

# Ingegneria del Web - A.A. 2007/08

## Progetto B: Web switch QoS-aware

Ing. Valeria Cardellini

### Requisiti del progetto

Lo scopo del progetto è realizzare in linguaggio C usando l'API del socket di Berkeley un **Web switch QoS-aware**. Tale tipologia di Web switch è in grado di fornire agli utenti del Web cluster un servizio differenziato mediante il supporto dei meccanismi per la qualità del servizio (QoS) tradizionalmente adottati nelle reti (classificazione dei servizi, controllo di accesso delle richieste, elevato utilizzo delle risorse, isolamento delle prestazioni).

I requisiti salienti del Web switch sono elencati di seguito.

- Il supporto delle funzionalità di base di un Web switch concorrente content-aware ed operante in modalità two-way (la gestione di connessioni TCP con i client e con i server del Web cluster, l'analisi della richiesta HTTP e la selezione di un target server, l'inoltro dei pacchetti del flusso di richiesta al target server e l'inoltro dei pacchetti del flusso di risposta al client);
- il supporto del protocollo HTTP/1.1 (in particolare, connessioni persistenti);
- il logging delle richieste;
- il supporto di meccanismi per la qualità del servizio (QoS):
  - la differenziazione delle richieste degli utenti in classi di servizio, anche in base ad informazioni sul client di livello applicativo;
  - il controllo di ammissione delle richieste, basato sulle classi di servizio e sullo stato di carico del cluster;
  - il servizio differenziato delle richieste, ottenibile mediante l'isolamento delle prestazioni tra le classi di servizio (politiche di scheduling oppure partizionamento delle risorse) ed un elevato utilizzo delle risorse del cluster.

Una descrizione dei meccanismi per la QoS adottati nelle reti si trova nel seguente testo: J.F. Kurose, K.W. Ross, "Reti di Calcolatori e Internet - Un Approccio Top-Down, Terza Edizione", Pearson Addison-Wesley, 2005.

Per approfondire lo studio dei meccanismi per la qualità del servizio nei Web cluster si possono considerare le soluzioni proposte in:

M. Andreolini, E. Casalicchio, M. Colajanni, M. Mambelli, "A cluster-based Web system providing differentiated and guaranteed services", *Cluster Computing*, Vol. 7, No. 1, pp. 7-19, 2004. [http://www.ce.uniroma2.it/courses/iw08/articoli/QoWS\\_cc04.pdf](http://www.ce.uniroma2.it/courses/iw08/articoli/QoWS_cc04.pdf)

V. Cardellini, E. Casalicchio, M. Colajanni, M. Mambelli, "Enhancing a Web-server cluster with Quality of Service mechanisms", *Proc. of 21st IEEE Int'l Performance, Computing, and Communications Conference*, pp. 205-212, Apr. 2002. <http://www.ce.uniroma2.it/publications/ipccc2002.pdf>

K. Gilly, S. Alcaraz, C. Juiz, R. Puigjaner, "Service differentiation and QoS in a scalable content-aware load balancing algorithm", *Proc. of 40th Annual Simulation Symposium*, pp. 185-193, 2007.

oppure altri lavori sull'argomento, reperibili ad esempio tramite <http://scholar.google.com/>.

Non è richiesto che il Web switch supporti URL diverse da http.

Il Web switch deve essere eseguito nello spazio utente e senza richiedere privilegi di root e deve essere in ascolto su una porta di default (configurabile). I parametri di configurazione dello switch e del cluster di server devono essere specificati tramite un apposito file di configurazione.

Il funzionamento del Web switch e la sua rispondenza ai requisiti possono essere testati tramite un browser oppure un tool a riga di comando quale cURL (disponibile all'indirizzo <http://curl.haxx.se/>).

## Opzionale

Il progetto può prevedere anche lo svolgimento di uno o più tra i seguenti punti:

- la realizzazione di una semplice interfaccia grafica per la configurazione del Web switch e per il monitoring a run-time delle sue prestazioni (tempo di CPU, uso della memoria, uso della rete, ...);
- la valutazione delle prestazioni del Web switch realizzato (in termini di tempo di risposta e di throughput ed all'aumentare del carico offerto al sistema) considerando due classi di servizio. Come tool di benchmarking, si consiglia `httperf`, reperibile all'URL <http://www.hpl.hp.com/research/linux/httperf/> con la relativa documentazione. Durante l'esecuzione dei test di benchmarking si consiglia di disabilitare il logging delle richieste sul Web switch.

## Consegna del progetto

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

Per poter sostenere l'esame nell'A.A. 2007/08, **entro mercoledì 12/3/2008** devono essere comunicate via e-mail a Valeria Cardellini le seguenti informazioni:

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

Per ogni comunicazione via e-mail è necessario specificare [IW] 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. 2007/08: i progetti scadranno inderogabilmente con l'ultima sessione d'esame dell'A.A. 2007/08.

La consegna del progetto deve avvenire **almeno dieci giorni** prima della data stabilita per l'appello di Ingegneria del Web in cui si intende sostenere la discussione del progetto e la prova orale (se non già superata). Il materiale relativo al progetto deve essere consegnato a Valeria Cardellini.

La consegna del progetto consiste in:

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

La relazione contiene:

- la descrizione dettagliata dell'architettura dell'applicazione 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 dell'applicazione;
- un esempio di funzionamento;
- un manuale per l'installazione, la configurazione e l'esecuzione dell'applicazione.

## **Valutazione del progetto**

I principali criteri di valutazione del progetto saranno:

1. rispondenza ai requisiti;
2. originalità;
3. efficienza;
4. leggibilità del codice;
5. modularità del codice;
6. organizzazione, chiarezza e completezza della relazione;
7. semplicità di installazione e configurazione del software realizzato in ambiente Linux.