**In Search of an Understandable Consensus Algorithm**

**摘要**

Raft是管理副本日志的共识算法。它提供了和Paxos相同的功能，并且与Paxos一样高效，但是它的结构与Paxos有一些不同；这使得Raft更容易理解也为构建实际系统提供了更好的基础。为了增强可理解性，Raft将共识的关键分为几块，分别是leader选举，log复制和安全性，并且实现了更强壮的一致性程度以减少需要考虑的情况。研究结果证明对于学生来说Raft比Paxos更容易学习。Raft同时包括一个新的机制去修改成员集群，使用重叠多数（Overlapping Majorities）以保证安全性。

**1 Introduction**

共识算法允许机器集群像一个团结的能够容忍部分成员失败的集体一样工作。正因为如此，他们在构建可靠的大规模软件系统中扮演了关键角色。Paxos[15,16]在过去的十年间主导了共识算法的讨论：很多的共识算法实现都基于Paxos或者受其影响，而且Paxos变成了教导学生共识算法的主要方式。

不幸的是，尽管有很多使Paxos更容易被理解的尝试 ，它还是有一点难以理解。此外，它的架构需要复杂的修改以支持实际的系统。因为如此，系统构建者和学生都不喜欢Paxos。

在与Paxos抗争之后，我们着手去探寻新的可以提供更好的系统构建和教育基础的共识算法。我们的方法不是一般的方法，因为我们的主要目标是可理解性：我们可以对实际系统定义一个共识算法并且用明显比Paxos更简单学习去描述它吗？此外，我们希望算法可以促进直观的对于系统构建者的必要的发展。对于算法如何工作，为什么这么做都是十分重要的。

该项工作的结果是一个叫做Raft的共识算法。在设计Raft过程中我们应用了一些特殊的技术以改变可理解性，包括分解（Raft分隔了leader选举，log复制和安全性）和状态空间减少（相对Paxos，Raft减少了非确定性等级和servers可以和其他不一致的方法）一个针对两所大学43个学生的研究显示Raft相对Paxos明显更好理解：同时学习了两种算法后，其中的33个学生对Raft的问题回答的比Paxos更好。

Raft在很多地方和一些共识算法相似（特别是，Oki和Liskov的Viewstamped Replication[29,22]），但是有一些新特性：

* Strong leader：Raft相对其他共识算法使用了更强的leadership构成。举例来说，log实体仅仅从leader到其他servers。这样简化了复制log的管理并且使得Raft更容易理解。
* Leader election：Raft使用随机定时器选举leader。该方式仅增加了一小部分其他算法早就有的心跳机制，可以使解决冲突更简单和更快。
* Membership changes：Raft对于更改server集群中集合的机制使用新共同共识（joint consensus）方式，这会使得在过渡时有两个不同配置的集合重叠。这样允许集群在配置修改时继续正常运作。

我们相信Raft比Paxos和其他共识算法优秀，不管是教育目的还是作为具体实现的基础。它比其他算法更简单，更易于理解；它的描述足够完整以满足实际系统的需求；它有很多的开源实现并且被很多公司使用；它的安全性属性被正式的指定和证明；它的效率也可以与其他算法相比。

论文的剩余部分介绍了复制状态机问题（第2节），讨论了Paxos的优劣势（第3节），描述了我们如何提高可理解性（第4节），现在的Raft共识算法（第5-8节），评价了Raft（第9节），并且讨论了相关工作（第10节）。

**2 Replicated state machines**