Parte implementativa:

Pacca dovrebbe scrivere tutta la parte relativa all’invio e alla ricezione dei file. Come avviene, come è stata implementata. La parte relativa ad Agrawala, la creazione dei vari figli del server. Il compito dei due figli del server, la gestione della memoria dinamica e delle pipe e dove vengono utilizzate (sempre Agrawala). Scrivere come il DNS fornisce gli indirizzi dei server presenti utilizzati per Agrawala. Scrivere come viene scritto il file di configurazione e il file di log con locking.

Marina dovrebbe scrivere tutta la parte riguardante l’operazione scrivi file, e le altre operazioni del server. Cosa avviene sul server nel momento in cui è richiamata l’operazione scrivi file. Come i vari server si scambiano le informazioni sullo stato dei file aggiornati nel momento del loro avvio. E il funzionamento del client. Risoluzione delle failure.

PS: Analizzare bene (??) il funzionamento di TCP e le problematiche dello scambio di messaggi.

Vedere insieme la parte di testing delle failure.

# Scrittura e modifica dei file

Una delle funzionalità offerte dal sistema realizzato consiste nella possibilità di scrivere file modificandone il contenuto.

L’utente dopo aver fatto una richiesta di scrittura può accedere ad un file esistente, oppure crearne uno nuovo se questo non esiste e sottomette al server le modifiche che desidera apportare.

Una volta finita l’operazione l’utente attraverso la digitazione del comando “commit” sul terminale ordina alla macchina di salvare il contenuto. A questo punto la macchina remota si occuperà di effettuare le modifiche e invierà un messaggio di conferma all’utente, preparandosi poi a ricevere una nuova richiesta di operazione da effettuare. Se. Invece, l’utente decidesse di annullare le modifiche scritte, esse verranno eliminate senza lasciare traccia sui file contenuti nel file system. Anche in questo caso la macchina confermerà di aver annullato l’operazione intrapresa e sarà pronta per una nuova operazione.

Per poter richiamare questa funzionalità l’utente deve semplicemente inviare una richiesta di operazione “scrivi file”, digitandola a console. Il client, riconosciuta l’operazione che si intende effettuare, si occupa di chiedere anche il nome del file che si vuole modificare o scrivere, dopodiché sottomette queste informazioni e invia i dati necessari attraverso il pacchetto applicativo, definito precedentemente, rispettivamente nei campi di *Tipo operazione* inserendo “scrivi file” e nel campo *Nome File* viene inserito il nome digitato dall’utente comprensivo di estensione.

In questo modo il server dispone delle informazioni necessarie per poter richiamare la funzione associata, *richiestaScritturaFile()*. In primo luogo viene creato un file temporaneo che conterrà le aggiunte apportate dall'utente e viene ricercato il file selezionato o creato a seconda delle condizioni sopra descritte. Una volta preparati gli elementi necessari alla memorizzazione delle modifiche, viene creato un ID casuale che permette di identificare lo scambio di messaggi durante le varie fasi.

Il server invia all’utente il messaggio di notifica che è pronto a ricevere, attraverso il pacchetto applicativo. Il client e il server si scambiano questi pacchetti contenenti nel campo messaggi le modifiche che si vogliono effettuare. Il pacchetto viene inizializzato con i dati relativi all’ID di transazione, al tipo di operazione e ai messaggi inviati.

Lo scambio di messaggi continua inserendo di volta in volta le modifiche nel file temporaneo fino a quando l’utente non sottomette un messaggio contenente il comando "*commit*" od il comando "*abort*".  
Nel primo caso la macchina deve rendere effettive le modifiche, prima che ciò avvenga però è necessario che il file in uso non sia in conflitto con altre macchine (ossia nessuno voglia modificare lo stesso file). Per questo motivo viene inviata una richiesta di permesso in scrittura del file a tutte le altre macchine presenti nel sistema.

Ogni server gestisce una lista di file che sono in uso dalla macchina, ogni processo in esecuzione utilizza un meccanismo di comunicazione tra processi ad area di memoria condivisa dove vengono inseriti i nomi dei file. Dopo aver inizializzato questo array di stringhe viene avviata una richiesta di scrittura verso tutte le altre macchine del sistema. I meccanismi che regolano la sincronizzazione tra i vari processi di modifiche sono trattati in un'altra sezione [riferimento ad agrawala].

Una volta inviata la richiesta di scrittura il server non deve far altro che aspettare la conferma da parte delle altre macchine. Anche in questo caso si è ricorso ad uno dei meccanismi di comunicazione interprocessi. in questo modo si comunica con gli altri processi sulla stessa macchina per avere l'ok a procedere.

Nel caso in cui il server che riceve la richiesta non stia lavorando sullo stesso file invia un messaggio di ok al richiedente, altrimenti vengono innescati degli algoritmi che permettano la scrittura in maniera ordinata del file in base all’algoritmo di Agrawala (che viene descritta in maniera più completa e dettagliata nel paragrafo apposito).

Agrawala, quindi, permette al server di effettuare le operazioni di scrittura in modalità atomica. Una volta ricevute le conferme da parte di tutti gli altri server, il file temporaneo viene reso permanente e cancellata la vecchia versione, dopodiché il file aggiornato viene inviato a tutte le altre macchine, in modo da non avere incoerenze nel file system.

Una volta terminato il processo il server invia una conferma dell'avvenuta operazione e l'utente sarà in grado di procedere con nuove operazioni.

Nel caso di abort, invece, il server si occuperà semplicemente di cancellare ogni traccia del file temporaneo contenenti tutte le modifiche inserite fino a quel momento, ed invia un messaggio di conferma di annullamento al client.

# Sincronizzazione del file system

Nel momento in cui un server viene avviato, in prima istanza deve sincronizzare il contenuto del suo file system con quello delle macchine già presenti, in modo che, dopo la fase di avvio, esso si trovi in uno stato coerente e aggiornato. Quindi, per prima cosa richiede al DNS gli indirizzi degli altri server e si mette in contatto con un altro server attivo al quale richiede tutti i dati presenti nel suo file system e li copia nella sua cartella locale.

Si è deciso di operare questa strategia perchè nonostante possa diventare oneroso con grandi quantità di file, questo tipo di operazione avviene solo nella fase di avvio, durante l'esecuzione dei processi avverranno solo degli aggiornamenti e questo giustifica l'onere iniziale del trasferimento dei vari file. In alternativa, per ottimizzare questo processo, il server avrebbe dovuto richiedere agli altri server solo gli aggiornamenti dei file effettivamente modificati rispetto a quelli presenti nel proprio file system locale, durante il periodo di inattività.

Se nessun server è attivo la sincronizzazione non avviene, altrimenti, dopo essersi connesso ad un'altra macchina, invia un messaggio di richiesta di aggiornamento attraverso il pacchetto applicativo. A questo punto il server contattato si occupa di inviare tutti i file presenti nella sua memoria locale. Una volta conclusa l'operazione la comunicazione viene terminata.

# Funzionamento client

Il client, una volta avviato, contatta il DNS, che si occupa di smistare le varie richieste ai vari server disponibili, in base all’algoritmo di tipo round robin. La funzione del DNS consiste semplicemente nel fornire l'indirizzo e la porta del server remoto da contattare, quindi il client inoltrerà una nuova richiesta di connessione al server indicato dal DNS. Può accadere però che il server non sia disponibile, per questo motivo è stato settato un tempo di timeout per connettersi al server. Se questo timeout scade il client provvede a fare una nuova richiesta al DNS.

Una volta connesso ad un server disponibile, questo permetterà di effettuare diversi tipi di operazione. Il client può chiedere di visualizzare la lista dei file presenti nella macchina remota, di visualizzare o modificare un file ed infine può richiedere di uscire dall'applicazione.