DICE: Quality-Aware DevOps For Big Data Applications

Giuliano Casale
*Imperial College London*
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)
What problems EU SMEs face?
An example from our 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

(+ others...)

Growth in sight, but ...
Mission: support SMEs in developing high-quality cloud-based data-intensive applications (DIAs)

- Horizon 2020 research project (4M€, 2015-18)
- 9 partners (Academia & SMEs), 7 EU countries
DICE: Quality-Aware DevOps for Big Data

Design → Prototype → Enhance → Deploy → Monitor

- Hadoop
- Cassandra
- Spark
- Storm
How to support DIA development?

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

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

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

©DICE 01/16/2018
What do we mean by Quality?

- Reliability
  - Availability
  - Fault-tolerance
- Efficiency
  - Performance
  - Costs
- Correctness
  - Privacy & security
  - Temporal metrics
DICE Framework
DICE Workflow - Dev

DICE Eclipse IDE

“I want to design for Big data”

Design

Transform to Formal Models

Simulate & Verify

Optimize

“Will the DIA meet SLAs and costs?”

©DICE 01/16/2018
"I want to quickly deploy and tune the DIA"

"I want an integrated DevOps toolchain"

Deploy

Quality Testing

Fault injection

Configuration Optimization

Enhance

DICER (TOSCA)

DDSMP

Optimal config

©DICE 01/16/2018
Following the DevOps paradigm, DICE delivers a unified toolchain for the enterprise team.
DICE Approach, Tools, Architecture and Methodology
Ingredients of technical approach

- DevOps

- Model-Driven Engineering

DevOps

Model-Driven Engineering

Analysis

Deployment blueprint
DICE incremental modeling and analysis

DICE Platform Independent Model (DPIM)

is implemented by

DICE Technology Specific Model (DTSM)

is deployed onto

DICE Deployment Specific Model (DDSM)

M2T transformation

TOSCA blueprint

Analysis & Optimization

Analysis

Analysis
DICE deployment, monitoring and testing

DICE Methodology

©DICE 01/16/2018
I’m defining the DPIM model for my application
DPIM-level simulation

A two VMs configuration will not be acceptable for this application
The verification step

I want to check if critical bolts could create problems at runtime.
Max>> Everything seems to be ok. Let’s focus on optimization...

Let’s identify the optimal configuration for Amazon reserved instances
Deployment step– creating the TOSCA YAML

I have got all I need to create the blueprint!
Deployment step – sending the TOSCA YAML to the deployment service

... and now let’s deploy!
Monitoring and detecting anomalies

Everything is fine today!
Quality testing

Features:
- Load generation for stream processing systems
- Load replay
- Load scaling (via Hidden Markov Model representation)

QT Tool workflow:
1. Max & Ophra add QT tool as a spout, provide (initial) load & deploys DIA
2. QT tool starts simultaneously with DIA
3. Injects load
4. Test results in DMon

Load injection into the DIA

Let’s stress our DIA!

©DICE 01/16/2018
Fault injection

And now let's check robustness!
Enhancement tool

- **Objective:** designed for iteratively enhancing the DIA quality

- **Functions:**
  - Providing a performance and reliability analysis
  - Updating UML models with analysis results
  - Anti-patterns Detection
  - Refactoring the design model

let's consider the proposed improvements!
Following the DevOps paradigm, DICE delivers a unified toolchain for the enterprise team.
Demonstrators

- News Asset
- Tax Fraud Detection Application
- Posidonia Operations

- News&Media Market
- e-Government Market
- Maritime Sector
News analysis demonstrator

Storm cluster
Configuring a Big data system

```java
102  drpc.port: 3772
103  drpc.worker.threads: 64
104  drpc.max_buffer_size: 1048576
105  drpc.queue.size: 128
106  drpc.invocations.port: 3773
107  drpc.invocations.threads: 64
108  drpc.request.timeout.secs: 600
109  drpc.childopts: "-Xmx768m"
110  drpc.http.port: 3774
111  drpc.https.port: -1
112  drpc.https.keystore.password: ""
113  drpc.https.keystore.type: "JKS"
114  drpc.httpcreds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugin
115  drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
116  drpc.authorizer.acl.strict: false
117  transactional.zookeeper.root: "/transactional"
118  transactional.zookeeper.servers: null
119  transactional.zookeeper.port: null
120
121  ## blobstore configs
122  supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
123  supervisor.blobstore.download.thread.count: 5
124  supervisor.blobstore.download.max_retries: 3
125  supervisor.localizer.cache.target.size.mb: 10240
126  supervisor.localizer.cleanup.interval.ms: 600000
```
DICE configuration optimization
DICE configuration optimization

Configuration Optimizer → Target Configuration

configuration parameters domains

experiment time

performance data repository

topology deployment

DICE      01/16/2018
Applying Bayesian Optimization to DevOps

- Default configuration
- Configuration recommended by expert
- CO after 100 iterations
Thanks!

DEVOPS FOR BIG DATA

WWW.DICE-H2020.EU