

Reingegnerizzazione del prototipo di gestione di macchine virtuali chiamato Octopus

Marco Mura
mura@di.unipi.it



4° Borsisti Day – 13/09/2013



- Rapida diffusione della virtualizzazione
 - migliore organizzazione delle risorse
 - sistemi isolati per sviluppo e test
 - facilita la creazione di sistemi ridondati
- Numero di VM da amministrare è o sarà presto insostenibile coi sistemi attuali
 - Da decine di persone per singolo computer a centinaia di VM per singolo amministratore
 - Quasi eliminazione della gestione fisica e focus su quella logica

Perche' di Octopus (1)

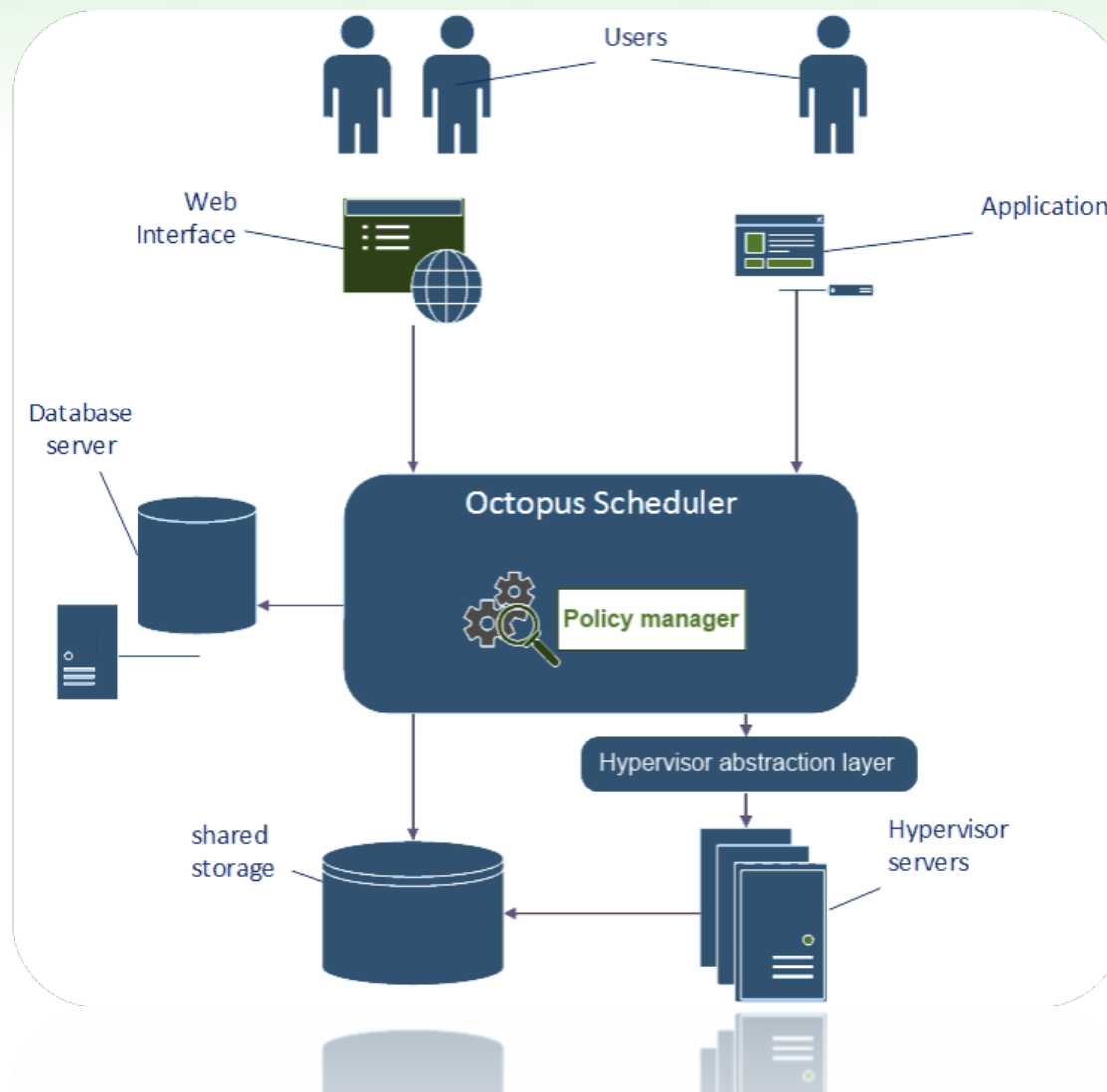
- Interfaccia web per gli utenti per creare e gestire le proprie VM in totale autonomia
- Template pre-configurati per creare le VM in pochi minuti

	Ubuntu 10.04 x64	Windows Server 2008 R2 x64
Installazione da zero	~ 23 minuti	~ 37 minuti
Istanziamento vm	~ 2 minuti	~ 4 minuti

Perche' di Octopus (2)

- Verifica dell'uso delle risorse e gestione di esse da parte del sistema mediante un sistema esperto
- L'amministratore può personalizzare il comportamento di Octopus scrivendo le regole del sistema esperto
- Conflitti del sistema risolti dal Sistema
- Scalabilità da uso personale su un singolo PC con hypervisor ad un cluster di HV

Architettura del sistema



- Politiche del sistema gestite dal sistema esperto CLIPS invece che da un Sistema di Workflow:
 - Il sistema viene popolato di fatti che descrivono l'ambiente attuale
 - Il sistema reagisce secondo regole, eseguendo gli script delle politiche
- Le politiche, completamente riscrivibili, sono espresse in un linguaggio che ricorda LISP

Politiche: esempio di fatti

- Facts: asseriscono qualcosa di vero, il loro insieme rappresenta una visione del mondo

(assert (node-name hypervis1))

(assert (node-memoryfree hypervis1 10654))

(assert (vm-name vm1))

(assert (vm-server vm1 hypervis1))

(assert (vm-owner vm1 mura))

- ...e anche gli eventi sono asseriti come fatti

(assert (event-creating-vm guid1 mura 1 1 2048))

(assert (event-starting-vm guid2))

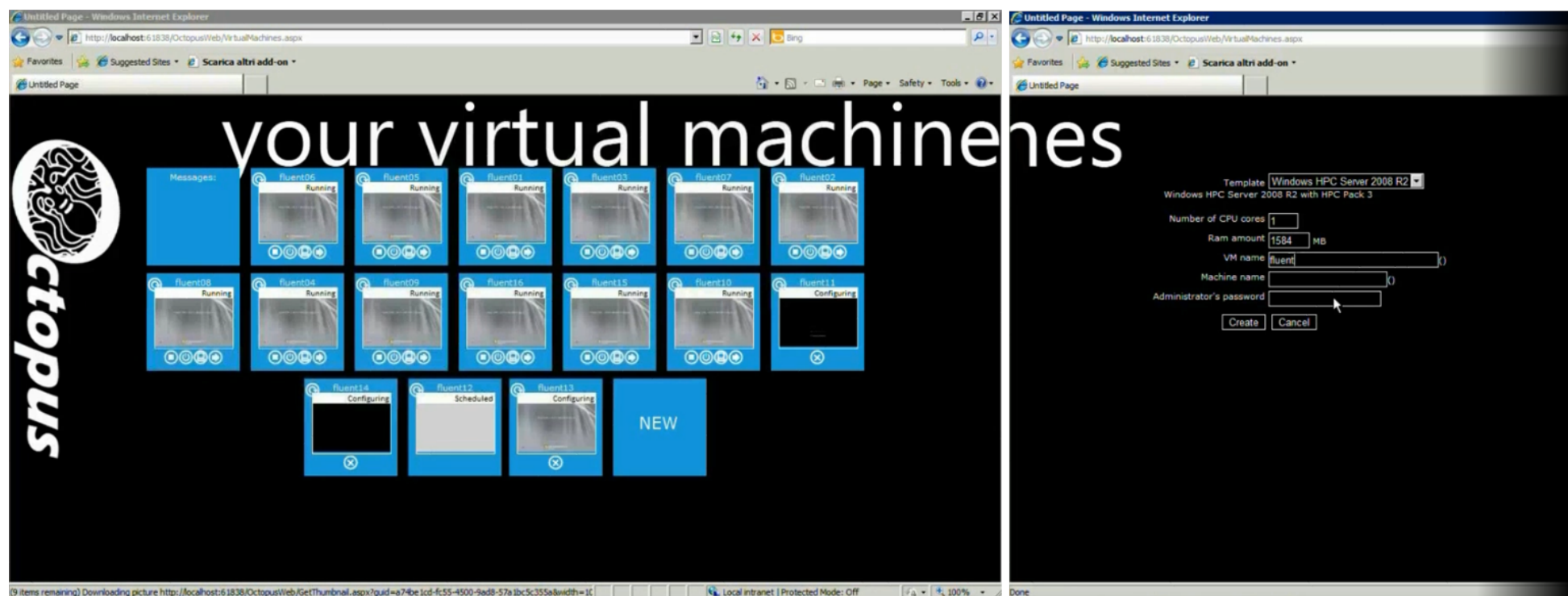
- Rules: descrivono come il sistema reagisce ai nuovi fatti asseriti

```
(defrule createVmRule
  ?f <- (event-creating-vm ?reqld ?owner ?numVm ?numCpu ?ramSize)
  =>
  (bind ?alreadyCreated (countVmPerOwner ?owner))
  (if (> (+ ?alreadyCreated ?numVm) ?*maxVmPerUser*)
    then (deny-newvm ?reqld (str-cat "max " ?*maxVmPerUser* " VMs per user"))
    else
      (bind ?nodes (assignNodesForNewVm ?numVm ?numCpu ?ramSize))
      (if (< (length$ ?nodes) ?numVm)
        then (deny-newvm ?reqld "insufficient resources for this request")
        else (create-newvm ?reqld ?nodes)
      )
    )
  )
  (retract ?f)
)
```

Sistemi esperti vs. Workflow

- Gli HV più utilizzati fanno uso di sistemi di workflow per orchestrare l'automazione
- -: non è ben definito il concetto di conflitto (il sysadmin garantisce la consistenza di insieme complesso di regole)
- I sistemi esperti esprimono flussi come concatenazione di attivazioni di regole
- +: le regole sono locali ed è possibile definire una politica di risoluzione conflitti

Interfaccia user-oriented



- Supportare diversi virtualizzatori
Attualmente supporta solo Microsoft Hyper-V
- Migliorare la gestione delle VM
Creazione di snapshot con branching;
Creazione di template da una VM esistente...
- Semplificazione dell'installazione
- Supporto a diversi domini di autenticazione
- Portabilita' su linux (grazie a Mono)

- Definizione di un'architettura aperta verso più HV
- Revisione dei modelli concettuali necessari (HV e storage multipli)
- Siamo a buon punto nella riscrittura del sistema rimuovendo le dipendenze da Hyper-V e passando ad un driver model
- Preparazione di un nuovo drop aggiornato del codice

Grazie dell'attenzione

- Marco Mura – mura@di.unipi.it
- Antonio Cisternino – cisterni@di.unipi.it
- Source code:
<http://octopus.codeplex.com>