Christieによるブロックチェーンの実装

155753A 氏名: 赤堀 貴一 指導教員: 河野 真治

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, 近年注目されたブロックチェーン技術を用いる. ブロックチェーンは分散システムとして注目されており, 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簡単に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として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検証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.

よって, Christie にブロックチェーンを実装し, 分散環境でのデータの整合性保持, 追跡を行う.

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.

ブロックは前のブロックと暗号化ハッシュでつながっており, 現在のブロックのハッシュは前のブロックのハッシュに依存して作られる. そのため, もしブロックを改ざんしたいとしたら, そのブロックにつながるすべてのブロックを改ざんしなければならない. しかし, その仕組みだけだと簡単に改ざんができてしまう. そのため, ブロックに付け加える場合にはある作業を行わせ, それによってある条件に収まる Hash を作らせる. 例えば, ビットコインだと Proof of Work という計算問題を解かせ, Hash を生成する. これは単純には

```
while(1){
    randomSeed(前のHash + nonce)
    // 0 < rand() < 10000
    このブロックのHash = rand() \% 10000
    if (このブロックのHash < 100){
        break
    }
    nonce ++
}
```

と言う問題を解くのと同義である. 実際には $0 < rand() < 10000$ はもっと大きな値である.

もし至るところでブロックが作られ, 競合すると, 競合したブロック同士で, つながっているブロックが多いものを正しいブロックとする.

通信は p2p で行われ, ブロックが承認された場合, 他のノードにブロードキャストされる.

3 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 CodeGear(以下 CG) と DataGear(以下 DG) という概念がある. CG はメソッドであり, DG は変数データに相当する. また, Christie には CodeGearManager(以下 CGM) と DataGearManager(以下 DGM) という概念もある. CGM はノードに当たり, 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, その CGM に DG を格納していき, Remote であれば接続した Remote 先の CGM 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.

CG は CGM によって実行されるが, 実行するには 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 する.