Использование базы данных Mnesia в чат-сервере

Юра Жлоба

Wargaming.net

Май 2019

Mnesia

Распределенная key value база данных,

встраиваемая в Erlang приложения.

Mnesia

Почему ее не рекомендуют использовать?

И почему все же используют?

Как она используется в чат-сервере?

Amnesia

1999 год

Изначально база данных называлась Amnesia.

Это название не понравилось кому-то из менеджмента.

”So we dropped the A, and the name stuck.” Joe Armstrong.

Amnesia

Традицию продолжила компания WhatsApp.

Они назвали свою БД

ForgETS.

Фичи

- Работает внутри эрланговской ноды,
  не нужно передавать данные по сети.

Фичи

- Работает внутри эрланговской ноды,
  не нужно передавать данные по сети.
- Хранит данные нативно (Erlang term),
  не нужно сериализовать/десериализовать данные.

Фичи

- Работает внутри эрланговской ноды,
  не нужно передавать данные по сети.
- Хранит данные нативно (Erlang term),
  не нужно сериализовать/десериализовать данные.
- Хранит данные в ETS/DEST таблицах,
  чтение и запись работают очень быстро.

Фичи

Работает внутри эрланговского кластера.

Данные доступны отовсюду в кластере (сетевая прозрачность).

Полная реплика данных на каждой ноде.

Фичи

- Транзакции (ACID).
- Вторичные индексы.
- Миграции (структуры таблиц и данных).
- Шардинг (fragmented tables).

API

- Базовые KV операции:
  read, write, delete.

API

- Базовые KV операции:
  read, write, delete.
- ETS/DETS API:
  lookup, match, select.

API

- Базовые KV операции:
  read, write, delete.
- ETS/DETS API:
  lookup, match, select.
- Fold:
  foldl, foldr.

API

- Базовые KV операции:
  read, write, delete.
- ETS/DETS API:
  lookup, match, select.
- Fold:
  foldl, foldr.
- QLC
  Query List Comprehension.

Query List Comprehension

```
qlc:q([X || X <- mnesia:table(shop)])
```

```
qlc:q([
    Xshop.item || X <- mnesia:table(shop),
    Xshop.quantity < 250
])
```

```
qlc:q([
    Xshop.item ||
    X <- mnesia:table(shop),
    Xshop.quantity < 250,
    Y <- mnesia:table(cost),
    Xshop.item =:= Ycost.name,
    Ycost.price < 2
])
```

Транзакции

Синхронные и ”обыкновенные”.

Pessimistic locking.

Медленные.

Но без них нет консистентности данных.

Консистентность данных

- Транзакции работают через 2PC.

Консистентность данных

- Транзакции работают через 2PC.
- Strict quorum protocol,
  все ноды должны подтвердить транзакцию.

Консистентность данных

- Транзакции работают через 2PC.
- Strict quorum protocol,
  все ноды должны подтвердить транзакцию.
- Гибкие настройки репликации,
  можно явно указать, на каких нодах и как хранить данные.

Консистентность данных

- Транзакции работают через 2PC.
- Strict quorum protocol,
  все ноды должны подтвердить транзакцию.
- Гибкие настройки репликации,
  можно явно указать, на каких нодах и как хранить данные.
- Неплохо переживает рестарты нод в кластере.

Консистентность данных

- Транзакции работают через 2PC.
- Strict quorum protocol,
  все ноды должны подтвердить транзакцию.
- Гибкие настройки репликации,
  можно явно указать, на каких нодах и как хранить данные.
- Неплохо переживает рестарты нод в кластере.
- Плохо переживает network partition.

С точки зрения CAP теоремы

- Это CA система,
  не устойчива к network partition (P).

С точки зрения CAP теоремы

- Это CA система,
  не устойчива к network partition (P).
- Можно повысить A отказавшись от C.
  dirty mode, без транзакций.

С точки зрения CAP теоремы

- Это CA система,
  не устойчива к network partition (P).
- Можно повысить A отказавшись от C.
  dirty mode, без транзакций.
- А если мне вообще не нужна репликация на несколько нод?
  Тогда просто бери ETS/DETS.

Репутация Mnesia

Мнение широко известных в узких кругах авторитетов.

Печальный опыт с персистентными очередями в RabbitMQ.

Слухи из Стокгольма от местных разработчиков.

Репутация Mnesia

Суть проблемы в том,

что если нода не была корректно остановлена, а упала,

то восстановление большой таблицы с диска может занять часы.

Репутация Mnesia

Downtime сервиса может длится несколько часов!

На этом про Mnesia можно было бы забыть и не вспоминать,

но ...

Применение Mnesia

Но её можно применить с пользой.

Задача

Кластер из нескольких эрланг-нод.

Нужно хранить пользовательские сессии,

так, чтобы они были доступны во всех нодах кластера.

Прежнее решение

- Сессии хранятся в MySQL.

Прежнее решение

- Сессии хранятся в MySQL.
- Данные консистентны и доступны все нодам.

Прежнее решение

- Сессии хранятся в MySQL.
- Данные консистентны и доступны все нодам.
- Latency больше, чем могло бы быть.

Задача

- Конечно, хочется иметь эту инфу прямо в ноде.

Задача

- Конечно, хочется иметь эту инфу прямо в ноде.
- Кешировать в ETS?

Задача

- Конечно, хочется иметь эту инфу прямо в ноде.
- Кешировать в ETS?
- Хорошо, а как обновить этот кэш на всех нодах?

Задача

- Конечно, хочется иметь эту инфу прямо в ноде.
- Кешировать в ETS?
- Хорошо, а как обновить этот кэш на всех нодах?
- Вот если бы был распределенный кэш ...

Задача

- Конечно, хочется иметь эту инфу прямо в ноде.
- Кешировать в ETS?
- Хорошо, а как обновить этот кэш на всех нодах?
- Вот если бы был распределенный кэш ...
- Постойте-ка, а Mnesia – это что?

Mnesia не вызывает проблем, если:

- Не нужно персистентное хранение данных.
- Не нужны сложные запросы с транзакциями.
- Данные относительно дешево реплицируются.

С Mnesia будут проблемы, если:

- Нужно хранить много данных.
- Нужно хранить их персистентно.
- Объем данных постоянно растет.
- Выполняются сложные запросы к данным.

Применение Mnesia

Все это – типичные сценарии использования типичной БД.

И все это – плохо для Mnesia.

Применение Mnesia

Идеальный сценарий для Mnesia:

in-memory хранение пользовательских сессий.

Применение Mnesia

В такой роли ее используют:

WhatsUp (на ранних этапах),

League of Legends Chat,

Discord,

Ejabberd.

Применение Mnesia

Mnesia – это не БД, это кэш :)

Еще раз про ключевые преимущества

- Данные прямо в памяти ноды,
  за ними не надо ходить по сети.
- Данные в нативном виде,
  их не надо сериализовать/десериализовать.
- Прозрачная репликация на все ноды кластера.

Что важно для нас

- Mnesia неплохо переживает рестарты отдельных нод в кластере.
- Потому что мы именно так обновляем кластер.
- Но нужно знать объем данных и время их репликации.
- Это этого зависит время downtime ноды при рестарте.

Вопросы?