

tesi di laurea

# **Un sistema per l'emulazione delle reti su cluster di macchine virtuali**

Anno Accademico 2007/2008

relatore

Ch.mo prof. Roberto Canonico

correlatore

ing. Pasquale Di Gennaro

candidato

Roberto Bifulco

Matr. 885/269



# Contesto e contributo

*Le soluzioni per la valutazione ed il testing di protocolli ed applicazioni di rete:*

- **TestBed:** *ricrea in laboratorio l'ambiente operativo;*
- **Simulazione:** *adopera un modello concettuale rappresentativo della realtà;*
- **Emulazione:** *combina elementi sintetici con elementi reali.*

*Approccio cluster-based per l'emulazione di rete:*

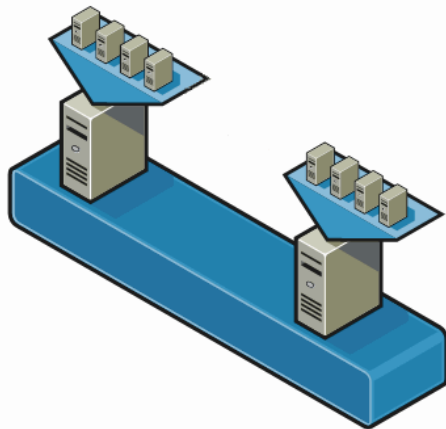
- Una infrastruttura omogenea facilmente gestibile
- Condivisione dell'infrastruttura fra più applicazioni
- Parallelismo degli esperimenti di emulazione

**Contributo:** approfondimento dell'analisi concettuale ed implementazione del sistema *software* per la realizzazione di un emulatore di rete *cluster-based* nell'ambito del progetto **Neptune**.

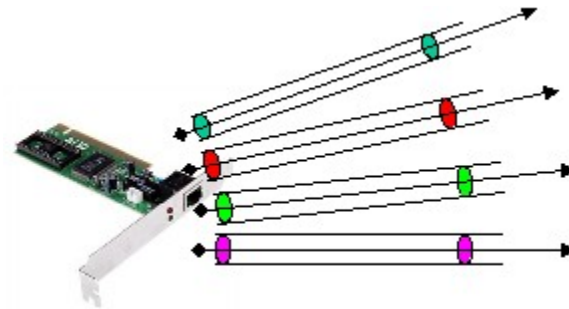
## La gestione delle risorse

**La soluzione ideale:** avere un meccanismo *on demand*, in cui le risorse vengono allocate all'esperimento secondo le effettive esigenze.

**Come realizzarla:** *node e link multiplexing*.



*Node multiplexing*

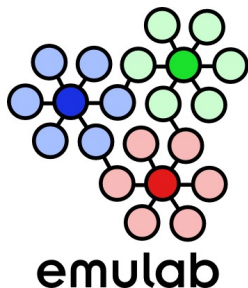


*Link multiplexing*



**Il progetto:** realizzare un emulatore di rete velocemente configurabile, multi-utente, accessibile da remoto (interfaccia web user-friendly), orientato al *multiplexing* delle risorse, che adopera l'approccio *cluster-based*.

### ***Progetti analoghi:***

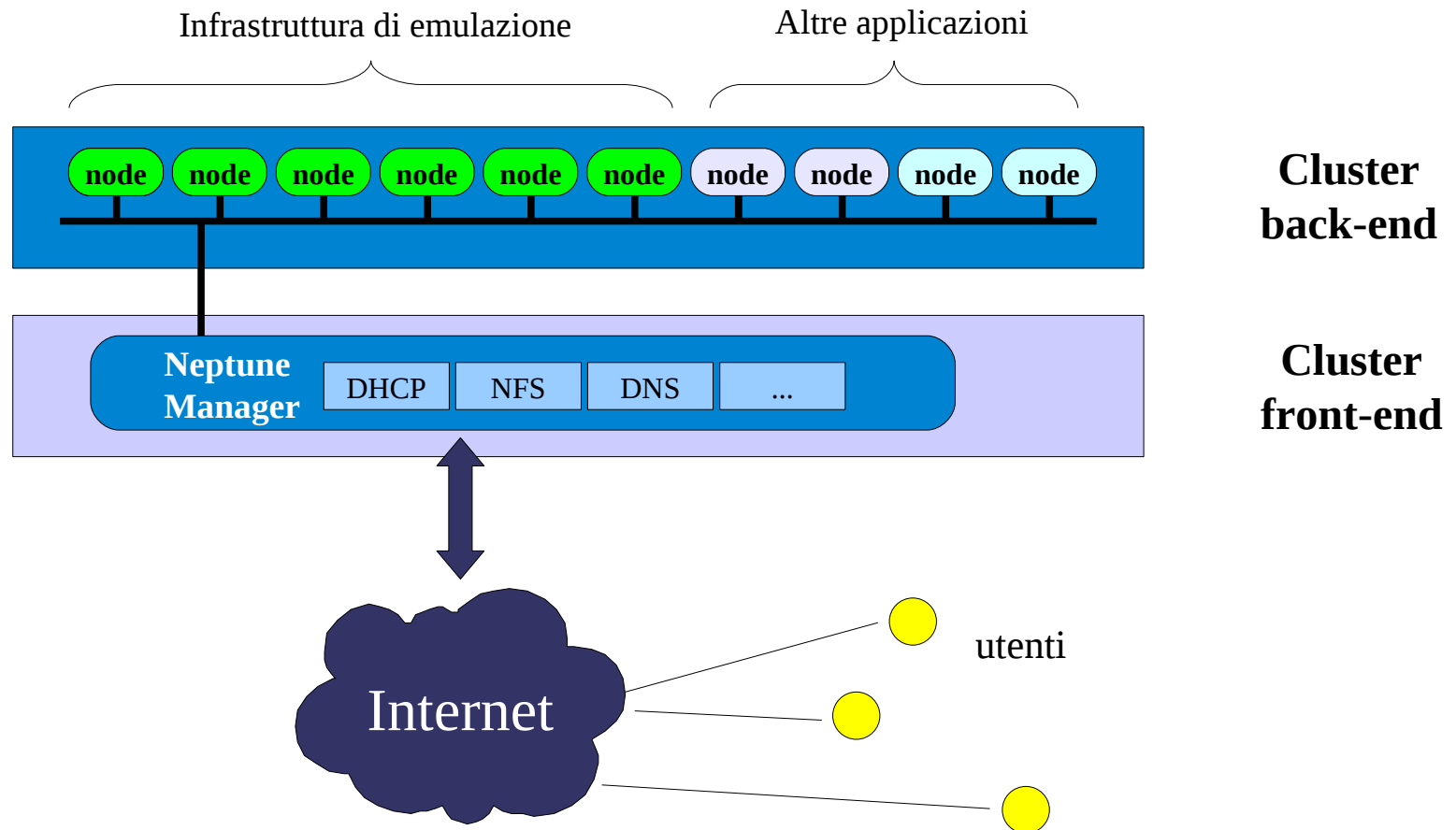


#### ***Differenze:***

***Neptune:*** adopera *hardware general purpose* e realizza dei *cluster virtuali* per supportare gli esperimenti.

***Emulab:*** adopera *hardware specializzato* ed il supporto alla virtualizzazione è limitato (Solo recentemente è stato presentato uno studio per estendere tale supporto: *Large-scale Virtualization in the Emulab Network Testbed*, M.Hiblet, R. Ricci, L. Stoller et al., Giugno 2008).

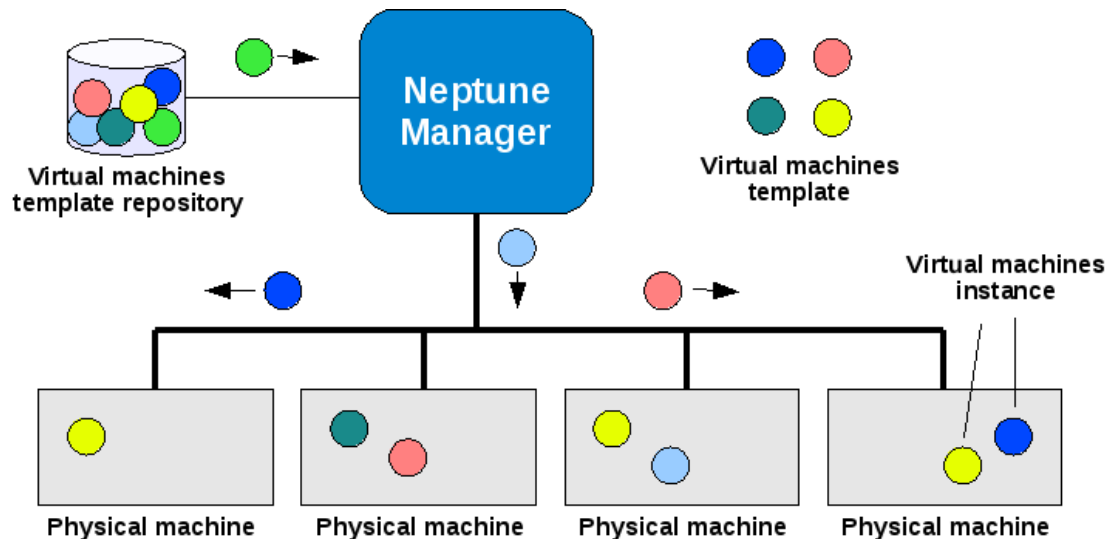
# L'organizzazione di Neptune



## Node multiplexing in *Neptune*

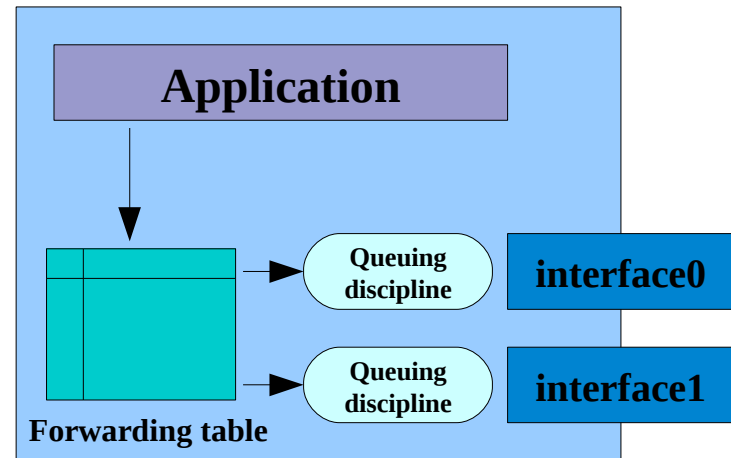
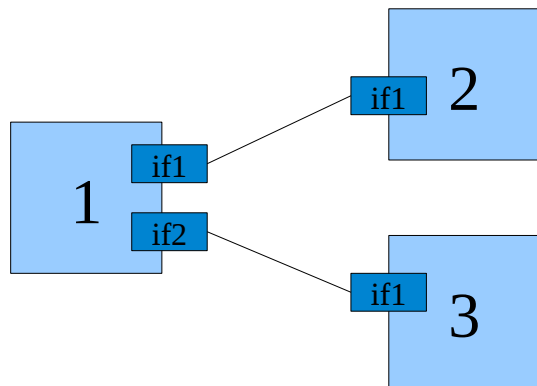
Per realizzare il *node multiplexing* Neptune adopera la virtualizzazione tramite la tecnologia **Xen** (paravirtualization). In futuro è previsto il supporto ad ulteriori tecnologie.

Approccio basato su *templating* per la realizzazione di cluster virtuali:



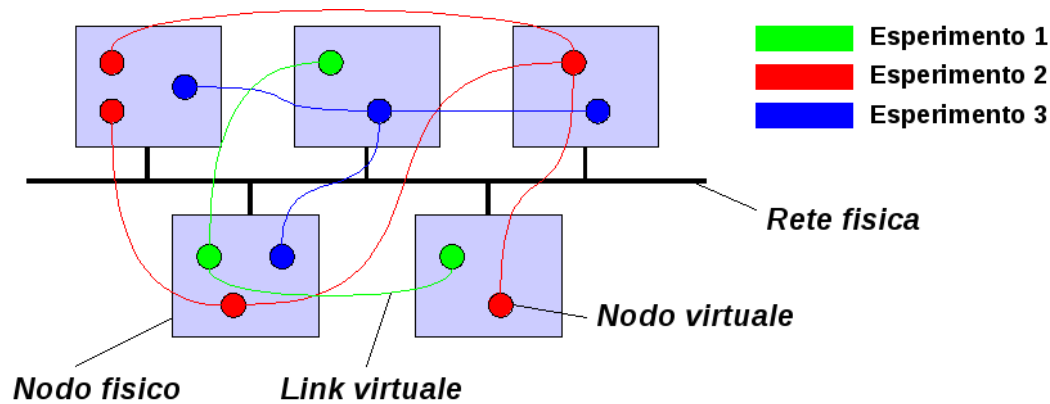
## Link multiplexing in *Neptune*

La tecnica adoperata: *one link per interface*



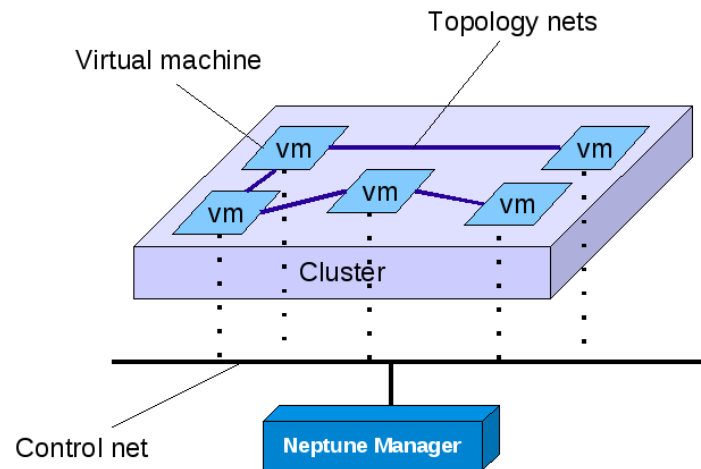
- Virtualizzazione delle interfacce di rete
- Utilizzo di *netem* e *traffic control* per la realizzazione delle caratteristiche del link (*bandwidth, delay, jitter, loss, ...*).

# Esempio di funzionamento:

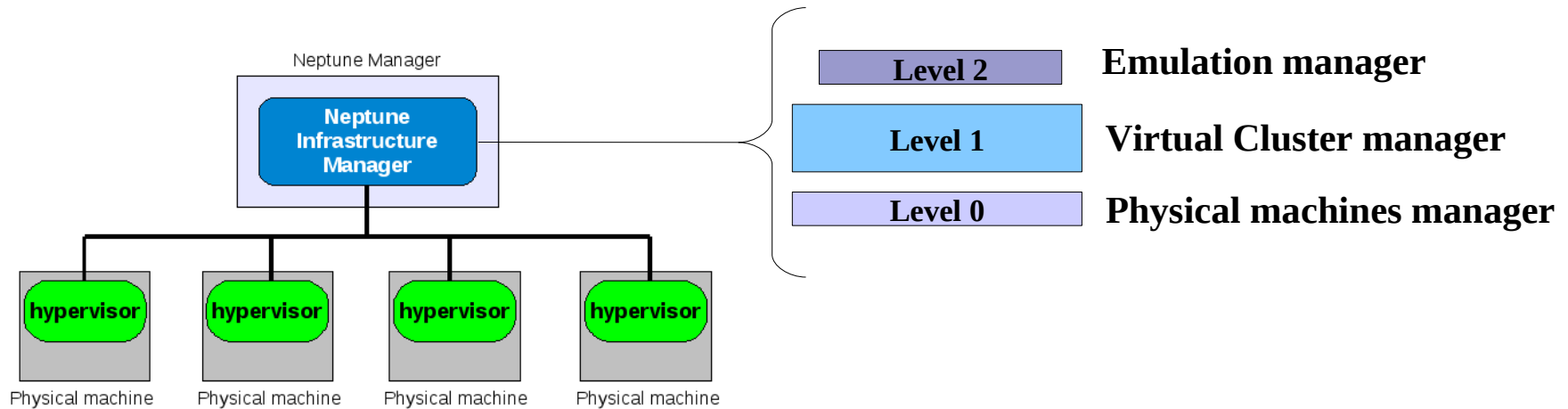


## Come accedo ai nodi virtuali?

Ciascun nodo virtuale è collegato “in modo trasparente” direttamente alla rete del cluster che assume il ruolo di “rete di controllo”



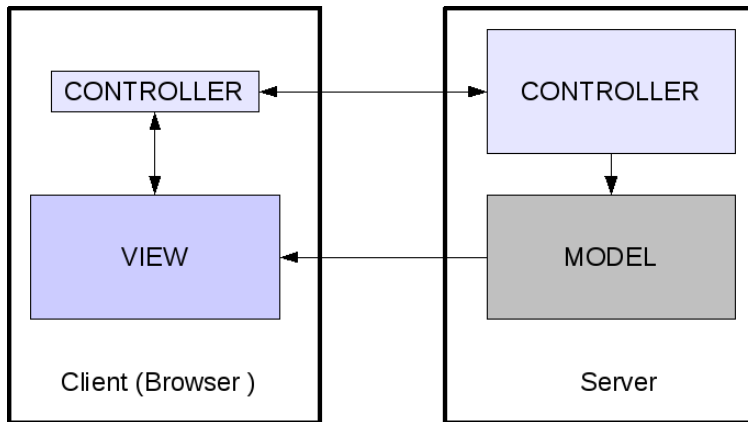
## L'applicazione realizzata



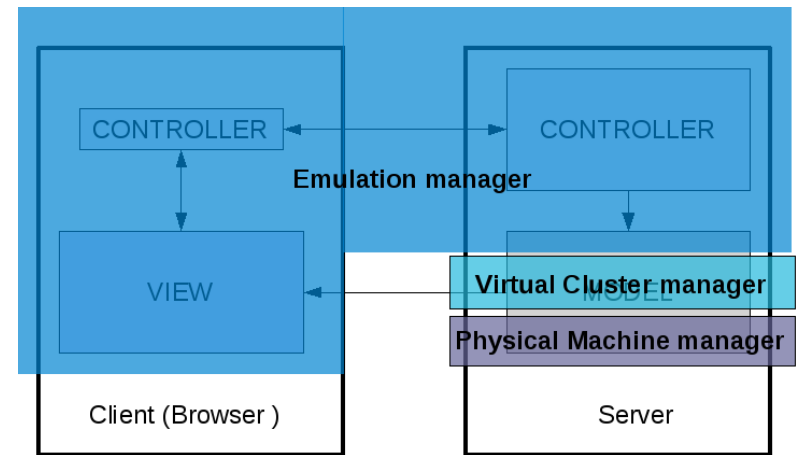
### *I livelli concettuali del Neptune Infrastructure Manager:*

- **Physical machines manager:** si occupa di gestire le macchine fisiche che costituiscono il cluster
- **Virtual Cluster manager:** crea e gestisce insiemi di macchine virtuali (virtual cluster)
- **Emulation manager:** introduce le funzionalità per l'emulazione, la gestione multi-utente e l'interfaccia grafica

## L'approccio MVC per realizzare l'interfaccia web



Distribuzione dei componenti fra client e server

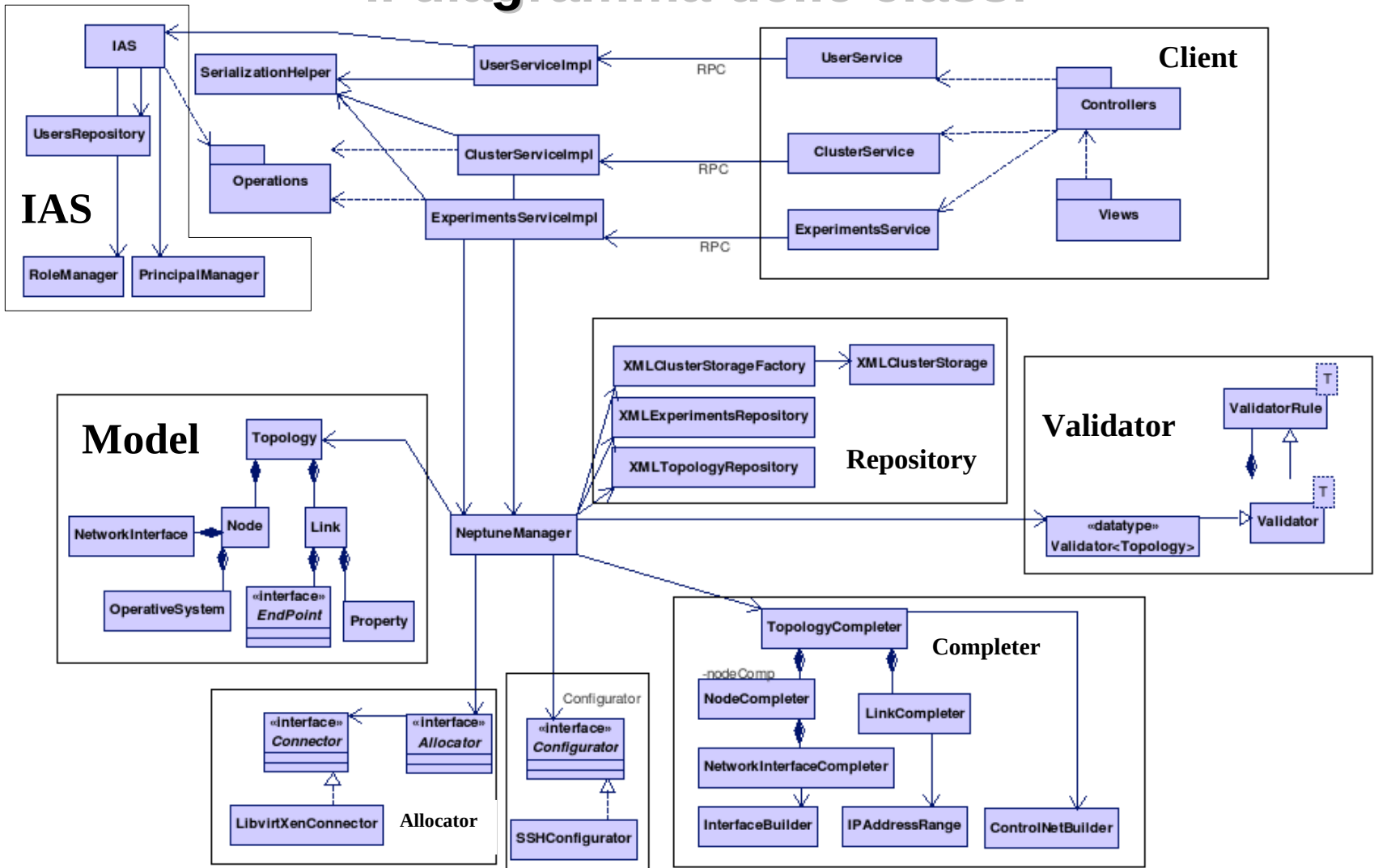


Corrispondenza con i livelli concettuali

### La tecnologia adoperata:

- **Java:** per la realizzazione delle funzionalità di Neptune
- **Servlet:** per l'esposizione dei servizi di Neptune tramite web
- **Google Web Toolkit:** per la realizzazione dell'interfaccia grafica *web-based AJAX*
- **Libvirt:** per l'accesso alle funzionalità degli *hypervisor*

# Il diagramma delle classi



## Il concetto di esperimento

Gli utenti possono creare esperimenti di emulazione che sono caratterizzati da:

- Identificativo
- Un utente che gestisce l'esperimento
- Risorse (La topologia di rete da emulare)

La topologia è un *insieme di nodi e link* che collegano fra loro tali nodi, ciascuno con opportune caratteristiche associate.  
La topologia è descritta in formato XML.

Gli utenti decidono l'allocazione e la deallocazione degli esperimenti da loro gestiti (previa autorizzazione dell'amministratore di sistema).

# La descrizione XML della topologia

## Descrizione di un nodo

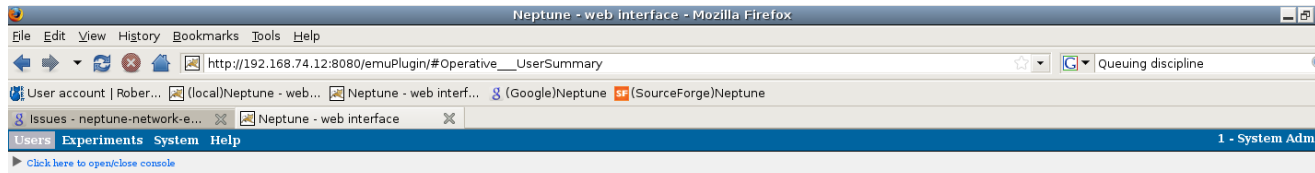
```
-<node id="node0">
  <description/>
  <cpu clock="0" name="" type=""/>
  <ram dimension="128"/>
  <hd dimension="2.0"/>
  -<network_interfaces>
    -<interface MAC="3E:DD:47:81:44:9D" name="eth0" type="">
      <interface_binding type="HIF"/>
      <ip address="192.168.2.108" subnetMask="255.255.255.0" type="control" version="4"/>
    </interface>
    -<interface MAC="9E:E1:A8:21:F5:D8" name="eth1" type="">
      <interface_binding type="HIF"/>
      <ip address="10.0.48.1" subnetMask="255.255.255.0" type="topology" version="4"/>
    </interface>
  </network_interfaces>
  <operative_system name="ubuntu" password="antoxps" username="root" version="810"/>
  -<software_configuration template="prova">
    <sshServer port="22"/>
    <software name="Apache Tomcat" type="Application Server" version="6.0"/>
  </software_configuration>
</node>
```

```
-<links>
  -<link id="link0" type="fullDuplex">
    -<end_point nodeId="node0" type="source">
      <multiplexing tech="OLPI"/>
      <interface name="eth1"/>
    </end_point>
    -<end_point nodeId="node1" type="destination">
      <multiplexing tech="OLPI"/>
      <interface name="eth1"/>
    </end_point>
    <property name="bandwidth" value="100kbps"/>
    <property name="delay" value="0ms"/>
    <property name="loss" value="0.0%"/>
    <property name="duplicate" value="0.0%"/>
    <property name="corrupt" value="0.0%"/>
  </link>
```

## Descrizione di un link

# L'applicazione in funzione: gestione utenti ed esperimenti

## Gestione degli utenti del sistema



| Username | Role         |
|----------|--------------|
| y        | User         |
| w        | User         |
| user     | User         |
| u        | User         |
| t        | User         |
| r        | User         |
| g        | User         |
| nasquale | System Admin |
| b        | User         |
| a        | User         |

Displaying 1 - 10 of 24

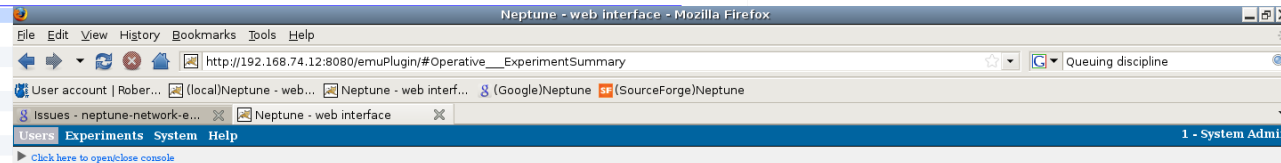


Il pannello utenti e' accessibile soltanto ai Syst registrati al sistema. Le informazioni mostrate Cliccando sul nome della colonna della tabella invertire il verso di ordinamento.

Cliccando sul nome di un utente si puo' accede

Per aggiungere un utente e' sufficiente accede Allo stesso modo, e' possibile eliminare un ute nome utente da eliminare, non verra' richiesta ulteriore confer

Se questo e' il primo accesso eseguito, si consiglia di creare un



| Experiment | Administrator | State     | Creation Date |
|------------|---------------|-----------|---------------|
| exp1       | 1             | UNDEFINED | 1227271633168 |
| exp2       | 1             | UNDEFINED | 1227271628888 |
| luigi2     | luigi         | DEFINED   | 1226890441731 |

Displaying 1 - 2 of 2

Il pannello esperimenti e' consente di accedere all'elenco degli esperimenti disponibili. Gli esperimenti visualizzati sono tutti quelli per i quali si hanno i diritti necessari all'accesso. Cliccando sul nome della colonna della tabella, e' possibile cambiare l'attributo rispetto al quale i nomi sono ordinati ed invertire il verso di ordinamento.

Cliccando sul nome di un esperimento si accede alla schermata dei dettagli dell'esperimento, da dove e' possibile gestire tutte le attivita' ad esso collegate.

Per creare un nuovo esperimento si puo' utilizzare la voce "New" del menu' "Experiment" presente nella parte alta dello schermo.



## Gestione degli esperimenti

# L'applicazione in funzione: Esperimento (1)

From this page you can edit the experiment's topology. To define or modify the topology, use one of the next methods. For new users, the topology editor is the simplest way to start. Note that you can move between experiment's tab without losing unsaved informations on the topology definitions, while if you exit this experiment view, all unsaved infos will be lost.

### Topology Editor

The topology editor provides a simple way to define or modify a topology. Using the topology tree showed on the left side of the editor, every topology component can be modified. When the topology is completed, it can be uploaded on the server for validation and saving. Please note that only a valid topology is actually saved on the server.

The topology editor helps the definition of the topology automatically generating nodes, network interfaces and links names. Moreover, it defines for every network interface a MAC address (editable by the user). It's also possible to define IP addresses for network interfaces, but this feature is intended for advanced users, from expert users. If you don't know how to assign a correct IP address to respect the topology correctness just don't define ip addresses: the system automatically generates any address.

**Topology**

- Nodes
  - WebServer
    - Interfaces
      - eth0 (control)
      - eth1
      - eth2
      - eth3
    - Softwares
  - WebClient0
  - DBMS
  - AdvClient
  - WebClient1
- Links
  - webLink [WebServer - WebClient0]
  - advClient-DBMS [AdvClient - DBMS]

```
graph LR;
  WC0[WebClient0] --- WS[WebServer];
  WC1[WebClient1] --- WS;
  WS --- DBMS[DBMS];
  DBMS --- AdvClient[AdvClient];
```

**Definizione della topologia di un esperimento tramite editor assistito:** la topologia può essere anche definita in modo soltanto concettuale, senza specificare parametri specifici come indirizzi IP o MAC che verranno generati automaticamente dal sistema.

## L'applicazione in funzione: Esperimento (2)

**Node Properties**

Node id:

Description:

▶ CPU

RAM (MBs):

HardDisk (GBs):

▼ Operative System

Name:

Version:

Username:

Password:

Software template:

Ok Cancel

**Link Properties**

Link id:

FullDuplex:

▼ Source

Node id:

Interface name:

▶ Destination

Bandwidth:

Delay:

Loss rate:

Duplication:

Corruption:

Link properties can be specified with more complete options using the [netem syntax](#)

Ok Cancel

# L'applicazione in funzione: Esperimento (3)

Gestione e visualizzazione dello stato dell'esperimento

usrExp  
Experiment Admin: user

**Allocation Map**

| Topology nodes | Cluster machines |
|----------------|------------------|
| node4          | chione           |
|                | node0            |
|                | neraide          |
|                | node1            |
|                | node2            |
|                | calice           |
|                | node3            |

Definizione manuale dell'allocazione dei nodi dell'esperimento

From this page you can manage experiment's active nodes. Active nodes are the experiment topology's nodes actually allocated.

| Node ID    | Status  | Control Address |
|------------|---------|-----------------|
| WebClient1 | PAUSED  | 192.168.2.106   |
| WebServer  | RUNNING | 192.168.2.102   |
| WebClientD | RUNNING | 192.168.2.103   |
| DBMS       | RUNNING | 192.168.2.104   |
| AdvClient  | RUNNING | 192.168.2.105   |

Press CTRL to select multiple nodes

Gestione dei nodi attivi dell'esperimento

## Conclusioni e sviluppi futuri

### ***Neptune*** oggi:

- Gestione multi-utente con sistema *Role Based Access Control (RBAC)*
- Gestione del ciclo di vita degli esperimenti (definizione, allocazione, distruzione)
- Gestione degli *hypervisor* (Xen) e delle macchine virtuali
- Definizione delle topologie tramite XML o tramite editor visuale
- Configurazione automatica ed *enforcement* delle topologie di rete

### ***Neptune*** domani:

- Progettazione di algoritmi per la (ri)allocazione dinamica delle risorse
- Implementazione della migrazione dei nodi virtuali
- Supporto a differenti tecnologie di virtualizzazione

**Neptune è un progetto *open source* distribuito secondo licenza GPLv3. Informazioni e risorse collegate a Neptune sono disponibili su Google Code: <http://code.google.com/p/neptune-network-emulator/>**