

Tworzenie szybkich stron www

Jak przyspieszyć strony internetowe

Metody przyspieszania

- Po stronie serwera (to będzie)
- Po stronie przeglądarki (tego dziś nie będzie)

Po stronie serwera

- Page Caching
- Action Caching
- Fragment Caching
- Data Caching
- Reverse Proxy (np: varnish) - to na osobną prezentację, ktoś inny może zrobić :)

Page Caching

- najszybsza metoda serwowania stron
- oparta na plikach (można też memcache użyć)
- cała strona jest statyczna
- istnieją możliwości dodania dynamicznej treści
- tylko dla stron publicznych (chyba, że zrobimy autoryzacje na serwerze www)

DEMO #1

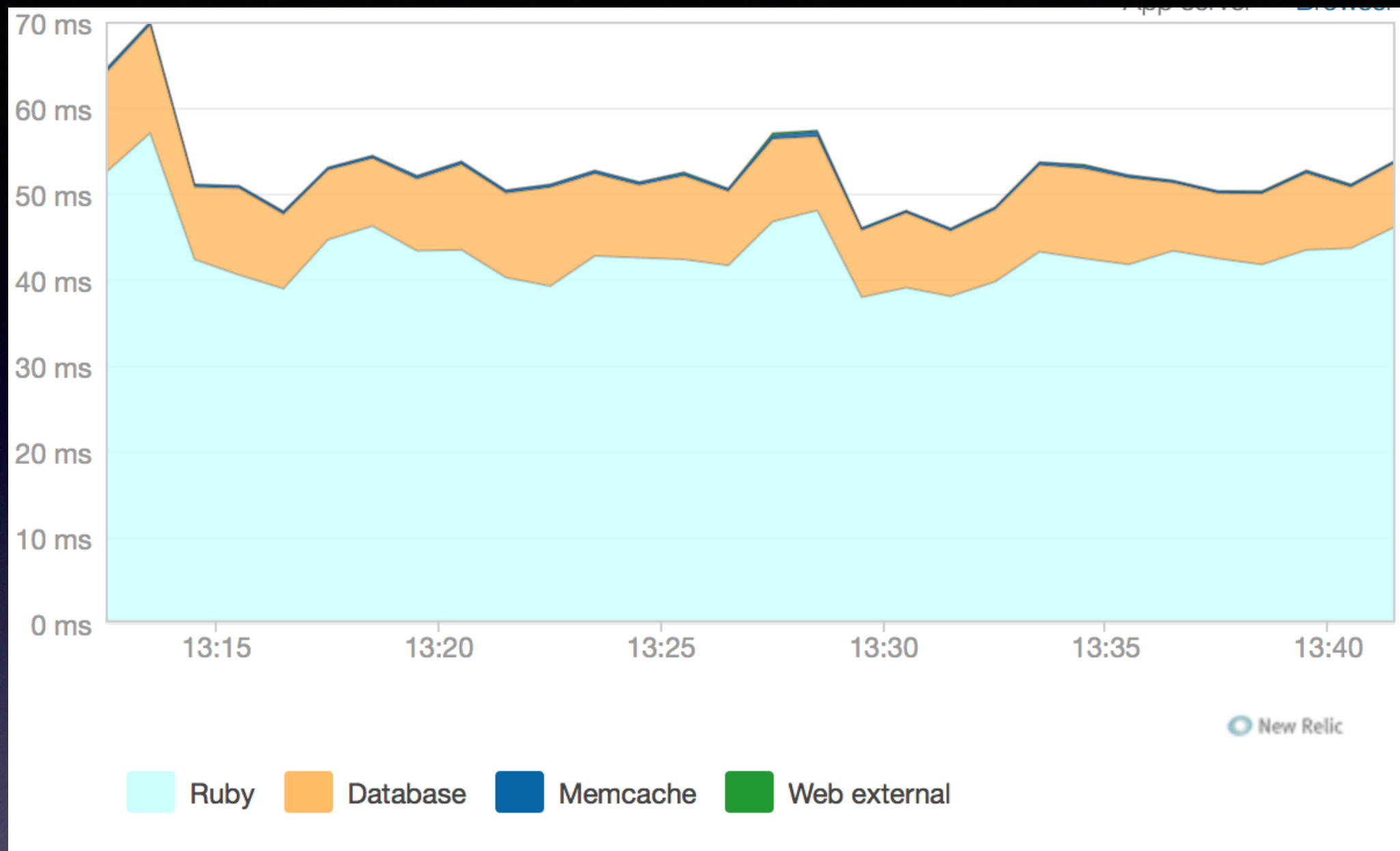
Action Caching

- Filtry przed akcjami nadal są wywoływane
- Można dzięki temu sterować widocznością strony (np tylko dla zalogowanych użytkowników, lub o określonych uprawnieniach)
- Wolniejsza od Page Caching bo dotyka serwera aplikacji

DEMO #2

Fragment Caching

- Caching poszczególnych elementów widoku (template)
- Możliwość automatycznego wygaszania cache
- Zastosowanie russian doll caching



Co cachować?

Wielu zapewne myśli, że to baza jest wąskim gardłem ale często wąskim gardłem jest rederowanie htmla.

W przypadku standardowych
stron rederowanie widoku
zabiera najwięcej czasu.

Russian doll caching

Polega na zagnieżdżaniu „zcachowanych” kawałków stron by zmaksymalizować ilość trafień w pamięć podręczną.



Poprzez zagnieżdżanie mamy
możliwość ponownego wykorzystania
pamięci podręcznej nawet jeżeli
niektóre elementy się zmieniły.


```
1 <%= cache @team do %>␣
2   <h1>Informacje o drużynie</h1>␣
3   <hr>␣
4   <h4><%= @team.name %></h4>␣
5   ␣
6   <h5>Gracze:</h5>␣
7   <ol>␣
8     <% @team.members.each do |member| %>␣
9       <%= cache member do %>␣
10        <li>␣
11          <strong>␣
12            <%= link_to member.name.humanize, [:edit, member] %>␣
13          </strong> ␣
14          <%= format_text(member.bio) %>␣
15        </li>␣
16      <% end %>␣
17    <% end %>␣
18  </ol>␣
19 <% end %>␣
```

Przykład zagnieżdżenia

Zespół składa się z wielu graczy. Zmiana danych jednego gracza spowoduje, że pamięć podręczna pozostałych nadal będzie poprawna.

DEMO #3

Problemy z jakimi można się spotkać

I jak je rozwiązać

Typowe problemy

- Zależność od czasu (np wiek aktora, jak dawno temu napisano komentarz)
- Zależność od `current_user` (wiele wersji tego samego fragmetu, np: uprawnienia)
- Brak nadrzędnego pojedynczego obiektu od którego można rozpocząć russian doll caching

DEMO #4

Do pobrania prezentacja i przykładowa aplikacja

https://github.com/madmax/neubloc_talk_31

PYTANIA?