

Rails 製アプリケーションを動かす 5 つの方法

okkez @ Ruby 関西

2008 年 05 月 17 日

今回対象の web サーバ

- Apache
- lighttpd
- WEBrick
- Mongrel
- Thin

CGI

- 設定は簡単
- すごく遅い
- Rails で使用したという情報は少ない

設定例

Virtual Host を使用する場合。

```
<VirtualHost *>
  ServerName cgi.localhost
  Options ExecCGI FollowSymLinks
  ErrorLog /var/log/apache2/qa_cgi-error.log
  CustomLog /var/log/apache2/qa_cgi-access.log combined
  DocumentRoot /home/xxx/ruby/temp/qa_cgi/public
  <Directory /home/xxx/ruby/temp/qa_cgi/public>
    AllowOverride All
    Order deny,allow
    Allow from All
  </Directory>
</VirtualHost>
```

.htaccess は CGI 用に編集する。環境変数は、RAILS_ROOT/config/environment.rb に設定する。
Virtual Host を使用しない場合は RAILS_ROOT/public/.htaccess を編集して RewriteBase を変更する。

FastCGI

- インストールは簡単
- 設定は簡単
- そこそこ速い
- 情報は豊富

設定例

Virtual Host を使用する場合

```
<IfModule mod_fcgid.c>
    IdleTimeout 3600
    ProcessLifeTime 7200
    MaxProcessCount 8
    DefaultMaxClassProcessCount 2
    DefaultInitEnv RAILS_ENV production
    IPCConnectTimeout 20
    IPCCommTimeout 300
</IfModule>
<VirtualHost *>
    ServerName fcgi.localhost
    Options ExecCGI FollowSymLinks
    ErrorLog /var/log/apache2/qa_fcgi-error.log
    CustomLog /var/log/apache2/qa_fcgi-access.log combined
    DocumentRoot /home/xxx/ruby/temp/qa_fcgi/public
    <Directory /home/xxx/ruby/temp/qa_fcgi/public>
        AllowOverride All
        Order deny,allow
        Allow from All
    </Directory>
</VirtualHost>
```

.htaccess を編集。ここでは mod_fcgid を使っているのに合わせて編集する。詳細は略。
Virtual Host を使用しない場合は RAILS_ROOT/public/.htaccess を編集して RewriteBase を変更する。

mod_ruby

- 設定はそんなに難しくない
- 速度はそこそこ
- あまり情報がない
 - 探しきれてないだけ？

設定例

Virtual Host を使用する場合の例

```
RubySafeLevel 0
<VirtualHost *>
  ServerName ruby.localhost
  RubyRequire apache/ruby-run
  <Files *.rb>
    SetHandler ruby-object
    RubyHandler Apache::RubyRun.instance
  </Files>
  ErrorLog /var/log/apache2/qa_ruby-error.log
  CustomLog /var/log/apache2/qa_ruby-access.log combined
  DocumentRoot /home/xxx/ruby/temp/qa_mod_ruby/public
  <Directory /home/xxx/ruby/temp/qa_mod_ruby/public>
    AllowOverride All
    Order deny,allow
    Allow from All
  </Directory>
</VirtualHost>
```

.htaccess を dispatch.rb を使用するように編集する。

dispatch.rb を編集する。

```
ADDITIONAL_LOAD_PATHS.reverse.each ....
```

で始まる行をコメントアウトする。

Virtual Host を使用しない場合は RAILS_ROOT/public/.htaccess を編集して RewriteBase を変更する。

Passenger (a.k.a mod_rails)

- インストールは簡単
- 設定は簡単
- そこそこ速い

設定例

Virtual Host を使用する場合。

```
<VirtualHost *>
  ServerName rails.localhost
  DocumentRoot /var/www/qa_mod_rails/public
  <Directory /var/www/qa_mod_rails/public >
    Options FollowSymlinks
    AllowOverride None
    Order allow,deny
    Allow from all
  </Directory>
  ErrorLog /var/log/apache2/qa_mod_rails-error.log
  CustomLog /var/log/apache2/qa_mod_rails-access.log combined
  RailsEnv production
  RailsMaxPoolSize 2
  RailsPoolIdleTime 600
</VirtualHost>
```

この例では、/var/www 以下にアプリケーションを設置しているがこれは良くない。Virtual Host を使うなら他の場所に置くべき。

mongrel_cluster + mod_proxy_balancer

- 設定は難しい
- 速いかと思ったら...
- 情報は探せば見つかる

設定例

```
<VirtualHost *>
  ServerName cluster.localhost
  Options ExecCGI FollowSymLinks
  ErrorLog /var/log/apache2/qa_cluster-error.log
  CustomLog /var/log/apache2/qa_cluster-access.log combined
  ProxyRequests Off
  <Proxy balancer://mongrel_cluster>
    BalancerMember http://127.0.0.1:8000 loadfactor=10
    BalancerMember http://127.0.0.1:8001 loadfactor=10
    BalancerMember http://127.0.0.1:8002 loadfactor=10
    BalancerMember http://127.0.0.1:8003 loadfactor=10
    BalancerMember http://127.0.0.1:8004 loadfactor=10
```

```

</Proxy>
# These directories should always be served up by Apache,
# since they contain static content. Or just let rails do it.
ProxyPass /images !
ProxyPass /stylesheets !
ProxyPass /javascripts !
ProxyPass /favicon.ico !

ProxyPass / balancer://mongrel_cluster lbmethod=byrequests
ProxyPassReverse / balancer://mongrel_cluster

DocumentRoot /home/xxx/ruby/temp/qa_mongrel/public
<Directory /home/xxx/ruby/temp/qa_mongrel/public>
    AllowOverride All
    Order deny,allow
    Allow from All
</Directory>

RewriteEngine On

# Important rule to prevent exposure of subversion files
# if you are deploying with Capistrano !
RewriteRule ^(.*/)?.svn/ - [F,L]

# Rewrite index to check for static
RewriteRule ^/$ /index.html [QSA]

# Rewrite to check for Rails cached page
RewriteRule ^([^.]+)$ $1.html [QSA]

# Redirect all non-static requests to cluster
RewriteCond %{DOCUMENT_ROOT}/%{REQUEST_FILENAME} !-f
RewriteRule ^/(.*)$ balancer://mongrel_cluster%{REQUEST_URI} [P,QSA,L]
</VirtualHost>

```

注意

mod_proxy の設定が出来ていないと動かない。

```

<IfModule mod_proxy.c>
    #turning ProxyRequests on and allowing proxying from all may allow
    #spammers to use your proxy to send email.

```

```

ProxyRequests Off

<Proxy *>
    AddDefaultCharset off
    #Order deny,allow
    #Deny from all
    #Allow from .example.com
    Order allow,deny
    Allow from all
</Proxy>

# Enable/disable the handling of HTTP/1.1 "Via:" headers.
# ("Full" adds the server version; "Block" removes all outgoing Via: headers)
# Set to one of: Off | On | Full | Block

ProxyVia On
</IfModule>

```

また、ちゃんと動かそうと思うと依存するモジュールが多いので注意が必要。私が設定した時は、`mod_proxy_http` がロードされていなくてハマりました。

ベンチマーク結果

```
% ab -n 1000 -c 10 http://rails.localhost/answers/summary/2
```

CGI

```

Server Software:      Apache/2.2.8
Server Hostname:      cgi.localhost
Server Port:          80

Document Path:        /answers/summary/2
Document Length:      23592 bytes

Concurrency Level:    10
Time taken for tests:  2704.89406 seconds
Complete requests:    1000
Failed requests:       0
Write errors:          0
Total transferred:    24117000 bytes
HTML transferred:     23592000 bytes
Requests per second:  0.37 [#/sec] (mean)
Time per request:     27040.894 [ms] (mean)

```

Time per request: 2704.089 [ms] (mean, across all concurrent requests)
Transfer rate: 8.71 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 2.9	0	31
Processing:	12516	27017 508.6	26991	27939
Waiting:	12487	26655 497.6	26628	27620
Total:	12516	27018 508.4	26991	27939

Percentage of the requests served within a certain time (ms)

50%	26991
66%	27061
75%	27114
80%	27163
90%	27291
95%	27455
98%	27624
99%	27686
100%	27939 (longest request)

FCGI

Server Software: Apache/2.2.8
Server Hostname: fcgi.localhost
Server Port: 80

Document Path: /answers/summary/2
Document Length: 23592 bytes

Concurrency Level: 10
Time taken for tests: 152.882008 seconds
Complete requests: 1000
Failed requests: 16
(Connect: 0, Length: 16, Exceptions: 0)
Write errors: 0
Non-2xx responses: 16
Total transferred: 23742984 bytes
HTML transferred: 23222064 bytes
Requests per second: 6.54 [#/sec] (mean)
Time per request: 1528.820 [ms] (mean)
Time per request: 152.882 [ms] (mean, across all concurrent requests)
Transfer rate: 151.66 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 2.6	0	27
Processing:	191 1508	8098.0	249	64522
Waiting:	191 1508	8098.0	249	64522
Total:	191 1508	8098.7	249	64522

Percentage of the requests served within a certain time (ms)

50%	249
66%	255
75%	267
80%	637
90%	819
95%	1214
98%	7424
99%	64031
100%	64522 (longest request)

mod_ruby

Server Software: Apache/2.2.8
Server Hostname: ruby.localhost
Server Port: 80

Document Path: /answers/summary/2
Document Length: 23592 bytes

Concurrency Level: 10
Time taken for tests: 203.246801 seconds
Complete requests: 1000
Failed requests: 0
Write errors: 0
Total transferred: 24117000 bytes
HTML transferred: 23592000 bytes
Requests per second: 4.92 [#/sec] (mean)
Time per request: 2032.468 [ms] (mean)
Time per request: 203.247 [ms] (mean, across all concurrent requests)
Transfer rate: 115.87 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 3.0	0	32

Processing:	449	2026	3656.3	1301	32864
Waiting:	401	1975	3656.8	1268	32863
Total:	449	2026	3658.2	1301	32864

Percentage of the requests served within a certain time (ms)

50%	1301
66%	1320
75%	1359
80%	1532
90%	3284
95%	3434
98%	3948
99%	26825
100%	32864 (longest request)

webrick

Server Software:	WEBrick/1.3.1
Server Hostname:	localhost
Server Port:	3000

Document Path:	/answers/summary/2
Document Length:	23592 bytes

Concurrency Level:	10
Time taken for tests:	158.277767 seconds
Complete requests:	1000
Failed requests:	0
Write errors:	0
Total transferred:	24065000 bytes
HTML transferred:	23592000 bytes
Requests per second:	6.32 [#/sec] (mean)
Time per request:	1582.778 [ms] (mean)
Time per request:	158.278 [ms] (mean, across all concurrent requests)
Transfer rate:	148.47 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 0.6	0	8
Processing:	276	1575 336.4	1518	3244
Waiting:	257	1552 336.2	1496	3224
Total:	276	1575 336.4	1518	3244

Percentage of the requests served within a certain time (ms)

50%	1518
66%	1531
75%	1686
80%	1711
90%	1759
95%	1790
98%	2793
99%	2989
100%	3244 (longest request)

mongrel

Server Software: Mongrel
Server Hostname: localhost
Server Port: 3000

Document Path: /answers/summary/2
Document Length: 23592 bytes

Concurrency Level: 10
Time taken for tests: 163.933646 seconds
Complete requests: 1000
Failed requests: 0
Write errors: 0
Total transferred: 24056000 bytes
HTML transferred: 23592000 bytes
Requests per second: 6.10 [#/sec] (mean)
Time per request: 1639.336 [ms] (mean)
Time per request: 163.934 [ms] (mean, across all concurrent requests)
Transfer rate: 143.30 [Kbytes/sec] received

Connection Times (ms)

	min	mean	mean[+/-sd]	median	max
Connect:	0	0	0.0	0	0
Processing:	194	1632	151.9	1645	2692
Waiting:	187	1601	154.4	1623	2691
Total:	194	1632	151.9	1645	2692

Percentage of the requests served within a certain time (ms)

50%	1645
66%	1654
75%	1659

80%	1663
90%	1724
95%	1898
98%	1912
99%	1922
100%	2692 (longest request)

Thin

```

Server Software:      thin
Server Hostname:     localhost
Server Port:         3000

Document Path:       /answers/summary/2
Document Length:     23592 bytes

Concurrency Level:    10
Time taken for tests: 144.303682 seconds
Complete requests:    1000
Failed requests:      0
Write errors:         0
Total transferred:    24024000 bytes
HTML transferred:     23592000 bytes
Requests per second:  6.93 [#/sec] (mean)
Time per request:     1443.037 [ms] (mean)
Time per request:     144.304 [ms] (mean, across all concurrent requests)
Transfer rate:        162.57 [Kbytes/sec] received

```

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 0.0	0	0
Processing:	1059	1442 110.9	1401	1972
Waiting:	1057	1440 110.9	1399	1970
Total:	1059	1442 110.9	1401	1972

Percentage of the requests served within a certain time (ms)

50%	1401
66%	1410
75%	1417
80%	1446
90%	1640
95%	1657
98%	1667

99% 1971
100% 1972 (longest request)

Ebb

動かなかった orz

mongrel_cluster + mod_proxy_balancer

Server Software: Mongrel
Server Hostname: cluster.localhost
Server Port: 80

Document Path: /answers/summary/2
Document Length: 23592 bytes

Concurrency Level: 10
Time taken for tests: 171.113394 seconds
Complete requests: 1000
Failed requests: 0
Write errors: 0
Total transferred: 24084000 bytes
HTML transferred: 23592000 bytes
Requests per second: 5.84 [#/sec] (mean)
Time per request: 1711.134 [ms] (mean)
Time per request: 171.113 [ms] (mean, across all concurrent requests)
Transfer rate: 137.45 [Kbytes/sec] received

Connection Times (ms)

	min	mean	mean[+/-sd]	median	max
Connect:	0	0	0.0	0	0
Processing:	556	1704	455.5	1650	2882
Waiting:	539	1633	441.6	1573	2797
Total:	556	1704	455.5	1650	2882

Percentage of the requests served within a certain time (ms)

50% 1650
66% 1977
75% 2130
80% 2173
90% 2266
95% 2352
98% 2532

99% 2547
100% 2882 (longest request)

Phusion-Passenger

Server Software: Apache/2.2.8
Server Hostname: rails.localhost
Server Port: 80

Document Path: /answers/summary/2
Document Length: 23592 bytes

Concurrency Level: 10
Time taken for tests: 153.686688 seconds
Complete requests: 1000
Failed requests: 0
Write errors: 0
Total transferred: 24192000 bytes
HTML transferred: 23592000 bytes
Requests per second: 6.51 [#/sec] (mean)
Time per request: 1536.867 [ms] (mean)
Time per request: 153.687 [ms] (mean, across all concurrent requests)
Transfer rate: 153.72 [Kbytes/sec] received

Connection Times (ms)

	min	mean[+/-sd]	median	max
Connect:	0	0 0.3	0	4
Processing:	575	1534 646.5	1323	3719
Waiting:	574	1506 646.8	1297	3708
Total:	575	1534 646.5	1323	3719

Percentage of the requests served within a certain time (ms)

50%	1323
66%	1388
75%	1426
80%	1517
90%	2823
95%	3305
98%	3474
99%	3546
100%	3719 (longest request)