323

NewSQLとは？

- NoSQLデータベースはビッグデータへの対応では成功したが、以下の問題点がある
  - ➢トランザクション処理を備えていない、もしくは弱い（ACID特性の一部が厳密には保証されない）
  - ➢複数データセットにまたがる問合せ（SQLでは結合問合せ）と問合せ最適化を備えていない、もしくは弱い
- NoSQLデータベースをベースに、RDBMSが備える上記の機能を実装しようという動きがあり、**NewSQL**と呼ばれている
- NewSQLの例：**Spanner**, VoltDB, ScaleDB, NuoDB, Clustrix, FoundationDB, **CockroachDB**

324

（参考）トランザクションが満たすべき性質

以下の4つの性質があり、頭文字を取って**ACID特性**と呼ばれることがある

- 原子性（atomicity, 不可分性とも呼ばれる）
- 整合性（consistency、一貫性とも呼ばれるがintegrityと区別するためここでは整合性と呼ぶ）
- 独立性（isolation）
- 永続性（durability）

325

ACID特性が保証されないとどうなるか?

「銀行口座間の振替」を例にとると次のようになる

- 口座1から引き出しているのに口座2に預け入れされていない（**原子性**の欠如）
- 口座1の引出額と口座2の預け入れ額が違っている（**整合性**の欠如）
- 複数の口座から同じ口座に同時に振替が実施されたとき、後の方の振替分しか入金されない（**独立性**の欠如）
- 口座振替作業の途中の障害で、口座振替の結果が失われる（**永続性**の欠如）

326

ACID特性でどう解決するか?

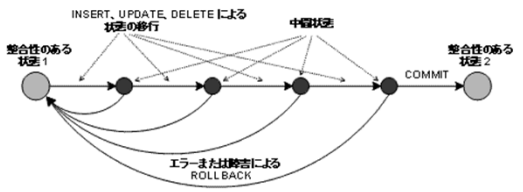
「銀行口座間の振替」の例は次のように行う

- 口座1からの引き出しと口座2への預け入れは、両方振替するか、両方しないかのどちらかである（**原子性**の保証）
- 口座1の引出額と口座2の預け入れ額を常に一致させる（**整合性**の保証）
- 複数の口座からの振替が同時に実施されても、両者は互いに影響しない（**独立性**の保証）
- 口座振替作業の途中で障害があっても、口座振替の結果が失われることはない（**永続性**の保証）

327

（参考）トランザクション処理

トランザクション処理：論理的には一つの操作だが、データベースに対して複数の操作が必要な処理
ACID特性の保証: ACID特性が保証された状態から出発して、更新操作を行った後、**COMMIT**を実施して次のACID特性が保証される状態に移行するか、または障害発生等で移行できないときは**ROLLBACK**で元の状態に戻す



出典　https://help.sap.com/

328

資料の修正

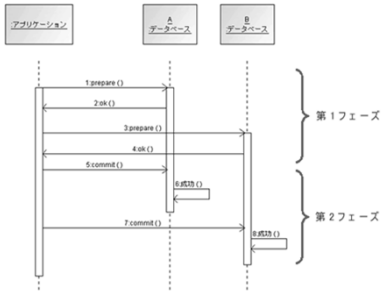
NoSQLデータベースでのトランザクション処理

- なぜNoSQLデータベースでトランザクション処理がほとんど実装されなかったのか?
  - ➢NoSQLデータベースでは、ビッグデータに対応するため複数サーバへの分散（**スケールアウト**）をしている
  - ➢COMMIT/ROLLBACKによるトランザクション処理をそのままスケールアウトすると、一つのトランザクションでサーバごとにCOMMITとROLLBACKが混在するためACID特性が保証されない恐れがある
    → 従来の解決方法　**2相コミット**
  - ➢スケールアウトでは処理効率とACID特性（特に整合性）の両立が困難

講義補助資料 No.11 (1/15)

2相コミット（two-phase commit） 329

分散された複数のデータベースサーバに対して、(1)COMMITの要求を行い、(2)全部が同意すればCOMMIT指示を、一つでも拒否すればROLLBACK指示を送る



出典 http://www.ogis-ri.co.jp/otc/hiroba/technical/DTP/step2/

2相コミットの問題点 330

- 同じデータに対する更新操作はACID特性を保証するため、同時には一つの操作のみロックを取得して、他の操作はロックが取得できるまでブロックする
- アプリケーションからのCOMMITの要求に対して、データベースサーバが同意または拒否を返してから、COMMITまたはROLLBACKの指示が来るまで、他の処理ができないようにブロックする
- ブロックが多発すると、処理の渋滞が発生し、スケールアウトの効果が抑えられてしまう

→RDBMSがスケールアップで性能向上を図る理由の一つ

NoSQLデータベースの対応 331

- 多くのNoSQLデータベースではスケールアウトの効果を優先して2相コミットを採用していないため、ACID特性（主に整合性）を厳密には保証しない
- 結果整合性：一時的には整合性が欠如することがあり得るが、更新操作の実行後に十分時間が経てばいずれは保証される

Spanner 332

- Googleが開発し、Cloud Platform上で提供
- Googleはリレーショナルデータベースサービスと呼んでいるが、以下のような開発の経緯から考えて、**NewSQLデータベース**と位置付けられる
  - **Bigtable**: 列指向型のNoSQLデータベースであり、Google Maps, Google Earth, YouTube, Gmailなど多様なGoogle Appで使われている
  - **Megastore**: Bigtableにトランザクション処理機能を付加
  - **Spanner**: さらにその上にSQLの問合せ機能を付加

Spannerでのトランザクション処理の効率化 333

- Spannerでは**TrueTime**という独自の時刻管理の手法を導入している
- TrueTimeでは、原子時計またはGPSを使用しており、分散データベース間で共通かつ非常に正確な時刻管理（誤差10ミリ秒程度）を行える
- 各操作に起動時刻が書かれており、時刻順に処理の順序を決めることで、更新操作が多発してもブロックを回避するように工夫して、分散環境でACID特性を保証し、かつ効率的なトランザクション処理を実装している

NoSQLデータベースでの問合せ 334

- NoSQLデータベースでは、リレーショナルデータベースの持つ結合問合せのような、複数のデータ単位（リレーションなど）にまたがる問合せを備えていないか、または制限がある
- なぜNoSQLデータベースでは結合問合せがほとんど実装されていないのか？
  - NoSQLはスケールアウト（複数サーバへのデータベースの分散）で効率化を行う
  - 結合問合せは複数のデータ単位の間での比較演算が必要となり、サーバ間の通信が多発するため効率が悪い

335

（参考）索引を用いた結合

表Sの列S.Yに索引が作成されていると仮定すると、索引を用いた結合の実行方法は、次の擬似コードで表現される

for i := 1 to m do

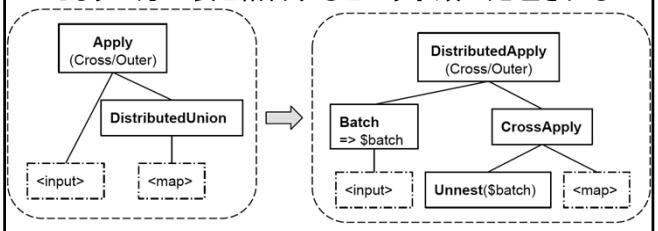
S.Yに関する索引により$r_i[X] = s_j[Y]$を満たすSのタプルs_jを探索

r_iとs_jを結合したタプルを出力

336

分散サーバ上での結合問合せ

- Spannerではデータベースを複数のサーバに分散（sharding）している
- 分散サーバ上での結合問合せは、結合する一方の表のタップルをまとめ（batch）、各batchを分散されたもう一方の表と結合するという手順で処理される



出典 D.F.Bacon et al., Spanner: Becoming a SQL System, SIGMOD'17

337

RDBMS, NoSQLとの比較

	RDBMS	NoSQL	Spanner
スキーマ	○	×	○
SQL	○	×	○
整合性	○（強い）整合性	×結果整合性	○（強い）整合性
トランザクション	○直列化可能性	×	○直列化可能性
拡張性	×垂直（スケールアップ）	○水平（スケールアウト）	○水平（スケールアウト）
DBの複製	△処理系に依存	△処理系に依存	○自動複製

（注）○、×はGoogleによる
出典 https://cloud.google.com/spanner/?hl=ja

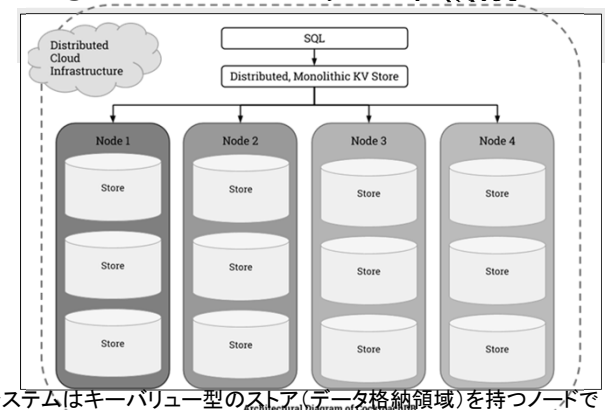
338

CockroachDB

- NewSQLデータベースの一つであり、Spannerに着想を得て開発された（Spannerは商用だがCockroachDBはオープンソース）
- NoSQLデータベースとしての特徴
  - ➢**キーバリュー型**のデータベース（SQLの表形式を内部でキーバリュー型形式に変換）
  - ➢複数サーバへの水平分散（スケールアウト）
- リレーショナルデータベースとしての特徴
  - ➢スキーマによるデータベースの定義が必要
  - ➢ACID特性を備えたトランザクション処理機能
  - ➢問合せ最適化機能を持つSQLインタフェース

339

CockroachDBのノード構成

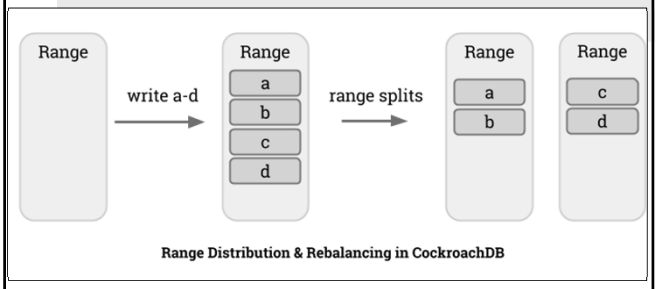


Architectural Diagram of CockroachDB

システムはキーバリュー型のストア（データ格納領域）を持つノードで構成されている（SQL層で隠ぺいされておりユーザには見えない）

340

データの複数ノードへの分散



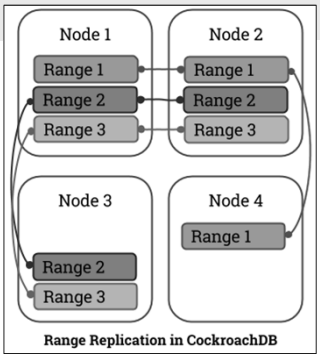
Range Distribution & Rebalancing in CockroachDB

キーバリュー型のストア中のデータは、キーの値の範囲に応じて複数のノードとストアに分散される

講義補助資料 No.11 (1/15)

341

データの複製

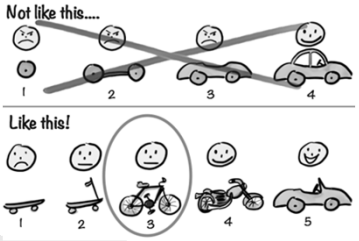


分散されたデータは、さらにあらかじめ決められた数（上の図では3）だけ複製がつくられて、複数のノードとストアに分散される

342

結合問合せ(join)についての注意

- CockroachDBでは結合問合せが実装されているが（現時点のバージョンでは）最適化はされない
- 結合問合せは複数のノードにまたがるので、最適化なしでは処理時間がかかる



343

NewSQLのまとめ

- NoSQLデータベースはビッグデータへの対応では成功したが、以下の問題点がある
  - トランザクション処理を備えていない、もしくは弱い（ACID特性、特に整合性が即時には保証されない）
  - 複数データセットにまたがる問合せ（SQLでは結合問合せ）と問合せ最適化を備えていない、もしくは弱い
- NewSQLは、NoSQLデータベースをベースに上記の機能を実装することで、ビッグデータに対応しRDBMSの機能も併せ持つデータベースの構築を目指している