

Azure na produkcji w Homebook.pl

Wrocław, 2016

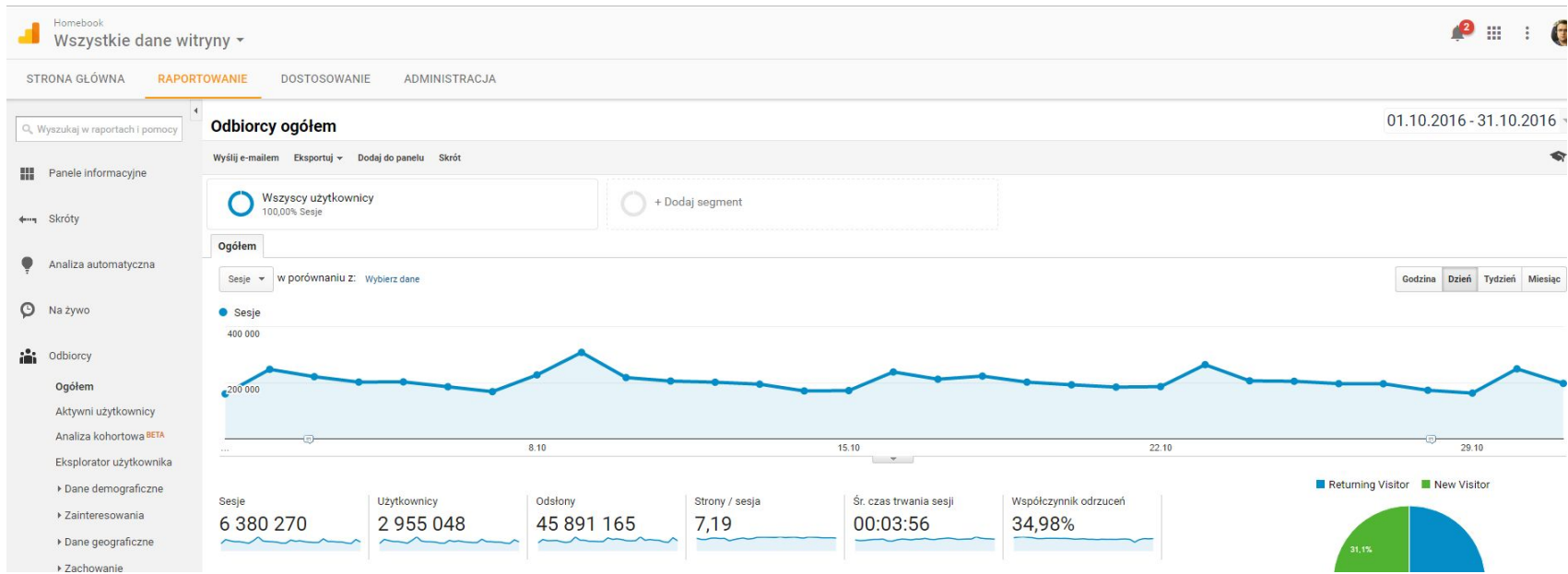
Agenda

- Kontekst decyzji
- Aplikacja
- SQL
- Zdjęcia
- Solr
- Inne
- Pytania?

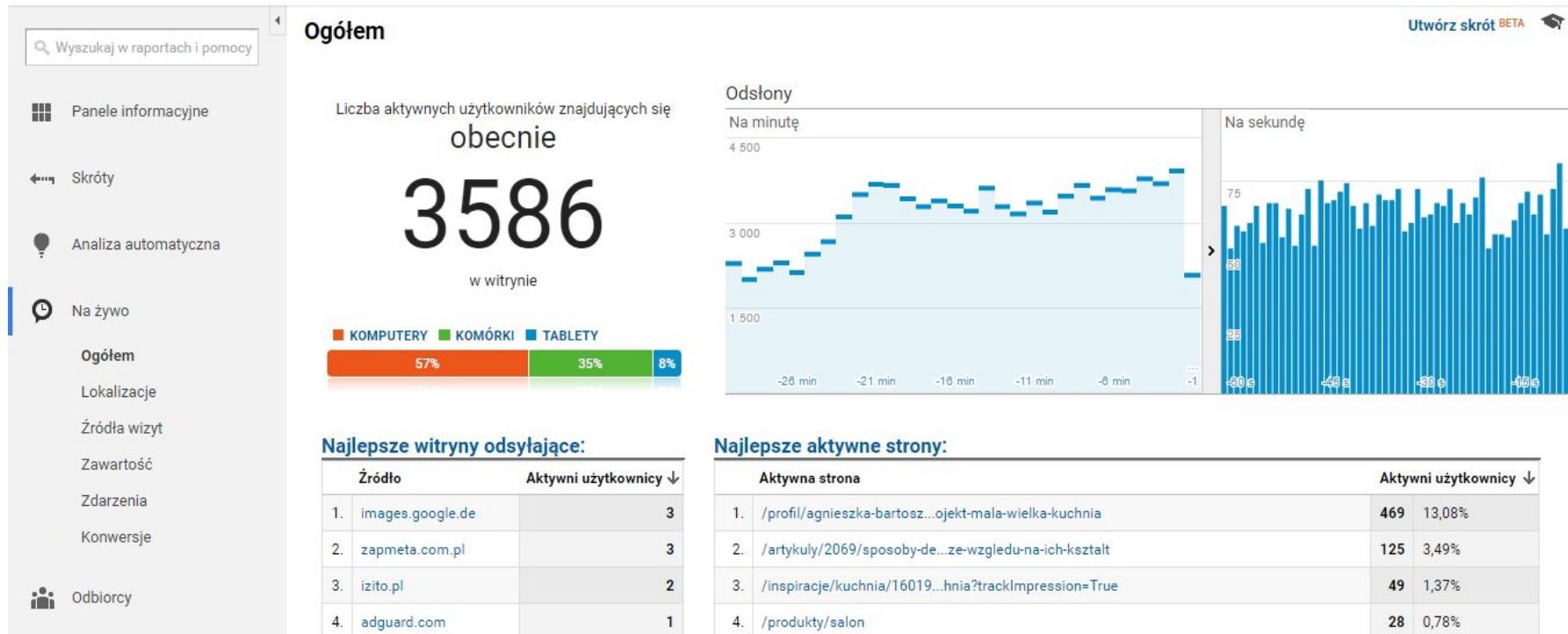
Kontekst decyzji 2013

- startowaliśmy
- nie jesteśmy softwarehousem
- obserwowaliśmy "niekontrolowany" wzrost w cenie
- mieliśmy doświadczenia z adminami
- dużo było w tym przypadku

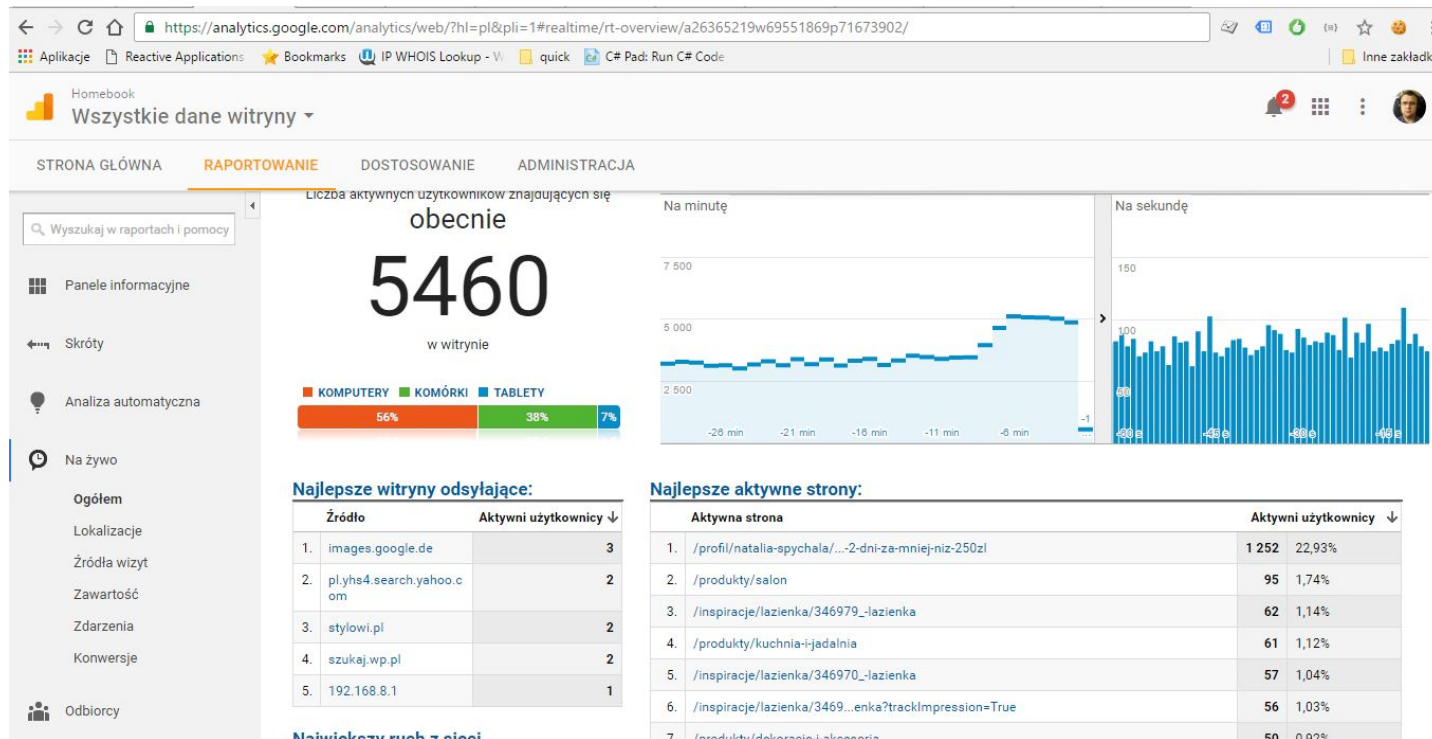
Październik 2016



Mamy charakterystyczny ruch



Mamy charakterystyczny ruch cd



Aplikacja

- .NET 4.5, C# 6.0, ASP.NET MVC 5.2.3
- Cloud service, ale zaczynaliśmy od WebApp
- 3-7 instancji, 8 cores 14GB ram, "ExtraLarge", A4 0.64\$/h
- Autoscaling, CPU 30%-40%/5min +-1 instancji
- Wiele instancji vs Cache

Dane, cache

- Pobieramy obiekty by id : `select from table where id in ()`
- Elementy rozmieszczamy w cache pojedynczo, nie wrzucamy do pamięci list prowadzących do duplikacji elementów
 - mniej chodzi o ilość pamięci, bardziej o spójność na etapie inwalidacji
 - lista produktów nie, lista kategorii tak
- Nie modyfikujemy elementów w cache
- Elementy w cache expirują albo zostają usunięte bezpośrednio lub pośrednio w ramach propagacji

Dane, cache cd

```
public async Task<Dictionary<int, IdeaBook>> GetByIds(IEnumerable<int> ideaBooksIds)
{
    var ideaBooksIdsList = ideaBooksIds.Distinct().ToList();
    var ideabooks = new List<IdeaBook>();
    var toGet = new List<int>();

    foreach (var ideaBookId in ideaBooksIdsList)
    {
        var ideabook = _cache[CacheKeyById(ideaBookId)] as IdeaBook;
        if (ideabook != null)
        {
            ideabooks.Add(ideabook);
        }
        else
        {
            toGet.Add(ideaBookId);
        }
    }

    if (toGet.Count > 0)
    {
        var dbResult = await GetIdeabooksFromDb(toGet);

        foreach (var ideabook in dbResult)
        {
            ideabooks.Add(ideabook);

            _cache.Insert(CacheKeyById(ideabook.Id), ideabook, null, DateTime.Now.Add(CachingDurationHelper.Long()),
                Cache.NoSlidingExpiration, null);
        }
    }

    return ideabooks.ToDictionary(i => i.Id, i => i);
}
```

Dane, cache cd

```
private async Task<IEnumerable<IdeaBook>> GetIdeabooksFromDb(IEnumerable<int> ideabooksIds)
{
    using (var connection = GetConnection())
    {
        return await connection.QueryAsync<IdeaBook>(
            "SELECT [Id], [Name], [UserId], [CreatedDate], [LatestActivity], [Description], [CategoryId],
            "WHERE [Id] in @ideabooksIds",
            new { ideabooksIds },
            commandTimeout: 5
        );
    }
}
```

Dane, cache cd

- Usowanie

```
public async Task RemoveIdeaBook(IdeaBook ideaBook)
{
    using (var connection = GetConnection())
    {
        await connection.ExecuteNonQuery("DELETE FROM [dbo].[IdeaBooks] WHERE [Id] = @id", new { id = ideaBook.Id });
    }

    _cache.Remove(CacheKeyById(ideaBook.Id));
    _cache.Remove(CacheKeyByUserId(ideaBook.UserId));
    _cache.Remove(CacheKeyForIdeaBooksCategories(ideaBook.Id));
}
```

Dane, cache cd

- Propagacja

```
namespace Frontend.Helpers.Caching
{
    public class CacheInvalidationPropagator
    {
        private static readonly TopicClient TopicClient = TopicClient.Create(ServiceBusHelper.Topics.InvalidateCache);

        public void Propagate(string cacheKey)
        {
            try
            {
                var message = new InvalidateMessage { Origin = ServiceBusHelper.InstanceId, CacheKey = cacheKey };

                TopicClient.Send(new BrokeredMessage(message)
                {
                    MessageId = Guid.NewGuid().ToString(),
                    TimeToLive = TimeSpan.FromMinutes(1)
                });
            }
            catch (Exception ex)
            {
                Elmah.ErrorSignal.FromCurrentContext().Raise(ex);
            }
        }
    }
}
```

Dane, cache cd

- Propagacja

```
SubscriptionClient.Create(topicName, subscriptionName).OnMessage(message =>
{
    var invalidationMessage = message.GetBody<InvalidateMessage>();

    if (invalidationMessage.Origin != ServiceBusHelper.InstanceId)
    {
        HttpRuntime.Cache.Remove(invalidationMessage.CacheKey);
    }
});
```

Output cache

- kluczowy - renderowanie widoków jest naprawdę kosztowne
 - zrozumieliśmy to jak dla aplikacji mobilnych wystawiliśmy api
- nie synchronizujemy, godzimy się na rozjazdy
- elementy specyficzne dla użytkownika doładujemy javascriptem
- nie mamy dla stron szczegółów
- varyByCustom="ViewVersion;FromCPC;UsersAvailableShopsSet"

SQL

- Tranzakcje P6, Dane P4, Dane R0 P1, Dane R1 P1
- "sql azure connection throttling"
- Point in time restore
- Subiektywnie...

Zdjęcia

- 300GB oryginalnych inspiracji w Blobach
 - 1TB za 25\$
 - 1TB transferu za 90\$
 - wysyłamy około 10MB/s, wieczorami nawet 20MB/s, $(10 \cdot 3600 \cdot 720) \sim 25\text{TB} \cdot 90\$ = 2,5\text{tys } \$ \dots$
- 2 x <https://imageresizing.net/> z OVH, Niemcy
- Nginx, proxy z DigitalOcean,
 - DNS Round-robin
 - 5TB w cenie
 - 1TB ponad limit 20\$

Solr

- 2 indeksy produktowe, 1 inspiracyjny
 - 8 cores, 16GB, 160GB 160\$
- Nginx, proxy DigitalOcean
 - 2 cores, 2GB, 40GB 20\$
- Indeksowanie z OVH

Szpiegula

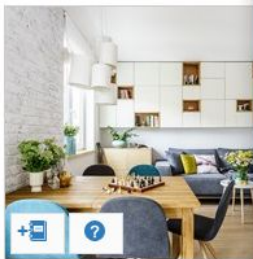
Włoski (294)

Kolonialny (791)

Art deco (1492)



3deko



SAJE ARCHITEKCI Joanna Morko

View: [Icons] | Preserve log | Disable cache | Offline No throttling

Filter [] | Regex | Hide data URLs | All | XHR JS CSS Img Media Font Doc WS Manifest Other

500 ms 1000 ms 1500 ms 2000 ms 2500 ms 3000 ms 35

Name	Status	Type	Initiator
vhdot.gif?itemId=383189&itemType=1&userId=df4e6b21-1536-44b4-a8cc-a8cb42320c83&date=1478859142880 data1.shmbk.pl	200 OK	gif	Other
vhdot.gif?itemId=370311&itemType=1&userId=df4e6b21-1536-44b4-a8cc-a8cb42320c83&date=1478859142880 data1.shmbk.pl	200 OK	gif	Other
vhdot.gif?itemId=373772&itemType=1&userId=df4e6b21-1536-44b4-a8cc-a8cb42320c83&date=1478859143176 data1.shmbk.pl	200 OK	gif	Other
vhdot.gif?itemId=359868&itemType=1&userId=df4e6b21-1536-44b4-a8cc-a8cb42320c83&date=1478859143177 data1.shmbk.pl	200 OK	gif	Other
vhdot.gif?itemId=294596&itemType=1&userId=df4e6b21-1536-44b4-a8cc-a8cb42320c83&date=1478859143177 data1.shmbk.pl	200 OK	gif	Other

<http://data1.shmbk.pl/vhdot.gif?itemId=359868&itemType=1&userId=df4e6b21-1536-44b4-a8cc-a8cb42320c83&date=1478859143177>

Szpiegula cd

- Składujemy po 1000 sesji w blobach, [mgravell/protobuf-net](#)



167	https://hsessions.blob.core.windows.net/20161110/167	11/11/2016 2:28:27 AM	3.28 MB
168	https://hsessions.blob.core.windows.net/20161110/168	11/11/2016 2:28:28 AM	3.66 MB
169	https://hsessions.blob.core.windows.net/20161110/169	11/11/2016 2:28:21 AM	3.54 MB
17	https://hsessions.blob.core.windows.net/20161110/17	11/11/2016 2:28:16 AM	3.78 MB
170	https://hsessions.blob.core.windows.net/20161110/170	11/11/2016 2:28:23 AM	3.63 MB
171	https://hsessions.blob.core.windows.net/20161110/171	11/11/2016 2:28:21 AM	3.36 MB
172	https://hsessions.blob.core.windows.net/20161110/172	11/11/2016 2:28:23 AM	3.69 MB
173	https://hsessions.blob.core.windows.net/20161110/173	11/11/2016 2:28:24 AM	3.07 MB
174	https://hsessions.blob.core.windows.net/20161110/174	11/11/2016 2:28:25 AM	2.92 MB
175	https://hsessions.blob.core.windows.net/20161110/175	11/11/2016 2:28:21 AM	2.65 MB
176	https://hsessions.blob.core.windows.net/20161110/176	11/11/2016 2:28:07 AM	2.58 MB
177	https://hsessions.blob.core.windows.net/20161110/177	11/11/2016 2:28:08 AM	1.24 MB

VisualSearch

- Prace trwają, ale jeszcze długa droga przed nami



AZzardo Plafon Caldo 50 AZzardo
1299 zł

[szczegóły »](#)



Lampa wisząca Acrylic 50cm
1219 zł

[szczegóły »](#)



Lampa wisząca Caldo 50, LC/LP2436-6PX
1299 zł

[szczegóły »](#)



Lampa wisząca Caldo 40, LC/LP2436-5PX
839 zł

[szczegóły »](#)



Plany

- ASP.NET Core
- Pracujemy nad automatycznym metkowaniem inspiracji
- Nad Homebookiem pracujemy we dwóch, ale jest milion tematów i w Domodi i Homebooku...

Pytania?

- Jeśli są to proszę o łatwiejsze ;)