Blockchain implements in Christie

155753A 氏名 赤堀貴一 指導教員：河野 真治

Abstract

Data corruption and inconsistency in computers causes severe problem. Therefore, detect corruption and inconsistency by Blockchain. The Blockchain is a distributed system. It is possible to compare between hash value and data which is correct. Even if there are incorrect operation and tampering, data are possible to recover with Blockchain.

We are developing Christie and GearsOS. Christie is a distributed framework and GearsOS is a Operating System. GearsOS Filesystem will refer to Christie. if implementing block chain in Christie and implementing it in GearsOS, makes it possible to detect data corruption and inconsistency in the GearsOS file system. In addition, it is possible to configure decentralized distributed network between Gears OS. if it does not configure, data is protected. So, our purpose can be achieved.

In this study, We implement Blockchain in Christie and run it in distributed environment on PC cluster.

1 研究目的

コンピュータにおいてデータの破損や不整合は深刻な異常を引き起こす原因となる. そのため, 破損, 不整合を検知するためにブロックチェーン技術を用いたい. ブロックチェーンは分散ネットワーク技術であり, データの破損や不整合をハッシュ値によって比較できる. そして, 誤操作や改ざんがあった場合でも, ブロックチェーンを用いることでデータの追跡が行える.

当研究室では分散フレームワークとして Christie を開発しており, これは GearsOS にファイルシステムに組み込む予定がある. そのため, Christie にブロックチェーンを実装し, GearsOS に組み込むことにより, GearsOS のファイルシステムにおいてデータの破損, 不整合を検知できる. また, GearsOS 同士による分散ファイルシステムを構成することができ, 非中央的にデータの分散ができるようになる. もし分散システムを構成しない場合でもデータの整合性保持は行え, 上記の目的は達成できる.

本研究では, Christie にブロックチェーンを実装し, 実際に学科の PC クラスタ上の分散環境で動かす.

2 ブロックチェーン

ブロックチェーンとは分散型台帳技術とも呼ばれ, 複数のトランザクションをまとめたブロックをつなげたものを, システムに参加しているすべてのノードが参照できる技術である. ブロックチェーンを実装することは次のようなメリットが有る.

- データの追跡, 検証が容易である.
- 中央管理者が存在しない. 単一障害点がない.

データの追跡, 検証が容易であるのは, ブロックの構造によるものである. ブロックの構造は簡易化すれば次のようなものである.

- 前のブロックの暗号化ハッシュ.
- タイムスタンプ.
- トランザクションリスト.

ブロックは図**??**のように hash でつながっている. 一つのブロックが変更されれば, その後に連なるブロックも整合性が保たれないため, これによってデータの整合性保持が行える.

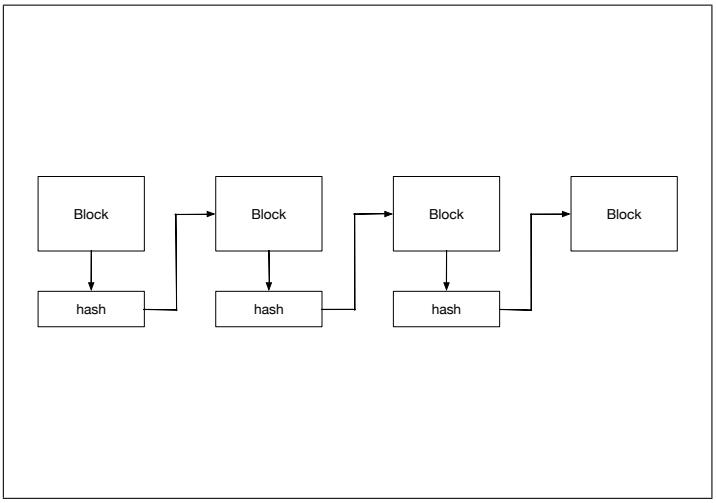

図 1: hash chain

トランザクション, ブロックともにノード間で伝搬され, ノードごとに検証される. そして検証を終え, 不正なトランザクション, ブロックであれば破棄する. 検証に通った場合は, トランザクションは Transaction Pool に Transaction を貯めておき, ブロックはブロックチェーンに取り組まれ, また検証したノードからトランザクション, ブロックがブロードキャストされる. ノード間は P2P で通信が行われている.

同時に異なるノードで複数のブロックができることを, fork という. これによってブロックチェーンの分岐が起こる. ブロックチェーンの分岐を収束させるにはコンセンサスアルゴリズムを使用する.

3　コンセンサスアルゴリズム

コンセンサスアルゴリズムとは, 一意の値を分散環境上で決めるためのアルゴリズムである.

今回は分散アルゴリズムとして Paxos を実装した.

4　Christie

Christie は当研究室で開発している分散フレームワークである. Christie は Java で書かれているが, 当研究室で開発している GearsOS に組み込まれる予定がある. そのため, GearsOS を構成する言語 Continuation based C と似た概念がある. Christie に存在する概念として次のようなものがある.

- CodeGear(以下 CG)
- DataGear(以下 DG)
- CodeGearManager(以下 CGM)
- DataGearManager(以下 DGM)

CG はクラス, スレッドに相当し, DG は変数データに相当する. CGM はノードであり, DGM, CG, DG を管理する. DGM は DG を管理するものであり, put という操作により変数データ, つまり DG を格納できる.

DGM には Local と Remote と 2 つの種類があり, Local であれば, Local の CGM が管理している DGM に対し, DG を格納していく. Remote であれば接続した Remote 先の CGM の DGM に DG を格納できる. DG を取り出す際にはアノテーションを付けることで, データの取り出し方も指定できる. Take, Peek という操作があり, Take は読み込んだ DG が消えるが, Peek は DG を消さずにそのまま残す. また, RemoteTake, RemotePeek というものもあり, リモート先を指定することにより, RemoteDGM からデータを取ることができる.

CG は CGM によって実行されるが, 実行するには CG に必要な DG が全て揃う必要がある. もし DG が全て揃わない場合, CGM はずっと listen し, データが揃うまで実行を待つ.

5　まとめ