Paxos Made Simple

摘要: 由简单英语表达的Paxos算法。

1. Introduction

被用来实现一个容错分布式系统的Paxos算法被认为太难去理解了，可能因为是原来的表述是由希腊语来表述的。实际上，这是其中最简单明了的分布式算法。下一节将展示这个公式算法如果满足我们期望的属性。最后一节解释了完整的Paxos算法，由构建分布式系统的状态机的共识应用获得——一个知名的途径，因为这是分布式系统理论中最常被引用到的主题。

2. The Consensus Algorithm

2.1 The Problem

假设有一个可以提出值的进程集合。一个共识算法确保被提出的值中的其中单独一个被选中。如果没有值被提出，则没有值被选中。如果一个值被选中，则进程们应该可以得知被选择的值。共识的安全要求是：

* 只有一个被提出的值可以被选择
* 只有一个值被选择
* 一个进度从不得知一个值被选中除非它确实被选中。

我们不会尝试去指定精确的活跃度要求。然而，该目标是为了确保有提出的值最终被选中，并且如果一个值被选中，则一个进程可以最终知道该值。

我们让算法中的这三种角色由三种代理类代表：proposers, acceptors, learners。在一种实现中，一个单独的进程可以表现为超过一种代理角色，但是代理至进程的映射不在考虑范围内。

假设代理们可以通过发送信息相互交流。我们使用传统异步，非拜占庭模型，也就是：

* 代理们以任意速率运行，可能因为停止而失败，可能重启。因为所有的代理在一个值被选中后失败然后重启，除非一些信息可以被一个失败并重启的代理记住，否则解决方案是行不通的。
* 消息可以通过任意时长被分发，可以重复，也可以丢失，但是不能错误。

2.2 Choosing a Value

最简单的选择一个值的方式是只有一个acceptor代理。一个proposer发送一个提案给acceptor，他将选择他收到的第一个提案的值。虽然简单，但是这个解决方案由于acceptor的失败可能导致后续的进程失败所以并不令人满意。

因此，让我们来尝试使用其他的方式来选择一个值。不同于只有一个acceptor，我们现在使用多个acceptor代理。一个proposer发送一个提案值给acceptor集合。一个acceptor可能accept（收到）提案值。当acceptor集合中的足够大部分人accept该值后，该值被选中。那么怎么算足够大呢？为了确保只有一个值被选中，我们可以使足够大的集合由代理中的多数组成。因为任意两个多数至少有一个共同的acceptor，所以如果一个acceptor只能accept最多一个值时，这是行得通的。（可以看到有很多的论文中都对多数如此概括）

在消息不丢失的情况下，我们希望一个值可以被选中即使只有一个proposer提出一个值。以下是要求：

P1. 一个acceptor必须accept他收到的第一个值。

但是该要求带来一个问题。一些值可以同时被不同的proposer提出，造成每个acceptor收到一个值的情况，但是没有一个值被acceptors中的多数所选中。即使只有两个值被提出，如果刚好分别被一半的acceptor选中，一个单独的acceptor的失败都会导致其无法得知哪个值被选中。

P1和一个值只有被acceptors中的多数选中才被选中的要求表明一个acceptor必须可以accept多个提案。我们通过给每一个提案分配一个自然数来持续跟踪一个acceptor可能accept的不同的提案，所以一个提案由一个提案序号和值组成。为了避免冲突，我们要求不同的提案有不同的序号。这依赖于其具体如何实现，所以目前我们只是假设如此。当一个提案被acceptors中的多数accept时一个值被选中。在这样的情景中，我们认为该提案（也就是该值）被选中。

我们可以允许多个提案被选中，但是必须保证所有被选中的提案有相同的值。通过提案序号，可以保证：

P2. 如果一个值为v的提案被选中，则所有被选中的更高序号的提案的值为v。

因为序号是完全排序的，条件P2保证了只有一个值被选中这个关键的安全属性。

为了被选中，一个提案必须被至少一个acceptor所accept。所以，我们可以使P2变为：

P2a. 如果一个值为v的提案被选中，则所有被任意acceptor选中的更高序号的提案值为v。

我们仍然维护P1以确保有提案被选中。因为沟通是异步的，所以一个提案可能被特定的从未收到过其他提案的acceptor c所选中。假设一个新proposer“苏醒”并且提出一个值不同的更高序号的提案。P1要求c去accept该提案，此时违反了P2a。为了同时维护P1和P2a，要求延伸P2a为：

P2b. 如果一个值为v的提案被选中，则所有由其他任意proposer提出的更高序号的提案值为v。

因为一个提案必须由一个proposer在其可以被一个acceptor accept之前提出，P2b实现P2a，也就是实现P2。

为了探索怎么满足P2b，让我们考虑我们怎么证明它拥有。我们假设一些序号为m且值为v的提案被选中并且任意序号n>m的提案的值也是v。我们通过使用n的induction来使证明更简单，所以我们可以证明序号为m值为v的提案在额外的假设之下，假设为所有被提出的序号为m..(n-1)的提案的值为v，其中i..j表示从i到j的序号集合。因为提案序号m被选中，必须有一些由多数的acceptors组成且所有acceptor都accept了它的集合C。结合之前的假设，m被选中的假设推导出：

所有C中的acceptor都accept了序号为m..(n-1)的提案，并且其中所有acceptor accept的序号为m..(n-1)的提案的值都是v。

因为任意由acceptors中的多数组成的集合S都包含了至少一个C中的成员，我们可以总结出一个序号为n的提案通过确保以下的不变条件保持值为v：

P2c. 对于任意v和n，如果一个值为v且序号为n的提案被提出，则有由acceptors中多数组成的集合S，(a) S中没有acceptor accept了序号小于n的提案，或者(b) v是更高序号的提案的值，其中所有的序号小于n的提案都被S中的acceptors所accept。

我们因此通过维护P2c来继续满足P2b。