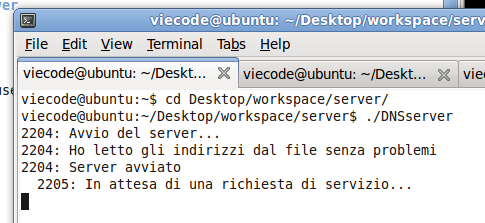
Funzionamento senza failure

Il flusso di eventi che caratterizzano il sistema secondo un funzionamento senza errori necessita di attenzione particolare verso le diverse fasi di scrittura dei file: un file modificato su un server replica deve essere inoltrato agli altri file-server della rete, oppure, uno stesso file modificato da due client distinti presso due file-server distinti deve, al termine delle operazioni, risultare identico su entrambe le copie. Verranno qui di seguito illustrati passo-passo entrambi i due casi.

# 1) Scrittura di un file e propagazione dell’aggiornamento

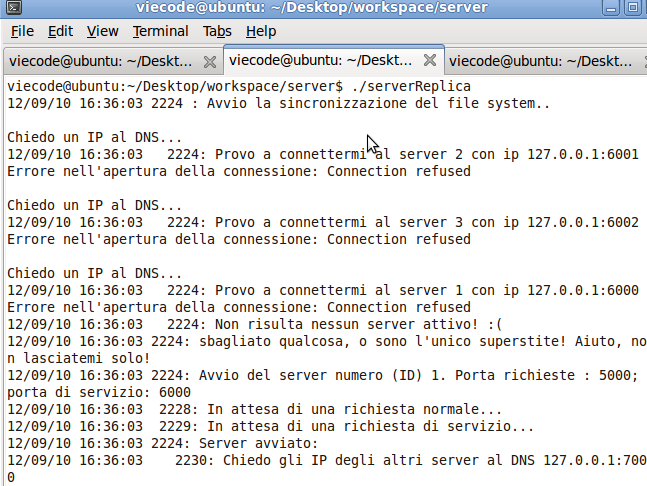
Per semplicità il sistema si compone di tre server, di cui uno viene lasciato spento, e due client interessati alla stessa risorsa.

Come primo passo viene avviato il server DNS

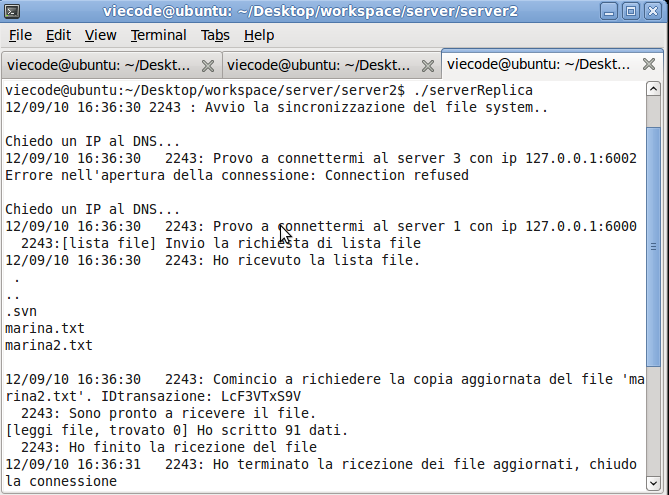


Vengono subito dopo avviati tre server replica, che nella cartella server dell’applicazione consegnata corrispondono agli eseguibili serverReplica con ID 1, localizzato nella cartella server,serverReplica, con ID 2, localizzato in nella sottocartellaserver2, e server Replica, con ID 3, localizzato nella sottocartella server3 Si ricorda che l’ID, il numero di porta di servizio e il numero di porta delle richieste dei client dei vari server, vengono letti in un file di configurazione durante la fase di avvio del server. Per cui questi valori possono essere modificati senza dover ricompilare l’intero progetto.

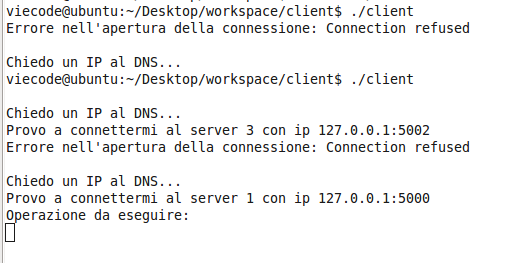
Il primo server lanciato prova a contattare gli altri server e non trova al momento alcun server attivo. E’ quindi l’unico server attivo nel sistema distribuito e non procede a sincronizzare il proprio file system con gli altri server durante la fase di avvio.



Il secondo server (ID 2), lanciato dopo il primo (ID 1), stabilisce una connessione con quest’ultimo, e procede a sincronizzare il proprio file system con il file system del server che ha contattato in modo da avere la stessa versione di tutti i file presenti nel file system distribuito. Nella figura in basso possiamo notare come, il server 2 non trovi il server 3. Difatti, ancora non si è proceduto all’avvio di tale server. Questa situazione, grazie alla gestione delle failure, non pone un problema in quanto il server 2 prosegue il proprio avvio cercando altri server attivi a cui chiedere l’aggiornamento del file system.

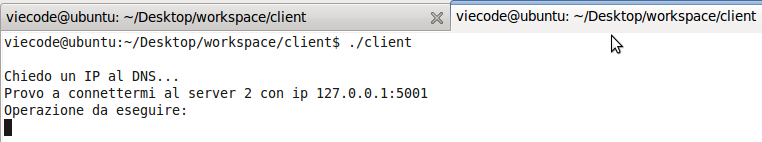


Viene fatto partire un client che si connette, dopo un primo tentativo fallimentare sul server con ID 3, spento, al server con ID 1.



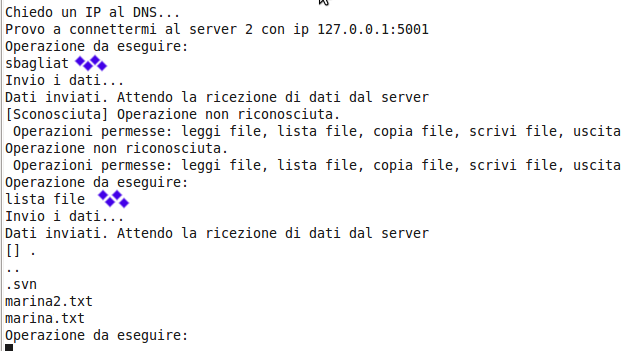
Da questo momento il server 1 farà da server primario per il client 1.

Viene intanto lanciato un secondo client che verrà utilizzato per la seconda dimostrazione. E’ possibile notare che il DNS, dopo aver inviato al client precedente (client 1) gli ID dei server 3 ed 1, continuerà l’assegnamento circolare, assegnando al secondo client il server con ID 2.

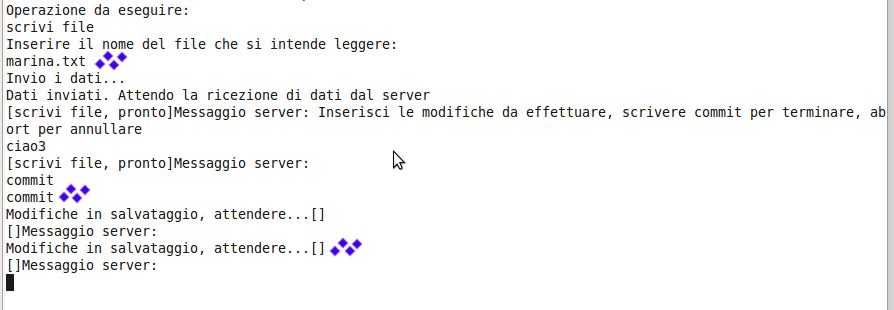


Tornando al client 1, questo chiederà al server ID 1 l’elenco dei file presenti, mediante il comando “lista file”.

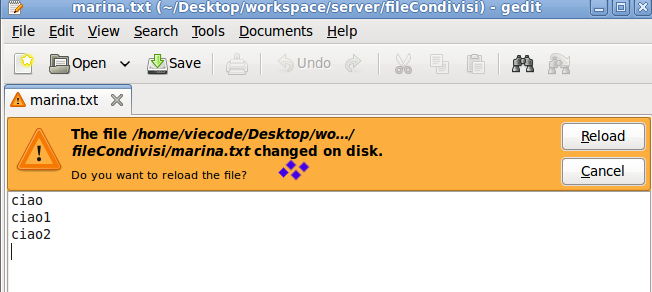
E’ stato volutamente digitato un comando errato (evidenziato in blu in figura) in modo da testare la gestione del server in caso di input incongruenti da parte del client. Il server risponde con la lista di comandi che è possibile digitare. In seguito alla specifica richiesta del client, viene stampata a video la lista di file testuali su cui è possibile effettuare modifiche.



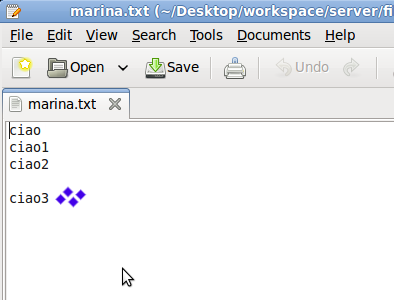
Si vuole scrivere del testo su un file. Il client digita adesso il comando “scrivi file”, a seguito del quale il server chiederà di specificare il nome del file da scrivere. Una volta specificato il nome e ricevuto l’ok dal server, viene inserito del testo di prova e, dopo aver dato invio, viene inserita la richiesta di salvare le modifiche, mediante il comando “commit” da parte del client.



Come è possibile notare con l’aiuto di un editor di testo tenuto nel frattampo aperto sulle risorse del server per comodità, viene indicato che che il file (presente sulla copia server con ID 1) è stato modificato.

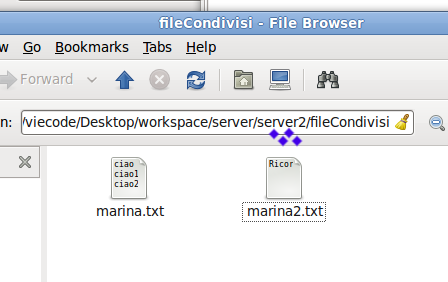


Cliccando su Reload, il file apparirà modificato come segue, mostrando al suo interno la nuova riga che prima era stata digitata dal client “ciao123”.



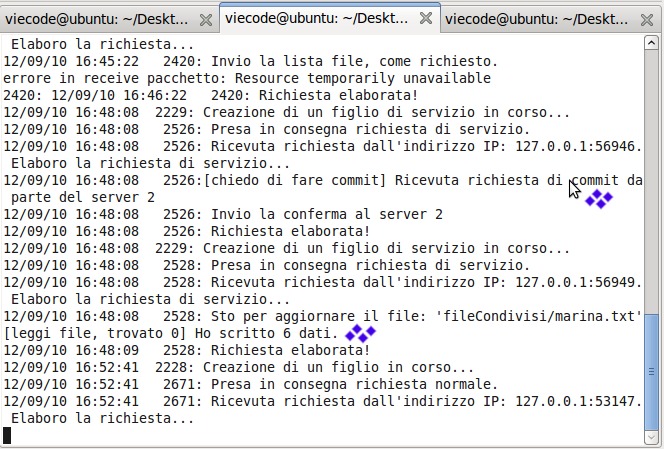
Le modifiche sono dunque state correttamente registrate sul server ID 1.

La nuova versione del file viene a questo punto propagata ai rimanenti server attivi nella rete, in questo caso il server ID 2. Andando ad esplorare manualmente la directory root del server 2, è infatti possibile notare la presenza del file “marina.txt” con le nuove stringhe inserite.

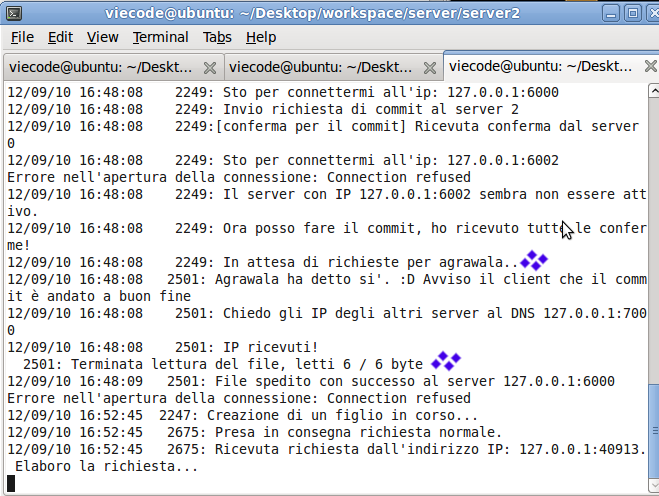


Per l’esecuzione di queste operazioni dal punto di vista dei server, vengono illustrati i passi che essi eseguono per registrare le scritture:

il server ID 1 prende in consegna la richiesta di commit del client ed avvia la richiesta di conferma con agrawala al server ID 2. Ricevuta la risposta, procede al salvataggio del file, ed in seguito alla propagazione dello stesso.



Il server ID 2, dalla sua parte, vede la richiesta, invia la risposta ed in seguito riceve il file aggiornato.

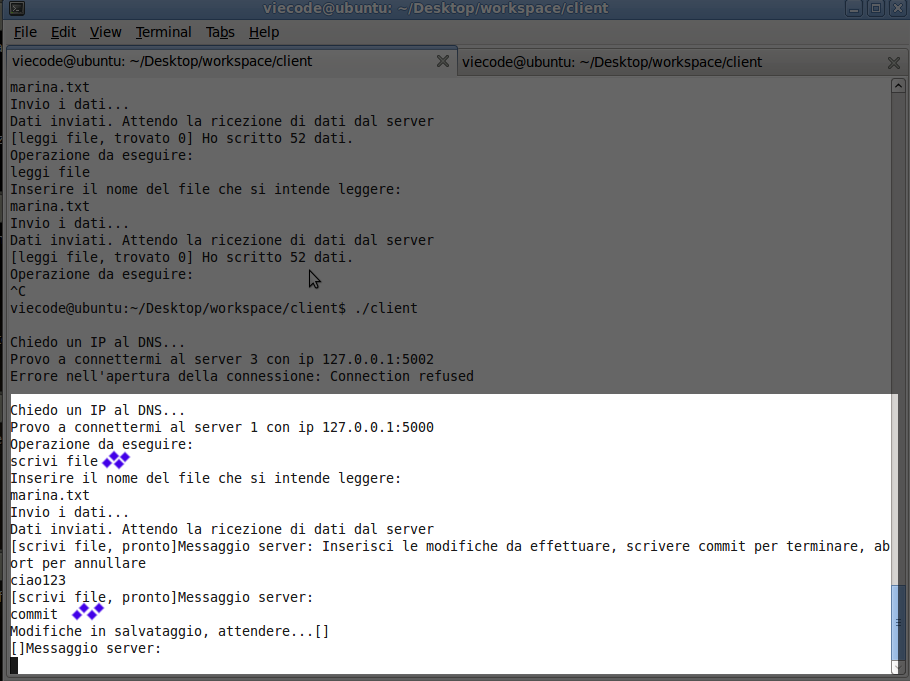


# 2) Gestione delle scritture concorrenti sullo stesso file

Nel caso di scrittura concorrente, viene innanzitutto svuotato il file “marina.txt”.

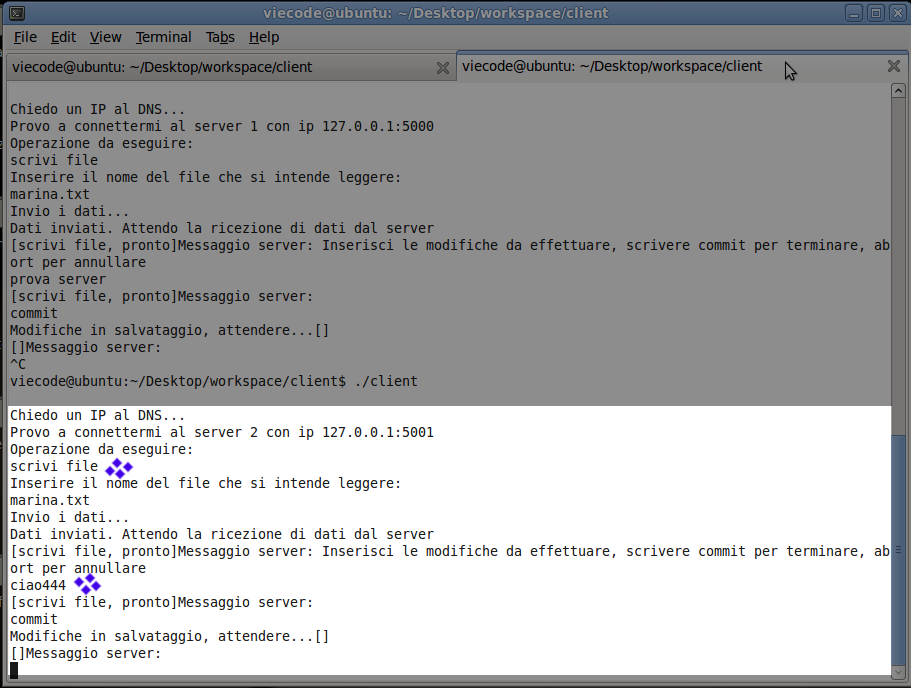
Il client 1, con le stesse operazioni descritte nel punto precedente, provvede dunque all’inserimento di una nuova stringa dentro al file, inizialmente senza lanciare il comando commit.

Viene digitato quindi il comando “scrivi file” ed inserita la stringa “ciao123”.

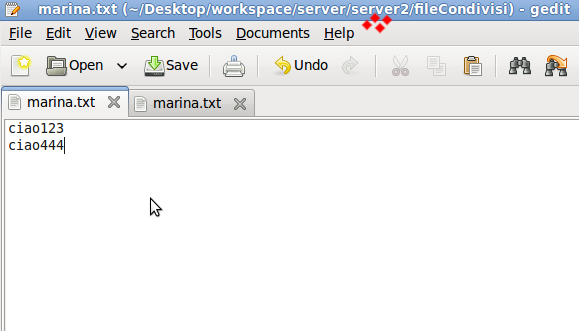


Un secondo client, richiede ad un secondo server l’apertura dello stesso file “marina.txt” digitando il comando “scrivi file” ed immettendo la stringa “ciao444”.

Nel frattempo il client 1 ha lanciato la richiesta di commit (figura in alto, seconda parte evidenziata in blu), ed ora anche il client 2 lancia la richiesta di commit, che andrà a buon fine in quanto il primo client ha già completato il suo lavoro.



Andando ad aprire il file su uno dei due server, esso conterrà infatti le stringhe nello stesso ordine sequenziale delle commit.



Sull’altro server il risultato sarà ovviamente lo stesso.

