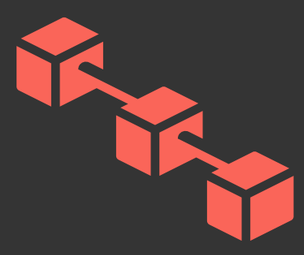
Amélioration de protocole de Blockchain

Implémenter une réparation et amélioration du consensus Abraxas, qui a été présenté récemment à la meilleure conférence de sécurité

Nathan Rouillé - Titouan Duhazé - Adam Chgour - Rayane Dakhlaoui - Aymane Hamdaoui
Encadrant: M. Rambaud

Qu'est-ce qu'un consensus de Blockchain ?

- Une blockchain est une suite de blocs chaînés qui contiennent des transactions.
- Un consensus est un algorithme décentralisé exécuté par chacun des nœuds pour assurer la sécurité de la Blockchain.



Safety de Jolteon

Nous avons dû comprendre le consensus Jolteon puis, à partir de sa description, prouver sa sécurité, c'est-à-dire prouver qu'une transaction validée est immuable !

Scan moi pour la preuve !

Protocol d'états stable pour le réplicas i

Propose

En entrant dans le round r, L_r multicasts un block $B = [id, qc_{high}, tc, r, v_{cur}(= 0), txn]$ avec $tc = tc_{r-1}$ si L_r entre dans le round r en recevant dans le round-$(r-1)tc_{r-1}$ et $tc = \perp$ sinon.

Vote

Chaque replica reçoit le block B.
Si $r = r_{cur}$ (empêche de voter pour un round précédent), $r > r_{vote}$ (un validateur ne revote pas 2 fois le même round) et ((1) $r = qc.r + 1$ ou (2) $r = tc.r + 1$ et $qc.r \geq \max\{qc_{high}.r \mid qc_{high} \in tc\}$) : les réplicas envoient leur threshold signature $\{id, r, v\}$ à L_{r+1} et $r_{vote} \leftarrow r$ (vérification des txn implicite car triviale avec les signatures)

Lock (1-chain lock rule)

Quand on reçoit un qc valide (formé de votes ou contenu dans une proposition ou time-outs) : $qc_{high} \leftarrow \max(qc_{high}, qc)$

Commit (2-chain commit rule)

S'il existe deux blocks certifiés adjacents dans la chaîne avec $B'.r = B.r + 1$ alors: les replicas commit B et tous ces ancêtres

Protocole de Pacemaker pour le réplicas i

(Avance de round soit par manque de progression soit parce que le round est complet)

Avancer de round

Si le replica reçoit ou forme un qc ou un tc pour le round $r - 1$ alors $r_{cur} \leftarrow \max(r_{cur}, r)$

Timer et Timeout

- Quand on entre dans le round r on envoie le TC du round $r - 1$ (si on en a un) à L_r puis on reset le timer à 0.
- Quand le timer atteint un temps τ, on arrête de voter pour le round r_{cur} et on multicast un message signé contenant notre threshold signature du round r_{cur} et notre $qc_{high}\langle r_{curi}, qc_{high}\rangle$
- Quand on reçoit un message de timeout valide ou un TC on exécute Advance round, Lock puis Commit.
- Quand on reçoit $2f+1$ messages de timeout on forme un TC (et on l'envoie à L_{r+1}) puis on fait Advance round.

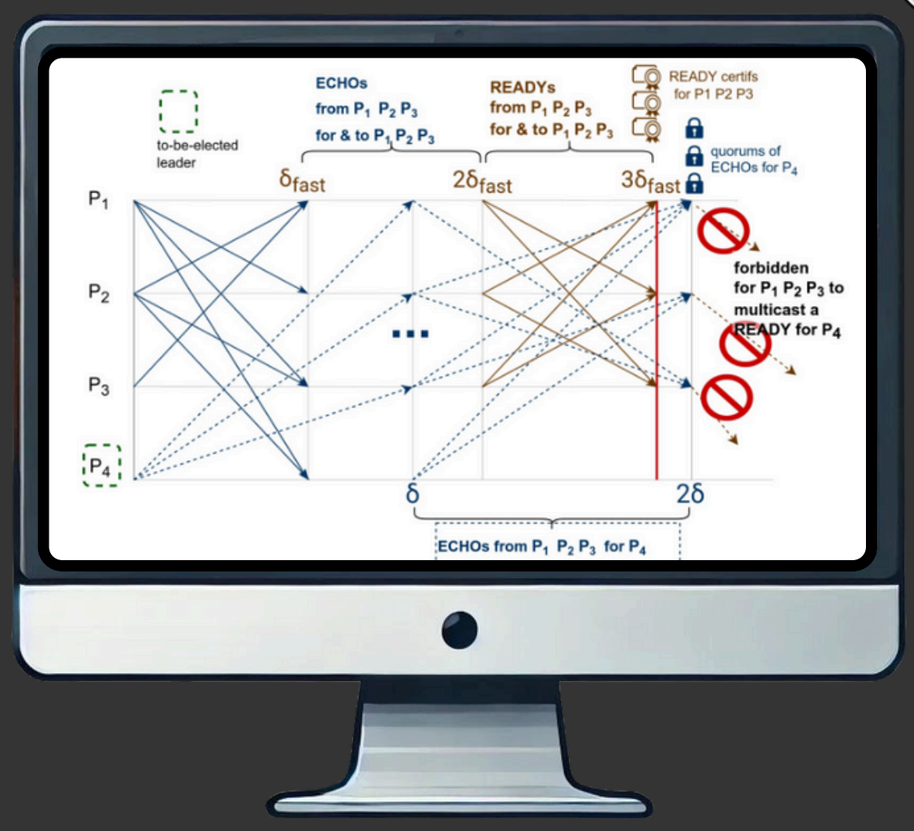
Qu'est-ce qu'Abraxas ?

C'est un consensus qui en combine 2 autres :

- Jolteon utilisé lorsque la blockchain évolue normalement
- 2-Phase-VABA lorsque celle-ci est attaquée

Implémentation de 2PAC

2PAC est le consensus asynchrone crée par M. Rambaud suite à sa découverte d'une faille dans la safety de 2-Phase-VABA. Nous avons du comprendre le consensus et l'implémenter en Rust.



Test de performances

Test en collaboration avec les autres groupes. Notre implémentation leur permet de tester les performances de ce consensus