SSL Accelerating Test Bench

SSL accelerating Test Method

Stefan Deelen & Maurits van der Schee (master students SNE at the UvA)

Supervised by: Jan Meijer (Surfnet)

Contents

- Objectives
- Test Method
- Scope
- Types of testing
- Other tests
- Conclusions & Future Work
- Questions

Objectives

Finding a test method which answers these questions:

1) What is the actual added value of an accelerator to a web server?

2) How to compare accelerator performance?

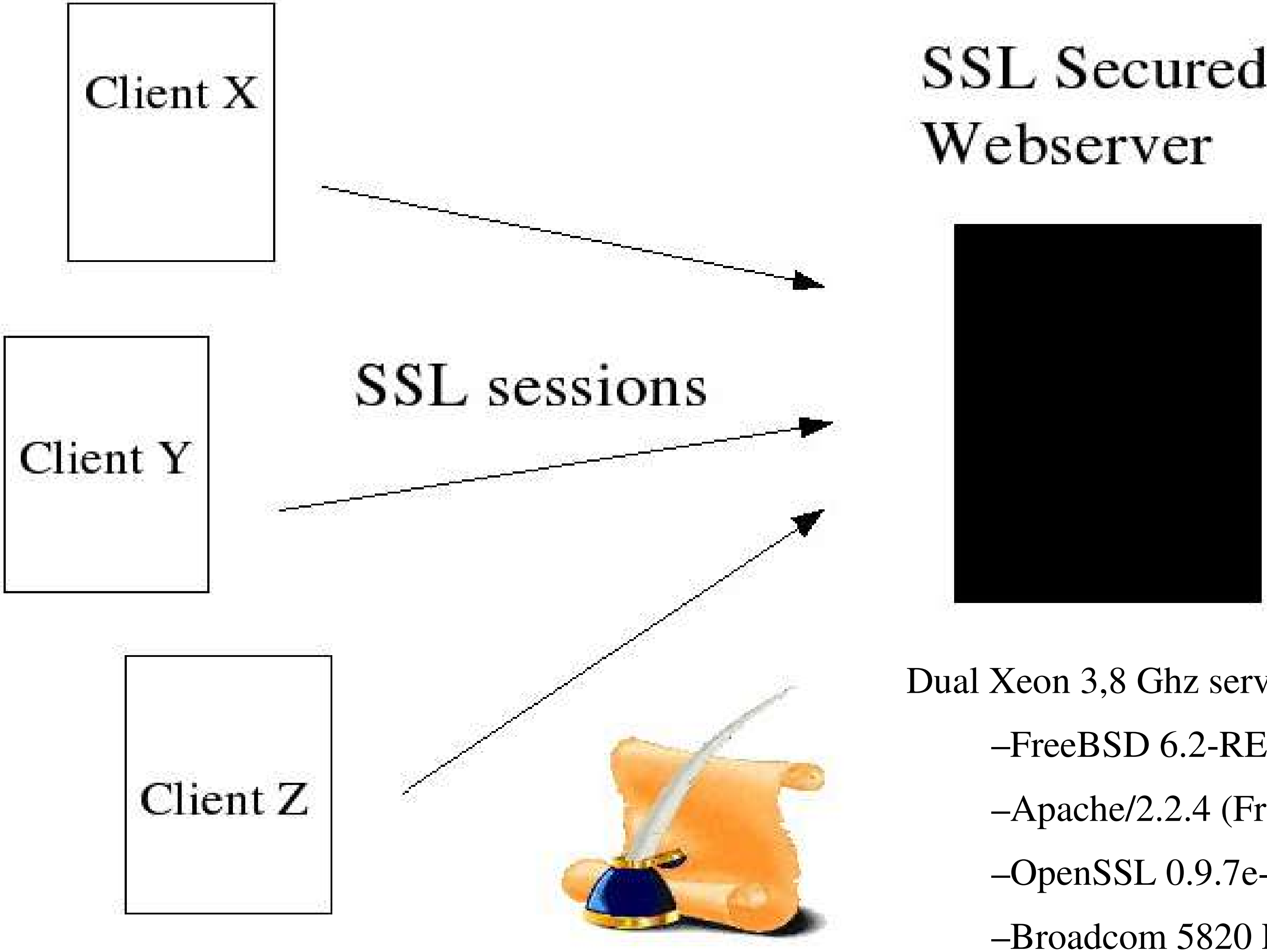
Our successful test approach:

Comparative testing

Performance with accelerator **X** vs. performance with accelerator **Y**

Web server performance **with** accelerator vs. **without** accelerator

SSL performance metric =
Max. number of unique SSL handshakes per second (TPS)



SSL Secured Webserver

Client X

Client Y

SSL sessions

Client Z

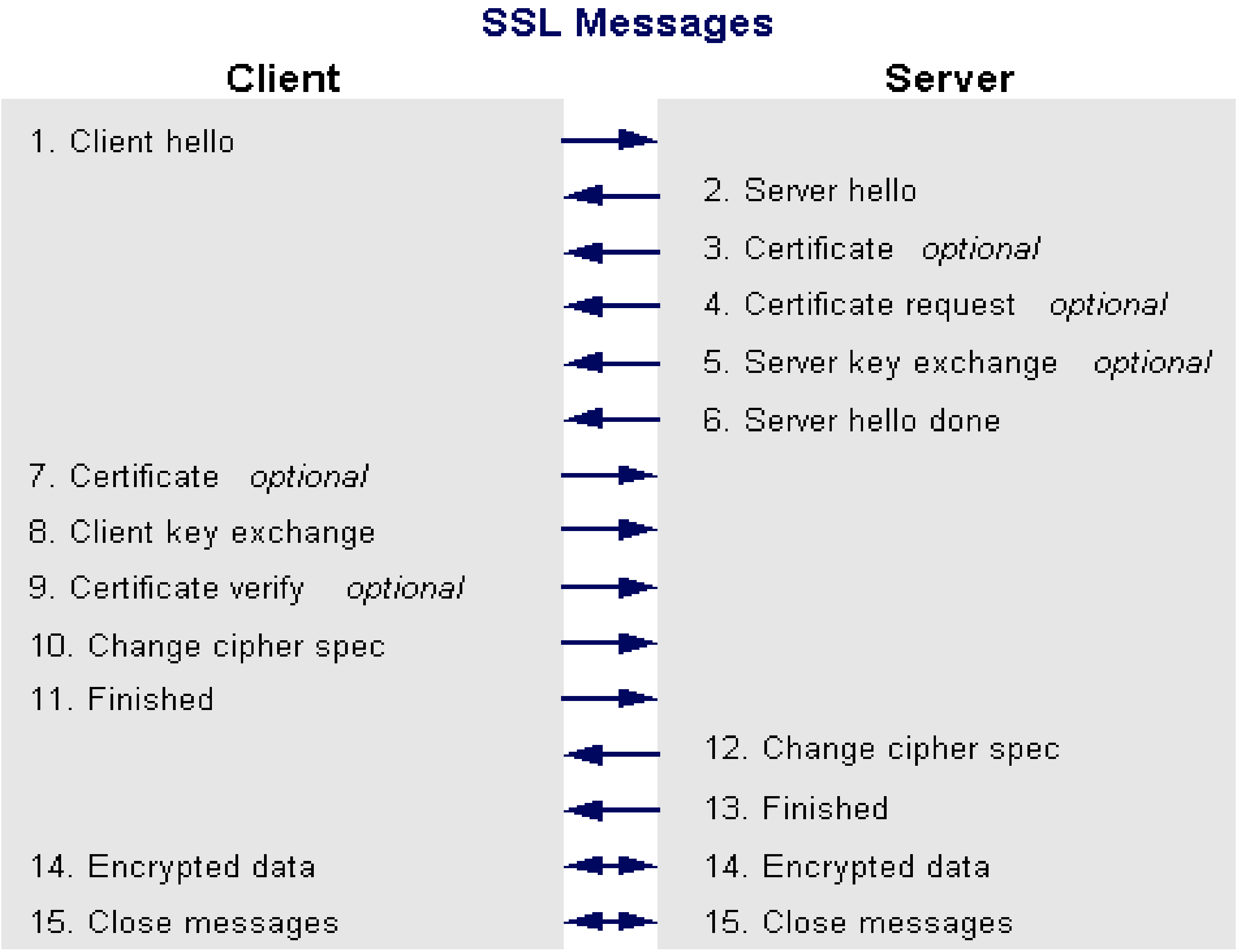
Dual Xeon 3,8 Ghz server

- –FreeBSD 6.2-RELEASE
- –Apache/2.2.4 (FreeBSD)
- –OpenSSL 0.9.7e-p1
- –Broadcom 5820 PCI crypto accelerator

Three clients running linux

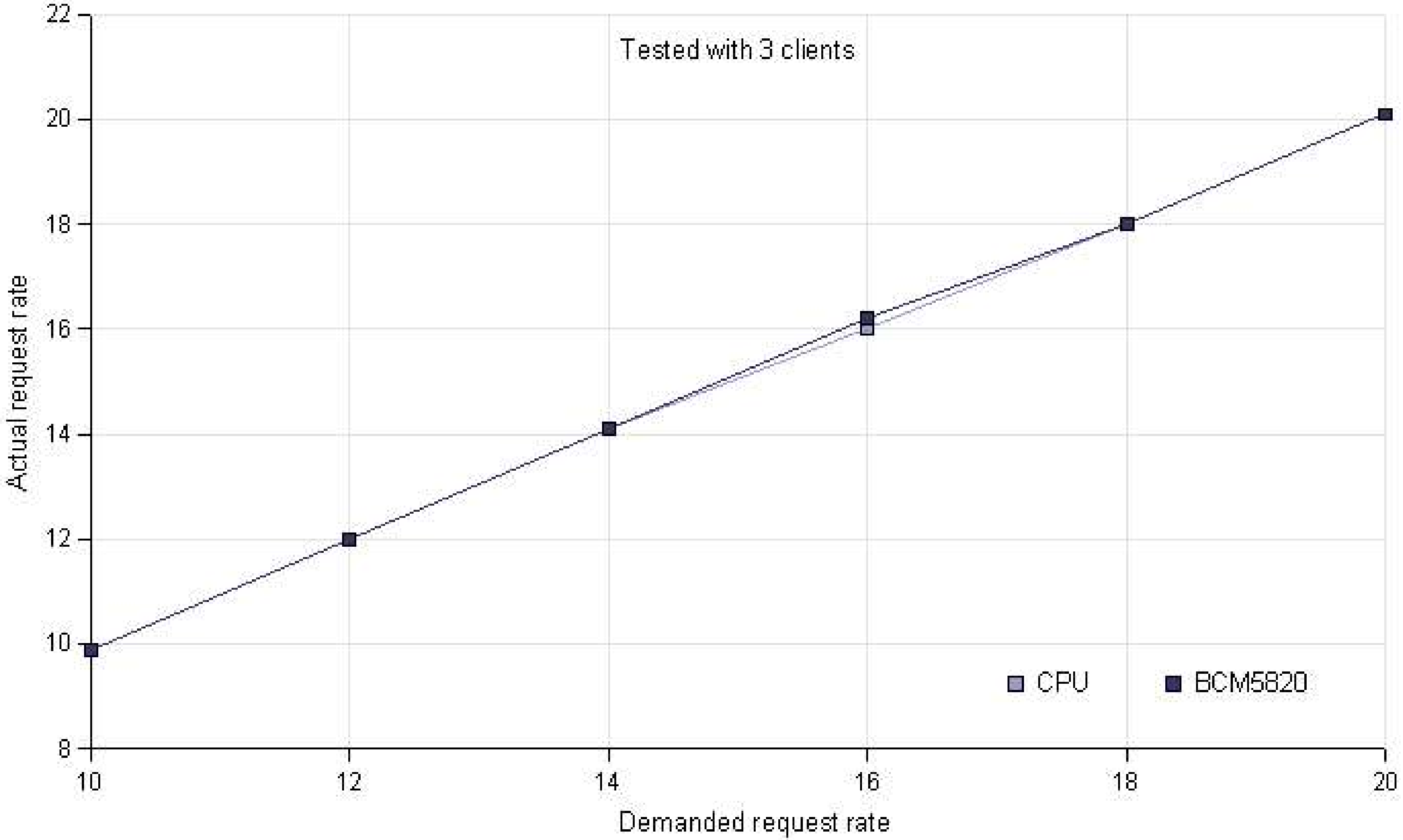
- – Ab, Httperf and autobench software
- – Connected through switched gigabit

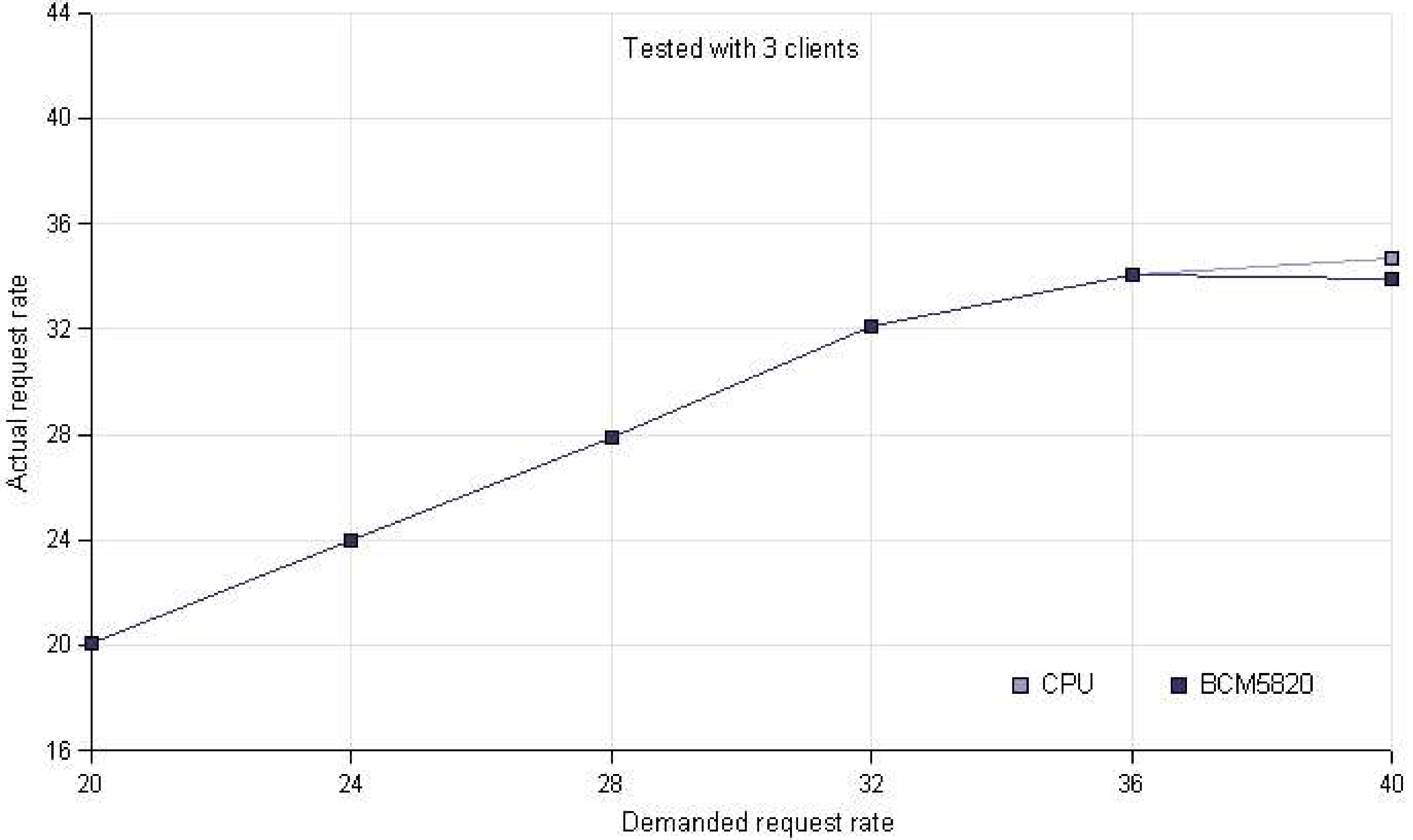
SSL in-balance: How many clients?

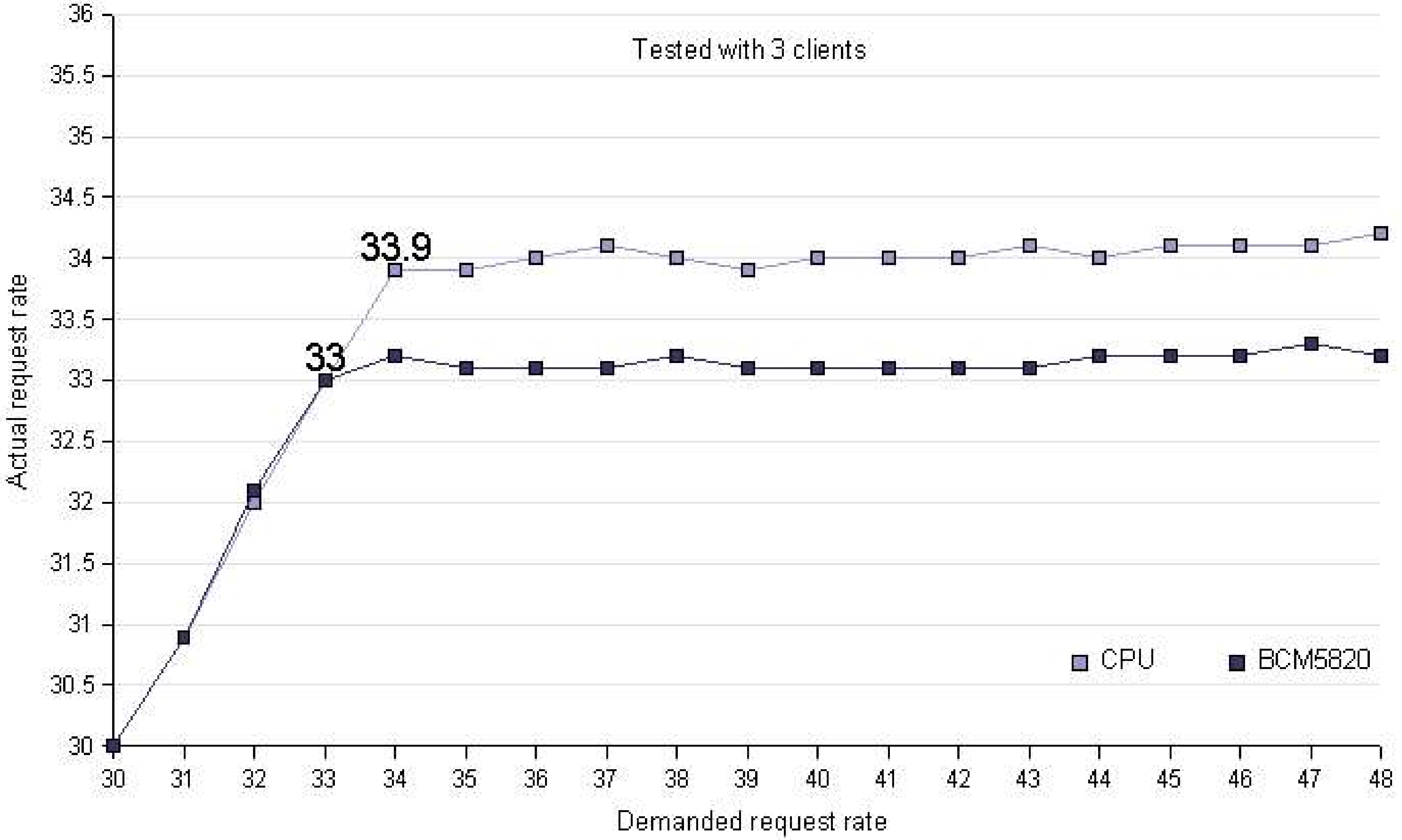


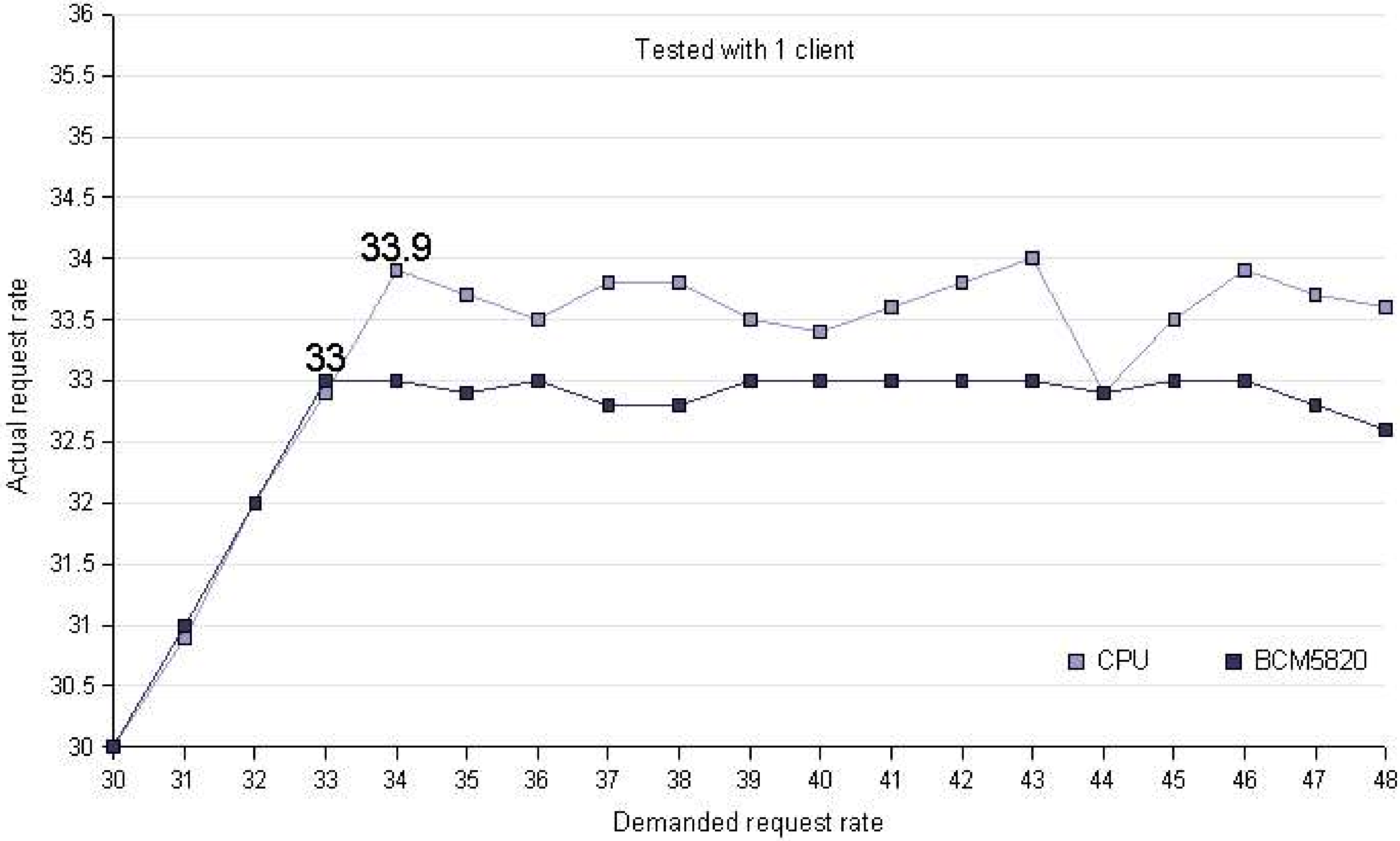
SSL Messages

Client		Server
1. Client hello	→	
	←	2. Server hello
	←	3. Certificate *optional*
	←	4. Certificate request *optional*
	←	5. Server key exchange *optional*
	←	6. Server hello done
7. Certificate *optional*	→	
8. Client key exchange	→	
9. Certificate verify *optional*	→	
10. Change cipher spec	→	
11. Finished	→	
	←	12. Change cipher spec
	←	13. Finished
14. Encrypted data	↔	14. Encrypted data
15. Close messages	↔	15. Close messages


Tested with 3 clients
Actual request rate
8
10
12
14
16
18
20
22
10
12
14
16
18
20
Demanded request rate
CPU
BCM5820


Tested with 3 clients
Actual request rate
44
40
36
32
28
24
20
16
20
24
28
32
36
40
Demanded request rate
CPU
BCM5820


Tested with 3 clients
Actual request rate
Demanded request rate
36
35.5
35
34.5
34
33.5
33
32.5
32
31.5
31
30.5
30
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
33.9
33
CPU
BCM5820



Tested with 1 client
Actual request rate
36
35.5
35
34.5
34
33.5
33
32.5
32
31.5
31
30.5
30
33.9
33
Demanded request rate
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
CPU
BCM5820

Test Operation

1. Use Autobench to do a quick test to find the saturation point
2. “Zoom into” the saturation point for more accurate results.
3. Add or remove clients to verify you hit a server limit

Research Scope

- Open source operating system
- OpenSSL
- SSL handshake (RSA cipher)
- Apache 2.2
- Benchmark tools “Autobench and Httperf”

Types of testing

- Black box
  - – Testing focused on software’s external attributes and behavior.
  - – From a user’s point of view.
- White box
  - – Testing with full knowledge of the algorithms, internal states, architectures, etc.
  - – From a developers point of view.

Gray box testing

- “Tests designed based on the knowledge of algorithms, internal states, architectures, or other high level descriptions of program behavior”. – Doug Hoffman
- Needed because black and white box testing do not allow for balanced testing
- Integral to the effective testing of Web applications

Other testing

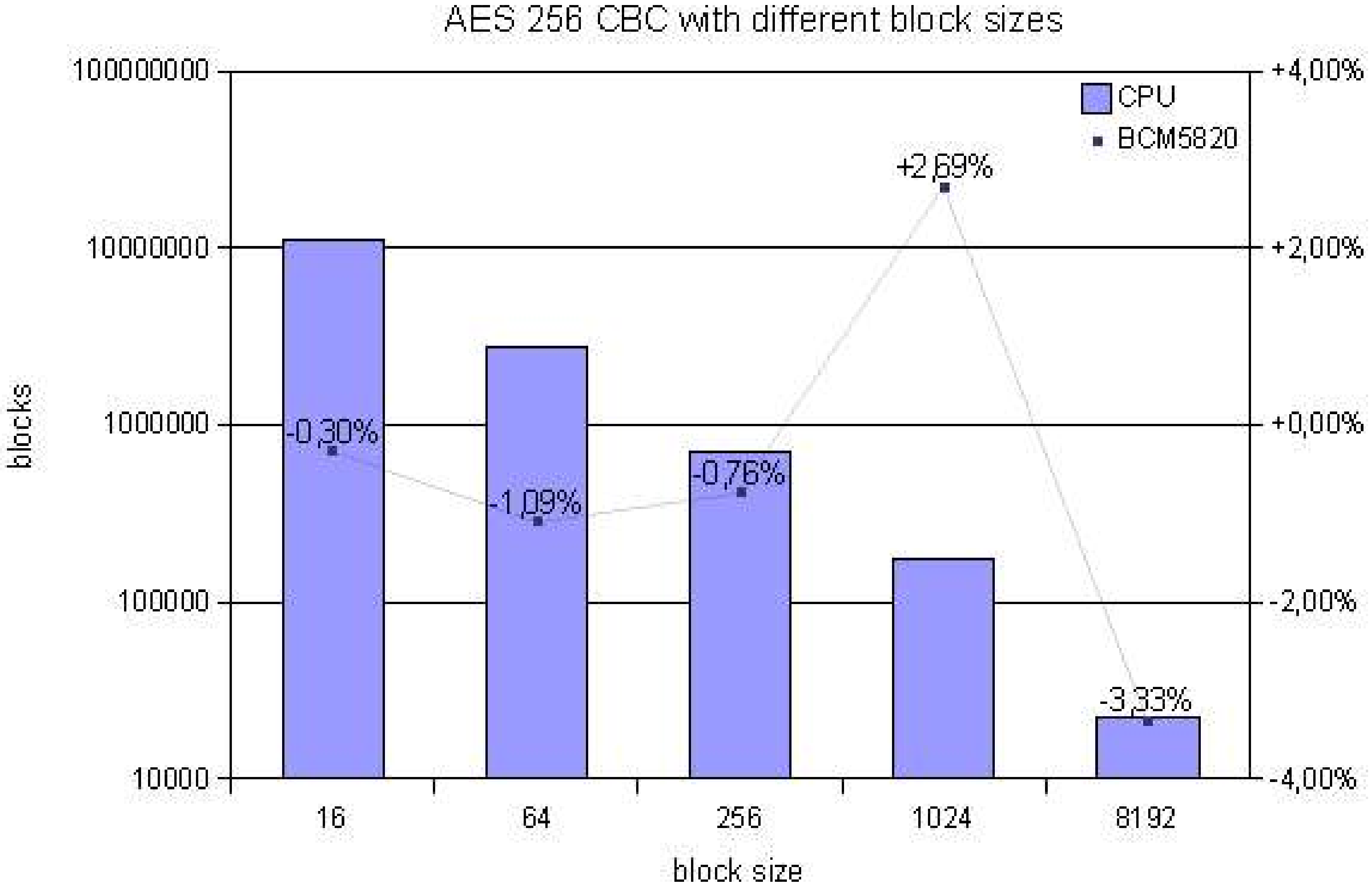
2. OpenSSL speed benchmark
    - Test the performance of the crypto library used by Apache
3. Single session
    - Test the response time of a single request

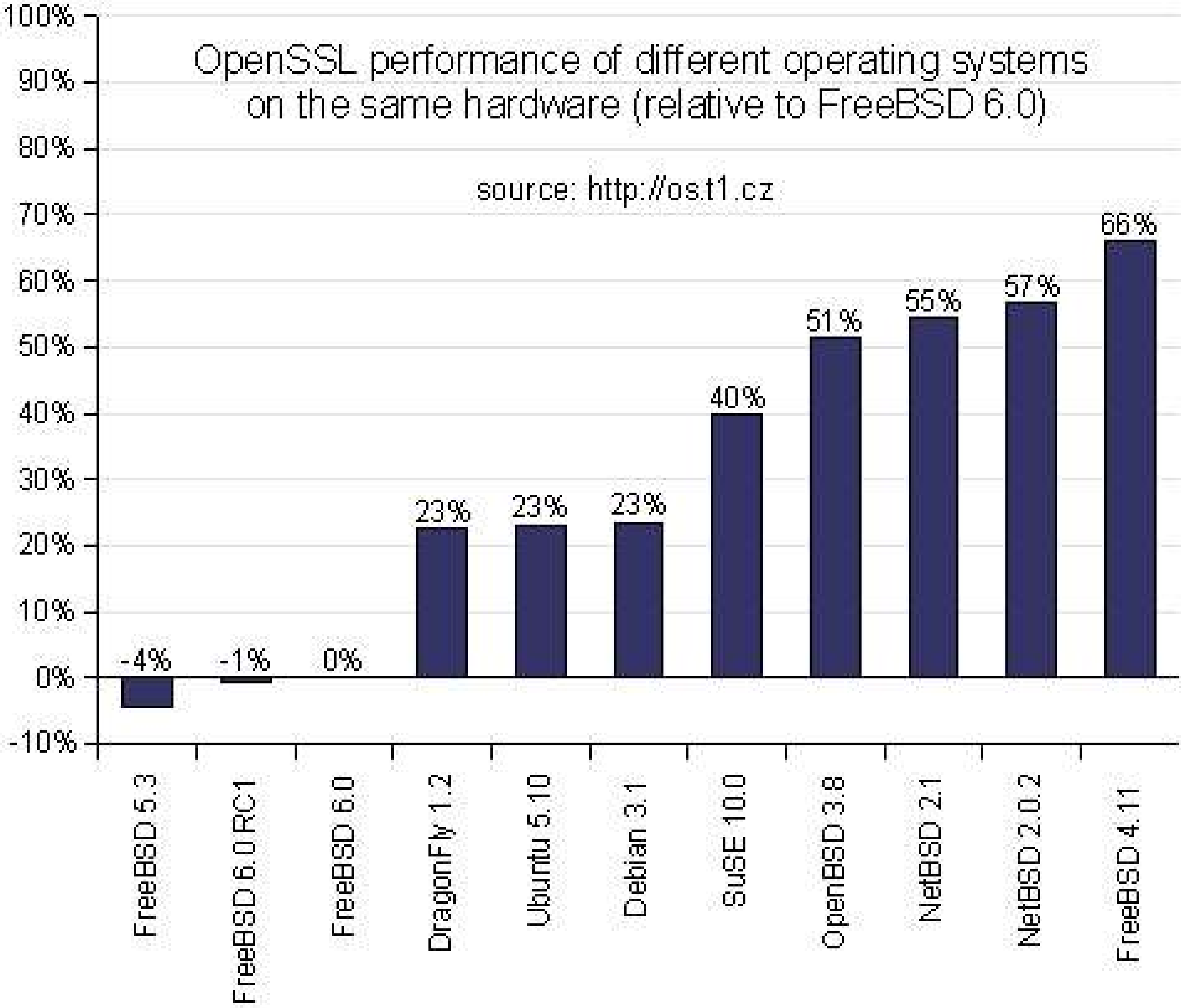
OpenSSL speed results

```
[root@test ~]# openssl speed rsa1024
Doing 1024 bit private rsa's for 10s:
  2989 1024 bit private RSA's in 9.97s
Doing 1024 bit public rsa's for 10s:
  48265 1024 bit public RSA's in 9.99s
timing function used: getrusage
rsa 1024 bits:
  sign      verify    sign/s   verify/s
  0.0033s   0.0002s   299.8    4832.8
```

Algorithm

- Accelerators may be optimized for certain algorithms and block sizes
- Algorithm balance can influence preformance
  - RSA vs DSA balance
  - Rebalanced RSA
- Driver may not be implemented optimal
  - CRT parameters used or not


AES 256 CBC with different block sizes
100000000
10000000
1000000
100000
10000
blocks
+4,00%
+2,00%
+0,00%
-2,00%
-4,00%
CPU
BCM5820
-0,30%
-1,09%
-0,76%
+2,69%
-3,33%
16
64
256
1024
8192
block size



OpenSSL performance of different operating systems on the same hardware (relative to FreeBSD 6.0)

source: http://os.t1.cz

OpenSSL speed conclusions

- Results show the actual encrypting performance of a system.
- Results may be heavily influenced by algorithm, driver and operating system
- Results are easy to compare
- But, you are not measuring the “added value” of the accelerator (it is white box)

Single session

- Httperf results are equal to Ab (Apache bench) results
- We measured a 2 ms difference between the situation with and without accelerator
- The handshake takes 7 ms longer (calculated value)
- We are not able to explain the difference

Httperf testing

- We used a 0 byte file to focus on handshake
- We used HTTP 1.0 to avoid keep-alive (and thus connection limits)
- We disabled caching on the client and server side (to simulate connections from different hosts)
- We measured the actual request rate (number of HTTP GET requests per second)

Autobench

- …is a Perl script (OS indepent)
- …automates doing series of Httperf tests
- …has a client/server architecture
- …enables you to do distributed tests
- …produces its results in a graph

Conclusions

- We developed a method that enables easy and comparable tests for SSL accelerators
- Gray box testing is needed to find the actual added value of an accelerator
- Choices in algorithm, operating system and drivers may multiply (!) performance
- Future work may prove this method useful for a wider scope

Future work

- Throughput testing
- Virtual users: script that emulates site visit
- Automated searching for saturation point
- Other (maybe better) testing software
- High performance accelerators and/or other algorithms may require an easy scalable client pool

Questions …?