

## RAFT

Com os avanços tecnológicos de virtualização e sistemas distribuídos, alguns problemas surgiram e que precisam ser resolvidos, e um destes problemas é o consenso entre todos os servidores replicados de que um conjunto de dados deve ser armazenado, atualizado ou excluído.

Servidores são suscetíveis a falhas, devido à inúmeros motivos causados por software, hardware ou redes. Essas falhas podem ser tão imprevisíveis quanto inevitáveis. Quando ocorre uma falha e seu serviço fica fora do ar, não apenas levará algum tempo para ser alertado sobre essa falha, mas também levará ainda mais tempo para resolver essa falha, ocasionando perda de clientes e receita ou danos graves à reputação da marca ou serviço. Uma maneira de reduzir o impacto de uma falha é iniciar mais instâncias do serviço, e é neste momento que o protocolo ou consenso RAFT, que foi proposto em um artigo de 2014, por Diego Ongaro e John Ousterhout, é utilizado. Por exemplo, se executarmos várias instâncias do nosso serviço em muitas regiões diferentes ao redor do mundo, mesmo que uma falhe e fique offline, ainda haverá outras em funcionamento, mantendo o serviço funcionando sem interrupção, criando um sistema distribuído.

O RAFT coordena todos os nós dentro de um sistema distribuído para chegarem a um acordo ou alcançarem consenso sobre o conjunto de dados ou estado dos sistemas. Cada nó pode se tornar um líder, um candidato à líder e um seguidor. Somente o nó líder poder interagir com os clientes, todos os outros nós interagem com o líder, os seguidores no caso. No sistema distribuído, somente há um líder, porém por um curto período, o sistema pode ficar sem nenhum líder, devido a uma falha. Neste momento o papel do candidato entra em evidência, podendo se tornar um líder.

O RAFT consiste em 2 principais etapas, eleição de líder único e replicação de log.

- Eleição de líder único:

Nesta fase, inicia-se um processo de votação para selecionar um nó líder entre os nós participantes no sistema distribuído. Esse processo começa quando um nó é inicializado ou quando o nó líder atual se torna indisponível, incluindo latência e/ou timeout.

Cada nó no sistema mantém um número de termo\* atual, aumentando sempre que um novo líder é eleito. Quando um nó é inicializado, ou um líder se torna indisponível, o Raft inicia uma nova eleição incrementando seu número de termo e enviando RPCs de RequestVote para os outros nós no sistema.

O RPC\* de RequestVote inclui o número de termo do nó, seu índice e termo de log atual, e a identidade do nó candidato. Os outros nós no sistema votarão no candidato se as seguintes condições forem atendidas:

- O número de termo do candidato é maior ou igual ao número de termo atual do nó.
- O log do candidato está atualizado, contendo todas as entradas de log do log atual do nó até certo índice.
- Se o nó votar no candidato, ele também atualizará seu número de termo atual para corresponder ao seu número de termo do sistema distribuído.

\*Número de Termo: como um relógio lógico, que indica que todos estão na mesma “hora”.

\*RPC: Remote Procedure Call – Protocolo de comunicação entre sistemas remotos.

Uma vez que o candidato tenha recebido votos da maioria dos nós no sistema, ele se torna o líder e inicia a fase de replicação de log. Porém um problema pode aparecer nesta fase, visto que o tempo de resposta entre os servidores pode ser afetado pela latência de rede, então o Raft utiliza um tempo aleatório para o fim da votação, entre 150 e 300 ms, caso nenhum líder seja eleito, inicia-se uma nova eleição, repetindo todo o processo.

- Replicação de Log

Para garantir consistência e confiabilidade, o algoritmo Raft utiliza um log replicado para armazenar atualizações do estado do sistema distribuído. Cada nó no sistema mantém sua cópia do log, que é atualizada através da replicação de log, que funciona enviando entradas de log do nó líder para os outros nós no sistema.

Cada entrada de log consiste em um comando que é aplicado ao estado compartilhado do sistema, juntamente com um número de termo que identifica o líder que emitiu o comando. O nó líder é responsável por receber novas entradas de log dos clientes, anexá-las ao seu log e replicá-las para os outros nós no sistema. Os outros nós no sistema são responsáveis por armazenar as entradas de log replicadas e aplicá-las às suas cópias do log.

O Raft é mais simples de entender e implementar que outros protocolos, como o PAXOS na rede blockchain. Além disto, o Raft oferece uma série de outros benefícios como convergência mais rápida e tolerância a falhas.