# MO809/MC964
# Tópicos em Computação Distribuída

## Introdução

Islene Calciolari Garcia

Instituto de Computação - Unicamp

Segundo Semestre de 2015

# Sumário

# Um pouco sobre mim...

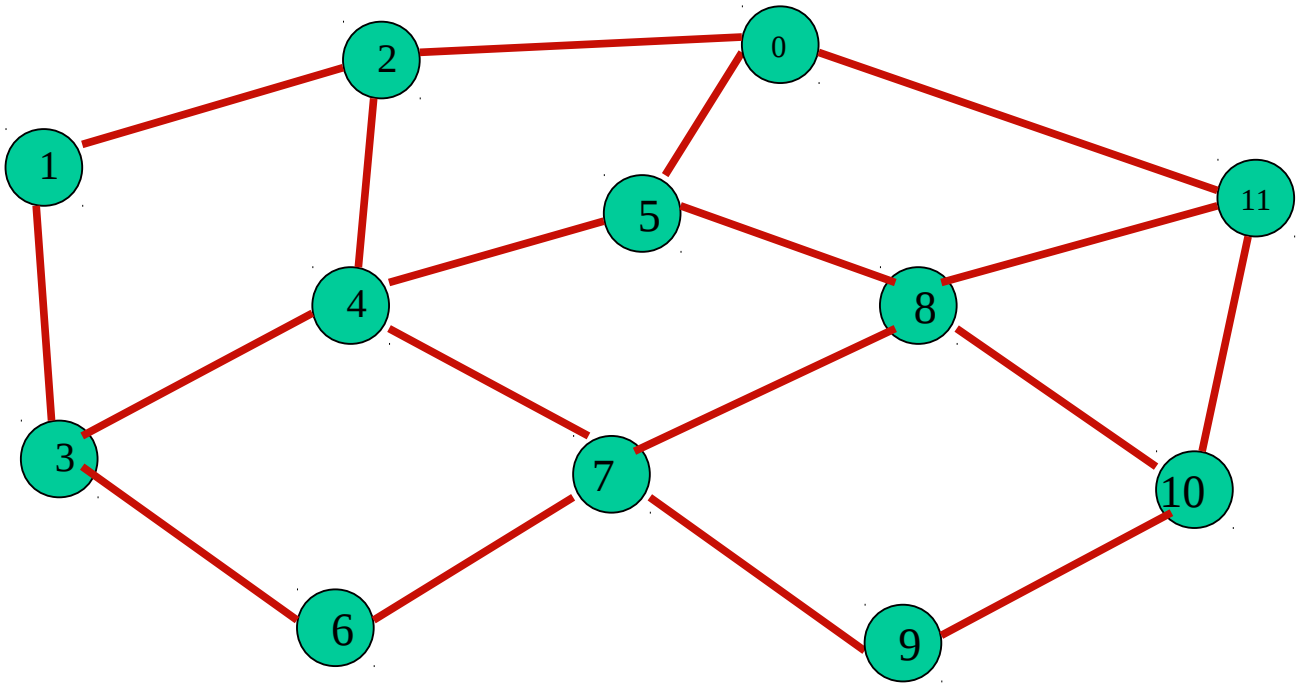
- Formação e filiação
  - Instituto de Computação—Unicamp
- Interesses de pesquisa
  - Sistemas distribuídos
  - Sistemas operacionais

# Objetivos

*Tópicos em . . .* $\Rightarrow$ abordagem livre

*Esta disciplina cobrirá tópicos teóricos e práticos de sistemas distribuídos com ênfase em Big Data.*
*Na parte teórica, serão abordados algoritmos de coordenação e consenso, técnicas de tolerância a falhas e o modelo de programação MapReduce.*
*Na parte prática, analisaremos o projeto e implementação de sistemas distribuídos reais, que tenham código disponível sob licença livre. Em particular, estudaremos o projeto Apache Hadoop e seu modelo de desenvolvimento.*

# What is a distributed system?




A channel may be physical (wired, wireless) or logical

**Abstract view**: It is a **network** of processes.

(The **nodes** are processes, and the **edges** are communication channels.)



Fonte: Prof. Ghosh

# What is a Distributed System?

*A collection of independent, autonomous hosts connected through a communication network.*

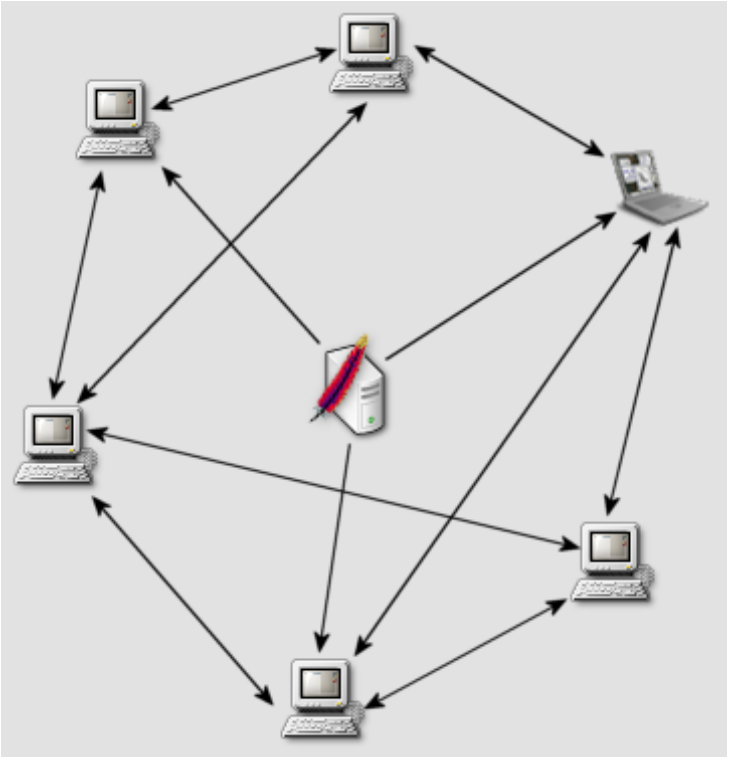
- No shared memory (must use the network)
- No shared clock


© 2014 Paul Krzyzanowski


Fonte: Prof. Paul Krzyzanowski

# Goal of a distributed system

The computers coordinate their activities and to share hardware and software and data, so that users perceive it as a single, integrated computing service with a well-defined goal.



Downloading music in Bittorrent



Fonte: Prof. Ghosh

# Examples

**Large networks** are very commonplace these days. Think of the world wide web. A few examples of distributed systems are:

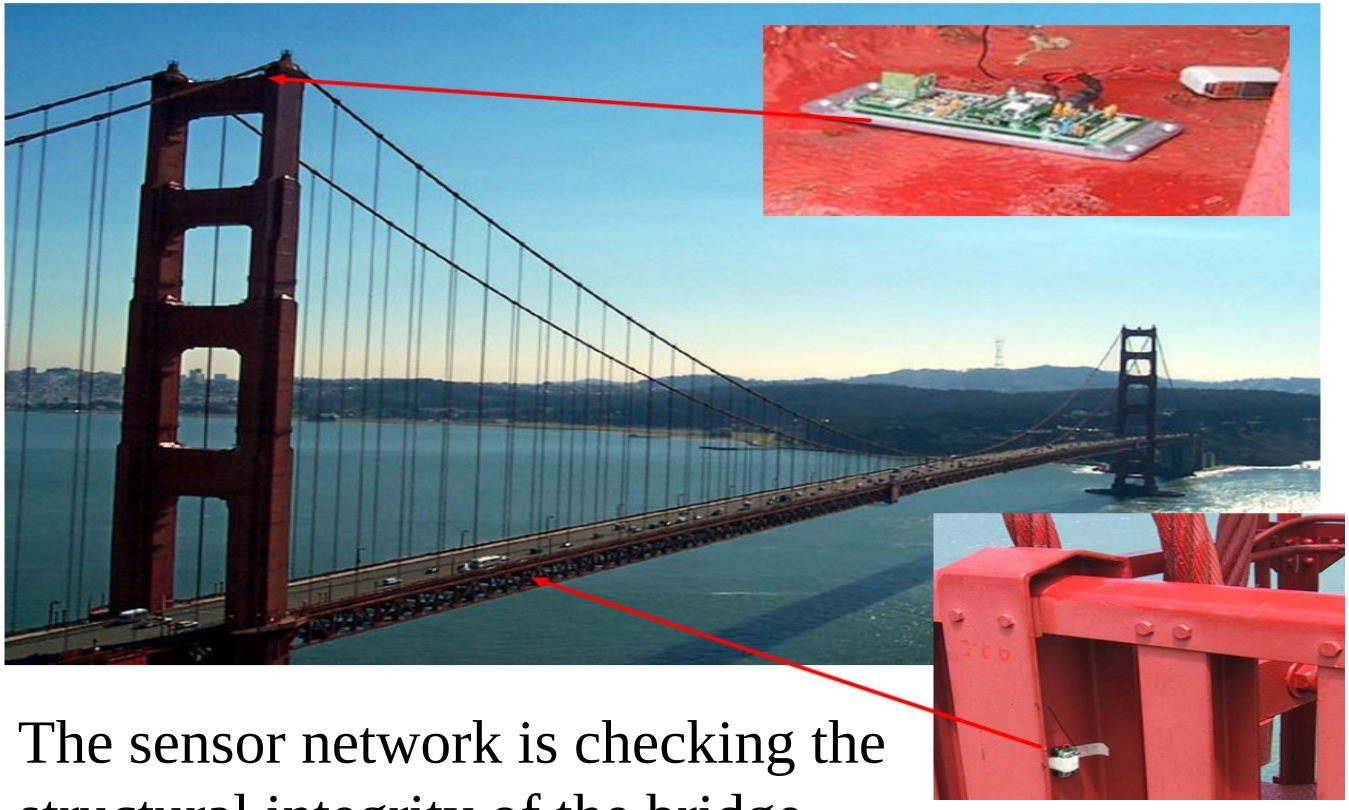
- eBay for internet-based auction
- Sensor networks
- BitTorrent (P2P network) for downloading video / audio
- Skype for making free audio and video communication
- Facebook (the oxygen of many people)
- Process control networks in engineering factories
- Computational grids (**OSG**, **Teragrid, SETI@hom**
- Network of mobile robots collectively doing a job
- Distance education, net-meeting etc.
- Netbanking
- Vehicular networking

What are these?



Fonte: Prof. Ghosh

# Sensor Network

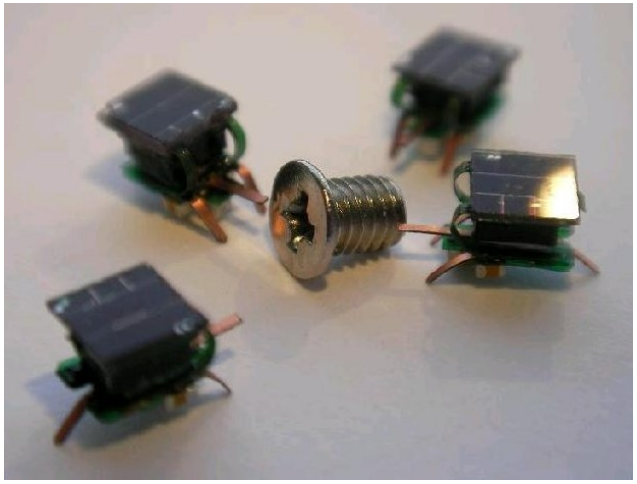


The sensor network is checking the structural integrity of the bridge



Fonte: Prof. Ghosh

# Mobile robots



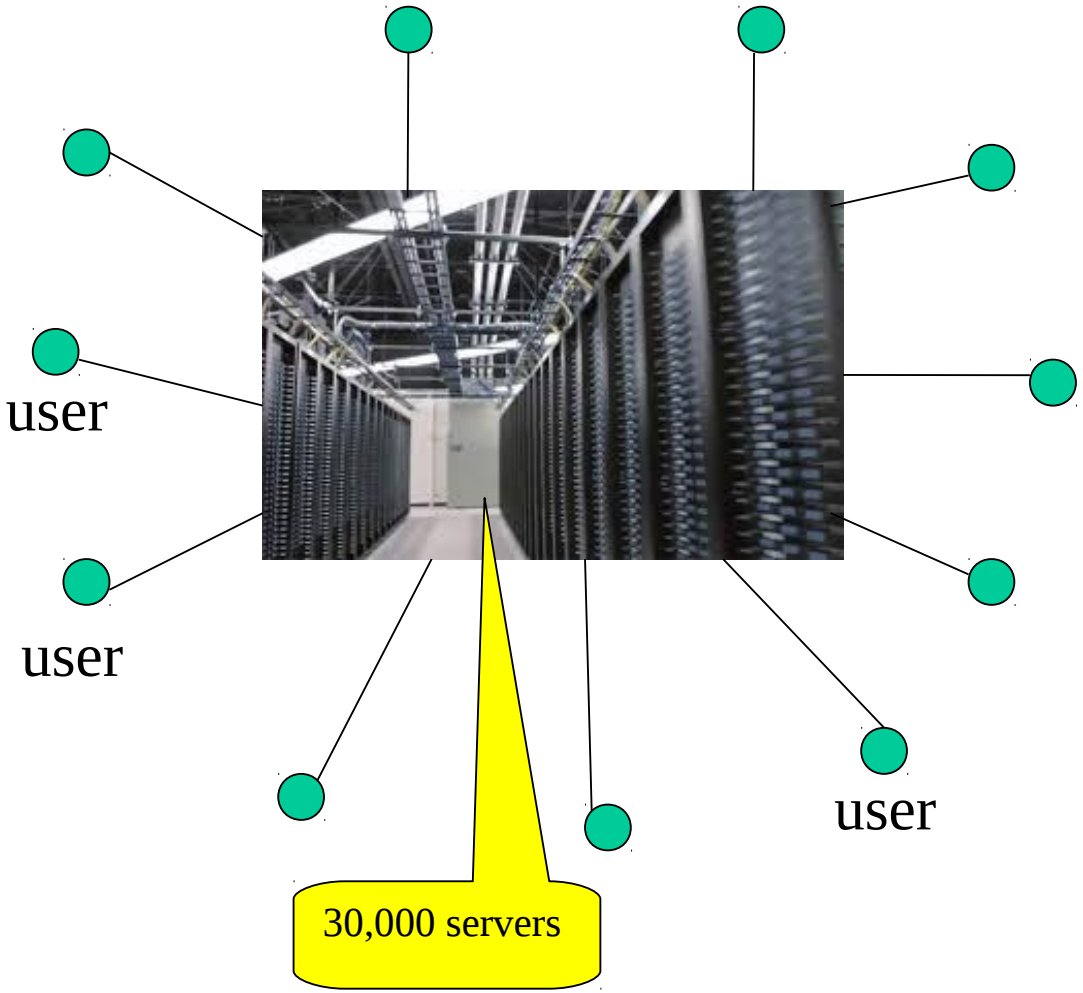
I-Swarm Robot
(See a video of the I-Swarm Robots on YouTube)

The I-Swarm project, consisting of 10 research institutes, is coordinated by Professor Heinz Wörn and Jörg Seyfried of the University of Karsruhe in Germany.



Fonte: Prof. Ghosh

# The Case of Facebook




The new Facebook data center in Prineville, Oregon. The new servers have been redesigned are networked, for energy efficiency, speed-up and for fault-tolerance.

The set up mimics client-server kind of operation, with the servers having a high level of parallelism. However, the network of servers also form a distributed system.



Fonte: Prof. Ghosh

# Objetivos

*Tópicos em . . .* $\Rightarrow$ abordagem livre

*Esta disciplina cobrirá tópicos teóricos e práticos de sistemas distribuídos com ênfase em Big Data.*
*Na parte teórica, serão abordados algoritmos de coordenação e consenso, técnicas de tolerância a falhas e o modelo de programação MapReduce.*
*Na parte prática, analisaremos o projeto e implementação de sistemas distribuídos reais, que tenham código disponível sob licença livre. Em particular, estudaremos o projeto Apache Hadoop e seu modelo de desenvolvimento.*

# Big Data

- *Volume*, *Velocity*, *Variety*, *Veracity*
- Escalabilidade
- Disponibilidade
- Tolerância a falhas
- Segurança
- Como as grandes empresas enfrentam estes desafios?
- Da prática para a teoria: muitas lições a serem aprendidas

# Estudo de caso: Twitter

# Twitter!

## Architecture and Scalability

Aditya B
05IT04

Fonte: Aditiya B

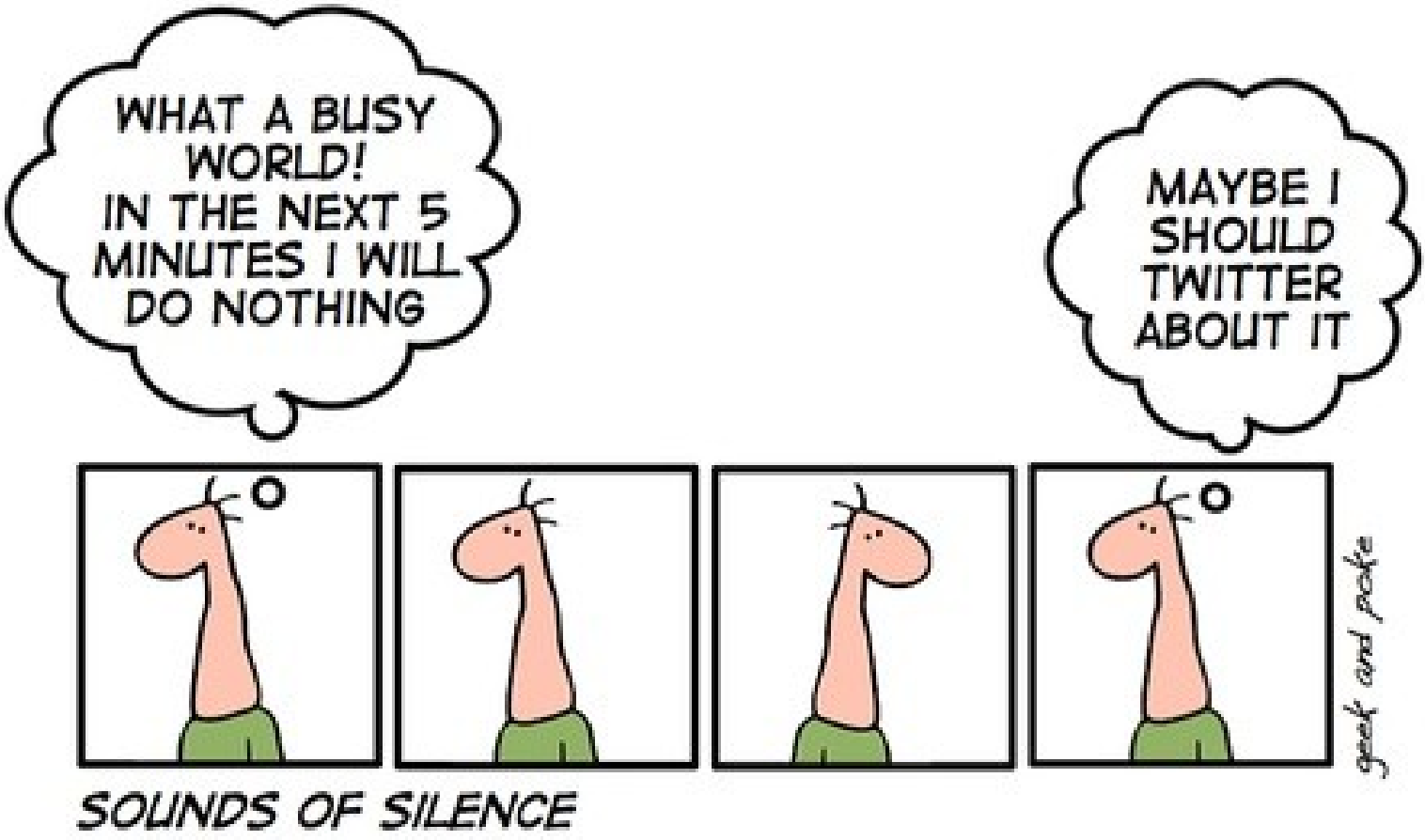
# WHAT IS TWITTER ?

Fonte: Aditiya B

- micro-blogging platform
- text-based posts
- 140 characters in length
- followers receive updates

**Its addictive**

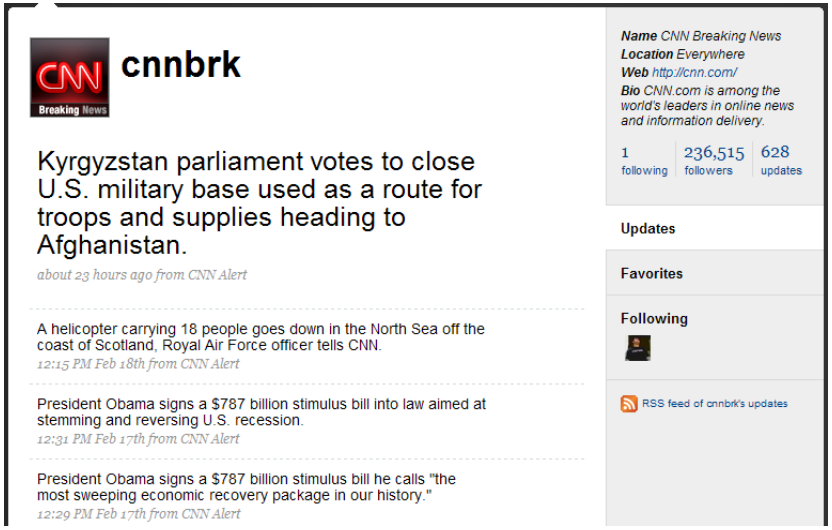
Fonte: Aditiya B




Fonte: Aditiya B


Fonte: Aditiya B

# Who uses twitter

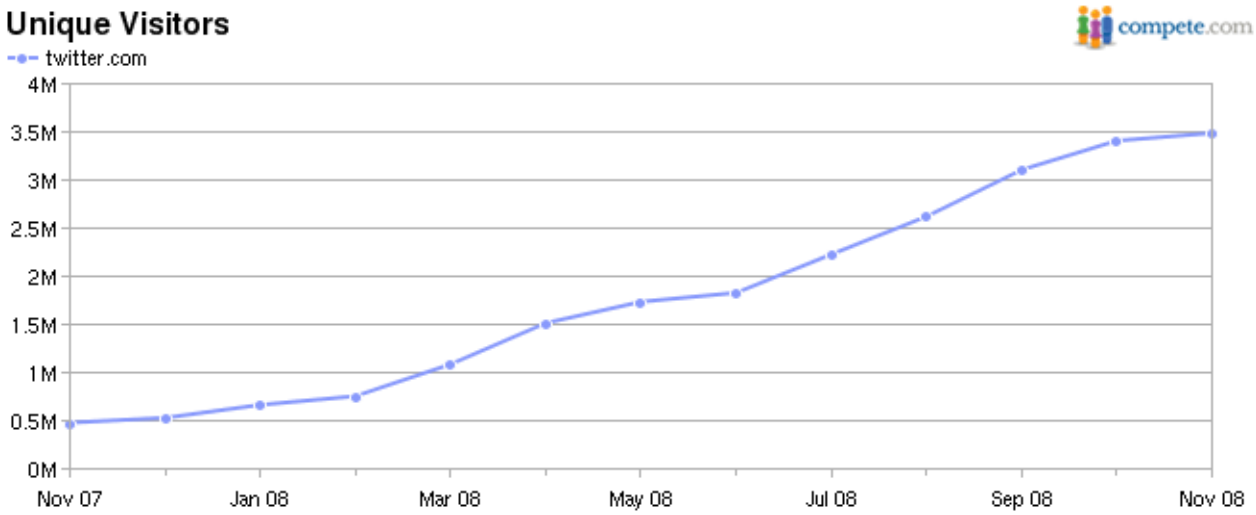






Fonte: Aditiya B

# Web Traffic

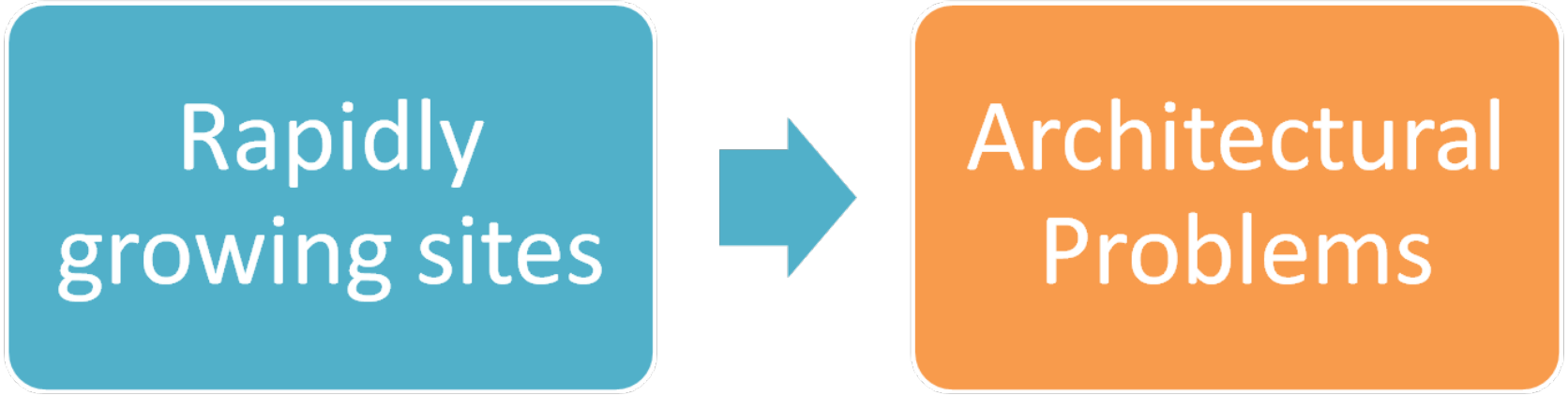



Twitter's web-based traffic

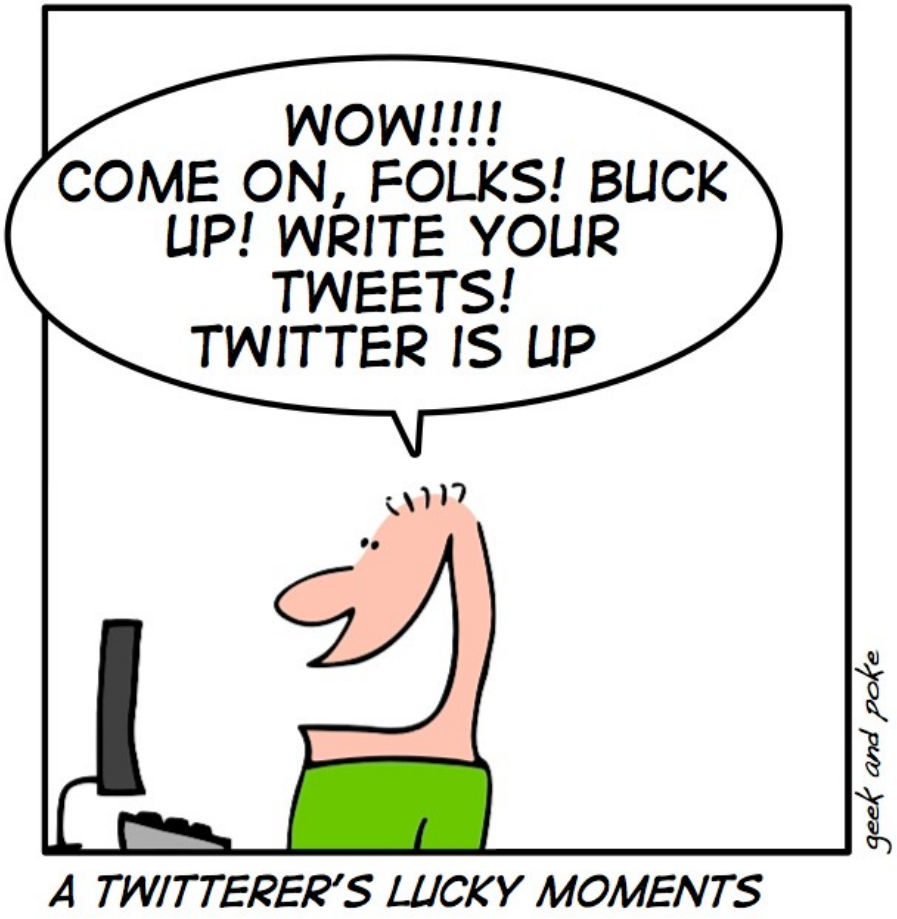
- Plus Twitter's API Traffic which is 10x the Site's

Fonte: Aditiya B

# As it often happens..




Fonte: Aditiya B




Fonte: Aditiya B

# Abuse Prevention

- Bots crawl the site and add everyone as friends.
- 9000 friends in 24 hours. It would take down the site.

**Saraha**

| 9000 | 14 | 2 |
|---|---|---|
| Following | Followers | Updates |

- **Be ruthless. Delete them as users.**


Fonte: Aditiya B

# Scalability -- Doing It Right

- Asynchronous event-driven design
- Partitioning/Shards
- Parallel execution
- Replication (read-mostly)

Fonte: Aditiya B

# Are we all doomed to go through this painful process when we are successful?

- Time-To-Market

Vs

- Architecture

***Good, Fast, Cheap - pick two***

:P

Fonte: Aditiya B

# LESSONS LEARNED

Fonte: Aditiya B

1. Talk to the community.
2. Treat your scaling plan like a business plan
3. Build it yourself
4. Build in user limits
5. Don't make the database the central bottleneck of doom
6. Make your application easily partitionable from the start

Fonte: Aditiya B

7. Optimize the database
8. Cache the hell out of everything
9. Most performance comes not from the language, but from application design
10. Turn your website into an open service by creating an API.

    Their API is the single most powerful reason for Twitter's success.

Fonte: Aditiya B

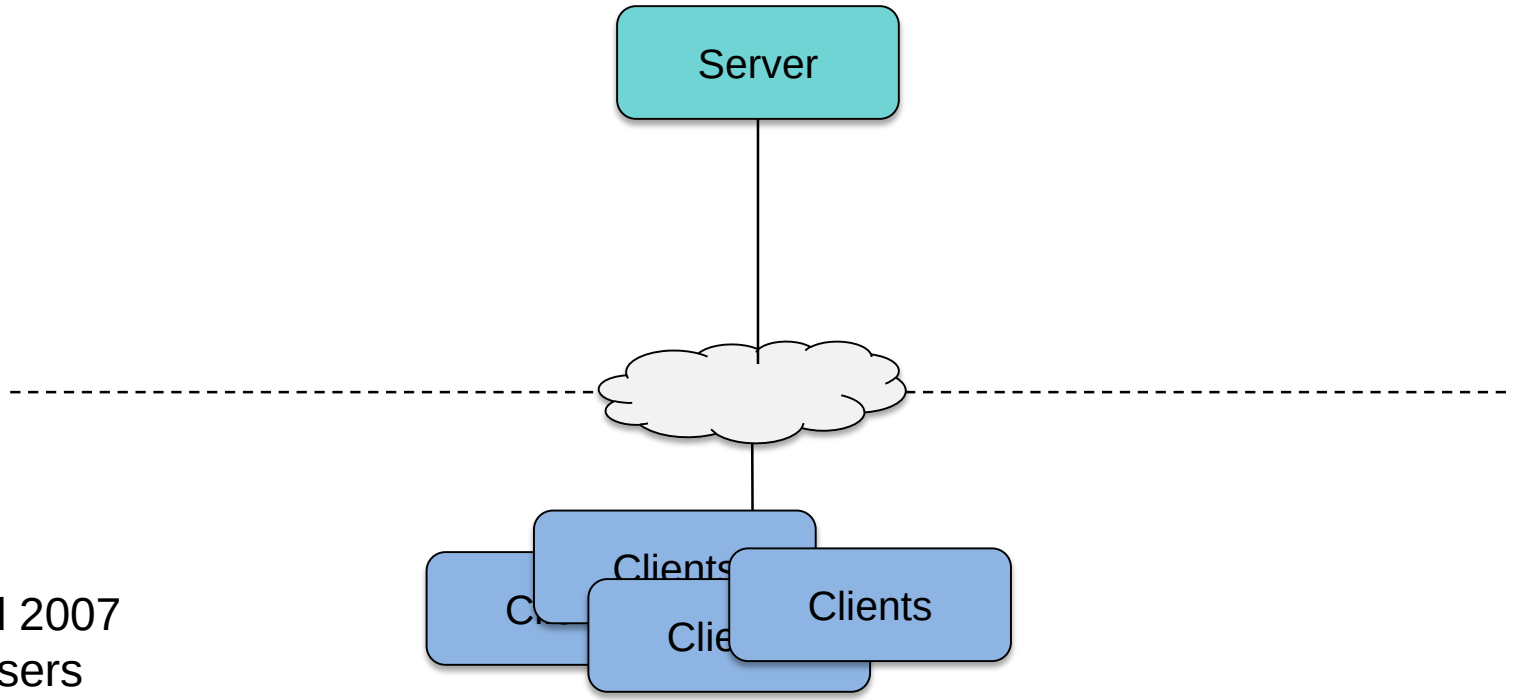
# References


- http://twitter.com/
- http://highscalability.com/scaling-twitter-making-twitter-10000-percent-faster
- http://www.slideshare.net/Blaine/scaling-twitter
- http://dev.twitter.com/2008/05/twittering-about-architecture.html
- http://www.danga.com/memcached/
- http://geekandpoke.com/


Fonte: Aditiya B

# Dropbox: architecture evolution: version 1

– One server: web server, app server, mySQL database, sync server




mid 2007
0 users

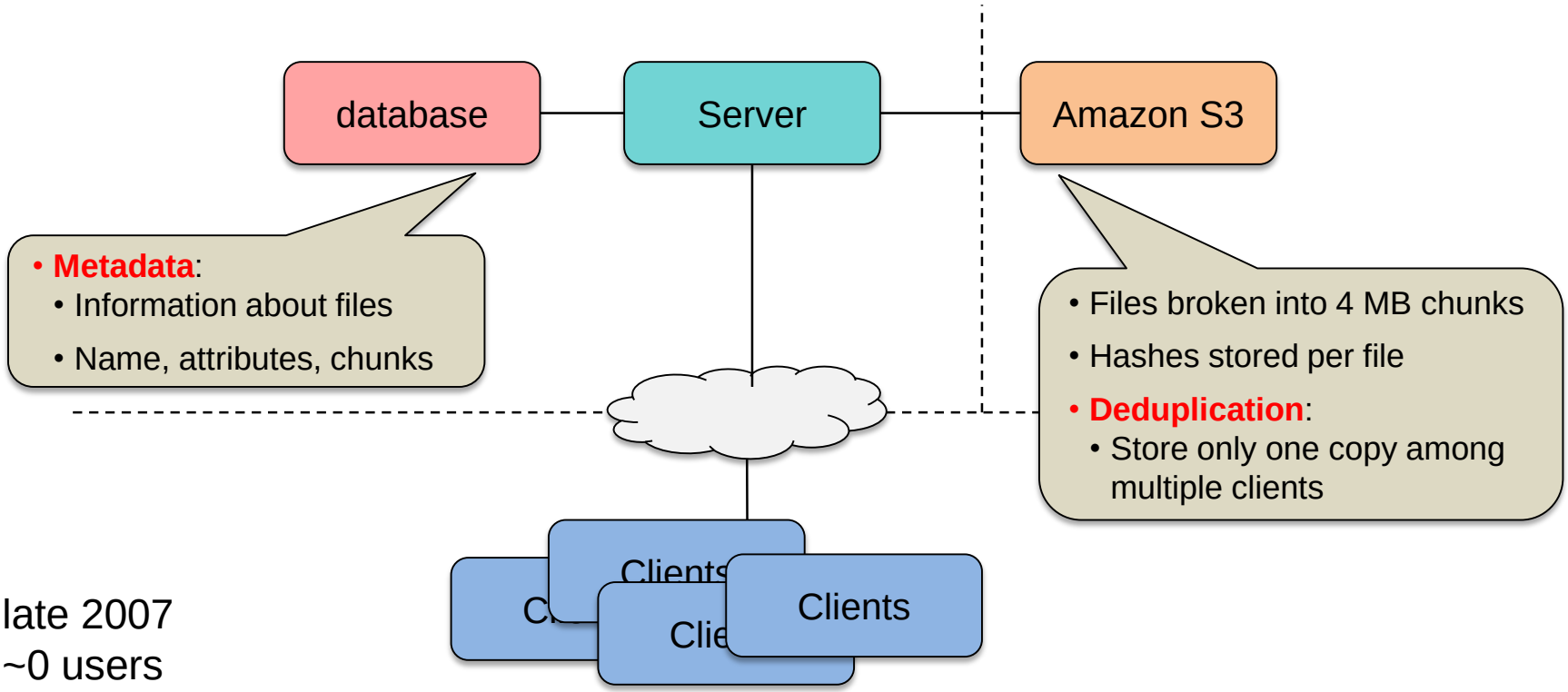
See http://youtu.be/PE4gwstWhmc


© 2014 Paul Krzyzanowski


Fonte: Prof. Paul Krzyzanowski

# Dropbox: architecture evolution: version 2

- Server ran out of disk space:
  moved data to Amazon S3 service (key-value store)
- Servers became overloaded: moved mySQL DB to another machine
- Clients periodically polled server for changes




late 2007
~0 users

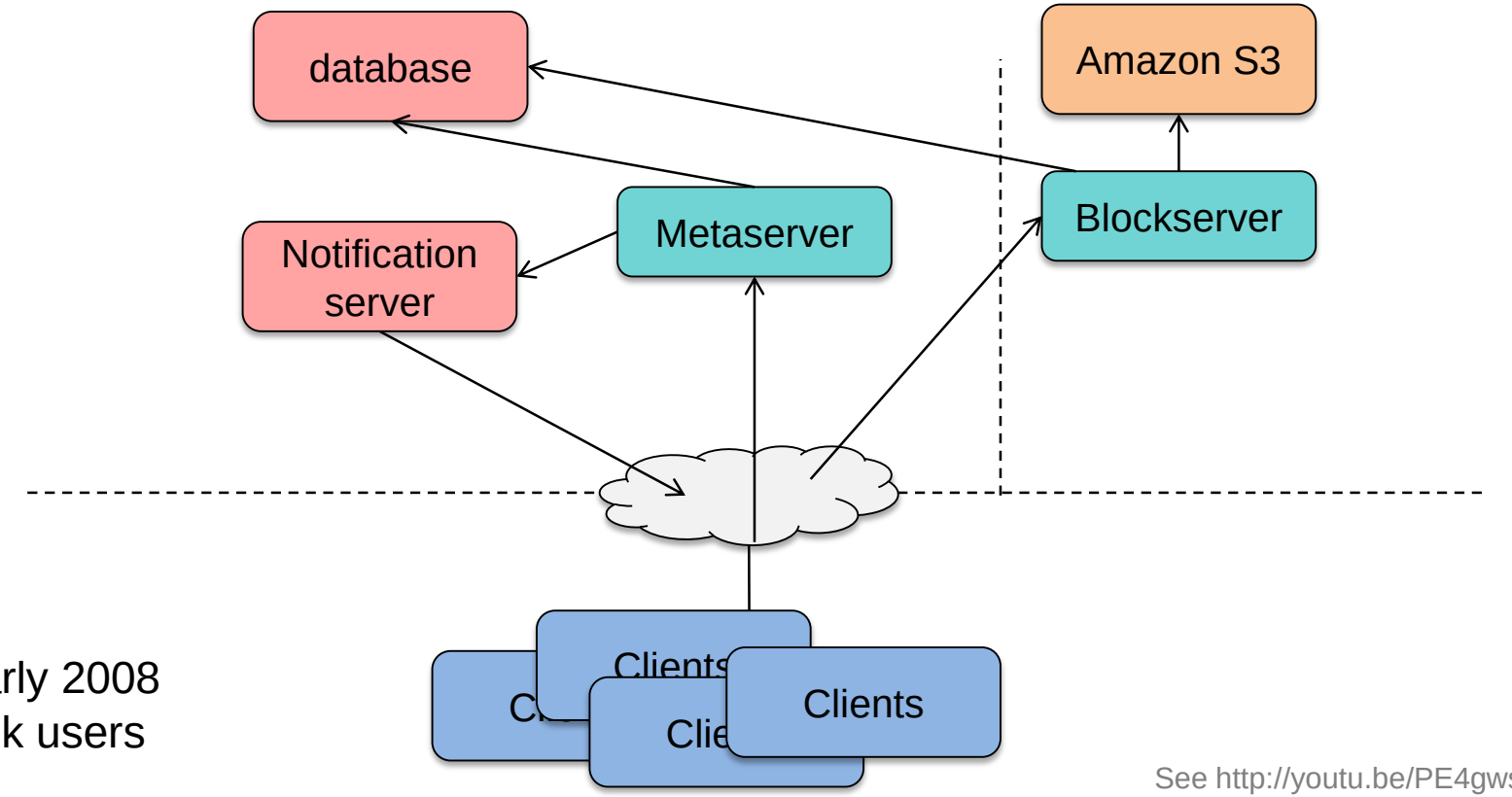
See http://youtu.be/PE4gwstWhmc


© 2014 Paul Krzyzanowski


Fonte: Prof. Paul Krzyzanowski

# Dropbox: architecture evolution: version 3

- Move from polling to notifications: add **notification server**
- Split web server into two:
  - Amazon-hosted server hosts file content and accepts uploads (stored as blocks)
  - Locally-hosted server manages metadata




early 2008
50k users

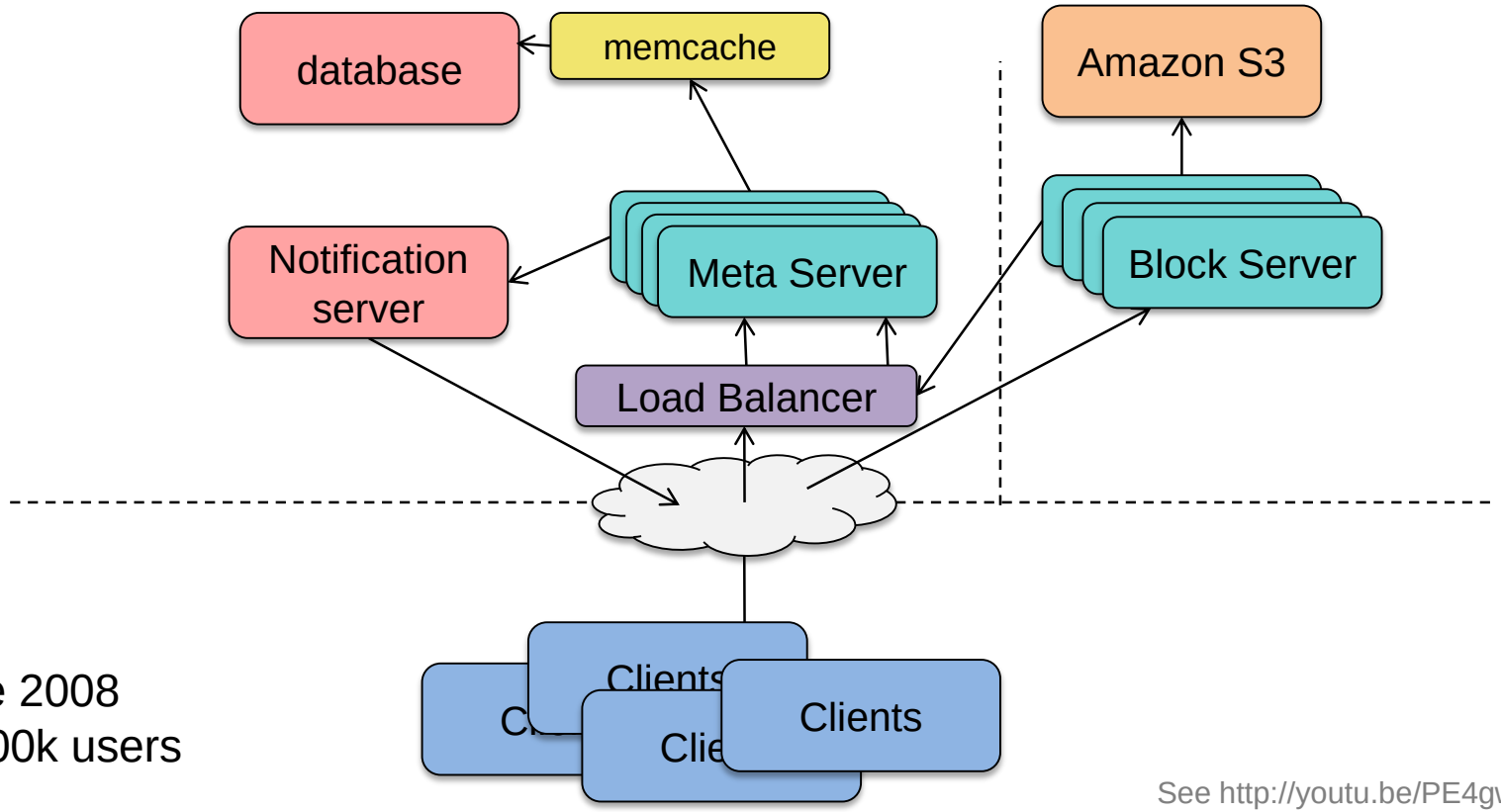
See http://youtu.be/PE4gwstWhmc


© 2014 Paul Krzyzanowski


Fonte: Prof. Paul Krzyzanowski

# Dropbox: architecture evolution: version 4

- Add more metaservers and blockservers
- Blockservers do not access DB directly; they send RPCs to metaservers
- Add a memory cache (memcache) in front of the database to avoid scaling




late 2008
~100k users

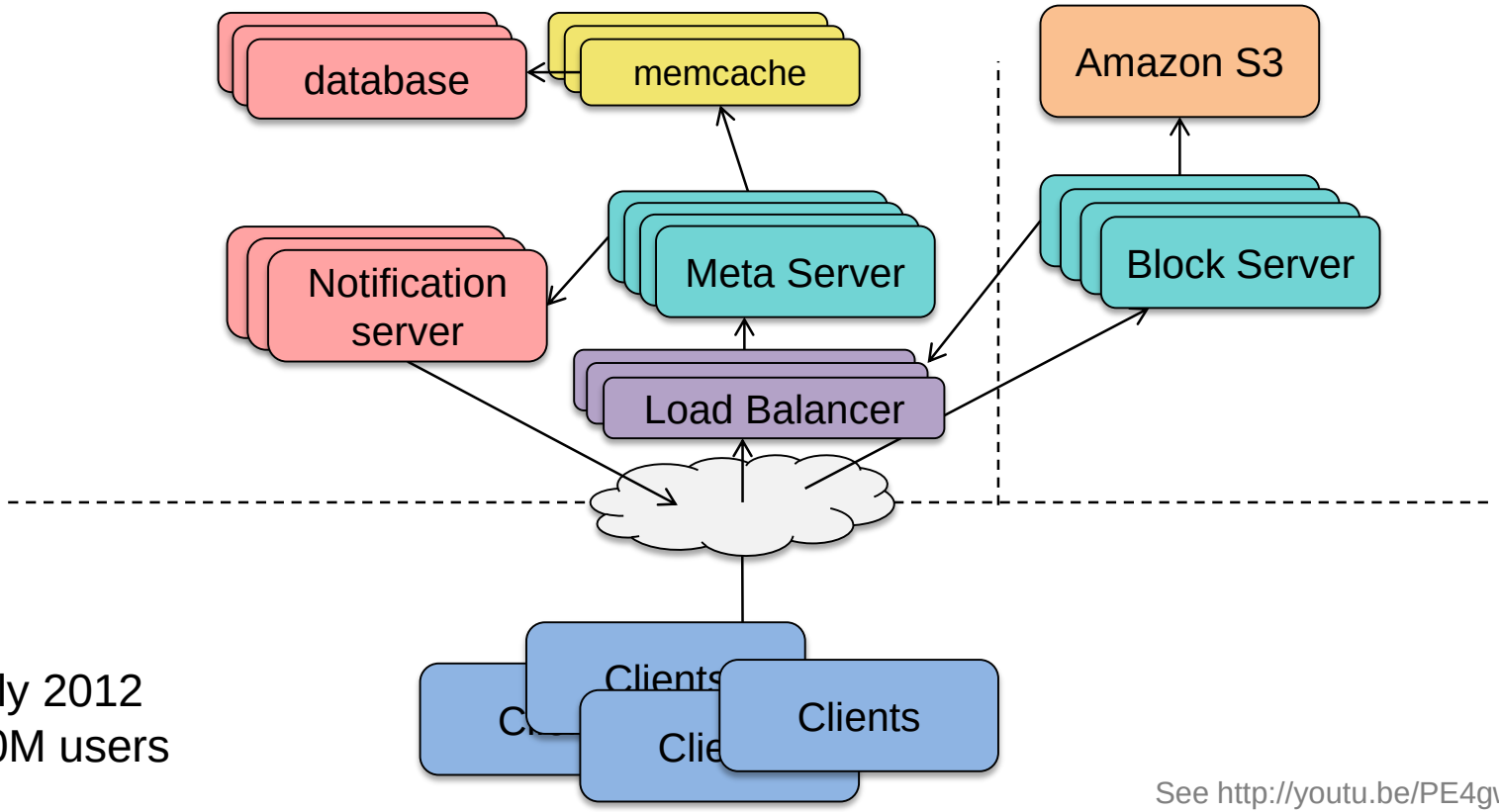
See http://youtu.be/PE4gwstWhmc


© 2014 Paul Krzyzanowski


Fonte: Prof. Paul Krzyzanowski

# Dropbox: architecture evolution: version 5

- 10s of millions of clients – Clients have to connect before getting notifications
- Add 2-level hierarchy to notification servers: ~1 million connections/server




early 2012
>50M users

See http://youtu.be/PE4gwstWhmc

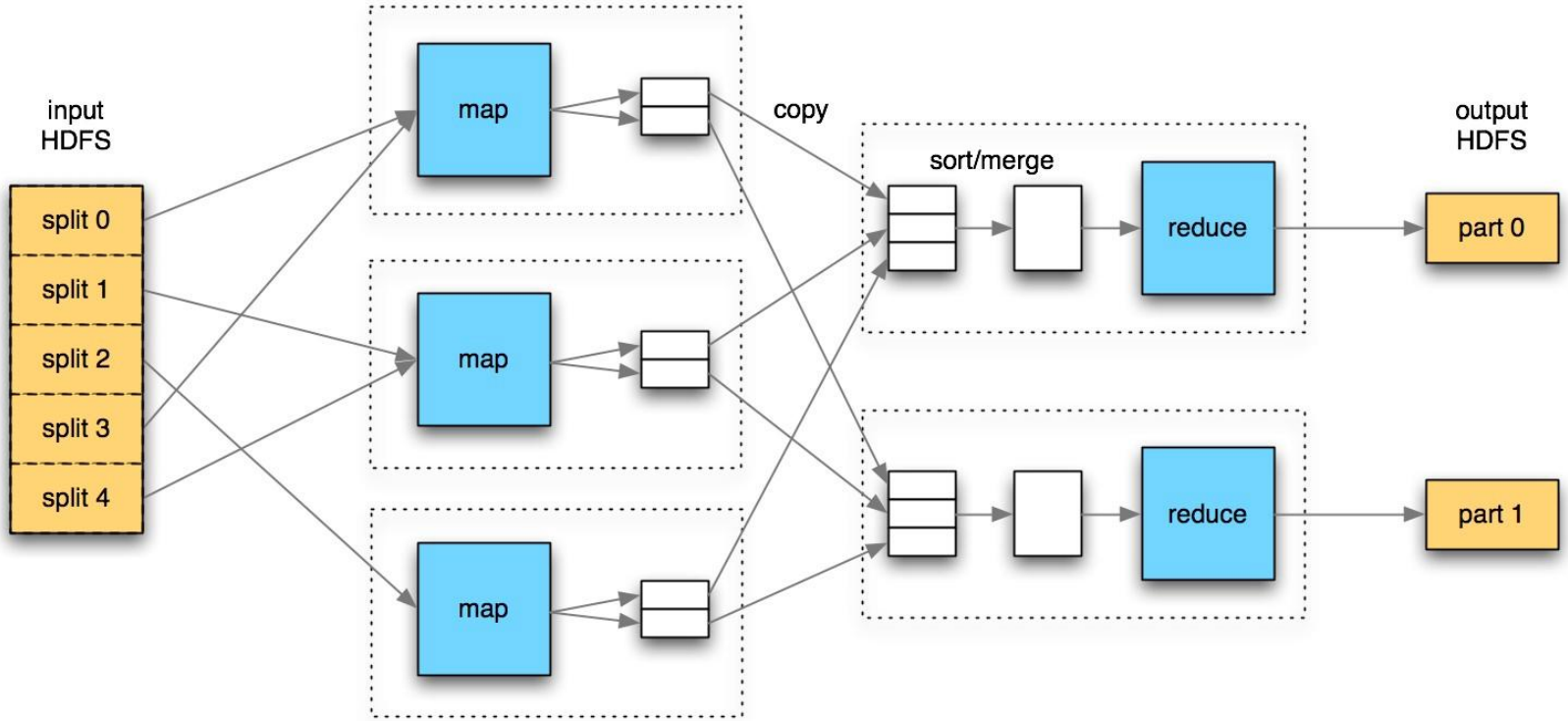

© 2014 Paul Krzyzanowski


Fonte: Prof. Paul Krzyzanowski

# História do projeto Hadoop

- 2002-2004: Doug Cutting e Mike Cafarella trabalham no projeto Nutch.
  - Nutch deveria indexar a web e permitir buscas
  - Alternativa livre ao Google
- 2003-2004: Google publica artigo sobre o Google File System e MapReduce
- 2004: Doug Cutting adiciona o DFS e MapReduce ao projeto Nutch
- 2006: Doug Cutting começa a trabalhar no Yahoo!
- 2008: Hadoop se torna um projeto Apache
- 2013: Yarn (Hadoop 2)

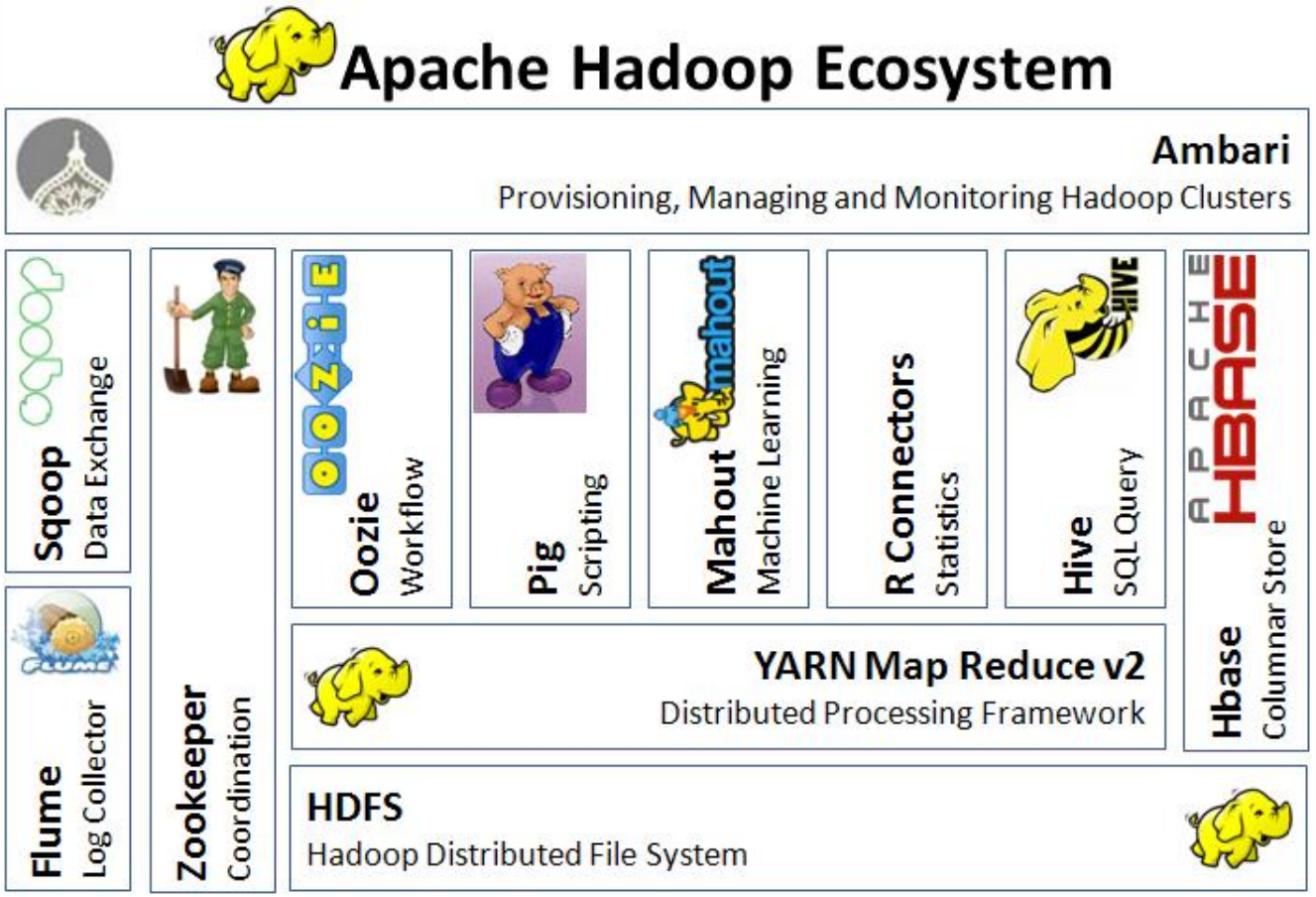
# MapReduce




Tom White

# Apache Hadoop Ecosystem

## Um ecossistema em evolução



# Explorando o Apache Hadoop


- Dimensão de usuário:
  - MapReduce
  - Outros sistemas: HBase, ZooKeeper, Hive, Pig
- Dimensão de desenvolvedor:
  - Como contribuir
  - Jira
- Comparação com outros sistemas/abordagens

# Objetivos

*Tópicos em . . .* $\Rightarrow$ abordagem livre

*Esta disciplina cobrirá tópicos teóricos e práticos de sistemas distribuídos com ênfase em Big Data.*
*Na parte teórica, serão abordados algoritmos de coordenação e consenso, técnicas de tolerância a falhas e o modelo de programação MapReduce.*
*Na parte prática, analisaremos o projeto e implementação de sistemas distribuídos reais, que tenham código disponível sob licença livre. Em particular, estudaremos o projeto Apache Hadoop e seu modelo de desenvolvimento.*

# Sistema de Rastreamento de Bugs e Melhorias

- Comunidades de software livre expõem seu desenvolvimento via ferramentas específicas



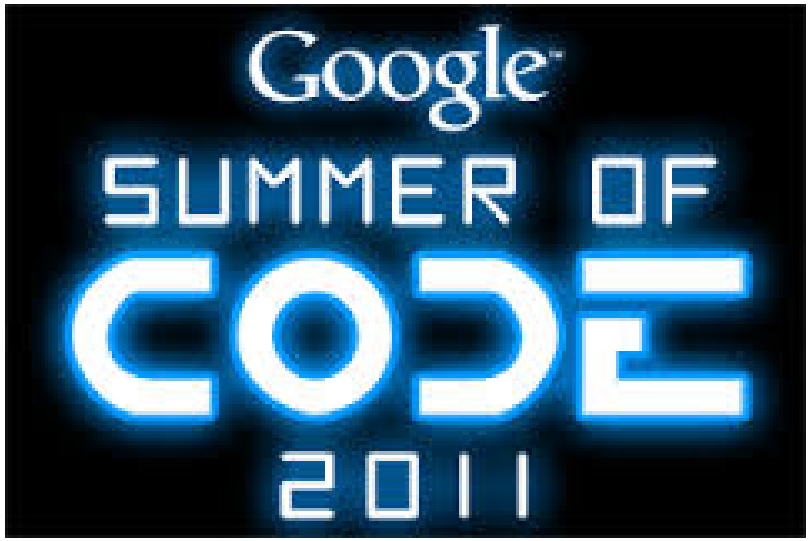
  - Bugs
    - só que não
    - simples ou complexos
    - prioritários ou não prioritários
  - Melhorias
    - Ideias para o avanço dos projetos
    - Entradas guarda-chuva
- Oportunidade para aprender e participar!

# Como escolher uma issue para trabalhar?

- A quanto tempo a issue está aberta?
- Quem reportou o problema ou melhoria?
- A comunidade deu algum retorno?
- Houve discussão?
- Existe possibilidade de contribuição?
- Se a issue estiver fechada? Valeria pelo estudo do processo, mas as chances de contribuição são mínimas...

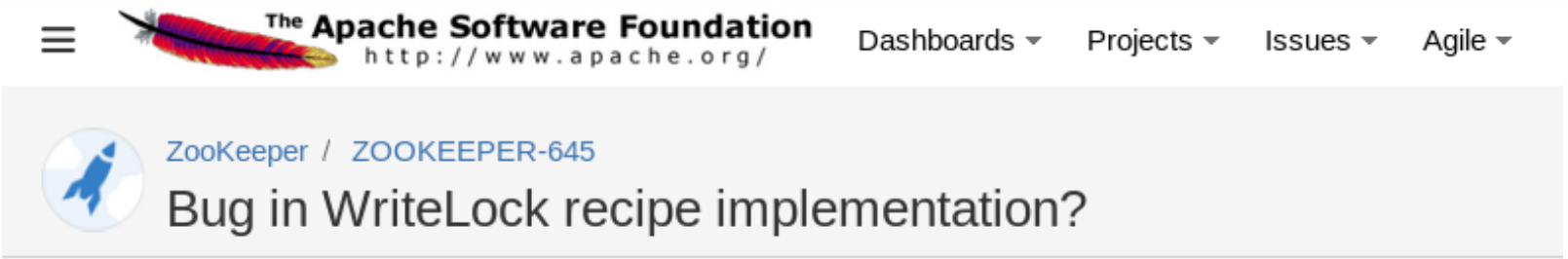
# Em busca da issue perfeita (2010)

- ZooKeeper: *Because coordinating distributed systems is a Zoo*
- Possível participação no GSoC 2011
  - *Flip bits not burgers*
  - Estudantes recebem cerca de US$ 5.000,00!!!
  - Google seleciona os projetos que selecionam os estudantes
  - Envolvimento inicial aumenta as chances de sucesso

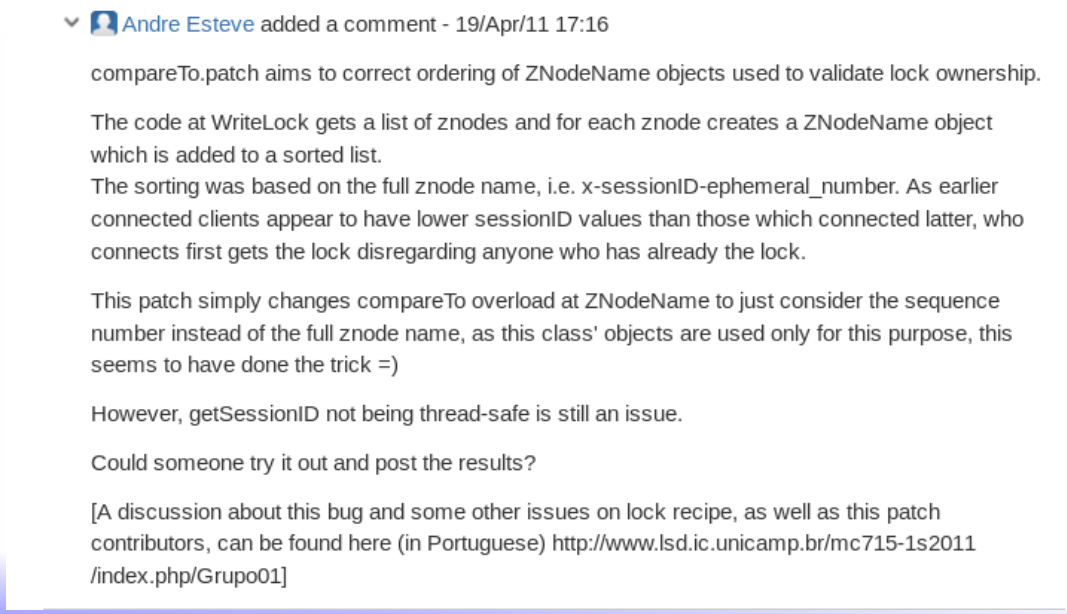



# A primeira contribuição! (2011)

- Turma pequena (quase todos tinha cursado no verão)
- Ainda as receitas do ZooKeeper




- Envio de patch



# E o retorno da primeira contribuição?

- Um mês depois...

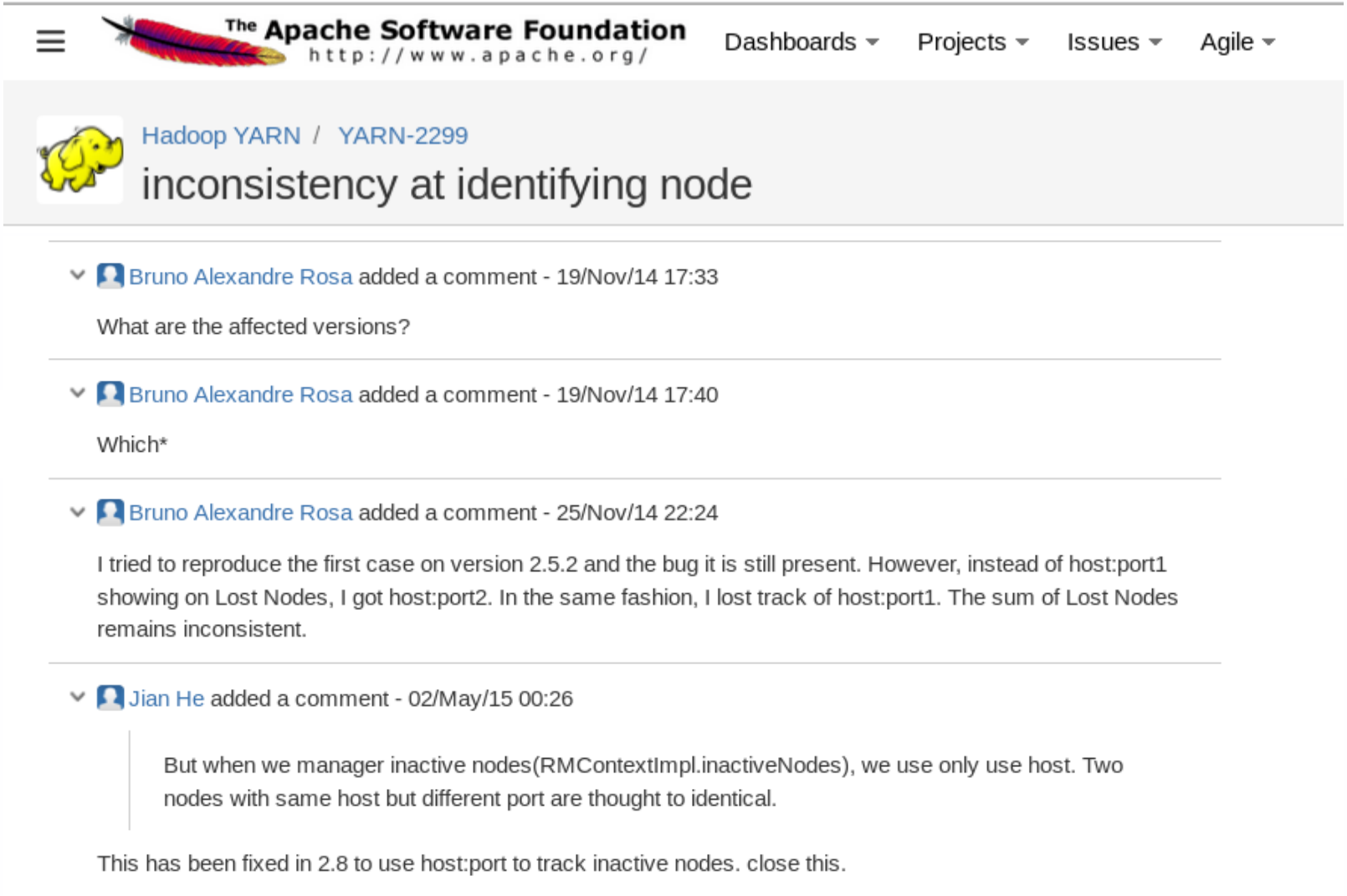
> Patrick Hunt added a comment - 23/May/11 20:04
>
> Hi, has anyone tried this?
>
> Matt Abrams added a comment - 30/May/11 23:09
>
> I've tried it. I did see the starvation behavior with the original compareTo method from ZNodeName. When I applied the compareTo patch the starvation issue went away.

- Mais de um ano depois
  - Patch seria combinado com outro e incluído
  - Grupo comemorou :-)
- O bug continua em aberto...

# Kazoo (2013)

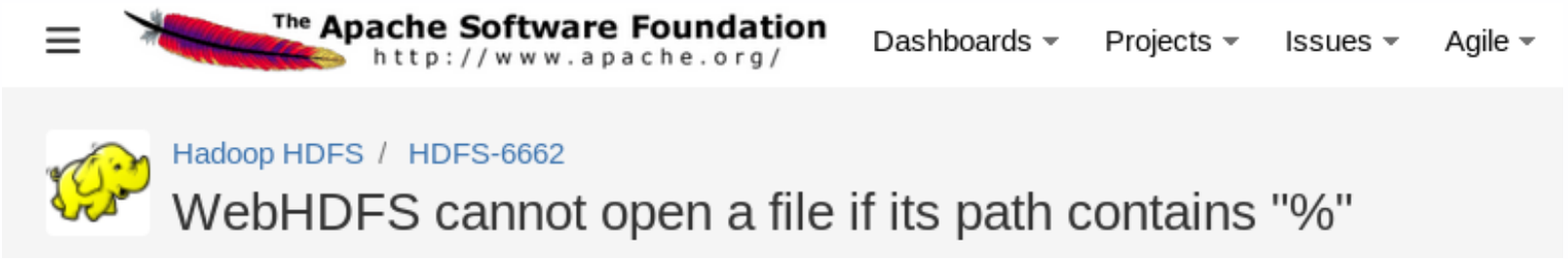
- Comunidade ZooKeeper demora para dar retorno?
- Netflix apoiou o desenvolvimento do Curator!
- Kazoo é uma interface em Python para facilitar o uso do ZooKeeper
- Comunidade mais receptiva!
- Aluno de mestrado contribuiu e seu patch foi aceito!


# E se formos completamente ignorados?

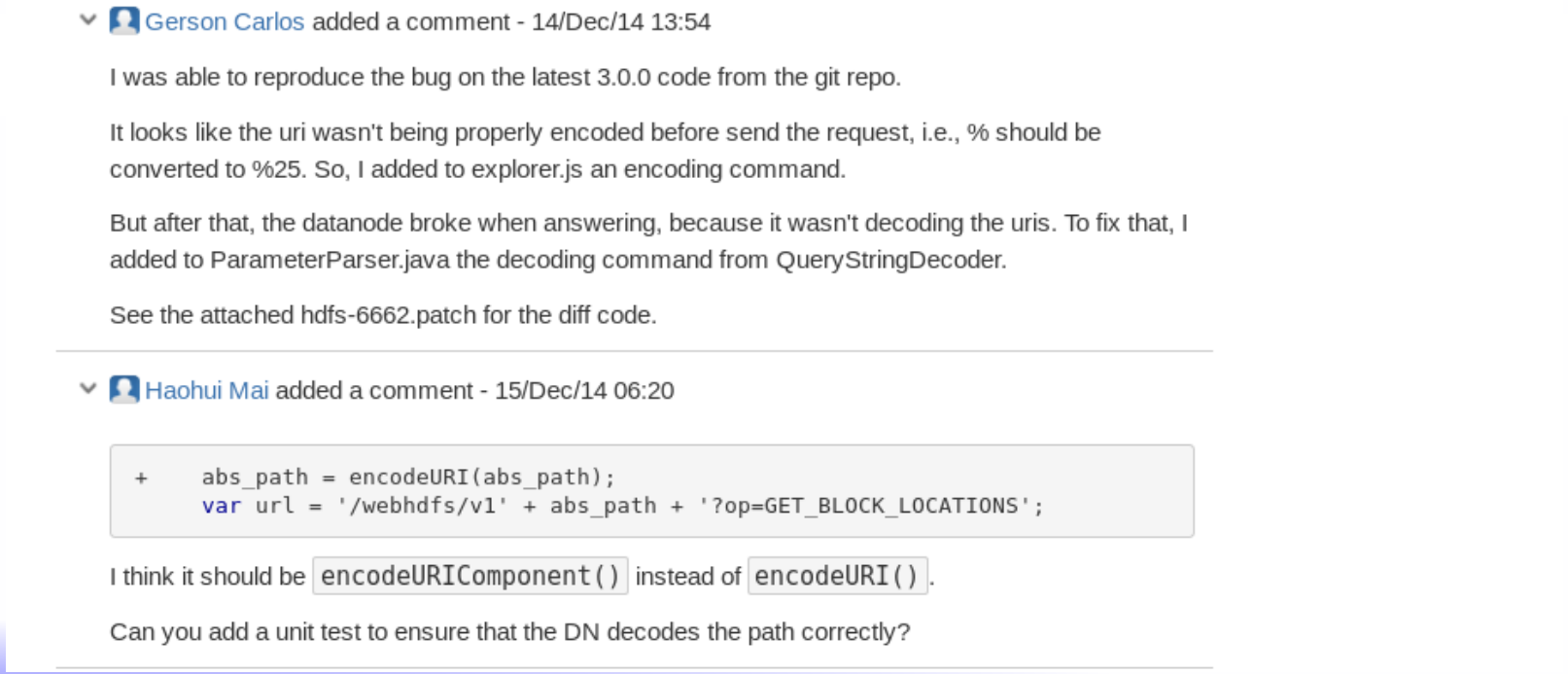


# A melhor experiência

- Bug simples




- Envio de patch e retorno rápido da comunidade



# A melhor experiência

- Uma surpresa

**Issue Links**

breaks

~~HDFS-7816~~ Unable to open webhdfs paths with "+" CLOSED

- Mais orientações da comunidade e
- final feliz! :-)

Hadoop HDFS / HDFS-6662

WebHDFS cannot open a file if its path contains "%"

Agile Board

**Details**

| | | | |
|---|---|---|---|
| Type: | Bug | Status: | CLOSED |
| Priority: | Critical | Resolution: | Fixed |
| Affects Version/s: | 2.4.1 | Fix Version/s: | 2.7.0 |
| Component/s: | namenode | | |
| Labels: | None | | |

**People**

Assignee: Gerson Carlos

Reporter: Brahma Reddy Battula

# Abordagem

- Estudo de tópicos em sistemas distribuídos (big data)
  - Sistemas de arquivos distribuídos
  - Modelo de programação MapReduce
  - Algoritmos para Coordenação
  - ...
- Análise de *issues* do projeto Apache Hadoop
- Experimentos práticos
- Artigos científicos

# Critério de Avaliação

- Projetos relacionados aos tópicos (6.0)
  - experimentos práticos ou
  - análise de *issues* ou
  - leitura/apresentação de artigos científicos
- Seminário (3.0)
- Participação em seminários/apresentações (1.0)
- Tabela de conversão de notas:
  $10 \ldots 8.5 = A$, $8.4 \ldots 7.0 = B$, $6.9 \ldots 5.0 = C$, $5.0 \ldots 0 = D$

# Principais Referências


- Distributed Systems: Principles and Paradigms, Andrew S. Tanenbaum and Maarten Van Steen, Second Edition, Pearson, 2007.
- Distributed Systems: Concepts and Design, George Coulouris, Jean Dollimore, Tim Kindberg and Gordon Blair, Fifth Edition, Addison Wesley, 2011.
- Hadoop: The Definitive Guide, Tom White, Fourth Edition, O'Reilly, 2015.
- The Hadoop Project