

Egrid Cache System

Angelo Leto

DRAFT

October 12, 2004

Contents

1	Introduzione	2
1.1	Precondizioni	2
2	Upload	3
2.1	Descrizione	3
3	Download	5
3.1	Descrizione	5
4	Esecuzione dei jobs in griglia	7
4.1	Descrizione	7

Chapter 1

Introduzione

Questo documento presenta una proposta per l'implementazione di un servizio di file cache che eviti il download della stessa porzione di dati in maniera analoga di un proxy cache web.

L'architettura qui proposta fa uso di tecnologie crittografiche per mantenere la coerenza con il sistema di autorizzazione dell'accesso ai files come definito sullo storage principale, nessun utente può accedere ad un dato nella cache se non ha accesso alla "master copy" sullo storage principale. L'evoluzione naturale di questo sistema prevede un sistema di metadati sui quali effettuare il controllo di accesso e lo storage dei parametri crittografici per il controllo dell'accesso.

1.1 Precondizioni

Il cache daemon girerà su tutti i nodi periferici, utilizzerà una directory sul filesystem sulla quale memorizzare le copie cache dei files richiesti, ad es.: `/tmp/egrid-cache/`; in tale directory sarà concesso leggere a tutti ma scrivere solamente all'utente root¹. L'algoritmo crittografico proposto per lo storage dei files è AES², tale misura vuole proteggere dal problema che root potrebbe leggere i files in cache, quindi da danni accidentali dovuti a cattiva gestione del contenuto della directory di cache. L'algoritmo di hashing proposto è MD5: un algoritmo per le checksums come MD5 è necessario per ottenere identificatori univoci per i files in griglia, oltre che per controllarne l'integrità. Per il cache daemon sarà possibile girare in modalità "no cache", anche se ciò sembra una contraddizione, è necessario nel caso del daemon che girerà sullo storage principale sul quale non è necessario il servizio di cache, ma che necessita invece del servizio di recupero dei files crittografati ad esempio durante l'esecuzione di un job, ma tale caso sarà trattato più in dettaglio in seguito.

Per comodità di esposizione, il servizio di cache viene descritto da due flow chart distinti rispettivamente per l'operazione di Upload e di Download.

¹-rw—— questo come vedremo non costituisce un problema dal punto di vista dell'autorizzazione in quanto le informazioni contenute nei files sono crittografate, comunque per scongiurare l'eventualità di attacchi di tipo brute force è possibile rendere leggibile l'intera directory solo a root.

²rimane possibile che qualora l'utente non ritenga necessario proteggere il file da occhi indiscreti, questi può decidere se utilizzare o meno la crittografia, con il vantaggio di una maggiore velocità nel recupero del file richiesto.

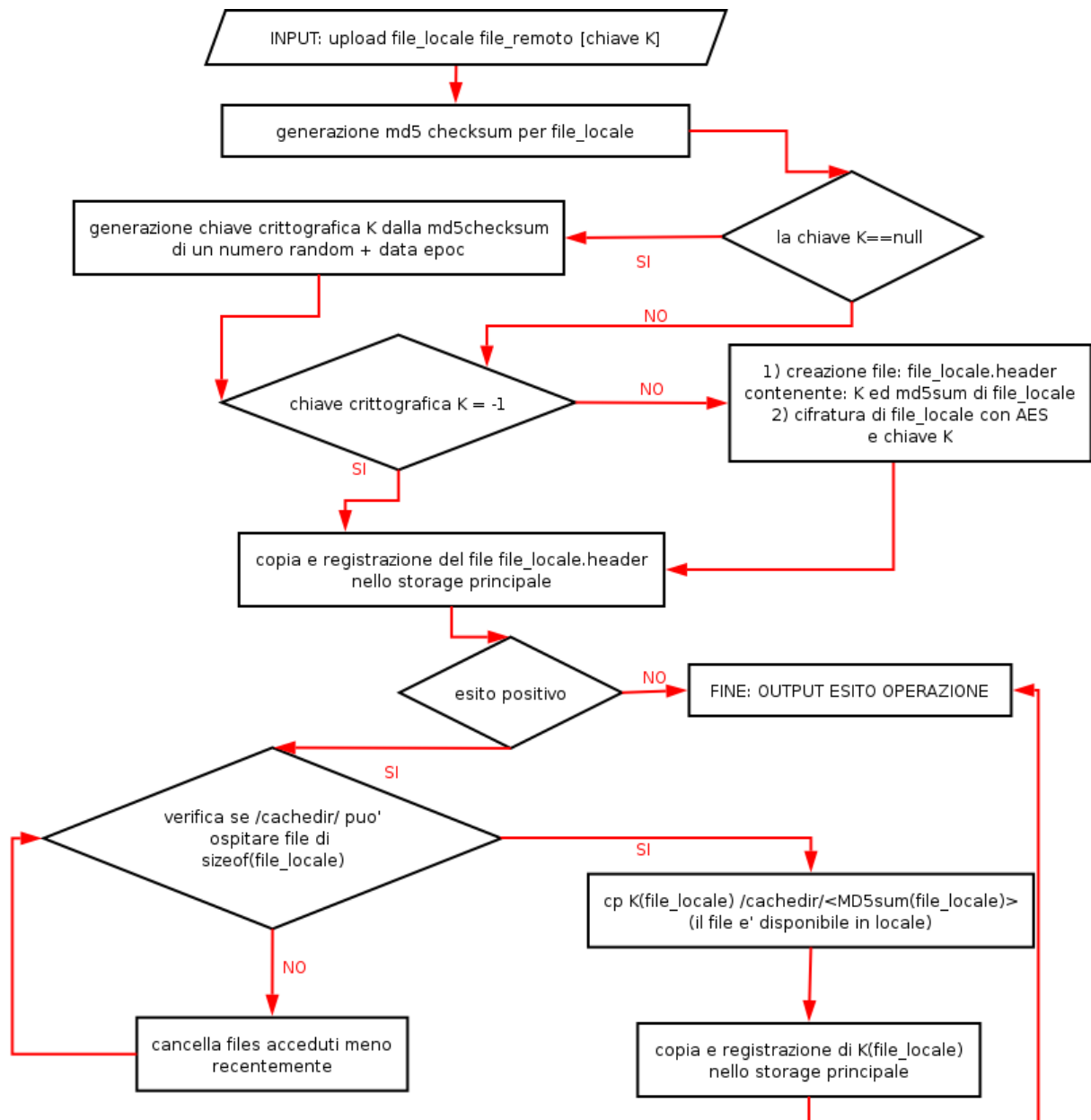
Chapter 2

Upload

2.1 Descrizione

Nel corso di questo paragrafo, dove si esplica il funzionamento del cache service in fase di Upload si farà riferimento al flow chart in figura1.

L'operazione di Upload di un file inizia quando l'utente richiede al cache daemon di caricare sullo storage principale un file *file_locale* nella posizione *file_remoto*. L'utente può inserire una stringa di caratteri che costituirà la chiave crittografica, (eventualmente -1 indica che non si esegue crittografia sul file) se nessun valore è immesso, la chiave sarà generata in modo random. In seguito viene estratta dal file *file_locale* la checksum MD5 per la creazione del file *file_remoto.header*. Questo file contiene la chiave crittografica K con la quale si oscurerà il file da uploadare, la MD5 checksum che servirà ad identificare univocamente il contenuto del file e per controllarne l'integrità; ad esempio quando un utente richiede un dato file attraverso il suo filename, il cache daemon controlla se la checksum del file che si intende scaricare corrisponde ad un file nella cache, in questo modo si identifica il contenuto de file master copy che potrebbe variare. A questo punto il file *file_remoto.header* viene posto sullo storage principale, il file da uploadare viene oscurato tramite la chiave K e viene posto in cache quindi reso disponibile in locale; successivamente inizia il trasferimento e la registrazione del file sullo storage principale.



Chapter 3

Download

3.1 Descrizione

Nel corso di questo paragrafo, dove si esplica il funzionamento del cache service in fase di Download si farà riferimento al flow chart in figura2.

L'operazione di Download inizia quando l'utente richiede al cache daemon un file tramite il suo nome logico e specificando il nome che il file dovrà assumere una volta giunto in locale, quindi il comando sarà ad es.:

```
$> getfile lfn_remoto pfn_locale
```

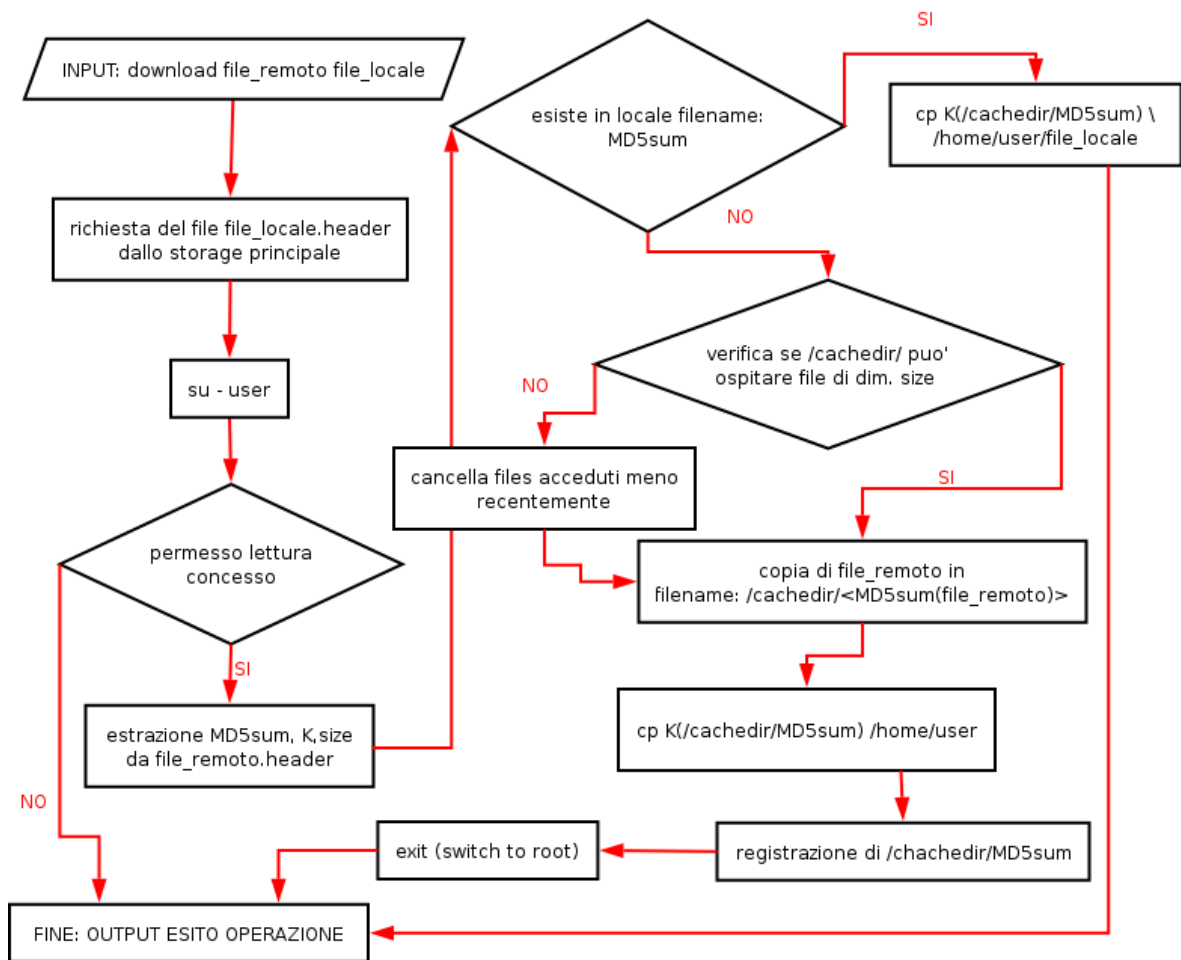
Per prima cosa il cache daemon (che gira come root) esegue il comando

```
$> su - user
```

utilizzando quindi le credenziali dell'utente *user* che ha lanciato il comando. Quindi chiede il file *lfn_remoto.header*, tale file contiene la chiave K con la quale è cifrato *lfn_remoto*. Dato che i files: *lfn_remoto.header* e *lfn_remoto* hanno sia gli stessi permessi che le stesse ownerships, l'utente è abilitato a leggere o entrambi i files o nessuno dei due. Nel secondo caso il cache daemon restituisce un messaggio di errore e l'operazione termina. Nel primo caso invece, il cache daemon estrae le informazioni presenti nel file *lfn_remoto.header*, controlla che la directory */tmp/egrid-cache/* non contenga un file il cui nome sia uguale alla hash estratta da *lfn_remoto.header*; sono possibili due casi:

1. **il file esiste:** il cache daemon copia il file locale nella destinazione prescritta in input dall'utente dopo averlo decifrato con la chiave K
2. **il file non esiste:** viene scaricato dallo storage principale nella directory di cache con un nome corrispondente alla sua MD5 checksum (valore contenuto in *lfn_remoto.header*), quindi provvede a copiare il file decrittano nella directory prescritta dall'utente come nell'azione intrapresa nel caso precedente.

Si può constatare che ha accesso alla chiave crittografica solo chi è in possesso di un certificato riconosciuto dall'infrastruttura di autenticazione della griglia ed è autorizzato ad accedere alla master copy del file come richiesto dai requisiti.



Chapter 4

Esecuzione dei jobs in griglia

4.1 Descrizione

Nel corso di questo paragrafo, si esplica il funzionamento del cache service in fase di esecuzione dei jobs in griglia.

Ogni volta che si esegue un job in griglia il recupero dei files necessari all'esecuzione vengono ottenuti tramite interrogazione del cache server, che esegue tutte le operazioni descritte per il download. Ci sono due casi:

1. job eseguito con files provenienti da storage principale (senza cache): il software richiede il file al proxy cache che controllate le credenziali e i permessi provvede al recupero del file, sul quale il programma è in grado di intervenire
2. job eseguito con files provenienti da storage con cache: identico al caso precedente, solo che il file potrebbe provenire dalla cache con una diminuzione del tempo di recupero.