Progetto di
Laboratorio di Programmazione di Rete
Corso di Laurea in Informatica Applicata
Anno Accademico 2002/2003

1 Scopo del Progetto

Si chiede di realizzare uno strumento che consenta ad un insieme di utenti che accedono alla rete da hosts diversi di *condividere* un insieme di files. La condivisione dei files avviene mediante la definizione di un *Name Server* centralizzato, che memorizza in una opportuna struttura dati *Naming* la corrispondenza tra il nome di ogni file condiviso e l'insieme di hosts che posseggono una copia di quel file. Per ogni diverso file, il *Name Server* registra al massimo *Nhmax* riferimenti ad hosts che possiedono quel file. I servizi offerti dallo strumento sono i seguenti :

- *Condivisione di un Nuovo File*: il client notifica al *Name Server* che intende mettere a disposizione un nuovo file, comunicando il nome del file. Il *Name Server* invia all'host un messaggio di avvenuta registrazione oppure un messaggio di errore, nel caso in cui esistano già *Nhmax* riferimenti a quel file.

- *Richiesta di un file*: il client richiede al *Name Server* un file, indicando il suo nome, oppure specificando un insieme di condizioni che consentono di individuare un insieme di files. Nel primo caso, se il nome esiste in *Naming*, si restituisce al client *l'elenco* di tutti gli hosts che posseggono quel file. Altrimenti, al client viene inviato un messaggio di errore. Nel secondo caso, si invia, per ogni file individuato, l'elenco degli hosts che posseggono quel file. Se non esiste alcun file che soddisfa alle condizioni, viene inviato un messaggio di errore.

- *Rimozione di un file*: il client notifica al *Name Server* che non intende più condividere un file *f*, di cui comunica il nome. Il *Name Server* elimina il nome dell'host su cui è in esecuzione il client, dalla lista dei nodi che mettono a disposizione *f*. Se quell'host era l'unico a mettere a disposizione *f*, la entrata relativa ad *f* viene eliminata da *Naming*. In entrambe i casi *Name Server* invia al client un messaggio di acknowledgment, oppure invia un messaggio di errore, nel caso in cui il nome del file da rimuovere non esista.

E' possibile che piu' hosts dichiarino di voler condividere files con lo stesso nome. In questo caso si suppone che il contenuto dei files sia *identico*.

2 Struttura del Sistema

Il sistema prevede la presenza di un *Name Server* centralizzato. E' necessario definire la struttura dati *Naming* che registra la corrispondenza tra nomi di files e riferimenti a tutti gli hosts che possiedono quel file. Per ogni file condiviso, *Naming* riporta un ulteriore valore *Reference* che approssima l'intervallo di tempo trascorso dall'ultima volta che quel file è stato richiesto da qualche utente. Il valore di *Reference* viene inizializzato a 0 quando viene inserito il primo riferimento al file e viene periodicamente incrementato dal *Name Server*. Ogni volta che un client richiede quel file, il valore di *Reference* viene riportato a 0. Quando il valore di *Reference* assume un valore *RefMax*, l'entrata relativa al file viene eliminata dalla struttura *Naming*. Ai client che avevano messo a disposizione il file viene comumicata l'avvenuta cancellazione.

E' necessario utilizzare il *meccanismo dei threads* sia per implementare il client, che il *Name Server*. Il client comprende due threads che realizzano, rispettivamente, il *client* vero e proprio ed un *server locale* che si occupa di gestire i download dei files locali richiesti da altri hosts. Consideriamo, ad esempio, la richiesta di un file da parte del client al *Name Server*. La figura 1 mostra l'esecuzione di questo comando sul client.

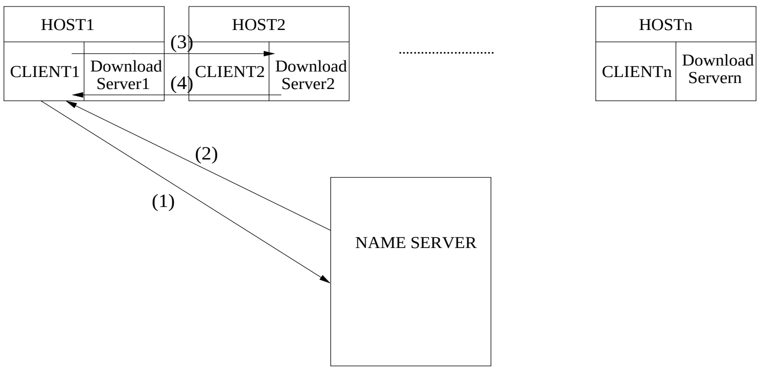

Figure 1: Struttura del client

Il client richiede a *NameServer* i nomi degli hosts che posseggono il file (collegamento (1) in figura). Il Name Server invia la lista l_H dei nomi degli hosts al client, altrimenti invia un messaggio di errore (2). Nel caso di risposta positiva da *Name Server*, il client sceglie uno tra gli hosts in l_H ed invia un messaggio di richiesta a quell'host (3). La richiesta viene gestita dal DownLoad Server dell'host remoto, che invia direttamente il file al Client (4). Se l'host scelto non é disponibile, il client ne sceglie un altro da l_H. Se nessuno degli hosts elencati in l_H é disponibile, il trasferimento fallisce.

Anche il *Name Server* deve utilizzare il meccanismo dei threads per realizzare un *Server Concorrente*. Il server genera un thread per ogni operazione richiesta dal client ed ogni thread generato opera autonomamente sulla struttura *Naming*.

La struttura complessiva del *Name Server* e' presentata in figura 2.

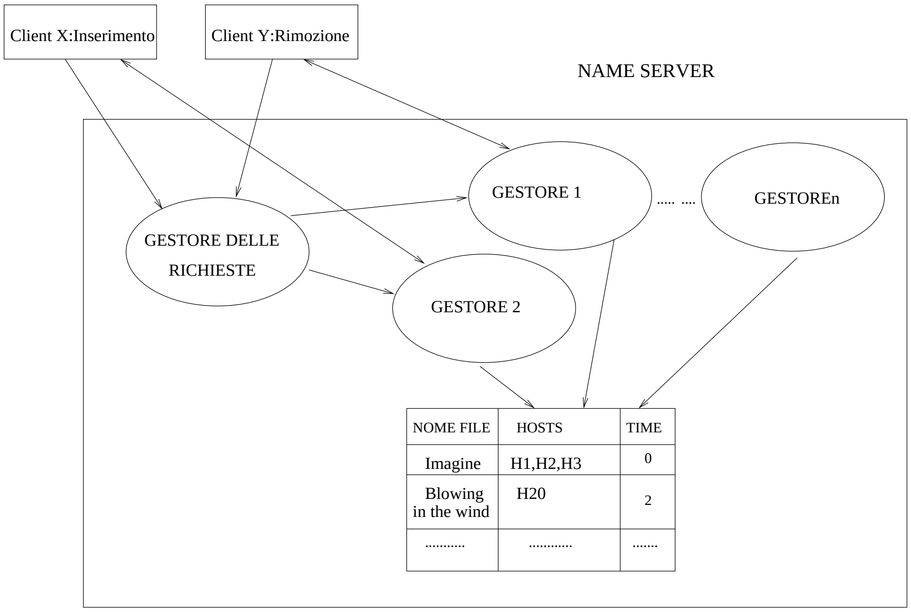

Figure 2: Struttura del server

3 Strumenti per l'implementazione dal Sistema

L'implementazione dello strumento deve utilizzare i seguenti strumenti

- il protocollo *UDP/IP* deve essere utilizzato per la comunicazione tra i clients ed il *Name Server*. Si richiede di definire un opportuno protocollo per implementare queste comunicazioni che consideri sia la possibilità di perdita dei dati nella trasmissione dei dati dal client al *Name Server*, sia quella di perdita della risposta inviata dal *Name Server* al client.

- il servizio di rete TCP/IP deve essere utilizzato per il trasferimento dei files tra gli hosts.

Tutte le eccezioni che si possono verificare a tempo di esecuzione vanno esplicitamente gestite. Per ogni caso intercettare l'eccezione più specifica e gestirla nel modo più idoneo.

4 Fault Tolerance

Si vuole gestire l'eventualità di una caduta del nodo su cui è in esecuzione il *Name Server* o di un qualsiasi client.
Per quanto rigurda la prima possibilità, si vuole evitare che una eventuale caduta del nodo su cui e' allocato il Name Server provochi la perdita delle informazioni memorizzate nella struttura *Naming*. Per questa ragione, si definisce un secondo NameServer di backup, *SlaveNameServer*, allocato su nodo diverso da quello su cui è in esecuzione il *Name Server* principale, che mantenga una copia della struttura *Naming*. Questa copia deve essere periodicamente aggiornata dal Name Server primario. Ogni client deve rivolgersi a *SlaveNameServer* nel caso in cui il *Name Server* primario non sia attivo. Inoltre, al momento della riattivazione del *Name Server* primario, esso si connette allo slave e ricarica la copia di *Naming*. Il salvataggio periodico della tabella nel *SlaveNameServer* è gestito da un ulteriore thread di *Name Server*. Le connessioni tra *Name Server* primario e *SlaveNameServer* devono essere realizzate con connessioni *TCP*.
Per quanto riguarda la caduta di un client, si deve considerare il caso in cui il trasferimento di files tra host possa essere interrotto per la caduta del mittente. In questo caso, il client che aveva richiesto il download del file, si rivolge ad un altro host che possiede lo stesso file, se questo esiste, e richiede l'invio della parte rimanente del file.

5 Modalità di svolgimento del Progetto

- Le prime tre parti del progetto *sono necessarie* per essere ammessi alla prova orale
- Il materiale consegnato deve comprendere:
  - Una stampa di tutto il codice dello strumento e di eventuale programmi utilizzati per il test delle funzionalità dello strumento
  - Una relazione di non piú di 20 pagine che descriva tutte le scelte effettuate.
- Il progetto deve essere consegnato via e-mail una settimana prima della data in cui si intende sostenere l'orale.