Xen, Alta Affidabilità e Assegnazione Dinamica delle Risorse

Riccardo M. Cefalà
Mirko Mariotti
Leonello Servoli

Incontro di lavoro della CCR INFN
Laboratori Nazionali di Legnaro
10-11 Dicembre 2008

Introduzione:
Il Sito INFNGrid di Perugia

- La Farm nasce dalla compartecipazione dei vari gruppi di ricerca e a partire dal 2004 è entrata a far parte di INFNGrid.
- Offre circa 200 CPU e 30TB di storage.
- Serve sia Utenza Locale che GRID.

Introduzione: Virtualizzazione @ Perugia

Crescente Impiego della Virtualizzazione (Xen) nel sito INFNGrid Perugia e il Dipartimento di Fisica dell'Università degli Studi di Perugia:

- Maggiore offerta di Servizi risparmiando Risorse
- Gestione più semplice rispetto Hardware Reale
- Aspetti intrinseci di Alta Affidabilità & Disponibilità
- Possibilità di sviluppo di soluzioni ad-hoc

Introduzione: Virtualizzazione @ Perugia

Molti servizi offerti nella nostra Sezione, sono da tempo implementati con successo su VM (DomU); tra questi, i principali componenti GRID:

- Esempi: *CE, UI, SE, WN*

Nel passato sono state impiegate varie tipologie di Storage per le VM (DomU):

- iSCSI, ATA over Ethernet, Fibre Channel, NFS

Virtualizzazione @ Perugia
DomU Diskless

- Un ulteriore sviluppo per quanto riguarda i DomU, è stato quello di realizzare un sistema completamente diskless.
- Ciò rende possibile avere un DomU interamente caricato in RAM e slegato da ogni altro Vincolo.
- Molti servizi possono essere ospitati da queste macchine.

Virtualizzazione @ INFN Perugia
DomU Diskless

- La realizzazione è avvenuta avvalendosi del sistema di sviluppo di distribuzioni di Gentoo (Catalyst).
- Abbiamo realizzato degli Stage derivati da quelli usati per produrre *LiveCD,* opportunamente modificati per permetterne il boot via rete.
- Una volta creata la macchina siamo in grado di caricarla su un Dom0 remoto che accede a un Reposostory di VM Comune.

Domini non Privilegiati su Storage Condiviso

Abbiamo appurato che l'impiego di un repository di Immagini di DomU centralizzato offre diversi vantaggi:

- Le immagini sono disponibili presso tutti i Dom0 che accedono allo Storage.
- I servizi che le VM offrono sono indipendenti da guasti Hardware dei Singoli Dom0.
- Vi è un unico repository da mantenere.

E i Domini Privilegiati?

E' possibile applicare gli stessi principi ai Dom0?

- Similmente e unitamente ai DomU su Storage Centralizzato, fornire un immagine generica di un Dom0 avviabile da rete significa che:

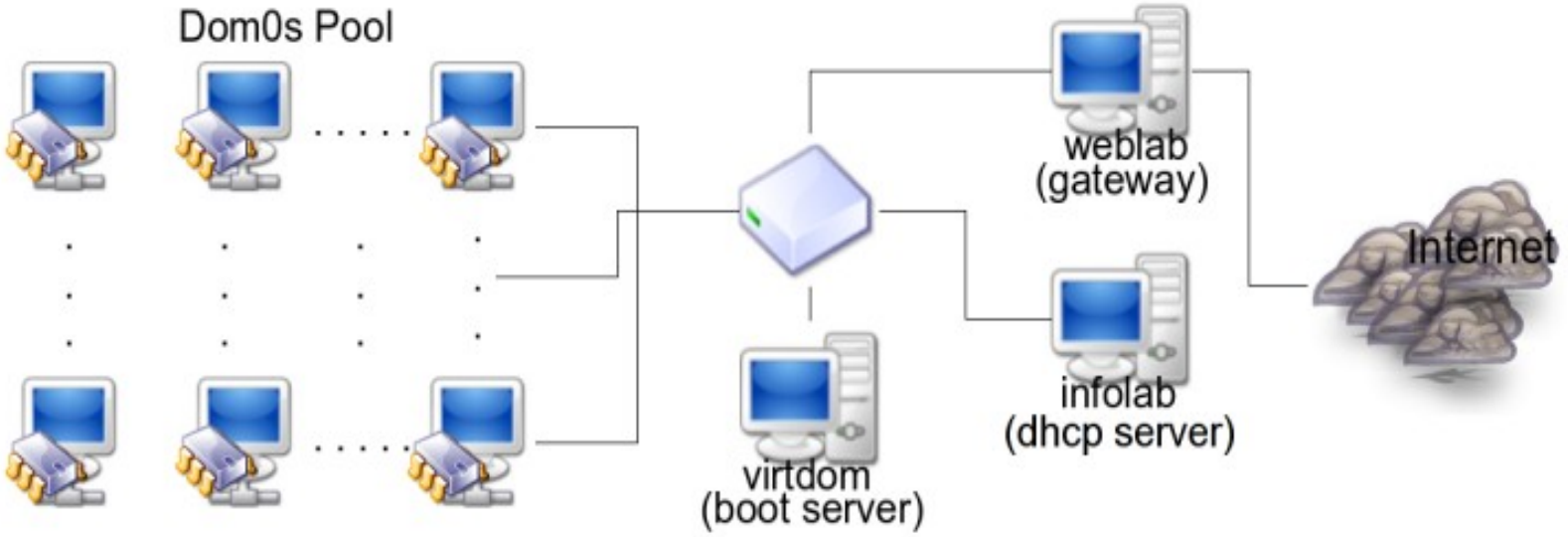
Ogni Macchina Disponibile può eseguire Macchine Virtuali*

A tale scopo è stata sviluppata una metodologia per l'esecuzione di Dom0 su NFS.

Dom0 via NFS

La realizzazione prevede l'impiego di:

- Un ambiente per il boot da rete (PXE, DHCP, TFTP, GRUB)
- Un filesystem di root su NFS
- Customizzazioni Varie



Dom0 via NFS (DHCP su Infolab)

- Configurazione standard per il boot via rete.

```
option space PXE;
option PXE.mtftp-ip               code 1 = ip-address;
option PXE.mtftp-cport            code 2 = unsigned integer 16;
option PXE.mtftp-sport            code 3 = unsigned integer 16;
option PXE.mtftp-tmout            code 4 = unsigned integer 8;
option PXE.mtftp-delay            code 5 = unsigned integer 8;
option PXE.discovery-control      code 6 = unsigned integer 8;
option PXE.discovery-mcast-addr   code 7 = ip-address;

class "pxeclients"
{
        match if substring (option vendor-class-identifier, 0, 9) = "PXEClient";
        option vendor-class-identifier "PXEClient";
        vendor-option-space PXE;

        option PXE.mtftp-ip 0.0.0.0;
}

option option-150 code 150 = text;
option option-128 code 128 = string;
option option-129 code 129 = text;
```

```
host post01 {
    next-server           192.168.0.100;
    hardware ethernet     00:11:2f:a8:b4:d4;
    fixed-address         192.168.0.101;
    filename              "pxegrub";
    option option-150     "/grub/post01.conf";
}
```

Dom0 via NFS (pxeGRUB)

- Il DHCP su virtdom fornirà i parametri di rete, il bootloader e il relativo file di configurazione.

- La seconda entry farà si che il bootloader contatti il secondo TFTP server (virtdom)

```
default 0
timeout 2
password xxxxx

title   Infolab terminal
        root (nd)
        kernel /kernel/ltsp-2.6.17.8 rw root=/dev/ram0
        initrd /initrd/ltsp-2.6.17.8.gz

title   On-demand virtual machines
        password xxxxx
        dhcp
        tftpserver 192.168.0.5
        root (nd)
        configfile /grub/post01.conf
```

Dom0 via NFS (TFTP, NFS su virtdom)

- virtdom fornirà un secondo file di configurazione per pxeGRUB specificando il filesystem via NFS offerto da virtdom stesso

```
#virtdom exports
/opt/ondemand/root              192.168.0.0/255.255.255.0(ro,no_root_squash,async)
/opt/ondemand/imgs              192.168.0.0/255.255.255.0(rw,no_root_squash,async)
```

```
#post-pIV-realtek - NFS root
default 0
timeout 2
title   Dom0 - NFS
        root (nd)
        kernel /hypervisors/xen-3.1.gz dom0_mem=32M
        module /kernels/kernel-2.6.20-xen-r6-pIV-rltk ip=dhcp root=/dev/nfs \
               nfsroot=192.168.0.5:/opt/ondemand/root
```

- Il Kernel dei Dom0 deve supportare rootfs over NFS

Dom0 via NFS (NFS Root Filesystem)

- Il filesystem di root (Debian etch) deve essere opportunamente configurato.
- Condiviso => Read Only. E le porzioni *Read/Write* necessarie al funzionamento (es. `/var`)?
- Impieghiamo una tecnica tipica dei *sistemi embedded*:

I file che devono essere scrivibili vengono sostituiti da link simbolici nelle loro posizioni orginali e spostati in una nuova posizione (`/flash`).
Ogni macchina creerà il tree delle locazioni scrivibili in RAM che verrà montato in `/flash` in fase di boot.

Dom0 via NFS (NFS Root Filesystem)

```
virtdom:~# ls -l /opt/ondemand/root/var/
[...]
drwxrwsr-x  2 root staff 4096 Oct 28  2006 local
lrwxrwxrwx  1 root root    15 Nov 16 18:37 lock -> /flash/var/lock
lrwxrwxrwx  1 root root    14 Nov 16 18:37 log -> /flash/var/log
drwxrwsr-x  2 root mail  4096 Oct 29 15:59 mail
drwxr-xr-x  2 root root  4096 Oct 29 15:59 opt
lrwxrwxrwx  1 root root    14 Nov 16 13:06 run -> /flash/var/run
drwxr-xr-x  3 root root  4096 Oct 29 16:01 spool
drwxr-xr-x  3 root root  4096 Nov 16 09:42 xen
```

```
virtdom:~# ls -l /opt/ondemand/root/flash/*
/opt/ondemand/root/flash/etc:
total 4
drwxr-xr-x 4 root root 4096 Nov 29 12:18 udev

/opt/ondemand/root/flash/var:
total 20
drwxr-xr-x 5 root root 4096 Nov 26 10:59 lib
drwxrwxrwt 2 root root 4096 Nov 16 13:07 lock
drwxr-xr-x 5 root root 4096 Nov 16 12:56 log
drwxr-xr-x 2 root root 4096 Nov 21 17:27 run
drwxrwxrwt 2 root root 4096 Nov 23 16:37 tmp
```

```
post23:~# ls -al /flash/
total 4
drwxrwxrwt  4 root root   80 Jan 18 11:34 .
drwxr-xr-x 20 root root 4096 Nov 26 09:46 ..
drwxr-xr-x  4 root root   80 Jan 18 11:34 etc
drwxr-xr-x  7 root root  140 Jan 18 11:34 var
```

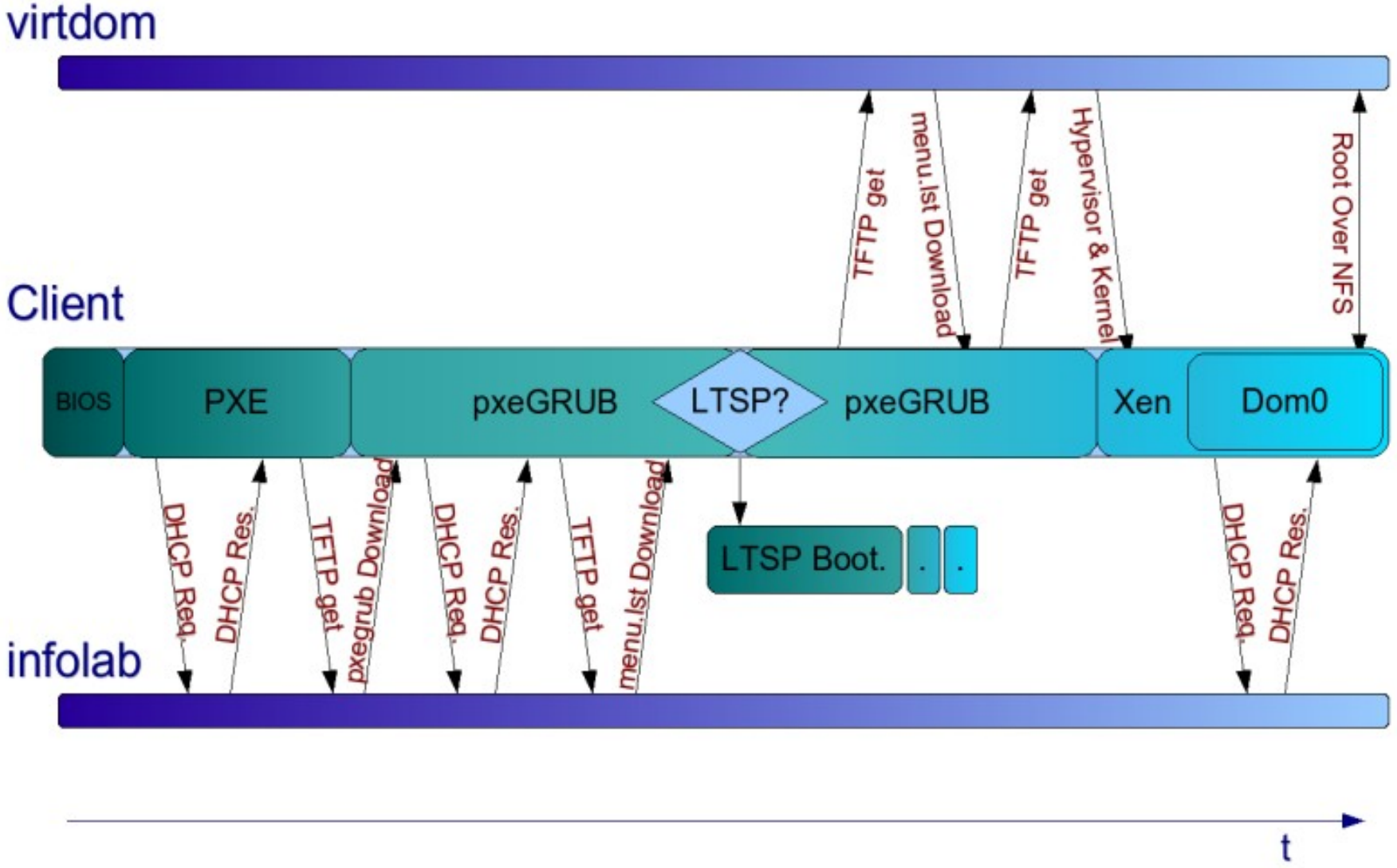
Link consistenti sia in chroot che sui Dom0.

Dom0 via NFS (Modifiche agli script di Xen)

- Configurazione della rete di Xen 3.1:
  - `eth0` → `peth0`
  - Crea un'interfaccia virtuale `eth0` e assegna essa l'IP.
- Ciò causa l'interruzione della connessione NFS!
- E' quindi necessario modificare il comportamento di default.

In Xen 3.2 il meccanismo è stato modificato in modo simile e non costituisce più un problema.

Dom0 via NFS (Timeline)



Dom0 via NFS (Sviluppi Futuri & Riferimenti)

Prossimamente prevediamo di realizzare Dom0 completamente Diskless (Senza NFS).

Il nostro Howto è pubblicato sul Wiki di Xen:

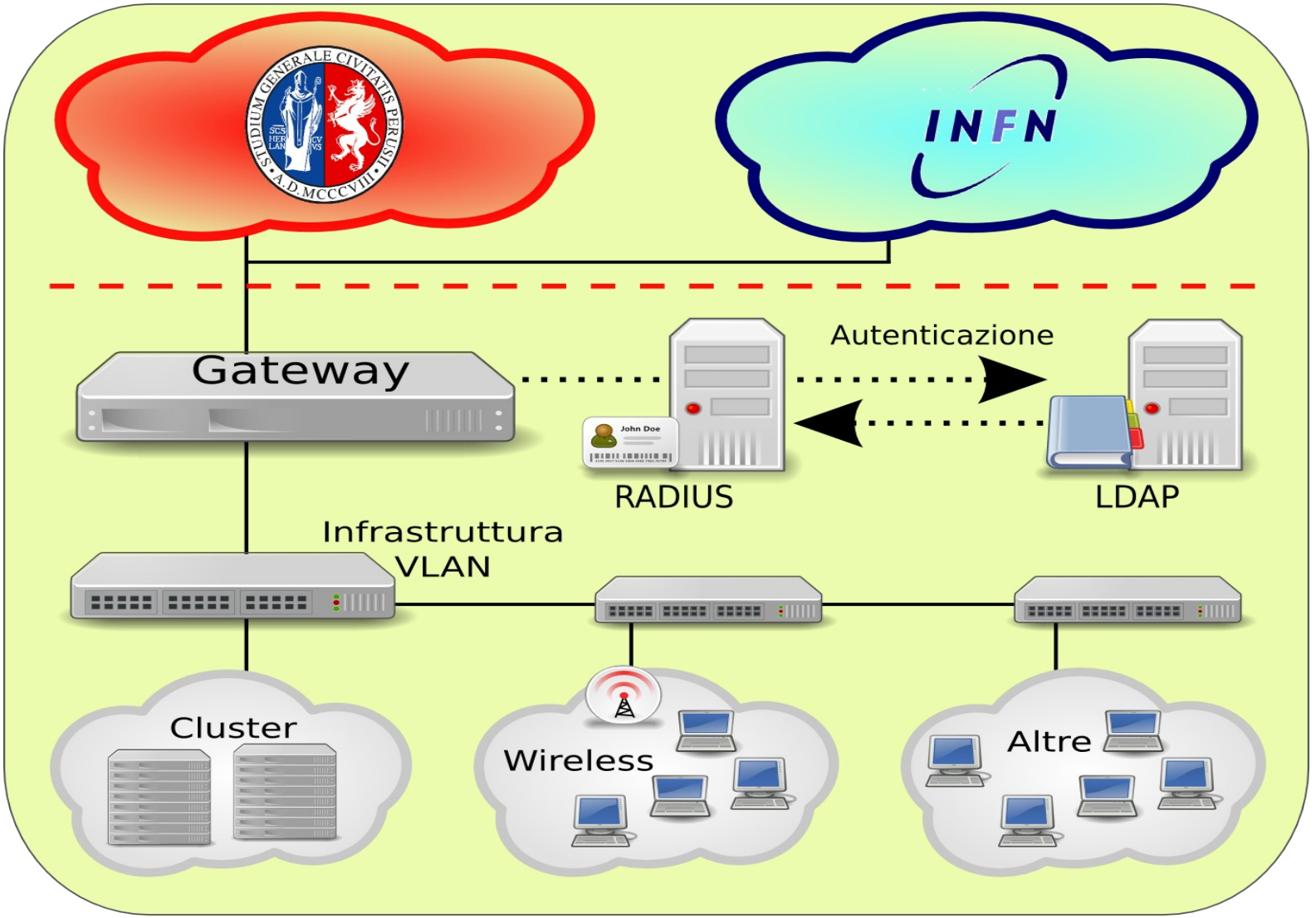
http://wiki.xensource.com/xenwiki/HowTos

Xen Dom0 root filesystem over NFS Howto

Virtualizzazione @ Perugia: The Net Result

- Come in molte sedi, l'infrastruttura di rete a Perugia poggia sulla tecnologia VLAN.
- L'integrazione di tale tecnologia con la versatilità delle macchine virtuali permette la realizzazione di scenari dinamici e altamente configurabili
- Ad esempio:
  - Tutte le macchine del "*Laboratorio d'acquisizione dati*" possono diventare Dom0 su un segmento di rete da tutt'altra parte nell'edificio.

Virtualizzazione @ Perugia: The Net Result



Virtualizzazione @ Perugia: The Net Result

- Il Gateway nella figura precedente è un DomU diskless direttamente collegato alla dorsale VLAN che “natta” le varie reti locali (Wireless, Labs, etc.).
- In precedenza il compito era svolto da una macchina fisica. Il tempo di ripristino dopo un guasto è di ordini di grandezza superiore rispetto a rilanciare la stessa Istanza Virtuale di Gateway su un altro Dom0.

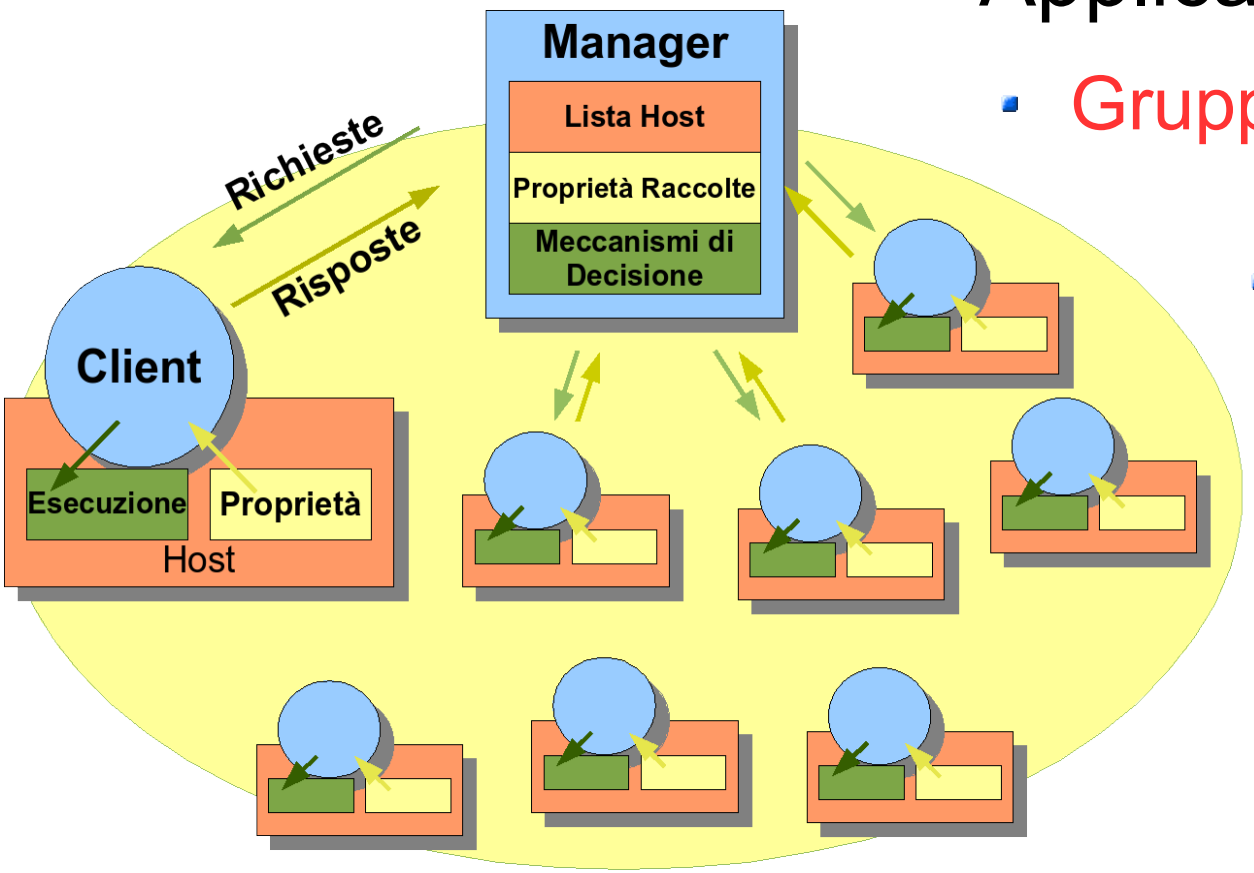
Gestione Dinamica delle VM (Introduzione)

- Gestire manualmente le VM diventa difficoltoso al crescere del numero.
- La Gestione Automatica delle VM apre la strada a nuove applicazioni per:
  - Ottimizzare l'impiego delle risorse
  - Risolvere problemi dovuti all'eterogeneeità degli ambienti
  - Applicazioni di Alta Affidabilità e Disponibilità
  - La Semplificazione delle operazioni di Manutenzione

Gestione Dinamica delle VM (Prototipo Introduzione)

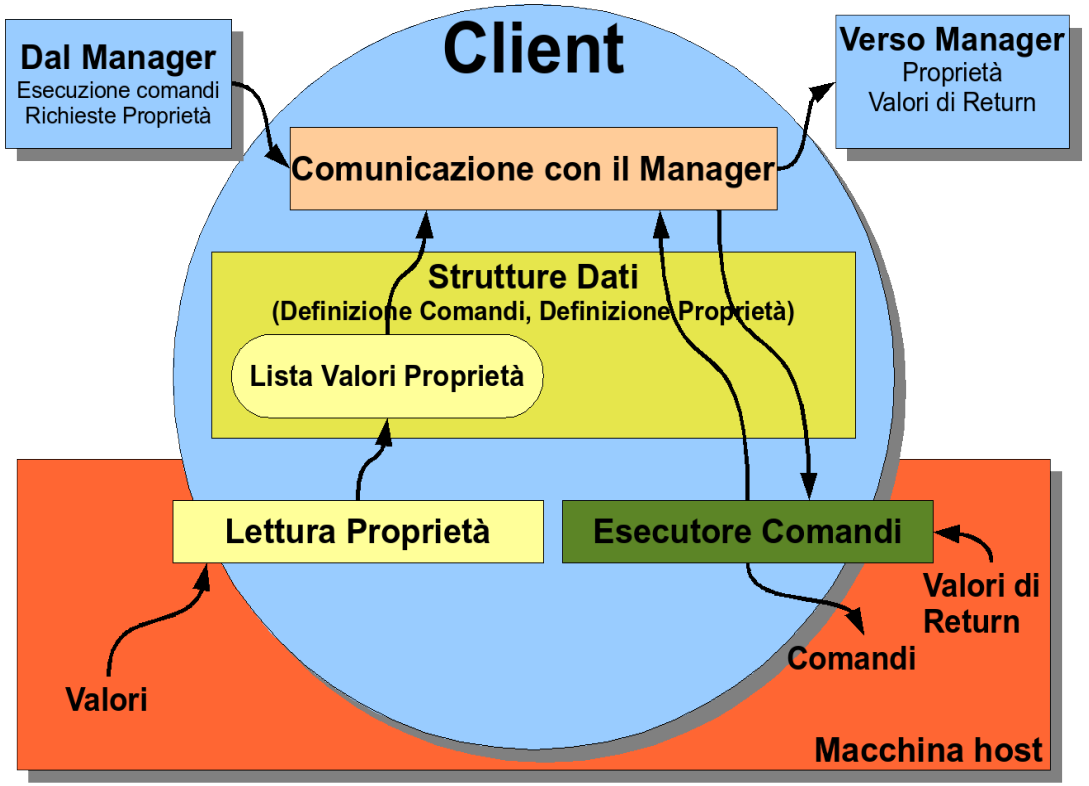
- Nell'ultimo anno è stato sviluppato un prototipo con lo scopo di automatizzare il processo di gestione delle VM
- L'applicazione deve essere flessibile e fornire strumenti per:
  - Definizione di classi di VM (Ambienti)
  - Monitoring delle Risorse
  - Controllo delle Richieste d'uso delle Risorse
  - Meccanismi di Decisione per la Gestione

Gestione Dinamica delle VM (Prototipo Architettura Generale)



- Applicazione Client-Server
  - Gruppi di Client e Manager
- Indipendenza da Strutture esistenti
  - Batch System
  - RMS

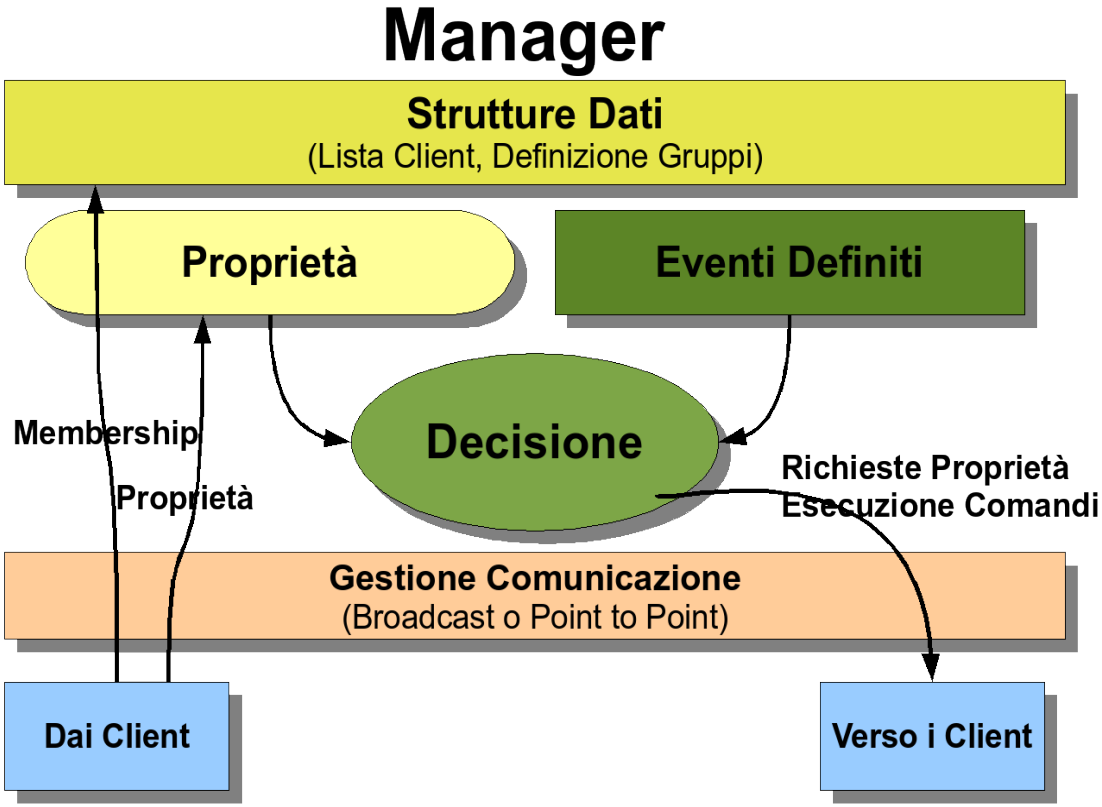
Gestione Dinamica delle VM (Prototipo - Client)



- Raccolta informazioni sistema: *Proprietà*
- Comunicazione Proprietà
- Esecuzione comandi per conto del Manager
- Plugin Comandi e Estrazione Proprietà
  - Ampie possibilità di configurazione

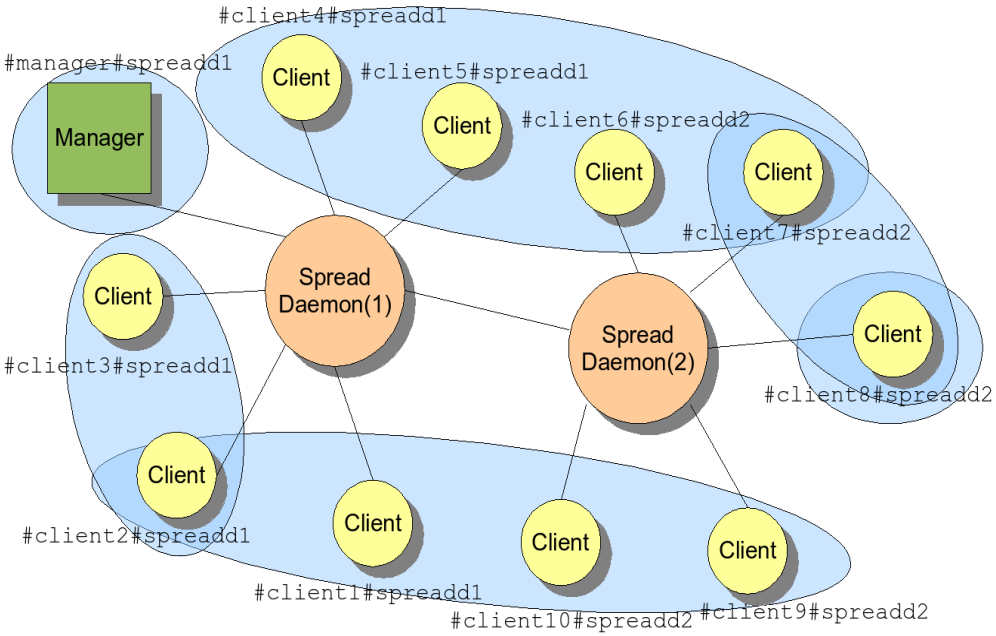
Gestione Dinamica delle VM (Prototipo - Manager)

- Invio Richieste Proprietà ai Client
- Aggregazione
- Logica di Decisione Basata su Eventi e Proprietà
- Invio Comandi ai Client



Gestione Dinamica delle VM (Prototipo – Comunicazione)

- Spread Toolkit: Group Communication System
- Definizione di Protocolli a livello Applicazione
- Gestione Gruppi
  - Multicast e Point to Point
- Qualità del Servizio
  - Garanzia di Consegna
  - Ordinamento
  - Efficienza



Gestione Dinamica delle VM (Prototipo - Comunicazione)

- Membership Messages

campi	MEMBERSHIP_MESSAGE	group	reason	extra
tipo	*Intero*	*Stringa*	*Intero*	*Stringa*
significato	*Id Messaggio*	*Nome Gruppo*	*Id. Evento*	*Nome Client*

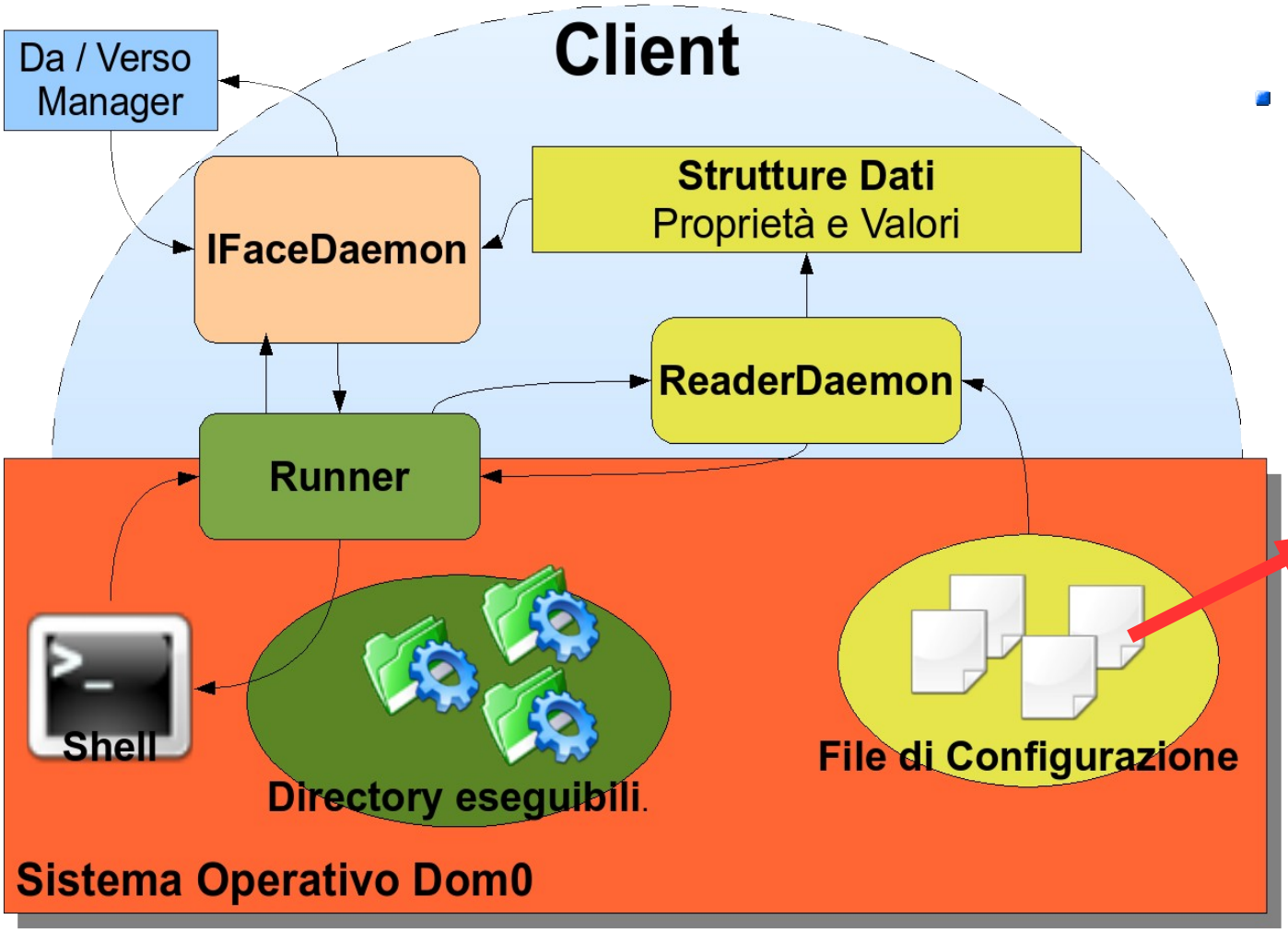
- Regular Messages

campi	REGULAR_MESSAGE	groups	message	msg_type	sender
tipo	*Intero*	*Lista*	*Stringa*	*Intero*	*Stringa*
significato	*Id Messaggio*	*Destinatari*	*"Payload"*	*Codice Mess.*	*Mittente*

- GETPROP, FETCHPROP
- EXECUTE
- CONTROL

- PROPVAL, PROPLIST, PROPERR
- EXECVAL, EXECERR
- CLB_CONTROL

Gestione Dinamica delle VM (Implementazione Client)

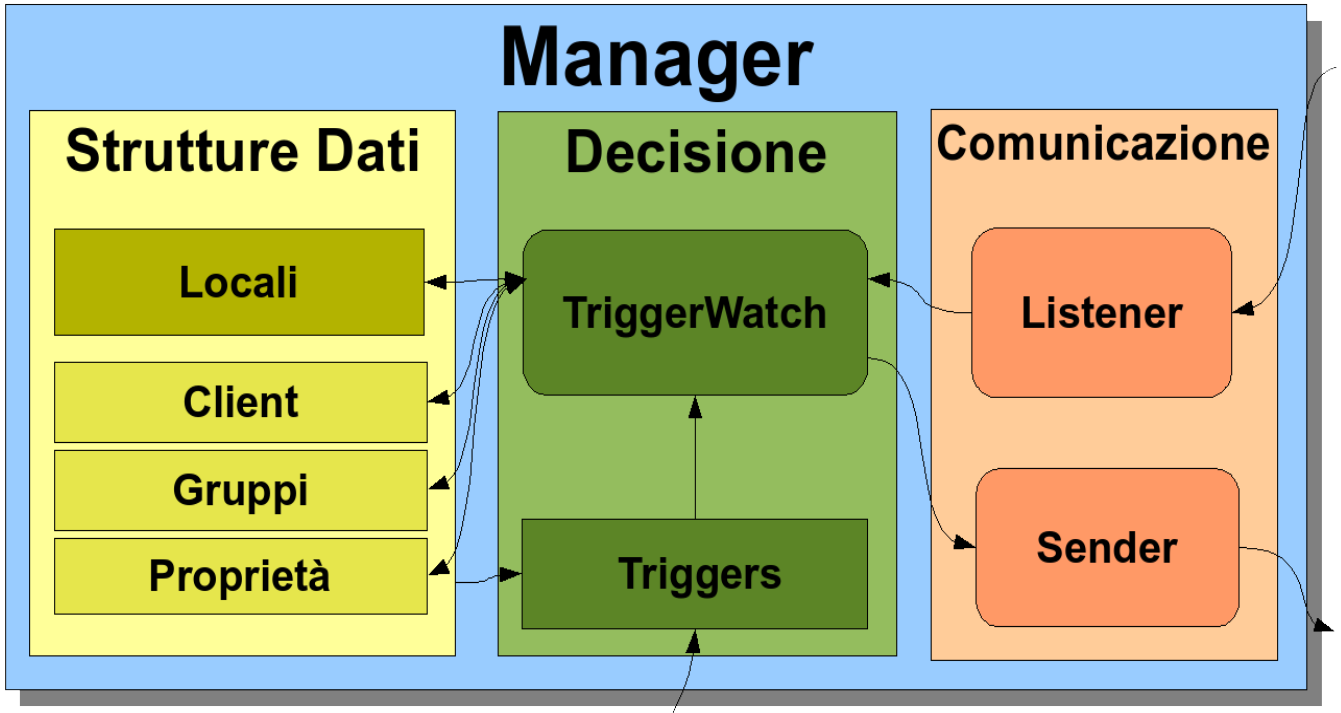


- Python
- Multithread
  - IFaceDaemon
  - ReaderDaemon
  - Runner

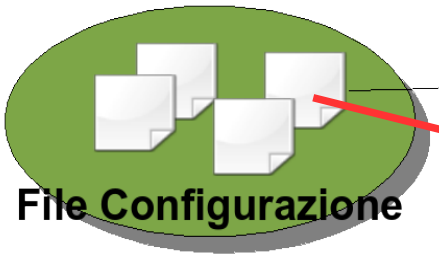
Esempio Proprietà:

```
[ram_free]
command = ram
parametes = MemFree
return_type = int
ticks = 60
```

Gestione Dinamica delle VM (Implementazione Manager)



- Python
- Multithread
  - Listener, Sender
  - TriggerWatch
- Trigger

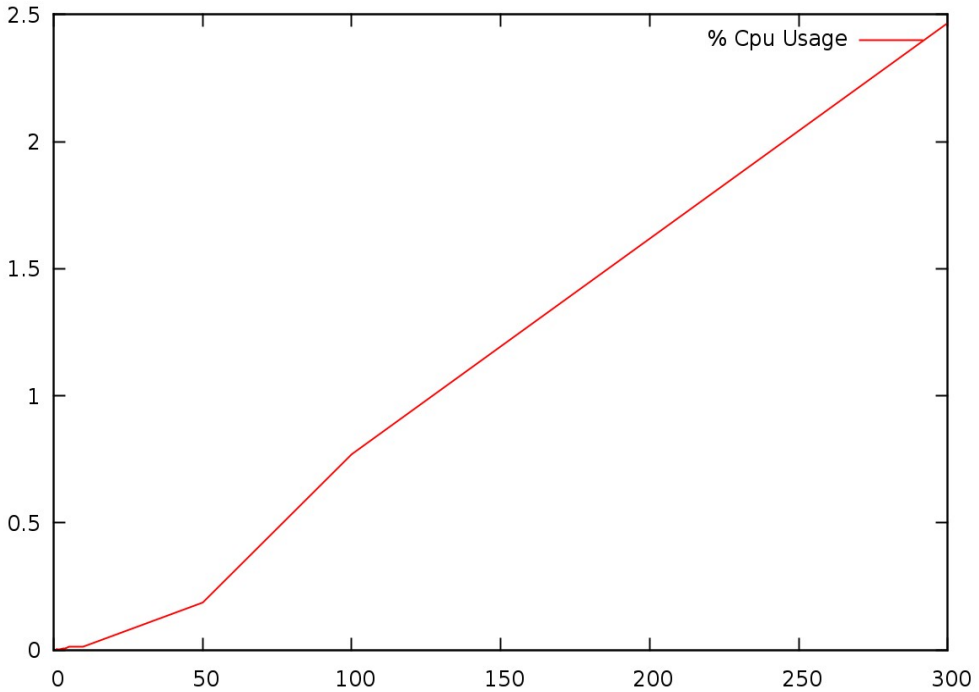


Esempio Trigger:

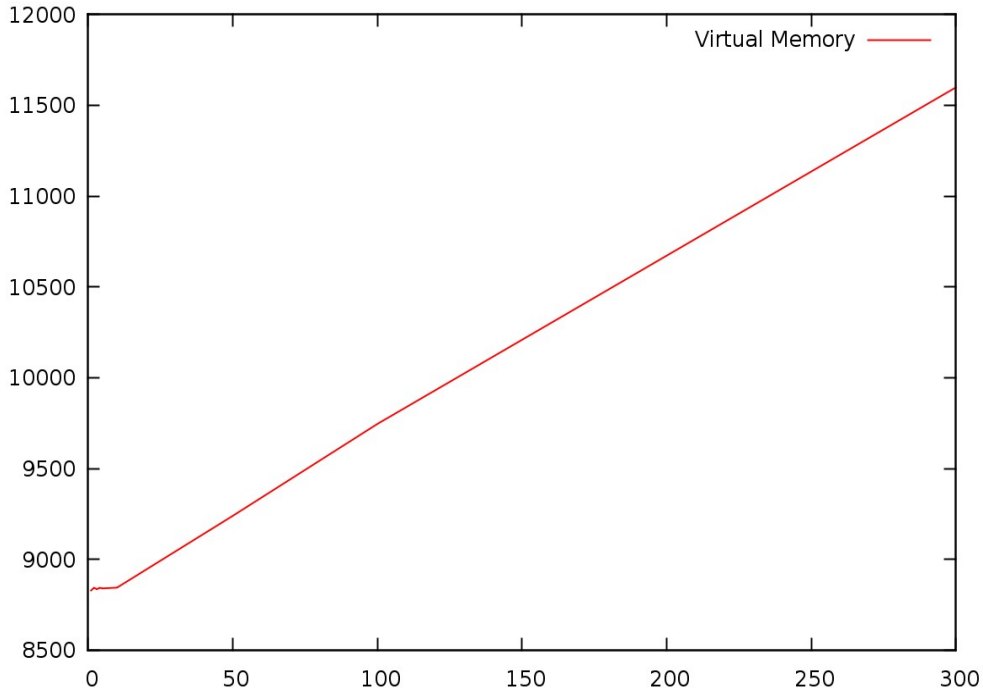
```
[client_query]
expr = { len(manager.clients) != 0
cmd = { sender.send_get_property()
ticks = 30
on_update_check = no
```

Gestione Dinamica delle VM (Prototipo – Test Scalabilità 1)

- Il prototipo è stato testato in varie condizioni fino a 300 Client ,1 Manager e messaggi di 10 - 100Kb

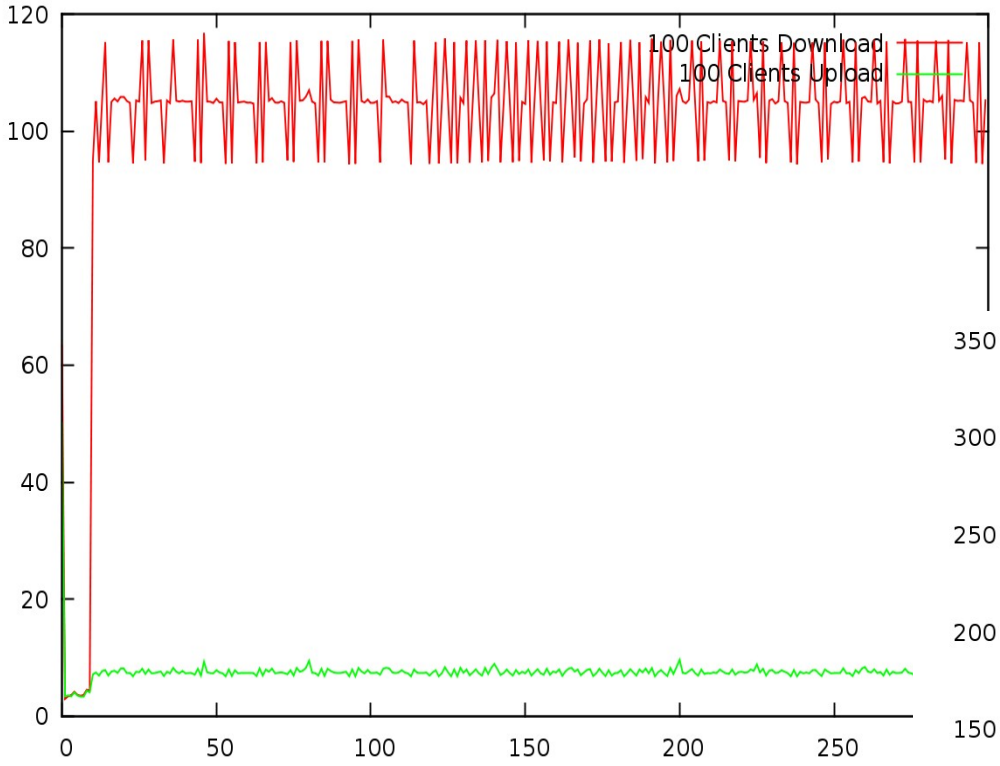


Manager CPU Load / Client (5min average)

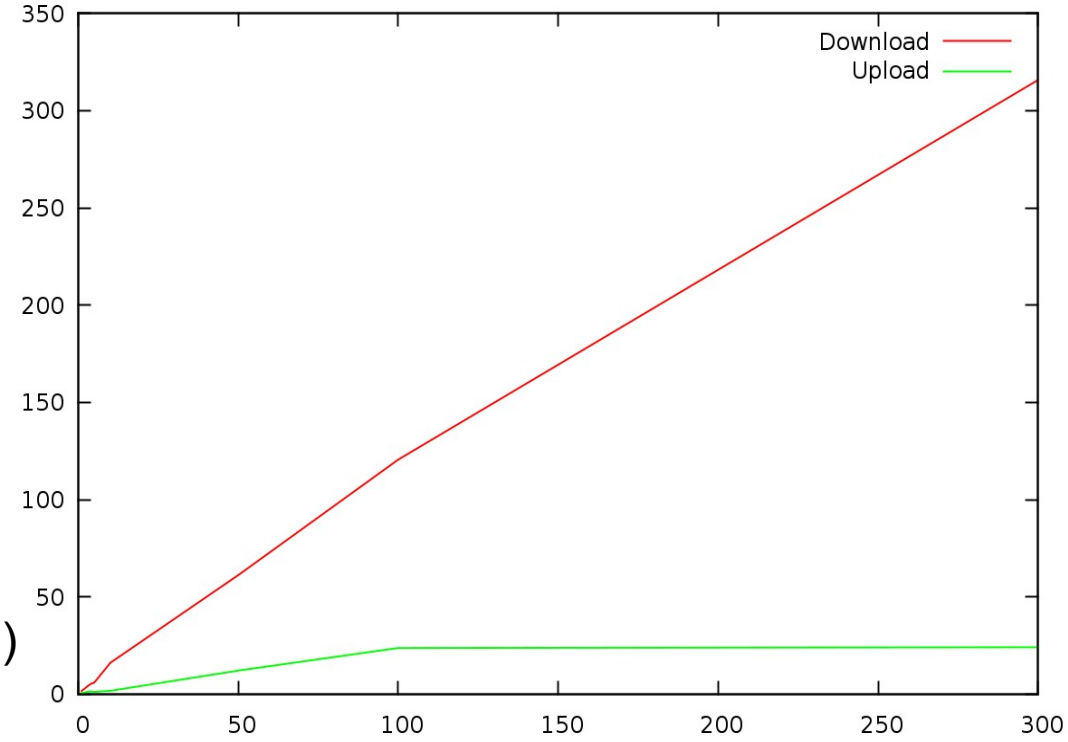


Manager RAM Usage/Clients (5min average)

Gestione Dinamica delle VM (Prototipo – Test Scalabilità 2)

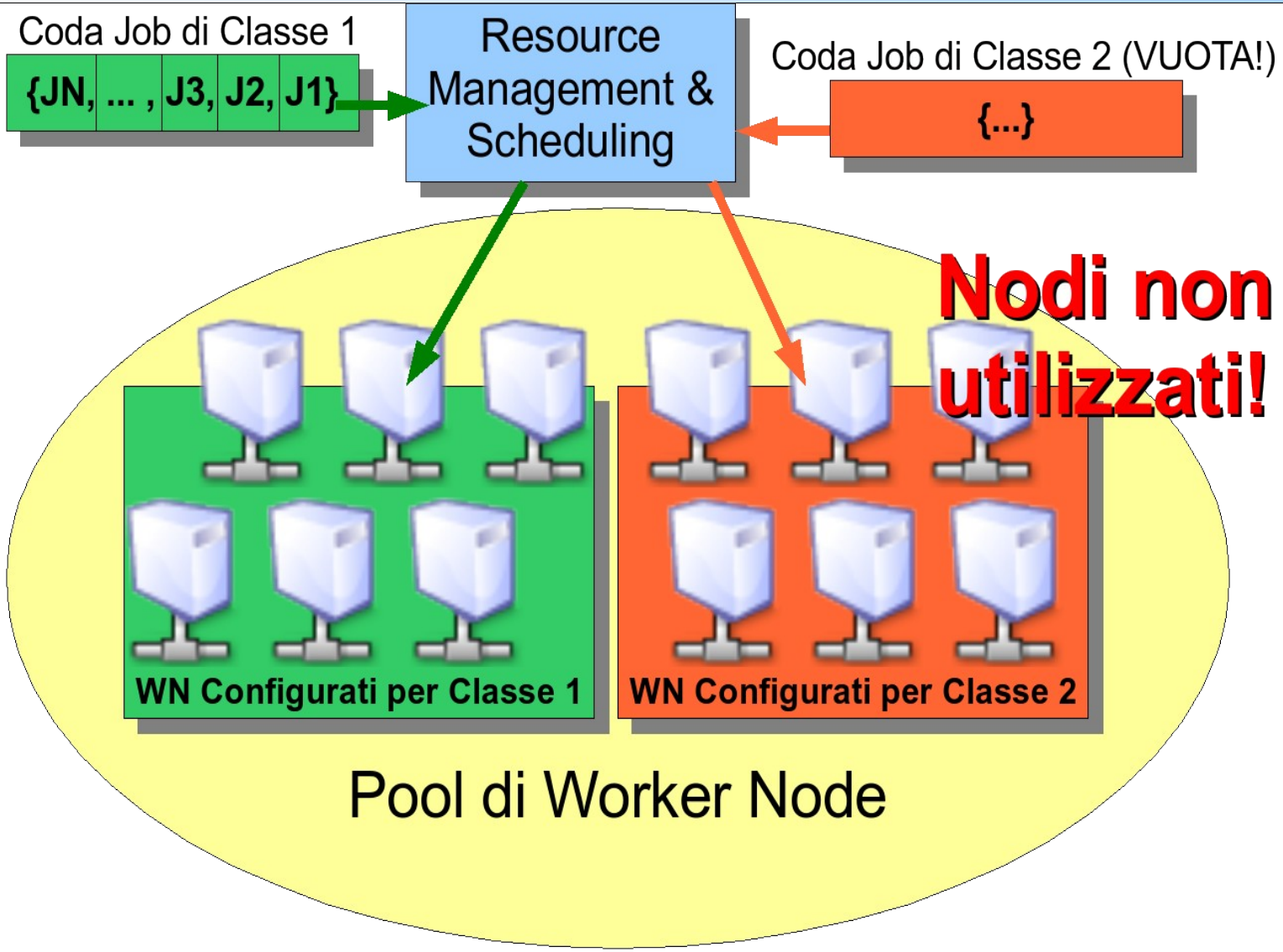


Instant Net Usage (100 Clients)



Manager Net Usage/Clients (5min average)

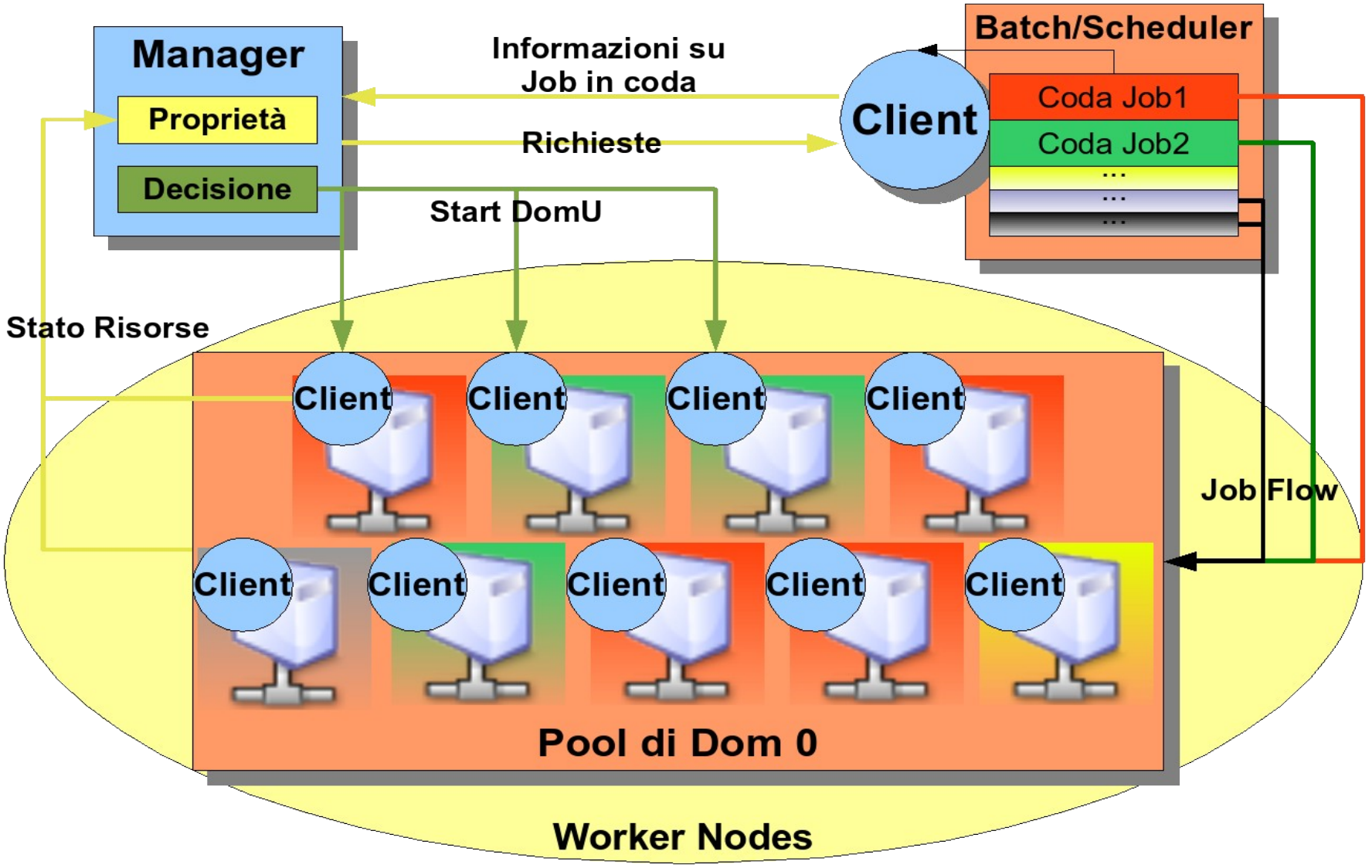
Gestione Dinamica delle VM (Esempio – Batch System)



Gestione Dinamica delle VM (Esempio – Batch System)

- Casi di Mutua Incompatibilità degli Ambienti:
  - Utenza locale necessita di software di tipo ingegneristico distribuito solo per Debian (caso reale).
  - Incompatibilità tra il software in uso e nuovi Sistemi Operativi (Aggiornamento SLC3 → SLC4).

Gestione Dinamica delle VM (Esempio – Batch System)



Gestione Dinamica delle VM (Esempio – Alta Affidabiltà)

- Precedenti esperimenti in Sezione hanno prodotto risultati riguardanti l'Alta Affidabilità.
- I Meccanismi di Gestione Dinamica delle VM possono integrare ulteriormente questi aspetti.

Gestione Dinamica delle VM (Esempio – Alta Affidabiltà)

- Il Manager Conserva il *Timestamp* dell'ultimo messaggio ricevuto da ogni Client.
- Un Trigger configurato appositamente può eseguire automaticamente il Fail-Over di una VM se essa non risponde da troppo tempo.

```
[client_alive_check]
expr = { len(manager.groups[“mailers”].values()) != 0}
cmd = {
        from time import time
        for c in manager.groups[“mailers”].values():
            d = time() - c.active_timestamp
            if (d > 120):
                print c.name, 'not active for more than 2 min!', d
        }
ticks =  -1
on_update_check = yes
```

Gestione Dinamica delle VM (Conclusioni & Sviluppi Futuri)

- Altri test sono in programma per verificare la capacità di gestione degli Ambienti Virtuali e le Politiche di Decisione.
- Verranno definite Metriche di Valutazione per stabilire l'impatto sulle code nei batch system (Testbed già realizzato).
- Un prototipo completo verrà implementato nel sito INFNGrid di Perugia per l'inizio del 2009.

Conclusioni

- Sono state mostrate le tecniche sviluppate sfruttando la Paravirtualizzazione (Xen):
  - DomU Diskless
  - DomU su Storage Centralizzato
  - Dom0 su NFS
  - Integrazione con la Rete Dipartimentale (VLAN)
  - Gestione Dinamica delle VM

Conclusioni

- Attraverso le varie soluzioni è già possibile usufruire di un ambiente flessibile e che si adatta ad essere ulteriormente espanso. Nel prossimo futuro si prevede:

- Consolidamento e Testing dell'Infrastruttura attuale
- Realizzazione di Dom0 con rootFS is RAM
- Realizzazione e Test di DomU su Filesystem Distribuiti
- Lo studio di applicabilità di programmi già esistenti (*Ganeti* by *google.code*)

Contatti

- Leonello Servoli - leonello.servoli@pg.infn.it
- Mirko Mariotti - mirko.mariotti@fisica.unipg.it
- Riccardo M. Cefalà - riccardo.cefala@pg.infn.it

Links

INFN-Grid Perugia

- http://grid.pg.infn.it

Gentoo Catalyst

- http://www.gentoo.org/proj/en/releng/catalyst/
- Xen Wiki HowTos
- http://wiki.xensource.com/xenwiki/HowTos

Xen

- http://www.xen.org