

# Benchmarking

## Mini Project – Benchmark the performance of a Web server using WebBench

TA: Tsung-Haun Cheng (鄭宗寰)

### **Important Dates:**

E-mail your report: 5/11 (Mon.) 23:59

DEMO your report: 5/12(Tue.) 15:00~18:00

TA Office: EECS (電資大樓) 701

E-mail: [raijin@cs.nctu.edu.tw](mailto:raijin@cs.nctu.edu.tw)

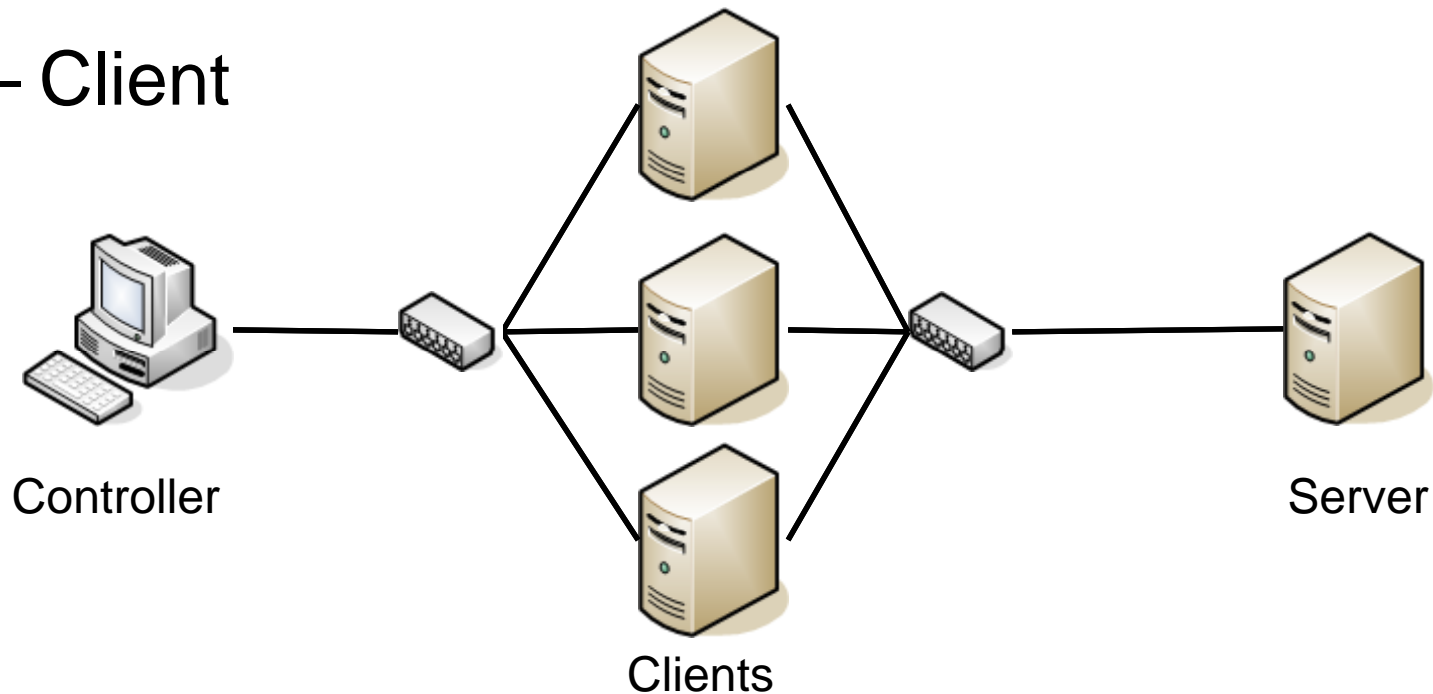
Contact: #56667-13 / 0919-848292

# Project Goals

- Be familiar with the usage of WebBench
- Learn to install and configure a Web server
- Study the performance of a Web server in various conditions

# Prerequisite

- **Web server**
  - Apache in Unix or IIS in Windows
- **Webbench (Windows)**
  - Controller
  - Client

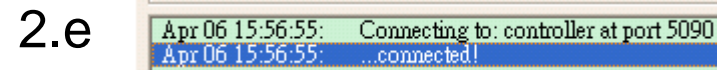
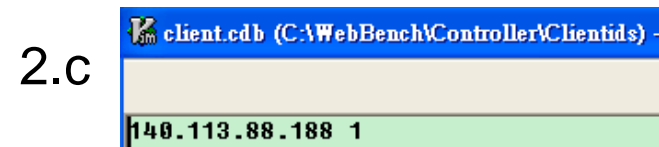
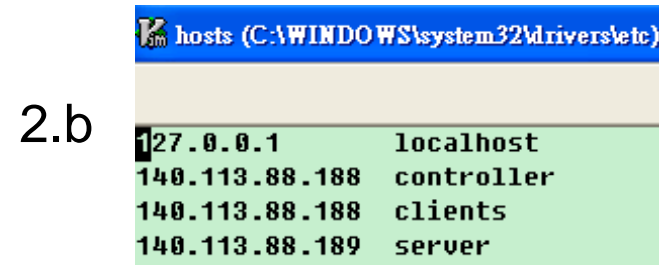


# 1. Setup Web Server

- a. Install a Web server to be tested
- b. Expand and install the standard workload in the document root of the server

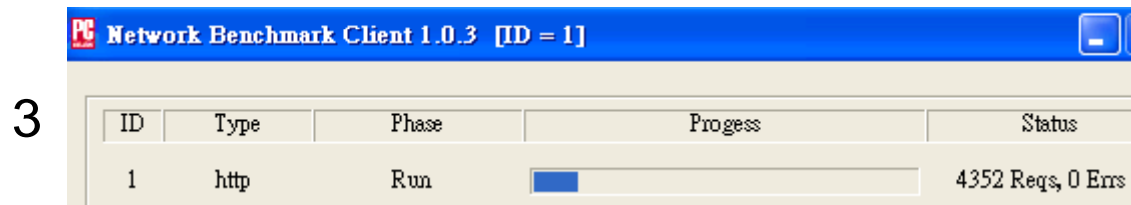
# 2. Setup one controller and at least one client (may be on the same host).

- a. Install controller and clients
- b. Edit Host file
- c. Edit CLIENT.CDB on controller
- d. Run controller and wait clients
- e. Run clients
- f. Select test suite "STATIC.TST"



# 3. Run

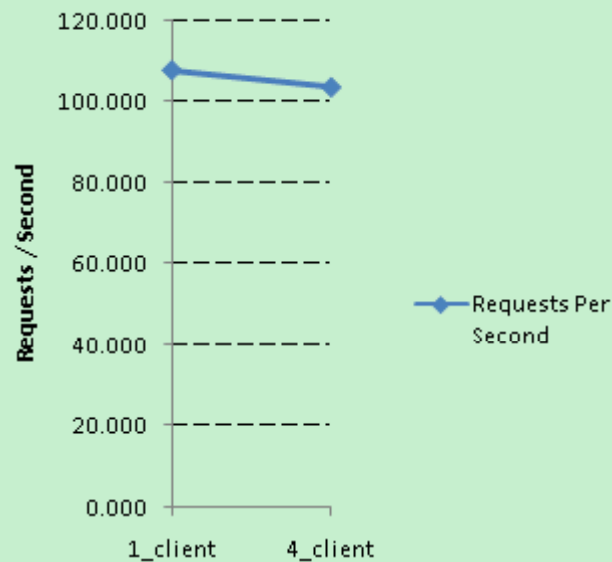
# 4. View Result



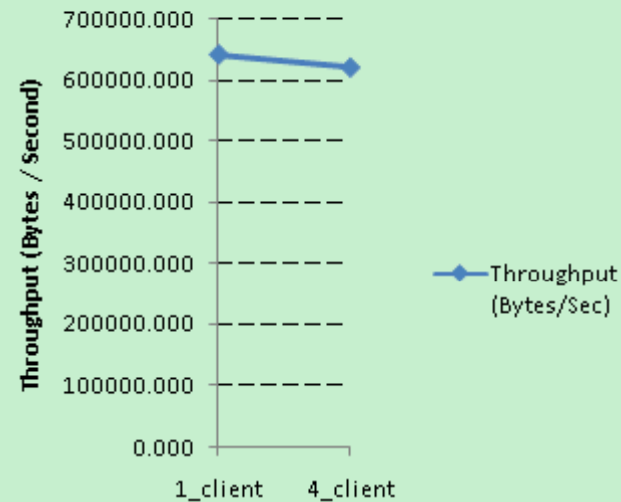
# View Results

Mix Name	Requests Per Second	Throughput (Bytes/Sec)	Test Information
1_client	107.700	643079.000	WebBench(TM) 3.0
4_client	103.575	622265.125	Start Suite: Mon Apr 06 16:02:44 2009
			Finish Suite: Mon Apr 06 16:02:44 2009
			Comments:

Overall WebBench  
Requests/Second



Overall WebBench  
Throughput  
(Bytes/Second)



# Edit test suite

The image shows the WebBench 5.0 Controller interface. The 'Suites' menu is open, showing options: Add..., Remove, Clear, Create/Edit..., and Edit... A blue arrow points from the 'Edit...' option to the 'Mixes in Suite' dialog box.

The 'Mixes in Suite' dialog box has a title bar: **Mixes in Suite - C:\WebBench\Controller\Suites\Webbench\static.tst**

Inside the dialog, there is a 'Suite List' section with a text box containing: `C:\WebBench\Controller\Suites\Webbench\static.tst`

Below the text box is a checkbox:  Save suite list when done

On the left side of the dialog, there are buttons: Add, Remove, New, Edit, Copy, Reorder, and Delete.

On the right side, there is a table with the following data:

Mix Name	Number of Clients	Number of Engines	Engine Types
1_client	1	1	http
4_client	4	4	http
8_client	8	8	http
12_client	12	12	http
16_client	16	16	http
20_client	20	20	http
24_client	24	24	http
28_client	28	28	http

At the bottom of the dialog are 'OK' and 'Cancel' buttons.

# Questions – (1/2)

1. What is the maximum throughput in the following conditions?
  - Tune the percentage of HTTP/1.0 traffic in the Mix definition window for 0%, 50%, 100%.
  - Tune the number of engines per client for 10, 20, 30
  - Tune the receive buffer size for 1KB, 2KB, 4KB.
  - Tune your web server (choose one directive)
    - Ref: <http://httpd.apache.org/docs/2.0/mod/directives.html>
  - (Bonus) Turn on pipelining option to see how pipelining accelerate the performance.
  - (Bonus) If you have a proxy, you can test the performance connections through it

# Questions – (2/2)

2. How does HTTP/1.1 increase the performance? Can you analyze it?
3. Is only one WebBench client enough to saturate your server? If not, how many are needed in your estimation.
4. How does the receive buffer size affect the performance?
5. Is server response time inversely proportional to the number of requests per second?
6. How much time have you spent in this project?
7. (Bonus) How does pipelining affect the performance? Discuss it.
8. (Bonus) Is the proxy or the Web server the bottleneck in your benchmark?



# Report Requirement

- Show the screenshots of the results (in graphs) for each benchmark
- Describe the meaning of each operation
  - Including web server and webbench
- Mail your report(doc) to [raijin@cs.nctu.edu.tw](mailto:raijin@cs.nctu.edu.tw)

# Benchmarking

## Term Project – Bottleneck analysis in virus-scanning mail proxy

TA: Tsung-Haun Cheng (鄭宗寰)

### **Important Dates:**

E-mail your report after presentation

TA Office: EECS (電資大樓) 701

E-mail: [raijin@cs.nctu.edu.tw](mailto:raijin@cs.nctu.edu.tw)

Contact: #56667-13 / 0919-848292

# Term Project – Bottleneck analysis in virus-scanning mail proxy

- Goal
  - Set up a virus-scanning mail proxy
  - Be familiar with the use of gprof
  - Be familiar with the use of email test tool
  - Study the program flow of ClamAV
  - Learn to analyze the bottleneck in a typical proxy server

# Procedure

- Install email test tool on Windows PC, and Postfix, AMaViSd and ClamAV on a Linux PC.
- Compile ClamAV for gprof.
- Send emulated email with various files (compressed, text, binary...) from the email test tool.
- Try to find out which functions are most frequently called and most time-consuming in virus scanning. (You may need to track the program flow of ClamAV).

# Questions

- List the top ten most frequently called functions and their percentage.
- List the top ten time-consuming functions and their percentage.
- Please examine the most time-consuming function and explain why.
- How much is the difference in virus-scanning time for text files and binary files?
- How much time is the decompression compared with virus scanning?
- How much is the execution time increased after turning on the `-pg` option for `gprof`?
- How much time have you spent in this project?