

Kernel-based Web switches providing content-aware routing

Mauro Andreolini, Michele Colajanni, Marcello Nuccio
Department of Information, Systems and Production
University of Roma Tor Vergata
Roma, Italy

IEEE Network Computing and Applications 2003

Speaker: Hicks Chen

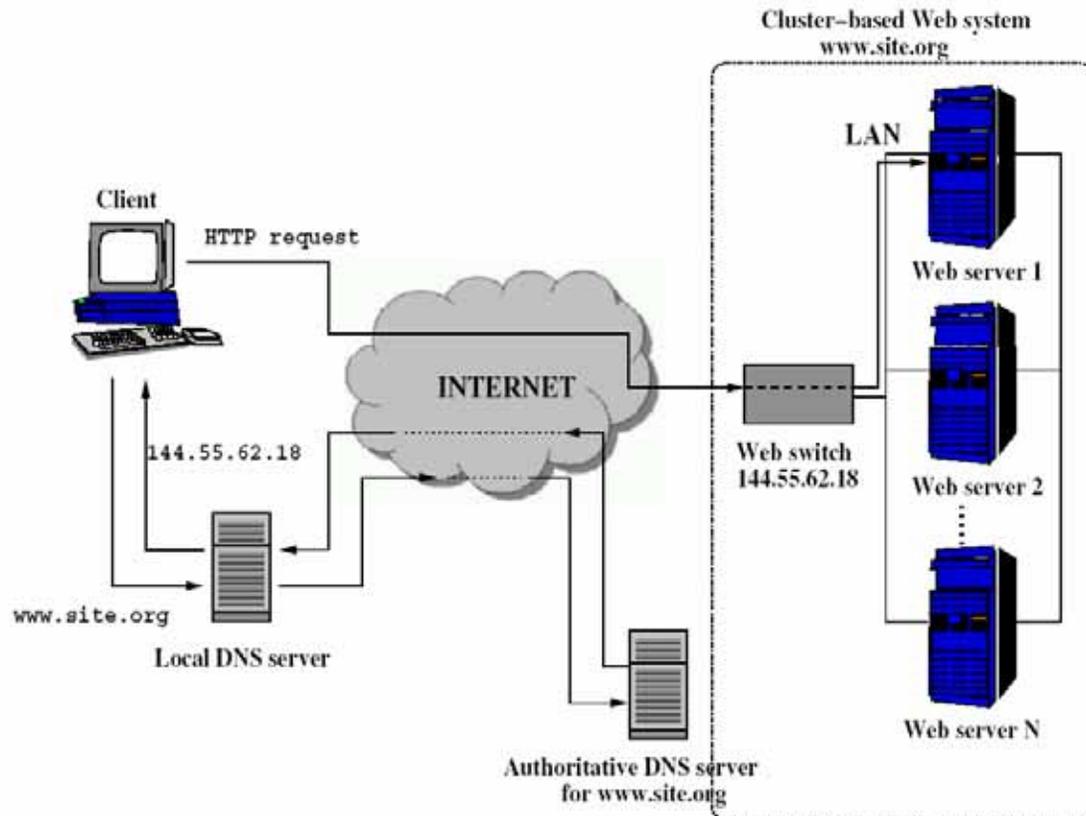


Outline

- **Introduction**
- **Two-way kernel-based Web cluster**
- **One-way kernel-based Web cluster**
- **Experimental results**
- **Conclusions**

Introduction

- *Locally Distributed Web-server Systems, briefly Web clusters*



Introduction

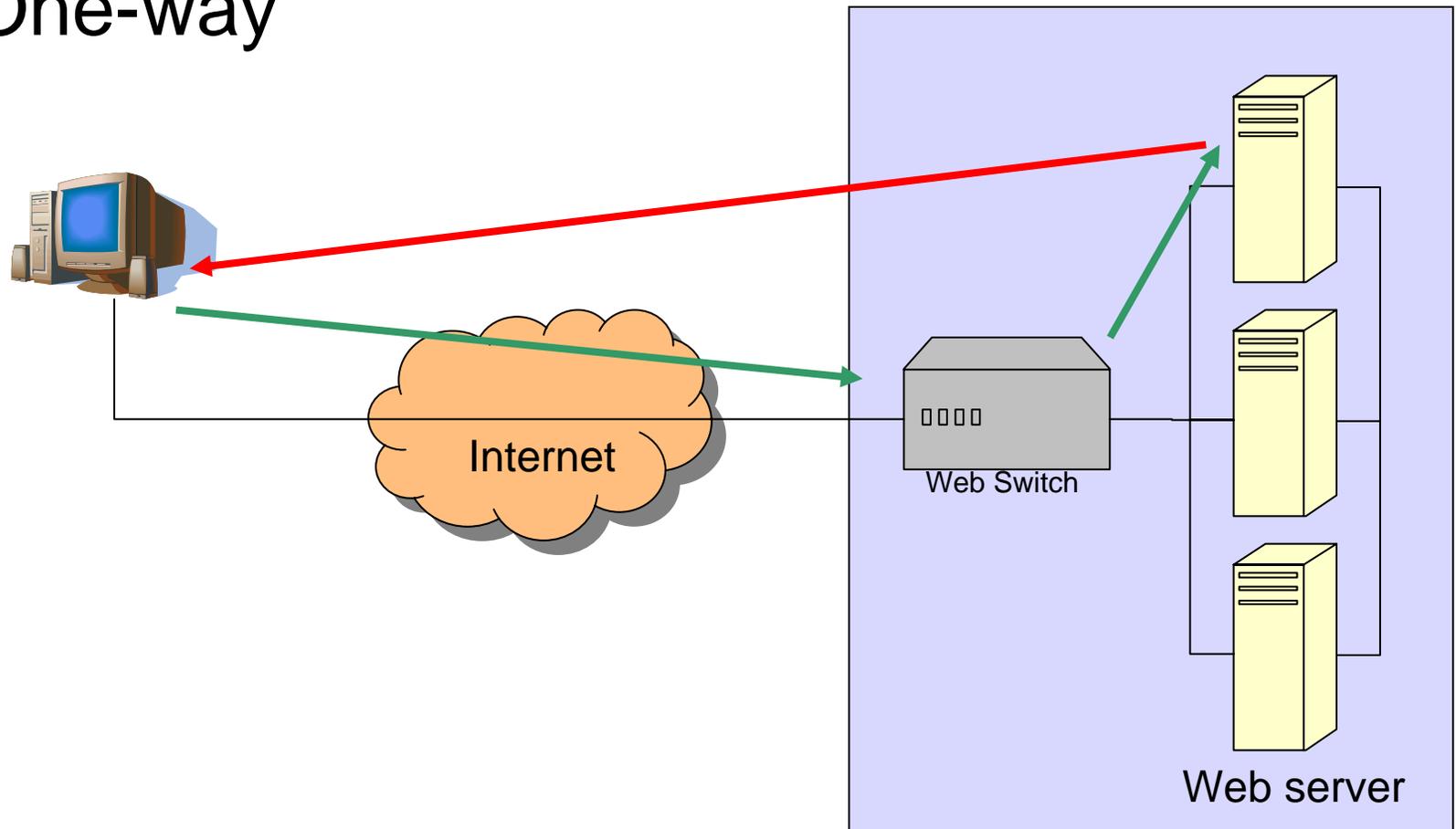
- Can be classified by protocol level:
 - layer-4 : Transport Layer
 - layer-7: Application Layer
- There are two main characteristics to classify layer-7 Web switches:
 - The first concerns the implementation layer:
 - kernel-based
 - application-based.

Introduction

- The second considers the flow of the packet traffic between the client and the Web cluster
 - One-way
 - The servers send response packets directly to the client.
 - Two-way
 - The outbound packets pertaining to a response pass again through the Web switch.

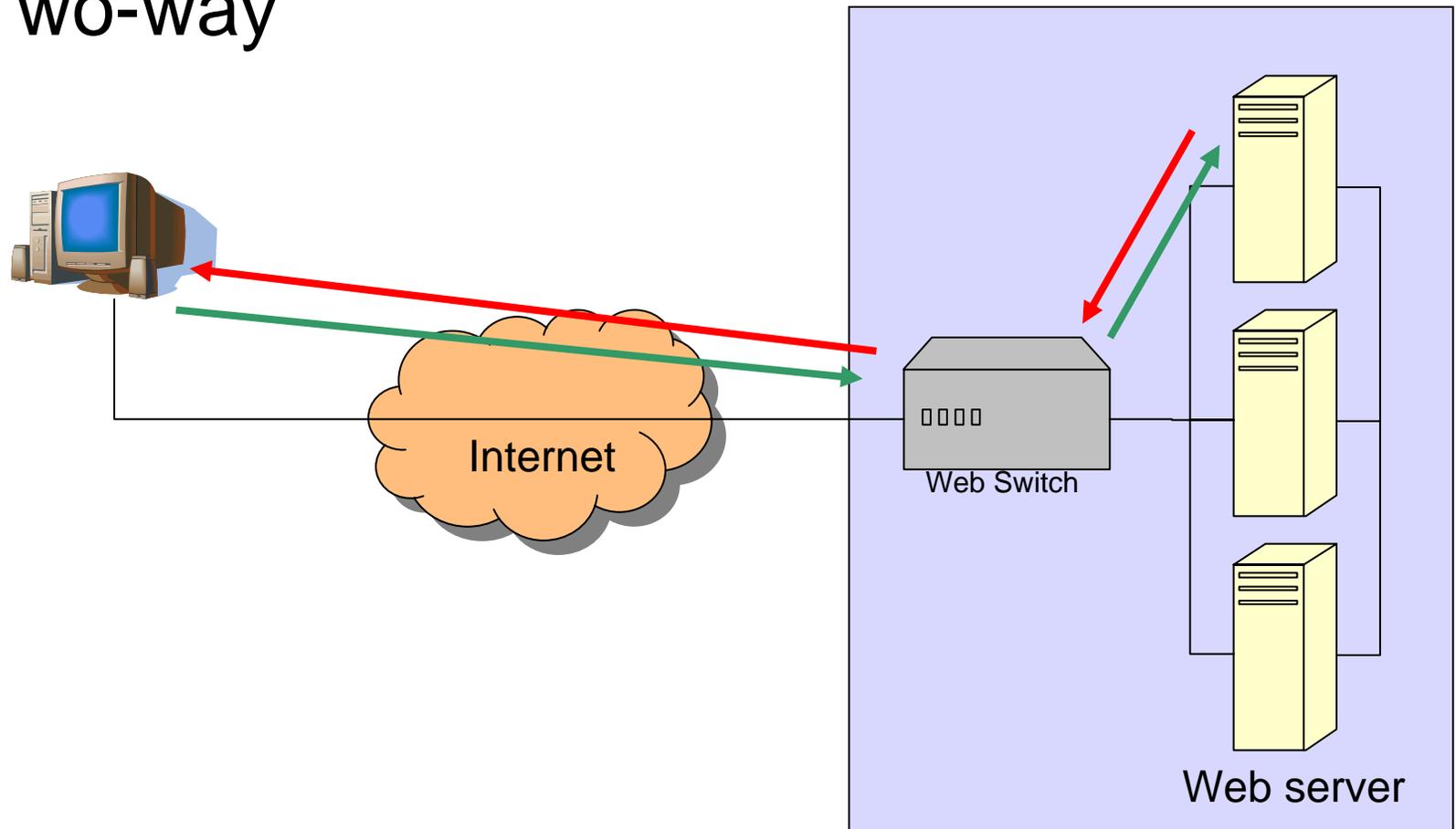
Introduction

- One-way



Introduction

- Two-way



Introduction

- The main focus of this paper is to provide a fair comparison among different 7-layer prototypes

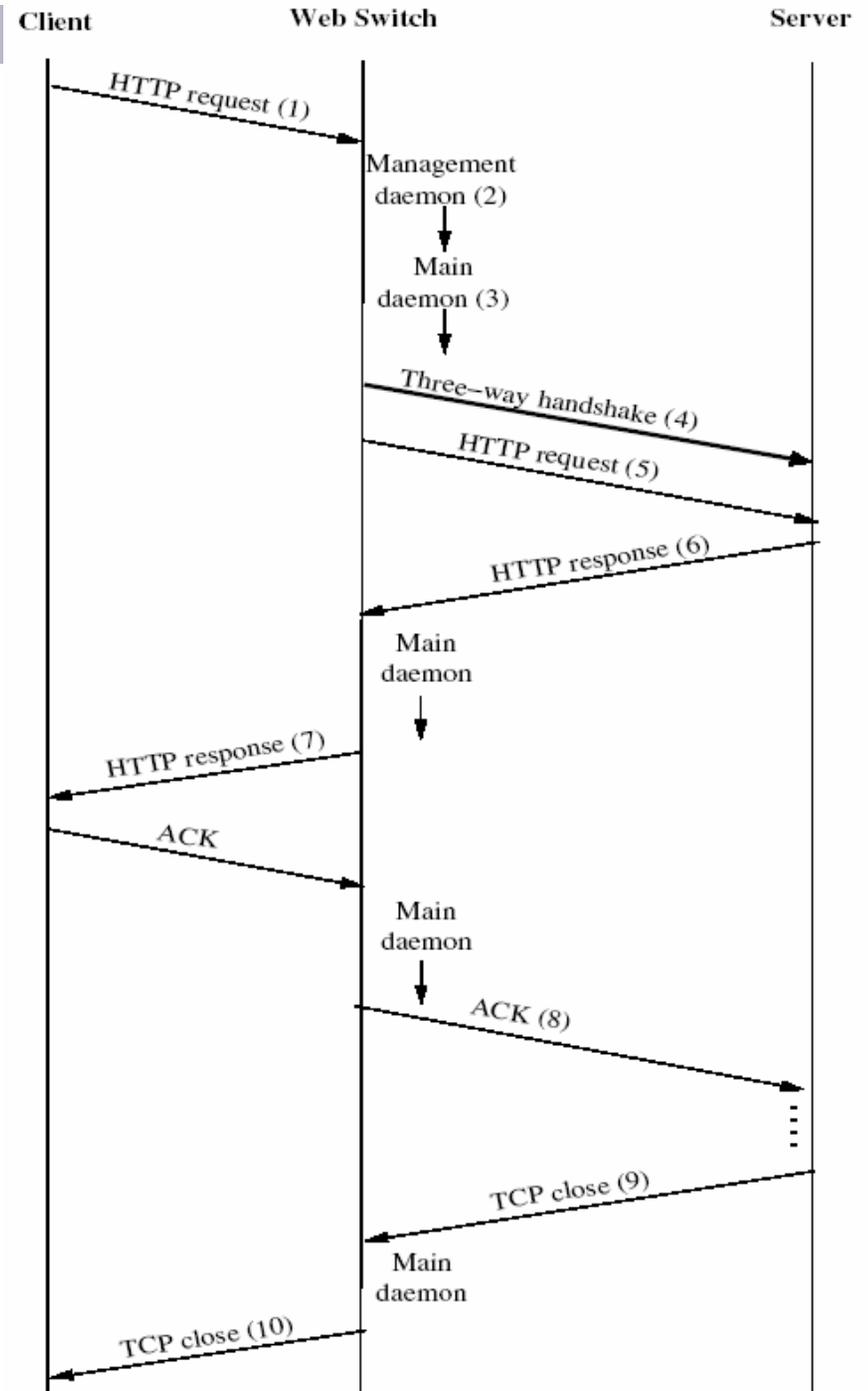
Table 1. Content aware Web switches.

<i>Name</i>	<i>Scheme</i>	<i>Switch implementation</i>	<i>Server implementation</i>	<i>References</i>	<i>Name</i>
TCP Gateway	two-way two-way	application level kernel level	n.a. n.a.	[6, 11] here	ClubWeb-2w-a ClubWeb-2w-k
TCP Splicing	two-way	kernel level	application level	[6, 10]	
TCP Handoff	one-way one-way	FreeBSD kernel Linux kernel	FreeBSD kernel Linux kernel	[6, 16] [3]	ScalaServer ClubWeb-1w

Two-way kernel-based Web cluster

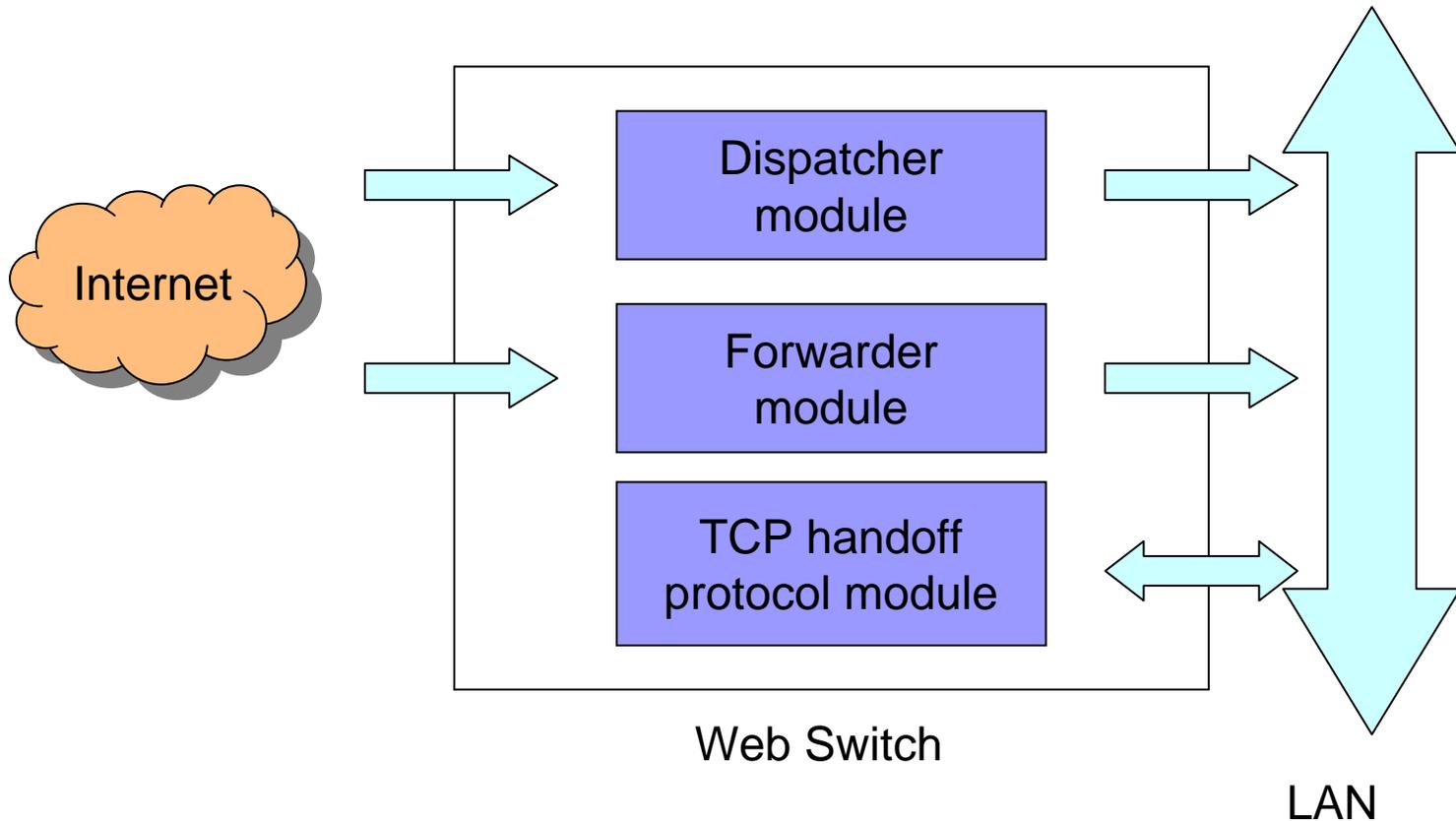
- This architecture is based on the kHTTPd server
- A bunch of *main daemons* are responsible for handling client requests.
- *Main* daemons are spawned by a *management daemon*

Two-way kernel-based Web cluster



One-way kernel-based Web cluster

- Based on the TCP Handoff approach



One-way kernel-based Web cluster

- Dispatcher module:
 1. parsing the HTTP requests
 2. choosing a Web server according to some content-aware dispatching policy

One-way kernel-based Web cluster

- Forwarder module:
 1. Intercepting Ethernet frames belonging to an already transferred TCP connection
 2. transmits them to the appropriate Web server

One-way kernel-based Web cluster

- TCP handoff protocol module :
 - Encapsulation and transmission of the complete state of a TCP connection
 - Notification of TCP connection close from a Web server
 - Notification of TCP connection duplication failure from a Web server

Experimental results

- Evaluating the performance and scalability of the two proposed kernel-based Web cluster architectures
- Compare them with the basic TCP Gateway solution

Experimental results

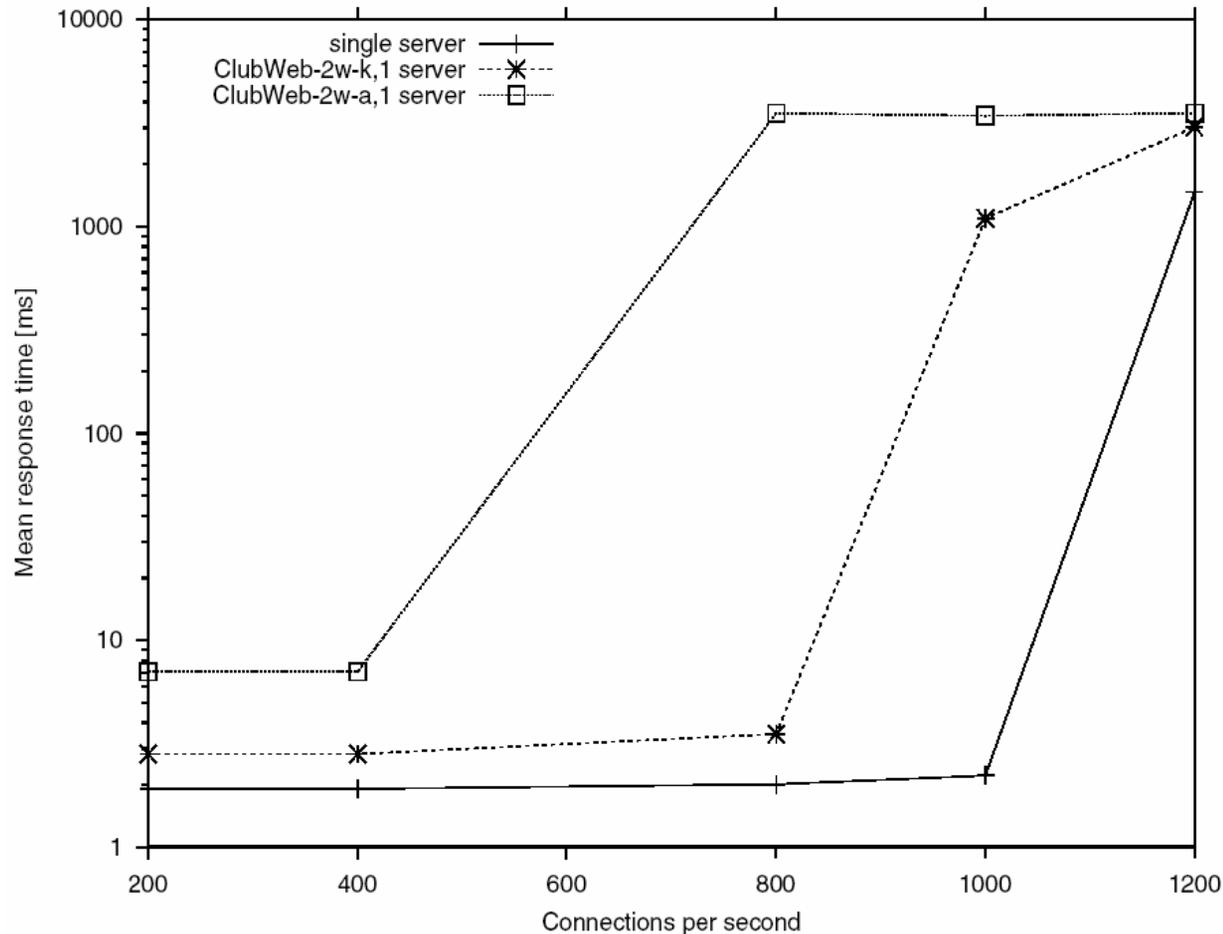


Figure 4. Overheads of kernel-based vs. application-based mechanism.

Experimental results

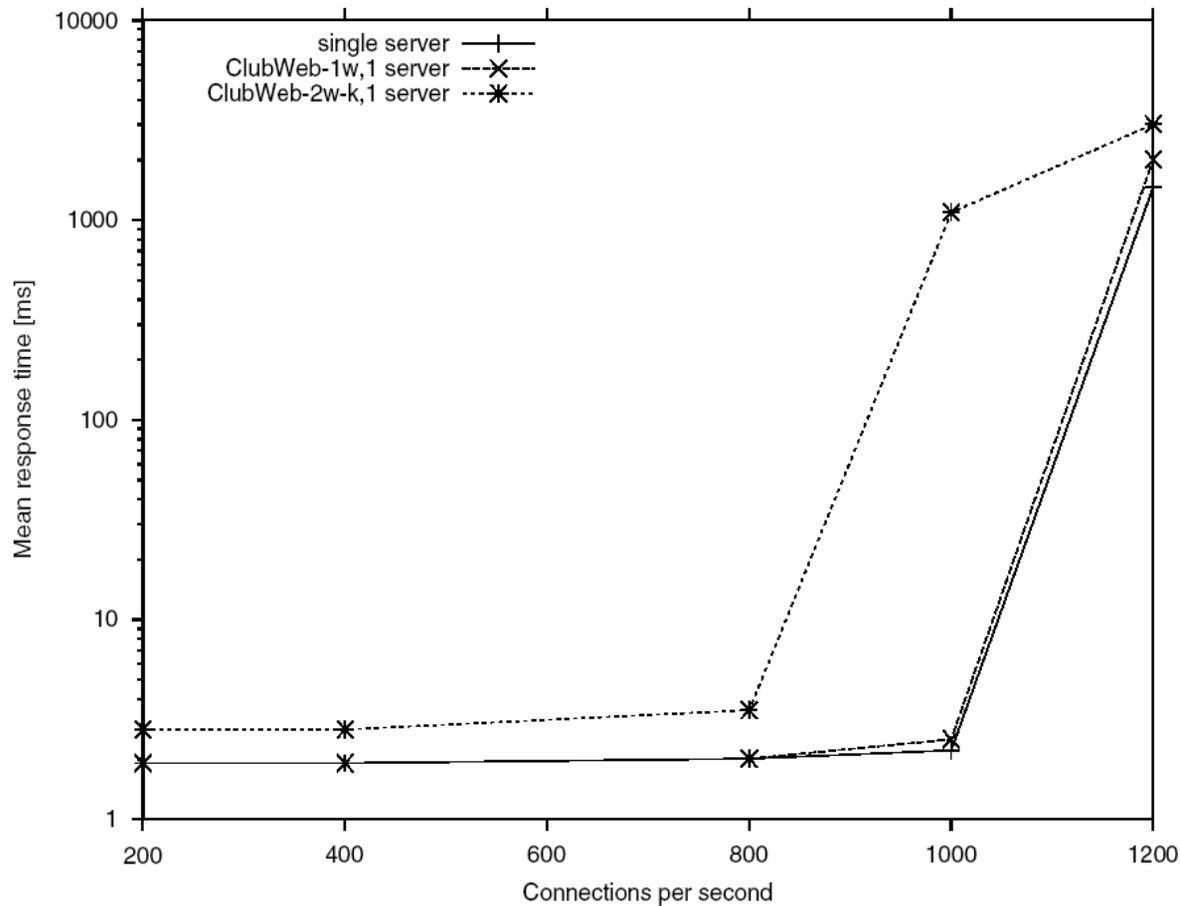


Figure 5. Overheads of one-way vs. two-way mechanism.

Experimental results

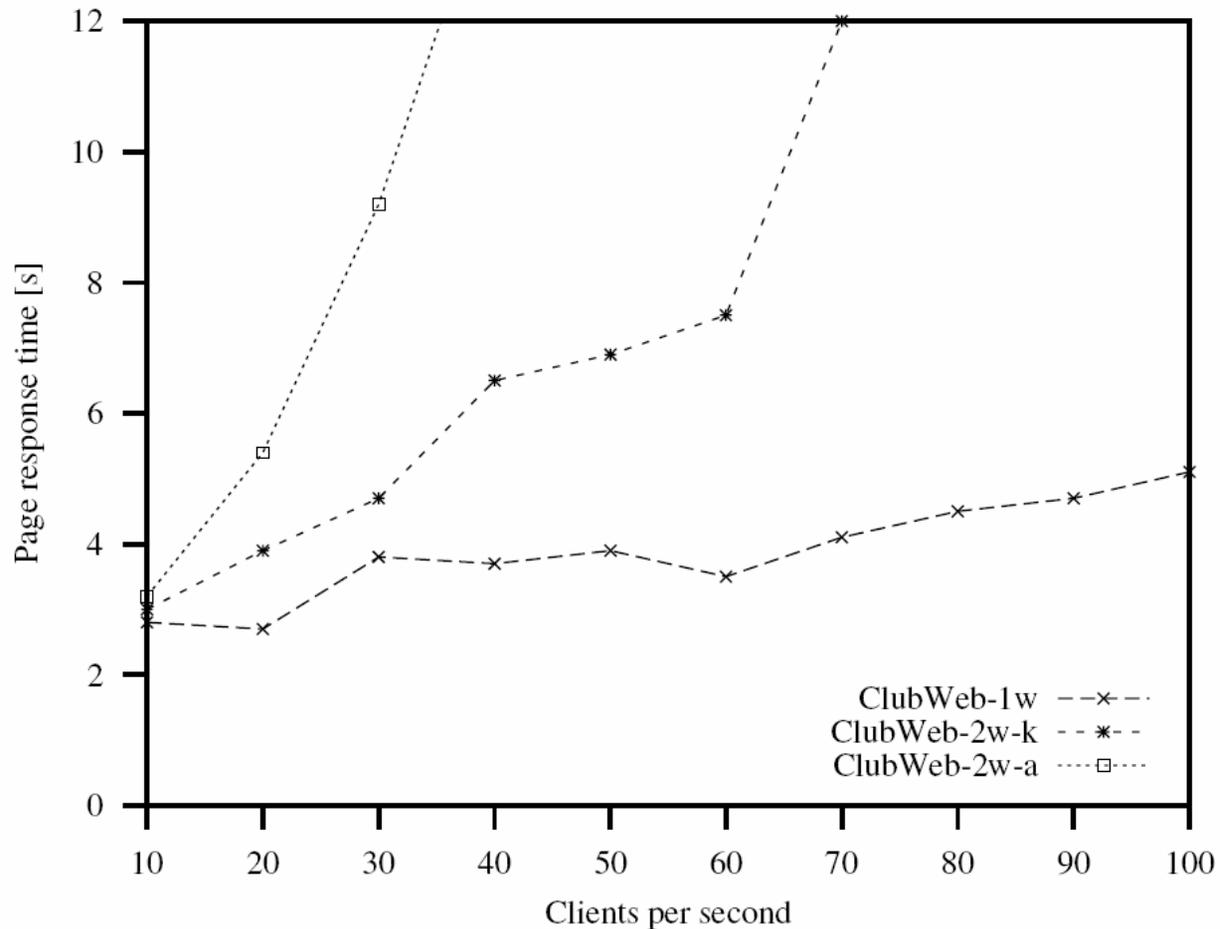


Figure 6. 90-perc. of page response time.

Experimental results

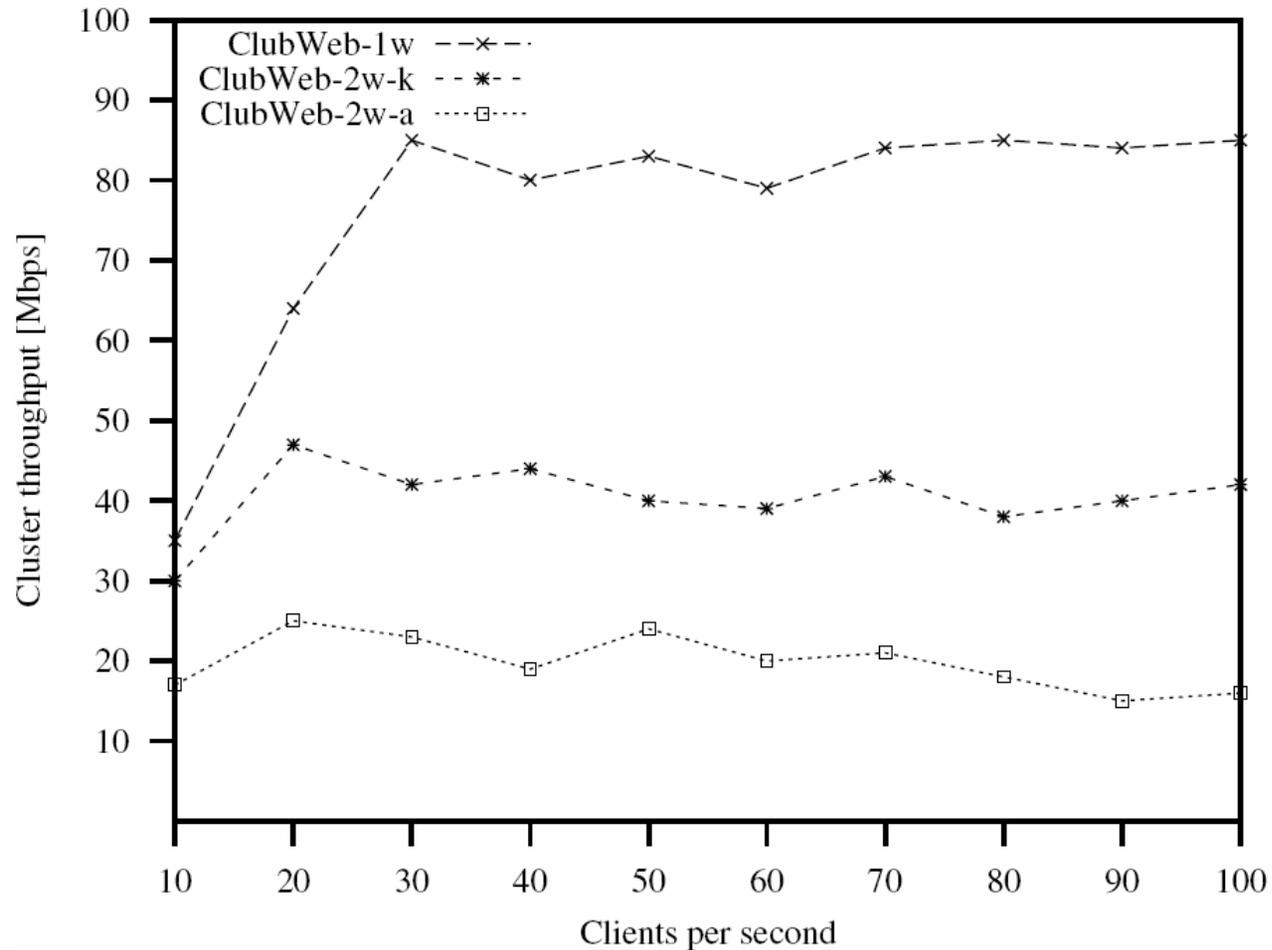


Figure 7. Throughput of the Web cluster.

Conclusions

- We have presented the high level design of a one-way and a two way Web cluster, that are implemented at the kernel level.
- We show that with current off-the-shelf PC hardware it is possible to implement a one-way Web cluster that performs very well