Automatic and Dynamic Configuration of Data Compression for Web Servers

Eyal Zohar

Yuval Cassuto

YAHOO! LABS

Technion Israel Institute of Technology

November 13, 2014

LISA'14

HTTP Compression

- Most web servers compress content for lower bandwidth and faster response

Sites supporting “gzip” format	92%
Sites supporting “deflate” format	9%
Average download size (compressed only)	22,526
Average uncompressed file	101,786
Compression ratio (best-median-worst)	11%-25%-53%

who: almost everybody

why: bw & overall latency

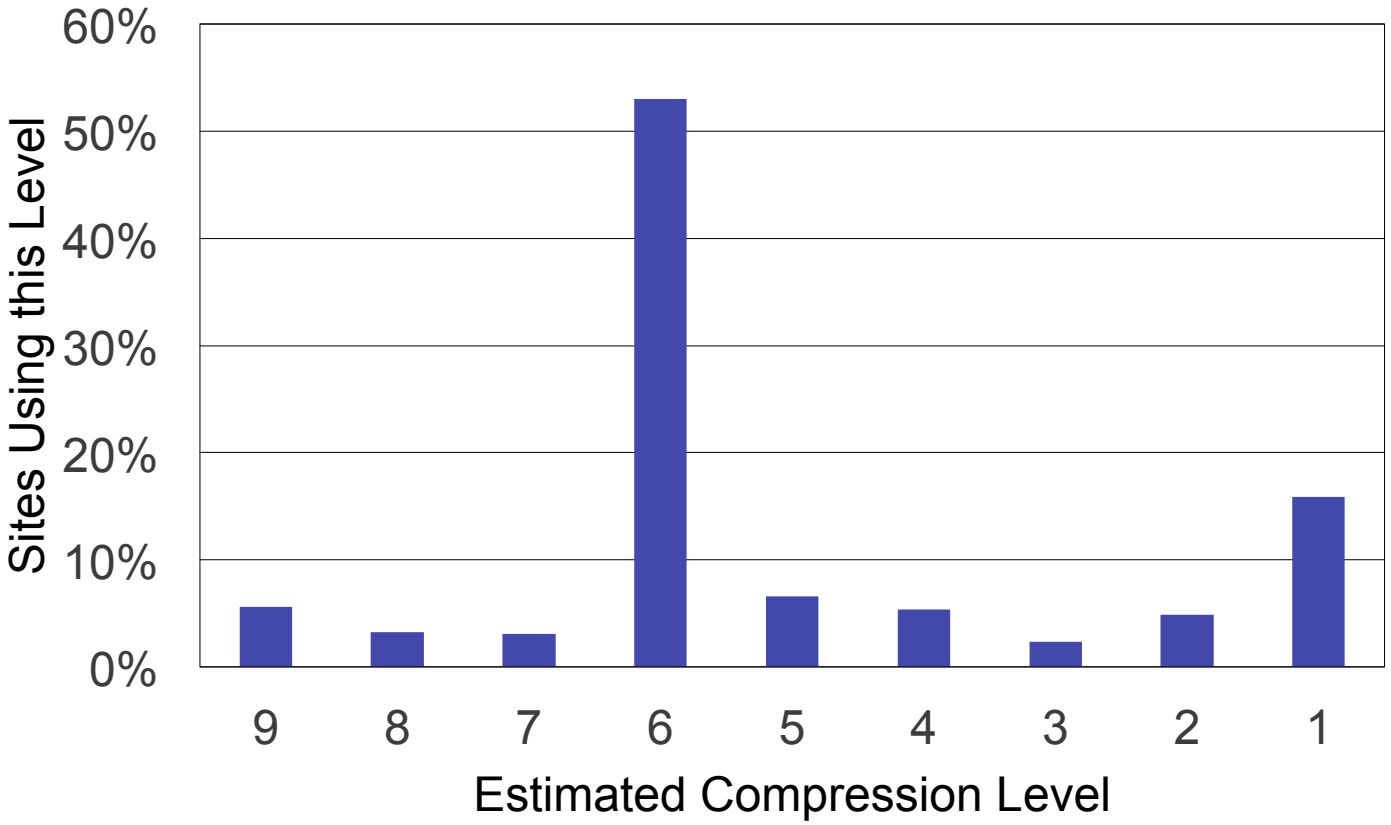
Compression Knobs

1. Enable

2. Set compression level

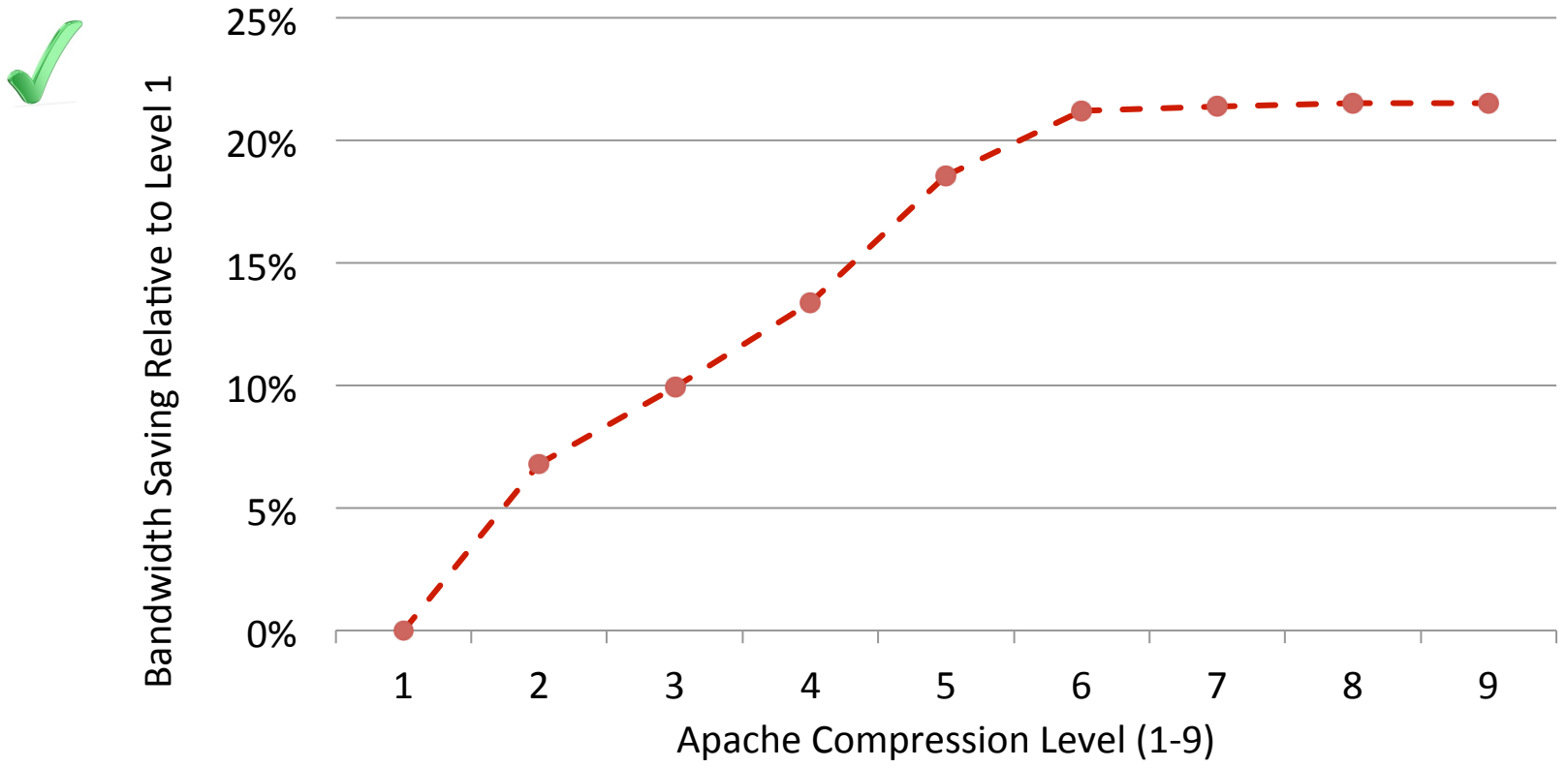
Which Level to Use?

- Top 500 sites – most pick the default (6)



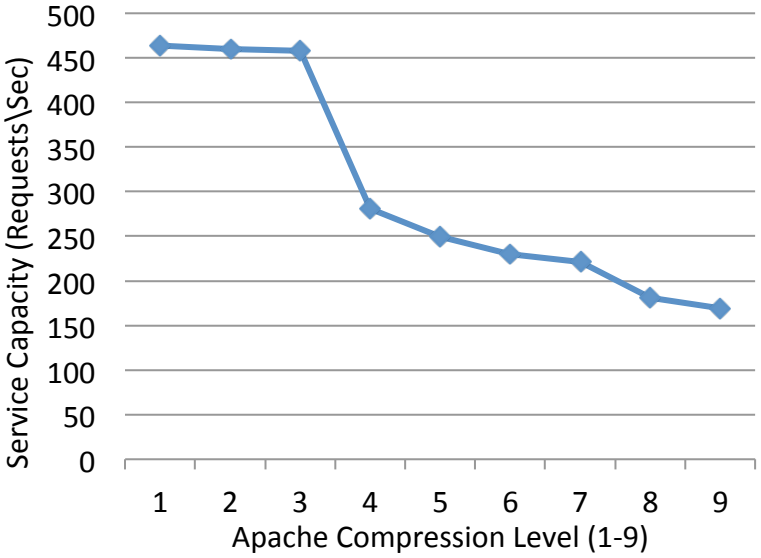
Which Level **Should** Use?

- Higher level ➔ better compression

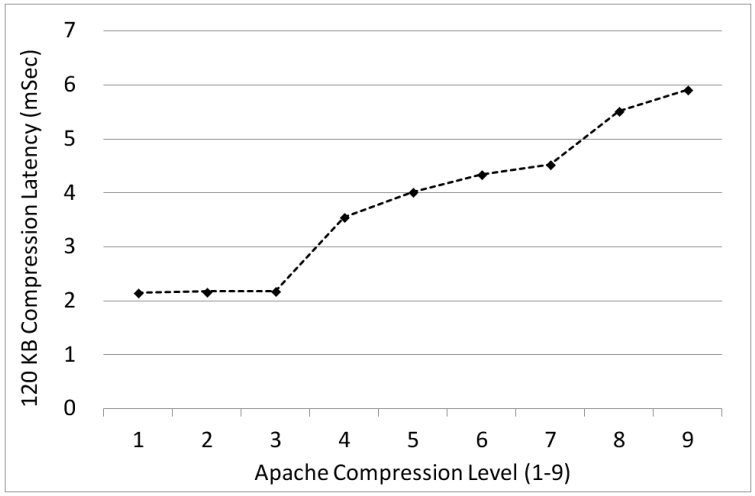


Compression Cost

- Higher level ➔ lower capacity (server)
  ➔ higher latency (user)



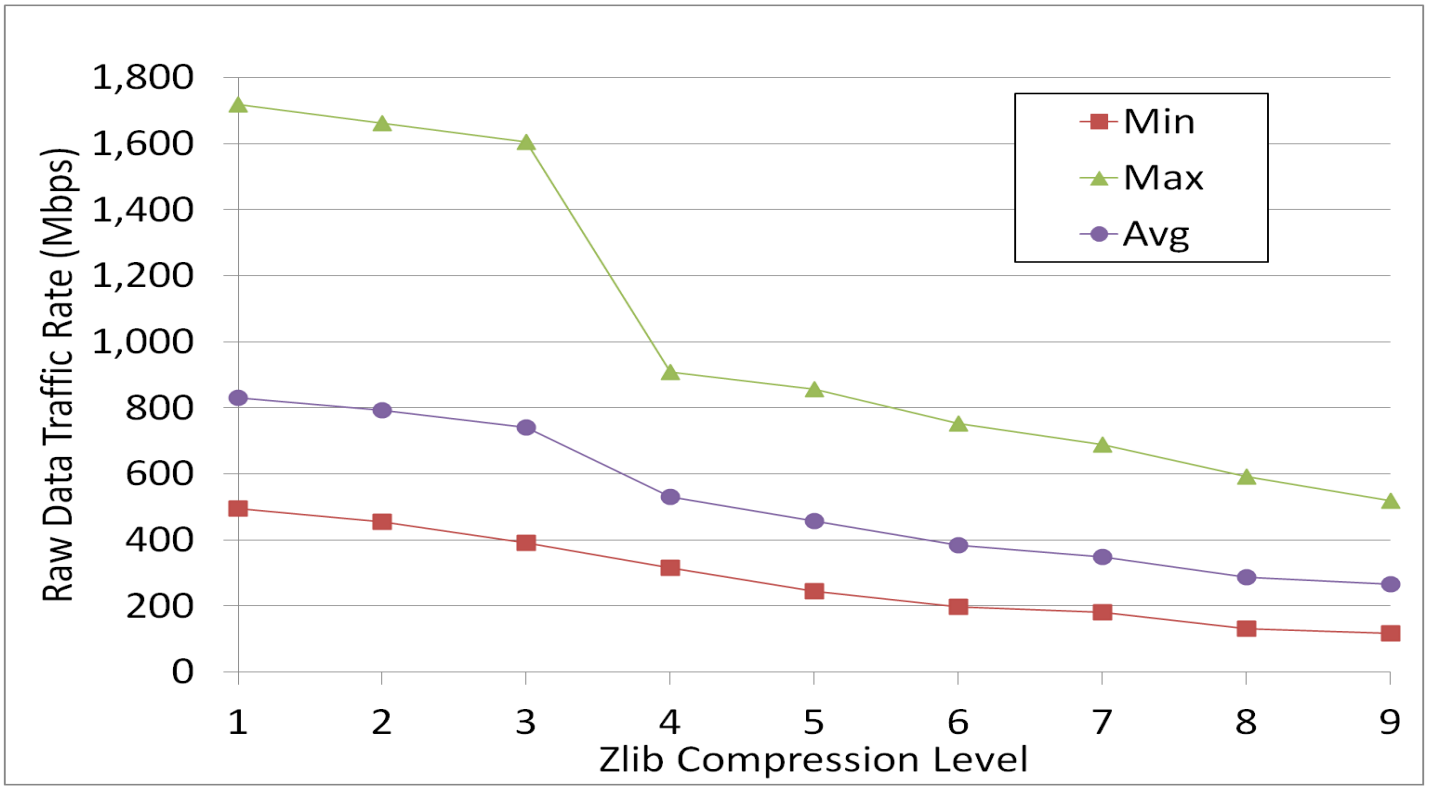
Capacity



Latency

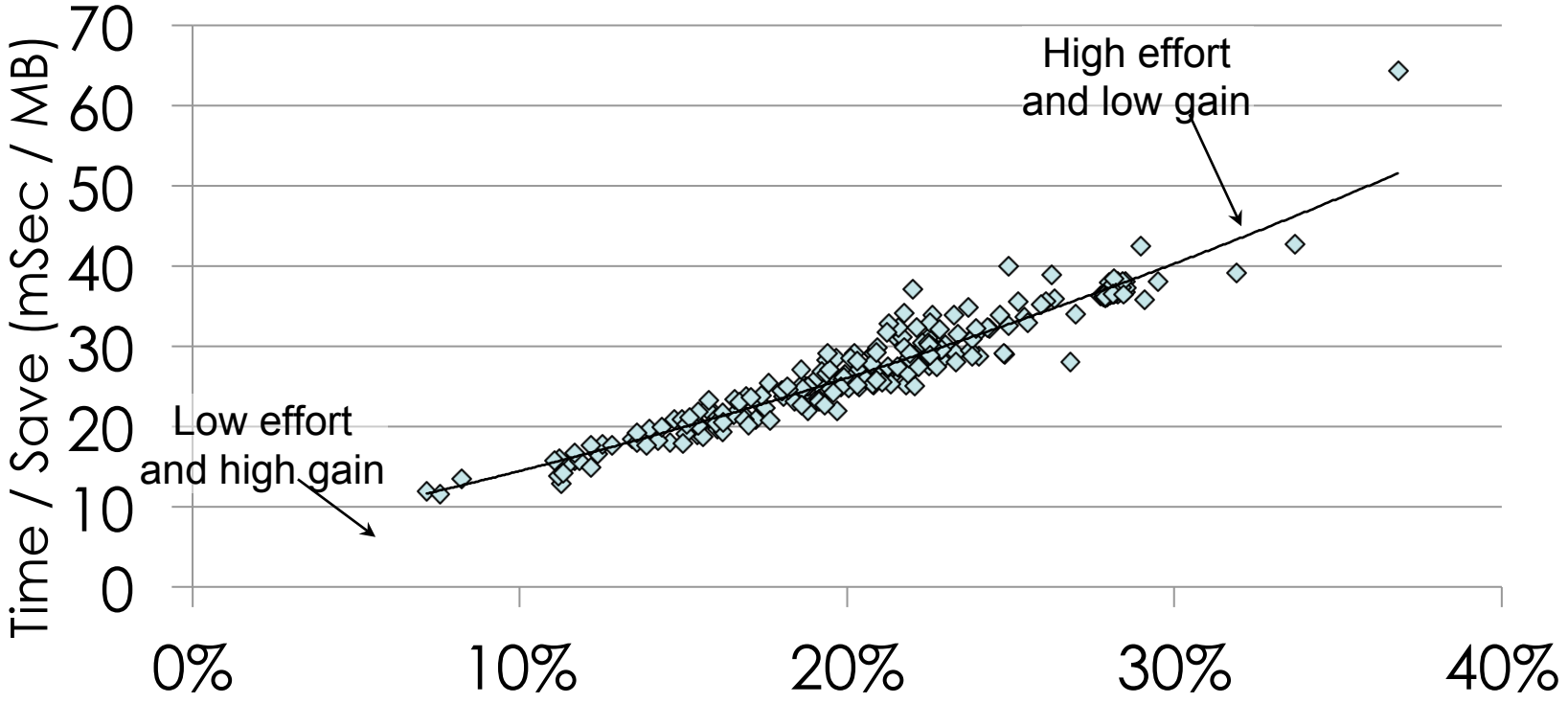
Content Matters

- Compression time depends on content
- All 500 sites, est. capacity (locally):



Performance vs. Effort

- **mSec/MB** – processing time to save 1MB

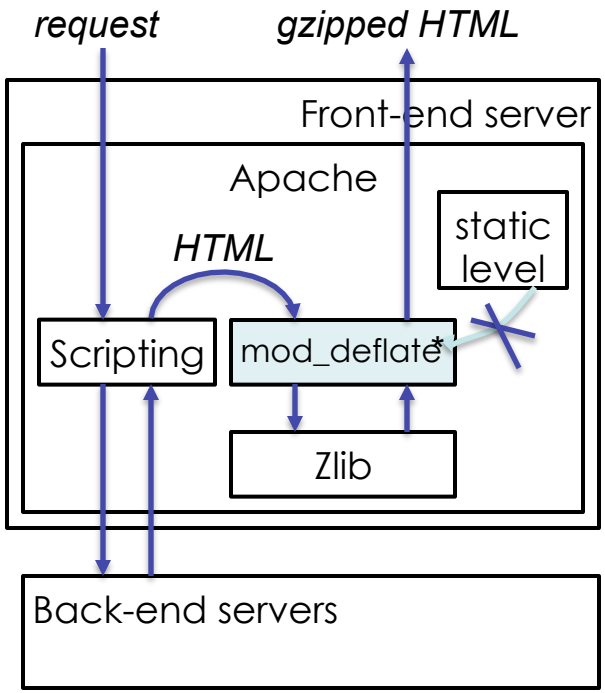


Compression under Dynamic State

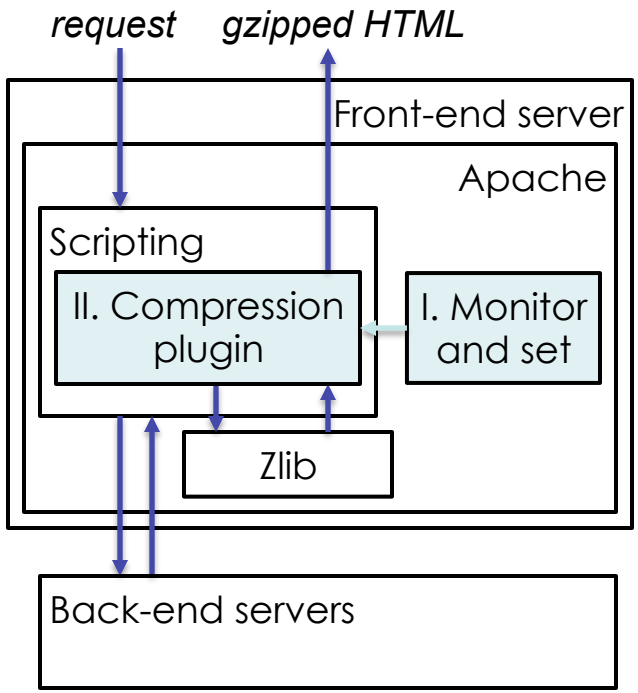
- Demand (popularity/time-of-day/attack)
- Hosting costs
- Hardware availability
- Content size
- Content compressibility
- Background processes (CPU)
- I/O performance
- …

Solution – 1) Auto Adjust

- Elastic level – according to load/latency



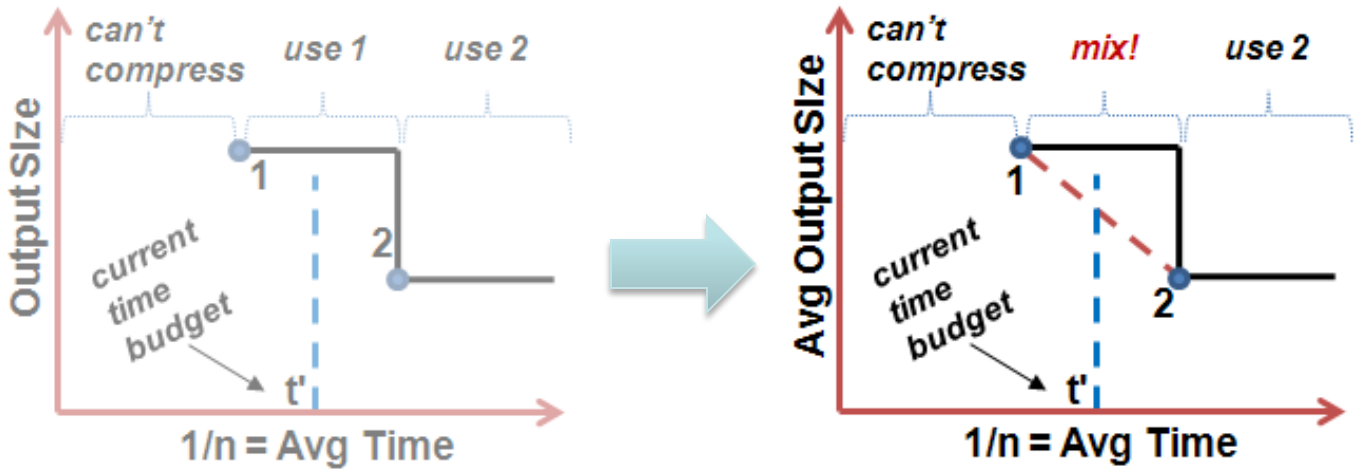
*Yevgeni Sabin, Alexander Chigirintsev ("Project A" 2012)



**Code available here:
http://eyalzo.com

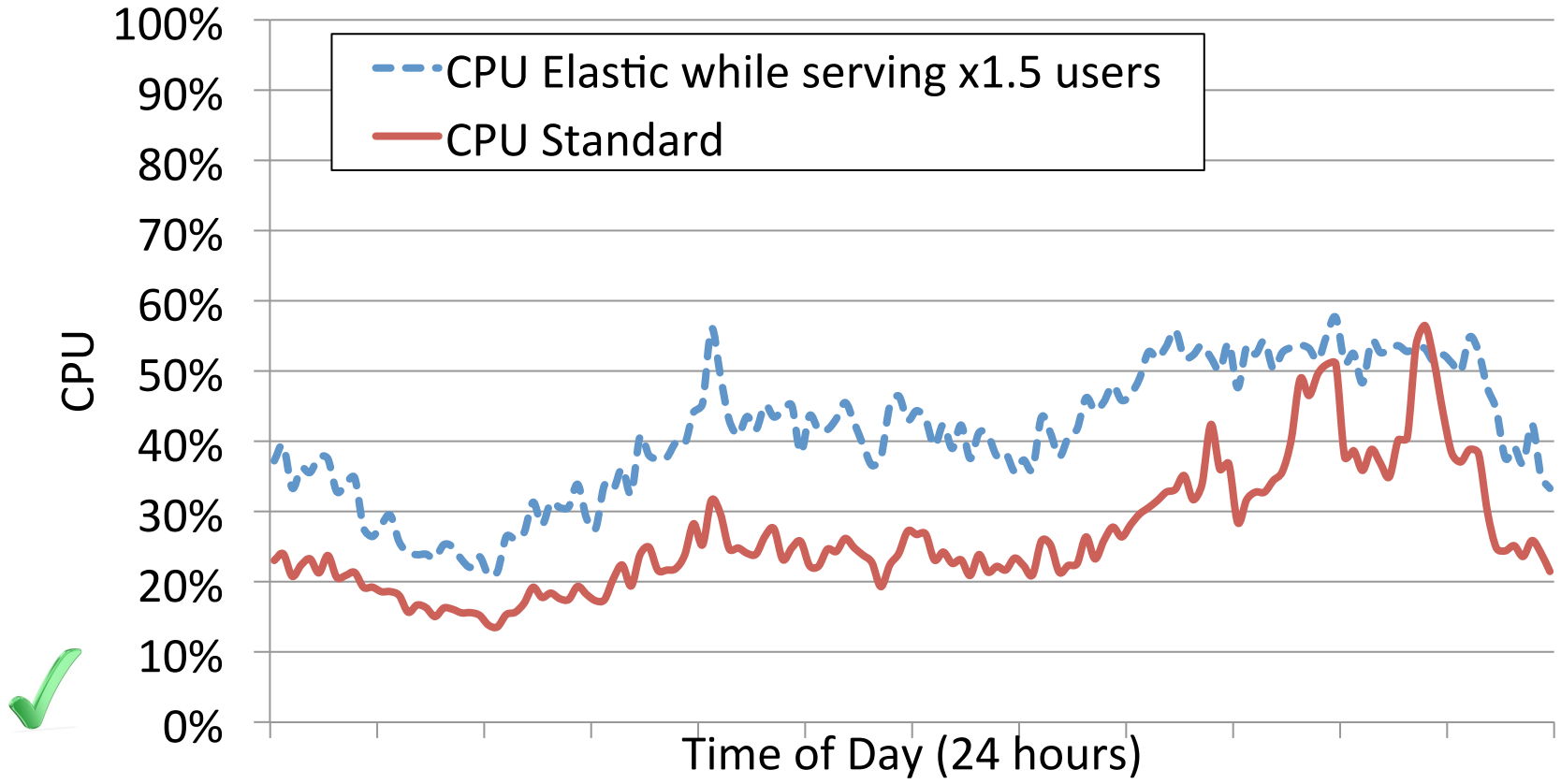
Solution – 2) Mix

- Implementing non-integer levels
  - 1.2 means (80% * level 1) + (20% * level 2)

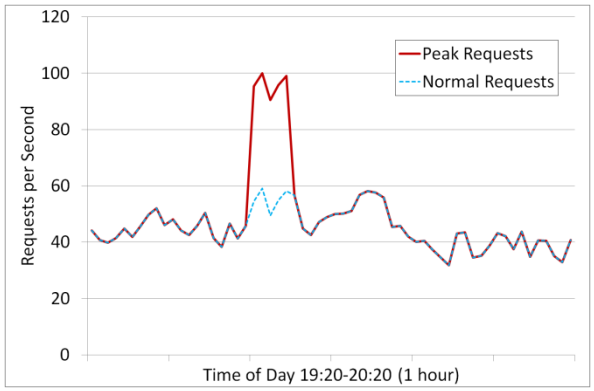


Experiment 1 – Serve More

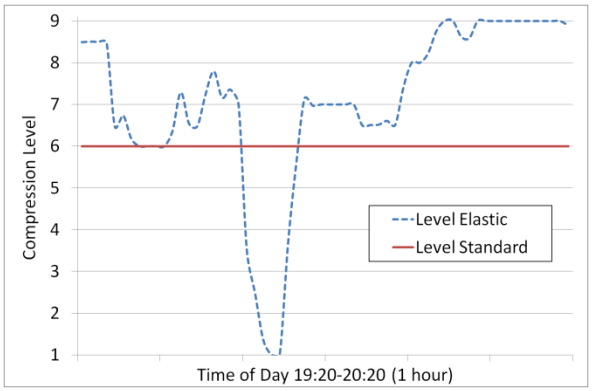
- Serving x1.5 requests



Experiment 2 – DoS Attack



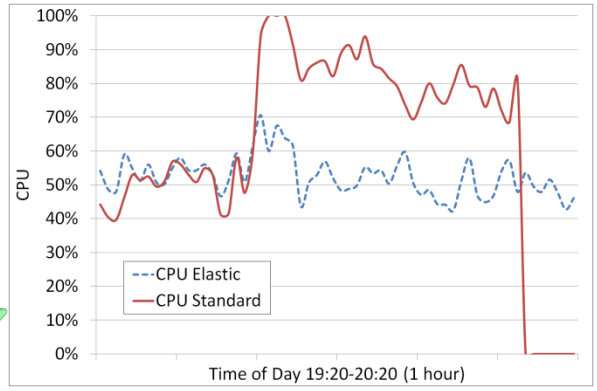
a) Requests



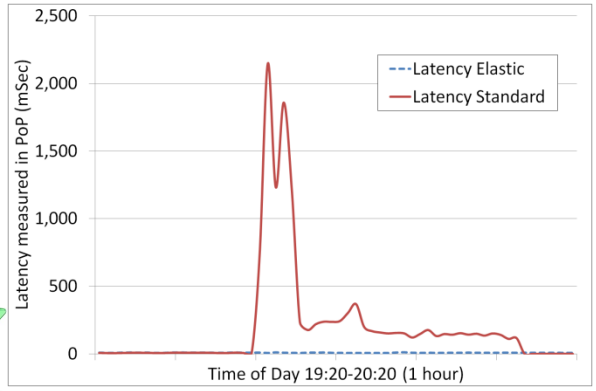
elastic

standard

b) Levels



c) CPU



d) Latency

Thank YOU!