

Project Heimdall

Proposta di implementazione per un webswitch
concorrente two-way di livello 7 (OSI) con
politiche di bilanciamento del carico stateless e stateful

Alessio Moretti - 0187698

Andrea Cerra - ??????

Claudio Pastorini - 0186256

Corso di Ingegneria di Internet e del Web - A.A. 2015/2016

Università di Tor Vergata

Ingegneria Informatica

Roma, 13 gennaio 2016

Indice

1	Example section	1
2	Introduzione	2
2.1	Perchè Heimdall?	2
2.2	Web-Switch di livello 7	2
3	Architettura	3
3.1	Server in ascolto	3
3.1.1	File di configurazione	3
3.1.2	Logging	3
3.1.3	Gestione degli errori	3
3.2	Pool manager	3
3.3	Scheduler	3
3.4	Worker	3
3.4.1	Gestione delle richieste	3
3.4.2	Gestione delle connessioni	3
3.4.3	Thread di lettura	3
3.4.4	Thread di scrittura	3
3.4.5	Thread di richiesta	3
3.4.6	Thread di watchdog	3
4	Ulteriori proposte	4
5	Politiche di scheduling	5
5.1	State-less: implementazione con Round Robin	5
5.2	State-aware: implementazione con monitor di carico	5
5.2.1	Modulo ApacheStatus	5
6	Performance	6
6.1	Test di carico	6
6.2	Comparazione con Apache	6
7	Future implementazioni	7
7.1	Webserver performante	7

Annotazioni	8
A Manuale per l'uso	9
B Vagrant	9
C Cluster virtuale	9
D Tool per i debug	9
D.1 GDB	9
D.2 Valgrind	9
E Tool per i test	9
E.1 PostMan	9
E.2 Telnet	9
E.3 HttPerf	9
E.4 Browser	9



Figura 1: Thor di Asgard, *figlio di Odino*

1 Example section

Sample text and a reference[1]. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec at lorem varius, sodales diam semper, congue dui. Integer porttitor felis eu tempor tempor. Proin molestie maximus augue in facilisis. Phasellus eros dui, blandit eu nibh ut, pharetra porta enim. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam ullamcorper risus pretium est elementum, eget egestas lorem fermentum. Etiam auctor nisi purus, vitae scelerisque augue vehicula sed. Ut eu laoreet ex. Mauris eu mi a tortor gravida cursus eget sit amet ligula.

2 Introduzione

2.1 Perchè Heimdall?

2.2 Web-Switch di livello 7

3 Architettura

3.1 Server in ascolto

3.1.1 File di configurazione

3.1.2 Logging

3.1.3 Gestione degli errori

3.2 Pool manager

3.3 Scheduler

3.4 Worker

3.4.1 Gestione delle richieste

Coda delle richieste

Chunk di dati

3.4.2 Gestione delle connessioni

Connessione

Richieste HTTP

Risposte HTTP

3.4.3 Thread di lettura

3.4.4 Thread di scrittura

3.4.5 Thread di richiesta

3.4.6 Thread di watchdog

4 Ulteriori proposte

5 Politiche di scheduling

5.1 State-less: implementazione con Round Robin

5.2 State-aware: implementazione con monitor di carico

5.2.1 Modulo ApacheStatus

6 Performance

6.1 Test di carico

6.2 Comparazione con Apache

7 Future implementazioni

7.1 Webserver performante

Annotazioni

- [1] Leslie Lamport, *L^AT_EX: a document preparation system*, Addison Wesley, Massachusetts, 2nd edition, 1994.

A Manuale per l'uso

B Vagrant

C Cluster virtuale

D Tool per i debug

D.1 GDB

D.2 Valgrind

E Tool per i test

E.1 PostMan

E.2 Telnet

E.3 HttPerf

E.4 Browser