

Reti di Calcolatori ed Ingegneria del Web - A.A. 2017/18

Progetto A: Web Server con adattamento dinamico di contenuti statico

Lo scopo del progetto è quello di realizzare in linguaggio C usando l'API del socket di Berkeley un Web server concorrente in grado di adattare dinamicamente il contenuto statico fornito in base alle caratteristiche del dispositivo del client. Il server dovrà fornire:

1. le funzionalità di base di un Web server (incluso il logging);
2. il supporto minimale del protocollo HTTP/1.1 (connessioni persistenti, metodi GET e HEAD);
3. l'adattamento dinamico (on-the-fly) di immagini. Tale adattamento dovrà essere effettuato almeno per il formato JPEG sulla base del fattore di qualità specificato dal client nell'header Accept; ad esempio, nel caso di Accept: image/jpeg; q=0.5, text/plain il server fornirà in risposta l'immagine JPEG richiesta con fattore di qualità pari a 0.5. L'adattamento potrà anche essere effettuato per altri formati di immagini ed applicando operazioni di manipolazione dell'immagine (ad es. conversione del formato, resizing, riduzione del numero di colori) sulla base di informazioni riguardanti il browser e contenute nell'header User-Agent (si consiglia di usare WURFL per ottenere le informazioni sulle caratteristiche dei dispositivi (<http://wurfl.sourceforge.net/>));
4. il caching della versione risultante dall'operazione di adattamento, allo scopo di ridurre l'overhead computazionale dell'adattamento. Implementare il caching in modo da gestire la presenza di versioni multiple di una stessa risorsa.

Per l'adattamento delle immagini usare un programma esterno, ad esempio:

- ImageMagick (in particolare, il programma convert), disponibile all'URL <http://www.imagemagick.org/>;
- la libreria Independent JPEG Group (in particolare, i programmi djpeg e cjpeg), disponibile all'URL <http://www.ijg.org/>.

Il server dovrà essere eseguito nello spazio utente ed essere in ascolto su una porta di default (configurabile); potrà essere testato tramite un browser reale o tramite un tool a riga di comando quale cURL, disponibile all'URL <http://curl.haxx.se/>.

Si richiede di valutare le prestazioni del Web server realizzato (non includendo le funzionalità di adattamento dei contenuti) in termini di tempo di risposta e throughput e confrontarle con quelle di Apache 2.x usando un semplice workload di tipo statico. La valutazione delle prestazioni può essere effettuata utilizzando il tool di benchmarking httpperf (disponibile all'URL http://www.hpl.hp.com/personal/David_Mosberger/httpperf.html).

Scelta e consegna del progetto

Il progetto può essere realizzato da un gruppo composto al massimo da 4 studenti.

Per poter sostenere l'esame nell'A.A. 2017/18, è necessario prenotarsi per il progetto, comunicando al docente le seguenti informazioni:

- nominativi ed indirizzi di e-mail dei componenti del gruppo;
- progetto scelto.

Per ogni comunicazione via e-mail è necessario specificare [IW17] nel subject della mail.

Eventuali modifiche relative al gruppo devono essere tempestivamente comunicate e concordate con il docente.

Il progetto è valido solo per l'A.A. 2017/18: i progetti scadranno inderogabilmente con la sessione d'esame di febbraio 2019.

La consegna del progetto deve avvenire almeno dieci giorni prima della data stabilita per la discussione del progetto.

La consegna del progetto consiste in:

- un CD-ROM contenente tutti i sorgenti (opportunamente commentati) necessari per il funzionamento e la copia elettronica della relazione (in formato pdf);
- la copia cartacea della relazione.

La relazione contiene:

- la descrizione dettagliata dell'architettura del sistema e delle scelte progettuali effettuate;
- la descrizione dell'implementazione;
- la descrizione delle eventuali limitazioni riscontrate;
- l'indicazione della piattaforma software usata per lo sviluppo ed il testing del sistema ;
- alcuni esempi di funzionamento;
- un manuale per l'installazione, la configurazione e l'esecuzione del sistema.

Valutazione del progetto

I principali criteri di valutazione del progetto saranno:

- rispondenza ai requisiti;
- originalità;
- efficienza;
- scalabilità;
- robustezza;
- leggibilità del codice;
- modularità del codice;
- organizzazione, chiarezza e completezza della relazione;
- semplicità di installazione e configurazione del software realizzato in ambiente Linux.