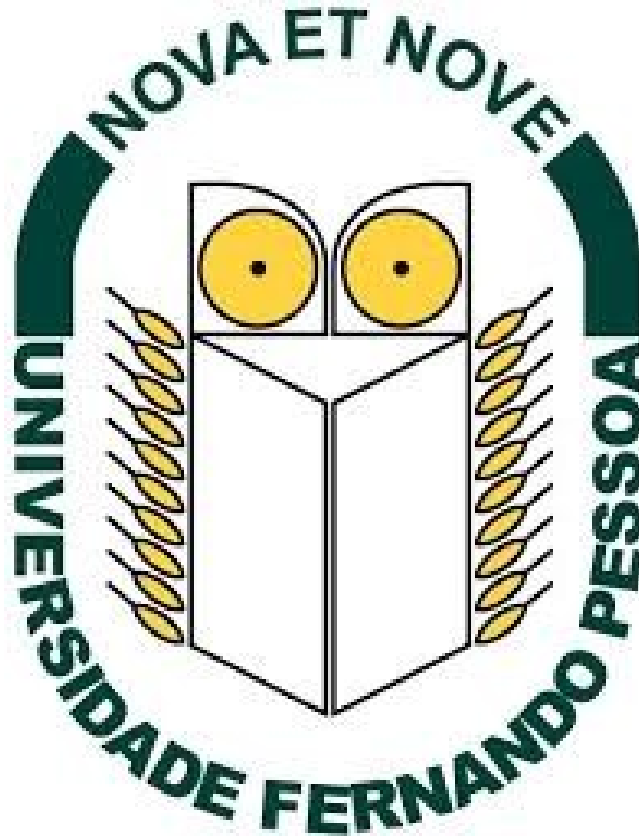


Universidade Fernando Pessoa

Curso de Engenharia Informática



Protocolo RAFT

Disciplina: Sistemas Distribuídos

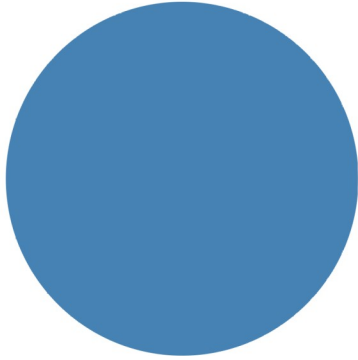
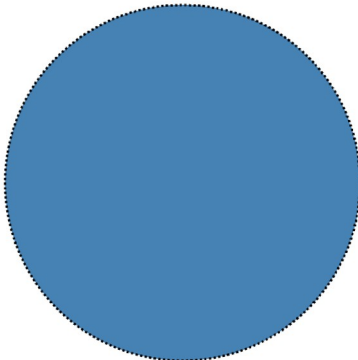
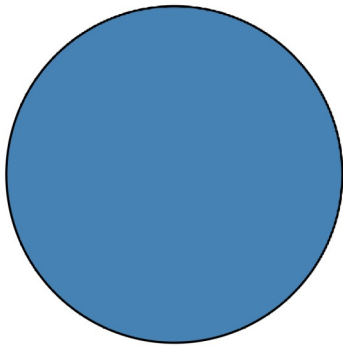
Realizado por: Luís Aguiar (36452)

O que é?

É um protocolo que ao ser implementado garante “distributed consensus” (consenso entre todos os valores das diversas bases de dados do servidor)

Como funciona?

Um nó (base de dados) pode estar em um de três estados:

Follower	Candidate	Leader
		

Leader Election (Processo de eleição do leader dos nós)

Overview do processo:

1. Todos os nós são iniciados no Follower state.
2. Se os Followers não receberem informações enviadas pelo Leader eles podem tornar-se Candidates.
3. O Candidate pede aos outros nós para votarem.
4. Os outros nós respondem ao Candidate com o seu voto.
5. O Candidate torna-se Leader se receber votos positivos da maioria dos outros nós.

Após esta eleição todas as alterações no sistema são obrigadas a passar pelo leader.

Detalhes de funcionamento interno:

Neste protocolo existem duas definições de timeout que controlam as eleições:

Election timeout → Tempo que um Follower espera até se tornar um Candidate (aleatório entre 150ms e 300ms).

Heartbeat timeout → Intervalo entre mensagens de Append Entries.

1. Depois do election timeout o Follower torna-se num Candidate e começa o novo election term (prazo de eleição).
2. Candidate vota em si próprio.
3. Envia mensagens de request vote para os outros nós.
4. Se o nó recetor ainda não tiver votado neste election term então ele vota no Candidate e dá reset ao election timeout.
5. Assim que um Candidate tem a maioria dos votos torna-se o nó Leader.
6. O Leader então começa a enviar mensagens de Append Entries para os seus Followers.
7. Os Followers depois respondem a cada mensagem de Append Entries.
8. Este election term continua até um dos Followers deixar de receber mensagens de presença (Append Entries) do Leader e tornar-se num Candidate.
9. Se o nó Leader parar o election term é repetido.

Apesar de pouco provável, **dois nós podem iniciar um processo de voto (election term) em simultâneo e obter o mesmo número de votos**. Caso isto aconteça recomeça-se o processo de Leader Election, processo este que se pode repetir até ser possível definir um Leader sem empates.

Log Replication (gestão de uma nova alteração):

Overview do processo:

1. A nova alteração é adicionada como uma entrada no log do nó Leader .
2. Esta entrada de log fica no estado uncommitted, desta forma não dá update ao valor do nó.
3. Antes de dar commit à entrada o nó primeiro replica-a para os nós Followers.
4. O Leader espera até a maioria dos nós terem escrito a entrada.
5. A entrada dá commit no nó Leader e este altera o seu estado.
6. O Leader notifica os Followers que a entrada foi committed.

Detalhes de funcionamento interno:

Assim que é eleito um Leader precisamos de replicar todas as alterações deste para todos os nós. Este processo é realizado utilizando as mensagens de Append Entries que usamos no Leader Election.

1. Um cliente envia uma alteração para o Leader.
2. A alteração é adicionada ao log deste.
3. A alteração é enviada para os Followers na próxima mensagem de Append Entries.

4. Esta alteração é confirmada (committed) assim que a maioria dos Followers a realizarem.
5. Uma resposta é enviada para o cliente.

No caso de se fazer uma partição nos nós, cada uma das partições fica com um Leader, mas caso se voltem a unir, o Leader eleito é o que tiver mais votos de todas as partições e os valores uncommitted são corrigidos.