BigchainDB 入門

BigchainDB GmbH, ベルリン, ドイツ

2017 年 5 月

ソース元：https://www.bigchaindb.com/whitepaper/bigchaindb-primer.pdf

■要旨

BigchainDB はスケーラブルなブロックチェーンデータベースで、「分散化」、「不変性」、「アセットの作成と転送のためのビルトインサポート」を含むブロックチェーン特性を持ったビッグデータデータベースです。分散型技術は以前からインターネットの一部でありましたが、Bitcoin の登場で新たな波が生まれました。特に、Bitcoin は、中央の信頼できる第三者に依存することなく、信頼できる不変のトランザクションレコード（台帳）を作成することが可能であることを示しました。多くの人は、Bitcoin の基盤となるブロックチェーン技術であらゆる種類の記録を保存出来ると考えていますが、Bitcoin 自体は、現状すべてのレコードを処理できるようにスケールすることはできません。 Bitcoin や類似のブロックチェーンをスケールするために、多くのアイデアやプロトタイプがあります。一方、BigchainDB はスケーリングの問題を解決するために異なるアプローチをとっています。実証済みのスケーラブルでビッグデータのデータベースをベースとしており、そこに ブロックチェーン特性を追加しています。

■ 1． 概要

BigchainDB はスケーラブルなブロックチェーンデータベースで、分散化、不変性、アセットの作成と転送のための API を含むブロックチェーン特性を持ったビッグデータデータベースです。この入門は、「BigchainDB がなぜ必要なのか」「ブロックチェーンの特性をどのように追加するのか」「分散型エコシステムの残りの部分にどのように適合するか」「どのように試すことができる」について説明します。

■ 2． ビットコインとその他の分散型プロトコル

初期のインターネット（ARPAnet）の 1 つの目標は、核戦争に耐えられる通信ネットワークを作り出すことでした。その目標を達成するために、設計者は、単一障害点である中央集権型の必要性を排除しました。そのため、主要なインターネットプロトコルは分散型となっており、他のインターネットベースのプロトコルも一部分散型となっています（例えば、電子メール、DNS、BitTorrent など）。しかし、時間が経つにつれて、インターネット上では多くの中央集権型サービスと API が普及してきました（Facebook や Google の検索など）。

2009 年に導入された Bitcoin は、分散型プロトコルであり、この状況に一石を投じました。その理由は、中央銀行や他の信頼できる第三者を必要としない、分権化された暗号化の動作が実証されたためです。信頼はプロトコルとその永続的な蓄積（すなわち長期的な Bitcoin ブロックチェーン）から生まれ、世界中の何千ものコンピュータに一貫して検証可能にコピーされます。

■ 3．ブロックチェーンは他のデータを記録できるのか？

Bitcoin が中央の所有者またはコントローラを必要とせずに信頼できる Bitcoin トランザクションのレコードを生成できる場合、Bitcoin または類似のテクノロジを使用して、その他の信頼できる重要な情報を記録できますか？出生証明書、大学の成績証明書、特許出願、土地所有権取引、美術所有権取引、サプライチェーン記録、研究室報告書、政府の透明性報告書などを考えてみてください。これらの例の中には、非常に高速にレコードが生成されるものがあります。Bitcoin で対応しきれるのでしょうか？

■ 4．Bitcoin と既存データベースの比較

現状の Bitcoin ネットワークでは、1 秒あたり 10 トランザクション（平均）未満のトランザクションしか記録出来ず、ノード数を増やしても改善されません。Bitcoin トランザクションには任意の情報を格納できますが、数十バイトもありません。 Bitcoin ブロックチェーンの総ストレージ容量は 10 分ごとに約 1 メガバイト増加しますが、全てのデータで 1 メガバイトになります。こちらも、ノードを増やしても保存容量は増加できません。さらに Bitcoin トランザクションが確定されるまでに数十分から数時間かかります。

一方、既存のビッグデータデータベースを見てみましょう。トランザクション性能は、1 秒あたり数百万回を処理可能であり、ノードを追加すれば性能は上がります。たとえば、Netflix は 2011 年に Cassandra(OSS の分散型 DB)をテストしました（図１）。プロットの左下に、50 個の分散型 Cassandra ノードが 1 秒あたり 174,000 回の書き込みを処理できることがわかります。 300 ノードまで増加すると、毎秒 110 万回の書き込みが可能になりました。 3 年後のフォローアップ調査では、わずか数十のノードで 1 秒あたり 100 万回の書き込みスループットが示されました。

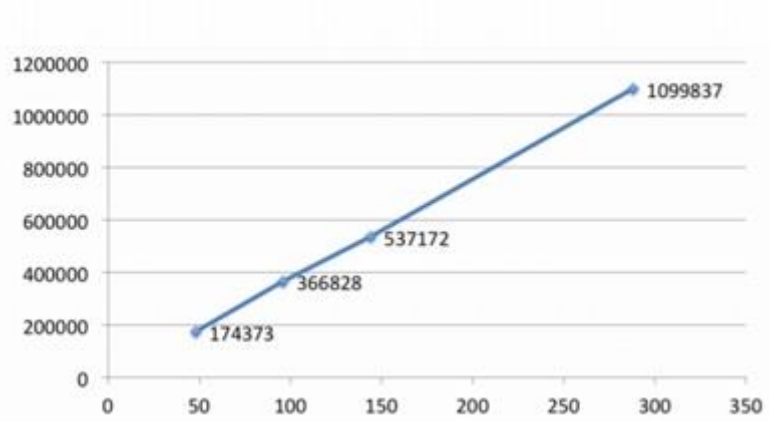

図１ ：Netflix の Cassandra テスト

（X 軸はノード数、Y 軸は書き込み回数）

既存データベースの他の特性を考えてみましょう。個々のレコードをかなり大きくする（メガバイト）ことが可能で、ノードを追加するにつれて（ペタバイト以上に）合計ストレージ容量を増大する事が可能です。世界規模の分散型データベースのトランザクションの待ち時間は、数百ミリ秒になることがあります（単一のデータセンター

では、主に光の速度によって決まります）。さらに、既存のデータベースはすべて、豊富なクエリと権限をサポートしています。

ストレージコストは、Bitcoin を使用する際の大きな制限です。 2017 年初期の時点で、Bitcoin の取引確認時間を短くしたい場合は、取引金額で約 25USD を支払う必要があります。これは、格納されている任意の 1 バイトあたり約 1USD、またはギガバイトあたり 1,000 万 USD です。 Amazon Glacier（長期利用可能かつ安全で耐久性のあるデータストレージ） のコストは、ギガバイトあたり 0.004USD/月なので 40USD あれば 800 年以上使えます。（実際には時間が経つにつれて安価になる可能性があります）。

Bitcoin や類似のブロックチェーンの調整や改善、スループットの向上、スケーラビリティの向上、ストレージ容量の増加、レイテンシの削減など、多くの提案があります。それらの提案は、ブロックチェーンのプロトコルにデータベースの要素を追加しています。

■ 5．BigchainDB による解決策

BigchainDB は異なるアプローチをとっています。大規模なデータベースをベースとした上で、 分散化、不変性、アセットの作成と転送などのブロックチェーン特性を加えています。

	Typical Blockchain	Typical Distributed Database	BigchainDB
High Throughput; increases with node count		✓	✓
Low Latency		✓	✓
High Capacity; increases with node count		✓	✓
Rich querying		✓	✓
Rich permissioning		✓	✓
Decentralization	✓		✓
Immutability	✓		✓
Built-in support for creation & transfer of assets	✓		✓

現在、データベースのバックエンドとして MongoDB または RethinkDB を使用できますが、高いパフォーマンス、強力な一貫性、最新のツールを備えた、信頼性の高いエンタープライズクラスのデータベースである MongoDB を推奨します。図 2 と図 3 は、BigchainDB フェデレーション(ID 認証)/クラスタのアーキテクチャを示しています。外部クライアントは、クラスタ内の BigchainDB Server の任意のインスタンスと通信できます。各 BigchainDB Server インスタンスは、MongoDB データベース内の 1 つのノードに関連付けられています。

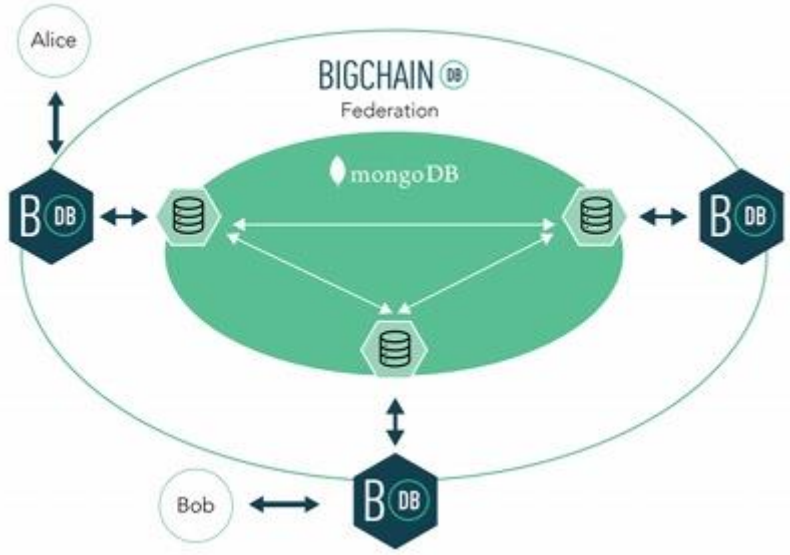

図 2 ：BigchainDB には 2 つの層からなるコンセンサスがあります。
下位レベル（MongoDB）は、トランザクションの順序とクラッシュ障害を処理します。
上位レベル（BigchainDB）のフェデレーションは、二重支払いや悪意のある攻撃を対処します。

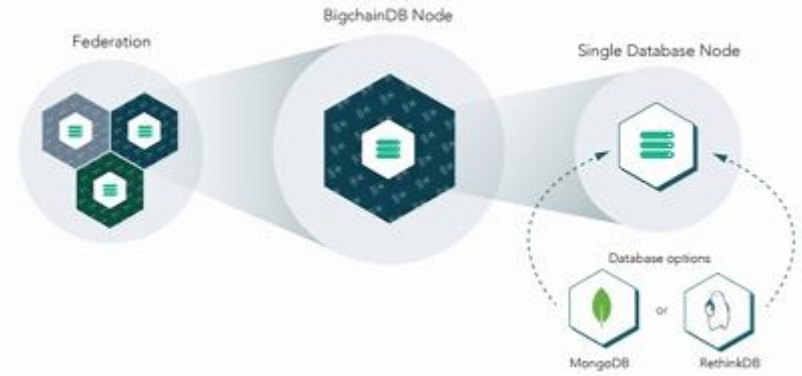

図 3 ：BigchainDB フェデレーション/クラスタには、いくつかの独立した BigchainDB ノード
が含まれています。各 BigchainDB ノードには、MongoDB（または RethinkDB）
データベースの 1 つのノードが含まれています。

【分散化】重要な要件の 1 つは、クラスタ内のすべてのノードを別の人や組織が所有して操作する必要があることです。これはコードによって処理されません。これは BigchainDB クラスタを監督する組織によって強制されなければならないものです。その組織では、参加者/ノードの追加や削除（ガバナンスが必要）などのルールとプロセスが必要です。民主主義/独裁政治の程度は、組織によって異なる場合があります。理想的には、ノードは多くの国、法的な管轄区域、ホスティングプロバイダーに置かれるべきであるため、1 つの問題はそれらすべてに影響しません。他のブロックチェーンと同様に、有効な転送トランザクションを作成できるのは、必要な秘密鍵を持つ人だけです。ノードオペレータを含む他の誰もそれを行うことはできません。取引する権限は分散されています。 BigchainDB のロードマップには、他の分散機能を追加するための計画が示されています。

【不変性】ブロックチェーンは強力な改ざん耐性があります。改ざんが行われても、暗号署名、ハッシュチェーン、マークルツリーの広範な使用のために、それが検出（および拒否）されます。ブロックチェーンコミュニティーでは、これらの特性はしばしば「不変性」と呼ばれます（これは通常の意味とは異なりますが、言語は生きたものであり、単語はしばしば付加的な意味で過負荷になります）。 BigchainDB はまた、暗号署名とハッシュを幅広く利用しています。たとえば、すべてのトランザクション、ブロック、投票が署名されます。また、トランザクションとブロックはハッシュによって識別されます。改ざん耐性も、データの複数のコピー（すなわち、高い複製因子）を有することによって増加する。 改ざん耐性は、セキュアなバックアップやセキュアなクラスタインフラストラクチャ（セキュアなネットワーク、セキュアなオペレーティングシステムなど）などの標準的な技術によっても増強できることに注目してください。他の改ざん防止機能を追加する計画については、BigchainDB のロードマップで概説されています。

【アセットの作成と転送のためのビルトインサポート】銀行は常に資産を移転するためにデータベースを使用していますが、データベース自体には資産の概念が組み込まれていません。その機能はアプリケーション層に追加されます。 BigchainDB には、大部分のブロックチェーンと同様に、組み込みのトランザクション妥当性チェック（例えば、署名のチェック、二重支出のチェック、出庫が有効なトランザクション内でのチェックなど）が組み込まれています。 ）。

【シビル耐性】一部のブロックチェーンネットワーク（Bitcoin など）では、誰でも自分のノードをネットワークに追加することができます。そのため、誰かが非常に多くのノードを追加してネットワークを効果的に制御できるという懸念があります。これはシビル攻撃と言われます。Bitcoin では、シビル攻撃を行なうためには法外なコストが必要なので起きにくい。BigchainDB ネットワークでは、ネットワークの背後にある統治組織がメンバーリストを管理しているため、シビル攻撃は問題ではありません。

■ 6． 分散型エコシステムにおける BigchainDB

BigchainDB は、分散型処理/スマートコントラクト（例えば、Ethereum や Enigma）、分散ファイルシステム（IPFS など）、コミュニケーション基盤（電子メールなど）を補完するものであり、高レベルの分散コンピューティングプラットフォーム（例えば、Monax や Tendermint）に含めることができます。また、ID 認証プロトコル、金融資産プロトコル（Bitcoin など）、知的財産資産プロトコル（SPOOL など）、グループロトコル（例えば pegged sidechains、Interledger など）と併用可能です。スマートコントラクトに対するスケーラビリティの向上は、完全分散型アプリケーションが BigchainDB のスケーラビリティ特性をより有効に活用するのに役立ちます。

BigchainDB はより集中化されたコンピューティングシステムでも動作します。 1 つのユースケースでは、ストレージの分散化が大きなメリットをもたらします。もう 1 つのユースケースは、スケーラビリティニーズが既存の分散処理テクノロジの機能よりも大きい場合です。この場合、BigchainDB は最終的には完全分散システムへの架け橋となります。

図 4 は、BigchainDB における完全分散化する場合と、中央集権型サービスからの拡張を行なう場合を示しています。

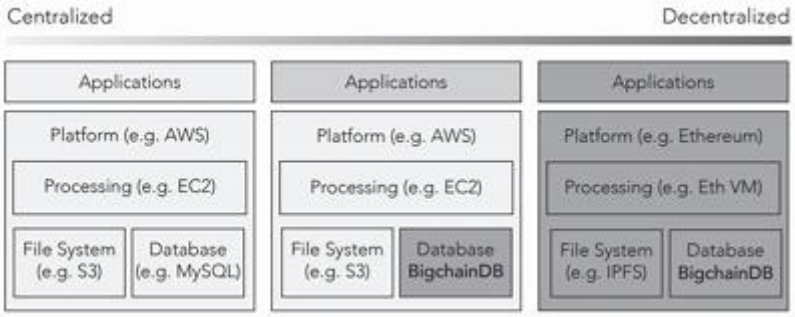

図 4 ：集中型クラウドコンピューティングエコシステムの基本コンテキスト（左）から、BigchainDB を別のデータベースとして追加して、分権化のメリット（中）を得ることができます。また、完全分散システム（右）にも適合します。

■ 7． BigchainDB を試す

BigchainDB を試してみたい場合、接続する BigchainDB クラスタが必要です。 やり方の 1 つは、自身のローカルマシンに、シングルノードの BigchainDB クラスタをインストールすることです。"BigchainDB Server docs"のクイックスタートに、インストール手順が記載されています。もう 1 つのやり方は、IPDB テストネットワーク（IPDB Foundation によって管理）など、既存の BigchainDB クラスタに接続することです。

接続する BigchainDB クラスタがあれば、BigchainDB HTTP API を使ってやり取りすることができます。BigchainDB は、WebSocket Event Stream API も提供しています。BigchainDB トランザクションを作るのに役立つ多くのツールやライブラリ（CLI、Python ドライバ、Javascript トランザクションビルダ、Java ドライバ、Go ドライバなど）があります。リスト 1 に、BigchainDB Python ドライバを使った Python コードの例を示します。

Listing 1: Example Code Using the Python Driver

```
from bigchaindb_driver import BigchainDB
from bigchaindb_driver.crypto import generate_keypair
bdb = BigchainDB('http://example-bdb-cluster.net:9984')
alice = generate_keypair()
new_asset = {
    'data': {
        'piece_name': 'Avocado Green',
        'artist_name': 'Parker Millson',
        'edition_number': 9
    }
}
prepared_creation_tx = bdb.transactions.prepare(
    operation='CREATE',
    signers=alice.public_key,
    asset=new_asset)
fulfilled_creation_tx = bdb.transactions.fulfill(
    prepared_creation_tx,
    private_keys=alice.private_key)
sent_creation_tx = bdb.transactions.send(fulfilled_creation_tx)
```

4 行目は、アリスという名前のユーザーの公開鍵/秘密鍵ペアを生成します。 5 行目～11 行目は、new_asset という名前の Python dict を作成します。12～15 行目は、アリスのために new_asset に基

づいてアセットを作成するための生の（符号なしの）BigchainDB CREATE トランザクションを作成します。16～18 行目は、アリスの秘密鍵を使用して、そのトランザクションに署名しています。 19 行目は、BigchainDB クラスタにそれを送信します。 Python driver docs には、さらに多くのサンプルコードがあります。

■８．製品化に向けて

前述のように、BigchainDB クラスタは、複数の異なる人や組織によって運用される必要があります。たとえば、単一の企業内では、各ノードを別のビジネスユニットで運用することができます。またはコンソーシアム内で、各参加組織はノードを実行できます。各オペレーターは、BigchainDB サーバーのインスタンス、MongoDB のインスタンス、MongoDB 用のストレージ、および潜在的に他のコンポーネントを含む「BigchainDB ノード」をインストールして実行する必要があります。 BigchainDB Production Nodes に詳細があります。

■９．結論

多くのアプリケーションでは、Bitcoin のように分散されたものを使用できますが、大量の任意のデータを短時間で、改ざん無く、低遅延で格納することができません。 BigchainDB は、その課題に対する私たちの答えです（分散化、不変性、資産の作成と転送のための組み込みサポートを含むブロックチェーン特性を備えた大規模なデータベース）。

詳細は、BigchainDB のウェブサイト(bigchaindb.com)を参照してください。

■備考

・ソース元で、9. Conclusion の「but capable」の部分は「but incapable」の誤り。
（BigchainDB CEO の Bruce に確認済）
・より仕組みの詳細については下記参照
https://www.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf