

Web Apps

Diego Casella

February 24, 2015



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

- 1 Introduzione
- 2 Remote Desktop
- 3 Remote Desktop - Fullscreen Application
- 4 Web Application
 - Wicket
 - Java Applet
 - Java Web Start
- 5 Conclusioni

Quando si vuole realizzare una applicazione che sia fruibile tramite web, slegata quindi da una installazione locale della stessa, le soluzioni percorribili sono sostanzialmente due:

- creare uno ambiente desktop remoto, nel quale ogni utente possa disporre di una propria home separata e riservata dove poter eseguire l'applicazione;
- eseguire l'applicazione direttamente nel browser;

Di seguito analizzeremo i vantaggi e gli svantaggi dei due approcci, confrontando le tecnologie più conosciute e diffuse.

Il Remote Desktop é una tecnologia attraverso la quale, come il nome stesso suggerisce, é possibile accedere ad una sessione desktop situata in una macchina remota.

Si tratta di una tecnologia che si presta bene nella situazione in cui:

- l'applicazione che si vuole remotizzare sia già esistente, e dunque il tempo/costi di sviluppo per convertirla in una applicazione in browser siano troppo alti;
- si abbia necessità di accedere al desktop remoto stesso, per eseguire operazioni di upload/download/modifica/creazione di files, o operazioni di versioning;

Il Remote Desktop viene realizzato attraverso una macchina remota, detta host, che contiene il software che si vuole rendere fruibile a terze parti, e che é configurata ad hoc per poter gestire sessioni desktop multiple. Per poter fornire un grado di protezione superiore spesso l'host viene settato per accettare solo connessioni criptate, attraverso le quali poi viene effettuato il tunnelling delle porte che interessano le effettive sessioni desktop.

Nel caso di sistema operativo GNU/Linux, uno dei programmi più diffusi per la creazione e gestione di desktop remoti é chiamato VNC. Esso é suddiviso in due applicativi differenti:

- un server, eseguito in background, che inizializza e gestisce le varie sessioni desktop;
- un client, sotto forma di applicazione standard, che consente di connettersi ad uno o più desktop remoti;

Dal momento che siamo interessati a capire come realizzare più sessioni desktop nel server remoto, ci focalizzeremo sul VNC server.

Per poter permettere ad un ipotetico utilizzatore *Bob* di accedere alla sessione desktop remota, l'amministratore del computer host deve:

- creare una nuova login e password per l'utente *Bob*;
- fermare la sessione VNC esistente, tramite il comando `service vncserver stop`;
- editare il file di configurazione situato in `/etc/sysconfig/vncservers` ;
- inserire una password per VNC riferita all'utente *Bob*, col comando `vncpasswd`;
- avviare una nuova sessione VNC, tramite il comando `service vncserver start`;

Esempio di file vncservers

```
VNCSERVERS="1:root 2:Fabio 3:Diego"  
VNCSERVERARGS[1]="-geometry 1600x900 -localhost"  
VNCSERVERARGS[2]="-geometry 640x480 -localhost"  
VNCSERVERARGS[3]="-localhost"
```

La coppia (*numero*, *utente*), separata dal carattere ":", indica in quale *DISPLAY* indirizzare la sessione desktop dell'utente associato. Inoltre tale numero, sommato al valore della porta 5900 usata di default da VNC, specifica attraverso quale porta é possibile connettersi a tale sessione desktop.

Esempio di file vncservers

```
VNCSERVERS="1:root 2:Fabio 3:Diego"  
VNCSERVERARGS[1]="-geometry 1600x900 -localhost"  
VNCSERVERARGS[2]="-geometry 640x480 -localhost"  
VNCSERVERARGS[3]="-localhost"
```

L'opzione *geometry* specifica la dimensione del desktop che verrà reso disponibile da VNC, tuttavia é un parametro che non é obbligatorio mettere in quanto l'utente può comunque cambiarne la risoluzione a piacimento.

Esempio di file vncservers

```
VNCSERVERS="1:root 2:Fabio 6:Bob"  
VNCSERVERARGS[1]="-geometry 1600x900 -localhost"  
VNCSERVERARGS[2]="-geometry 640x480 -localhost"  
VNCSERVERARGS[6]="-localhost"
```

L'opzione *localhost* invece istruisce VNC di ignorare le connessioni provenienti dalle interfacce di rete, accettando solo quelle che sono originate dal server stesso. Quest'ultima opzione é fondamentale in quanto obbliga chi vuole connettersi al server VNC di instaurare per prima cosa una connessione protetta al server stesso, e poi fare un tunnelling della sessione desktop interessata.

Ora che il setup del server é stato completato, vediamo come l'utente finale riesce a connettersi alla sua corrispettiva sessione desktop remota.

Dal momento che, come detto in precedenza, il server accetta solo connessioni provenienti dal server stesso, occorre per prima cosa instaurare una connessione protetta al server stesso, dal quale effettuare poi il tunnelling della porta interessata.

Remote Desktop - Connessione (2)



Per poter realizzare tale connessione e tunnelling, sotto ambiente Windows é possibile utilizzare il client *PuTTY*, specificando il server al quale volersi connettere, la login per l'autenticazione, ed il port forwarding.

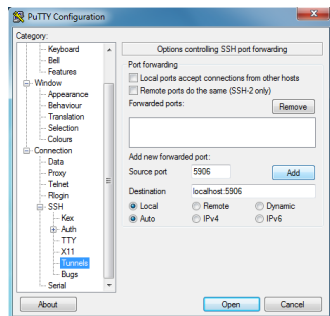
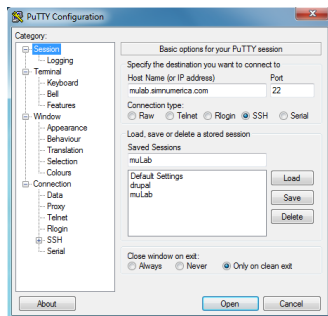


Figure: Esempio di configurazione di PuTTY

In ambienti Unix invece é sufficiente aprire un terminale e digitare il comando

```
ssh -L 5906:localhost:5906 -l Bob remote.host.com
```

In entrambi i casi comunque, per poter procedere occorre inserire la password associata al desktop remoto di Bob .

Remote Desktop - Connessione (4)



Giunti a questo punto occorre utilizzare un client VNC, come ad esempio TightVNC, per potersi connettere al desktop remoto.

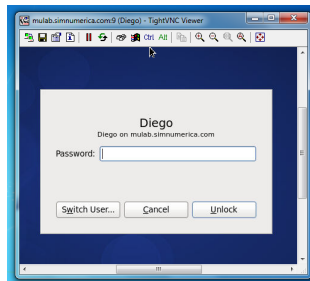
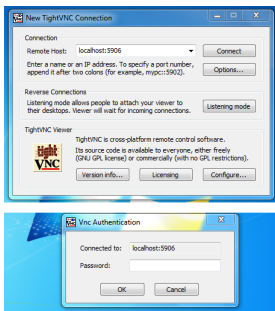


Figure: Esempio di connessione tramite TightVNC

- accesso al proprio spazio di lavoro da qualsiasi pc con una connessione internet;
- l'utente finale non deve preoccuparsi di effettuare upgrade del software;
- chiudendo VNC viewer, la sessione remota rimane persistente: si può dunque lanciare il software per eseguire calcoli laboriosi, e monitorarlo di tanto in tanto;
- possibilità di utilizzare altri programmi presenti nell'host, come ad esempio toolchain di sviluppo, programmi di creazione di report (openoffice), ecc... ;

- sono necessari l'uso di due software specifici per potersi connettere remotamente;
- é inoltre necessario inserire 3 password, di cui almeno 2 obbligatorie;
- se si esegue il logout nella sessione desktop remota, non é più possibile effettuare ulteriori login finché l' amministratore non riavvia il VNC server;
- i tempi di risposta dipendono molto dalla qualità della connessione a disposizione;

Un caso particolare del Remote Desktop é rappresentato dalla possibilità di configurare ed utilizzare un window manager apposito, che faccia apparire solamente l'applicazione di interesse in modalità fullscreen, disabilitando tutto ciò che riguarda taskbar, menu di sistema, analogamente a quanto accade negli ATM o nei terminali per comprare i biglietti ferroviari.

Se da un lato questa soluzione permette di impedire che l'utente esegua involontariamente un logout dalla sua sessione, d'altro lato gli impedisce di usare qualsiasi altro programma che possa tornargli utile per il suo workflow. Quindi sarà necessario che questi tool vengano inclusi nell'applicativo principale, con notevole dispendio di tempo.

Un'altra tipologia di tecnologia per rendere disponibile un'applicazione da più locazioni differenti, senza dover obbligare l'utente ad installare e configurare tale software, é rappresentata dalla categoria delle *Web Application*. Per *Web Application* si intende un gruppo molto esteso ed eterogeneo di applicazioni che, per funzionare, necessitano semplicemente di un browser web.

Apache Wicket é la prima tipologia di framework per realizzare applicazioni web: si tratta di un framework scritto in Java per lo sviluppo di siti web dinamici/web applications in cui la generazione delle pagine web viene demandata ad un servlet Java.

In Wicket viene usata una sintassi XHTML standard per creare i template delle pagine web. In essi, sono presenti dei tag con una proprietà speciale denominata "*wicket:id*": essa rappresenta la congiunzione fra l'elemento della pagina web, e il componente Java ad esso associato, che diventa responsabile dell'output nella pagina web finale dello stesso. La pagina infine è semplicemente il componente di più alto livello, accoppiato ad esattamente un template XHTML, dal quale dipartono i vari componenti figli.

Esempio di Applicazione Wicket: HelloWorld.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN"  
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">  
<html xmlns="http://www.w3.org/1999/xhtml"  
    xmlns:wicket="http://wicket.apache.org/dtds/data/wicket-xhtml1.3-  
strict.dtd">  
<body>  
    <span wicket:id="message" id="message">Errore nella  
visualizzazione del messaggio</span>  
</body>  
</html>
```

Esempio di Applicazione Wicket: HelloWorld.java

```
package org.test.wicket;  
import org.apache.wicket.markup.html.WebPage;  
import org.apache.wicket.markup.html.basic.Label;  
public class HelloWorld extends WebPage {  
    public HelloWorld() {  
        add(new Label("message", "Hello World!"));  
    }  
}
```

- netta separazione fra core e presentazione;
- uso di linguaggio di alto livello per realizzare pagine web (PHP é piú ostico);
- possibilità di ridisegnare l'interfaccia senza toccare la logica di controllo;
- il codice risiede tutto nel web server;

- sintassi elaborata per realizzare anche semplici layout di elementi;
- il server deve incaricarsi di gestire le sessioni utente, a differenza dei Remote Desktop;
- tutte le utility presenti in una sessione desktop devono essere reimplementate nel software;
- se il servlet Java é sotto carico di lavoro elevato o crasha, tutti gli utenti ne risentono;

Un'altra tecnologia per usufruire di applicativi web tramite browser é rappresentata dalle Applet Java, ovvero bytecode Java che viene caricato dal browser web ed interpretato non appena l'utente visita la pagina contenente tale applet.

Sostanzialmente, l'entry point é una istanza della classe JApplet (o sue sottoclassi), che viene eseguita dal browser stesso.

Esempio di Applicazione Wicket: HelloWorld.java

```
package org.test.japplet;  
import javax.swing.JApplet;  
import javax.swing.JLabel;  
public class HelloWorld extends JApplet {  
    public HelloWorld() {  
        add(new JLabel(" Hello World!"));  
    }  
}
```

- uso di linguaggio di alto livello per realizzare una applicazione web complessa;
- sgrava il server dall'onere delle computazioni, che vengono eseguite nel browser dell'utente;

- il bytecode di ogni singola classe viene salvato nella cache del browser per essere eseguito, quindi applicazioni complesse richiederanno più tempo per l'avvio;
- non é possibile gestire le sessioni utente, a meno che non si utilizzi un servlet di appoggio (gestendone quindi la comunicazione, oltre al fatto che tale servlet deve gestire ogni utente);
- tutte le utility presenti in una sessione desktop devono essere reimplementate nel software;
- le sandbox dei browser limitano le funzionalità solitamente presenti (lettura/scrittura su file ad esempio, memoria disponibile);

Java Web Start é un altro framework per la creazione di Web Apps, che differisce leggermente dai due framework visti in precedenza. Java si promuove come linguaggio "*write once, run everywhere*" ma il problema é che l'installazione dell'applicazione, con annesse dipendenze ed upgrade dell'applicazione stessa, risulta difficile da gestire per l'utente medio.

Per questa ragione, Sun ha creato il Java Web Start: l'utente finale deve solo installare tale framework una volta e, ad ogni utilizzo di un applicativo basato su Java Web Start, il framework stesso si occupa di tenerlo costantemente aggiornato, senza ulteriori interventi esterni.

L'applicativo Web Start viene scaricato nel computer quando l'utente, provvisto della runtime Web Start, clicca su un particolare link che la runtime stessa identifica. Si avvia così il processo di download dell'applicazione e dei componenti necessari al funzionamento della stessa; da quel punto in poi, l'app sarà disponibile direttamente dal pc dell'utente senza l'utilizzo di un browser come intermediario.

- uso di linguaggio di alto livello per realizzare una applicazione web complessa;
- sgrava il server dall'onere delle computazioni, che vengono eseguite nel pc dell'utente;
- intero applicativo disponibile nel computer dell'utente;
- funzionamento anche in assenza di connessione ad internet;

- intero applicativo disponibile nel computer dell'utente;

