

# Operations in the Cloud

Marius Stuparu  
DevOps @ SDL Language Weaver

# Summary

The Cloud, AWS Web Services

Best Practices in the Cloud

High Availability and Scalability in the Cloud

Alternative Open Source Solutions

# What is cloud computing?



IaaS, PaaS, SaaS



**amazon**  
web services™

# Who is using it?



# Why?

- Zynga thought 200 thousand daily active users on Farmville would count as a success (2009).
- 1 million new net users every single week
- Within a year, FarmVille had more than 50 million monthly active users
- CityVille on AWS was able to scale up to **~60 million** active monthly users in the first 2 months!!!

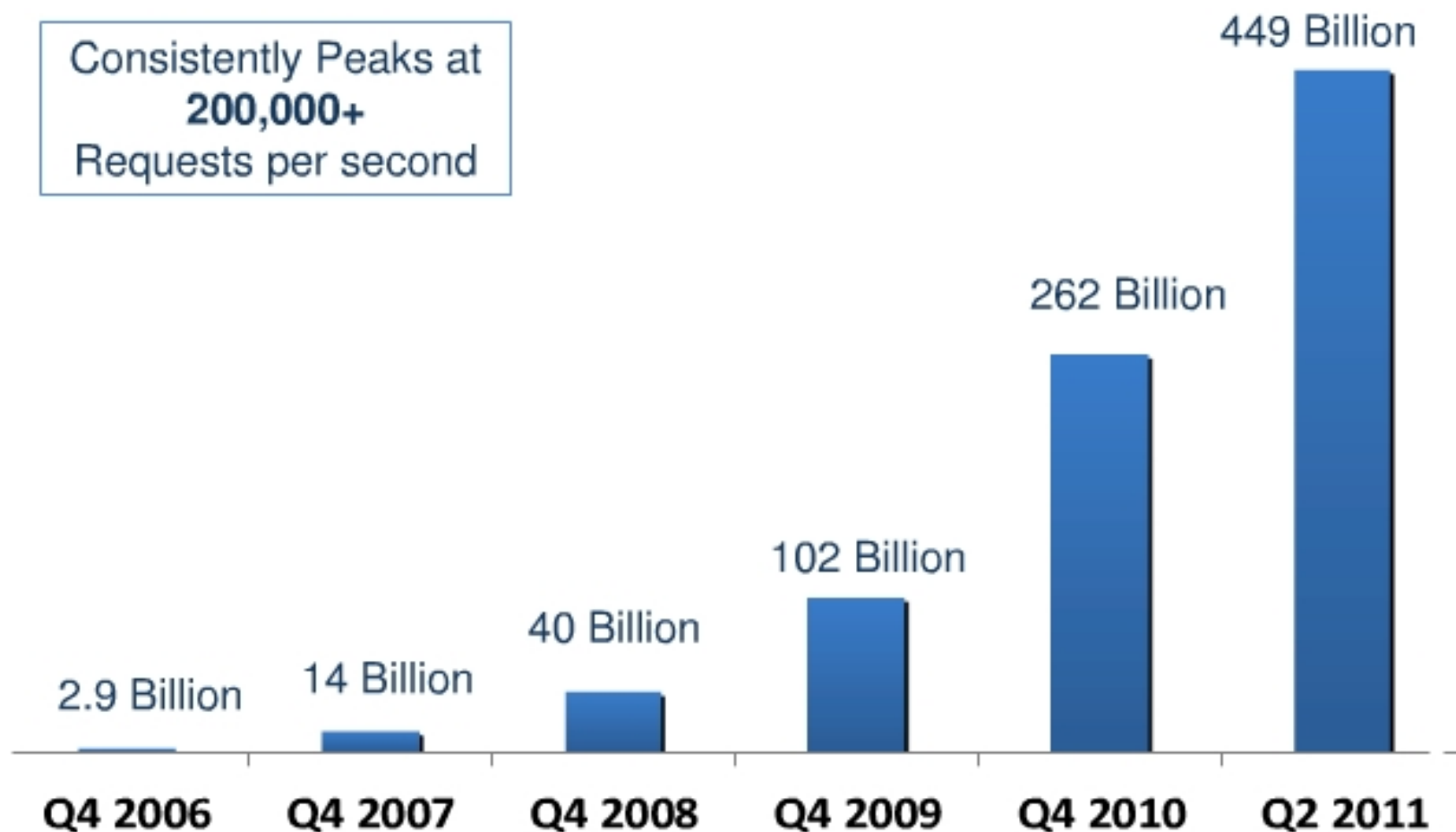
# AWS

- Amazon started really simple
- Ec2 - Virtual Machines service
- S3 - Storage service



# The Cloud Scales: Amazon S3 Growth

Consistently Peaks at  
**200,000+**  
Requests per second



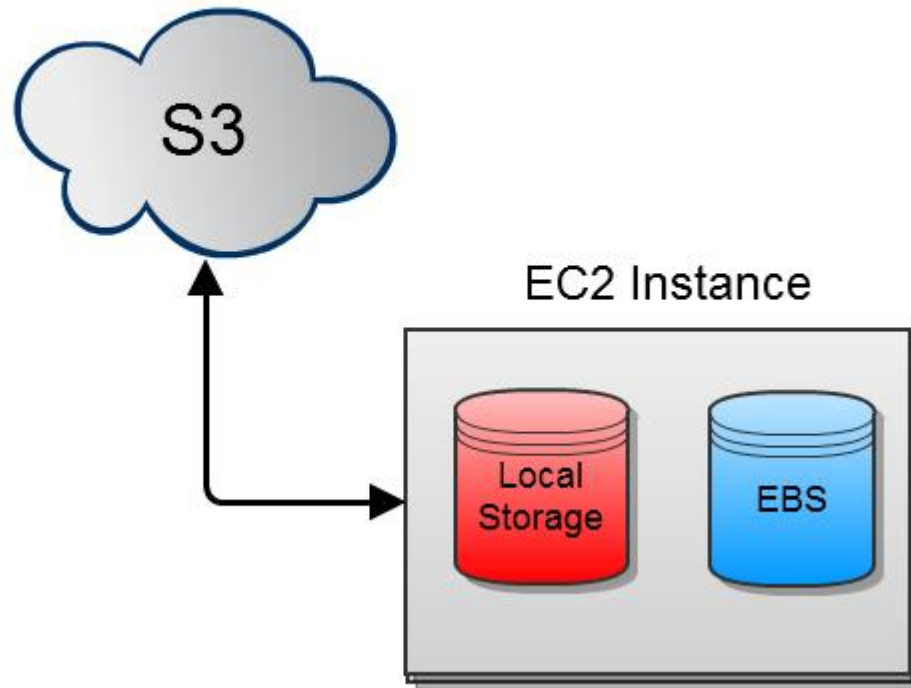
Total Number of Objects Stored in Amazon S3



# Elastic Block Store(EBS)

- local storage is volatile
- use EBS for persistent storage(network accessible block storage volumes).
- try to keep persistent data on S3 or RDS.  
EBS performance varies

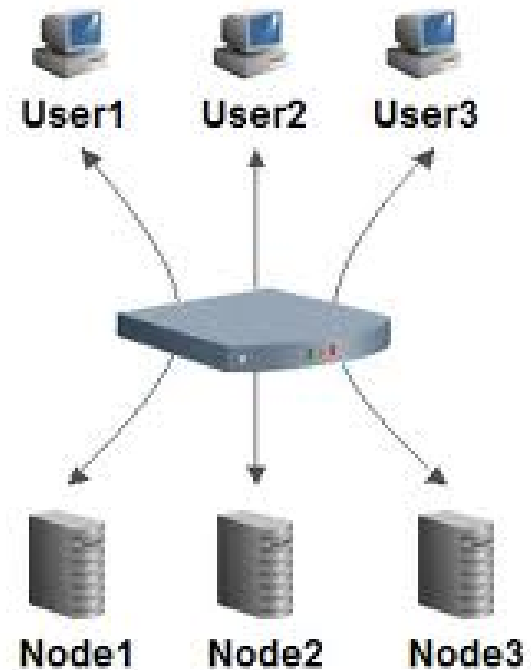
# Elastic Block Store(EBS)



# Elastic Load Balancers

Cool things:

- ELB- Availability zones
- SSL termination



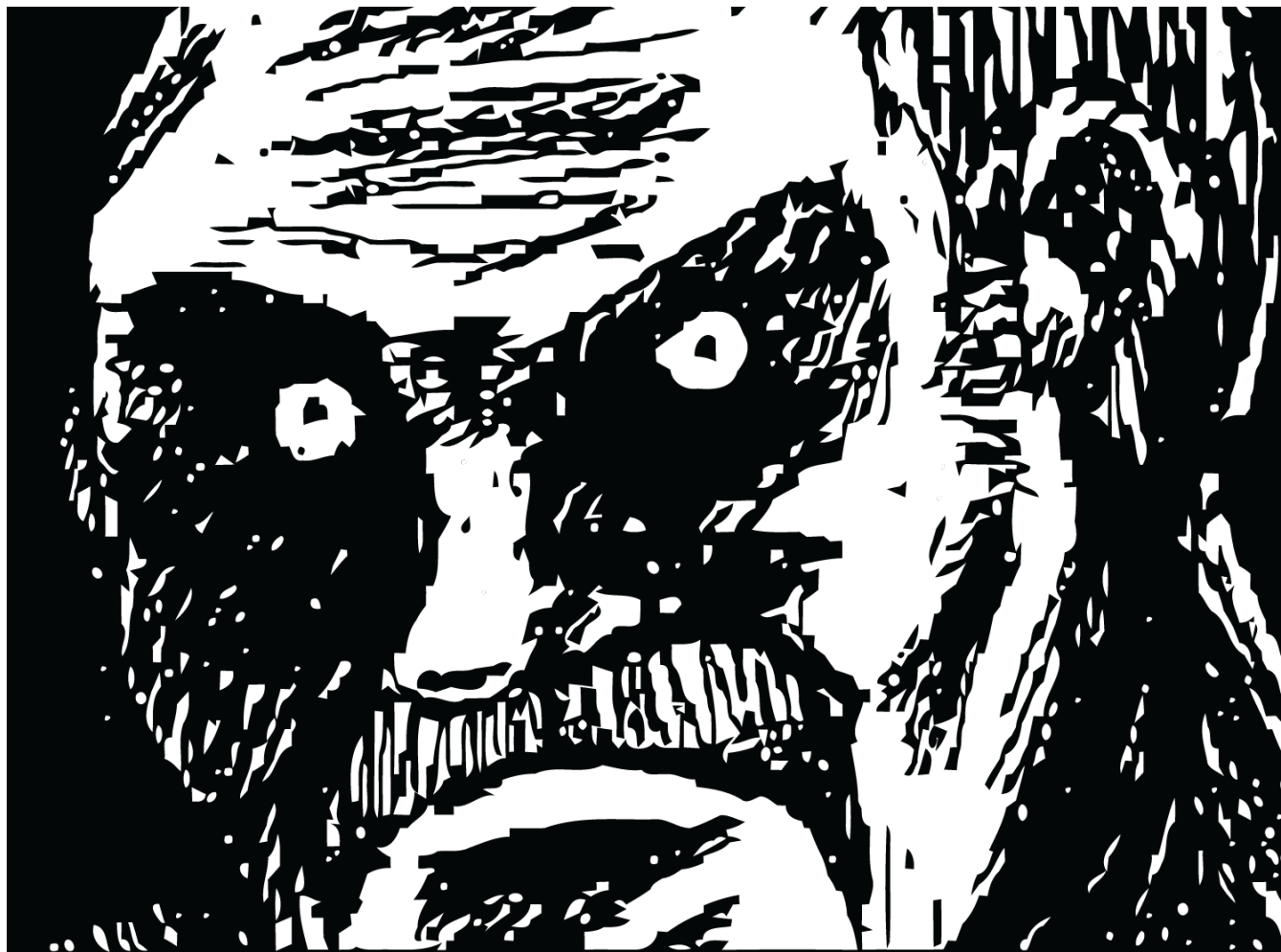
# Relational Database Service

- RDS (Multi AZ availability, fail-over ~5min)
- Easy to launch replicas and offload read traffic (3 clicks away)
- Backup using PITR, Snapshots

# RIGHT SCALE<sup>®</sup>

- Infrastructure as code
- Configuration management
- Orchestration
- Automated Provisioning/Auto Scaling
- Repeatable/Reproducible Servers  
(cloning your servers)

# April 21 2011 Server Down



# Eliminate Single Point of Failures

- architect around these problems
- decouple your components (queues)
- build asynchronous systems and scale horizontally
- make your applications as stateless as possible
- use multiple cloud providers (AWS, RackSpace, GoGrid, Linode.)

# Netflix Chaos Monkey





# All about being fast

- Cache database responses, objects, fully formed html (ElastiCache)
- **CDN**
- Follow the Sun
- Try to touch metal only when necessary, use local storage or SAN, avoid NFS

# DNS Management

- Route 53 LBR
- LBR = Latency Based Routing

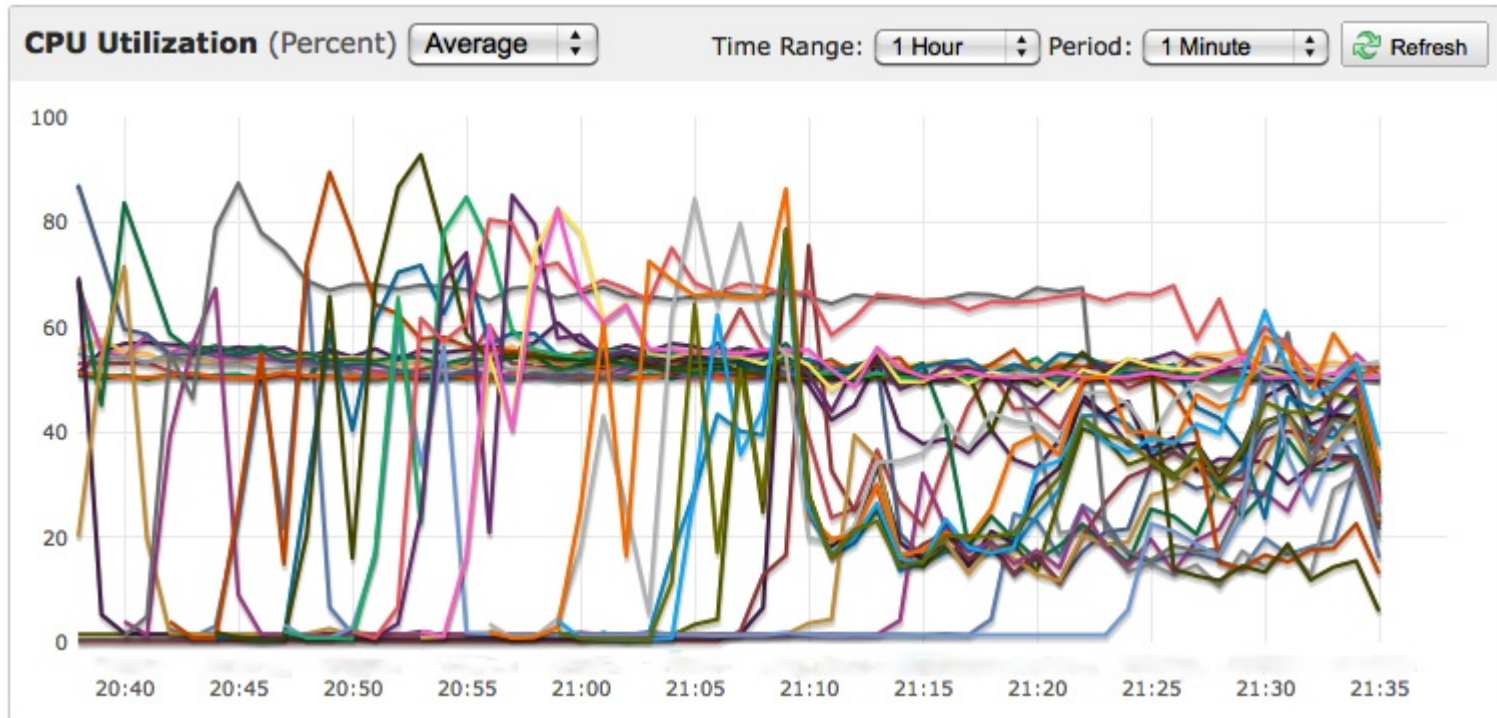
what it does:

- route your user to the closest server which runs your application

# Monitor and graph everything

- RightScale collectd, AWS CloudWatch
- NewRelic
- PingDom, Catch Point, Uptrends
- Nagios, Cacti, Zabbix
- Splunk

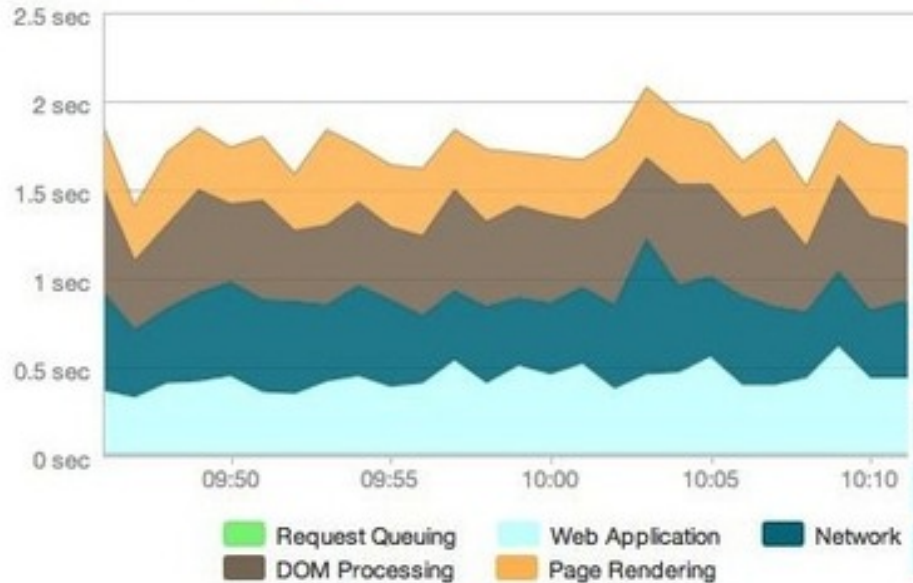
# AWS Cloudwatch



# New Relic - RUM

Browser page load time ?

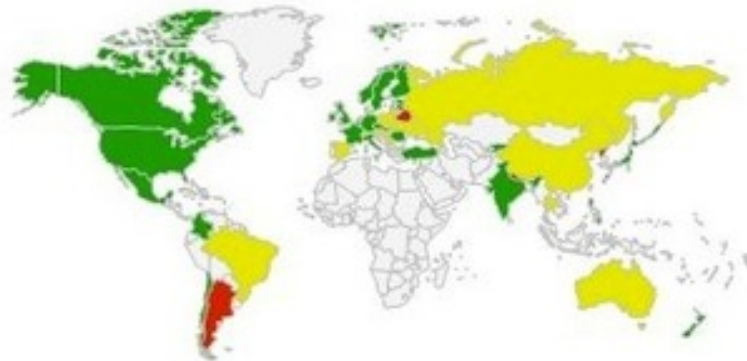
Average: 1.7 sec



Sort by

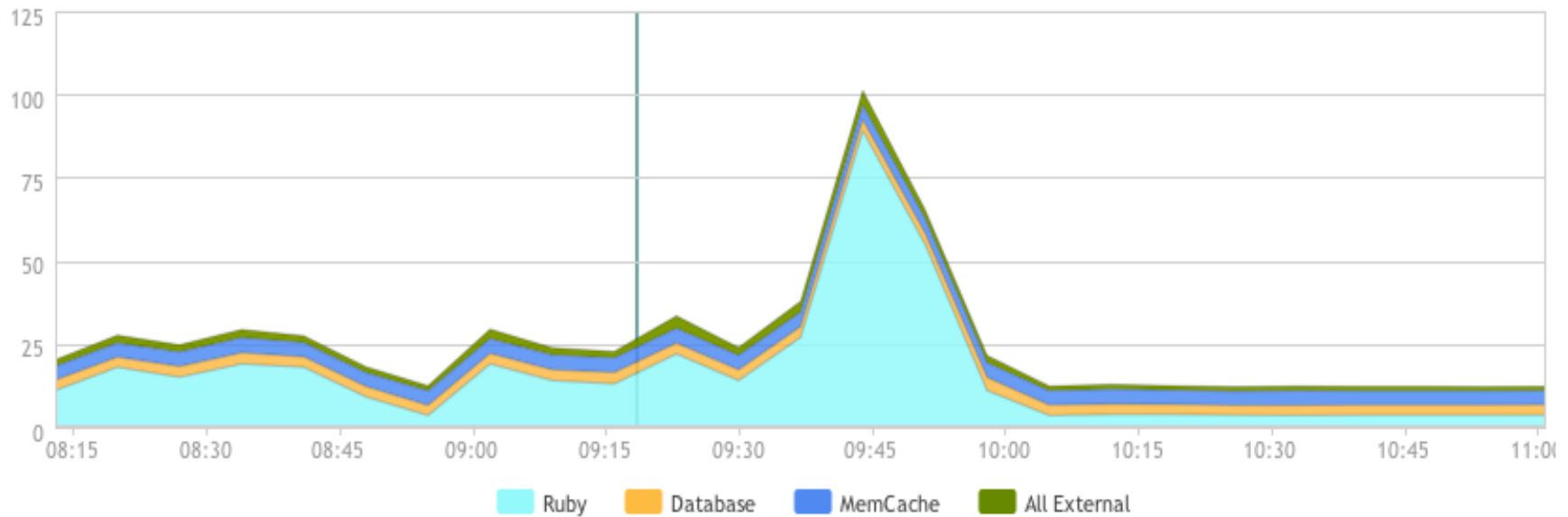
Average front-end time

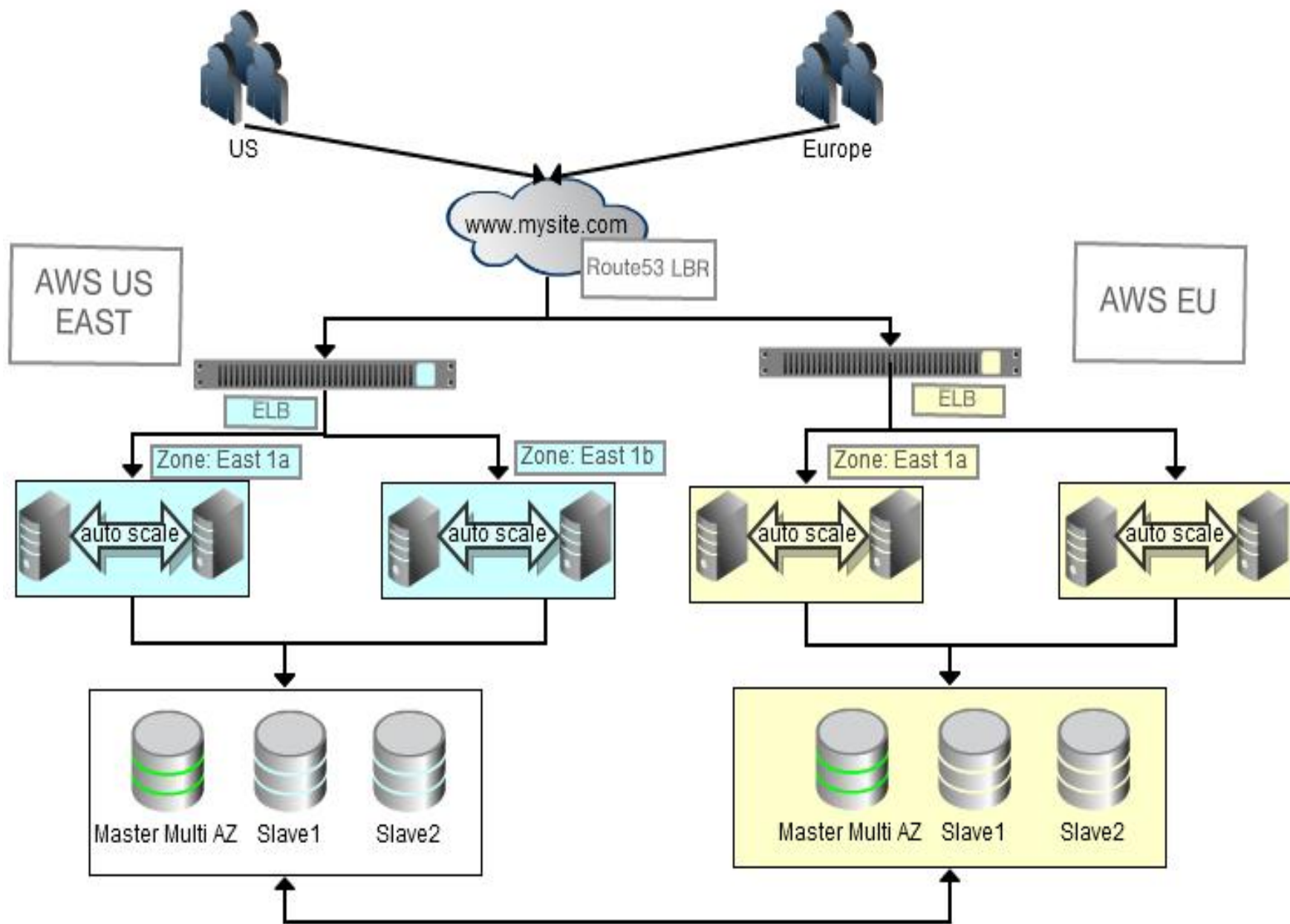
Windows IE 7	5.1 sec
Windows IE 8	3.6 sec
Windows Firefox 3.6	3.4 sec
Windows Firefox 4.0	3.4 sec
Windows IE 9	3.3 sec
Windows Chrome 11	3.0 sec
Mac Safari 5.0	2.6 sec
Mac Firefox 3.6	2.4 sec
Mac Firefox 4.0	1.8 sec
Mac Chrome 11	1.4 sec



# New Relic - Application Monitoring

Average response time, broken down by tier (ms)





# Open Source Alternatives



Foreman (no logo) - web ui for puppet



# Other useful tools

- Git (Github)
- Vagrant
- If you have a python stack look at boto  
(Python interface to Amazon Web Services)

**Q&A**

**Thank You!**

Slides: <http://www.slideshare.net/mstuparu/osom-operations-in-the-cloud>

Contact information:

marius@ec2.ro / mstuparu@sdl.com