

複数DBとRails



Kohei Suzuki (@eaglemt)

Cookpad Inc.

アウトライン

- ◆ 複数 DB を Rails から扱う
 - ◆ DB 分割
 - ◆ r/w splitting
- ◆ 複数 DB トラブル
- ◆ 最近の悩み

複数DB

- ◆ Rails のバージョンアップ (3.2 -> 4.1) を担当した
- ◆ 特に ActiveRecord の変更に対応していたら Rails における複数 DB と向き合うことに……

複数DBの種類

- ◆ いくつか種類がある
 - ◆ 単にテーブルが別の DB に存在している (DB 分割)
 - ◆ master-slave 構成で r/w splitting している
 - ◆ シャーディング
- ◆ シャーディングは分からないので、上2つについて

DB分割

- ❖ `establish_connection` で接続先を変える

```
class Db1Model1 < ActiveRecord::Base  
  establish_connection :"#{Rails.env}_db1"  
end
```

```
class Db1Model2 < ActiveRecord::Base  
  establish_connection :"#{Rails.env}_db1"  
end
```

DB分割

- ◆ これでもできるけど、コネクション数が急増する
- ◆ app は大丈夫でも DB が耐えられない

ARの接続

- ◆ Model.connection は、クラス階層を上りながら establish_connection している最初のモデルクラスの接続プールを使う
- ◆ 単一 DB のときは ActiveRecord::Base のみ
- ◆ すべてのモデルが ActiveRecord::Base の接続プールを使う

ARのコネクション

- ◆ 各モデルでそれぞれ `establish_connection` すると、モデルの数に比例したコネクションが作られる
- ◆ 数百のモデルがあると現実的ではない

コネクション共有

- ❖ `establish_connection` するだけの抽象クラスを定義する

```
class Db1Base < ActiveRecord::Base
  self.abstract_class = true
  establish_connection :"#{Rails.env}_db1"
end
```

コネクション共有

- ◆ db1 にあるテーブル (モデル) は Db1Base を継承するようにする

```
class Db1Model1 < Db1Base  
end
```

```
class Db1Model2 < Db1Base  
end
```

コネクション共有

- ◆ コネクションプールのために基底クラス Db1Base を用意する
- ◆ 継承した Db1Model1, Db1Model2 は Db1Base のコネクションプールを使う

r/w splitting

- ◆ 接続先を切り替えたいときに `establish_connection` する
- ◆ メリット
 - ◆ Rails 標準の機能だけでできる
- ◆ デメリット
 - ◆ `establish_connection` の度に接続し直すことになる
 - ◆ 頻繁な切り替えには向かない

r/w splitting

```
class SummaryBase < ActiveRecord::Base
  self.abstract_class = true
  establish_connection :("#{Rails.env}_summary")

  def self.with_master
    establish_connection :("#{Rails.env}
_summary_writable")
    yield
    ensure
      establish_connection :("#{Rails.env}_summary")
    end
  end
end
```

r/w splitting

- ◆ コネクションを維持しつつ、頻繁に切り替えたい
- ◆ モデルがコネクションプールを1種類しか持てないので、ルールから外れなければならない
- ◆ できるだけ標準的な方法で実現するために switch_point という gem を作成して移行
- ◆ https://github.com/eagletmt/switch_point

r/w splitting

```
SwitchPoint.configure do |config|  
  config.define_switch_point :main,  
    readonly: : "#{Rails.env}_slave"  
  
  config.define_switch_point :bargain,  
    readonly: : "#{Rails.env}_bargain_slave",  
    writable: : "#{Rails.env}_bargain"  
  
  ...  
end
```

r/w splitting

```
class MainBase < ActiveRecord::Base  
  self.abstract_class = true  
  use_switch_point :main  
end
```

```
class BargainBase < ActiveRecord::Base  
  self.abstract_class = true  
  use_switch_point :bargain  
end
```

```
class User < MainBase; end  
class Recipe < MainBase; end
```

```
class Bargain::Shop < BargainBase; end  
class Bargain::Leaflet < BargainBase; end
```

r/w splitting

```
User.with_readonly { User.find(id) }
```

```
Recipe.with_writable do  
  Recipe.create(recipe_params)  
end
```

複数DBトラブル

DBを跨ぐJOIN

- ◆ モデル A、B が別の DB にある
- ◆ A と B は JOIN できない
- ◆ しかし、includes を使ったときに特定の状況下では ActiveRecord によって JOIN が使われてしまう

DBを跨ぐJOIN

```
class A < ActiveRecord::Base  
end
```

```
class B < ActiveRecord::Base  
  belongs_to :a  
end
```

```
irb(main):001:0> B.includes(:a).pluck(:id)  
(0.2ms) SELECT `bs`.`id` FROM `bs`  
        LEFT OUTER JOIN `as` ON `as`.`id` = `bs`.`a_id`
```

DBを跨ぐJOIN

```
class A < ActiveRecord::Base  
end
```

```
class B < ActiveRecord::Base  
  belongs_to :a  
end
```

```
irb(main):001:0> B.preload(:a).pluck(:id)  
(0.2ms) SELECT `bs`.`id` FROM `bs`
```

database_rewinder

- ❖ database_cleaner に代わる gem
- ❖ @amatsuda さんが作った
- ❖ *This strategy was originally devised and implemented by Shingo Morita (@eudoxa) at COOKPAD Inc.*
- ❖ https://github.com/amatsuda/database_rewinder#credit

database_rewinder と複数DB

- ❖ 複数 DB 環境で最初の DatabaseRewinder.clean_all がうまく動かない問題
 - ❖ ActiveRecord::Base 以外のコネクションをうまくとれてない
- ❖ 最終的に v0.4.1 で解決
- ❖ https://github.com/amatsuda/database_rewinder/compare/v0.3.0...v0.4.1
- ❖ *TODO: Write documentation and specs about multiple databases.*

database_rewinderと複数DB

- ◆ 複数の DB が異なる adapter を使っているときにテーブルが消されない問題
- ◆ MySQL + PostgreSQL など

database_rewinder と複数DB

- ❖ database_rewinder の初期化時にはまだ別 DB の adapter が読み込まれていないことが原因
- ❖ https://github.com/amatsuda/database_rewinder/pull/21

最近の悩み

複数DB + 複数アプリケーション

- ◆ 複数のアプリケーションがモデルを共有している
 - ◆ cookpad.com 本体
 - ◆ Web API サーバ
 - ◆ 管理画面
 - ◆ バッチ

複数DB + 複数アプリケーション

- ◆ すべてのアプリケーションがすべての DB に接続するわけではない
- ◆ 管理画面からしか使われない DB
- ◆ 管理画面とバッチからしか使われない DB
- ◆

複数DB + 複数アプリケーション

- ◆ モデルは各アプリケーションで共有されている
- ◆ 一方で、そのアプリケーションから実際には使われない DB の接続設定を `database.yml` に書きたくない
- ◆ 例えばバッチからしか使われない DB を更新した時に、Web API サーバにも `database.yml` を配布する必要があるような状況は避けたい

複数DB + 複数アプリケーション

- ❖ モデルが共有されているので、起動時に `database.yml` に書かれていない DB への `establish_connection` が実行されてしまう
- ❖ `ActiveRecord::AdapterNotSpecified` を握り潰すモンキーパッチ!!!
- ❖ 非常につらい…… 何か良い方法があれば是非