

**Politecnico di Milano**  
**Facoltà di Ingegneria dell'Informazione**



**Corso di Laurea Magistrale in Ingegneria Informatica**  
**Dipartimento di Elettronica, Informazione e**  
**Bioingegneria**

**Towards Virtual Machine Consolidation in**  
**OpenStack**

**Advisor: Sam Jesus Alejandro Guinea Montalvo**

**Master thesis by:**

**Giacomo Bresciani matr. 804979**

**Lorenzo Affetti matr.**

**Academic Year 2013-2014**



*dedica...*



# Ringraziamenti

Milano, 1 Aprile 2005

*Affear*



# Table of Contents

List of Figures	ix
List of Tables	1
<b>1 Introduction</b>	<b>3</b>
<b>2 State of the art</b>	<b>5</b>
2.1 Introduction . . . . .	5
2.2 Virtual Machine consolidation . . . . .	6
2.3 Cloud test environments . . . . .	6
2.3.1 Vagrant . . . . .	6
2.3.2 Chef . . . . .	7
2.3.3 Puppet . . . . .	7
2.3.4 Docker . . . . .	7
2.3.5 Dockenstack . . . . .	7
<b>3 aDock</b>	<b>9</b>
<b>4 FakeStack</b>	<b>11</b>
<b>5 Simulation tools</b>	<b>13</b>
<b>6 Nova Consolidator</b>	<b>15</b>
<b>7 Evaluations</b>	<b>17</b>
<b>8 Conclusions and future works</b>	<b>19</b>

## TABLE OF CONTENTS

---

Appendices	21
Bibliography	23



# List of Figures



# List of Tables



# Chapter 1

## Introduction



# Chapter 2

## State of the art

### 2.1 Introduction

At the beginning of the development of the thesis we were mainly focused on the implementation of a module for OpenStack that would allow to implement different consolidation algorithms and test them to see their real impact on a cloud system in terms of resource allocation. During the first phases we faced with the problem of running, testing and benchmarking our code in an OpenStack environment: to deal with aspects like scheduling, VM placement, and consolidation we needed an highly configurable system that would allow us to run simulations and benchmark to evaluate the goodness of our solution. We wanted it to be fully customizable to match different requirements and let the user customize a lot of aspect such as the structure of the environments, the number of compute nodes (the nodes that host the Virtual Machines), their fake characteristics or the OpenStack services to run. Secondly we needed a way to automatically simulate, in a repeatable way, the workload generated from user applications that normally run on an OpenStack installation. At last we realized that it would be very useful to show the real time data of the simulations to analyze the behavior of the system in different configurations. Therefore we decided to develop aDock, a suite of tools for creating performant, sandboxed, and configurable cloud infrastructure experimentation environments that developer, sysadmins and researchers can exploit to access a fully functional cloud installation of OpenStack. For that reason this chapter is divided into two sections that describe the state of the arts of Virtual Machine

consolidation and of Cloud test environments.

## 2.2 Virtual Machine consolidation

In the cloud world is fundamental, as in any engineering field, to be able to test environment configurations and algorithms, both to analyze the behavior of tested code that integrates with a real environment and to benchmark and collect data for researches and experimentations. Unfortunately it can be expensive and complex to create and manage a cloud test environment in terms of time, resources and expertises, especially if the hardware resources like server machines or network infrastructures are limited. Fully understand and handle an OpenStack installation is not easy, especially for non sysadmins like developers or researchers, it has infact an high learning curve and often it is necessary a lot of time to achieve a good and desired result. To reduce the impact of these complications are available some tools that make the process of setting up a cloud infrastructure experimentation environment more easy and manageable. The three main ones are Chef, Puppet, and Dockenstack. In order to achieve the goal of running the desired sandboxed environment on limited hardware resources the three solution are used together with two core technologies for the cloud world, Vagrant and Containers (Docker), both outlined below.

## 2.3 Cloud test environments

### 2.3.1 Vagrant

Vagrant is a software, written in Ruby, that allows to create configure and manage virtual development environments. It is a wrapper around virtualization software such as VirtualBox, KVM, VMware and could be used together with configuration management software such as Chef and Puppet. One of its major benefits is that, given a Vagrantfile, is simple as run a command to have a full pre-configured development environment: `Vagrant up` In addition there is a big collection of Vagrantfile available online and free to download.



**2.3.2 Chef**

**2.3.3 Puppet**

**2.3.4 Docker**

**2.3.5 Dockenstack**



# Chapter 3

## aDock



# Chapter 4

## FakeStack



# Chapter 5

## Simulation tools





## Chapter 6

### Nova Consolidator



# Chapter 7

## Evaluations



## Chapter 8

### Conclusions and future works



# Appendices





# Bibliography