Big Data and DevOps in the Cloud: the role of monitoring and DICE approach

Dana PETCU

web.info.uvt.ro/~petcu, Dana.Petcu@e-uvt.ro
Outline

- Motivation
- DICE approach:
  - starting point, main ingredients and scope
  - workflow and architecture
  - a simple example of modeling and analysis
- DICE monitoring platform
- Concept implementation status

Subscribe to WG4: Obj. „improving the programmability of data management and analysis“
The Rapid Growth of Big Data

- Software market rapidly shifting to Big data
  - 27% compound annual growth rate through 2017 (IDC)
  - Popular technologies such as Spark, Hadoop, and NoSQL boost Big Data adoption and revenues from new services

Business issue: 65% of Big data projects still fail (CapGemini’15)
What problems SMEs face?
An example from DICE consortium

Traditional market: Legacy software systems

- Learning curves
- Initial prototype
- Risk of failure
- Fast-paced market

Customers with legacy data now ask for Big Data technologies

©DICE 7/13/2016
What Are Your Primary Concerns About Using Big Data Software?

- Big data expertise is scarce and expensive: 38%
- Data warehouse appliance platforms are expensive: 33%
- We aren’t sure how big data analytics will create business opportunities: 31%
- Analytical tools are lacking for big data platforms like Hadoop and NoSQL databases: 22%
- Our data’s not accurate: 21%
- Hadoop and NoSQL technologies are hard to learn: 17%
- We don’t have enough data: 13%
- Hadoop and other NoSQL technologies lack management features: 12%
- Other: 2%
- I have no concerns about big data analytics: 16%

Note: Multiple responses allowed
Skill shortage is among the main causes of Big data project failures [Gartner]

- Software written by inexperienced developers does not fulfil reliability, safety, efficiency requirements

Integrating Quality Assurance (QA) practices in application development can reduce failures

- Important for SMEs, often no dedicated quality teams
- QA is an endemic problem in EU software research
  - e.g., ISTAG Report calls to define environments: “for understanding the consequences of different implementation alternatives (quality, robustness, performance, maintenance, evolvability, ...)”
Mission: support in developing high-quality cloud-based data-intensive applications (DIAs)

- H2020 ICT 9 Call/2014 – Software engineering
- 9 partners (Academia & SMEs), 7 EU countries

Imperial College London (IMP)
IEAT (IEAT)
Universidad Zaragoza (ZAR)
Politecnico Di Milano (PMI)
XLAB (XLAB)
NetFective (NETF)
ATC (ATC)
Flexiant (FLEXI)
ProDevelop (PRO)
How to support DIA development?

Characterize Data Properties
- Volume
- Variety
- Location
- Velocity
- Privacy

Big Data Technologies
- NoSQL
- Spark
- Hadoop
- Storm
- Cloud

Development Methods & Tools
- UML
- Integration
- Delivery
- QA
- DevOps

©DICE 7/13/2016
Building blocks for DIAs today

- Distributed storage
  - Cassandra
  - HDFS
- Lambda architecture
- Cloud infrastructure
- Data streaming
- Distributed computation

Diagram:
- Data Source
- Coordinator (Kafka)
- Serving Layer
- Speed Layer
- Batch Layer
- Orchestrator (Hadoop Cluster)
- Data Store
Open questions

- How many DIA technologies do I need to know and combine?
- How can I reduce incidents, increase robustness, improve performance?
- Which resources, how many do I need for my applications?
- How do we configure the deployed technologies?
- What caused the performance decay since my last commit?
- What performance constraints (SLAs) are violated at runtime?

DICE objectives

- Simplify the life of designers and reduce effort/cost
- Provide assessment for quality properties
- Simplify deployment
- Ensure that lessons learned in Ops are considered in design activities
DICE delivers the **first** quality-driven framework for DIAs:

- UML profile for DIAs
- UML-driven agile delivery via DICE IDE
- Ecosystem of quality prediction and testing tools for Big data technologies
- Iterative cycles of refinements through analysis of test data
1. **Requirements for data** (volume, velocity, ...)  
2. **Requirements for data technologies**  
3. **Agile delivery** of DIAs via IDE & DevOps methodology  
4. Model-based **quality analysis and verification**  
5. Rapid & optimized **deployment** via TOSCA  
6. **Semi-automated feedback analysis** of monitoring and test data to improve design
Demonstrators

- ATC
- NETFECTIVE
- proDEVELOP

- News Asset
- Tax Fraud Detection Application
- Posidonia Operations

- News&Media Market
- e-Goverment Market
- Maritime Sector

©DICE 7/13/2016 DICE RIA - Overview
What do we mean by Quality?

- Reliability
  - Availability
  - Fault-tolerance

- Efficiency
  - Performance
  - Costs

- Correctness
  - Safety
  - Temporal metrics
DevOps is “A set of practices and tools to reduce the time to commit a change to production, while ensuring high-quality.” (Bass et al.,’15)
Model driven engineering in a nutshell

A single model...

... transformations...

... many targets

Other models

Analysis tools

TOSCA blueprint

Code or scripts

©DICE
MDE and DevOps: a possible synergy
DICE scope

• Continuous architecting
• Continuous delivery
• Continuous testing and monitoring in canary environment
• Continuous feedback to development
DICE incremental modeling and analysis

DICE Platform Independent Model (DPIM)

M2M transformation

DICE Technology Specific Model (DTSM)

M2M transformation

DICE Deployment Specific Model (DDSM)

M2T transformation

TOSCA blueprint

Analysis

Analysis

Analysis & Optimization
DICE deployment, monitoring and testing

- DICE deployment
- Monitoring
- Testing
- Enhancement
- Fault Injection
- Quality Testing
- Anomaly Detection
- Trace Checking
- Configuration optimization
- Deployment
- TOSCA blueprint
- Running DIA Comp (MW, VM)
- Running DIA Comp (MW, VM)
A toy example of incremental modeling and analysis: word count example

<<Source Node>> --> <<Computation Node>>

Word Count (Batch, Machine Learning)

<<Storage Node>>

<<DICE::ResponseTime
30 sec on average>>

DPIM
A toy example of incremental modeling and analysis: word count example

- **Source Node**
- **Storage Node**
- **Computation Node**
  - Word Count (Batch, Machine Learning)
  - **ResponseTime** 30 sec on average
- **HDFS**
- **WC mapper**
- **WC reducer**
- **Job Configuration**
- **Framework Configuration**
- **HadoopMR**
- **nTask=3...**

©DICE 23
A toy example of incremental modeling and analysis: word count example

Configuration defined by analysis & optimization, improved from runtime data by configuration optimization
Components role in DICE architecture

IDE based on Eclipse

Profile
Verification
Simulation
Optimization

Repository & CI
Configuration Optimization
Delivery
Monitoring

Anomaly Detection
Trace Checking
Enhancement
Quality Testing
Fault Injection (Resilience)

Running DIA Comp
MW
VM

Running DIA Comp
MW
VM
DICE arch current implementation

IDE based on Eclipse

Profile

Verification

Simulation

Optimization

Repository & CI

Anomaly Detection

Trace Checking

Enhancement

Quality Testing

Monitoring

Configuration Optimization

Delivery

Fault Injection (Resilience)

Running DIA Comp

Running DIA Comp

Running DIA Comp

D2.1

D3.2

D3.5

D4.1

D5.1

D5.1

WP1

WP5

©DICE

DICE – Approach and architecture
DICE Monitoring Platform

**The problem:**
- DIA generate Big Monitoring Data
- Heterogeneous data generated and collected

**What does the tool do?**
- Collects monitoring data generated during DIA execution
- Visualization of collected data

**Innovation:**
- Integrates in a unique platform monitoring data from multiple Big Data technologies
- Easy to deploy, control and scale
- Default selection of metrics across DICE tool chain
- Easy integration with existing system monitoring solutions (e.g. Nagios, Ganglia)

**Impact & stakeholders:**
- Reduce cost and time of deploying and maintaining a monitoring solution
- Big Data operators, start-ups on Big Data technologies
## Why Another Monitoring Tool?

<table>
<thead>
<tr>
<th></th>
<th>Nagios</th>
<th>Ganglia</th>
<th>Sequence IQ</th>
<th>Apache Chukwa</th>
<th>Sematext</th>
<th>DataDog</th>
<th>D-Mon</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elasticity</td>
<td>None</td>
<td>None</td>
<td>-</td>
<td>None</td>
<td>Yes ?</td>
<td>-</td>
<td>Yes (Y2)</td>
</tr>
<tr>
<td>Deployment model</td>
<td>VM</td>
<td>On-premise</td>
<td>As a service</td>
<td>On-premise</td>
<td>As a Service / On-premise</td>
<td>As a service</td>
<td>As a service / On-premise</td>
</tr>
<tr>
<td>Installation</td>
<td>-</td>
<td>Manual / via CMS</td>
<td>-</td>
<td>Manual / via CMS</td>
<td>-</td>
<td>-</td>
<td>as services using REST API</td>
</tr>
<tr>
<td>Big Data frameworks support</td>
<td>Poor</td>
<td>Poor</td>
<td>Hadoop 2.x</td>
<td>Hadoop 2.x</td>
<td>Good and Extensible</td>
<td>Good (no Spark, Storm)</td>
<td>Good and Extensible</td>
</tr>
<tr>
<td>Visualization</td>
<td>User defined</td>
<td>Predefined</td>
<td>Predefined ?</td>
<td>Predefined</td>
<td>User-defined</td>
<td>User-defined</td>
<td>User-defined</td>
</tr>
<tr>
<td>Analytics</td>
<td>Alerts</td>
<td>???</td>
<td>ML support</td>
<td>Anomaly detection</td>
<td>Alerts</td>
<td>Alerts, correlation s</td>
<td>Anomaly detection (Y2)</td>
</tr>
<tr>
<td>Real-time data support</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Licensing</td>
<td>Freemium</td>
<td>BSD</td>
<td>Commercial</td>
<td>Apache 2</td>
<td>Freemium</td>
<td>Freemium</td>
<td>Apache 2</td>
</tr>
</tbody>
</table>
DICE Monitoring Platform (D-Mon) Highlights

- (near) Real-time data
- Fully distributed
- Horizontal scaling, elasticity
- High availability
- Small footprint on nodes
- Easy deployment (integration with DICE Deployment Service)
- Offers a selection of representative performance and reliability metrics across DICE-supported technologies
- D-Mon controller service (HTTP REST API) (developed in Y1)
- Query results as JSON, CSV, raw text, OSLC Perf Mon (selected)
- Customizable visualization
Monitoring Platform Architecture

Micro-services architecture

First release due at M18
OSLC Perf Mon

- Open Services for Lifecycle Collaboration Performance Monitoring Specification Version 2.0
- Semantic vocabulary
- RDF/XML format

```xml
<rdf:Description rdf:about="http://dice-h2020.eu/rec001#cpuutil10">
  <ems:numericValue rdf:datatype="http://www.w3.org/2001/XMLSchema#integer">10</ems:numericValue>
  <ems:unitOfMeasure rdf:resource="http://dbpedia.org/resource/Percentage"/>
  <ems:metric rdf:resource="http://open-services.net/ns/perfmon#CpuUsed"/>
  <dcterms:title>CPU Utilization</dcterms:title>
  <rdf:type rdf:resource="http://open-services.net/ns/ems#Measure"/>
</rdf:Description>
```

<table>
<thead>
<tr>
<th>Metric</th>
<th>RDF/namespace</th>
<th>Source</th>
<th>Measurement Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU Speed</td>
<td>pm:CpuSpeed</td>
<td>CPUfreq from collectd</td>
<td>Idle, nice, User, Wait-IO, System, SoftIRQ, IRQ, Steal</td>
</tr>
<tr>
<td>CPU Utilization</td>
<td>pm:CpuUsed</td>
<td>cpu plugin for collectd</td>
<td>free, used</td>
</tr>
<tr>
<td>Percentage Disk Space Used</td>
<td>pm:DiskSpaceUsed</td>
<td>df plugin for collectd</td>
<td>MemHeapUsedM, MemHeapCommittedM, MemHeapMaxM</td>
</tr>
<tr>
<td>Heap Usage</td>
<td>pm:HeapMemoryUsed</td>
<td>JVM Context</td>
<td></td>
</tr>
<tr>
<td>Real Memory Utilization</td>
<td>pm:RealMemoryUsed</td>
<td>memory plugin for collectd</td>
<td>used, buffered, cached, free</td>
</tr>
</tbody>
</table>
D-Mon Deployment

- Deployments on
  - Flexiant Cloud Orchestrator (FCO)
  - Open Stack (cloud platform at West University of Timisoara)
  - ATC’s premises
- 2 testbed deployments on FCO and Open Stack
  - 4-node cluster with Cloudera Distribution for Hadoop (CDH 5.4.7): YARN, HDFS, Spark + other services
  - 4-node cluster with Apache Storm

- Technical details
  - Python (~80%) + Bash + Ruby
  - +22,000 Lines of Code
  - ~100 classes
  - +30 external libraries
# DICE Timeline

<table>
<thead>
<tr>
<th>Year</th>
<th>Deliverables</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan 2016</td>
<td>• DICE simulation tools</td>
</tr>
<tr>
<td></td>
<td>• DICE verification tools</td>
</tr>
<tr>
<td></td>
<td>• Monitoring and data warehousing</td>
</tr>
<tr>
<td></td>
<td>• DICE delivery tools</td>
</tr>
<tr>
<td>Jul 2016</td>
<td>• Release of the main DICE Tools as independent tools</td>
</tr>
<tr>
<td>Jan 2017</td>
<td>• First release of the integrated framework</td>
</tr>
<tr>
<td>Jul 2017</td>
<td>• Second release of the integrated framework</td>
</tr>
<tr>
<td>Jan 2018</td>
<td>• Completion of demonstrators and consolidation of results</td>
</tr>
</tbody>
</table>
# Research & Innovation KPIs

<table>
<thead>
<tr>
<th>RESEARCH &amp; INNOVATION KPI</th>
<th>ACHIEVEMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Release of DICE framework under an open source non-viral license</td>
<td>All tools are due for non-viral open source release, some with partial release.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Technical Asset</th>
<th>GitHub Code Release</th>
</tr>
</thead>
<tbody>
<tr>
<td>DICE Deployment Service</td>
<td><a href="https://github.com/dice-project/DICE-Deployment-Service">https://github.com/dice-project/DICE-Deployment-Service</a></td>
</tr>
<tr>
<td>FCO plug-in for Cloudify</td>
<td><a href="https://github.com/dice-project/DICE-FCO-Plugin-Cloudify">https://github.com/dice-project/DICE-FCO-Plugin-Cloudify</a></td>
</tr>
<tr>
<td>DICE TOSCA library</td>
<td><a href="https://github.com/dice-project/DICE-Deployment-Cloudify">https://github.com/dice-project/DICE-Deployment-Cloudify</a></td>
</tr>
<tr>
<td>DICE Chef Cookbooks</td>
<td><a href="https://github.com/dice-project/DICE-Chef-Repository">https://github.com/dice-project/DICE-Chef-Repository</a></td>
</tr>
<tr>
<td>DICE Jenkins plug-in</td>
<td><a href="https://github.com/dice-project/DICE-Jenkins-Plugin">https://github.com/dice-project/DICE-Jenkins-Plugin</a></td>
</tr>
<tr>
<td>DICE Configuration Optimization – BO4CO Module</td>
<td><a href="https://github.com/dice-project/DICE-Configuration-BO4CO">https://github.com/dice-project/DICE-Configuration-BO4CO</a></td>
</tr>
<tr>
<td>DICE Verification tool</td>
<td><a href="https://github.com/dice-project/DICE-Verification">https://github.com/dice-project/DICE-Verification</a></td>
</tr>
<tr>
<td>DICE Simulation tool</td>
<td><a href="https://github.com/dice-project/DICE-Simulation">https://github.com/dice-project/DICE-Simulation</a></td>
</tr>
<tr>
<td>DICE Profiles</td>
<td><a href="https://github.com/dice-project/DICE-Profiles">https://github.com/dice-project/DICE-Profiles</a></td>
</tr>
<tr>
<td>DICE Fault Injection tool</td>
<td><a href="https://github.com/dice-project/DICE-Fault-Injection-Tool">https://github.com/dice-project/DICE-Fault-Injection-Tool</a></td>
</tr>
<tr>
<td>DICE D-Mon monitoring platform</td>
<td><a href="https://github.com/dice-project/DICE-Monitoring">https://github.com/dice-project/DICE-Monitoring</a></td>
</tr>
<tr>
<td>DICE Metamodels</td>
<td><a href="https://github.com/dice-project/DICE-Models">https://github.com/dice-project/DICE-Models</a></td>
</tr>
</tbody>
</table>
Questions?

www.dice-h2020.eu
Others

- **HPC for Science and Technologies, workshop**
  - September 24-27, Timisoara
  - Extended deadline: 25 July
  - Publication by IEEE CSP

- **SESAME-NET, HPC for SME**
  - Call for joining the network, for small&medium HPC centers
  - [http://sesamenet.eu](http://sesamenet.eu)

- **NATRES Cluster: recommendations for WP18-19**
  - [https://eucloudclusters.wordpress.com/new-approaches-for-infrastructure-services/](https://eucloudclusters.wordpress.com/new-approaches-for-infrastructure-services/)