

An overview

Luigi De Russis

# Social Network Technologies



POLITECNICO  
DI TORINO



# Prerequisite

2

Did you know...

- what is a [data center](#)?
- the difference between a logical server and a physical server?
- the difference between cache and database access?

3

Let's think!

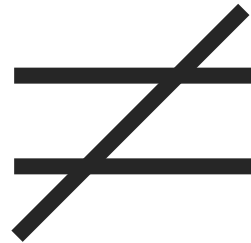
# A “typical” web site?

4

Traditional web  
site

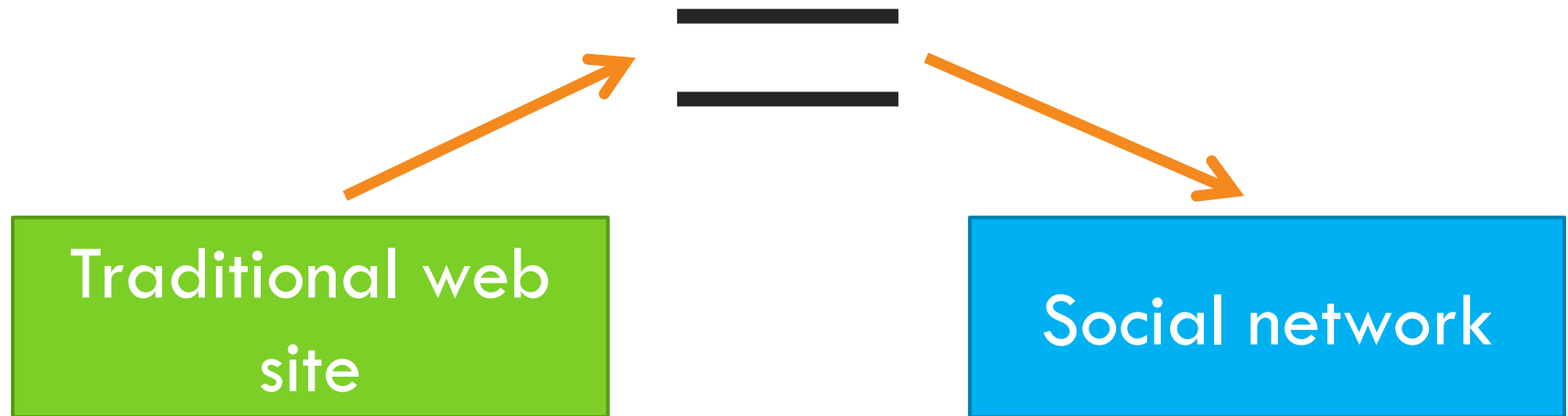


Social network



# A “typical” web site?

5



In the beginning, probably...

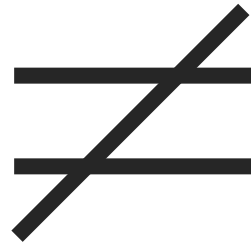
# A “typical” web site?

6

... but soon or later...

Traditional web  
site

Social network



ONALS

COMMERCIAL COMPANIES

LOCAL AUTHORITIES

Small businesses  
Industries

Age Concern  
Scope Mencap  
Mind, etc

National

net 300 /  
s/ other  
health profs  
signals

SOME OF  
GIVING  
LIVING  
COMPLAINERS

Small  
businesses  
Industries

Why?

GPs

HOBBIES

Family  
happy  
happy

COMMERCIAL MARKET

Team  
Manager

SCHOOLS

the 2000  
past years  
policy work

General

developing  
work  
opportunities  
etc

only  
SINGLE  
PEOPLE

Environment  
60

multinational

Small business

# Web site: tools and technologies

8

What tools and technologies are we using for building a website?



# Web site: tools and technologies

9

For example...

## Java/JSP

- JVM + JDK + J2EE
- Tomcat (or similar)
- MySQL (or similar)

## PHP

- PHP
- Apache
- MySQL (or similar)

# Web site: tools and technologies

10

## Java/JSP

- JVM + JDK + J2EE
- MySQL (or similar)
- Tomcat (or similar)

## PHP

- PHP
- MySQL (or similar)
- Apache server

Typically, a vertical stack  
(with one programming language)

# Now let's try with these sites...

11

The Facebook logo, consisting of the word "facebook" in white lowercase letters on a dark blue rectangular background.The Twitter logo, featuring the word "twitter" in blue lowercase letters followed by a blue bird icon.The Pinterest logo, featuring the word "Pinterest" in a red, cursive script font.

# Now let's try with these sites...

12



facebook



PHP  
MySQL  
Apache



twitter



Rails  
MySQL  
Unicorn



Pinterest



Django  
MySQL  
Tornado



Django  
PostgreSQL  
Gunicorn



Instagram  
Fast beautiful photo sharing

# Now let's try with these sites...

13

facebook

PHP  
MySQL

twitter 

Rails  
MySQL

Are you really, really sure?

Django  
MySQL  
Tornado

Django  
PostgreSQL  
Gunicorn

Pinterest

Instagram  
Fast beautiful photo sharing

# Not so sure?!

14

Let's have a look at some pages and projects...

<https://github.com/twitter>

<https://github.com/facebook>

<https://code.facebook.com/projects/>

<https://engineering.twitter.com/opensource/projects>

What did you notice?

# What did you notice?

15

A lot of different components

- Twitter
  - ▣ tracing system, package manager, various servers, NoSQL database, caching system, etc.
  
- Facebook
  - ▣ code-related tools, code transformer, various servers, distributed file system, caching system, NoSQL database, etc.

# What did you notice?

16

## A lot of different languages

### □ Twitter

- ▣ Java, Scala, Ruby, C++, C, Objective-C, Shell scripting, Python, JavaScript

### □ Facebook

- ▣ PHP, OCaml, C++, JavaScript, Python, Java, Objective-C, Processing, C, Ruby, Shell scripting, Haskell, Emacs Lisp, ActionScript



# Now, have a look again at this...

17

Web site: tools and technologies

9

Java/JSP	PHP
<ul style="list-style-type: none"><li>□ JVM + JDK + J2EE</li><li>□ MySQL (or similar)</li><li>□ Tomcat (or similar)</li></ul>	<ul style="list-style-type: none"><li>□ PHP</li><li>□ MySQL (or similar)</li><li>□ Apache server</li></ul>

Typically, a vertical stack  
(with one programming language)

Social Networks Technologies 14/03/2013

Impressive?!



This is **ONLY** the tip of the iceberg...

# Social Network Characteristics

19

- Wildly popular over last few years
  - ▣ Facebook has more than 1 billion (monthly active) users
  - ▣ Twitter has more than 600M users
- Distributed across the planet
- Changed how content is created and consumed
- Explosion of smartphones
  - ▣ photos and video are now easy to shoot and share
  - ▣ e.g., Facebook has more than 350M photos uploaded each day

20

# Facebook as an example...

# On December 2014...

21

More than **1 billion**  
monthly active users

More than **3.9 trillion**  
feed actions processed  
per day

More than **200 billion**  
monthly page views

Over **500 TB** new  
data ingested per day

**4.75 billion** content  
items shared per day

**890 million** active  
users per day

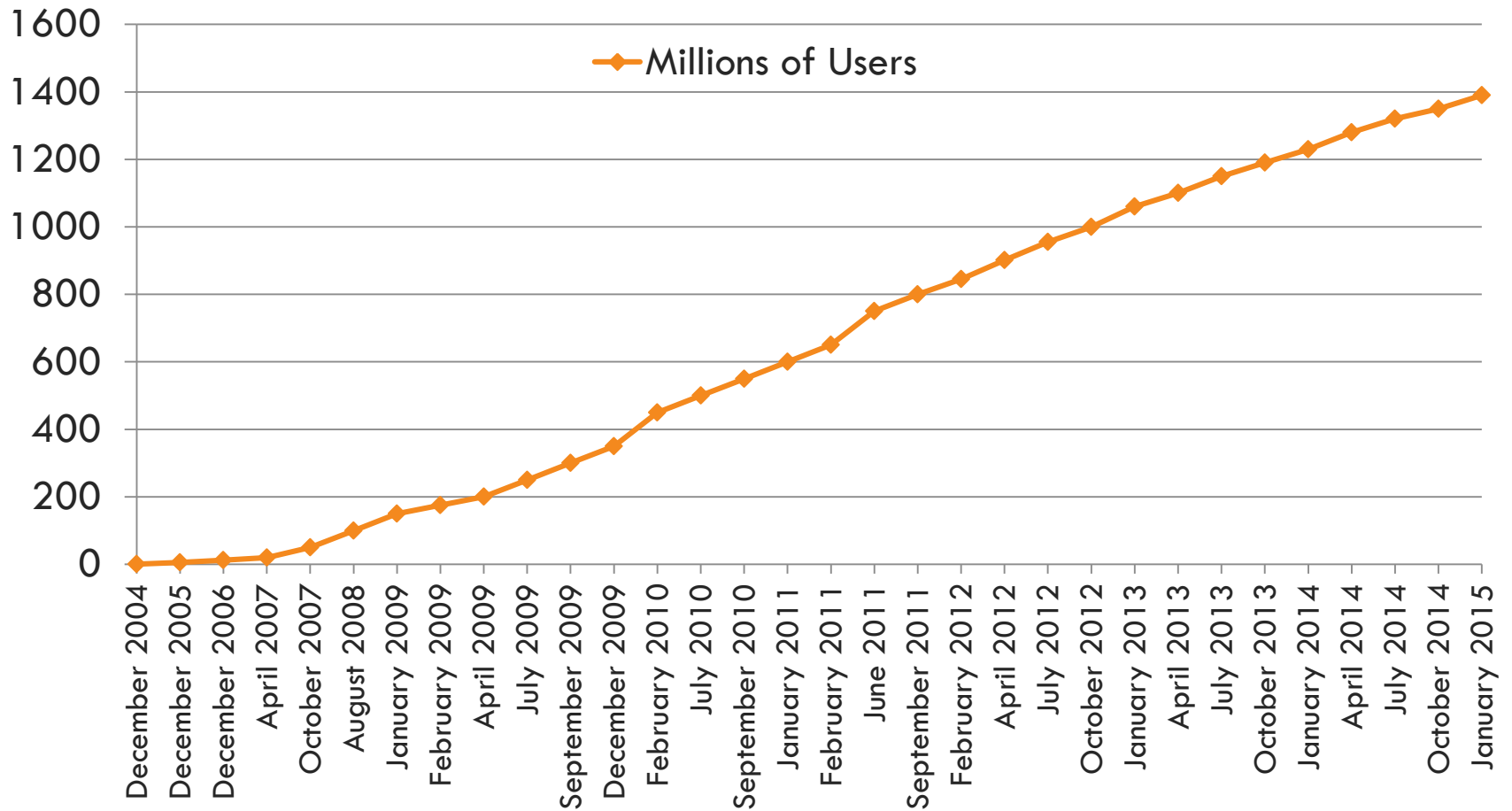
**100 million** search  
queries per day

**350 million**  
photos uploaded  
per day

**4.5 billion**  
'Likes' per day

# Growth Rate

22



# Reality Check

23

Each active user wants to write and upload something NOW

No matter how “big” the content is

Data and services have to be available 24/7, everywhere

No perceptible delay is allowed, in any case

# Reality Check

24

Each active user wants to see and share something NOW  
(she sees the recent status of her friends and pages, anyway)

No matter where the information is (geographically speaking)

No perceptible delay is allowed, in any case

Moreover, new features and applications are added  
continuously to Facebook



# Reality Check

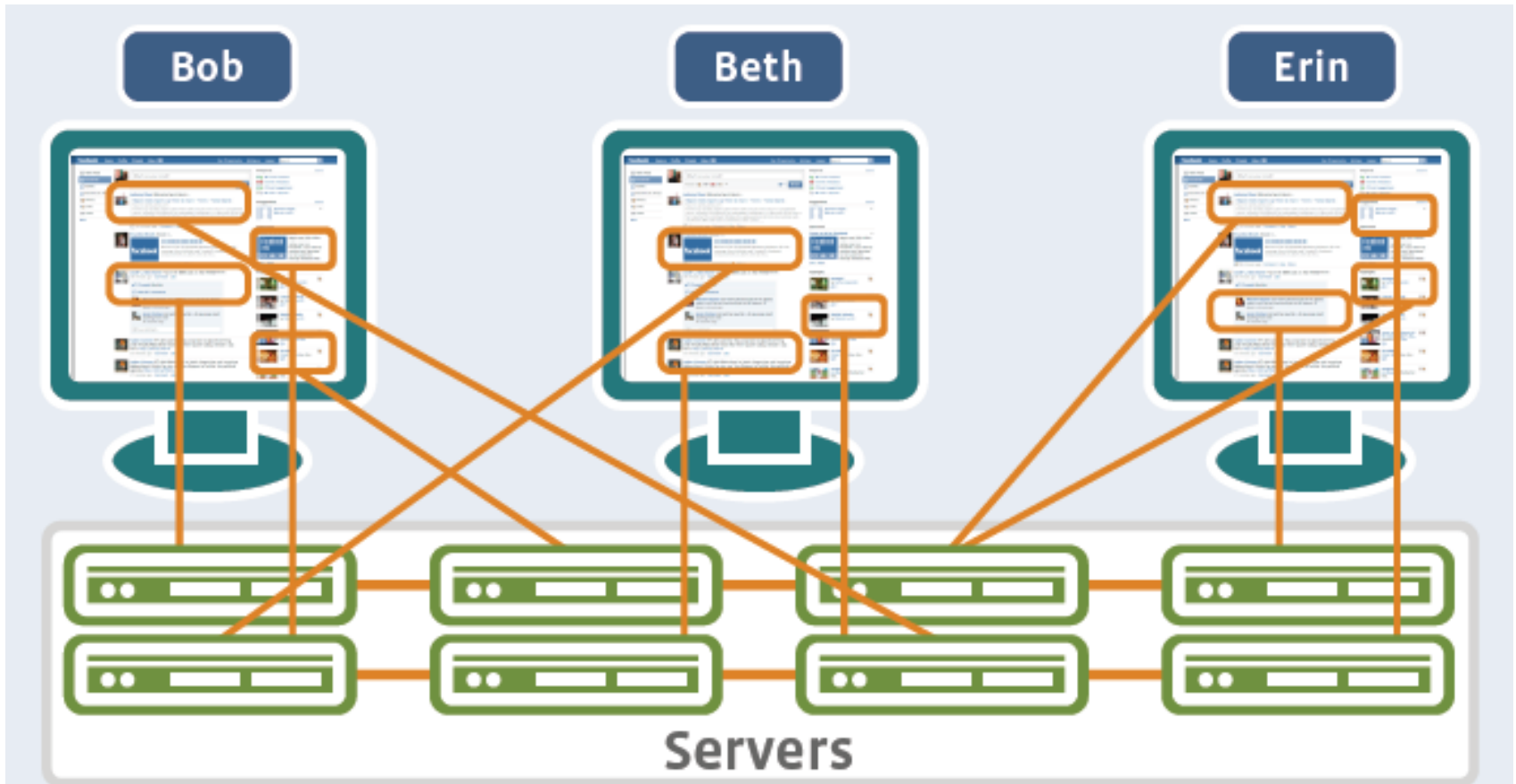
25

“Rendering a single page of Facebook involves hundreds of machines examining tens of thousands of pieces of data from dozen of services - all in real time.”

- from the Infrastructure page  
of the Facebook Newsroom (2014)

# Reality Check

26



# The main problem

27

## Scalability

The ability of a system, network or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth.

# The main problem

28



Affected by  
several factors

## Scalability

The ability of a system, network or process to handle a growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth.

29

# How to handle such situations?

# Solution

30

- No standard solutions...
- ... each social network made different choices
  - ▣ strongly dependent from the original core
- We are in the Cloud Computing realm
  
- We are going to analyze briefly the Facebook and the Pinterest cases

31

# How to handle such situations?

Facebook architecture at 100 feet

# Servers and Data Centers

32

- Facebook has 4 data centers
  - ▣ Prineville, Oregon
  - ▣ Forest City, North Carolina
  - ▣ Luleå, Sweden
  - ▣ Altoona, Iowa



- They build their servers and data centers from the ground up (efficiently)
- Servers and data center design is open source
  - ▣ see The Open Compute Project (<http://opencompute.org>)



# Complex Infrastructure

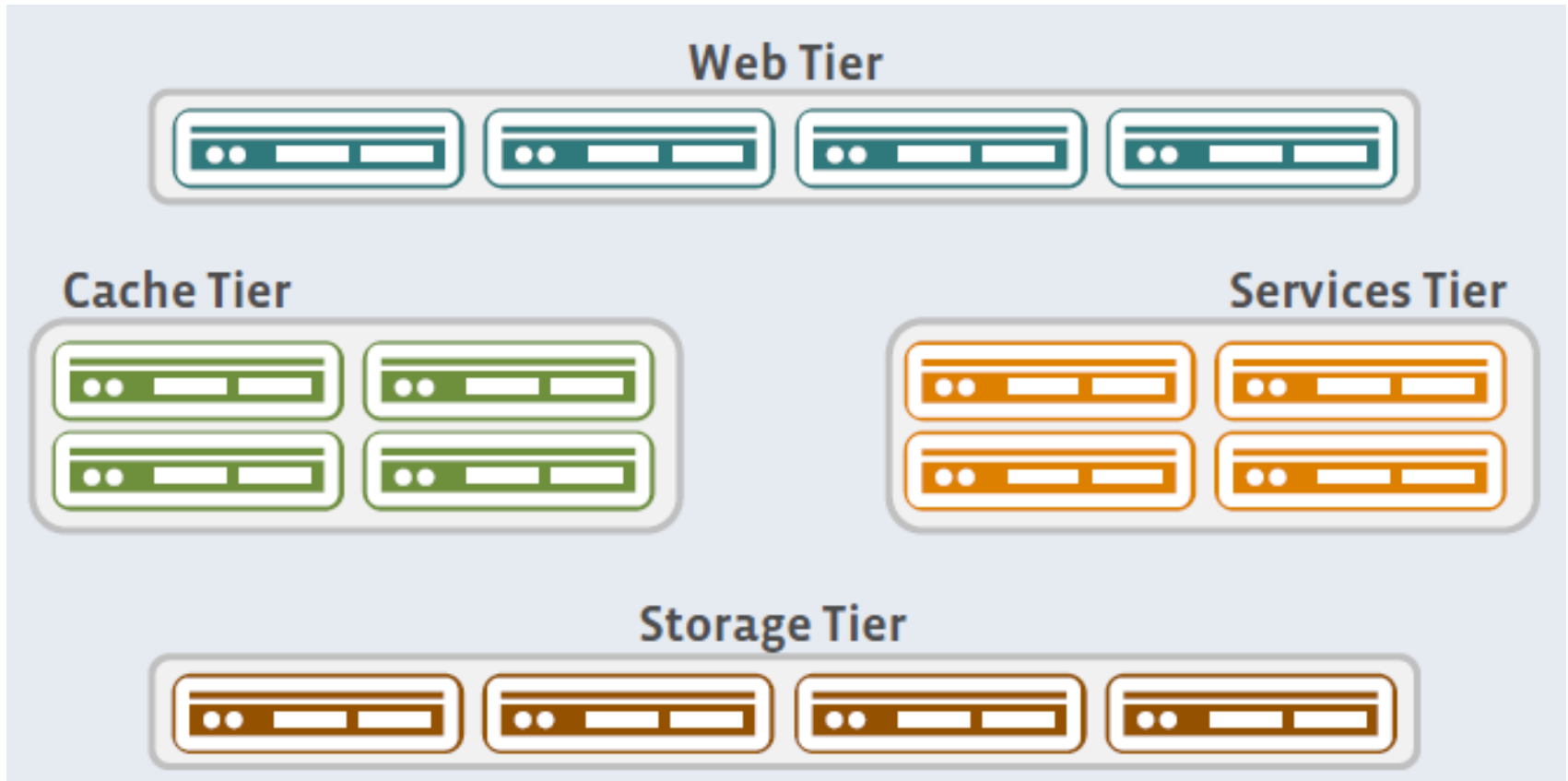
33

- Large number of software components
  - ▣ Multiple storage systems
  - ▣ Multiple caching systems
  - ▣ Hundreds of specialized services
- Failure is routine!

*Keep things as simple as possible!*

# Software Architecture

34



# Web Tier

35

- ❑ Gather Data from the other Tiers
- ❑ Runs PHP code
- ❑ Widely used for web development
- ❑ One single source tree for all the entire code
- ❑ Same “binary” on every web tier box



# Web Tier

36

- At the beginning: Zend Interpreter for PHP
  - ▣ reasonably fast (for an interpreter)
  - ▣ rapid development
    - no recompiling
  - ▣ but, at scale, performance matters!
  
- Then: HipHop compiler for PHP
  - ▣ 400% faster
  - ▣ but slow down development
    - they add an HipHop interpreter
    - but compiler and interpreter sometime disagree

# Web Tier

37

- Now: HipHop Virtual Machine
  - ▣ The best of both worlds
  - ▣ 9x increase in web request throughput
  - ▣ 5x reduction in memory consumption
  
- All this is open source
  - ▣ e.g., you can find HipHop Virtual Machine at <http://hhvm.com>

# Storage Tier

38

- Multiple storage systems
  - ▣ MySQL
  - ▣ Hbase (NoSQL) - Messaging and Insight
  - ▣ Haystack (BLOBS)

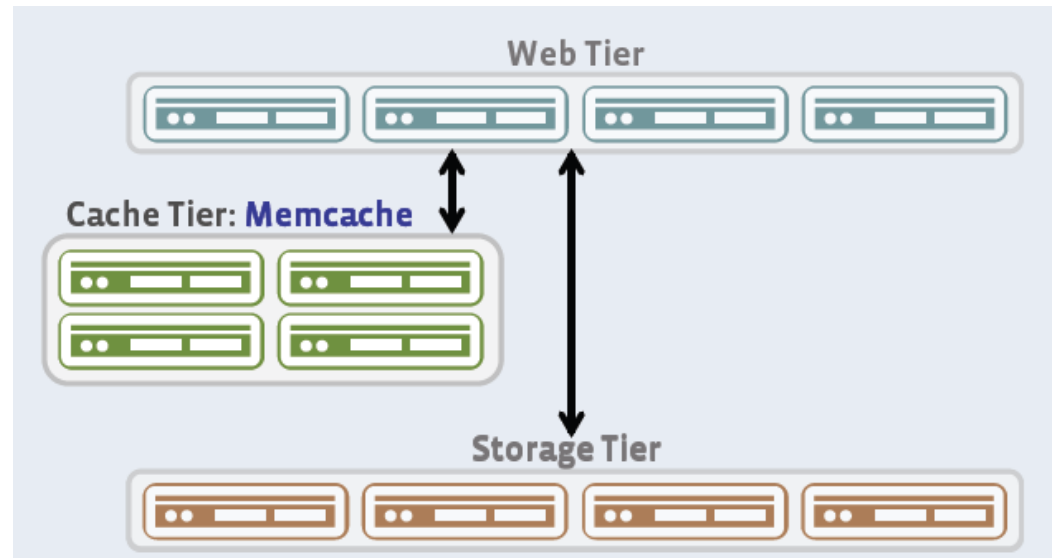


- BLOB: Binary Large Objects (Photos, Videos, Email attachments, etc.)
  - ▣ large files, no updates/appends, sequential reads

# Cache Tier

39

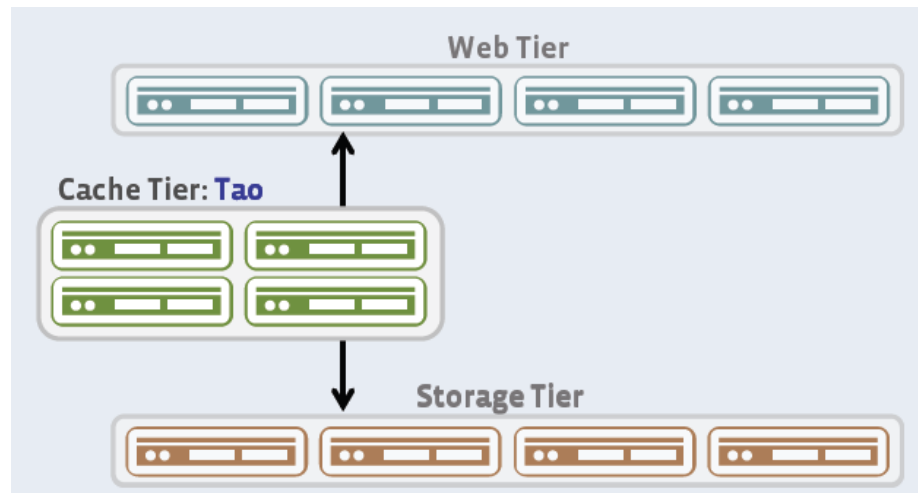
- Memcache
  - ▣ speak only with the Web Tier
  - ▣ do one thing very well
  - ▣ improved performance by 10x
  - ▣ key-value store



# Cache Tier

40

- Tao
  - ▣ abstract the Storage Tier
  - ▣ in production for more than a 3-4 years
  - ▣ higher CPU load than memcache
  - ▣ used for the social graph





# Service Tier

41

## Life before Services

Example: Wish your friend a Happy Birthday

Web Tier



Cache Tier



**Inefficient and Messy**

- Potentially access hundreds of machines
- Solution: Nightly cron jobs
- Issues with corner cases

**What about more complex problems?**

**Solution:** Build Specialized Services

Storage Tier



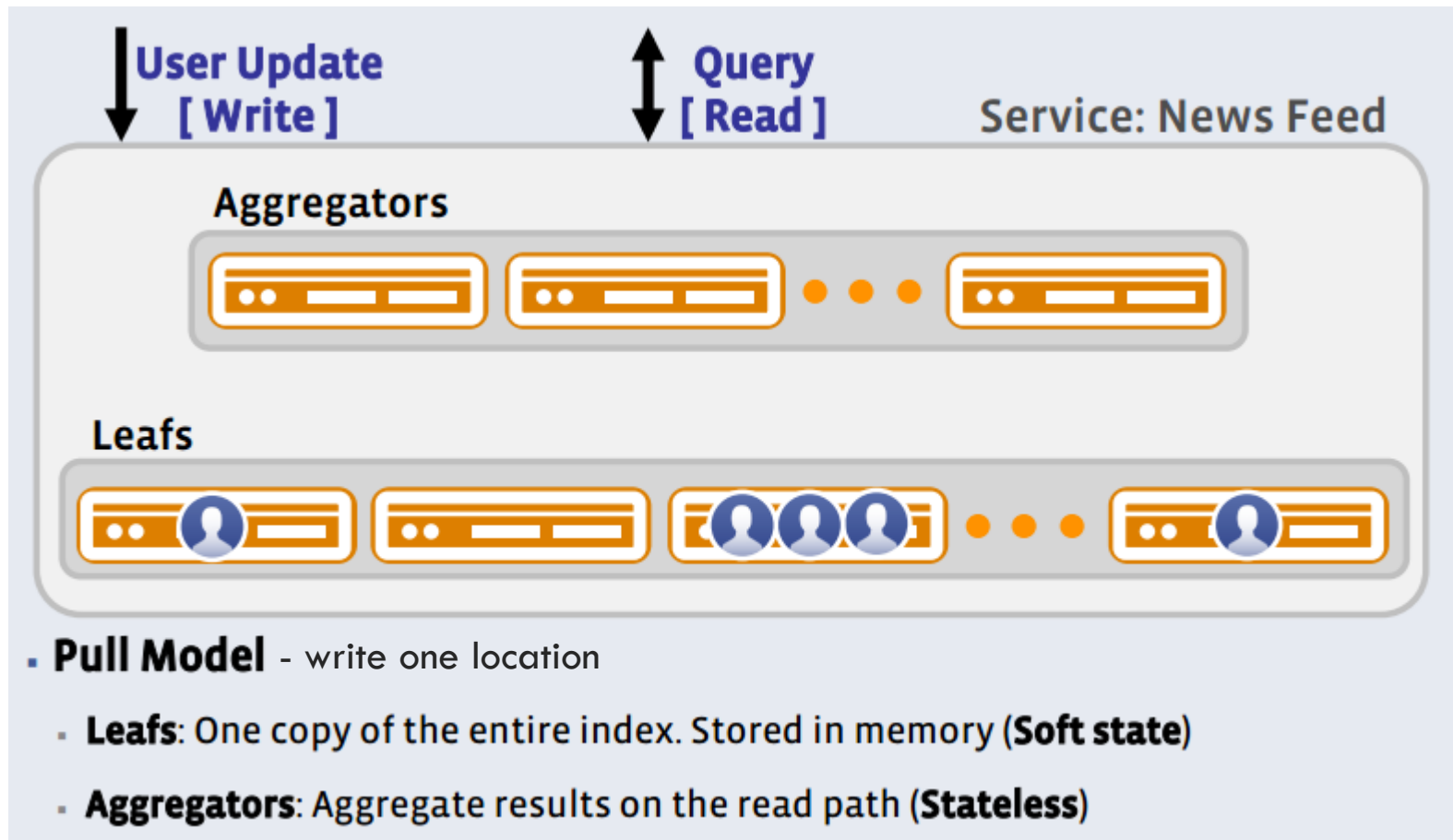
# Service Tier

42

- Example: News Feed
  - ▣ one of the hundreds of services at Facebook
- Characteristics
  - ▣ real-time distribution
  - ▣ **writers** can potentially broadcast to very large audience
  - ▣ **readers** wants (and have) different and dynamic ways to filter data
- The service should maintain an index and rank the data (in multiple ways)

# News Feed Service

43



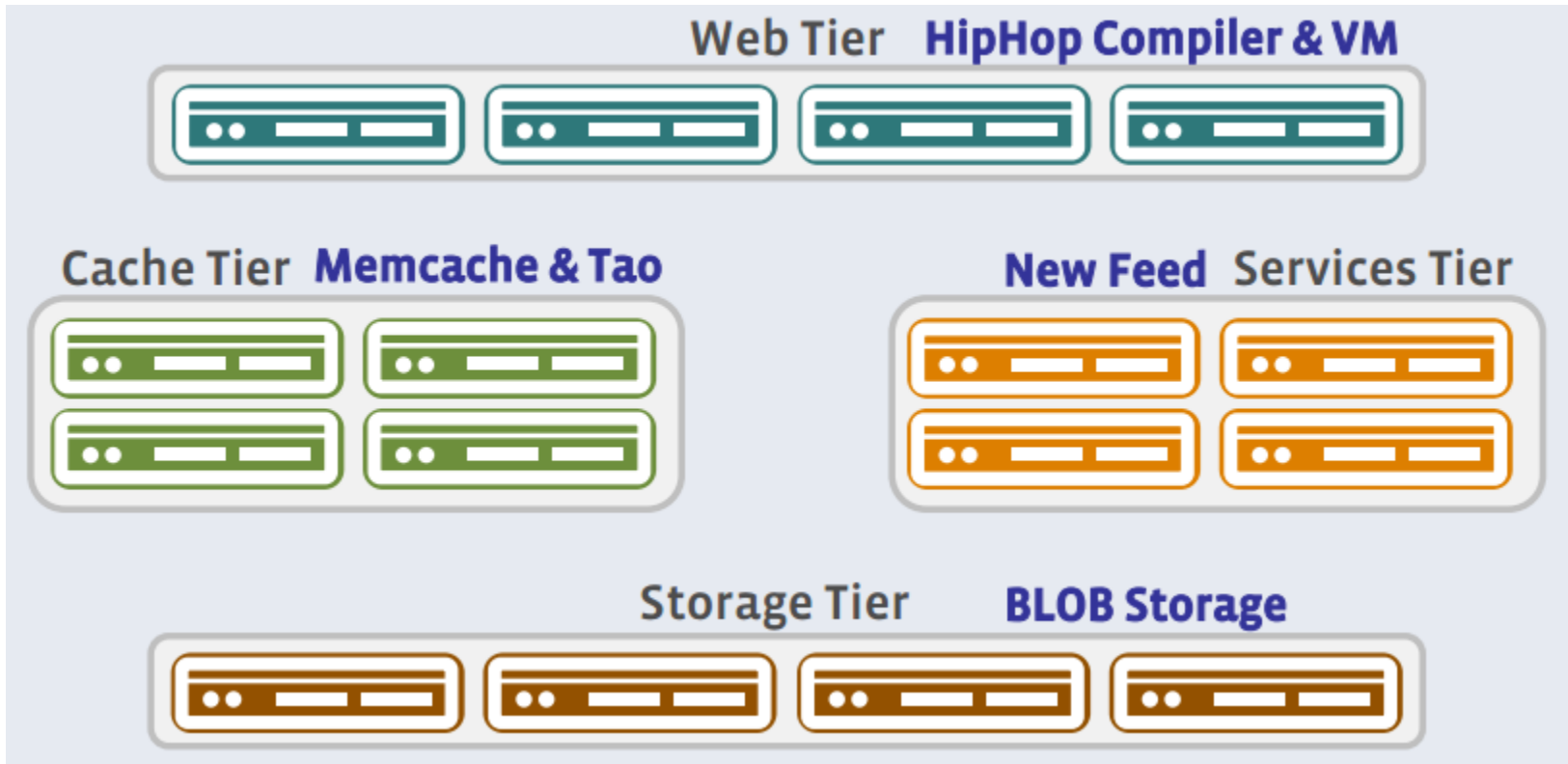
# News Feed Service

44

- 1 000s of machines
  - ▣ leafs are in multiple sets, and each set has the entire index
- Dealing with (daily) failures
  - ▣ hardware/software, server/network, intermittent/permanent, etc.
  - ▣ if a **leaf** is inaccessible, failover request to a different set
  - ▣ if an **aggregator** is inaccessible, “just” pick another
- More leafs than aggregators
  - ▣ Reads are more expensive than writes
- High network load between aggregator and leafs
  - ▣ fundamental to keep a full leaf set within a single rack on machines

# Software architecture

45



46

# How to handle such situations?

Pinterest architecture at 100 feet

# Infrastructure

47

- Pinterest follows some core principles when choosing a system:
  - ▣ keep it very simple
  - ▣ don't re-invent the wheel
  - ▣ go with proven and solid technologies when you can
- It runs on Amazon EC2

# Software Architecture

48

- Web Tier
  - ▣ Heavily-modified Django (Python)
  - ▣ Tornado is the chosen Web Server
- Storage Tier
  - ▣ MySQL
  - ▣ Amazon S3 for photos
- Cache Tier
  - ▣ Redis (key-value storage)
  - ▣ memcache



49

# Conclusions

# Conclusions

50

- Most Social Networks starts like “traditional” website
- They change and/or evolve their hardware and software infrastructure when:
  - ▣ users grow
  - ▣ some functionalities are added/revised
- Scalability is a relevant problem...
- They are complex entities and, sometimes, required complex or innovative solutions

# References




51

- Facebook Newsroom, <http://newsroom.fb.com>
- Facebook Investor Relations, <http://investor.fb.com>
- HPCA 2012 Facebook Keynote, [http://www.ece.lsu.edu/hpca-18/files/HPCA2012\\_Facebook\\_Keynote.pdf](http://www.ece.lsu.edu/hpca-18/files/HPCA2012_Facebook_Keynote.pdf)
- Scaling Pinterest, <http://highscalability.com/blog/2013/4/15/scaling-pinterest-from-0-to-10s-of-billions-of-page-views-a.html>

# License



52

- This work is licensed under the Creative Commons “Attribution-NonCommercial-ShareAlike Unported (CC BY-NC-SA 3,0)” License.
- You are free:
  - **to Share** - to copy, distribute and transmit the work
  - **to Remix** - to adapt the work
- Under the following conditions:
  - **Attribution** - You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).  

  - **Noncommercial** - You may not use this work for commercial purposes.  

  - **Share Alike** - If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.  

- To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc-sa/3.0/>