Toward A Model-Driven Design Tool for Big Data Architectures

Michele Guerriero
Saeed Tajfar
Damian Andrew Tamburri
Elisabetta Di Nitto
Introduction

Lambda architecture

Cloud infrastructure

Distributed storage

Data streaming

Distributed computation

Data Source → Coordinator (Kafka) → Serving Layer → Speed Layer → Batch Layer → Data Store

Cloud infrastructure:
- HDFS
- Cassandra

Data streaming tools:
- Kafka
- Spark
- Storm

Cloud infrastructure tools:
- AWS
- OpenStack
Questions

How many Big Data technologies do I need to know and combine?

Which resources, how many do I need and how do we configure the deployed technologies?

What if I want to know properties and performance of my application?

Goals

Simplify software design and reduce costs

Simplify Deployment

Support Analysis
Model-Driven Engineering

Analysis

Deployment blueprint
Model-Driven Big Data Design Architecture

Platform Independent Model (DPIM)

M2M transformation

is implemented by

Technology Specific Model (DTSM)

M2M transformation

is deployed onto

Deployment Specific Model (DDSM)

M2T transformation

TOSCA blueprint

Analysis

Analysis

Analysis & Optimization
DTSM Meta-Model

DPIM Model

M2M Transformation

Core DTSM Package

<<Uses>>

Storm Package

<<Uses>>

HadoopMR Package

<<Uses>>

Spark Package

<<Uses>>

Oryx 2 Package

<<Uses>>

Extensibility

...
The HadoopMR DTSM Package

Application Logic

Reducer
- type: ReducerType
  - [0..*] hasReducer

Mapper
- type: MapperType
  - [0..*] hasMapper

InputFormat
- type: InputFormatType
  - path: String
  - numberOfSplits: Int

OutputFormat
- type: OutputFormatType
  - path: String

Job
- jobId: String
- isJobCompleter: EBoolean
- numOfReduceTasks: Int
- numOfMapTasks: Int
- fileSystemConfig: Config

Framework Configuration

Scheduler
- numberOfHosts: int
  - type: SchedulerType

Partitioner
- type: PartitionerType

Combiner
- type: CombinerType

HadoopMRApplication
- appId: String

[1..1] hasScheduler
hasPartitioner [1..1]
[1..1] hasCombiner
• Build Deployment Topology
• Standards adoption (TOSCA)
• Deployment Technological Packages

DDSM Meta-Model
Transformations Set
Conclusion and Future Works

- Preliminary steps toward the **model-driven engineering of Big Data applications**.

- **Great potential** behind MDE for data-intensive applications!

- Future steps:
  - increase **models expressiveness** (data quality, privacy concerns)
  - **validation** against industrial case studies
  - increase **automation** mechanisms
  - **technological support**
Thanks!