FLINK: MODEL-DRIVEN DEMPOLYMENT

Introduction

New models of computing have been growing up rapidly and are become widely popular with the amount of data growing at an exponential rate.

For the computation of these data in the present era, we need high speed processing techniques, which in turn leads to a curtail the process of software development cycle.

Background

**DICER** is a tool developed with the goal of supporting the deployment and management of Big Data applications. Main goal of DICER is to exploit deployment models specified in accordance with the DICE Deployment Specific metamodel, in order to speed up the deployment process.

DICER is a tool developed in the context of the DICE H2020 European Project enabling the model-driven deployment of data-intensive applications (DIAs) leveraging the Infrastructure-as-Code (IaC) paradigm. More specifically, DICER adopt the OASIS Topology and Orchestration Specification for Cloud Applications standard and is able to automatically generate IaC for DIAs in the form of TOSCA blueprints from stereotypes UML models.

Using Eclipse UML Papyrus and the DICER Eclipse plugins, we are able to design the UML Deployment Diagram of its DIA using the DICER **UML profile** to enrich the model with DIA-specific deployment concepts by applying standard UML stereotypes. By using the UML stereotypes, we can configure various properties of our model, both at the infrastructural level (i.e. on some number of VMs) and also at the platform level.

How to extend profile ????????????????????????

In this project we are trying to deploy **Flink** which is

an open-source framework for distributed stream processing. It allows exactly-once semantics for stateful computations along with that it supports stream processing and windowing with event time

semantics, supports flexible windowing based on time, count, or sessions in addition to data-driven windows. It’s lightweight fault tolerance structure allows the system to maintain high throughput rates and provide exactly-once consistency guarantees at the same time. Even processing of large volume can be done at faster way using Flink because of it high throughput and low latency. Flink can run in the cloud or on premise and on a standalone cluster or on a cluster managed by YARN or Mesos.

To deploy the models that has been generated in the DDSM into the TOSCA blueprint we need to generate TOSCA code. To generate this codes, we can use the **ATL Transformation language**. ATL-Transformation Language is a model transformation language and toolkit. In the field of Model-Driven Engineering (MDE), ATL provides ways to produce a set of target models from a set of source models. Developed on top of the Eclipse platform, the ATL Integrated Environment (IDE) provides a number of standard development tools (syntax highlighting, debugger, etc.) that aims to ease development of ATL transformations. It provides a way to produce a number of target models from a set of source models. An ATL transformation program is composed of rules that define how source model elements are matched and navigated to create and initialize the elements of the target models.

Papyrus