

RUTGERS

THE STATE UNIVERSITY
OF NEW JERSEY

Programming with CometCloud

Moustafa AbdelBaky, Javier Diaz-Montes, and Manish Parashar

NSF Cloud and Autonomic Computing Center (CAC)

Rutgers Discovery Informatics Institute (RDI²)

Rutgers, The State University of New Jersey

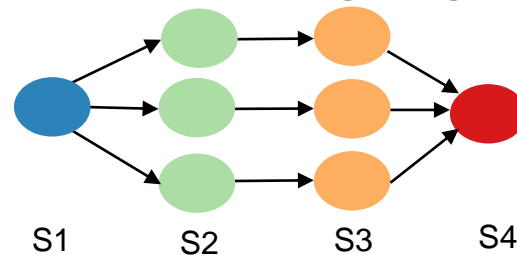


Introduction

- Learn how to use CometCloud to execute a workflow using autonomic in a federation
 - If application is deployed, we just simple create a workflow description
- Learn how to create a new application using the APIs
 - We need to create two components to Generate tasks and to Compute tasks (similar to master/worker models)
 - We also need to prepare the worker machine (e.g., VM image)
- Learn how to deploy CometCloud to create a federation
 - Download at <http://tinyurl.com/getcomet>
- Visit <http://tinyurl.com/cometbitbucket>

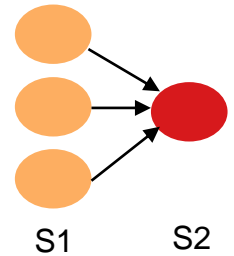
Defining a Workflow

- A workflow is composed by a set of stages
- A stage has a set of tasks with no dependencies (bag of tasks)
- Objective and constraints of a stage are enforced by selecting a scheduling policy (each stage can have a different one)
- Dependencies are defined across stages, DAG + loops
- A dependency can be blocking or non-blocking
 - Blocking – all tasks of a stage have to finish before the following stage starts
 - Non-Blocking – every time a task finishes we evaluate if there are tasks that can be executed in following stages



Simple Workflow Definition

- Define a simple two stages map/reduce workflow
- Four sections, namely stages definition, scheduling policies, dependencies, and loops (optional)



```

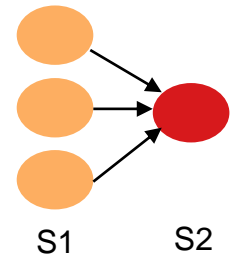
<xflow name="SimpleWorkflow">
<!-- Stages definition -->
<stages>
<!-- Stage 1 -->
<stage id="S1" type="AppGenerateClass" value="tassl.application.cometcloud.sample.GenerateTasks" method="map"/>
<stage id="S1" type="PropertyFile" value="./sample.properties"/>
<stage id="S1" type="Application" value="cbir"/>
<stage id="S1" type="InputData">
  <InputData value="jdiaz@sierra.futuregrid.org:/home/inputs/" zone="zoneA" site="siteSierra" constraint="zoneA,siteIndia,siteAlamo"/>
  <InputData value="jdiaz@hotel.futuregrid.org:/home/inputs/" zone="zoneA" site="siteHotel" constraint="zoneA,siteIndia"/>
  <InputData value="jdiaz@alamo.futuregrid.org:/home/inputs/" zone="zoneB" site="siteAlamo" constraint=""/>
</stage>
<stage id="S1" type="Results" value="" zone="" site="" constraint="zoneA,siteIndia,siteSierra,zoneB"/>
<!-- Stage 2 -->
<stage id="S2" type="AppGenerateClass" value="tassl.application.cometcloud.sample.GenerateTasks" method="reduce"/>
<stage id="S2" type="PropertyFile" value="./sample.properties"/>
<stage id="S2" type="Application" value="cbir"/>
<stage id="S2" type="InputData">
  <InputData value="jdiaz@alamo.futuregrid.org:/home/inputs/" zone="zoneB" site="siteAlamo" constraint=""/>
</stage>
<stage id="S2" type="Results" value="jdiaz@sierra.futuregrid.org:/home/output/" zone="zoneB" site="siteSierra" constraint="siteSierra"/>
</stages>

```

```

<!-- Scheduling Policies -->
<objectives>
  <objective id="S1" type="DeadlineLocalityAware" value="1377" />
  <objective id="S2" type="DeadlineLocalityAware" value="4776" />
</objectives>
<!-- Dependencies -->
<transitions>
  <transition from="S1" to="S2" blocking="true"/>
</transitions>
<!-- Loops -->
<loops>
  <loop from="S1" to="S2"/>
</loops>
</xflow>

```



Policy Name	Description
MinRunningTime	Minimum completion time
DeadlineLocalityAwareProc	Set of resources needed to complete all tasks within a given deadline while satisfying an objective function Objective function: performance machine
DeadlineLocalityAware	Objective function: trade-off between the highest performance and the lowest estimated transfer time.
DeadlineLocalityAwareCost	Objective function: cheapest available resource that can complete the task within the given deadline.
DeadlineLocalityAwareData	Objective function: estimated transfer time is minimum.
BudgetConstraint	Finds a solution that minimizes the total execution time (i.e., critical path) while keeping the cost of the solution within the given budget.

Command Line Interface

- workflowClient.sh
 - serverPort <port>
 - serverAddress <serverAddress>
 - (-regWorkflow <workflowFile> |
 - checkStatus <workflowId> |
 - cancelWorkflow <workflowId> |
 - getSupportedApps |
 - getResults <workflowId> -user <remoteUserId> -path <local path>)
- Parenthesis means that you need to specify one of the options

Developing Applications

- <http://tinyurl.com/developcomet>

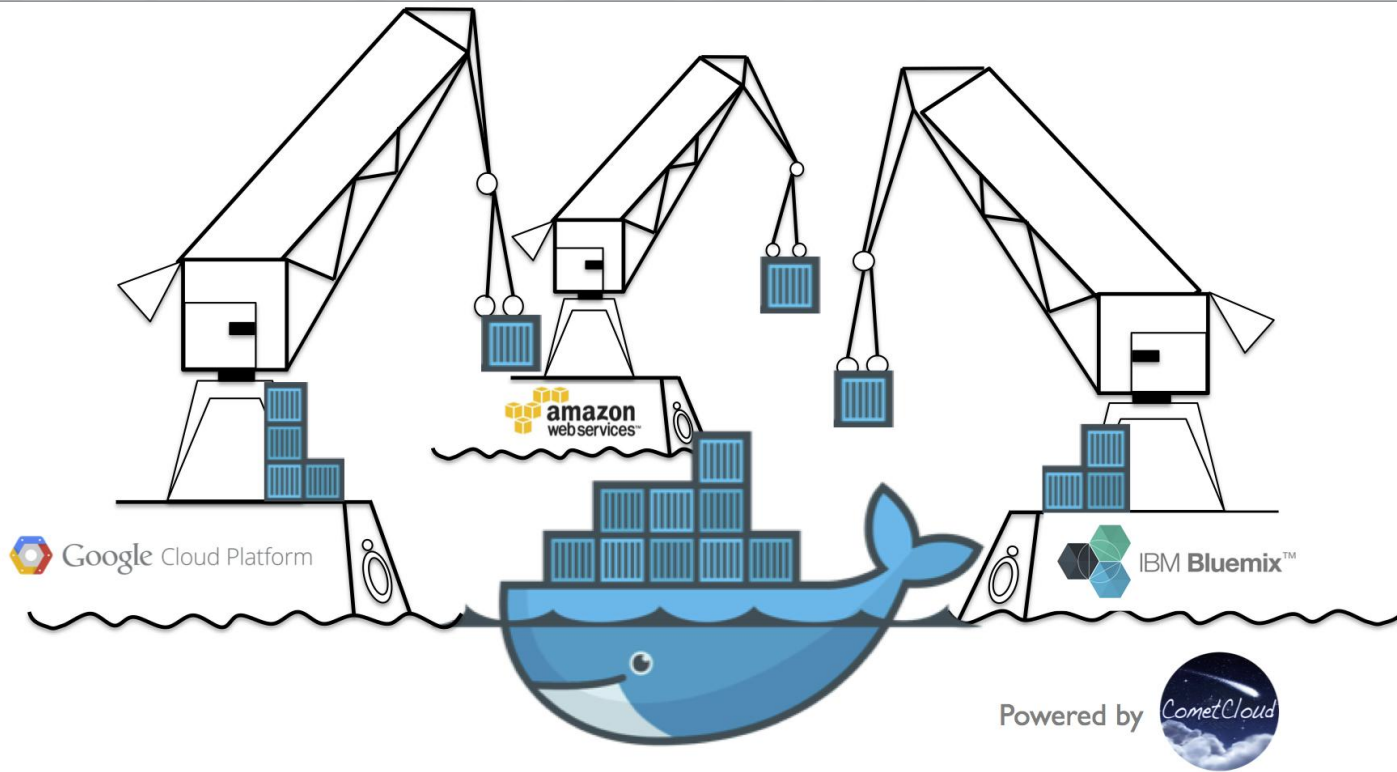
Quick Start Deployment

- <http://tinyurl.com/runcomet>

Supported Resources

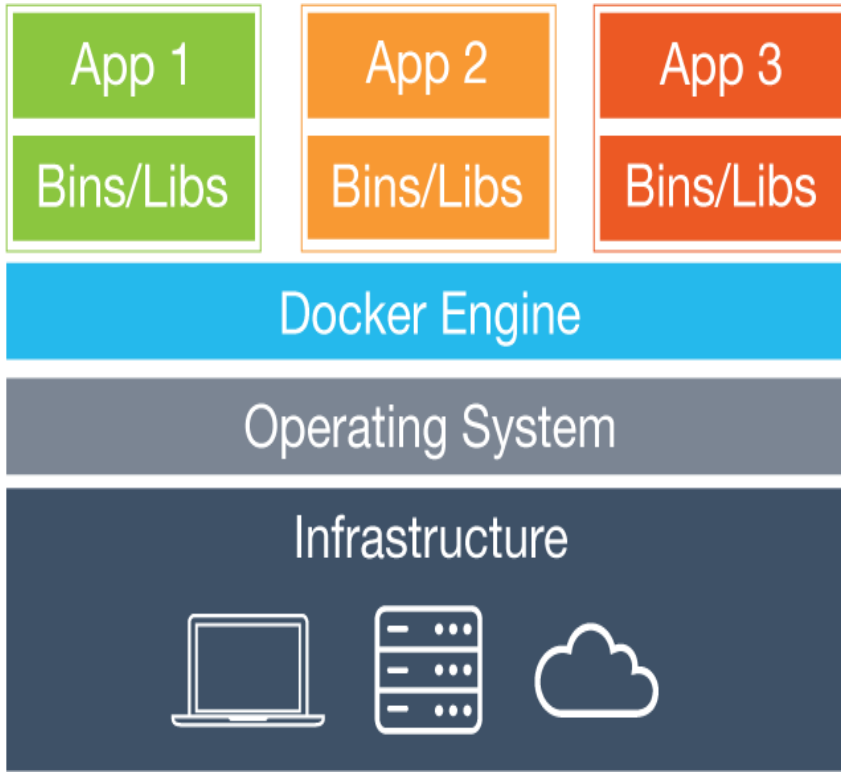
- Clusters
- Cloud Resources

ProviderType	Platform	API-used
openstack_ec2	OpenStack EC2	boto
openstack_nova	OpenStack Nova	novaclient
nimbus_ec2	Nimbus EC2	boto
aws_ec2	AWS EC2	boto



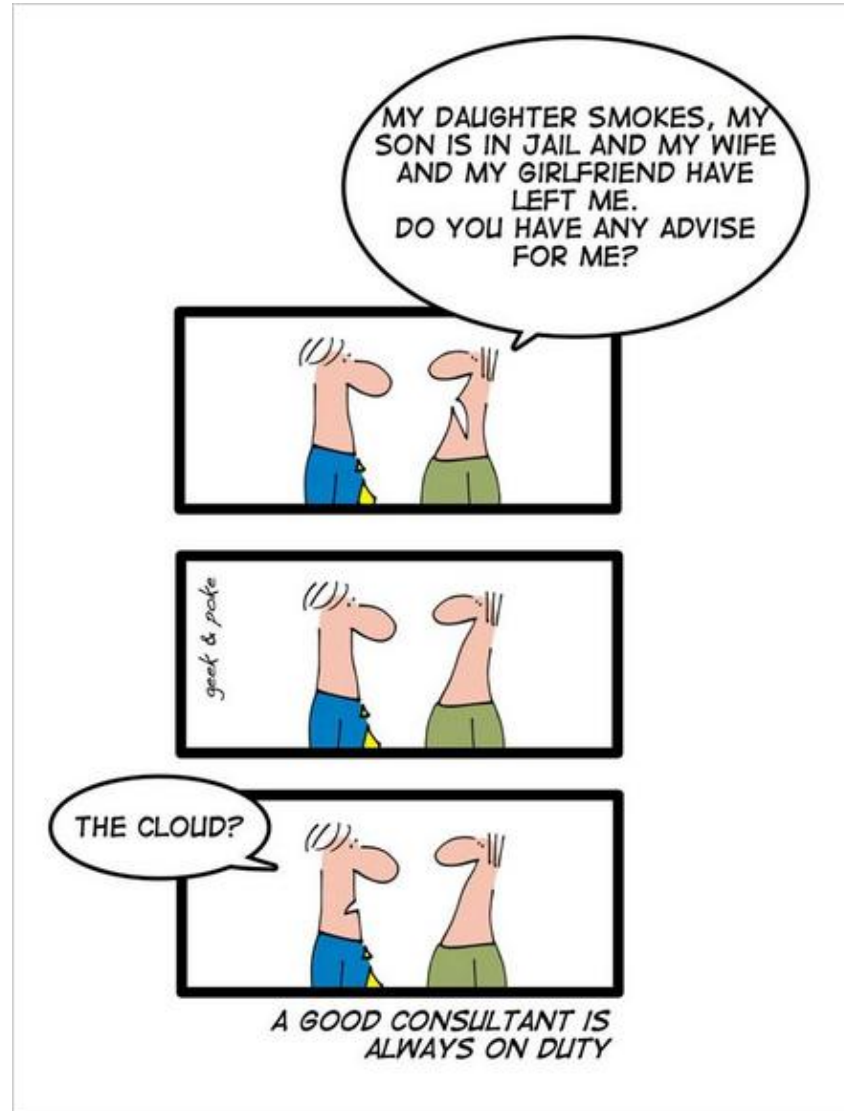
C-PORTS: DOCKER CONTAINERS ACROSS MULTIPLE CLOUDS AND DATACENTERS

Docker the new Cloud?



The “Container” Concept. Deploy everything you need to run your software in one container.

Image: Docker website <https://www.docker.com/whatisdocker>



Docker Across Clouds

- Container deployments are currently focused at the single-site level
 - “On-premise” data center, single cloud provider, single availability zone
- The goal of C-Ports is to investigate the use of containers across a federation of resources and adapt to multiple availability zones, multiple cloud providers, as well as hybrid [public/private] cloud infrastructure

