寻找一种易理解的共识算法
（扩展版本）

Diego Ongaro and John Ousterhout
Stanford University

Translated into Simplified Chinese from raft.github.io/raft.pdf
by xdsdmg@163.com

摘要

Raft 是一种管理复制日志的共识算法，它提供了和 (multi-)Paxos 等效的结果，且它和 Paxos 一样有效，但结构不同于 Paxos，这使得 Raft 更易理解，且能够为构建实际系统提供更好的基础。为了提高可理解性，Raft 将共识划分为 Leader 选举、日志复制及安全性等核心模块，并且它加强了一致性以减少需要考虑的状态。用户研究表明，对于学生而言，Raft 相交于 Paxos 更易理解。Raft 还提供了一种用于改变集群成员的新机制，其使用重叠大多数（overlapping majorities）[1]来保证安全性。

1 Raft 共识算法

你好呀 A paragraph of text goes here. Lots of text. Plenty of interesting text. Text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text text. More fascinating text. Features galore, plethora of promises.

1.1 Raft 基础

一个 Raft 集群含有若干个服务节点，通常为 5 个，系统能够容忍最多两个节点发生故障。在任何给定的时间，服务节点的状态为 leader、follower 及 candidate 的其中一个。正常运作时，只有一个 leader 节点，其他节点皆处于 follower 状态。

Raft 将时间分割为任意长度的 term，详见图 2。term 使用连续数字编号。每个 term 以选举开始，在选举中，一个或多个 candidate 节点尝试成为 leader 节点，详见章节 1.2。如果某个 candidate 节点赢得了选举，那么它将在此 term 的剩余时间内担任 leader 节点。在一些情况下，选举会导致投票分裂，term 将以没有 leader 节点的状态结束，然后一个新 term（新选举）将快速开始。Raft 保证在一个 term 中至多只有一个 leader 节点。

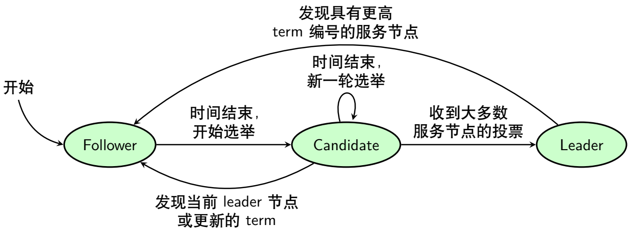

图 1：服务节点状态。follower 只应答来自其他服务节点的请求。如果一个 follower 未收到任何通信，它将成为 candidate 并开始选举。收到来自整个集群大多数服务节点投票的 candidate 将成为新的 leader 节点。leader 节点通常会一直运作至失败为止。

不同的服务节点可能在不同的时刻观察到 term 的变化。term 在 Raft 中作为逻辑时钟 [2]，且服务节点能够根据 term 检测过期信息，比如过期的 leader 节点。每个服务节点存储当前的 term 编号，term 编号随时间单调递增。服务节点在相互通信时交换彼此当前的 term 信息，如果一个服务节点的 term 小于其他节点的，它会将自己的 term 更新为较大的值。如果 candidate 或 leader 节点发现自己的 term 已过期，它将立即恢复至 follower 状态。如果服务节点收到携带过期 term 的请求，它将拒绝此请求。

1.2 Leader 选举

2 Footnotes, Verbatim, and Citations

Footnotes should be places after punctuation characters, without any spaces between said characters and footnotes, like so.[2] And some embedded literal code may look as follows.

```
int main(int argc, char *argv[])
```

---

[1] 这里翻译得不准确，需要调整

[2] Remember that USENIX format stopped using endnotes and is now using regular footnotes.

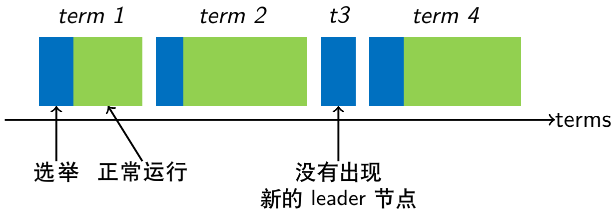

图 2: 时间被分割为 term，并且每个 term 以选举开始。选举成功后，一个 leader 节点将管理集群直至 term 结束，选举也可能失败，在此情况下，term 将以没有选举出 leader 节点结束。不同的节点可能在不同的时刻观察到 term 的变化。

```
{
    return 0;
}
```

Now we're going to cite somebody. Watch for the cite tag. Here it comes. Arpachi-Dusseau and Arpachi-Dusseau co-authored an excellent OS book, which is also really funny [1], and Waldspurger got into the SIGOPS hall-of-fame due to his seminal paper about resource management in the ESX hypervisor [3].

The tilde character (~) in the tex source means a non-breaking space. This way, your reference will always be attached to the word that preceded it, instead of going to the next line.

And the 'cite' package sorts your citations by their numerical order of the corresponding references at the end of the paper, ridding you from the need to notice that, e.g, "Waldspurger" appears after "Arpachi-Dusseau" when sorting references alphabetically [1,3].

It'd be nice and thoughtful of you to include a suitable link in each and every bibtex entry that you use in your submission, to allow reviewers (and other readers) to easily get to the cited work, as is done in all entries found in the References section of this document.

Now we're going take a look at Section 3, but not before observing that refs to sections and citations and such are colored and clickable in the PDF because of the packages we've included.

3 Floating Figures and Lists

Here's a typical reference to a floating figure: Figure 3. Floats should usually be placed where latex wants then. Figure3 is centered, and has a caption that instructs you to make sure that the size of the text within the figures that you use is as big as (or bigger than) the size of the text in the caption of the figures. Please do. Really.

In our case, we've explicitly drawn the figure inlined in latex, to allow this tex file to cleanly compile.

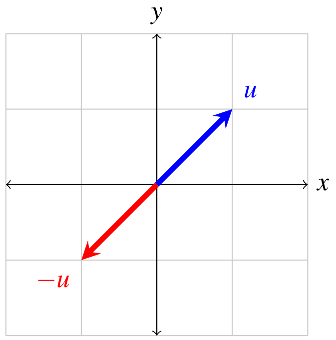

图 3: Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text. Text size inside figure should be as big as caption's text.

But usually, your figures will reside in some file.pdf, and you'd include them in your document with, say, \includegraphics.

Lists are sometimes quite handy. If you want to itemize things, feel free:

fread a function that reads from a `stream` into the array `ptr` at most `nobj` objects of size `size`, returning returns the number of objects read.

Fred a person's name, e.g., there once was a dude named Fred who separated usenix.sty from this file to allow for easy inclusion.

The noindent at the start of this paragraph in its tex version makes it clear that it's a continuation of the preceding paragraph, as opposed to a new paragraph in its own right.

3.1 LaTeX-ing Your TeX File

People often use `pdflatex` these days for creating pdfs from tex files via the shell. And `bibtex`, of course. Works for us.

Acknowledgments

The USENIX latex style is old and very tired, which is why there's no \acks command for you to use when acknowledging. Sorry.

Availability

USENIX program committees give extra points to submissions that are backed by artifacts that are publicly available. If you made your code or data available, it's worth mentioning this fact in a dedicated section.

References

[1] Remzi H. Arpaci-Dusseau and Arpaci-Dusseau Andrea C. *Operating Systems: Three Easy Pieces*. Arpaci-Dusseau Books, LLC, 1.00 edition, 2015. `http://pages.cs.wisc.edu/~remzi/OSTEP/`.

[2] Leslie Lamport. *Time, clocks, and the ordering of events in a distributed system*, page 179–196. Association for Computing Machinery, New York, NY, USA, 2019.

[3] Carl A. Waldspurger. Memory resource management in VMware ESX server. In *USENIX Symposium on Operating System Design and Implementation (OSDI)*, pages 181–194, 2002. `https://www.usenix.org/legacy/event/osdi02/tech/waldspurger/waldspurger.pdf`.