

ORM, Connection Пулы, Индексация, Миграции



Hibernate VS Mybatis



АКАДЕМИЯ

Преимущества и недостатки Hibernate



Плюсы

- Поддерживается spring'ом из коробки.
- Простая конфигурация.
- Хорошая абстракция
- Множество различных классов, хелперов, билдеров и т.д.
- Можно быстро написать проект
- Большое сообщество
- Есть кеширование



Минусы

- Тяжеловесный.
- Тяжело поддерживаемый (легко поддерживать только в небольших проектах).
- Сложен в использовании в больших проектах.
- Performance по сравнению с Mybatis меньше.
- Магия, магия и еще раз магия.

Преимущества и недостатки MyBatis



Плюсы

- Удобен в конфигурации.
- Прост в использовании.
- Удобен при сложной структуре базы данных.
- Только разработчик отвечает за то как быстро будет работать приложение.
- Множество вспомогательных плагинов.
- Performance больше чем у Hibernate в больших проектах.



Минусы

- Нет кеширования из коробки.
- Почти весь sql код придется писать в xml
- Сложен в использовании в больших проектах.
- Документация.
- Хороший плагин-платный

Что лучше подходит для вашего проекта

- Если у вас много простых объектов без сложных связей друг с другом – **Hibernate**
- Если у вас сложные запросы к базе на агрегацию данных – **MyBatis**
- Если вы готовы поручить фреймворку создавать и обновлять для вас схему – **Hibernate**
- Если у вас работа с хранимыми процедурами – **MyBatis**
- Если вы не хотите писать запросы сами, и работать с JPQL – **Hibernate**
- Если вы хотите писать свой ORM (а его так или иначе придётся писать) – **MyBatis**



VS



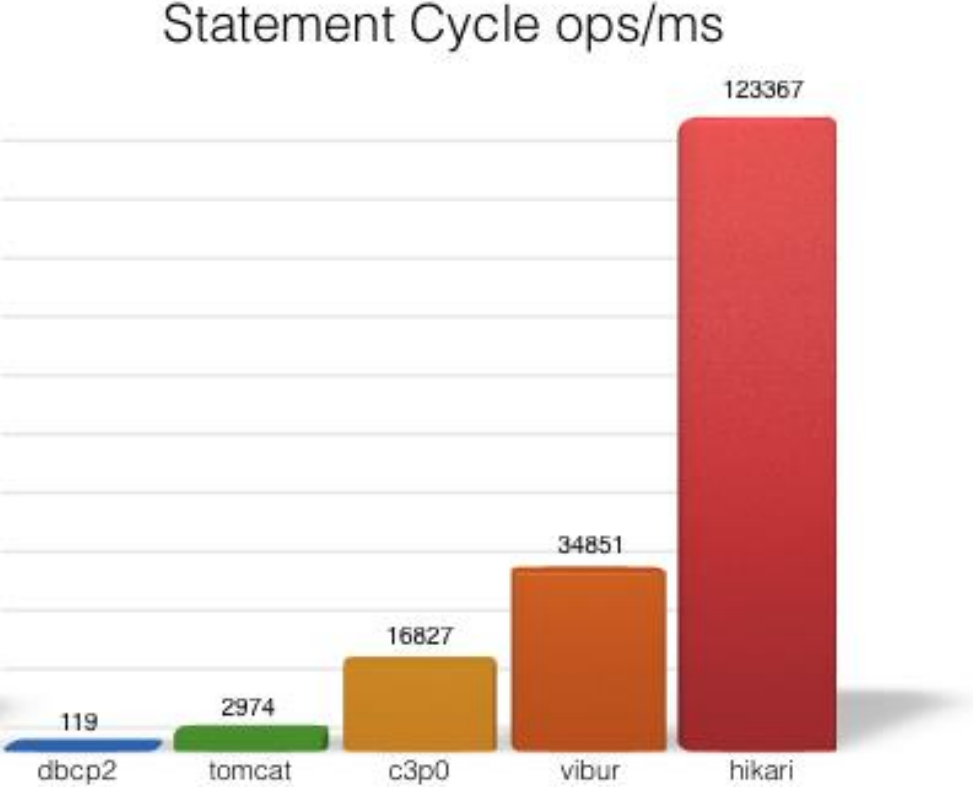
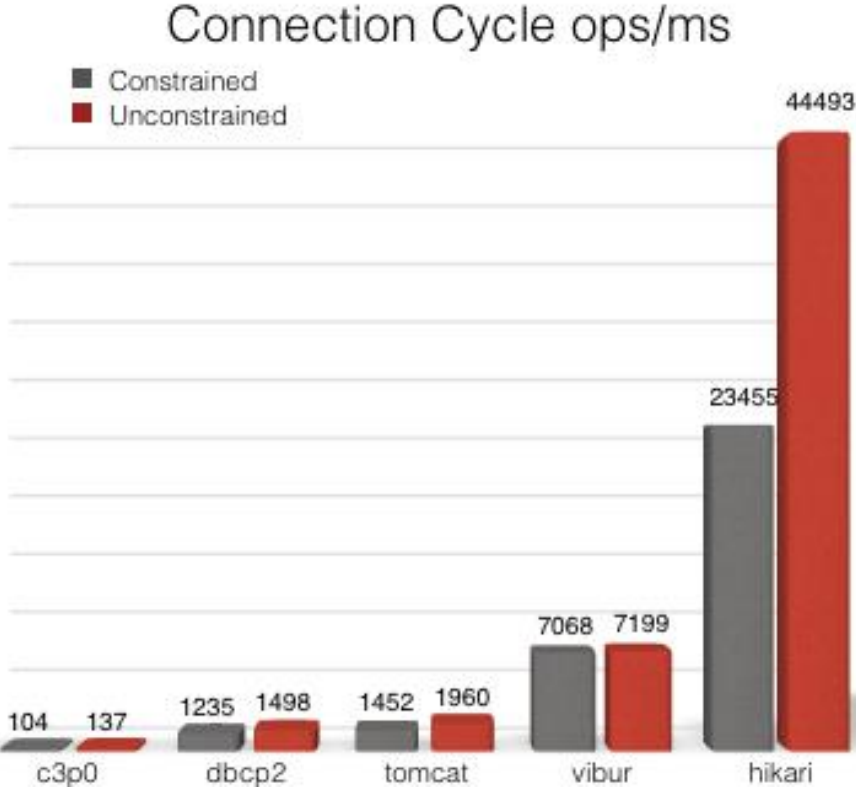
HikariCP

Самый быстрый пул соединений на java



АКАДЕМИЯ

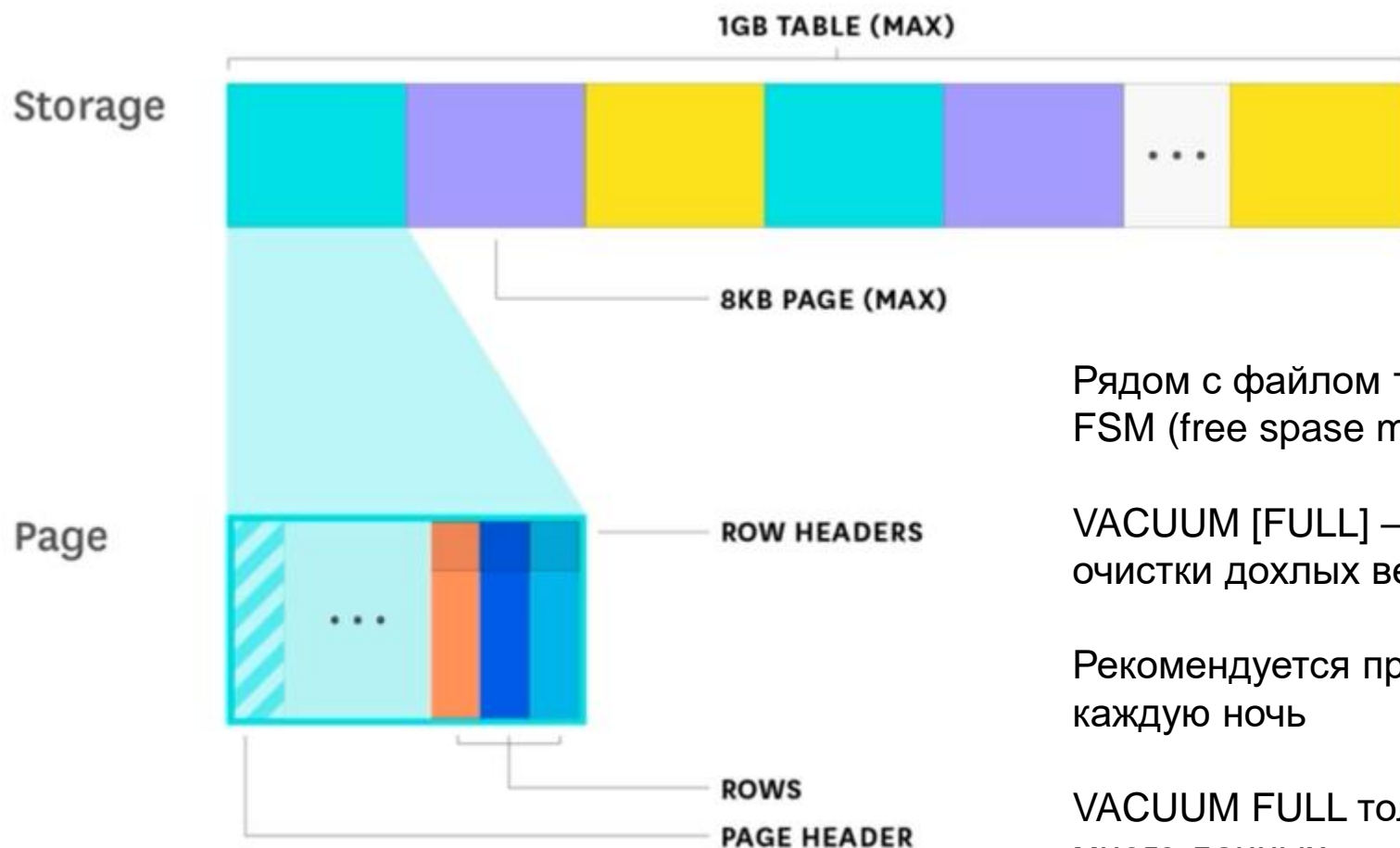
HikariCP



Посмотрим конфигурацию + примеры MyBatis



Устройство таблиц



Рядом с файлом таблицы лежит файл FSM (free space map)

VACUUM [FULL] – команда для очистки дохлых версий строк

Рекомендуется проходить VACUUM каждую ночь

VACUUM FULL только если удалили много данных

Индексы

- Индекс – объект БД, который можно создать и удалить.
- Позволяет искать значения без полного перебора.
- Служит для оптимизации выборки «небольшого» числа записей.
- «Небольшое число» – число относительно общего количества записей в таблице.
- По Primary Key b Unique столбцам индекс создается автоматически.
- Индексы не бесплатны.



SELECT amname FROM pg_am;

- B-tree (сбалансированное дерево)
- Hash-индекс
- GIST (обобщенное дерево поиска)
- GIN (обобщенный обратный)
- SP-GIST (GIST с двоичным разбиением)
- BRIN (блочно-диапазонный)

Методы сканирования

- Индексное (index scan)
- Исключительно индексное сканирование (index only scan)
- Сканирование по битовой карте (bitmap scan)
- Последовательное сканирование (sequential scan)

Balanced Tree

- Создается по умолчанию (CREATE INDEX index_name ON table_name (column_name))
- Поддерживает операторы: <, >, <=, >=, =
- Поддерживает LIKE 'abc%' (но не '%abc')
- Индексирует NULL
- Сложность поиска $O(\log N)$

Hash

- `CREATE INDEX index_name ON table_name USING HASH (column_name);`
- Поддерживает только оператор “=“
- Не отражается в журнале предзаписи (WAL)
- Не рекомендуется к применению(но это не значит что его не нужно применять)
- Сложность поиска $O(1)$

Специализированные индексы

GIST (обобщенное дерево поиска)

- Для индексации геометрических типов данных, полнотекстового поиска

GIN (обобщенный обратный)

- Jsonb, полнотекстовый поиск, диапазоны

SP-GIST (GIST с двоичным разбиением)

- Для данных которые подразумевают естественную упорядоченность

BRIN (блочно-диапазонный)

- Для огромных объемов данных которые подразумевают естественную упорядоченность

Как решить проблему?

Если есть проблема – надо понять откуда растут ноги

1) EXPLAIN

- EXPLAIN query позволяет посмотреть на план выполнения запроса
- EXPLAIN ANALYZE query прогоняет запрос, показывает план и реальность

2) ANALYZE

- Собирает статистику по данным таблицы
- Планировщик смотрит на статистику при построении запроса
- ANALYZE [table_name [(column1, column2...)]]
- Запускать как минимум один раз в день
- Avtovacuum в том числе запускает ANALYZE

Посмотрим на практике ускорение работы



Миграции



АКАДЕМИЯ

Какие мы используем

- **Flyway** – это инструмент с открытым исходным кодом, работающий под лицензией Apache License 2.0, который помогает выполнять автоматические миграции баз данных на основе версий.



- **Liquibase** – это библиотека, которая предоставляет набор инструментов для реализации ручной и автоматической миграции базы данных.
- **Преимущества** - последовательное выполнение всех необходимых обновлений для получения самой последней версии базы данных, автоматическое создание и выполнение операций отката для неудачных обновлений и независимое от базы данных определение операций обновления.

ChangeSet

ChangeSet – единица манипуляции Liquibase у которой есть определенный набор параметров и настроек. Список всех изменений состоящий из множества changeSet-ов отслеживается в databasechangelog

Основные команды:

update – обновляет базу данных на основании логов.

updateSQL – прогоняет скрипт обновления без непосредственного обновления.

rollback – откатывает изменения на определенную дату, тэг или количество changeSets.

rollbackSQL - прогоняет отката без непосредственного отката.

status – показывает состояние миграции (количество потенциальных сетов, ошибки и тд).

changelogSync – синхронизирует существующие changeset с логами в changelog.

diff – показывает разницу реализованных и потенциальных изменений БД.

clearChecksums – удаляет хэши из файла changelog.

history – история операций и запущенных миграций.

Основные атрибуты `changeSet`

`id` – уникальный идентификатор (обязательный атрибут)

`author` – псевдоним, логин или другое имя автора (обязательный атрибут)

`dbms` – тип базы данных (или несколько) для которых предназначен

`runAlways` – использует `changeset` при каждом запуске, даже если он был использован ранее

`runOnChange` – использует `changeset` если он был изменен

`contexts` – задает контексты `changeset` для возможности запуска конкретных групп

`labels` – задает лэйблы для возможности запуска конкретных групп

`ignore` – не выполняет помеченный `changeset`

Возможные вложенные тэги

- `comment` – комментарий к `changeset`
- `preConditions` – условия необходимые для использования сета
- `validChecksum` – явно указанная валидная хэш сумма
- `rollback` – явно прописываемый скрипт отката (в том или ином виде)

Домашнее задание

1. Написать свои репозитории (с контроллерами и сервисами) для других таблиц разными способами. Надо чтоб работало.
2. Написать свой changeset по аналогии с представленным для создания базы данных.

* Подключить логирование к mybatis и liquibase



Всем спасибо!

Лига –
лучший
старт
карьеры!

Мы в Лиге!

Умножай
знания –
верь в мечту!

Каждый
день – новый
челлендж!



Владислав Сыров

Разработчик



АКАДЕМИЯ