最后要提一下，为了能够使高级的理念理解起来简单，这篇博客文章故意删除了一些数学计算方法。如果你想更加深入的了解，那么一定要读一下这篇论文，这样你才能更好地理解误判率、检测失效的平均时长和网络负载等的上限。

我希望这篇博客文章能让你对一种非常流行的成员资格协议如何运行的有一个概要的理解。如果你还有问题的，请在发表在下面的评论中。

关键词：

相关阅读

AMQP基本概念 (/2016/01/1172923.html)
实用模式之中介者模式 (/2016/01/1172919.html)
在 Swift 中编写 watchOS 2 Hello World 程序 (/2016/01/1172918.html)
如何恢复Windows8下IE的跳转列表功能 (/2016/01/1172912.html)
Installation Oracle11gR2 RAC---常见报错处理 (/2016/01/1172883.html)
Android中使用log4j (/2016/01/1172878.html)
Installation Oracle11gR2 RAC---创建数据库 (/2016/01/1172877.html)
Installation Oracle11gR2 RAC---安装database (/2016/01/1172874.html)

说点什么吧…

发布