Performance Benchmarking in Systems

L’évaluation de performance en système

Emmanuel Cecchet
University of Massachusetts Amherst

UMASSCS
DEPARTMENT OF COMPUTER SCIENCE

Lass Laboratory for Advanced Systems Software

& UMass Digital Data Forensics Research
WHY ARE WE BENCHMARKING?

- Because my advisor told me to do it?
- Because others are doing it?
- Because I can’t get my paper published without it?
Performance

Faster is better?
Bigger is better?
Scalable is better?

What about manageability?

Which is the right metric?

Hardware counters
Throughput
Latency
Watts

KEEPING THE LINES OF WORKING MORE LAZY.
EXPERIMENTAL METHODOLOGY

- Limiting performance bias
  - Performance sensitive to experimental setup.
EXPERIMENTAL ENVIRONMENT

- Software used
  - OS
  - Libraries
  - Middleware
  - JVMs
  - Application version
  - Compiler / build options
  - Logging/debug overhead
  - Monitoring software

- Hardware used
  - Cpu / mem / IO
  - Network topology
SCI NETWORK PERFORMANCE AND PROCESSOR STEPPING
OUTLINE

- How Relevant are Standard Systems Benchmarks?
- BenchLab: Realistic Web Application Benchmarking
- An Agenda for Systems Benchmarking Research
SPEC BENCHMARKS

- http://www.spec.org
- Benchmark groups
  - Open Systems Group
    - CPU (int & fp)
    - JAVA (client and server)
    - MAIL (mail server)
    - SFS (file server)
    - WEB
  - High Performance Group
    - OMP (OpenMP)
    - HPC
    - MPI
  - Graphics Performance Group
    - APC (Graphics applications)
    - OPC (OpenGL)
TYPICAL E-COMMERCE PLATFORM

- Virtualization
- Elasticity/Pay as you go in the Cloud
TYPICAL E-COMMERCE BENCHMARK

- Setup for performance benchmarking
  - Browser emulator
  - Static load distribution
  - LAN environment

![Diagram of e-commerce benchmark setup](image-url)
**Open vs Closed**

- *Open Versus Closed: A Cautionary Tale* – B. Schroeder, A. Wierman, M. Harchor-Balter – NSDI’06
  - response time difference between open and close can be large
  - scheduling more beneficial in open systems

(a) Closed system

(b) Open system

(c) Partly-open system
TYPICAL DB VIEW OF E-COMMERCE BENCHMARKS

- Direct SQL injection
TPC-W BENCHMARK

- Open source PHP and Java servlets implementations with MySQL/PostgreSQL
- Browser Emulators have significant variance in replay
WHY IS TPC-W OBSOLETE?

- HTTP 1.0, no CSS, no JS...

- And seriously… did you recognize Amazon.com?
RUBiS BENCHMARK

- Auction site (a la eBay.com)
- Many open source implementations
  - PHP
  - Java: Servlet, JEE, Hibernate, JDO...
- Everybody complains about it
- Everybody uses it

Why?
- It is available
- It is small enough to be able to mess with it
- Others are publishing papers with it!
WEB APPLICATIONS HAVE CHANGED

- Web 2.0 applications
  - Rich client interactions (AJAX, JS...)
  - Multimedia content
  - Replication, caching...
  - Large databases (few GB to multiple TB)
- Complex Web interactions
  - HTML 1.1, CSS, images, flash, HTML 5...
  - WAN latencies, caching, Content Delivery Networks...
## More Reasons Why Benchmarks Are Obsolete?

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>HTML</th>
<th>CSS</th>
<th>JS</th>
<th>Multimedia</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>RUBiS</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>eBay.com</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>31</td>
<td>38</td>
</tr>
<tr>
<td>TPC-W</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>amazon.com</td>
<td>6</td>
<td>13</td>
<td>33</td>
<td>91</td>
<td>141</td>
</tr>
<tr>
<td>CloudStone</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>21</td>
<td>28</td>
</tr>
<tr>
<td>facebook.com</td>
<td>6</td>
<td>13</td>
<td>22</td>
<td>135</td>
<td>176</td>
</tr>
<tr>
<td>wikibooks.org</td>
<td>1</td>
<td>19</td>
<td>23</td>
<td>35</td>
<td>78</td>
</tr>
<tr>
<td>wikipedia.org</td>
<td>1</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td>36</td>
</tr>
</tbody>
</table>

Number of interactions to fetch the home page of various web sites and benchmarks
STATE SIZE MATTERS

- Does the entire DB of Amazon or eBay fit in the memory of a cell phone?
  - TPC-W DB size: 684MB
  - RUBiS DB size: 1022MB

- Impact of CloudStone database size on performance

<table>
<thead>
<tr>
<th>Dataset size</th>
<th>State size (in GB)</th>
<th>Database rows</th>
<th>Avg cpu load with 25 users</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 users</td>
<td>3.2</td>
<td>173745</td>
<td>8%</td>
</tr>
<tr>
<td>100 users</td>
<td>12</td>
<td>655344</td>
<td>10%</td>
</tr>
<tr>
<td>200 users</td>
<td>22</td>
<td>1151590</td>
<td>16%</td>
</tr>
<tr>
<td>400 users</td>
<td>38</td>
<td>1703262</td>
<td>41%</td>
</tr>
<tr>
<td>500 users</td>
<td>44</td>
<td>1891242</td>
<td>45%</td>
</tr>
</tbody>
</table>

CloudStone Web application server load observed for various dataset sizes using a workload trace of 25 users replayed with Apache HttpClient 3.
OUTLINE

- How Relevant are Standard Systems Benchmarks?
- BenchLab: Realistic Web Application Benchmarking
- An Agenda for Systems Benchmarking Research
BENCHMARK DESIGN

Traditional approach (TPC-W, RUBiS…)

Workload definition + Web Emulator → Application under Test

BenchLab approach

HTTP trace → Real Web Browsers → Application under Test
BenchLab: Trace recording

- Record traces of real Web sites
- HTTP Archive (HAR format)
BenchLab WebApp

- JEE WebApp with embedded database
- Repository of benchmarks and traces
- Schedule and control experiment execution
- Results repository
- *Can be used to distribute / reproduce experiments and compare results*

**Web Frontend**

- Results upload
- Trace download
- Experiment start/stop
- Browser registration
- Experiment scheduler

**Database**

- Traces (HAR or access_log)
- Results (HAR or latency)
- Experiment Config
- Benchmark VMs

- Upload traces / VMs
- Define and run experiments
- Compare results
- Distribute benchmarks, traces, configs and results
**BenchLab Client Runtime (BCR)**

- Replay traces in real Web browsers
- Small Java runtime based on Selenium/WebDriver
- Collect detailed response times in HAR format
- Can record HTML and page snapshots
- Upload results to BenchLab WebApp when done
Wikimedia foundation Wikis

- Wikimedia Wiki open source software stack
  - Lots of extensions
  - Very complex to setup/install
- Real database dumps (up to 6TB)
  - 3 months to create a dump
  - 3 years to restore with default tools
- Multimedia content
  - Images, audio, video
  - Generators (dynamic or static) to avoid copyright issues
- Real Web traces from Wikimedia
- Packaged as Virtual Appliances
**WIKIPEDIA DEMO**

- **Wikimedia Wikis**
  - Real software
  - Real dataset
  - Real traces
  - Packaged as Virtual Appliances

- **Real Web Browsers**
  - Firefox
  - Chrome
  - Internet Explorer

---

<table>
<thead>
<tr>
<th>Name</th>
<th>Method</th>
<th>Status</th>
<th>Type</th>
<th>Size</th>
<th>Time</th>
<th>Timeline</th>
</tr>
</thead>
<tbody>
<tr>
<td>site-wide-5674991483.css_V183552863...</td>
<td>GET</td>
<td>200</td>
<td>text/css</td>
<td>30.15KB</td>
<td>141ms</td>
<td>7.92s</td>
</tr>
<tr>
<td>websiteGridCSS-websiteGridCSS-358...</td>
<td>GET</td>
<td>200</td>
<td>text/css</td>
<td>21.36KB</td>
<td>149ms</td>
<td>11.72s</td>
</tr>
<tr>
<td>navPackedSprites-US-22_V183711641...</td>
<td>GET</td>
<td>200</td>
<td>image/png</td>
<td>7.11KB</td>
<td>200ms</td>
<td>15.83s</td>
</tr>
<tr>
<td>transparent-pixel_V192234675.gif</td>
<td>GET</td>
<td>(from cache)</td>
<td>image/gif</td>
<td>(from cache)</td>
<td>88ms</td>
<td></td>
</tr>
<tr>
<td>site-wide-13640146130.js_V183531323...</td>
<td>GET</td>
<td>200</td>
<td>application/X-javascript</td>
<td>133.61KB</td>
<td>403ms</td>
<td></td>
</tr>
<tr>
<td><a href="http://www.amazon.com">www.amazon.com</a></td>
<td>GET</td>
<td>Pending</td>
<td>Pending</td>
<td>0B</td>
<td>10ms</td>
<td></td>
</tr>
<tr>
<td>51skuPkUrXl-_SL500_Plstb-sticker-ar...</td>
<td>GET</td>
<td>200</td>
<td>image/jpeg</td>
<td>7.10KB</td>
<td>170ms</td>
<td></td>
</tr>
<tr>
<td>s9-widget-combined-min_V183042506...</td>
<td>GET</td>
<td>200</td>
<td>text/css</td>
<td>8.65KB</td>
<td>127ms</td>
<td></td>
</tr>
</tbody>
</table>
HTTP VS BROWSER REPLAY

- Browsers are smart
  - Parallelism on multiple connections
  - JavaScript execution can trigger additional queries
  - Rendering introduces delays in resource access
  - Caching and pre-fetching

- HTTP replay cannot approximate real Web browser access to resources
Typing Speed Matters

- Auto-completion in search fields is common
- Each keystroke can generate a query

```
GET /api.php?action=opensearch&search=W
GET /api.php?action=opensearch&search=Web
GET /api.php?action=opensearch&search=Web+
GET /api.php?action=opensearch&search=Web+2
GET /api.php?action=opensearch&search=Web+2.0
```
**JavaScript Effects on Workload**

- Browser side input validation
- Additional queries during form processing
LAN vs WAN load injection

- Deployed BCR instances in Amazon EC2 data centers
  - As little as $0.59/hour for 25 instances for Linux
  - Windows from $0.84 to $3/hour
- Latency
  - WAN latency >= 3 x LAN latency
  - Latency standard deviation increases with distance
- CPU usage varies greatly on server for same workload (LAN 38.3% vs WAN 54.4%)

![Graph showing browser perceived latency over time with US East, US West, Europe, Asia regions]

<table>
<thead>
<tr>
<th></th>
<th>US East</th>
<th>US West</th>
<th>Europe</th>
<th>Asia</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average latency</td>
<td>920ms</td>
<td>1573ms</td>
<td>1720ms</td>
<td>3425ms</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>526</td>
<td>776</td>
<td>906</td>
<td>1670</td>
</tr>
</tbody>
</table>
OUTLINE

- How Relevant are Standard Systems Benchmarks?
- BenchLab: Realistic Web Application Benchmarking
- An Agenda for Systems Benchmarking Research
OPEN CHALLENGES - METRICS

- Manageability
  - Online operations
  - Autonomic aspects

- HA / Disaster recovery
  - Fault loads
  - RTO/RPO

- Elasticity

- Scalability
  - Private cloud
  - Internet scale

- Cacheability
  - Replication
  - CDNs
OPEN CHALLENGES - WORKLOADS

- Capture
  - Quantifiable overhead
  - Complex interactions
  - Correlation of distributed traces
- Separating trace generation from replay
- Scaling traces
- Security
  - Anonymization
  - Content of updates
- Replay
  - Complex interactions
  - Parallelism vs Determinism
  - Internet scale
OPEN CHALLENGES - EXPERIMENTS

- Experiment automation
  - Capturing experimental environment
  - Reproducing experiments
  - Minimizing setup bias

- Experimental results
  - Certifying results
  - Results repository
  - Mining/comparing results

- Realistic benchmarks
  - Applications
  - Workloads
  - Injection
CONCLUSION

- Benchmarking is hard
- Applications are becoming more complex
  - Realistic workloads/interactions
  - Realistic applications
- BenchLab for Internet scale Benchmarking of real applications
- A lot to explore...
Q&A

http://lass.cs.umass.edu/projects/benchlab/

DILBERT™ by Scott Adams

I HAVE A STUPID QUESTION...

THERE ARE NO STUPID QUESTIONS.

THAT'S RIDICULOUS... IF THERE ARE NO STUPID QUESTIONS THEN WHAT KIND OF QUESTIONS DO STUPID PEOPLE ASK? DO THEY GET SMART JUST IN TIME TO ASK QUESTIONS?

WERE YOU GOING TO ASK ME SOMETHING?

SEE... NOW THERE'S A STUPID QUESTION.