



Ruby on Rails環境構築

hajime.miyauchi@gmail.com

内容

- はじめに
- 目的
- 今回構成する環境
- Ruby/RubyGems/Railsのインストール
- Webアプリサーバのインストールと設定
- リバースプロキシのインストールと設定
- 最後に

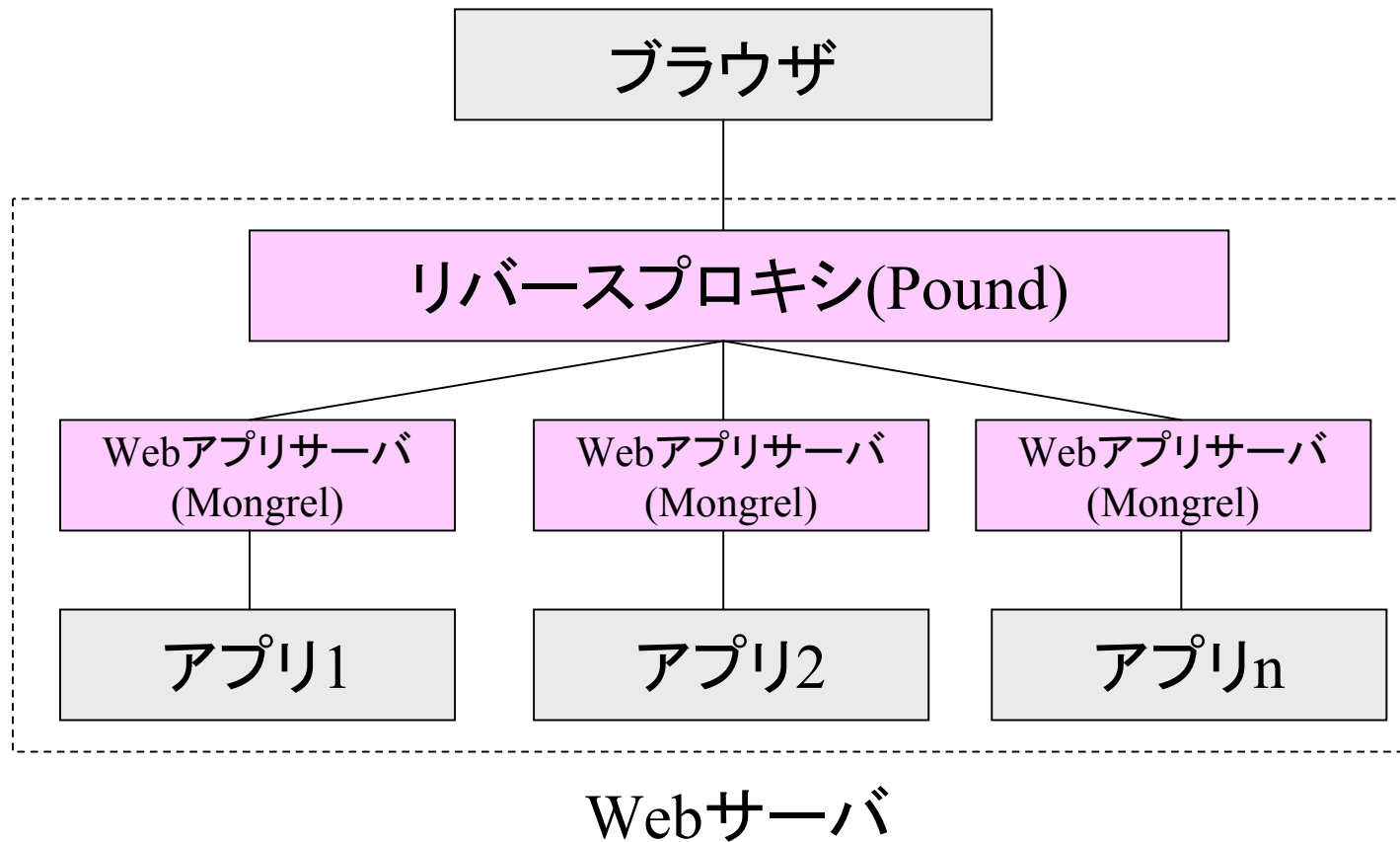
はじめに

- Ruby on RailsはWebアプリサーバ同梱。
- なのでこんな感じで起動します。
 - `$ cd /home/hajime/demo_app`
 - `$ ruby script/server`
- Webアプリサーバが起動。
- 公開完了。
 - `http://localhost:3000/`
- 超簡単。Apacheとか要りません。

目的

- 複数アプリを同じポート(80番)で動かしたい。
 - http://demo.example.com/
 - http://demo2.example.com/
- サーバ起動時に自動起動させたい。
 - `$ /etc/init.d/rails start`
 - みたいに。
- もっと高速に
 - 同梱のWebアプリサーバ(WEBrick)はPureRuby
 - 実装は美しいが速度優先ではない。

今回構築する環境



今回使用するソフト

- OS
 - CentOS 5.2
- Webアプリケーションサーバ
 - Mongrel
 - Mongrel Cluster
- リバースプロキシサーバ
 - Pound

Rubyのインストール

```
$ yum install zlib-devel openssl-devel  
$ wget ftp://ftp.ruby-lang.org/pub/ruby/1.8/ruby-1.8.7-p72.tar.gz  
$ tar xvfz ruby-1.8.7-p72.tar.gz  
$ cd ruby-1.8.7-p72  
$ ./configure  
$ make  
$ sudo make install  
$ sudo ln -s /usr/local/bin/ruby /usr/bin/
```

RubyGemsのインストール

```
$ wget  
  http://rubyforge.org/frs/download.  
  php/45905/rubygems-1.3.1.tgz  
$ tar xvfz rubygems-1.3.1.tgz  
$ cd rubygems-1.3.1  
$ sudo ruby setup.rb
```

Railsのインストール

```
$ sudo yum install sqlite-devel
```

```
$ sudo gem install rails
```

```
$ sudo gem install sqlite3-ruby
```

デモアプリを作ってみる

```
$ cd /var/www/
```

```
$ rails demo ← 作成
```

```
$ cd demo
```

```
$ ruby script/server ← 起動
```

→ <http://localhost:3000/>

にアクセスできるようになった。

Webアプリサーバ のインストールと設定

- Mongrel/MongrelClusterを使用
- Mongrelとは
 - 高速が謳い文句のWebアプリサーバ
- MongrelClusterとは
 - Mongrelのプラグイン
 - 単一アプリを複数のMongrelプロセスで起動 (クラスタ化)
 - 複数アプリをOS起動時に自動起動可能

Mongrel/MongrelCluster のインストール

```
# gem install mongrel
# gem install mongrel_cluster
# ln -s /usr/local/bin/mongrel_rails
  /usr/bin
# ln -s
  /usr/local/bin/mongrel_cluster_ctl
  /usr/bin
```

→ リンク貼ってパス通しておきます。

Mongrel/MongrelCluster のインストール(2)

```
# cd /usr/local/lib/ruby/gems/1.8/  
# cd gems/mongrel_cluster-1.0.5/  
# cd resources/  
# cp mongrel_cluster /etc/init.d/
```

→起動スクリプトを設置します。

Mongrel/MongrelCluster のインストール(3)

```
# chmod 755  
  /etc/init.d/mongrel_cluster  
# chkconfig --level 345  
  mongrel_cluster on
```

→ランレベル3、4、5で
自動起動するよう設定

Mongrelの起動ユーザ作成

```
# groupadd mongrel
```

```
# useradd -g mongrel mongrel
```

Mongrel Cluster用の ディレクトリを作成

```
# mkdir /etc/mongrel_cluster  
# chown mongrel.mongrel  
  /etc/mongrel_cluster  
# mkdir /var/run/mongrel_cluster  
# chown mongrel.mongrel  
  /var/run/mongrel_cluster
```

デモアプリの 設定ファイルを作成

```
# cd /var/www/demo ←アプリディレクトリに移動
# mongrel_rails cluster::configure ¥
--environment=development ¥
--port=3000 ¥
--num-servers=3 ¥
--log=/var/www/demo/log/production.log ¥
--pid=/var/run/mongrel_cluster/demo.pid ¥
--chdir=/var/www/demo ¥
--user=mongrel ¥
--group=mongrel
```

デモアプリを自動起動に追加

```
# chown mongrel.mongrel -R  
/var/www/demo  
# ln -s  
/var/www/demo/config/mongrel_cluster.yml  
/etc/mongrel_cluster/demo.yml  
→ 自動起動させるためにシンボリックリンク  
を張ります。
```

Mongrel Cluster起動！

```
# /etc/init.d/mongrel_cluster start
```

```
http://localhost:3000/
```

```
http://localhost:3001/
```

```
http://localhost:3002/
```

3プロセス起動。

リバースプロキシの インストールと設定

- Pound
 - リバースプロキシ/ロードバランサ
- Slashdot.orgで使われてました。
- 80番ポートで待ち受けて、下位のMongrel達に振り分けます。
- 振り分け条件はドメインやパス等で指定可能。

Poundのインストール

```
# wget  
http://www.apsis.ch/pound/Pound-  
2.4.4.tgz  
# tar xvfz Pound-2.4.4.tgz  
# cd Pound-2.4.4  
# ./configure  
# make  
# make install
```

Poundの設定ファイル

- /usr/local/etc/pound.cfg
に作成。
- <http://demo.example.com/>
にアクセスしたらdemoアプリ表示

Poundの設定ファイル

```
User "nobody"  
Group "nobody"  
LogLevel 0  
Alive 30  
ListenHTTP  
  Address 192.168.x.x  
  Port 80  
  xHTTP 2  
  Service  
    HeadRequire "Host: demo.example.com"  
    Backend  
      Address 127.0.0.1  
      Port 3000  
    End  
  End  
End
```

Pound起動

```
# /usr/local/sbin/pound  
-f /usr/local/etc/pound.cfg
```

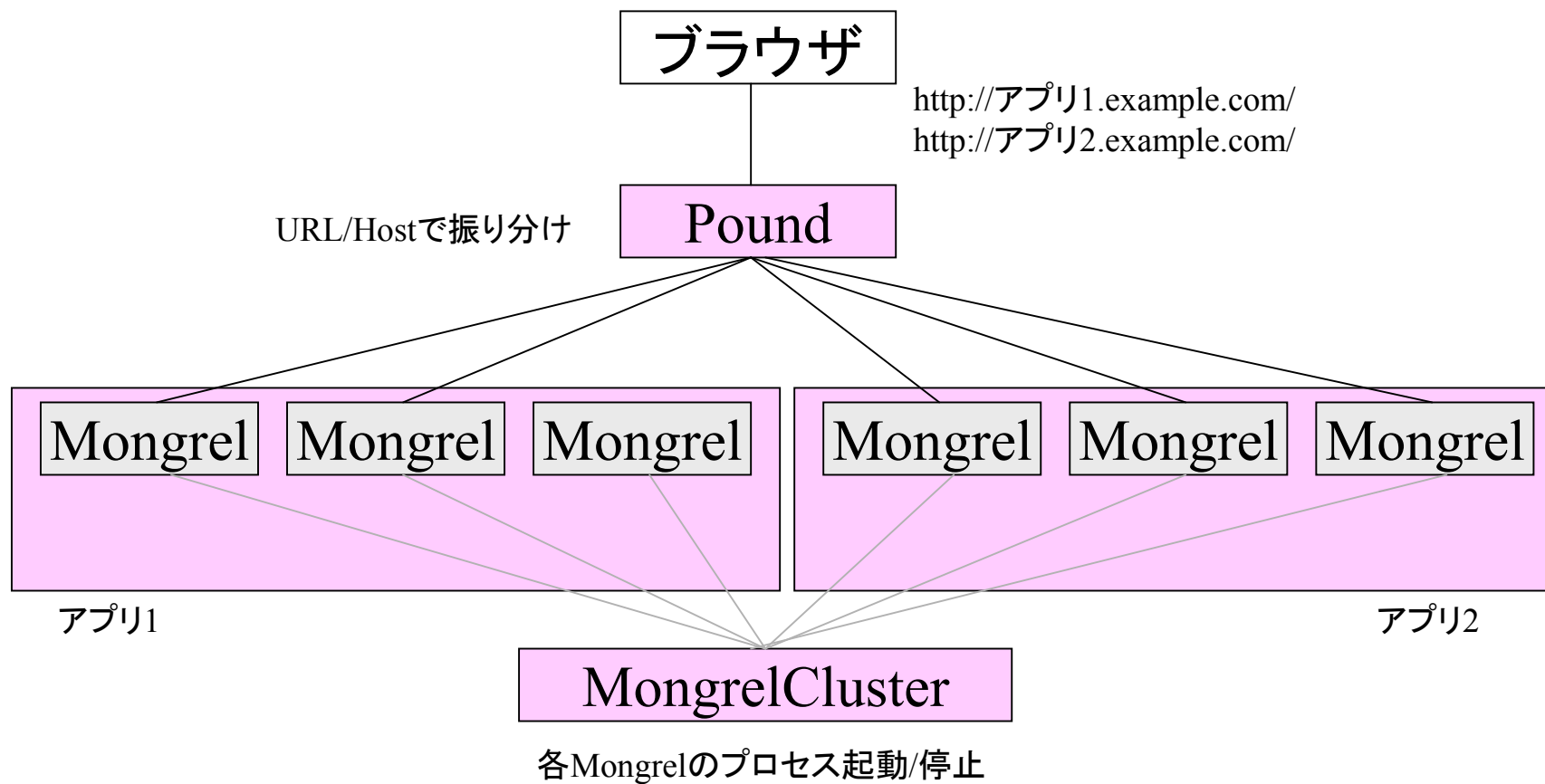
<http://demo.example.com/>

にアクセスできることを確認。

複数アプリの起動

- MongrelClusterの設定
 - /etc/mongrel_cluster/にアプリの数だけ設定ファイルのリンクを作る。
 - アプリ毎にポートを分ける。
- Poundの設定ファイルに追記する。
 - Serviceディレクティブを増やす。
 - それぞれのアプリのポートに振り分ける。

最後に



その他の選択肢

- Pound以外の選択肢
 - nginx(高速なロードバランサ)
 - Apache 2.2 + mod_proxy_balancer
- Mongrel以外の選択肢
 - lighttpd(軽量なWebサーバ)
 - Thin(Ruby web server)
- これだけあればok
 - Passenger(mod_rails for Apache)

ご清聴ありがとうございました