WP3: the properties that are likely to be verified are definitely latency and throughput. Any property which is consistent with these two is then an...

<<enumeration>>
StormOpMode
- local

Storm is a distributed computation framework written predominantly in the Clojure programming language. Originally created by Nathan Marz[1] and team at BackType,[2] the project was open sourced after being acquired by Twitter.[3] It uses custom created "spouts" and "bolts" to define information sources and manipulations to allow batch, distributed processing of streaming data. The initial release was on 17 September 2011.[4]

A Storm application is designed as a topology in the shape of a directed acyclic graph (DAG) with spouts and bolts acting as the graph vertices. Edges on the graph are named streams, and direct data from one node to another. Together, the topology acts as a data transformation pipeline. At a superficial level the general topology structure is similar to a MapReduce job, with the main difference being that data is processed in real-time as opposed to in individual batches. Additionally, Storm topologies run indefinitely until killed, while a MapReduce job DAG must eventually end.[5]

Storm became an Apache Top-Level Project in September 2014,[6] and was previously an Apache Incubator project since Septem...

TopologyBuilder exposes the Java API for specifying a topology for Storm to execute. Topologies are Thrift structures in the end, but since the Thrift API is so verbose, TopologyBuilder greatly eases the process of creating topologies.

The topology is a virtual DAG composed of spouts and bolts. Altough virtual, Storm allows designers and developers to access the topology besides building it.

TopologyBuilder
name : EString
setSpout()
setBolt()

builds

Topology
build : EString
Reliable : EBoolean
name : EString
getTopology()
createTopology()
topologyBuilder()

logical specification

Component
name : EString
Emit()

reads from

DIASource

preprocesses

DIAFilter

these would be elements that we inherit from the upper layer specifications

Spout
inputSource : EString
open()
nextTopo()

programmers should limit themselves to implementing bolts and spouts also recalling other DIAelements after the FW has been set...

these bolt-spout relations are what models the...

DIAStorage

processed by

stores in

Bolt
inputMessage : EString
outputMessage : EString
Prepare()
Execute()

further processing

Every Bolt may be associated with a set of bolts by a typed association called StreamGrouping, for example,...

Grouping
FieldsGrouping
ShuffleGrouping
GlobalGrouping
AllGrouping
PartialKeyGrouping
DirectGrouping
LocalShuffleGrouping

ReplicationFactor
Number

The grouping is a device to replicate multiple bolts and spouts... The replicaton factor by grouping predicates that there is a replication associated to the typed association (i.e., the Grouping)

becomes

cross-functional processing

uses

DIAmain
opMode : StormOpMode

Component is the class of all elements that are part of a...

becomes

the DIAMain element is the substitute of whatever the original DPIM and DTSMCore instances contained before the package (storm in this...

TopologyConfigurator
parameter : EString
path : EString
maxSpout : EInt
ZookeeperConnectionTimeout : EDouble
getParameter()
setParameter()

NOTE: This is the Storm.Yaml file that gives the standard and default configuration parameters of Storm. Pooyan will send me a list of parameters to be extended here so that this class matches the expected parameters that are needed in the Simulator they are using for analysis.

cluster manager

depends on

clustered via

specifies

instance of

Because of consistency across tasks, every Executor can only be associated with tasks that are part of the same topology

Nimbus
stormFW : EString
UI : EString

Zookeeper

Supervisor
stormFW : EString

reference

file management

is managed by

is part of

these are the essential phisical elements that should then be mapped into the CloudML language and eventually in TOSCA-YAML. The principle here is that you will specify the necessary numbers in the configuration parameters for the topology configuration - this is then fed into Nimbus who will enact Zookeeper and Supervisor distribution across the cluster. The assumption is obviously that Storm is actually set-up already (e.g., thorugh Chef) as part of the YAML blueprint that will be produced feeding instances of this meta-model to the DICE Xtext grammar. As for other technological packages a distinction is made for the logical viewpoint elements (top part of this meta-model) and the deployment viewpoint elements.

feeds into

has

task consistency

Worker
ReplicationFactor : EInt

Executor
replicationFactor : EInt

Task
replicationFactor : EInt

when the bolts and spouts are aggregated for runtime operation they become Tasks, e.g., allocated to a VM