

2024 年 10 月 13 日
データサイエンス C レポート

aaa

情報経営システム工学分野 B3
学籍番号 : 24336488
氏名 : 本間三暉

1 いま MySQL2024～MySQL の最新情報を語る

1.1 セミナーの内容の概略

MySQL8.4 が発表された。Innovation Release(MySQL9.X.X) という積極的に変化させていくバージョンと LTS Version(MySQL8.4.X) という 8 年間サポートされる安定バージョンがある。

LTS 版では同一メジャーバージョンではデータ形式の変更を行わないため、アップグレードやダウングレードが比較的行きやすい。

Innovation Release はいわゆる「やんちゃするバージョン」である。非互換の修正が含まれることもあり、サポート期間は次のバージョンまで (3 ヶ月) なので、保守していく労力がとんでもないので本番環境などではあんまりおすすめしない。

ここまでだと IR 版はデメリットばかりのように思えるが、最新機能にいち早く触れられたり、次の LTS の機能を知りたかったり、勇者におすすめである。とにかく IR 版はわくわくする。

Ver8.4.1 や 9.0.0 は特定の条件下でデータが壊れるという致命的な欠陥があったため取りやめられた。LTS は 8.4 や 9.7 である。

LTS 版への経緯としては MySQL8.0 のメリットである開発の高速化を取りたい一方、デメリットである不安定性という問題を解決するために LST 版と IR 版の 2 つのバージョンを作った。

最新の注意事項として、8.0 から 8.4 に上げた途端に認証できない状態となる事がある。mysql_native_password という古いパスワード形式を使っている場合、認証プラグインがデフォルトで無効化されるため、プラグインを有効にすれば接続できるようになる。これは非推奨のパスワード形式なので、早めに最新の形式にすべき。

MySQL の情報源として、MySQL 公式マニュアルやとみたまさひろ氏の「MySQL Parameters」、ツイッターや Bluesky などでは #masql_jp が挙げられる。

1.2 感想

サークル活動などでデータベース言語に触れる機会も多く、急遽バージョンを下げたりすることも多々あるので、SQL の詳しいバージョンの関係は覚えておきたい。

2 ビジネスを止めずにシステムリプレースを進める為の CDC という選択肢

2.1 セミナーの内容の概略

システムリプレースとは、既存のシステム (サービス) やソフトウェアを新しいシステムやソフトウェアに置き換えることを指す。システムリプレースの手法として、徐々に新システムに移行するフェーズアプローチと一度にすべてのシステムを新しいものに切り替えるビックバンアプローチがある。システムの規模感によって手法を選択する。今回はフェーズアプローチを主に紹介する。

サービス開始から複数年が経過して、コードベースが大きい。作った人が退職し、なぜこの機能があるのかわからない。機能リリース優先で追加されたため同修正していいかわからないなどがある。

開発の限られた氏リプレースをすべてシステムリプレースを割り当てることは出来ない。なので、一部ずつ切り替えながらやるフェーズアプローチを選ぶ。

なんのためにシステムリプレースするのか。

フレームワークのバージョンを上げる、TypeScript で書き直すなどいろいろあるが、きれいな旧システムを作ってしまう可能性がある。

例えば、クーポンには審査の概念があり、承認されたクーポン飲みを使うことができるというコードがあったとき、ただ移植するなら簡単だが、ヒアリングすると現在は承認の確認等はないという。

CSC の紹介。フェーズアプローチは一部ずつ切り替えながら進めていく手法なので旧システムとリプレースさせた新システムが並行稼働する。データベースをリモデリングしたいけどみたくないとき、CDC：Change Data Capture と呼ばれる、データベースの変更を検知し、別のアプリケーションにイベントが発生したことを伝える。トリガーベースのパターンとログベースのパターンがある。

トリガーベースの CDC は名著『データベース・リファクタリング』でも紹介されている手法。メリットとしてはほとんどの DBMS で利用可能で、デメリットとしてデータベースに追加の負荷がかかる。また、SQL で書く必要がある。

ログベースの CDC は DB のトランザクションログを直接読み取る手法。メリットとして、処理を別アプリケーションに切り出すことができるので追加のオーバーヘッドが発生しない。デメリットとして、各ツールの運用をキャッチアップする必要がある。

AWS DMS を使う方法と Debezium を使う方法がある。Apache Kafka

運用フローを一度見返す過程でユーザが今まで抱えていた問題を解決できるいい機会となる。

2.2 感想

この発表は、システムリプレースの現実的な課題と方法論について実用的な知見を提供している。フェーズアプローチによる旧システムと新システムの並行稼働や CDC の活用で、段階的な移行と運用改善の可能性が感じられた。

3 『LEADING QUALITY』から考えるソフトウェア品質とビジネス価値

3.1 セミナーの内容の概略

数年でハードな競争環境になってしまった。

品質とは、ある時点でそれが問題となる誰かにとっての価値である。エンジニアとして生きていくなれば、”品質”とは切り離せない関係である。

品質は絶対的なものと相対的なものがある。品質という話をする際に切り離せないテストの話をしていない。品質の専門家がテストのことしか知らないということが日本ではある。

ビジネス価値と企業の関係。

虚栄の評価指標：成果につながっているようで、行動に繋がりにくく、ビジネスの成功とも関連しない数値。

意味のある学び。

そもそも OSC で営利企業の話をするのかに対して、文化とは組織のオープンソースである。という結論を出していた。

文化が価値を持つのは交換しても価値が減らないからである。

戦略と文化は組織の両輪である。

3.2 感想

ゴールデン・サークルで深掘りをしていたりしてとても良かった。

ゴールデン・サークルに加えて How-to と Who を付け足した説明が説得力があってとても良かった。

ザ・ゴールを読むべきらしいので今度読んでみようと思う。

4 再: 自宅サーバーを始めてみよう!2024 年版

4.1 セミナーの内容の概略

インフラに関わるなら IPtables だけでは足りない。NFtables もやらないといけない。CentOS 10 Stream を触っていると、Intel x86_64 v3 以上でないと boot すらしない。

自宅サーバーを始めてみよう！とはいえ、Linux を使うといずれ不幸せになるかもしれない。

セキュリティのお話。ログを監視しながら fail を連発する IP を自動的に BAN する。iptables, ip6tables とかの Ban4ipd を使う。

4.2 感想

この発表では、Linux 自宅サーバー運用におけるセキュリティ強化の必要性が強調されている。特に、NFtables を活用し、ログ監視と自動 BAN 機能で不正アクセス対策を講じる重要性を実感した。

5 Docker + GPU で生成 AI に挑戦してみよう！

5.1 セミナーの内容の概略

高火力 DOK はここが便利！

- 実行時に CPU を選べる。
- Docker イメージを実行できる。
- 使った時間だけ課金
- API で自動化

現状だと、インタラクティブな処理には向いておらず、外部接続機能が近日提供予定である。

- アプリケーションを Docker イメージ化
- コンテナレジストリーへ登録
- 高火力 DOK で実行！
- 成果物を回収

オープンソース要素どこ行った.....? 今すぐ使える高火力 DOK のイメージ例 <https://github.com/shimataro/dok-example>

5.2 感想

高火力 DOK の利便性と柔軟性が強調されており、特に CPU 選択や課金体系が実用的で魅力的だ。一方で、インタラクティブな処理やオープンソース要素の不足が課題として感じられる内容だった。

参考文献

- [1] 10/12 Open Source Conference 2024 Nagaoka<https://ospn.connpass.com/event/325425/>