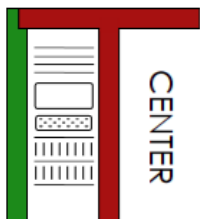




# self-service cloud of virtual machines

Antonio Cisternino, Maurizio Davini, Marco Mura - ITC Center  
Università di Pisa



# Introduction

---

- ▶ In 2010 Patterson: Cloud is a business model rather than a technology
- ▶ Cloud definition is still «cloudy» and is declined in several flavors (IaaS, PaaS, SaaS)
- ▶ Goal: use a computing infrastructure without knowing where and how much resources are used
- ▶ Keywords: flexibility (server), elasticity (client)
- ▶ The plug problem... What is the right abstraction between the cloud and users?
- ▶ So far VMs (and x86) look the better interface for general purpose cloud services



# Architecture

---

- ▶ Cluster of Hyper-V manager
- ▶ Implemented in F# and WMI
- ▶ Operations:
  - ▶ Create/Destroy VM
  - ▶ Live and off line migration
  - ▶ Diff disk imaging
- ▶ Self service Web
- ▶ Expert system for VM orchestration and policy

Octopus

Hyper-F

WMI Hyper-V calls

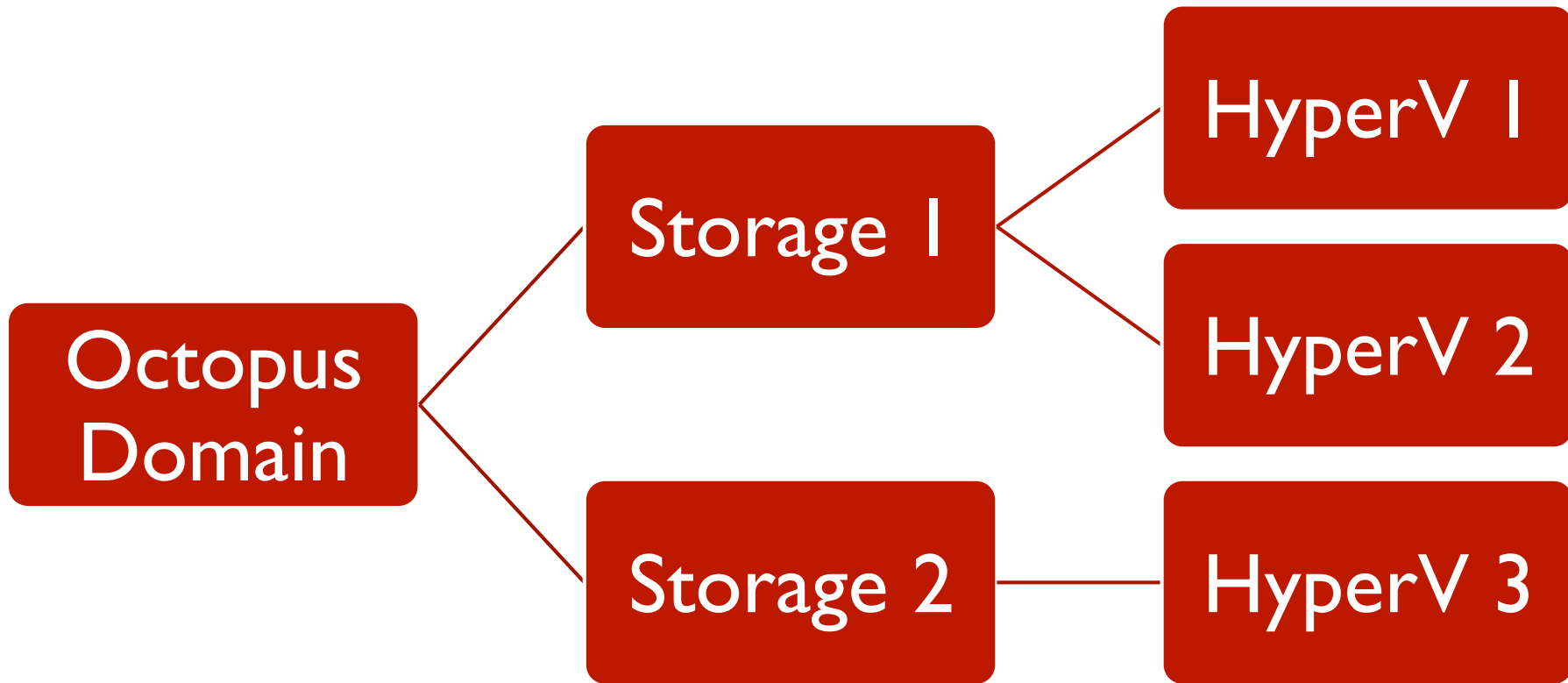
Shared Storage

DHCP  
Networking  
DNS



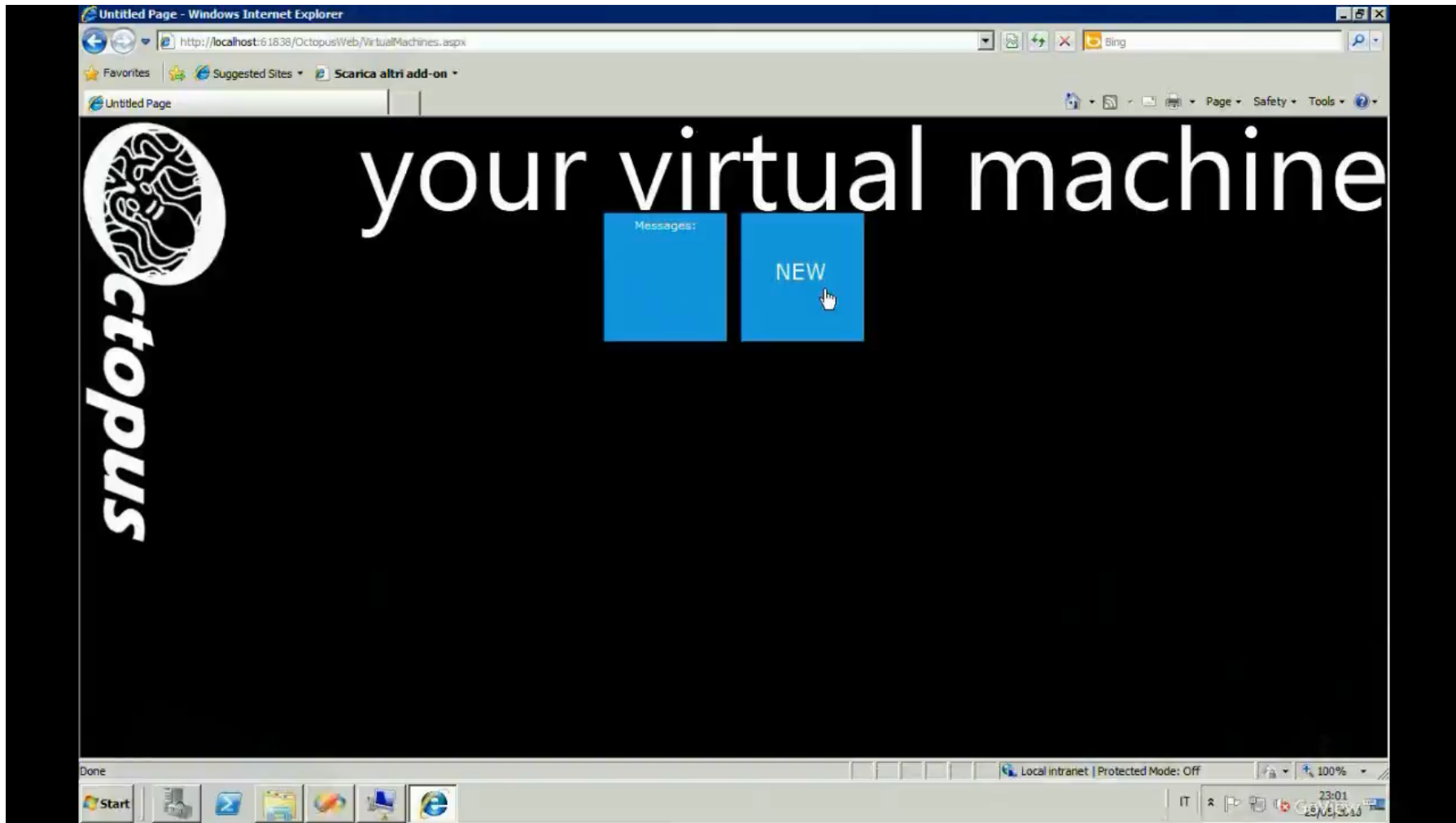
# Model

---



# Creating a cluster of VMs

---



# Checking VMs from mobile

---



# Rule based VM management

---

- ▶ Using a rule based approach to resource management
- ▶ Example: moving VMs to optimize power consumption (packing VMs on HW nodes)
- ▶ Use of CLIPS expert system with the (great) .NET wrapper
- ▶ Octopus runtime asserts facts about the cloud (users, VMs, VM status, counter etc)
- ▶ Rule based systems allow to express efficiently arbitrary complex policies



# CLIPS environment

---

- ▶ This is still a prototype!
- ▶ Rule example

```
(require "Octopus")
```

```
(octopus-create-vm "Pippo" "LinuxImg" 2 2048 1024)
```

```
(defrule suspend-when-idle (octopus-vm-idle ?x) =>  
  (if (octopus-suspend-vm ?x) then  
    (printout t ?x " suspended") else  
    (printout t ?x " failed to suspend"))))
```

```
(assert (octopus-vm-idle "Pippo"))
```

```
(run)
```





# The experiment

---

- ▶ HW: Acer AR585, 2U, 4cpu (8 cores), 32Gb RAM, 8NUMA (Windows)
- ▶ SW: Fluent 12.1, Win2008 R2 e Hyper-V
- ▶ Test: Fluent test con ~1 mln cells
- ▶ Run (using MPI from HP and MS):
  - ▶ 16 processes in shared memory
  - ▶ 16 processes without shared memory
  - ▶ Virtual Cluster of 16 nodes (1 core, 1584Mb) and 1 head node (2 cores, 2048Mb) and MS HPC v3 b2



# An interesting result

Hardware											
	hpmpi				msmpi				Total Avg	Total	Total
	Virtual cluster				Virtual cluster				Iter (s)	StdDev	#exp
Precision	hpmpi				msmpi				Total Avg.	Total	Total
double	Avg. Iter (s)	StdDev	#exp	Avg. Iter	StdDev	#exp	Avg. Iter	StdDev	#exp	Average	
single	11,72	0,77	10	13,17	1,008	10	11,81	1,333	1,760	12	
double	12,17	1,06	9	13,27	1,26	5	12,40	1,11	14		
no	11,57	0,37	10				11,57	0,37	10		
Shrd mem	12,84	1,06	9	13,27	1,26	5	12,99	1,11	14		

**Virtual execution is faster than native on the same HW!!!**



# Conclusions

---

- ▶ VMs and x86 are a great way for having a standard API between software and cloud for general purpose computing
- ▶ Lease renew is an efficient way for stimulating users at VM recycling
- ▶ It is possible to let users request their own resources
- ▶ Rule based expert systems are a promising way to express complex policies for cloud management.

