High Scalability by Example – How can Web Architecture scale like Facebook, Twitter?

accenture

High performance. Delivered.

JAX 2011, 03.05.2011

Robert Mederer, Markus Bukowski

Speaker

Robert Mederer
Lead Technology Architect
Anni-Albers-Straße 11
80807 München
Mobil: +49-175-57-68012
robert.mederer@accenture.com
accenture Technology Solutions

Markus Bukowski
Senior Software Engineer
Kaistraße 20
40221 Düsseldorf
Mobil: +49-175-57-64217
markus.bukowski@accenture.com
accenture Technology Solutions

Experience

2000 - 2005: Technology Architect and Software Engineer in several projects

2006: Technical Architecture Lead, Integration and Execution Architecture for Location-Based Service Provider

2009: Technical Architecture Lead, Frontend and Execution Architecture Government Agency

2009/2010: Technical Architecture and front-office integration build lead, Integration and Execution Architecture Financial Services Agency

2011: Architect and Performance Engineer for Location Based Services Platform

2006 - 2009: Software Engineer in several projects (during studies) for mainly telecommunication companies

2009 - 2011: Senior Software Engineer in several projects

2009/2010: Coach and Software Engineer for a Health Insurance Fund

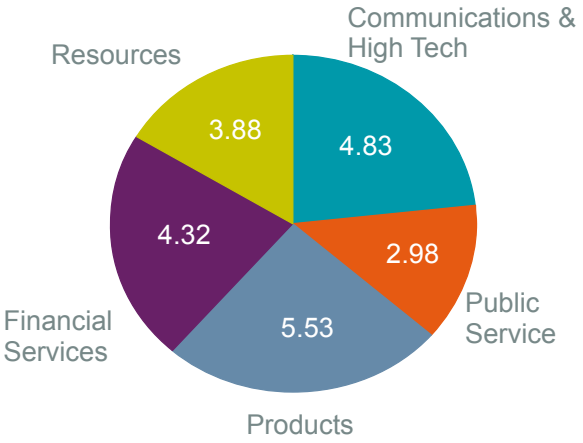
2011: Technical Architecture Lead during the development of a Document Management Solution for a Government Agency

Accenture
High performance achieved

Company Profile

- Global **management consulting, technology services and outsourcing** company
- **215.000** employees
- Rank 47 among the **"Best Global Brands 2008"**
- **Top 100** Employer
- 28 of the DAX-30-Companies
- 96 of the Fortune-Global-100
- More than three-quarters of the Fortune-Global-500
- **87 of our Top 100-clients** have been with us for 10 or more years

Worldwide Revenues $21.6 billion (in US$ billion, as of August 31, 2010)



Local Accenture … ???

Geographic unit

- Austria
- Switzerland
- Germany

Employees

- Ca. 6000
- We are hiring!

Exciting Technology work

- Large scale projects (100+ people / multiple years)
- Most challenging requirements
  - Stock Exchange / Banking / Trading Systems
  - AEMS Mobility Platform
  - Large Scale Web Applications (> 1M page views / day)
  - Batch Architectures

Agenda

- **Introduction**
  - **Scalability & CAP Theorem**
  - Traditional websites & Social media websites
  - High Scalability in Numbers
- Accenture – High Scalability by Example
- Architecture at Internet Scale
- Common concepts of Scalability
- Conclusion

Scalability

A system's capacity to uphold the same performance under heavier volumes.

(Patterns for Performance and Operability: Building and Testing Enterprise Software, Chris Ford et. al., 2008)

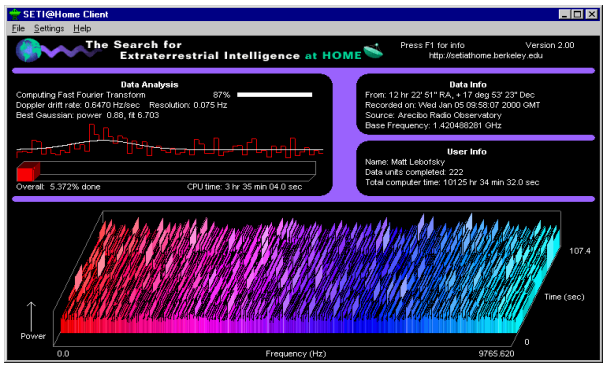
Vertical Scalability

- Is achieved by increasing the capacity of a single node
  - CPU,
  - Memory,
  - Bandwidth, …
- Simple Process
  - Application is generally not affected by those changes
- Classical Example are Super Computers like
  - HP Integrity Superdome
  - IBM Mainframe

Source: Hewlett-Packard

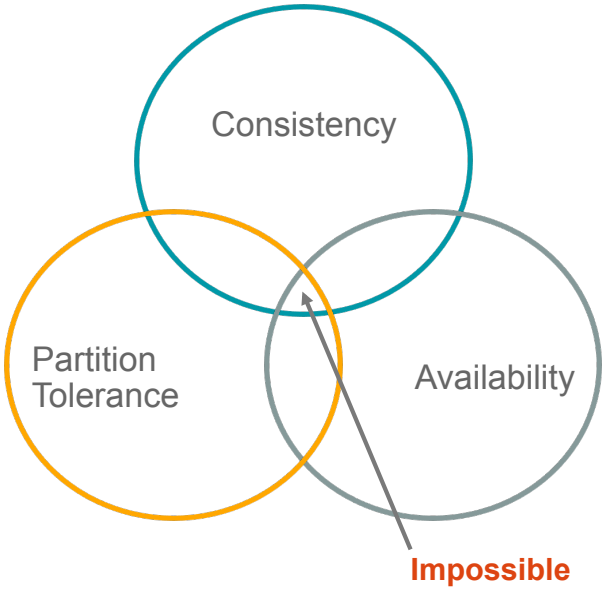
Horizontal Scalability

- Application is spread on a cluster with several nodes
- Nodes can be added to scale out
- Produces overhead
  - Keep cluster consistent
  - Node error detection and handling
  - Communication between nodes
- May be used to increase reliability and availability
- Distributed Systems and Programs like
  - SETI@Home
  - World Wide Web
  - Domain Name Service



Source: Space Sciences Laboratory, U.C. Berkeley

CAP Theorem (Brewer's Theorem)

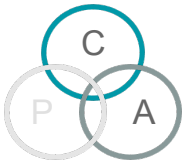


- **Consistency** – all clients see the same data at the same time
- **Availability** – all clients can find all data even in presence of failure
- **Partition Tolerance** – system works even when one node failed

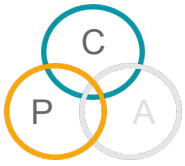
Source: PODC-keynote, Towards Robust Distributed Systems, Dr. Eric A. Brewer, 2000

CAP Theorem

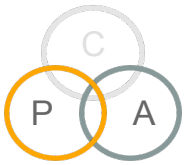
Normally, two of these properties for any shared-data system



Consistency + Availability
- High data integrity
- Single site, cluster database, LDAP, etc.
- 2-phase commit, data replication, etc.



Consistency + Partition
- Distributed database, distributed locking, etc.
- Pessimistic locking, etc.



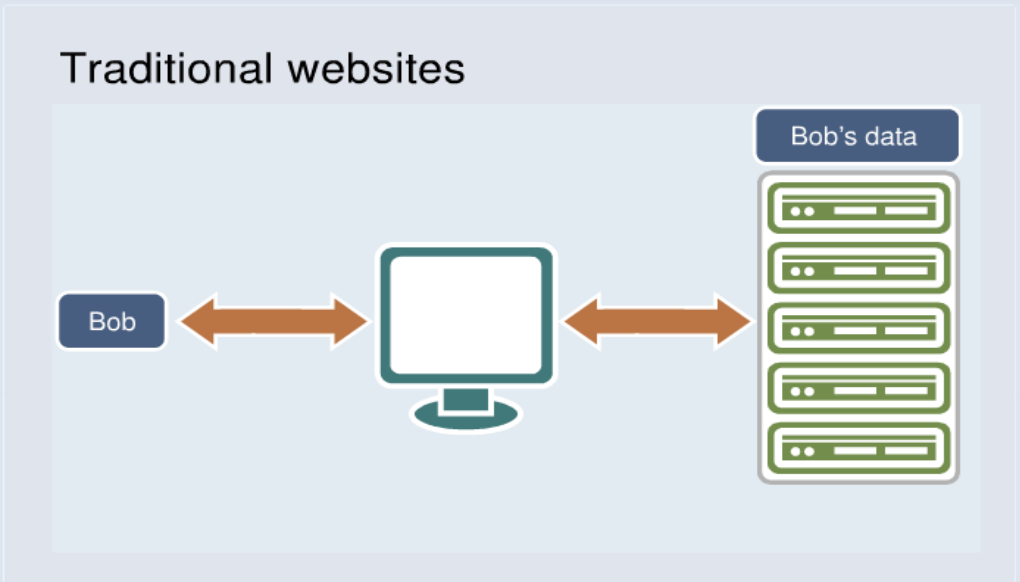
Availability + Partition
- High scalability
- Distributed cache, DNS, etc.
- Optimistic locking, expiration/leases (timeout), etc.

Source: "Architecting Cloudy Applications", David Chou

Agenda

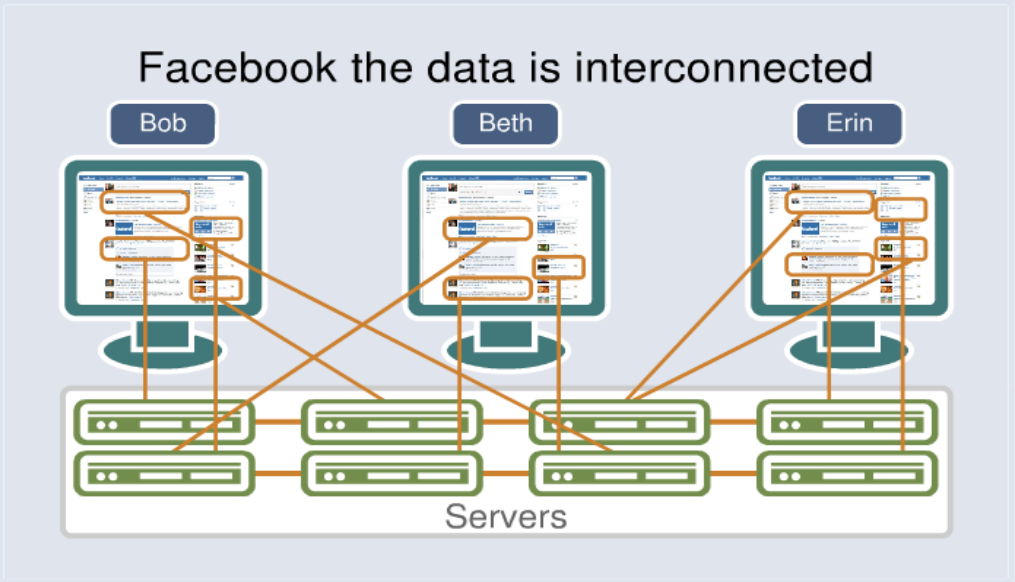
- Introduction
  - Scalability & CAP Theorem
  - **Traditional websites & Social media websites**
  - High Scalability in Numbers
- Accenture – High Scalability by Example
- Architecture at Internet Scale
- Common concepts of Scalability
- Conclusion

Traditional Websites



(Scaling the Social Graph: Infrastructure at Facebook, Jason Sobel, QCon 2011)

Social Media Websites



(Scaling the Social Graph: Infrastructure at Facebook, Jason Sobel, QCon 2011)

Agenda

- Introduction
  - Scalability & CAP Theorem
  - Traditional websites & Social media websites
  - **High Scalability in Numbers**
- Accenture – High Scalability by Example
- Architecture at Internet Scale
- Common concepts of Scalability
- Conclusion

High Scalability in Numbers

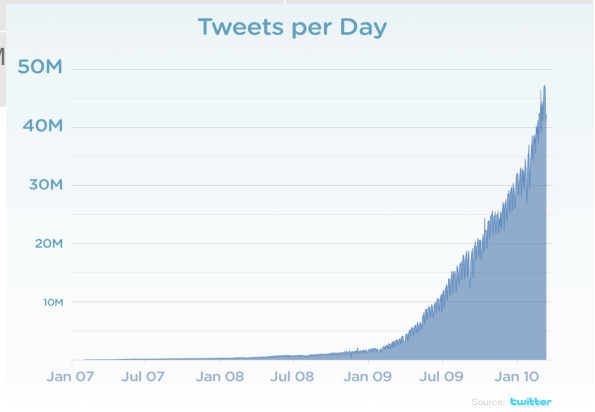
The World Is Obsessed With Facebook (Video)

Source: http://www.youtube.com/watch?v=xJXOavGwAW8

High Scalability in Numbers

facebook		twitter	
Active users	500.000.000	New accounts per day	500.000
Active users logging in daily	250.000.000	Tweets per day (a year ago)	140.000.000 (50.000.000)
Active mobile users	250.000.000	M	

In 20 minutes …	
Links shared	1.000.000
Status updates	1.851.000
Photos uploaded	2.716.000
Comments made	10.208.000



Source: http://www.facebook.com/press/info.php?statistics, http://www.allfacebook.com/

High Scalability in Numbers

LinkedIn	
Members	>100.000.000
Connections between Members	1.300.000.000
Visitors per month	128.000.000
% of global internet users who visit LinkedIn daily	4%

Source: http://blog.linkedin.com, http://press.linkedin.com/about/, LinkedIn Demographics 20111

Agenda

- Introduction
- **Accenture – High Scalability by Example**
  - The Royal Wedding Website
  - US Based Professional Sport League .com Website
- Accenture – High Scalability by Example
- Architecture at Internet Scale
- Common concepts of Scalability
- Conclusion

Agenda

- Introduction
- Accenture – High Scalability by Example
- **Architecture at Internet Scale**
  - Facebook
  - Twitter
  - LinkedIn
  - Challenges
- Common concepts of Scalability
- Conclusion

Facebook – Key Facts

- Biggest Social Network

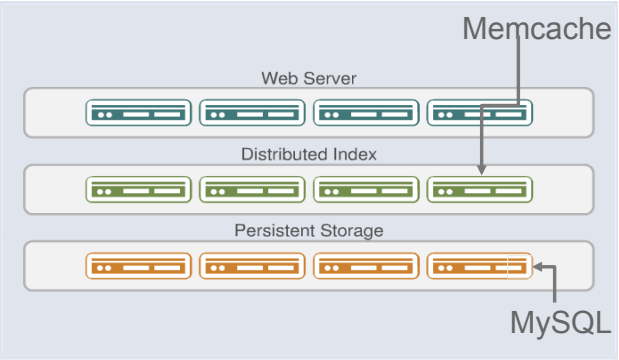
- People are linked together
- People share and exchange information in real-time

- Facebook provides a
  - A personalized **News Feed** containing news about friends
  - **Picture Service** including people tagging / linking
  - **Messaging Service** to stay in connect with friends
  - Platform to play **Games** with friends

Facebook – Architecture (1)

- Facebook „main“ architecture is based on a LAMP stack
- The application logic resides in Web Server layer **(PHP based)**
- The application layer takes care of the data distribution

- Some services do not fit **(C++, Java)**
  - Search
  - Ads
  - People You May Know
  - Multifeed



Invalidation logic is implemented in the application!

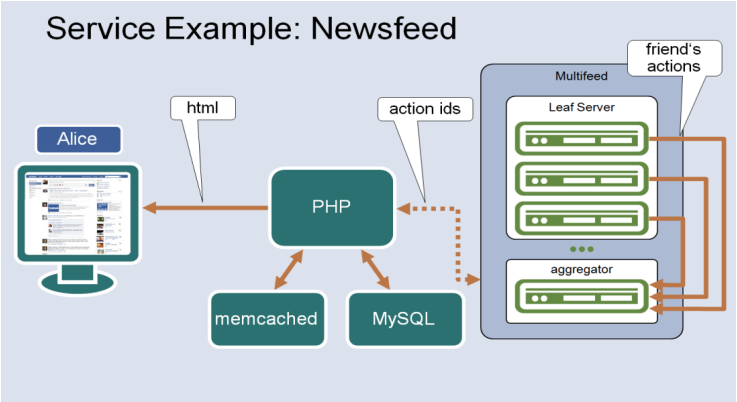
Facebook – Architecture (2)

Multifeed / Newsfeed

- Distributed System

- Every user gets
  - **45** best rated updates
  - on every reload

- Every DB-Update results in a Scribe Notification
  - Leaf Nodes / Server are notified



Source: “High Performance at Massive Scale – Lessons learned at Facebook”, Jeff Rothschild, 2009
Inspired By: “Architecting Cloudy Applications”, David Chou, 2010

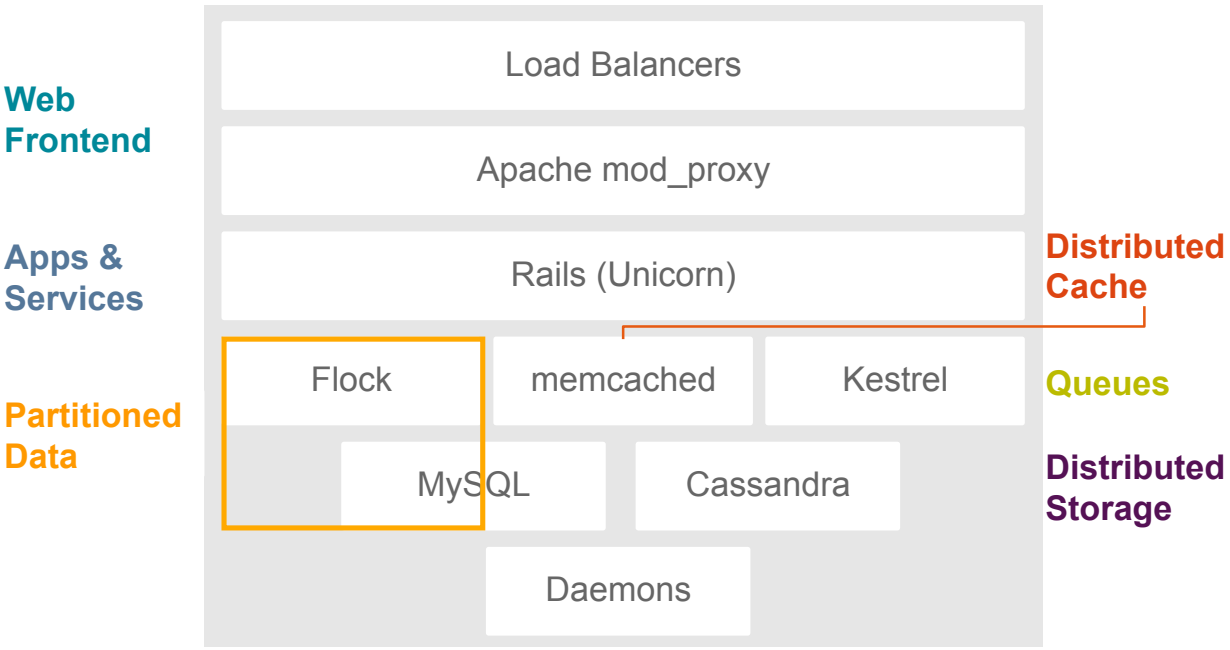
Agenda

- Introduction
- Accenture – High Scalability by Example
- **Architecture at Internet Scale**
  - Facebook
  - **Twitter**
  - LinkedIn
  - Challenges
- Common concepts of Scalability
- Conclusion

Twitter – Key Facts

- Twitter is a real-time information network
- Twitter's key characteristics
  - **Tweets** are used for sharing information
  - **Timeline** of tweets must be kept
  - A **Social Graph** is maintained to deliver information

Twitter - Architecture



Source: "Architecting Cloudy Applications", David Chou

Agenda

- Introduction
- Accenture – High Scalability by Example
- **Architecture at Internet Scale**
  - Facebook
  - Twitter
  - **LinkedIn**
  - Challenges
- Common concepts of Scalability
- Conclusion

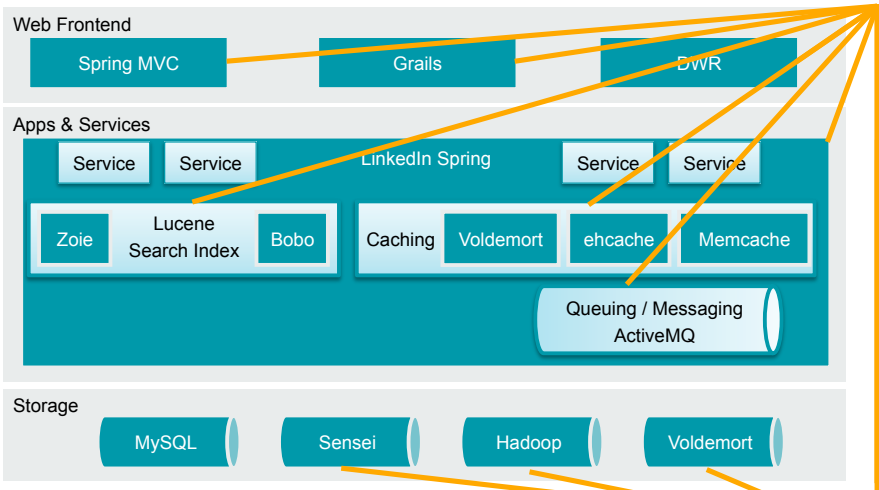
LinkedIn – Key Facts

- A social media networking platform to find connections to recommended job candidates, industry experts and business partners
- 99% pure Java

LinkedIn – Applications

- Groups
- Job market
- Mailing
- Profiles (Friends and Companies)
- Mobile Applications (Android, iPhone, Blackberry, Palm)

LinkedIn - Architecture



LinkedIn Spring

- Spring Framework (IoC, DI, AOP, Separation of Concerns,..)
- LinkedIn extensions

Storage Valdemort

- Valdemort used as a major distributed key-value data cache for the social graph
- Advantage:
  - Fast concurrent read
  - Fast index building

- Keep core data ACID
- Keep replicated and cached data BASE
- Cache data on a cheap memory (memcached)
- Use a hint to route the client to his / her's ACID data

Source: Project Voldemort, Jay Kreps Source: http://sna-projects.com

Agenda

- Introduction
- Accenture – High Scalability by Example
- Architecture at Internet Scale
- **Common concepts of Scalability**
- Conclusion

Common concepts of Scalability

parallelization

asynchronous

idempotent operations

7 Habits of Good Distributed Systems

partitioned data

fault-tolerance

optimistic concurrency

shared nothing

Source: “Architecting Cloudy Applications”, David Chou
Source: highscalability.com

Common concepts of Scalability
Hybrid architectures

- **Scale-out** (horizontal)
  - –BASE
  - –focus on “commit”
  - –optimistic locking
  - –shared nothing
  - –maximize scalability
- **Scale-up** (vertical)
  - –ACID
  - –availability first
  - –pessimistic locking
  - –transactional
  - –favor accuracy/consistency

Most distributed systems realize both

Agenda

- Introduction
- Accenture – High Scalability by Example
- Architecture at Internet Scale
- Common concepts of Scalability
- **Conclusion**
  - **Questionnaire**
  - **Déjà vu - Common concepts**
  - **Don’t forget the Infrastructure**

Conclusion
Questionnaire

- **Is there a need to scale my application?**
  - Vertical scaling is more easy to achieve
  - Use horizontal scaling only when required
- **Is there a plan to proof your designed solution?**
  - Plan to do a lot of **realistic** Proof-of-Concepts
- **Is there a one size fits all solution?**
  - NO!
- **How important is ACID?**
  - Is BASE enough?
  - Can a NoSQL solution be used?

Conclusion
Déjà vu - Common concepts

- **Asynchronous Processing**
  - Keep Facebook, Twitter and LinkedIn in mind
  - Asynchronous writes and even UI
- **Data Partioining**
  - You must understand your data
  - Is a hybrid solution as used by LinkedIn applicable?
- **Design To Failure**
  - Lessons learned from Amazon‘s cloud solution outage

Conclusion
Don’t forget the Infrastructure

- **Disk IO compared to RAM is slow**
  - Avoid „slow“ IO access whenever possible
  - Caching, Caching, Caching
  - Bulk Updates (Batch) over small updates

- **Network Latency**
  - Keep in mind that there is a network
  - Where are the users located?
  - How are your users connected? mobile application?
  - Is there a need for a datacenter distribution?

- **Hardware configuration**
  - Is the CPU Power Saving working right?

Questions ?