BFT

Grok

Byzantine Fault Tolerance、ビザンチン障害耐性
具体的なコンセンサスアルゴリズムではなく概念。

概要

ノードの一部が期待しない動作(悪意があったり故障していたりなどのビザンチン障害)をしてもコンセンサスを形成してシステム全体の整合性と信頼性を保つ。
ノードの約3分の1がビザンチン障害を持つ。障害を持つノード数を `f` とした場合、全体のノード数が `3f + 1` 以上であれば保証できる。

基本的なしくみ

1. リクエストの送信
    クライアントがリクエストをシステムに展開
        トランザクションの展開やブロックの承認などか
2. 合意形成プロセス
    ノード間でメッセージを交換し、リクエスト内容の正当性を検証
    各ノードが他のノードのメッセージをチェックし、悪意ある動作を検出
    通常、複数ラウンドの投票や検証が行われて合意に達する
3. 結果の確定
    十分な数のノード(通常は3分の2以上)が同意すると、そのリクエストがシステム全体の合意として確定する

特徴

悪意があるノードが存在しても条件を満たしていれば正しい合意が得られる
合意が確定すると覆らない。つまりファイナリティがある。
許可型ネットワークに向いている
    ノード同士で通信し合うし、信頼するノード数が必要なのでかな？
通信が多いので遅くなりがち。

使用例

詳細は Grok を読んでおくれ。

PBFT (Practical Byzantine Fault Tolerance)

- Hyperledger Fabric で使われている

Tendermint

- Cosmos で使われている

HotStuff

- Diem で使われている

他のアルゴリズム

Consensus Algorithms Compared: PoA vs IBFT vs Raft