HotStuff

https://arxiv.org/pdf/1803.05069

HotStuff consensus. HotStuff is a variant of the BFT... - by Gerald Lee - Medium

Grok

HyperBFT の元になったアルゴリズム。
リーダー主導型BFT で部分同期通信モデルを採用している。
ビザンチン耐性としては他の BFT系と同じ強度を持ち、3分の2よりも多数を正直なノードが占めていれば成り立つ($n = 3f + 1$)。 `f=10` なら `n=31 (t=21)` あればよい。

部分同期通信モデル

BFT うんぬんではなく通信の遅延などをどう定義するかというモデルの話のようだ。
部分同期通信モデルは、ある時点までは非同期で振る舞い(つまり遅延が不定)、それ以降は固定の最大遅延時間以内に通信が完了する(つまり同期)という考え方をする。
その「ある時点」を GST(Global Stabilization Time、グローバル安定化時間)と呼ぶ。その時点は未知で、プロトコルで規定したりするものでもない。

- すべてのメッセージは最終的には相手に届く、という前提
  - その代わり遅延は無限と考え、いつ届くのかは分からない
- GST 以前の非同期的な期間でも安全性を常に維持するようにする。
  - Safety と呼び、具体的には不正な決定をしないようにする。
- GST 以降の同期的な期間は最終的な決定になることを保証するようにする。
  - Liveness と呼ぶ。

HotStuffでは.....

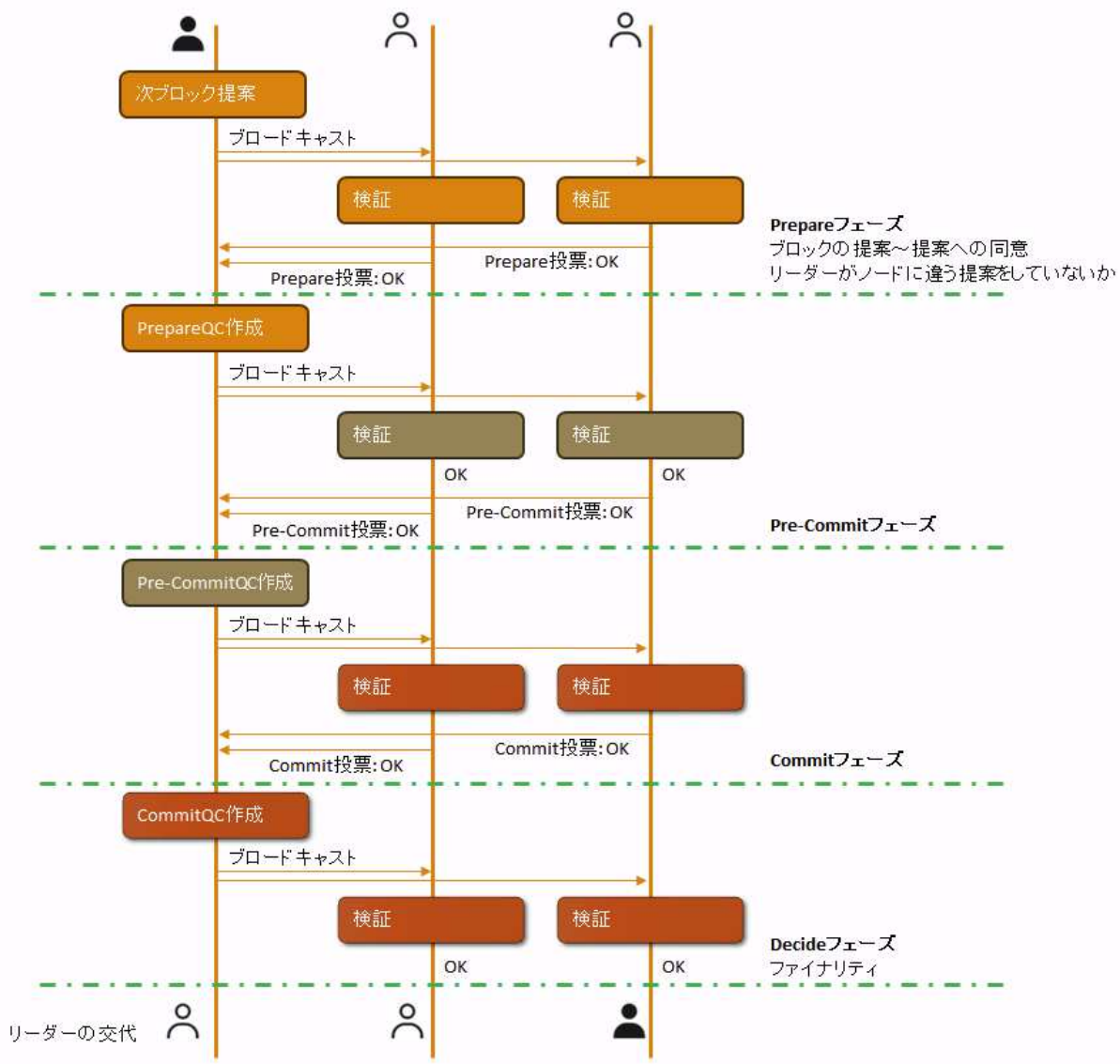
- メッセージが速く届く場合、つまり固定の最大遅延時間以内にメッセージが届くような状態は「今は GST以降の同期的期間」と考える。
  - 特に問題はないのでばんばん処理を進めればよい。
- メッセージが届かない場合、GST はわからないが HotStuff はリーダーが交代する「View チェンジタイムアウト」が存在するのでリーダーを交代してやり直す。

という感じで、GST を考えないというか自分で決めるというか、そういう感じである。

(Basic) Hotstuff

3フェーズで段階的にコミットしていく。
リーダーはビュー（View: ラウンド）ごとに選出され、タイムアウトでView Changeが発生する。
1回のビューは「準備(prepare)」「事前コミット(pre-commit)」「コミット(commit)」となる。
最後の CommitQC をブロードキャストするところを「Decide」フェーズと呼んだりもする。

1. Prepare Phase（準備フェーズ）
   リーダーが新ブロックの提案をブロードキャスト
   各ノード(レプリカと呼ぶようだ)はそれを検証し、問題なければリーダーに Prepare投票(vote)を送信
   リーダーは「全ノードの 3分の2 + 1」以上の Prepare投票を集め、PrepareQC（Quorum Certificate）を作成
      リーダーがレプリカに違うブロックの提案をしないよう
      QCは十分な数のノードがそのブロックを承認したことを証明
2. Pre-Commit Phase（事前コミットフェーズ）
   リーダーは PrepareQC をレプリカにブロードキャスト
   QC を受け取ったノードは検証してリーダーに Pre-Commit投票を送信
   リーダーは Pre-Commit投票を集め、Pre-CommitQCを作成
3. Commit Phase（コミットフェーズ）
   リーダーは Pre-CommitQCをブロードキャスト
   QC を受け取ったノードは現在のブロックのコミット準備をしてリーダーに Commit投票を送信
   リーダーは Commit投票を集め、CommitQC を作成
4. Decide Phase
   リーダーは CommitQC をブロードキャスト(Decideメッセージと呼ぶ)
   レプリカは返信不要
5. View Change（ビュー変更）
   リーダー交代



リーダーの交代というか Viewの切り替わり時にも通信が発生し、そのときに highQC というデータを作るそうだ。
ブロック提案にはその highQC も含んでいる。

同じようなことを繰り返している理由

Prepareフェーズ：提案の初期合意

Pre-Commitフェーズ：提案のロック

Commitフェーズ：コミットの準備

Decideフェーズ：ファイナライズ

段階的にコミットしているのだが、PBFT のように 2フェーズで足りないというわけではなく View Change(リーダーの交代)での通信複雑度が PBFT だと $O(n^3)$ になるところを4フェーズにすることで $O(n)$ にできているそうだ。

HotStuffのViewChange

レプリカが新しいリーダーに new-view メッセージを送信(O(n))

各 new-view メッセージは highest prepareQC を含む(O(1))

リーダーは (n - f) 個の new-view メッセージから highQC を選択して提案をブロードキャスト (O(n))

認証子(authenticators): O(n)

デジタル署名とMAC

Chained HotStuff

QCを再利用するということと、楽観的に「Prepareフェーズまで通ったブロックはそのままファイナライズされるだろう」と考えてパイプラインにする。

いわゆるファイナリティを得られるのは 3ブロック後ということになる。

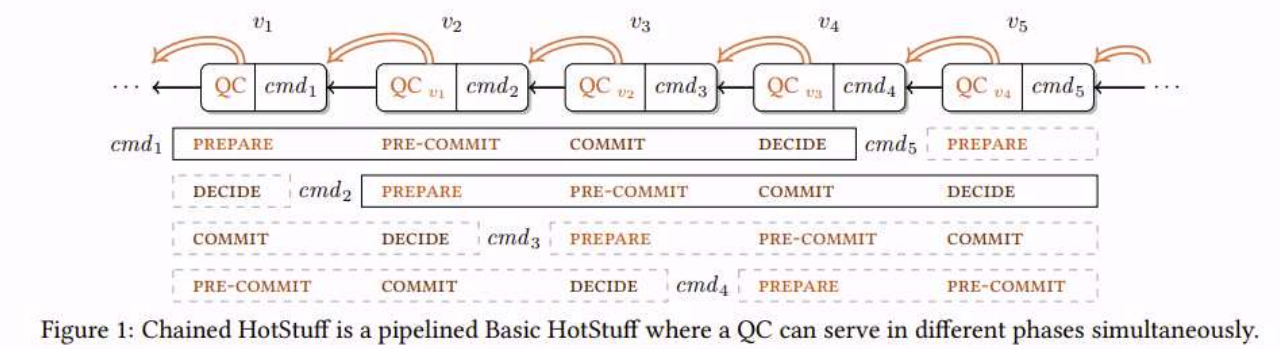


Figure 1: Chained HotStuff is a pipelined Basic HotStuff where a QC can serve in different phases simultaneously.

Hyperliquid は「one-block finality」とうたっているが、HyperBFT がそうであるとは書いていない。「Every order, cancel, trade, and liquidation happens transparently with one-block finality inherited from HyperBFT.」なので HyperCore 側のことを指しているのかもしれない。

Quorum Certificate(QC)

QC は閾値署名を中心に暗号計算をして得られる。

Basic HotStuff では以下の QC を作って展開される。

PrepareQC: 提案ブロックの初期合意を証明

Pre-CommitQC: ブロックのロックを証明

CommitQC: ブロックのファイナリティを準備

Chained HotStuff ではこれらを genericQC として統一し、Three-Chain(3連続のQC)でファイナリティを達成したと見なす。

1. Partial Signatureの生成

   レプリカはリーダーから受信したメッセージの正当性を検証(View番号やフェーズの情報、署名の検証など)

   成功したら、レプリカは Partial Signatureを生成してリーダーに送信

secp256k1 や BLS署名アルゴリズムなど

これが「投票」になる

2. Partial Signatureを結合して QCを作成

リーダーはレプリカから Partial Signature を受け取った署名を結合して単一の閾値署名を生成

生成した閾値署名を含めたメッセージ QC を作ってレプリカにブロードキャスト

3. QCの検証

1 に戻る

DiemBFT (LibraがDiemになったのは2020年12月1日。2021年1月には売却済み)

State Machine Replication in the Libra Blockchain 2019/10/24

State Machine Replication in the Libra Blockchain 2020/05/26

State Machine Replication

DiemBFT v4: State Machine Replication in the Diem Blockchain 2021/08/17

Diemチームは解散し、一部メンバーが立ち上げた Aptos Labs が技術的に引き継いだ感じ(売却先ではなさそう)。

Sui も Diemチームの人が立ち上げた人たちが立ち上げたらしい

改良

同期状態で遅延を制限するラウンド同期メカニズム

リーダーに障害があっても提案をコミットできる nil ブロック提案

参加者の攻撃対象領域を削減する安全なハードウェアenclave内で実行

Narwhal & Bullshark - Sui

機能を、通信関係の Narwhal、コンセンサスの Bullshark に分けた