ブロック作成時のハッシュ化研究

SK3A　2220042 文家俊

1. 前書き

本研究では、ブロック作成の効率を向上させる方法を検討しました。前回の研究から、速度の改善が応用範囲の拡大に寄与する可能性が示唆されており、また、パソコンへの負荷も考慮する必要があります。これらの課題に対し、最適なアプローチを模索しました。

2. ハッシュ関数の選定とツールの活用

Sha256 を含む 9 種類のハッシュ関数を選定しました。そのうち 8 種類は比較的利用頻度が低く、あまり注目されていませんでした。これらの性能を検証することで、新たな選択肢を見出すことを目的としています。実験では、GOMAXPROCS で CPU コア数を調整し、Goroutine による並行処理でタスクを効率化しました。また、Pprof を活用し、CPU やメモリ使用状況の可視化と最適化を行いました。

3. 1 コアと 4 コアの比較結果

1 コアで実行した場合、1MB 以下のファイルでは処理時間の差があっても、全体的な時間大体五分以内ですが、大きなファイルでは 30 分から 1 時間以上かかるケースが確認されました。これは 1 コア環境では並行処理が行えないと考えられます。 一方、4 コア環境では全体の処理時間が大幅に短縮されました。また、小さなファイルでも処理時間のばらつきが抑えられ、安定性が向上しました。大規模データでも並行処理による効果が顕著でした。

4. ハッシュ関数の性能分析

9 種類のハッシュ関数を比較した結果、Murmur3 が Sha256 よりも優れた結果を示しました。特に大規模データでは、Murmur3 の処理速度が Sha256 より平均 30%以上速いことが確認されました。一方、Blake3 も高速でしたが、安定性では Murmur3 が最も優れていました。これらの結果から、軽量ハッシュ関数は処理効率を向上させる有力な選択肢であるといえます。

5. 結論

4コア環境は、1コア環境と比較して処理時間短縮と安定性向上において効果的であることが明らかになりました。また、Murmur3のような軽量ハッシュ関数は速度と効率性に優れており、ブロック作成の最適化に寄与する可能性が高いと考えられます。これらの成果は、ブロックチェーン技術のさらなる実用化や応用範囲の拡大に大きく貢献するものです。