Tsurugi OLTP

2021年度 Tsurugiユーザー会

Suguru ARAKAWA

NAUTILUS

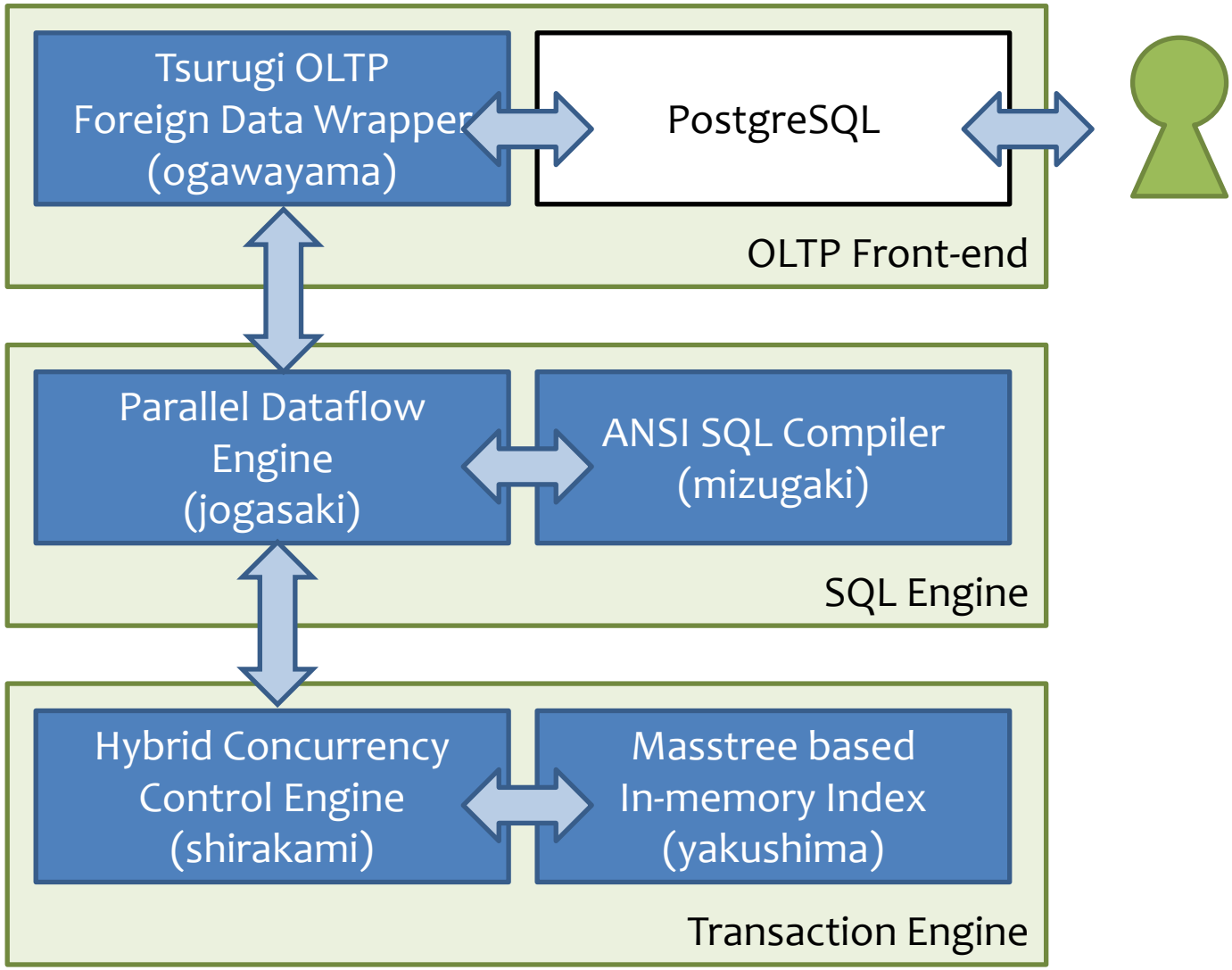
Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

Tsurugi OLTPの特徴

- 現代的アーキテクチャ向けにOLTP処理を最適化
  - マルチコア・大容量メモリー向けに設計
    - 最大で数百万トランザクション/秒規模
    - 大規模な問い合わせ(OLAP)処理も多数のコアとメモリーを活かして高速に処理
- 短いオンライン処理と長いバッチ処理の共存
  - 大量の細かなオンライン処理を捌きながら、裏で長時間のトランザクション処理も稼働できる
  - かつ、これらをSERIALIZABLEに行う
- PostgreSQL経由でアクセス可能
  - フロントエンドにPostgreSQLを配置し、従来のRDBMSのような利便性を確保
  - 性能確保のため、独自方式でのアクセスも提供

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

Tsurugi OLTPアーキテクチャ概観



Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

現代的アーキテクチャへの対応 (マルチコア)

- ロックに頼らない楽観的並行性制御
  - 細かなロックはマルチコアと非常に相性が悪い
    - コア間の通信が多発
    - ロックの待ち合わせによりコンテキストスイッチが多発
  - SILOベースの方式を採用
    - コア間の通信を削減してひとまとめに行うことで、マルチコアでも性能が出る
    - ただし、楽観的な操作なのでAbort率が高まる
- SQLの並列データフロー処理
  - SQLをデータフロー処理に見立てて並列化
    - 余剰コアを活かして長い計算を高速に処理
    - 余剰コアがない場合や短い計算は並列化を抑制してオーバーヘッドを削減

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

現代的アーキテクチャへの対応 (大容量メモリー)

- インメモリーインデックス
  - 単純に大容量のメモリーにインデックスを載せればいいという話ではない
    - ディスクに保存しやすい形式から、ロックフリー処理を活用しやすいインデックスの構造へ
    - ロックフリー処理を活用しないと、コアごとに干渉しあってマルチコアの性能が稼げない
- インメモリーデータフロー処理
  - SQL上の計算もメモリーを多用することで高速化
  - メモリーとCPUの性能格差が広がり、従来方式では性能を稼ぎにくい
    - 高度で複雑なアルゴリズムより、キャッシュを利用しやすい単純なアルゴリズムが勝つ場合も多い

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

現代的アーキテクチャと長時間の処理

- 楽観的並行性制御と長時間処理の相性が悪い
  - 単純な楽観方式ではバッチのような長時間の処理は困難
  - 事前に書き込み先を予約する(write preservation)方式で長時間処理を保護
    - 保護されたトランザクションは、他のトランザクションに邪魔されなくなる
- ハイブリッド方式でワークロードの混在を実現
  - 楽観方式で現代的アーキテクチャ上のメリットを活かし、長時間処理を保護することでAbort率も下げている
    - 従来は難しかった、オンライン処理の裏側で複雑な長時間の処理を稼働できるように
    - これらはSERIALIZABLEで動作し、データの不整合も起きにくい
  - 書き込みの予約先がオンライン処理と衝突する場合は、実施時間帯を分ける等、従来の手法で行える

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

PostgreSQLフロントエンド

- PostgreSQLを介してSQLを処理
  - ユーザー管理やスキーマ管理はPostgreSQLのものを使える
  - SQLコンソールやJDBC, ODBCなどでTsurugi OLTPにアクセス可能
- これだけではTsurugi OLTPを活かしきれない
  - 接続部分が性能のオーバーヘッドになる
  - 書き込み先の予約など、Tsurugi OLTP特有機能の細かな制御ができない
- → より高度な要求のために別口もご用意

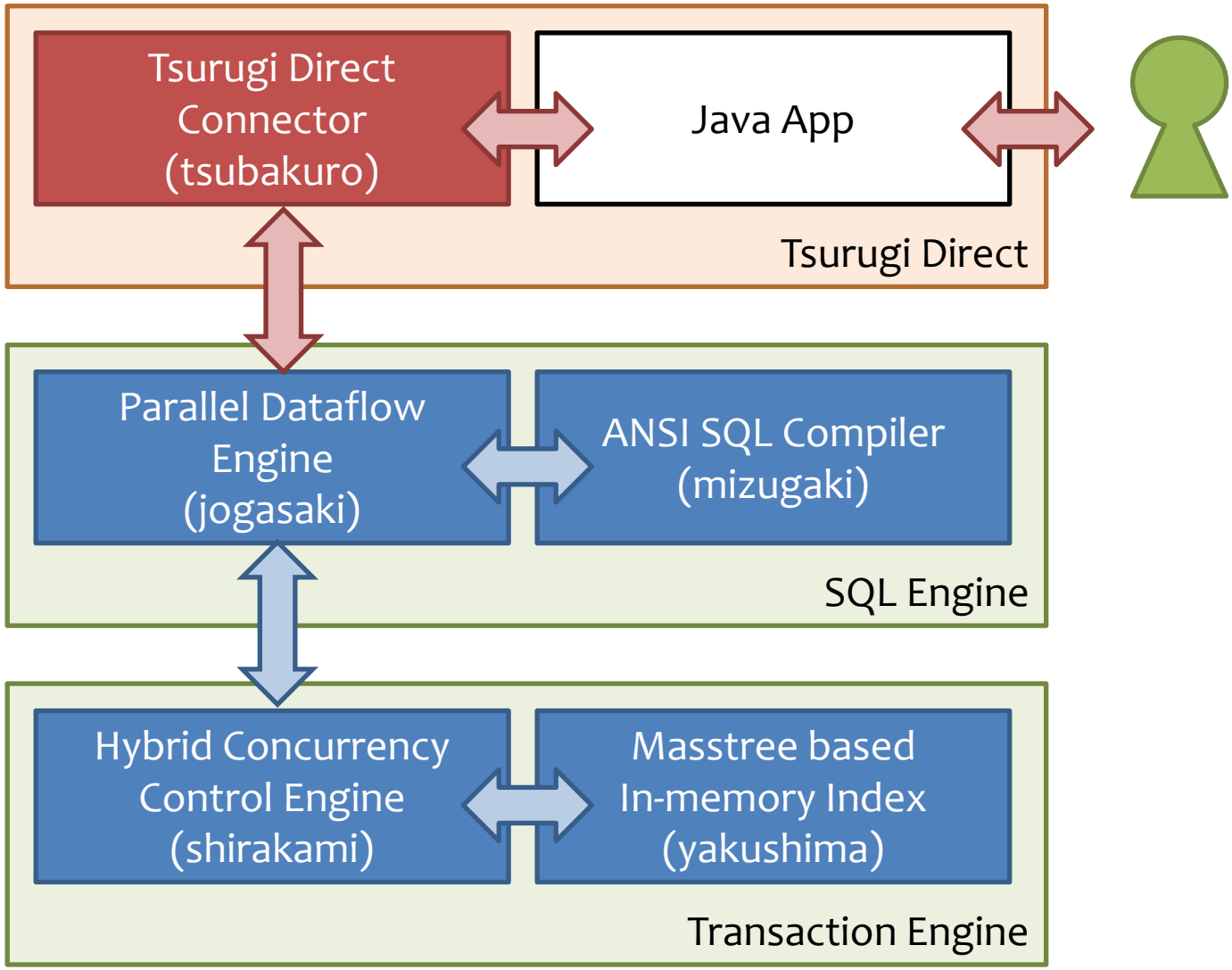
Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

別口でのアクセス (1) - Tsurugi Direct

- Tsurugi OLTPに直接アクセスするための機構
  - PostgreSQLをバイパスするため、オーバーヘッドが載らない
  - 独自のAPIを利用し、Tsurugi OLTPの全機能を活用可能
    - 同一トランザクションから複数のSQLを非同期に発行
    - 楽観方式のAbortを適切な方法で処理
    - 細やかなwrite preservationの制御や、並列数の制御など
- プログラムをJavaで記述し、UDFや外部プログラムとして実行
  - UDFの場合は共有メモリー方式、外部からはHTTP経由でTsurugiと通信
  - 独自APIのため、JDBCの資産は利用できない

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

Tsurugi Direct - アーキテクチャ図



Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

Tsurugi Direct API

- 2レベルのAPIを提供
  - 低水準API
    - 高速に動作するが、非同期プログラミングの知識を要する
    - 少し内部が透けて見える
  - 高水準API
    - 低水準APIをもとにした、やや低速だが利便性の高いAPI
    - Javaらしいプログラミングが可能
    - 気に入らなければ自作も可能
- 他言語バインディングについて
  - Pythonなど、他言語への移植も検討中
  - 通信路はJavaに依存しない方法で作成しているので、移植もそこまで面倒ではない
  - 手間の関係で来年度以降に計画中

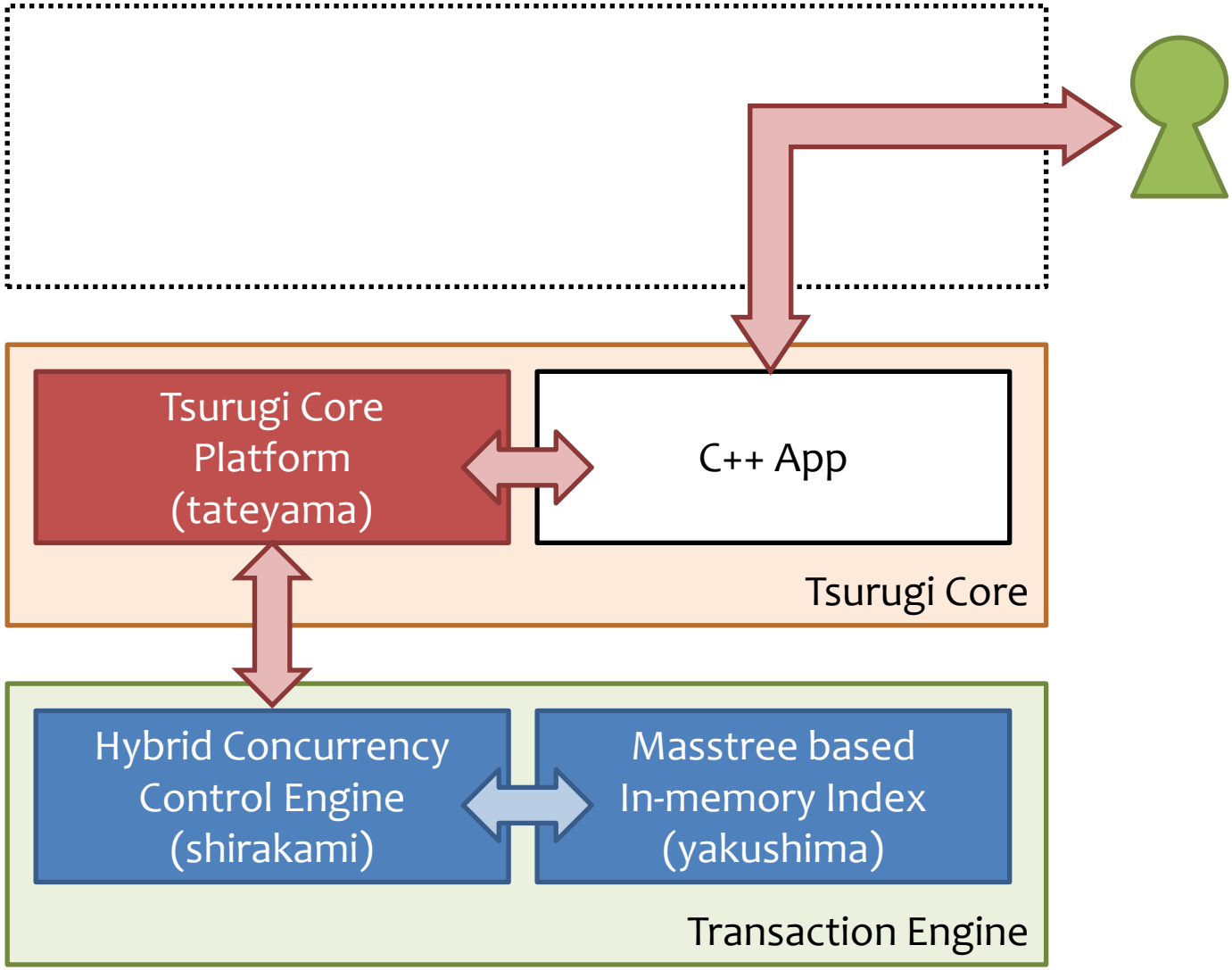
Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

別口でのアクセス (2) - Tsurugi Core

- Tsurugiのトランザクションエンジンを直接操作
  - SQLを利用せず、Key-Valueストアとしてアクセス
    - SQL実行エンジンをバイパスするため、限界に近い性能を出せる
  - C++でプログラムを書き、Tsurugi OLTP内で直接実行
    - SQL実行エンジンと同格のサービスとして、HTTP経由でアクセス
  - 特別な要求に対応するための機構
    - Tsurugi Direct以上の性能や、ハードウェア特有機能を利用する等
- 各アクセス方式は共存可能
  - どの方式でも共通のトランザクションエンジンを操作するため、データの一貫性は広く保たれる
  - Key-Valueストア(Tsurugi Core)で大量に書き込み、SQL (PostgreSQL) で分析などが可能

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

Tsurugi Core - アーキテクチャ図



Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

New SQL (?) としての Tsurugi OLTPの使い分け

- PostgreSQL
  - 既存ツールとの接続や、簡単な分析処理等に有利
  - プログラミングレスで行える
- Tsurugi Direct
  - 性能や、バッチ処理の細かな制御が必要になる場合
  - 独自のAPIを利用する必要はあるが、Javaで書ける (Python計画中)
- Tsurugi Core
  - 上記でどうしようもない場合に、最後の手段として
  - 細かなトランザクションエンジンへの理解が重要

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.

まとめ

- 現代的アーキテクチャ向けにOLTP処理を最適化
  - マルチコアに適した並行性制御方式を採用
  - 大容量メモリーを活かしたインデックスとSQL処理
- 短いオンライン処理と長いバッチ処理の共存
  - ハイブリッドな並行性制御により、現代的アーキテクチャの良さを殺さずに混在ワークロードを実現
- 様々な方式でアクセス可能
  - PostgreSQL, Tsurugi Direct/Core など、複数のアクセス方式を提供
  - データは共通しているため、目的に応じて使い分け

Copyright © 2021 Nautilus Technologies, Inc. All rights reserved.