What Drives Innovation?

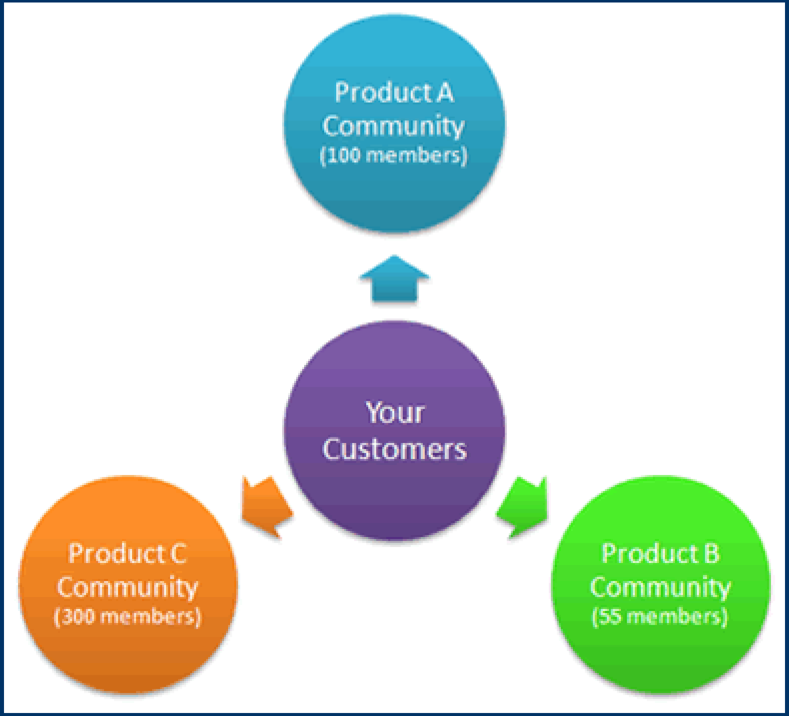
STEFANO FAGO

Reflecting on what drives Technological Innovation in the last 5 year and which interesting solutions, strategies and technologies were born

So... What Drives Innovation?

Social Networking & Social Marketing!

flickr
YouTube
facebook
digg
myspace.com a place for friends
friendster
reddit
StumbleUpon Discover new sites
meetup
del.icio.us social bookmarking


Product A Community (100 members)
Your Customers
Product C Community (300 members)
Product B Community (55 members)

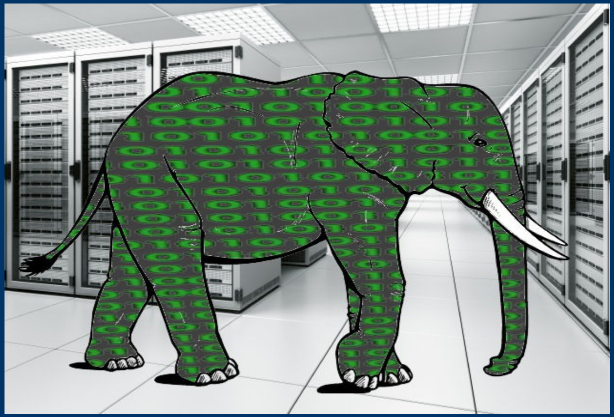
Social Networking & Social Marketing

...for both we have '***Big' Number***, so they need:

- ***Big Data*** to support
- ***Big Performance*** to support
- ***Big Scalability*** to support
- ***Big Ability*** to evolve fast

So... What about Big Data?

Big Data has introduced lots of needs!



Technically speaking we can find some discussion branches:

- Custom Persistence
- Polyglot Persistence
- Streaming and Analitical tools for Big Data

Custom Persistence?

The classical Database System has a lot of features but it requires different aspects to be managed:

- Important Installation
- Configuration
- Complex Partitioning
- Performance tuning
- Stable but Complex way to design data-topologies

Custom Persistence?

Wait!

...but you need:

- Fast ***Cache***,
- Fast ***Data-Harvesting***,
- Fast ***Data-Evolution***
- Fast ***Use&Forget Data-Design***

Same Data can evolve rapidly and can be used in many different ways, so I cannot waste time to normalize what changes fast!

NoSql, HybridSql, WhatEverSql: ***I Need My Persistence but I don't want to waste money!***

Polyglot Persistence?

Every Tool has a Role! Why do we forget it?

Social Networking and Marketing, both have volatile data but when we talk about money, we need:

- Transactional ACID-idity
- Solid Backup & Restore
- D&R solutions
- Known and Stable ways to install and manage.

So, let's speak different languages and we can use the right tool for the right job

Streaming and Analitical Tools

People change mind fast, people change their needs fast, so ***people produce and people consume informations fast***!

If I can manage and understand moods and needs, I can follow the market, I can find what will happen and maybe I can drive my market share better!

So I need to have continuous data flow and I need to be able to analyze those data...

New ***Distributed File System*** raised, New Super ***Batch System*** raised and ***Complex Events Processing*** systems have new Role and Installation, all around the World!

...and...What About Performance?

Processing ***Big Number*** means also using all CPU Power, exploiting MultiCore Architectures.

So how can I do it? ***Concurrency***? ***Parallelism***?

- Concurrency can cost a lot, so I need ***Lock-Free*** Data Structure and I need some way to use ***all CPU Cores***.

- New interest in Functional Programming and Actor Concurrency Model and/or Message Paradigms.

- A more simple use of Threads: less Threads more Events.

...and...What about Performance?

First winner is Erlang: it's Functional, has ***Actors Framework*** and Message Passing Implementation.

In main scripting languages, Python and Ruby have different solutions that work on ***Events*** and ***Non-Blocking I/O***.

It's started also a technological race in the mainstream languages on developing ***Concurrency Frameworks*** and ***Parallel Frameworks***.

THE REBELLION BEGINS
Erlang
The Movie
ERLANG SOLUTIONS PRESENTS
AN ETC PRODUCTION "ERLANG - THE MOVIE" MIKE WILLIAMS JOE ARMSTRONG ROBERT VIRDING BJARNE DÄCKER FEATURING FUNCTIONAL PROGRAMMING
BASED ON A NOVEL BY AGNER KRARUP ERLANG SPECIAL EFFECTS BY ERICSSON SCREENPLAY BY SYNAPSE MOBILE NETWORKS PRODUCTION DESIGN BY KLARNA DIRECTOR OF PRODUCTION MOBILE ARTS EXECUTIVE PRODUCER TAIL-F SYSTEMS
PRODUCED BY CORELATUS CREATIVE DESIGNER PROCESSONE SET MANAGER SJÖLAND & THYSELIUS TESTING SUPERVISOR QUVIQ
ERLANG
tail-f
corelatus
ERICSSON
klarna
process one
Sjöland&Thyselius Creative Software Solutions
O'REILLY
Mobile Arts
QuviQ
synapse mobile networks
Erlang

...and...What about Performance?

- Messaging Systems are also evaluated. So, standard ways to make messaging aren't good. ***Enterprise MOM*** are ***slow and incomplete***.
- New custom products raised using ***raw sockets***, ***nosql db***s and/or ***brokerless*** solutions.
- CPU load also means the right policies to ***divide workload between clients and servers***.
- ***Stateless Systems*** and ***REST*** infrastructures, they win being able to process data fast and use completly CPU abilities
- Big Amount of ***Central Memory used like a Cache*** and/or lie an ***Embedded Database***, also to optimize CPU work.

...and...What About Scalability?

Scaling Out, Scaling Up however I need to Scale!

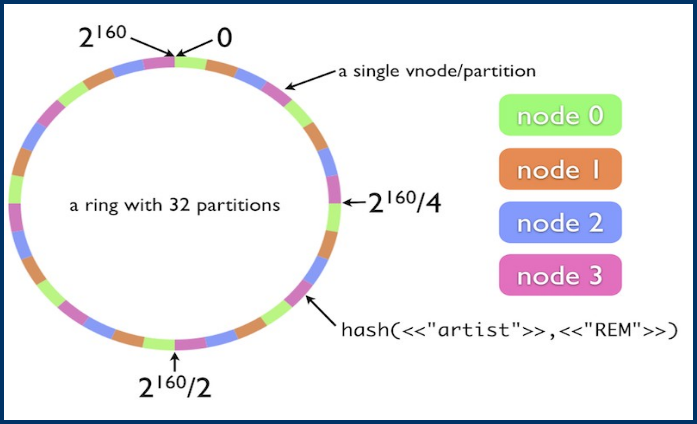
Different directions were followed:

1) Algorithms and Data Structure that make it possible to ***divide systems in nodes*** that are ***easy to classify*** and ***easy to find***:

Consistent hashing

Probabilistic Data Structures

Fast Protocols, often Text-Based



2) Using Central Memory like a database and using silos of memory

...and...What About Scalability?

3) Studying ***specific I/O profile*** to create tuning plans and specific topologies about SQL and NOSQL databases load, ***both in writing and reading***

4) Using ***Scripting Languages*** or scripted versions of mainstream languages, for end-point elements and some core parts:

- **Change, rapidly, functional code; when it's possible, code is changed to be optimized**
- **Easy to Replicate and Distribute on different nodes**

...and...What About Scalability?

5) Using light communications:

- Simple protocols
- Brokerless messaging

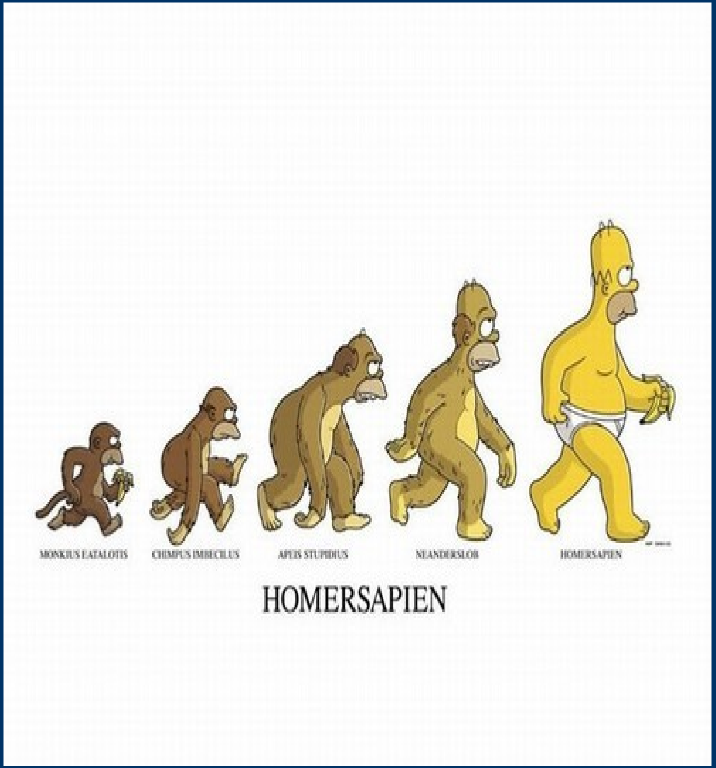
6) Specific load profile to divide Client and Server abilities. Profile and algorithms relative to:

- Data Locality
- Frequency Updates in Client and Server
- Dimensions of data messages

Evolving Fast?? It's possible??

Create elements that fit specific needs; so it's necessary to have tools for:

- Research
- Innovation
- Spikes code
- Logical Mindset to reuse effectively what is 'in house' and available, trash what doesn't work: it always learns but not to be afraid to trash code.



Evolving Fast??

Tools for fast refactoring of the code:

- IDE
- Best Practices (...better to say Good Practices)
- Skilled people, Engaged people: ***I'm good to make things but I also need to stay more hours on screen; so please let it be pleasant!***
- Agile-Team communications and team management: chats, mails,microblogging, stantard blogging

Evolving Fast??

Tools to ***manage*** and ***understand*** Problems:

- Bug tracking tools
- Versioning tools
- Monitoring Tools
- Script and Languages to create the best Test Suite (Tests need resources)
- Client Simulations, Network Simulations

Evolving Fast??

Creation of Servers, Databases setup and rapid management of Nodes:

- Choosing the best OS and platform that fits specific needs
- Skilled people, Engaged people
- Virtualization

...and then??

There's a lot of work to do... so let

Thank You All for the Attention!