Distributed Computing in the Cloud

Universitat de València

11-04-2017
Iwan van der Kleijn

@soyrochus

ivanderk@capgemini.com
devonfw is targeted at

...creating multi-device applications and multi-platform (micro) services in the Cloud or existing IT infrastructures with the focus on short time-to-market and great user experience...
devonfw consists of ...

a series of Open Source & industry standard frameworks, used to create stand-alone, production-grade applications which can "just run", but which are fundamentally modular in nature so they can be effortlessly extended with external components & libraries.

This is supported by a rich and highly automated toolchain, standardized architecture blueprints and industry proven best practices, patterns and code conventions.
devonfw - Technology stacks to choose from

development

Server

- Java
- .NET Core
- Node.js

Client

- Angular
- Sencha
- Xamarin

runtime

App servers

- Oracle WebLogic
- WebSphere
- Apache Tomcat

Cloud

- docker
- Cloud Foundry

Microservices

- Netflix OSS
- Spring Boot
- fabric8

Persistence

- Oracle DB2
- SQL Server
- Redis
devonfw is build on a community. All aspects of devonfw are build in cooperation with and are open to people from all over Capgemini and to a large extend from people outside of Capgemini (the Open Application Standard Platform).
The active Community around devonfw is the most valuable asset for our initiative! International collaboration in Capgemini especially Rightshore is working!
Paradigm

The **interface** is the system

The **API** is the system
devonfw as part of your integration infrastructure (rather than hub or PaaS)

In a recent survey *) the majority of respondents cited integration issues in the Cloud as a top concern, second only to data security and privacy. It becomes critical to avoid that Enterprise data becomes segregated into cloud silos and to guarantee that applications will talk to each other within the cloud and between the cloud and enterprise.

A strong value proposition of devonfw is that it can help solve these connectivity issues. Not by being a single monolithic integration hub with a single point of failure, but rather by being part of the standard infrastructure of a company, providing the "glue" between all of the components within the Enterprise ecosystem without “turning into a PaaS”.

*) Report by Saugatuck Technology with Gartner
Automate & Industrialize

- More focus on standards, patterns, blueprints
- Even more focus on code generation & meta programming
- UI & API focus
- Flat pyramid
- Development centers
dev-on time; on target
(A|The) (New|Real) World
What you will encounter when you leave the University even at “big, boring” companies
‘Big Data’ is similar to ‘small data’, but bigger in size but having data bigger it requires different approaches:

– Techniques, tools and architecture

and aim to solve new problems or old problems in a better way

Big Data **generates value** from the storage and **processing (analysis)** of very large quantities of digital information that cannot be handled with traditional computing techniques
What is BIG DATA

Walmart handles more than 1 million customer transactions **every hour**.

Facebook handles 400 billion photos from its user base. **Each day 300 million** new photos

Decoding the human genome originally took 10 years to process; now it can be achieved in one week.
What is BIG DATA

• A typical PC might have had 10 gigabytes of storage in 2000.

• Today, Facebook ingests 500 terabytes of new data every day.

• Boeing 737 will generate 240 terabytes of flight data during a single flight across the US.

• The smart phones, the data they create and consume; sensors embedded into everyday objects will soon result in billions of new, constantly-updated data feeds containing environmental, location, and other information, including video.
An IoT platform for Building Energy Management Solutions

- The Siemens Building Technologies Division’s web-based Energy & Sustainability Navigator platform will continue to unlock the hidden potential of building data, provide transparency to support decision making processes that increase building operating efficiency, and drive cost savings for customers. Using this platform Siemens Building Technologies’ customers currently save 10.5 million tons of CO2 per year.

- Using Devonfw Siemens is now further developing this Navigator platform to help corporate real estate owners drive business results and meet energy efficiency goals, while maximizing the lifecycle potential of their customers’ real estate assets.

Devonfw is “buzzword” ready. It is being used:

- centered around IoT
- in the Cloud
- for real-time systems
- with Micro-services
- and huge volumes of data
IoT – Big Data – Micro service show case

TollCollect manages automatic toll tracking & payment

- 1 million trucks with on-board GPS device
- Send location data and metrics in real-time
- Trucks route is calculated continuously
- Distance travelled over toll highways is automatically detected and calculated
- Electronic bills are send automatically to the truck owners

Devonfw is “buzzword” ready. It is being used:

- centered around IoT
- in the Cloud
- for real-time systems
- with Micro-services
- and huge volumes of data

* Commercial product
The End of Moore’s Law?
Moore's law

The number of transistors on a chip will double approximately every 18 months

Gordon E. Moore, 1965
A super computer in your pocket

77 gigaflops > 80 megaflops
No matter how fast processors get, software finds new ways to eat up the extra speed

“...Andy Giveth, and Bill Taketh away ..”
The End of Moore Law?

Free and regular performance gains, even without releasing new versions or doing anything special

The Free Lunch
The End of Moore Law?

Intel CPU Trends
(sources: Intel, Wikipedia, K. Olukotun)

- Dual-Core Itanium 2
- Pentium 4
- Pentium
- 386

The information contained in this presentation is proprietary. © 2017 Capgemini. All rights reserved. Rightshore® is a trademark belonging to Capgemini.
Why You Don’t Have 10GHz Today

Hardware crossed a boundary in the early 2000s:

– chips got big enough, cycle speed got fast enough
– a signal can no longer reach the whole chip in a clock cycle
– problems with heat dissipation
The End of Moore Law?

The Multicore Era

Processor manufacturers have turned towards multi-core processors

Capable of doing multiple calculations in parallel

CPU speeds are likely to stay relatively flat in the near future
The Concurrency Revolution

The performance lunch isn’t free any more
Want to benefit from the continued throughput advances in new processors?
You will need to develop well-written concurrent applications
In parallel computing, Amdahl's law is mainly used to predict the theoretical maximum speedup for program processing using multiple processors.

Amdahl’s law states that in parallelization, if $P$ is the proportion of a system or program that can be made parallel, and $1-P$ is the proportion that remains serial, then the maximum speedup that can be achieved using $N$ number of processors is $\frac{1}{(1-P)+(P/N)}$.

If $N$ tends to infinity then the maximum speedup tends to $\frac{1}{(1-P)}$.

Speedup is limited by the total time needed for the sequential (serial) part of the program. For 10 hours of computing, if we can parallelize 9 hours of computing and 1 hour cannot be parallelized, then our maximum speedup is limited to 10x.
We will NOT SOLVE (and therefore on this talk NOT focus) all the challenges we face related with the huge growth of data with concurrency on multi-core systems.

We will take a look at another approach to tackle parallelism & scaling: distributed systems.
Distributed Computing
A distributed system is one in which hardware or software components located at networked computers communicate and coordinate their actions only by message passing.

In the term distributed computing, the word distributed means spread out across space. Thus, distributed computing is an activity performed on a distributed system.

These networked computers may be in the same room, same campus, same country, or in different country.
The use of concurrent processes that communicate by message-passing has its roots in operating system architectures studied in the 1960s.

The study of distributed computing became its own branch of computer science in the late 1970s and early 1980s.

The first conference in the field, Symposium on Principles of Distributed Computing (PODC), dates back to 1982, and its European counterpart International Symposium on Distributed Computing (DISC) was first held in 1985.
In distributed system each processor have its own memory. The computational entities are called computers or nodes. In distributed computing a program is split up into parts that run simultaneously on multiple computers communicating over a network.

Distributed computing is a form of parallel computing.
Working Of Distributed System:

- Machine A
- Machine B
- Machine C

1. Distributed applications
2. Middleware service
3. Local OS
4. Network
Types Of Distributed Computing:

Grid computing

Multiple independent computing clusters which act like a “grid” because they are composed of resource nodes not located within a single administrative domain. (formal)

The creation of a “virtual supercomputer” by using spare computing resources within an organization.

Cloud computing

Cloud computing is a computing paradigm shift where computing is moved away from personal computers or an individual application server to a “cloud” of computers. Users of the cloud only need to be concerned with the computing service being asked for, as the underlying details of how it is achieved are hidden. This method of distributed computing is done through pooling all computer resources together and being managed by software rather than a human.
The main motivations in moving to a distributed system are the following:

- Inherently distributed applications.
- Performance/cost.
- Resource sharing.
- Flexibility and extensibility.
- Availability and fault tolerance.
- Scalability.
Goals & Characteristics

- **Making Resources Accessible.** The main goal of a distributed system is to make it easy for the users (and applications) to access remote resources, and to share them in a controlled and efficient way.

- **Distribution Transparency.** An important goal of a distributed system is to hide the fact that its processes and resources are physically distributed across multiple computers.

- **Openness.** An open distributed system is a system that offers services according to standard rules that describe the syntax and semantics of those services.

- **Scalability.** Scalability of a system can be measured along at least three different dimensions.

- **Concurrency:** Concurrency arises naturally in distributed systems from the separate activities of users, the independence of resources and the location of server processes in separate computers.

- **Fault tolerance:** Fault tolerance cares the reliability of the system so that in case of failure of hardware, software or network, the system continues to operate properly, without significantly degrading the performance of the system.
Architecture

- Client Server Architecture
- 3-Tier Architecture
- N-Tier Architecture
- Tightly Coupled
- Peer To Peer
Examples of distributed systems and applications of distributed computing include the following:

Telecommunication networks:

– Telephone networks and cellular networks
– Computer networks such as the Internet

Network applications:

– World wide web and peer-to-peer networks
– Massively multiplayer online games and virtual reality communities

Real-time process control:

– Aircraft control systems
– Industrial control systems

Parallel computation:

– Scientific computing, including cluster computing and grid computing and various volunteer computing projects
– Distributed rendering in computer graphics
Advantages vs challenges

Economics

Speed

Inherent distribution of applications

Reliability

Extensibility and Incremental Growth

Distributed custodianship

Data integration

Complexity

Network problem

Security
Into the Cloud
Amazon AWS
What is AWS Offering?

Low Ongoing Cost: pay-as-you-go pricing with no up-front expenses or long-term commitments.

Instant Elasticity & Flexible Capacity: (scaling up and down) Eliminate guessing on your infrastructure capacity needs.

Speed & Agility: Develop and deploy applications faster Instead of waiting weeks or months for hardware to arrive and get installed.

Apps not Ops: Focus on projects. Lets you shift resources away from data center investments and operations and move them to innovative new projects.

Global Reach: Take your apps global in minutes.

Open and Flexible: You choose the development platform or programming model that makes the most sense for your business.

Secure: Allows your application to take advantage of the multiple layers of operational and physical security in the AWS data centers to ensure the integrity and safety of your data.
The Amazon Web Services Universe

Management Interface

Cross Service Features

Platform Services

Infrastructure Services
AWS offering

Global Infrastructure – Regions | Availability Zones | Edge Locations
Amazon Elastic Compute Cloud (EC2)

A web service that provides **resizable compute capacity** in the cloud.

EC2 allows **creating Virtual Machines (VM) on-demand**. Pre-configured templated Amazon Machine Image (AMI) can be used to get running immediately. Creating and sharing your own AMI is also possible via the AWS Marketplace.

Auto Scaling allows **automatically scale of the capacity up** seamlessly during demand spikes to maintain performance, and **scales down** during demand lulls to minimize costs.

**Elastic Load Balancing** automatically distributes incoming application traffic across multiple Amazon EC2 instances.

Provide tools to build **failure resilient applications** by launching application instances in separate Availability Zones.

Pay only for resources actually consume, **instance-hours**.

**VM Import/Export** enables you to easily import virtual machine images from your existing environment to Amazon EC2 instances and export them back at any time.
Amazon Elastic Block Store (EBS)

Provides block level storage volumes (1 GB to n TB) for use with Amazon EC2 instances.

- Multiple volumes can be mounted to the same instance.
- EBS volumes are network-attached, and persist independently from the life of an instance.
- Storage volumes behave like raw, unformatted block devices, allowing users to create a file system on top of Amazon EBS volumes, or use them in any other way you would use a block device (like a hard drive).

EBS volumes are placed in a specific Availability Zone, and can then be attached to instances also in that same Availability Zone.

Each storage volume is automatically replicated within the same Availability Zone.

EBS provides the ability to create point-in-time snapshots of volumes, which are persisted to Amazon S3.

- These snapshots can be used as the starting point for new Amazon EBS volumes, and protect data for long-term durability.
- The same snapshot can be used to instantiate as many volumes as you wish.
- These snapshots can be copied across AWS regions.
Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web.

Write, read, and delete objects containing from 1 byte to n terabytes of data each. The number of objects you can store is unlimited.

Each object is stored in a bucket and retrieved via a unique, developer-assigned key.

- A bucket can be stored in one of several Regions.
- You can choose a Region to optimize for latency, minimize costs, or address regulatory requirements.
- Objects stored in a Region never leave the Region unless you transfer them out.

Authentication mechanisms are provided to ensure that data is kept secure from unauthorized access.

- Objects can be made private or public, and rights can be granted to specific users.

S3 charges based on per GB-month AND per I/O requests AND per data modification requests.
Amazon VPC lets you provision a logically isolated section of the Amazon Web Services (AWS) Cloud.

You have complete control over your virtual networking environment, including:

- selection of your own IP address range,
- creation of subnets, and
- configuration of route tables and network gateways.

VPC allows bridging with an onsite IT infrastructure with an encrypted VPN connection with an extra charge per VPN Connection-hour.

There is no additional charge for using Amazon Virtual Private Cloud, aside from the normal Amazon EC2 usage charges.
HPC in the Cloud
High Performance Computing (HPC) vs. High Throughput Computing (HTC)

HPC: High performance computing (cluster computing)
- Tightly clustered
- Latency sensitive

HTC: High throughput computing (grid computing)
- Less inter-node communication
- More horizontal scalability (pleasingly parallel)
**CfnCluster** is a tool used to build and manage High Performance Computing (HPC) clusters on AWS.
What is MPI?

Message Passing Interface

—“De facto” standard
—Not an “official” standard (IEEE, IETF, ...)

Written and ratified by the MPI Forum

—Body of academic, research, and industry representatives

MPI is two spec documents:

—MPI-1 and MPI-2
—Specified interfaces in C, C++, Fortran 77/90

Software implementations of spec
—Mostly host-side software
What is MPI?

“Middleware”
- Sits between the application and network
- Simplifies network activity to the application

Source code portability
- Run apps on commodity clusters and “big iron” supercomputers

Scientists and engineers
- Don’t know or care how network works
- Not computer scientists
- Sometimes not even [very good] programmers

Parallel computing
- Using tens, hundreds, or thousands of servers in a single computational program
- Intended for high-performance computing
<table>
<thead>
<tr>
<th>Request</th>
<th>Bid Price</th>
<th>Spot Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$1.00</td>
<td>$0.20</td>
</tr>
<tr>
<td>2</td>
<td>$0.55</td>
<td>$0.20</td>
</tr>
<tr>
<td>3</td>
<td>$0.50</td>
<td>$0.20</td>
</tr>
<tr>
<td>4</td>
<td>$0.33</td>
<td>$0.20</td>
</tr>
<tr>
<td>5</td>
<td>$0.20</td>
<td>$0.20</td>
</tr>
<tr>
<td>6</td>
<td>$0.18</td>
<td>$0.20</td>
</tr>
<tr>
<td>7</td>
<td>$0.15</td>
<td>$0.20</td>
</tr>
<tr>
<td>8</td>
<td>$0.10</td>
<td>$0.20</td>
</tr>
<tr>
<td>9</td>
<td>$0.05</td>
<td>$0.20</td>
</tr>
</tbody>
</table>
Computing at the Fermi National Accelerator Laboratory

Lead United States particle physics laboratory

- Funded by the Department of Energy
- ~100 PB of data on tape
- High Throughput Computing characterized by:
  - “Pleasingly parallel” tasks
  - High CPU instruction / Bytes IO ratio
  - But still lots of I/O. See Pfister: “In Search of Clusters”

Focus on Neutrino Physics

- Including the NOvA Experiment

Strong collaborations with international laboratories

- CERN / Large Hardron Collider (LHC) Experiments
- Brookhaven National Laboratory (BNL)
- Lead institution (“Tier-1”) for the Compact Muon Solenoid (CMS)
Data centric distributed processing in the Cloud
a.k.a.
The “Business World”
Kafka

- Very high performance
- Elastically scalable
- Low operational overhead
- Durable, highly available
- Rewindable consumption

- Focus on HDFS

(The Hadoop Distributed File System is a distributed file system designed to run on commodity hardware)

- Push model
- No rewindable consumption

**Message queues**
- ActiveMQ
- TIBCO

**Log aggregators**
- Flume
- Scribe

- Low throughput
- Secondary indexes
- Tuned for low latency

APACHE kafka™
A distributed streaming platform
What is pub sub?

- **Producer**
  - publish(topic, msg)

- **Topic 1**
- **Topic 2**
- **Topic 3**

- **Consumer**
  - subscribe
  - msg

Publish subscribe system
Volume of data in LinkedIn (activity tracking & Logging)

- 20B events/day
- 3 terabytes/day
- 150K events/sec
Efficiency: simple storage

- Each topic has an ever-growing log
- A log == a list of files
- A message is addressed by a log offset

```
[nnarkhed@nnarkhed-md kafka-logs]$ tree -s
.
|-- [ 4096] novels-0
`-- [119878441] 00000000000000000000.kafka
`-- [ 4096] short stories-0
   `-- [119817774] 00000000000000000000.kafka

2 directories, 2 files
```
Efficiency #2 & Safety

Efficient operations

- Batch send and receive
- No message caching in JVM
- Rely on file system buffering
- Zero-copy transfer: file -> socket

Multi subscribers

- 1 file system operation per request
- Consumption is cheap
- SLA based message retention
- Rewindable consumption

Guarantees

- Data integrity checks
- At least once delivery
- In order delivery, per partition
Automatic load balancing

Producer

Broker

Consumer

Producer

Broker

Consumer
Performance - AWS

- 2 AWS EC2 instances
  - 16 2.0 GHz cores
  - 6 7200 rpm SATA drive RAID 10
  - 24GB memory
  - 1Gb network link
- 200 byte messages
- Producer batch size 200 messages

- Producer batch size = 40K
- Consumer batch size = 1MB
- 100 topics, broker flush interval = 100K
- Producer throughput = 90 MB/sec
- Consumer throughput = 60 MB/sec
- Consumer latency = 220 ms
Latency vs throughput

(100 topics, 1 producer, 1 broker)
Scalability

(10 topics, broker flush interval 100K)

Throughput in MB/s

101 190 293 381

1 broker 2 brokers 3 brokers 4 brokers