

# Scaling Kamona



Bruno de Jesus Braga, [bdejesus@nd.edu](mailto:bdejesus@nd.edu)  
Fernando Beletti, [fbeletti@nd.edu](mailto:fbeletti@nd.edu)

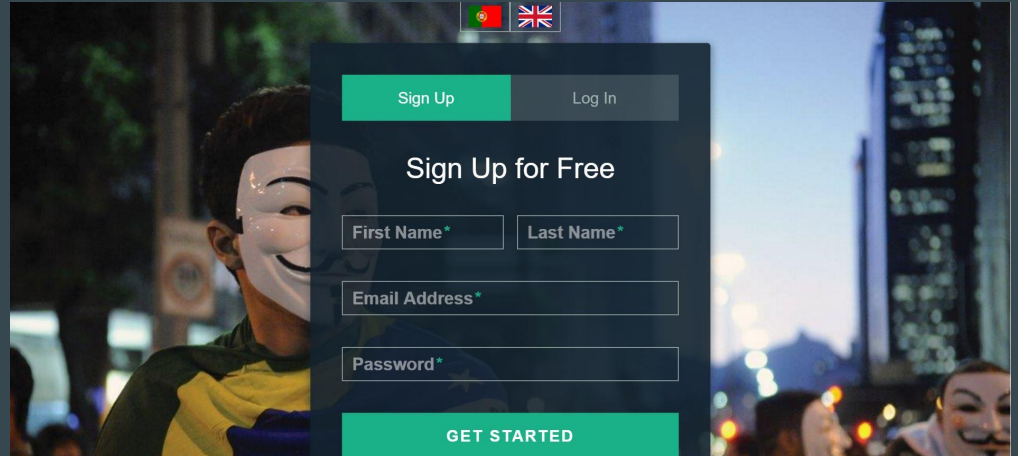
# What is Kamona?

- ❖ Kamona is a discussion website, in which users can discuss about a particular subject and propose solutions to that problem.

<b>34</b> votes	<b>56</b> views	<b>What do you think about public education?</b> <small>Created by: Fernando Beletti, March 2016</small>
<b>2</b> votes	<b>45</b> views	<b>Should full time public schools be added on our education?</b> <small>Created by: Paulo Braga, may 2016</small>
<b>99</b> votes	<b>87</b> views	<b>Political parties positions were added into public school books!</b> <small>Created by: Bruna Surfistinha, August 2016</small>

# What are the options?

- ❖ Create a new discussion topic
  - Education
  - Politics
  - Economics
  - Security
- ❖ Participate on a discussion
- ❖ Post comments and Images
- ❖ Be part of good and organized discussions



# Overview

- ❖ The website aims brazilian economics/political issues.
- ❖ There is a growing brazilian community around the US and Europe.
- ❖ Kamona is hosted in Brazil, accessing the website from these locations results in big latency and low performance.
- ❖ As seen in classes we can use Amazon Web Services to distribute the website and solve these latency problems.
- ❖ The database side uses Firebase API, that scales itself automatically.

# Challenges

- ❖ Implement a cache to make queries faster
- ❖ Memcached: Based on get/set functions

When reading:

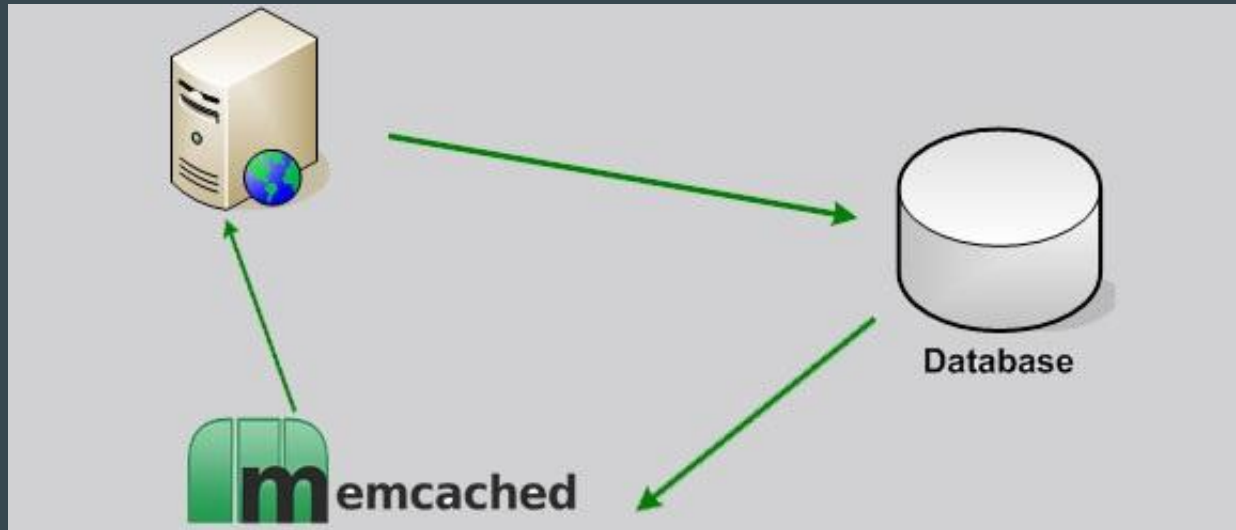
1. Check if value is on cache.
2. If it is, use it, if not get from database.

When writing

1. Update value on cache
2. Update value on database

# Node.js Memcached

- ❖ Fully featured memcached for node.js
- ❖ Easy to implement and to install
- ❖ Combined with CloudFront, could create an even faster experience for users



# But it did not work.

```
1  * Created by brunobraga on 4/20/16.
2  */
3
4  var http = require('net');
5  var Memcached = require('memcached');
6  var memcached = new Memcached('localhost:11211');
7  var lifetime = 86400; //24hrs
8  var Firebase = require("firebase");
9
10 http.createServer(function (req, res) {
11     memcached.get('snap', function( err, result ){
12         if( err ) console.error( err );
13         if(!result){
14             var Users = new Firebase("https://brunobraga.firebaseio.com/kamona/users");
15             var snap;
16             firebase.once("value", function(snapshot) {
17                 snap = snapshot.val();
18                 memcached.set('snap', JSON.stringify(snap), lifetime, function( err, result ){
19                     if( err ) console.error( err );
20                     console.dir( result );
21                 });
22             });
23         }
24         else snap = result;
25     });
26     ;
27
28     }).listen(1337, '0.0.0.0');
29     console.log('Server running at http://127.0.0.1:1337/');
```

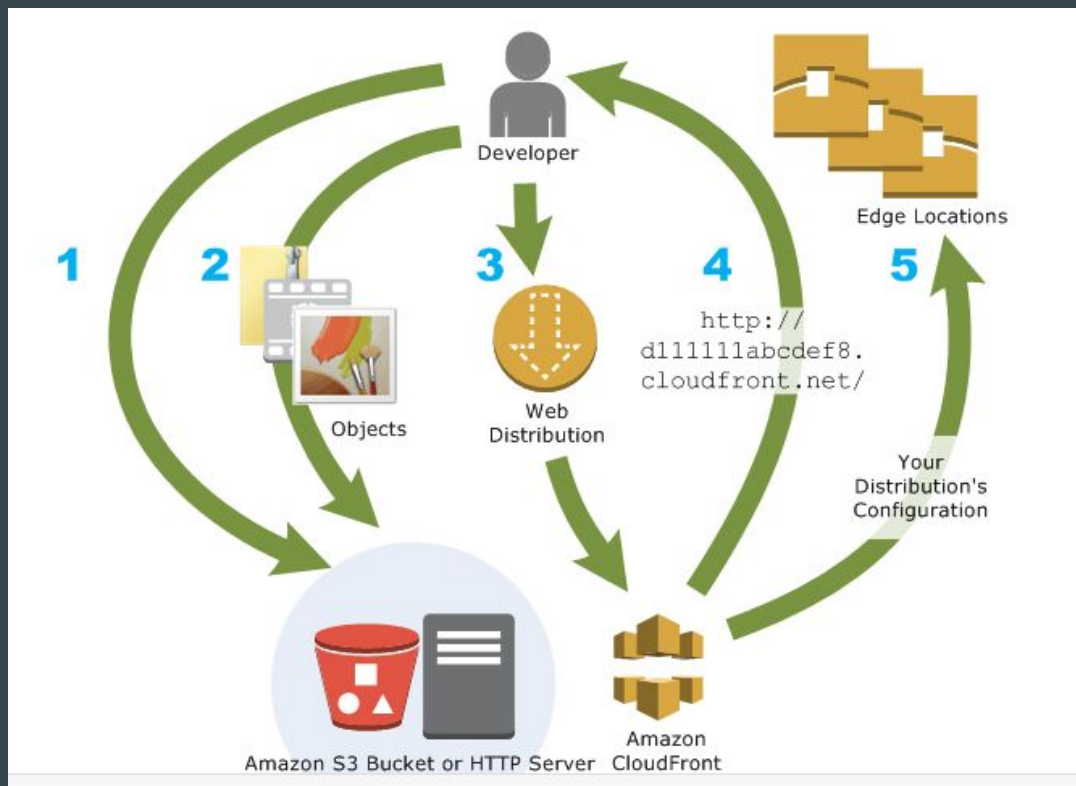
# Solutions

- ❖ Initially all the static data from the website was moved to S3 servers so users could make faster requests to this files.
- ❖ In a second moment, all static files were moved to CloudFront, resulting in even faster access to the website.
- ❖ Currently the website is hosted using Amazon Web Services
- ❖ The domain provided by Amazon was:

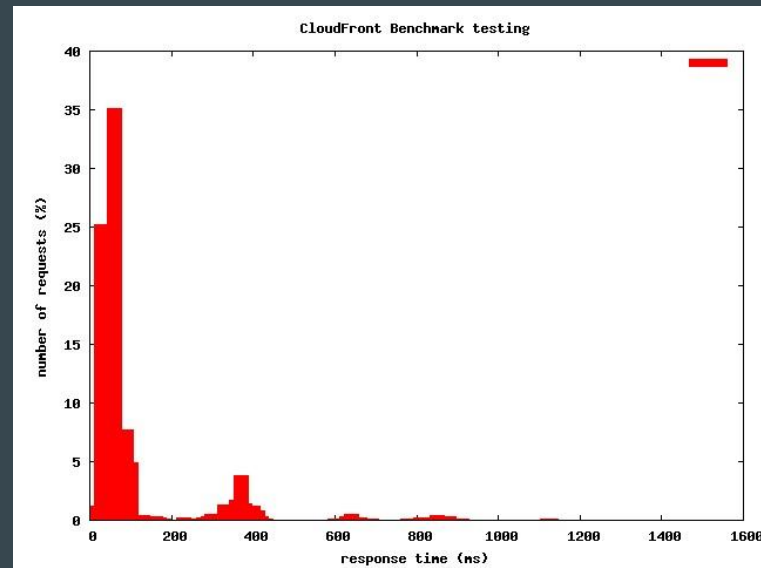
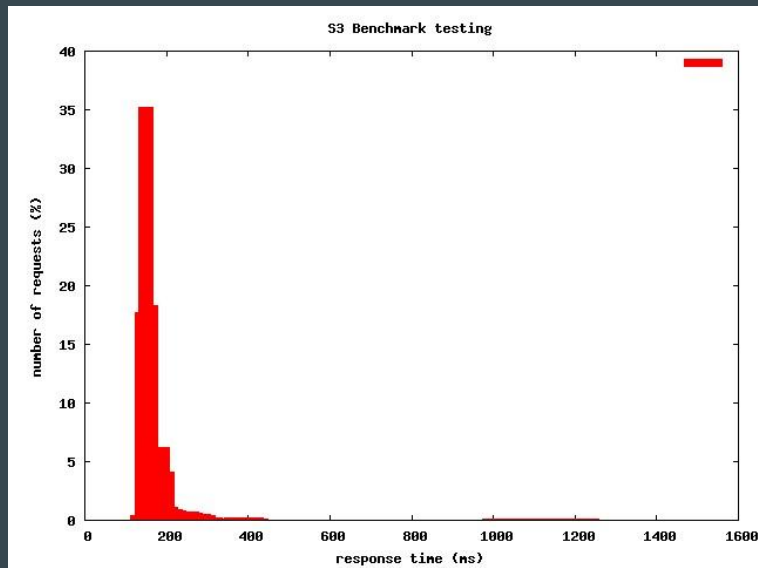
<http://dlinijhv8u5dmq.cloudfront.net/index.html>



# Structure



# Results



ab -k -c 300 -n 18000 \$URL  
Ireland -> US West

# Results



# Results

	Single User	Single User
	S3	CloudFront
Visually complete	4,6 s	3,6 s
Time to first byte	807 ms	120 ms
Time to start render	2093 ms	903 ms

Simulated using Chrome connection from Ireland client

Tool used: <http://www.webpagetest.org/>

# Conclusion and Future Work

- ❖ Using AWS to distribute the website increased the performance for users far from the original host.
- ❖ Elastic Load Balancing can be used in the future to scale the website automatically, creating new instances as the website suffers from traffic increase.
- ❖ Memcached or other caching technologies can be added to improve users experience.