Leszek Krupiński, PHPCon 2014

SKALOWANIE APLIKACJI PHP

H

14

320×320

L

A

P

F

W:618
C:318

BP:5.2ST:4.0
sp: 5.0
Powiększenie:1.2

F

W:739
C:342

CO TO JEST
SKALOWALNOŚĆ?

Skalowalność (ang. *scalability*) - zapewnienie coraz wydajniejszej pracy w miarę zwiększania liczby elementów składowych.

— WIKIPEDIA

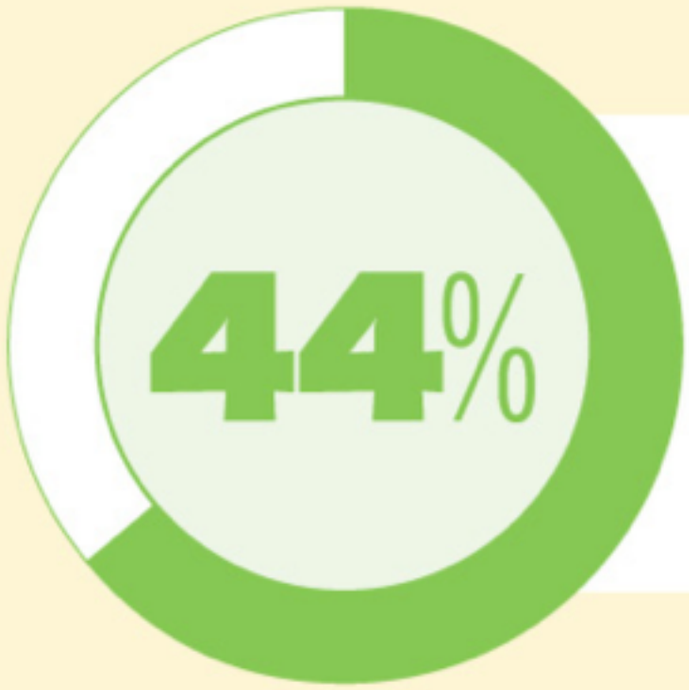
SKALOWANIE WERTYKALNE

SKALOWANIE HORYZONTALNE

PO CO DBAĆ O **SKALOWALNOŚĆ**?

ZADOWOLENIE **KLIENTA**

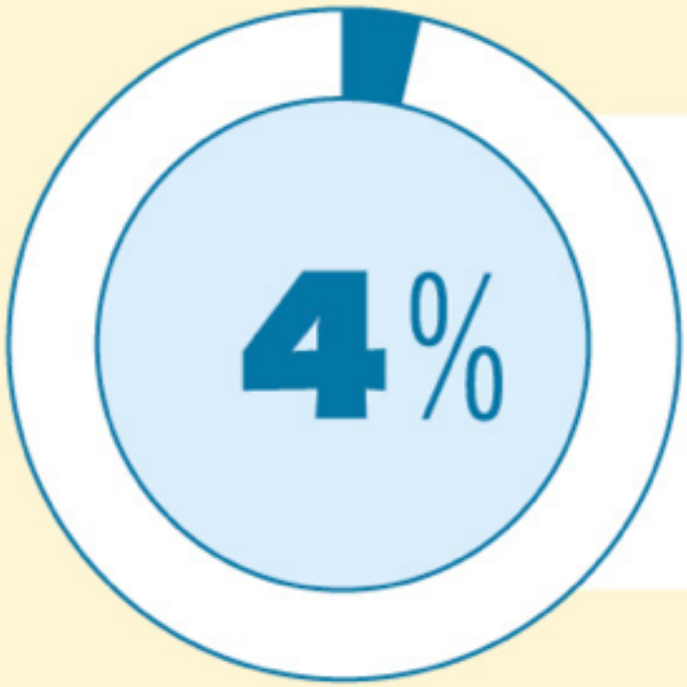
bit.ly/slowwebspeed



44% of users say that slow online transactions make them

ANXIOUS

about the success of the transaction.



4% of people have

THROWN THEIR PHONE

while using a slow mobile site.

JAKIE SĄ **KRYTYCZNE WARTOŚCI**?

HOW FAST is fast enough?

57%

of online consumers

WILL ABANDON A

SLOW PAGE

AFTER 3 SECONDS

We want you to be able to flick from one page to another **as quickly as you can flick a page in a book.**

SO WE'RE aiming very, very HIGH ...at something like

100

MILLISECONDS

Urs Hölzle,
Senior VP Operations, Google

HOW YOUR BRAIN
perceives page load times

0.1 SECOND
Feels instantaneous

1 SECOND
Lets you think seamlessly

10 SECONDS
Keeps your attention...barely

+10 SECONDS
Loses you

0 2 4 6 8 10 12 13 14

WYMIERNE
KORZYŚCI

15,4% WIĘCEJ POBRAŃ
FIREFOKSA PO
PRZYSPIESZENIU STRONY
MOZILLI **O 2,2s**

14% WIĘCEJ WPŁAT NA KAMPANIĘ BARACKA OBAMY PO PRZYSPIESZENIU STRONY ZGŁOSZENIOWEJ **O 60%**

WYLICZENIA WALMARTU I AMAZONU:

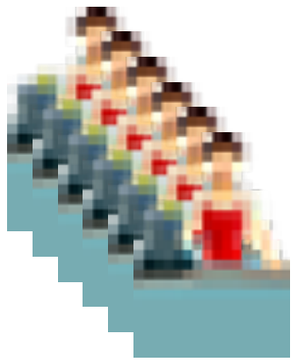
ZYSKI WIĘKSZE O 1% ZA KAŻDE 100ms

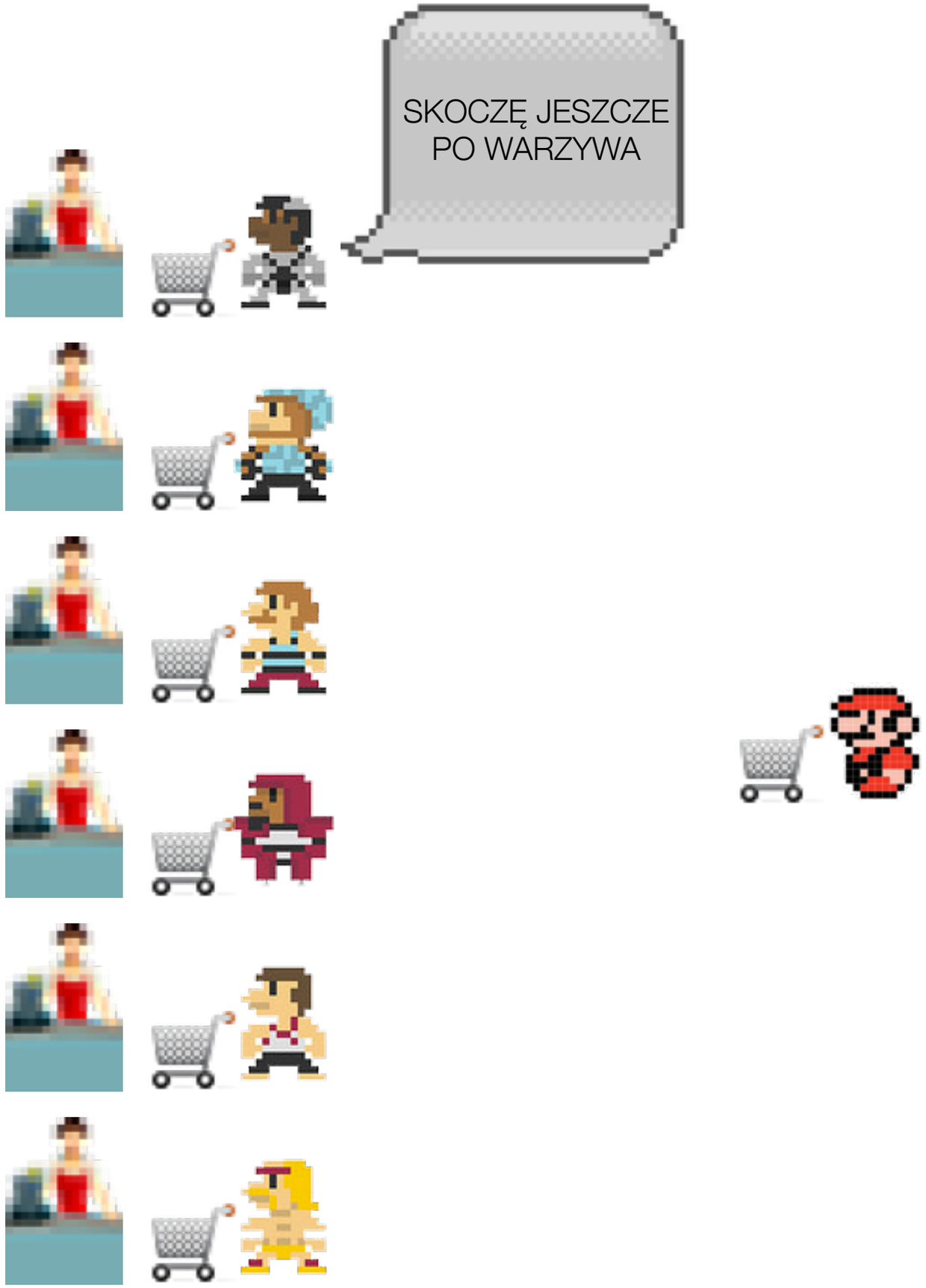
WYLICZENIA YAHOO:

9% WIĘCEJ RUCHU W SERWISIE ZA KAŻDE 400ms

PODSTAWY

HTTP REQUEST




SKOCZĘ JESZCZE
PO WARZYWA

CO MOŻNA
ZROBIĆ?

- **SZYBSZA OBSŁUGA KLIENTA**
- **WIĘCEJ KAS**
- **KOLEJNE SKLEPY**

KUPMY

WIĘKSZY

SERWER

KUPMY

WIĘCEJ

SERWERÓW

BEZ
ODPOWIEDNIEJ
ARCHITEKTURY
NIE MA TO SENSU

BYĆ MOŻE WYSTARCZY

OPTYALIZACJA

APLIKACJI

WYCIŚNIĘCIE
MAKSIMUM Z
MOŻLIWOŚCI MASZYNY

OPTYMALIZUJ **ISTOTNE** FRAGMENTY KODU

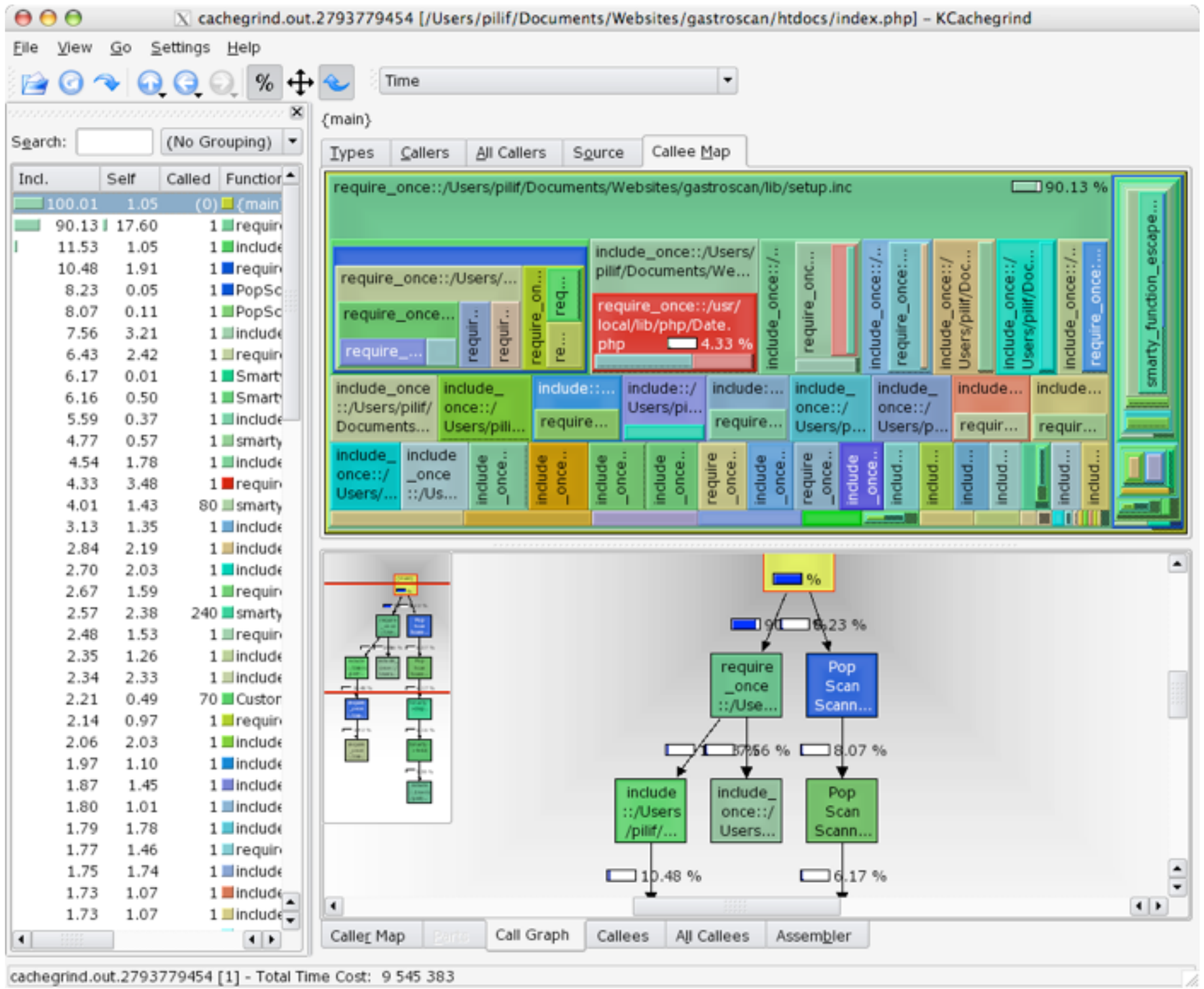
Premature optimization is the root of all evil.

— Donald Knuth

UPEWNIJ SIĘ, ŻE
TO **NA PEWNO PHP**
JEST WINNE

NARZĘDZIA

xdebug + cachegrind



webgrind of /Library/WebServer/Documents/musikfreak/index.php

http://localhost/webgrind/

Google

webgrind v0.8

profiling in the browser

Show 90% of musikfreak/index.php (cachegr in percent update

Hide PHP functions

/Library/WebServer/Documents/musikfreak/index.php

cachegrind.out.musikfreak_news__XDEBUG_PROFILE12702 @ 2008-06-23 14:05:21

418 different functions called in **148** milliseconds (**109** shown)

Function	Invocation Count	Total Self Cost	Total Inclusive Cost
Zend_Loader::loadClass	11	7.51	20.11
Zend_Controller_Front->dispatch	1	5.89	62.75
include::/Library/WebServer/Documents/musikfreak/app/views/scripts/news/index.phtml	1	5.12	8.07
Zend_Cache::factory	1	3.77	7.42
php::unserialize	4	3.24	3.24
include_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Front.php	1	3.20	10.29
require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Router/Rewrite.php	1	3.04	3.78
require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Action.php	1	2.57	2.57
{main}	1	2.47	99.99
require_once::/Library/WebServer/Documents/zend/library/Zend/View.php	1	2.22	2.49
Zend_Controller_Action_Helper_ViewRenderer->getInflector	3	2.01	10.72
require_once::/Library/WebServer/Documents/zend/library/Zend/Cache/Backend/File.php	1	1.55	1.55
require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Action/Helper/Abstract.php	1	1.45	4.02
require_once::/Library/WebServer/Documents/zend/library/Zend/Filter/Inflector.php	1	1.41	2.12
include_once::/Library/WebServer/Documents/zend/library/Zend/Config/Ini.php	1	1.40	1.41
Zend_Loader::loadFile	1	1.39	1.44
Zend_Loader_PluginLoader->load	6	1.28	3.45
require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Request/Http.php	1	1.28	1.62
require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Dispatcher/Standard.php	1	1.27	1.71
php::crc32	2	1.18	1.18
Zend_Controller_Dispatcher_Abstract->_formatName	7	1.17	1.25
Zend_Controller_Front->getDispatcher	8	1.15	2.95
require_once::/Library/WebServer/Documents/zend/library/Zend/Controller/Action/HelperBroker.php	1	1.15	5.78

Display a menu

XHProf

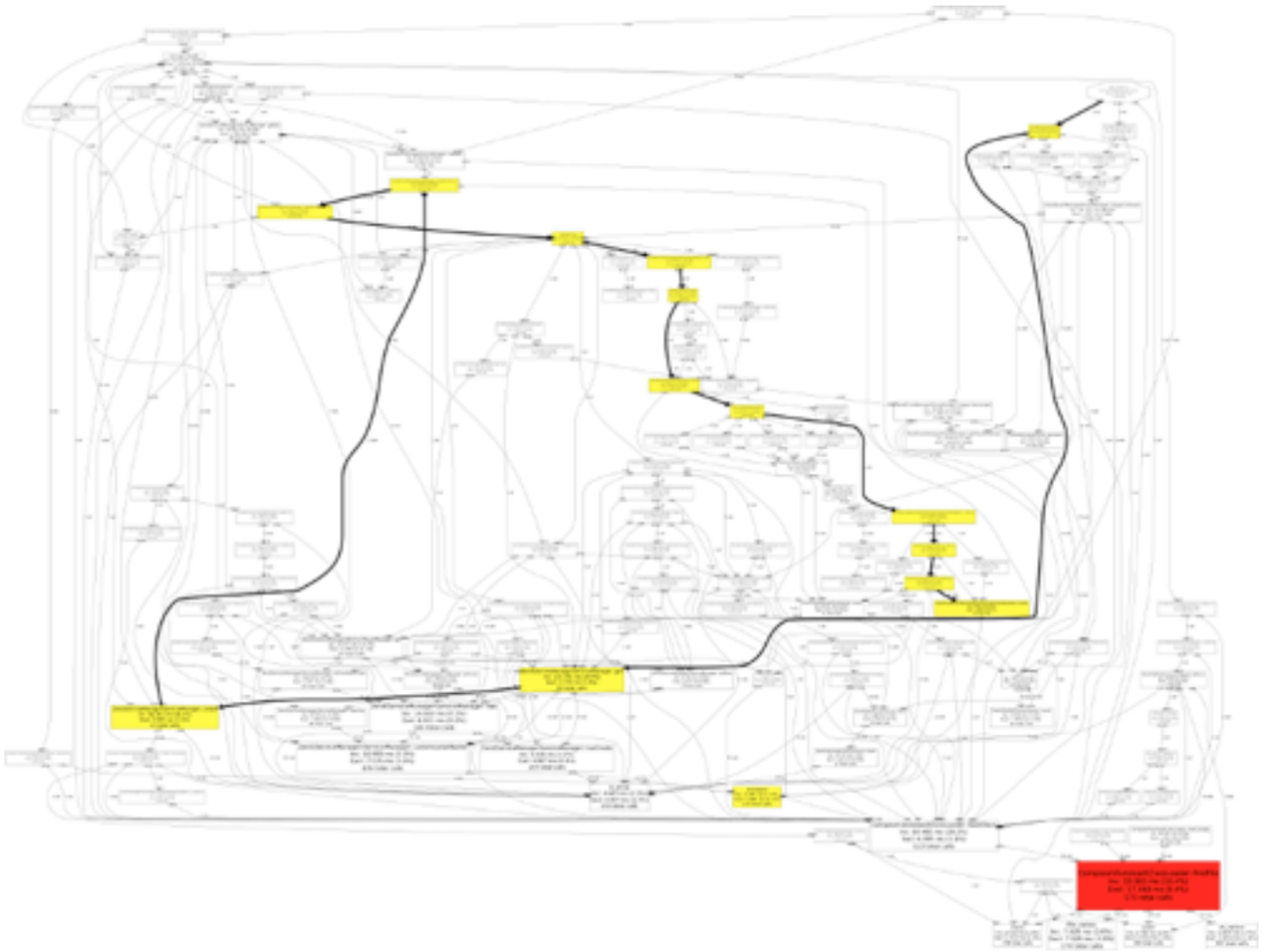
Overall Summary

Total Incl. Wall Time (microsec): 44,801 microsecs
Total Incl. CPU (microsecs): 44,002 microsecs
Total Incl. MemUse (bytes): 2,459,584 bytes
Total Incl. PeakMemUse (bytes): 2,508,032 bytes
Number of Function Calls: 1,513

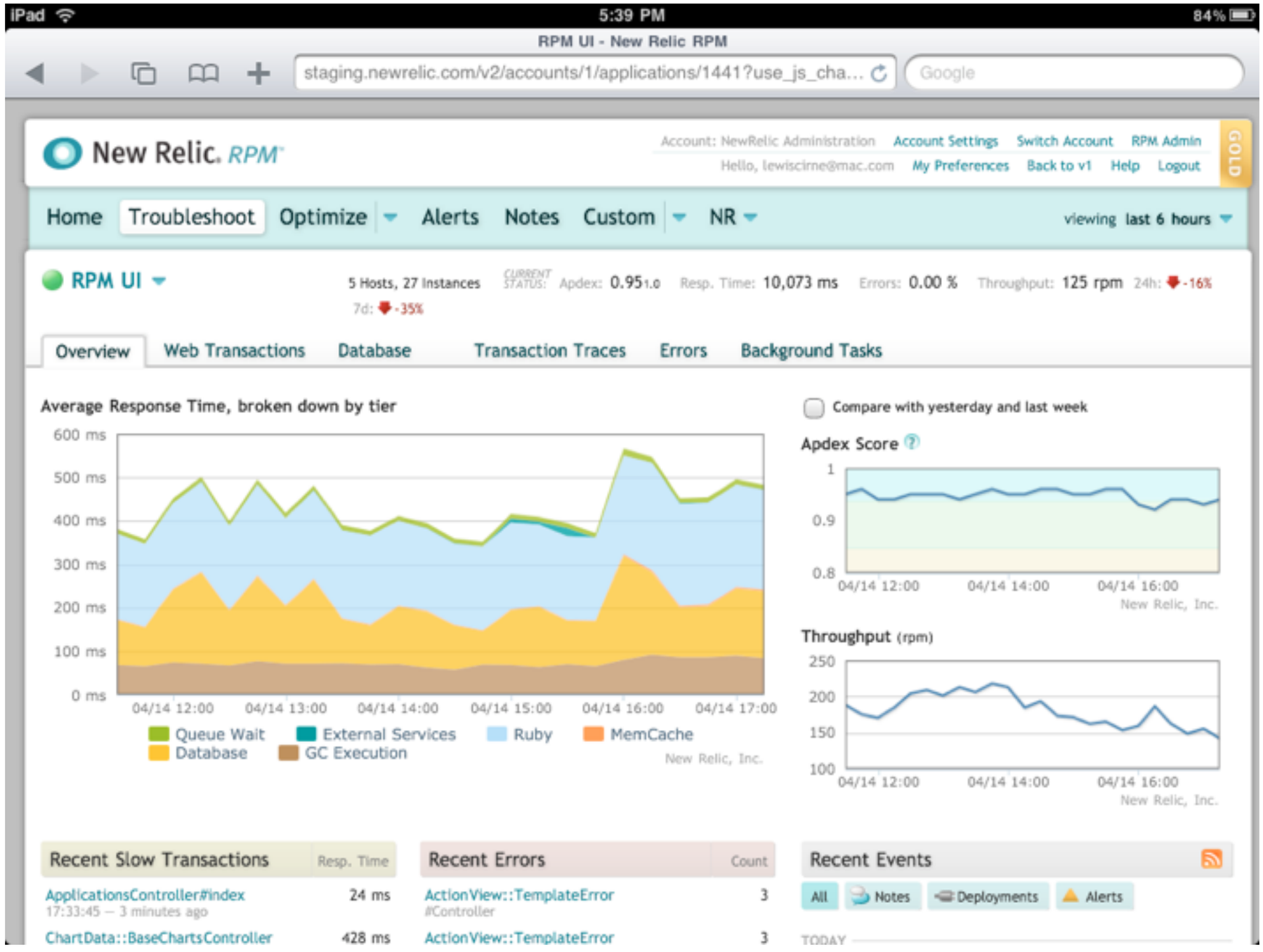
[View Full Callgraph]

Displaying top 100 functions: Sorted by Excl. CPU (microsec) [display all]

Function Name	Calls	Calls%	Incl. Wall Time (microsec)	IWall%	Excl. Wall Time (microsec)	EWall%	Incl. CPU (microsecs)	ICpu%	Excl. CPU (microsec)	ECPU%	Incl. MemUse (bytes)	IMemUse%	Ex MemU (byte
run_init::Dispatcher/Standard.php	1	0.1%	3,899	8.7%	174	0.4%	8,001	18.2%	4,001	9.1%	192,096	7.8%	7,2
load::Loader/PluginLoader.php	1	0.1%	1,099	2.5%	1,099	2.5%	4,001	9.1%	4,001	9.1%	70,956	2.9%	70,9
ucfirst	23	1.5%	18	0.0%	18	0.0%	4,000	9.1%	4,000	9.1%	2,368	0.1%	2,3
load::Bootstrap/Bootstrap.php	1	0.1%	310	0.7%	310	0.7%	4,000	9.1%	4,000	9.1%	7,992	0.3%	7,9
load::Controller/Action.php	1	0.1%	1,194	2.7%	1,194	2.7%	4,000	9.1%	4,000	9.1%	76,392	3.1%	76,3
run_init::Dispatcher/Abstract.php	1	0.1%	3,228	7.2%	1,178	2.6%	4,000	9.1%	4,000	9.1%	140,752	5.7%	1,7
load::Controller/Front.php	1	0.1%	1,657	3.7%	1,657	3.7%	4,000	9.1%	4,000	9.1%	108,104	4.4%	108,1
load::Loader/Autoloader.php	1	0.1%	1,304	2.9%	1,304	2.9%	4,000	9.1%	4,000	9.1%	74,624	3.0%	74,6
load::controllers/IndexController.php	1	0.1%	188	0.4%	188	0.4%	4,000	9.1%	4,000	9.1%	10,744	0.4%	10,7
load::Router/Route.php	1	0.1%	1,334	3.0%	1,334	3.0%	4,000	9.1%	4,000	9.1%	81,500	3.3%	81,5
load::Request/Http.php	1	0.1%	1,886	4.2%	1,886	4.2%	4,000	9.1%	4,000	9.1%	135,104	5.5%	135,1
Zend_Controller_Plugin_Abstract::preDispatch	1	0.1%	1	0.0%	1	0.0%	0	0.0%	0	0.0%	480	0.0%	4
Zend_Controller_Dispatcher_Abstract::getParam	4	0.3%	6	0.0%	6	0.0%	0	0.0%	0	0.0%	1,572	0.1%	1,5
Zend_Controller_Dispatcher_Standard::isDispatchable	1	0.1%	438	1.0%	43	0.1%	0	0.0%	0	0.0%	16,188	0.7%	1,0
Zend_Controller_Request_Abstract::setDispatched	2	0.1%	4	0.0%	4	0.0%	0	0.0%	0	0.0%	984	0.0%	9
Zend_Controller_Plugin_Broker::preDispatch	1	0.1%	8	0.0%	7	0.0%	0	0.0%	0	0.0%	1,096	0.0%	6

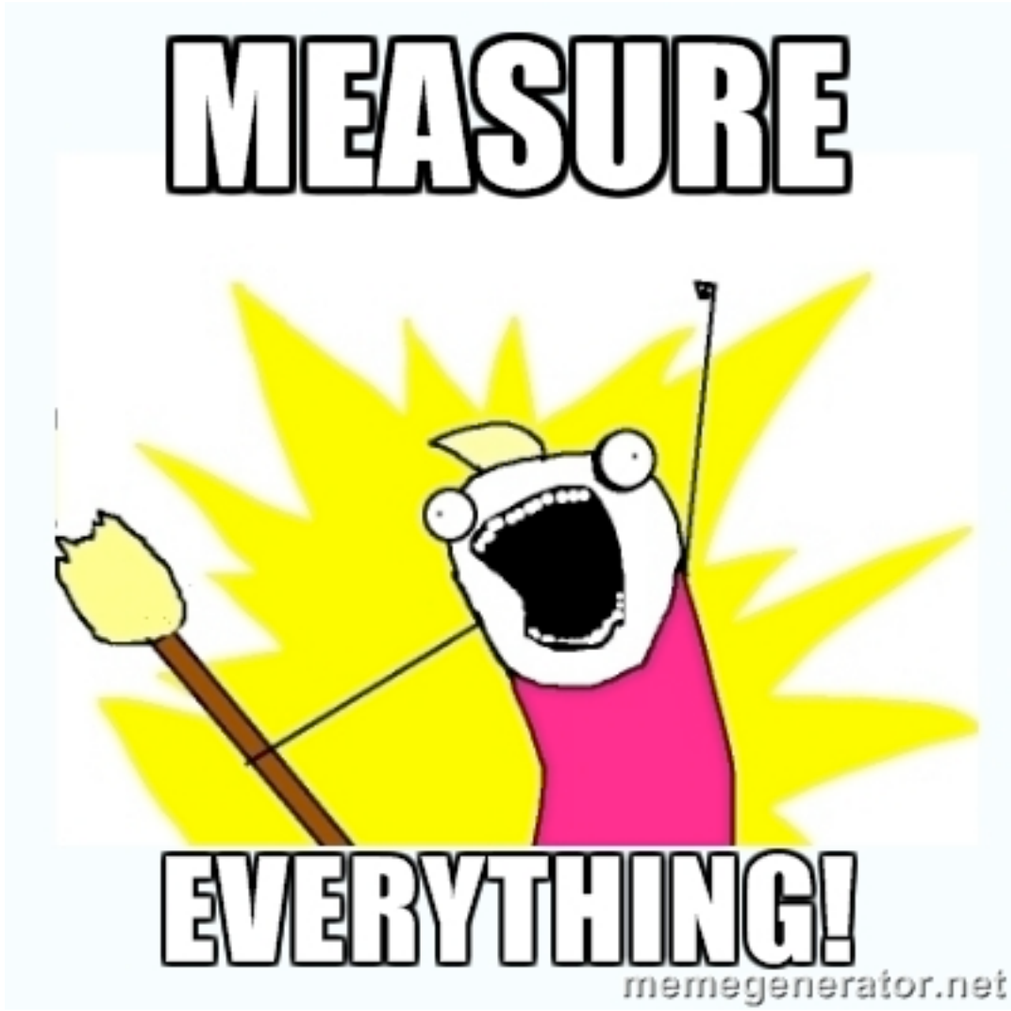


New Relic


iPad
5:39 PM
84%
RPM UI - New Relic RPM
staging.newrelic.com/v2/accounts/1/applications/1441?use_js_cha...
Google
New Relic. RPM
Account: NewRelic Administration
Account Settings
Switch Account
RPM Admin
Hello, lewiscirne@mac.com
My Preferences
Back to v1
Help
Logout
GOLD
Home
Troubleshoot
Optimize
Alerts
Notes
Custom
NR
viewing last 6 hours
RPM UI
5 Hosts, 27 Instances
CURRENT STATUS:
Apdex: 0.95 1.0
Resp. Time: 10,073 ms
Errors: 0.00 %
Throughput: 125 rpm
24h: -16%
7d: -35%
Overview
Web Transactions
Database
Transaction Traces
Errors
Background Tasks
Average Response Time, broken down by tier
600 ms
500 ms
400 ms
300 ms
200 ms
100 ms
0 ms
04/14 12:00
04/14 13:00
04/14 14:00
04/14 15:00
04/14 16:00
04/14 17:00
Queue Wait
External Services
Ruby
MemCache
Database
GC Execution
New Relic, Inc.
Compare with yesterday and last week
Apdex Score
1
0.9
0.8
04/14 12:00
04/14 14:00
04/14 16:00
New Relic, Inc.
Throughput (rpm)
250
200
150
100
04/14 12:00
04/14 14:00
04/14 16:00
New Relic, Inc.
Recent Slow Transactions
Resp. Time
ApplicationsController#index
17:33:45 — 3 minutes ago
24 ms
ChartData::BaseChartsController
428 ms
Recent Errors
Count
ActionView::TemplateError
#Controller
3
ActionView::TemplateError
3
Recent Events
All
Notes
Deployments
Alerts
TODAY

NEWRELIC POKAŻE
KTÓRE KOMPONENTY
DOMINUJĄ OBCIĄŻENIE

METRYKI


MEASURE
EVERYTHING!
memegenerator.net

STATSD

```php
<?php

$start = microtime(true);
wyślij_maila();
StatsD::timing("podsystemy.mail", (microtime(true) - $start) * 1000);

?>
```

MIN, MAX, 90th, COUNT, MEAN

GRAPHITE

graphite

Command-Line Interface
Logged in as cdavis, logout (edit profile)
Documentation

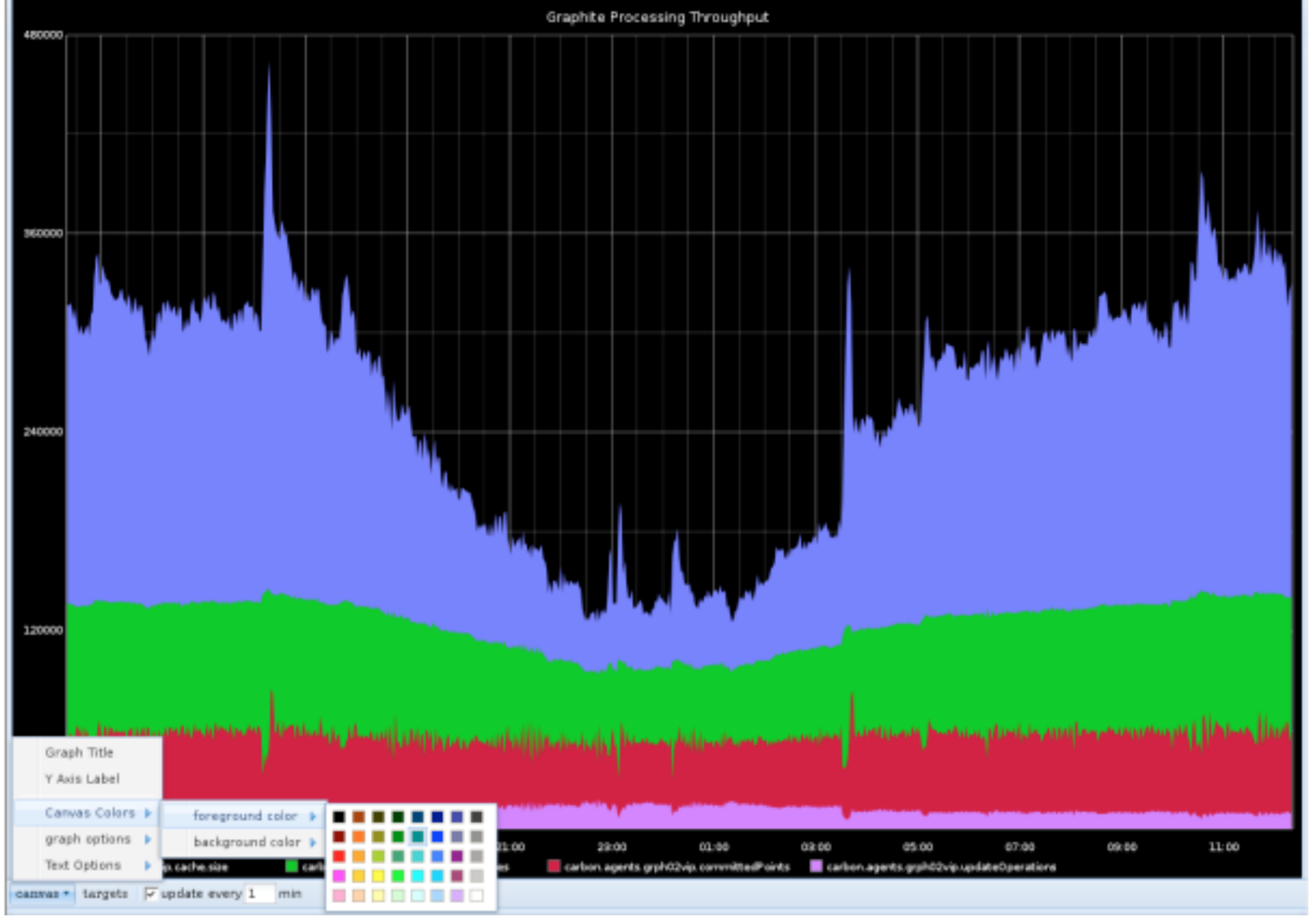
production
pre-production

Tree | Search | Auto Completer

- Graphite
  - *
  - PRO2
  - Perf
  - afe
  - avs
  - carbon
    - *
    - agents
      - grph01vip
        - *
          - avgUpdateTime
          - cache
          - committedPoints
          - creates
          - errors
          - pointsPerUpdate
          - updateOperations
      - grph02vip
        - *
          - avgUpdateTime
          - cache
            - *
              - queries
              - queues
              - size
          - committedPoints
          - creates
          - errors
          - pointsPerUpdate
          - updateOperations
      - pypedUtilization
      - pypes
  - database
  - hosting
  - lds
  - ods
- My Graphs
  - myTest
  - test123
- User Graphs

Graphite Processing Throughput

FROM: - range UNTIL: - range




graphite>load c3
Loading view c3
graphite>draw PROD.streambase.tbs-shop.all.jiniIn.
PROD.streambase.tbs-shop.all.jiniIn.LocationFinderService.
PROD.streambase.tbs-shop.all.jiniIn.SessionEventService.
PROD.streambase.tbs-shop.all.jiniIn.ShopService.
PROD.streambase.tbs-shop.all.jiniIn.stdevLatency
PROD.streambase.tbs-shop.all.jiniIn.avgLatency
PROD.streambase.tbs-shop.all.jiniIn.count
PROD.streambase.tbs-shop.all.jiniIn.countFailed
PROD.streambase.tbs-shop.all.jiniIn.maxLatency
PROD.streambase.tbs-shop.all.jiniIn.minLatency
commits
carbon.agents.grph01vip.committedPoints
carbon.agents.grph02vip.committedPoints
consumption
carbon.pypes.egpype02_prod_o_com_2002.consumedMessages
carbon.pypes.egpype01_prod_o_com_2002.consumedMessages
caches
carbon.agents.grph01vip.cache.queues
carbon.agents.grph02vip.cache.queues
carbon.agents.grph01vip.cache.size
carbon.agents.grph02vip.cache.size
pdateLatency
carbon.agents.grph01vip.avgUpdateTime
carbon.agents.grph02vip.avgUpdateTime
updates
carbon.agents.grph01vip.updateOperations
carbon.agents.grph02vip.updateOperations
cacheQueries
carbon.agents.grph01vip.cache.queries
carbon.agents.grph02vip.cache.queries
receives
carbon.pypes.egpype02_prod_o_com_2002.logicalMessages
carbon.pypes.egpype01_prod_o_com_2002.logicalMessages
pointsPerUpdate
carbon.agents.grph01vip.pointsPerUpdate
carbon.agents.grph02vip.pointsPerUpdate

ANALIZA LOGÓW:
LOGSTASH, GRAYLOG2

PHP

ZAKTUALIZUJ **PHP**

STANDARD NA **2014**:
NGINX + PHP-FPM

NGINX:

SZYBKOŚĆ, ELASTYCZNOŚĆ

PHP-FPM:

ROZDZIELENIE PROCESÓW

WŁĄCZ **OPCODE CACHE**

ZOPTYMALIZUJ

FRAMEWORK

WŁĄCZ **CACHE**

WYŁĄCZ

AUTOGENEROWANIE

WYŁĄCZ **ZBĘDNE**

MODUŁY

OPTYMALIZACJA
BAZY DANYCH

GŁÓWNA PRZYCZYNA
PROBLEMÓW Z
WYDAJNOŚCIĄ

SLOW LOG?

PERCONA TOOLKIT

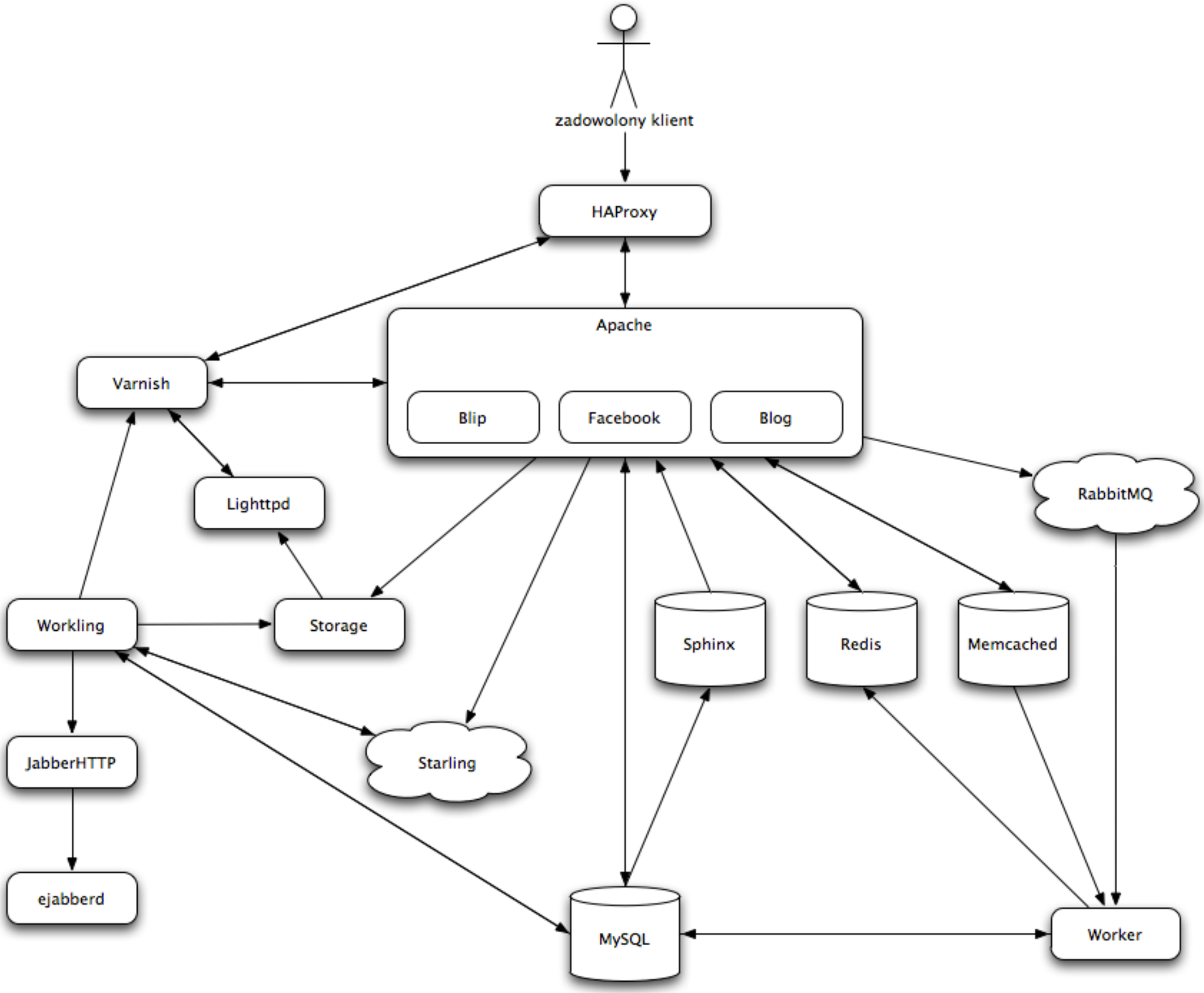
INDEKSY

QUERY CACHE

ALTERNATYWNE SILNIKI

NOSQL

ARCHITEKTURA


zadowolony klient
HAProxy
Apache
Varnish
Blip
Facebook
Blog
Lighttpd
RabbitMQ
Workling
Storage
Sphinx
Redis
Memcached
JabberHTTP
Starling
ejabberd
MySQL
Worker

DOBIERZ
KOMPONENTY

FULL TEXT SEARCH?

SPHINX.

STRUKTURY
DANYCH? **REDIS.**

BUFORUJ DANE

- WYNIKI ZAPYTAŃ
- **ODPOWIEDZI API**
- KONFIGURACJA
- **GENEROWANE DANE**

NAJCZĘŚCIEJ
UŻYWANE:
W PAMIĘCI

MEMCACHE

MOŻLIWOŚĆ WYDZIELENIA **WSPÓŁDZIELONEGO** SERWERA MEMCACHE

VARNISH

NAGŁÓWKI **HTTP**

bit.ly/httpcache

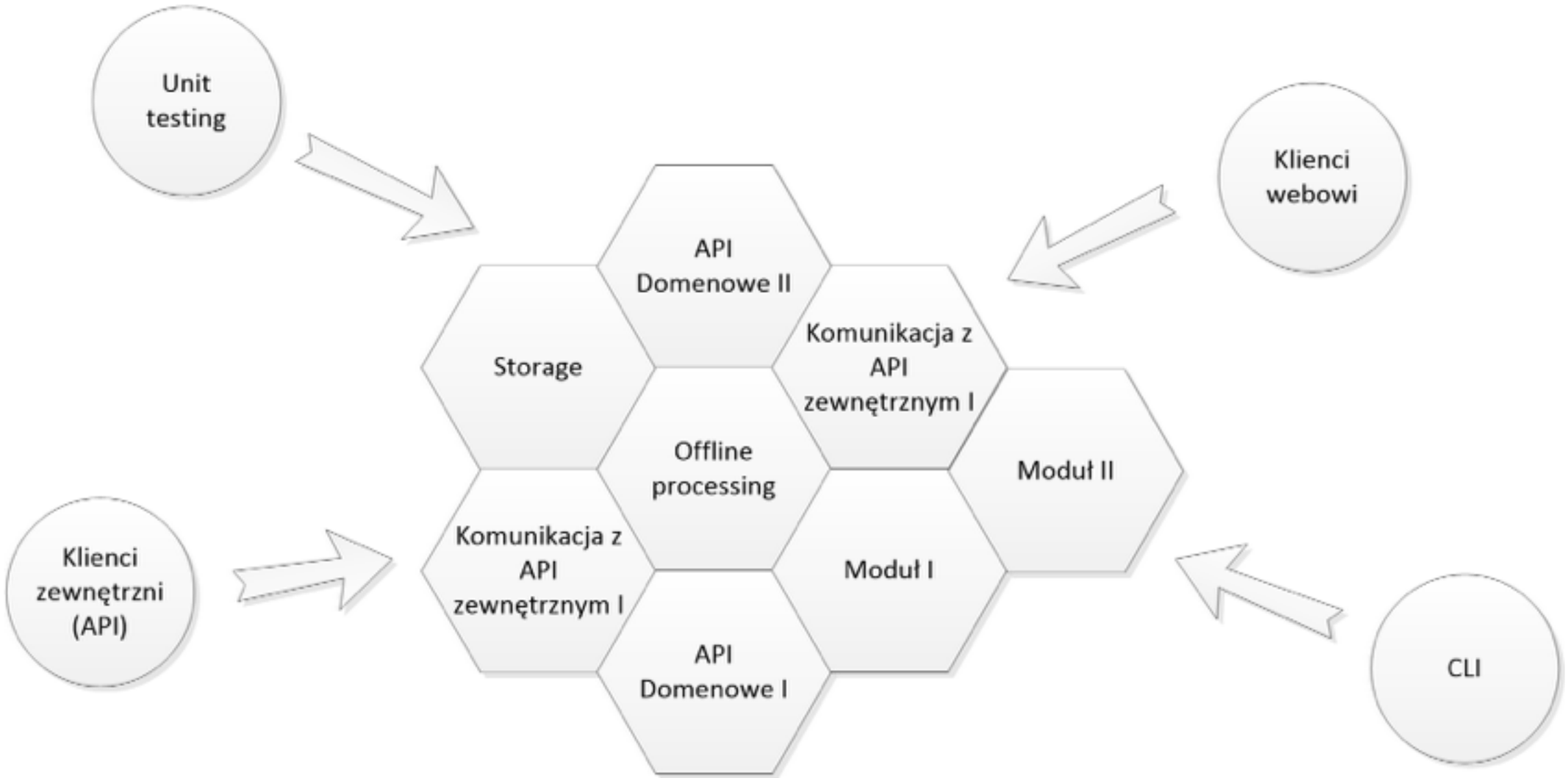
MAX AGE

LAST MODIFIED

ETAG

ROZDZIEL
SKŁADNIKI

ARCHITEKTURA
HEKSAGONALNA


Unit testing
Klienci webowi
API Domenowe II
Storage
Komunikacja z API zewnętrznym I
Offline processing
Moduł II
Klienci zewnętrzni (API)
Komunikacja z API zewnętrznym I
Moduł I
API Domenowe I
CLI

PRZETWARZAJ

ASYNCHRONICZNIE

0.0 Przyjmij request

0.1 Przeanalizuj dane

0.2 Wyślij mailing

...

...

...

...

...

...

...

...

5.2s Wyświetl wyniki

GDZIEŚ TU UŻYTKOWNIK KLIKA “RELOAD”

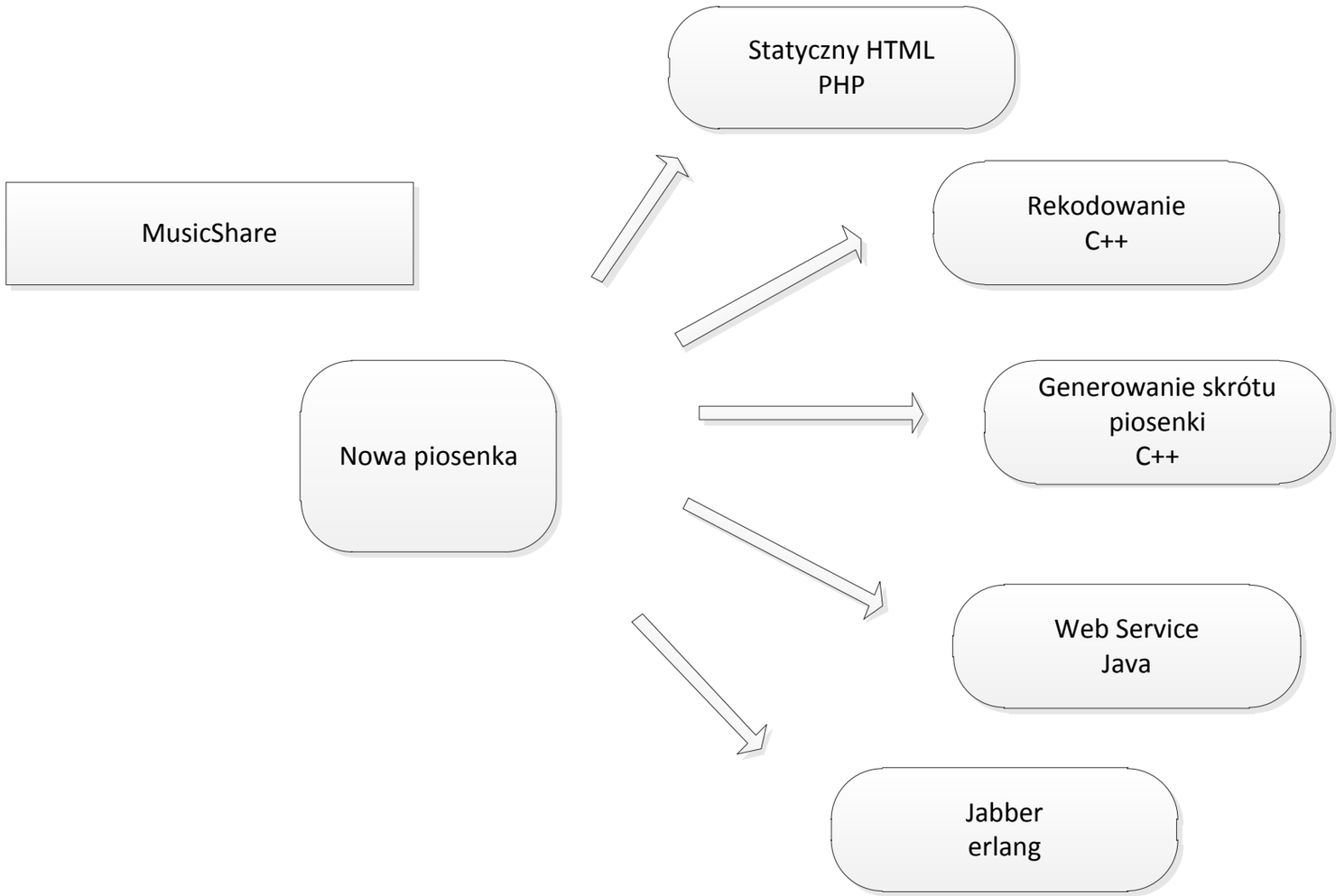
Aplikacja

0.0 Przyjmij request
0.1 Przeanalizuj dane
0.2 Przekaż dane do kolejki
0.3 Wyświetl komunikat

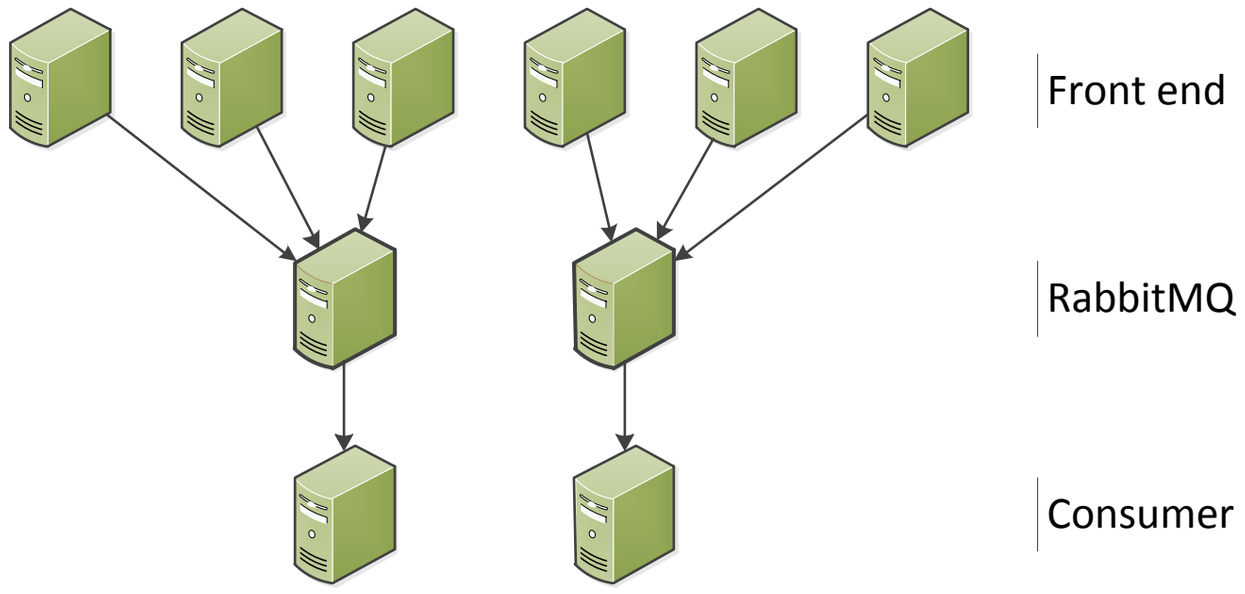
Kolejka

0.0 Przyjmij dane
0.1 Wyślij mailing
...
...
...
...
...
5.0 Zapisz informację o wyniku

DECOUPLING


Statyczny HTML
PHP
MusicShare
Rekodowanie
C++
Nowa piosenka
Generowanie skrótu
piosenki
C++
Web Service
Java
Jabber
erlang

LOAD **BALANCING**



POZA **JEDEN** SERWER

LOAD BALANCER

VARNISH

HAPROXY

NGINX

PROBLEMY

DANE

WSPÓŁDZIELONE

SESJE

PERSISTENT
LOAD BALANCER

WSPÓLNY

MEMCACHE

INNE DANE

PLIKI **STATYCZNE**

WSPÓŁDZIELONY
SYSTEM PLIKÓW

API

ARCHITEKTURA DLA
BAZ DANYCH

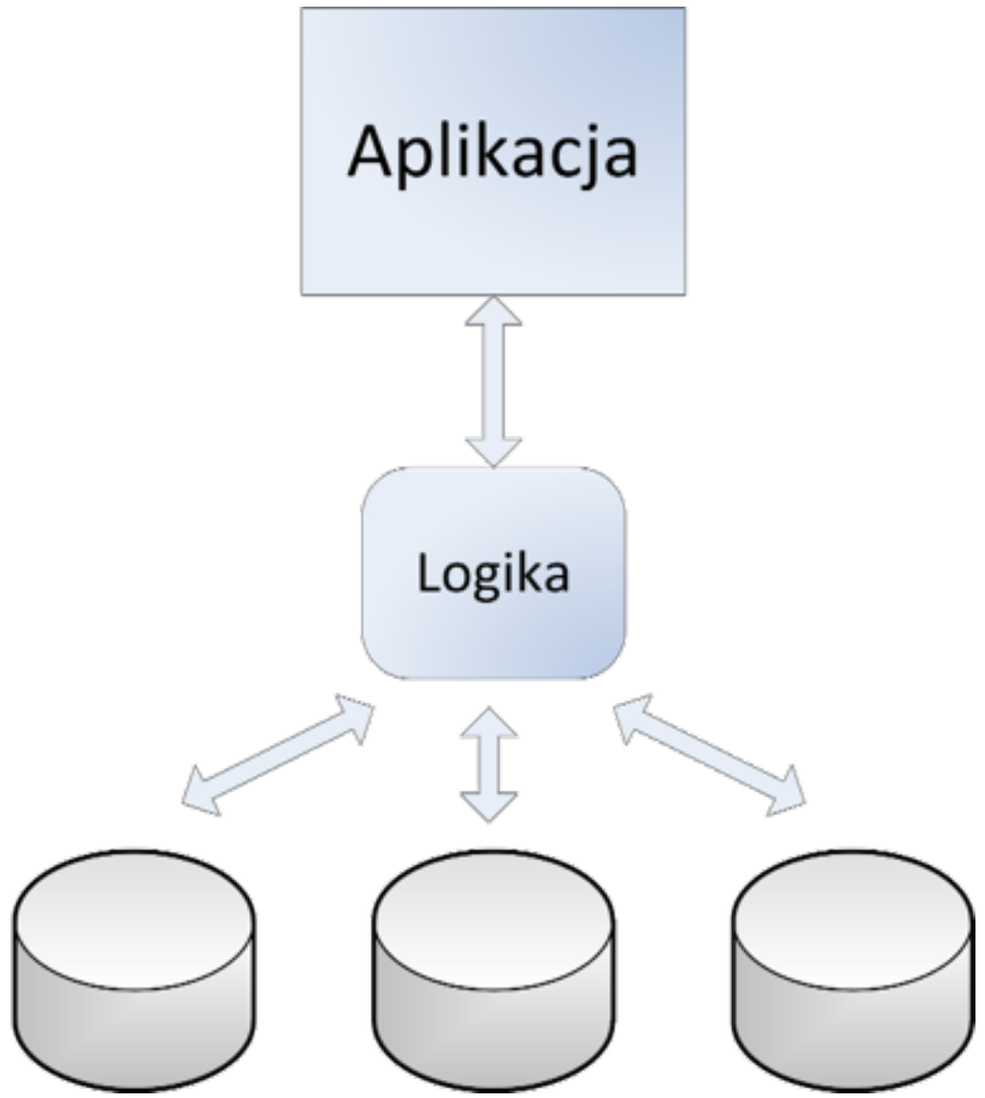
REPLIKACJA

MASTER-SLAVE

SHARDING

ID	First name	Last Name
1	Jan	Kowalski
2	Anna	Kwiatkowska
3	Zenon	Jaskóła
4	Józef	Nowak
5	Krzysztof	Boberek
6	Witold	Kwiecień

ID	First name	Last Name	Shard
1	Jan	Kowalski	1
2	Anna	Kwiatkowska	2
3	Zenon	Jaskóła	3
4	Józef	Nowak	1
5	Krzysztof	Boberek	2
6	Witold	Kwiecień	3


Aplikacja
Logika

PROBLEMY

RESHARDING

JOIN

NA **KONIEC**

NIE OPTYMALIZUJ

PRZEDWCZEŚNIE

BUDUJ ARCHITEKTURĘ

PERSPETYWICZNIE

ŻYCZĘ WSZYSTKIM
TAKICH PROBLEMÓW
:-)

https://joind.in/11839

PYTANIA?

DZIĘKUJĘ
ZA UWAGĘ