

Computernetzwerke CE26

Raft Consensus Algorithmus

Der Raft Consensus Algorithmus wurde von Diego Ongaro und John Ousterhout entwickelt. Vor dem Raft Algorithmus war der Paxos Algorithmus in der Lehre weit verbreitet. Dieser ist jedoch schwerer zu verstehen. Deswegen haben Diego und John den Raft Algorithmus entwickelt um Studenten und anderen Menschen einen einfacheren Consensus Algorithmus zur Verfügung zu stellen.

Um den Raft Algorithmus besser verstehen zu können gehen wir alles einmal durch. Wir haben eine Gruppe von Servern z.B. 5 Server. Jeder dieser Server kann unter bestimmten Umständen einen von drei Zuständen annehmen. Zu diesen Zuständen zählen der Leader-State, der Follower-State und der Candidate-State. Jeder Server beginnt als Follower. Sobald die electionTimeout Zeit eines Followers abgelaufen ist, wechselt dieser in den Candidate-State und versucht mithilfe von RPC's Stimmen von den anderen Servern zu bekommen. Wenn der Candidate die Mehrheit der Server erreicht und eine Antwort auf seinen RPC erhalten hat wird dieser der neue Leader. Der neue Leader versucht mittel RPC's eine Neuwahl zu unterbinden.

Ab diesem Zeitpunkt agieren die anderen Server passiv und reagieren nur auf den Leader. Sollten sie von einem Client eine Anfrage erhalten geben sie diese an den Leader weiter. Der Leader wird jetzt für eine Zeitperiode (Term) als Schnittstelle zwischen der Servergruppe und dem Clienten dienen. Der Leader nimmt die Anfragen des Clienten an und versucht diese in den Protokollen der Follower zu replizieren. Dazu sendet er RPC's an die Follower bis er eine Antwort der Follower erhält. Wenn die Mehrheit der Follower geantwortet hat wird die Anfrage im Protokoll des Leader gespeichert (committed) und er sendet nochmals einen RPC an die Follower damit diese es ihm gleich tun. Danach kann die Anfrage des Clienten in der State Machine des Leader und der Follower bearbeitet werden und der Leader gibt dem Clienten nach der Bearbeitung eine Antwort auf seine Anfrage.

Ein Leader versucht in seiner Amtszeit die Logs seiner Follower identisch mit seinem Log zu halten. Dazu überprüft er mit jedem RPC's die Log-Einträge und sollten Unstimmigkeiten erkannt worden sein zwingt der Leader, den Follower sein Log zu aktualisieren. Dabei sucht der Leader den letzten Eintrag in seinem Log, welcher mit dem des Followers übereinstimmt und löscht alle Folge Einträge im Log des Followers damit er seine Log-Einträge im Follower speichern kann.

Ist der Term, also die Zeitperiode des Leaders, vorbei so gibt es für den näch-

sten Term eine neue Wahl. Dabei ist zu beachten, dass der neue Leader alle Log-Einträge aus den vorhergegangenen Term besitzt. Das wir bei der Wahl überprüft, denn durch das Sammeln der Stimmen werden die Logs der Server verglichen und so muss jeder Eintrag im Log des Candidate auch in mindestens einem Log der Follower sein. Sollte dies nicht der Fall sein kann dieser Server kein Leader werden.