Performance Benchmarking in Systems

L'évaluation de performance en système

Emmanuel Cecchet

University of Massachusetts Amherst

UMASSCS
DEPARTMENT OF COMPUTER SCIENCE

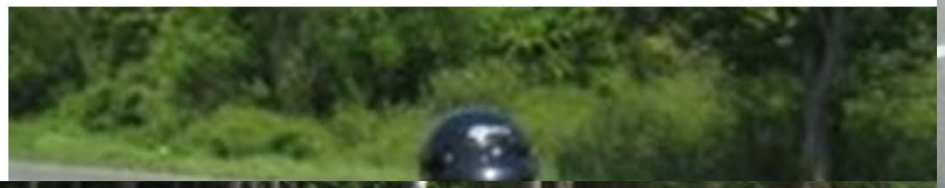
LASS Laboratory for Advanced Systems Software

& UMass Digital Data Forensics Research

WHY ARE WE BENCHMARKING?

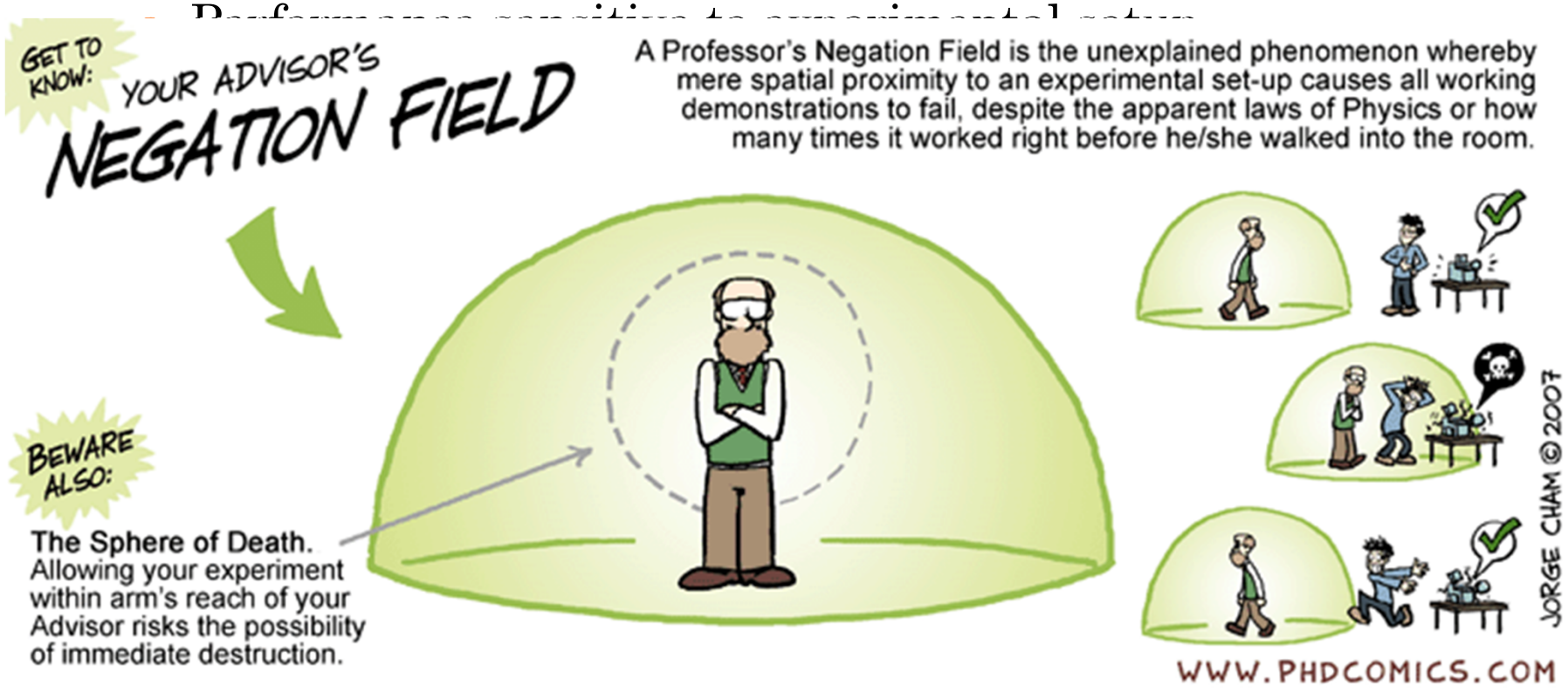
- Because my advisor told me to do it?
- Because others are doing it?
- Because I can't get my paper published without it?

Performance



EXPERIMENTAL METHODOLOGY

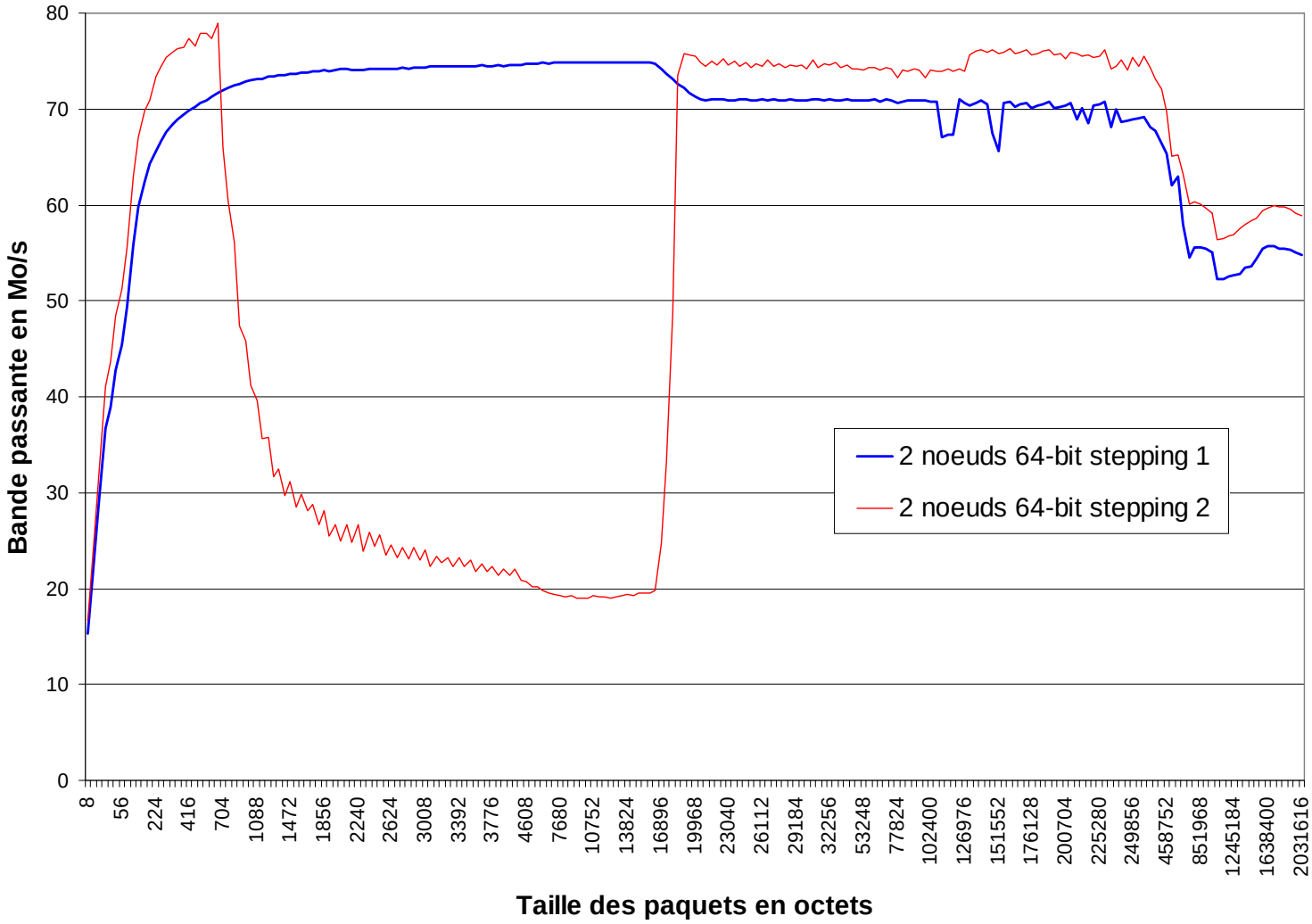
- Limiting performance bias
- ***Producing Wrong Data Without Doing Anything Obviously Wrong!*** – T. Mytkowicz, A. Diwan, M. Hauswirth, P. Sweeney – Asplos 2009
  - Performance sensitive to experimental setup



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

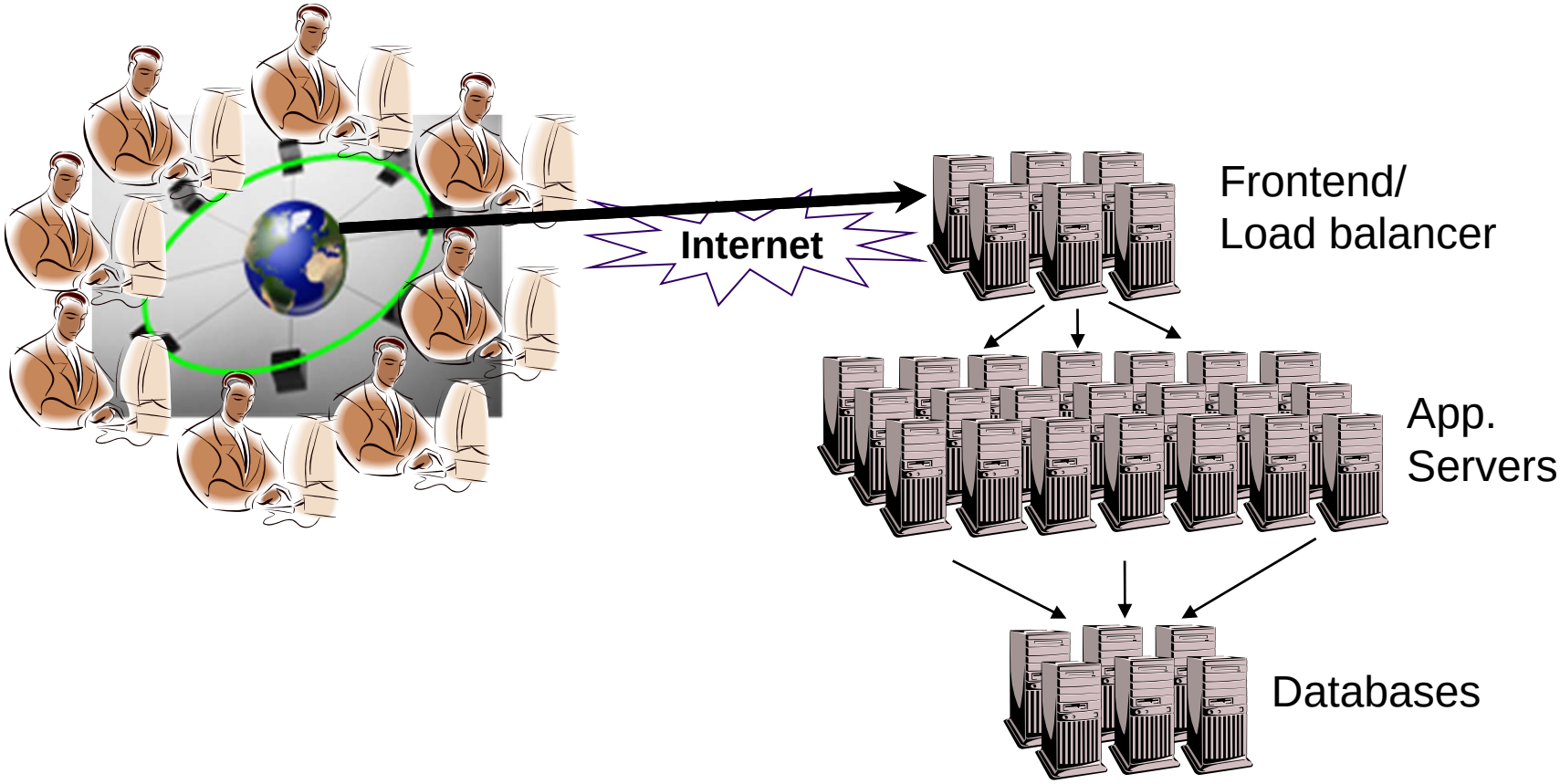
- How Relevant are Standard Systems Benchmarks?
- BenchLab: Realistic Web Application Benchmarking
- An Agenda for Systems Benchmarking Research

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)

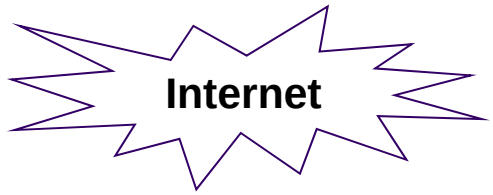
TYPICAL E-COMMERCE PLATFORM

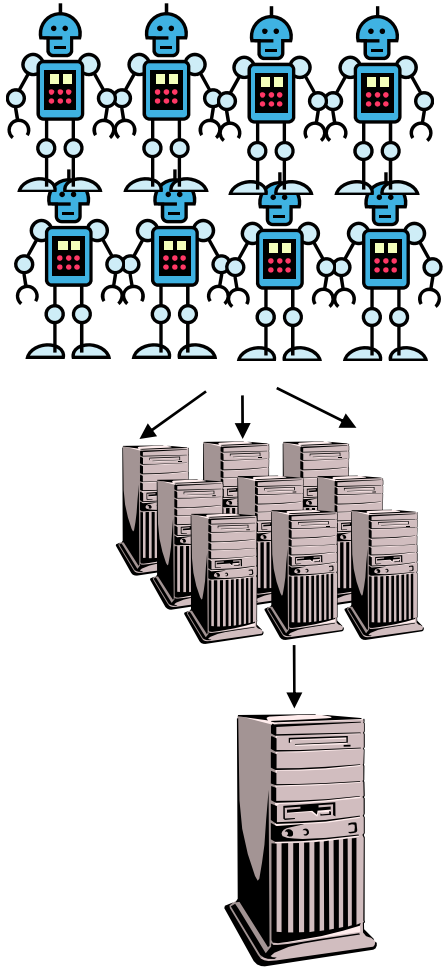
- Virtualization
- Elasticity/Pay as you go in the Cloud



TYPICAL E-COMMERCE BENCHMARK

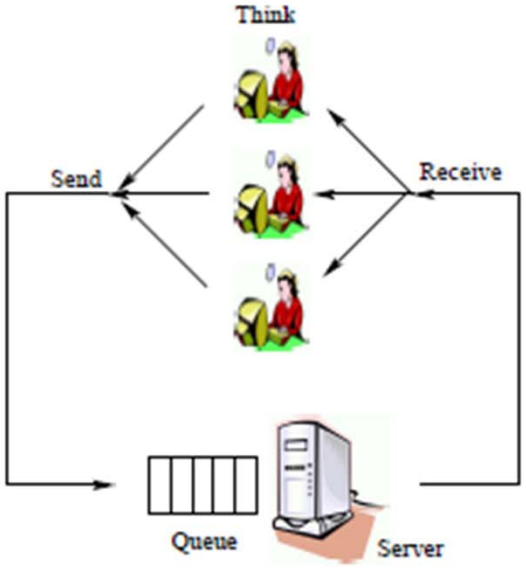
- Setup for performance benchmarking
  - Browser emulator
  - Static load distribution
  - LAN environment





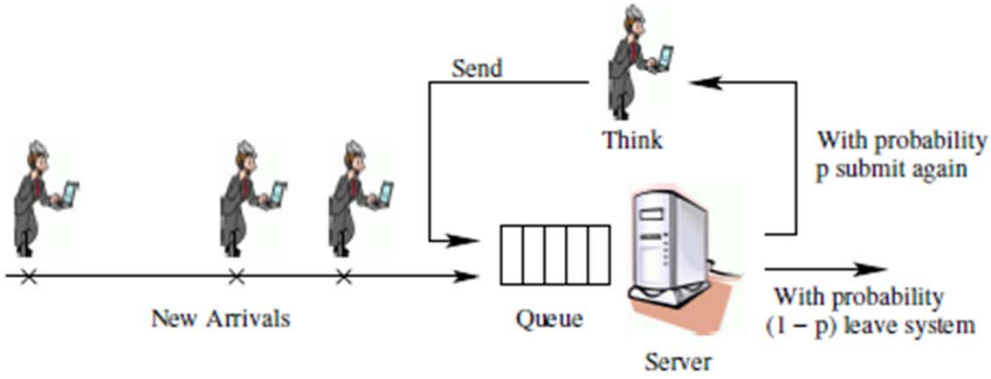
Open vs Closed

- ***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

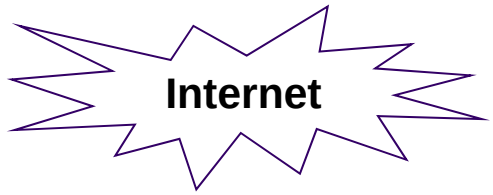
(b) Open system

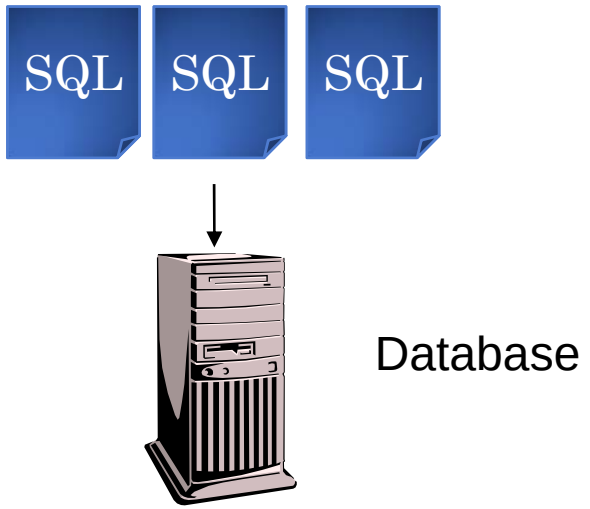


(c) Partly-open system

TYPICAL DB VIEW OF E-COMMERCE BENCHMARKS

- Direct SQL injection

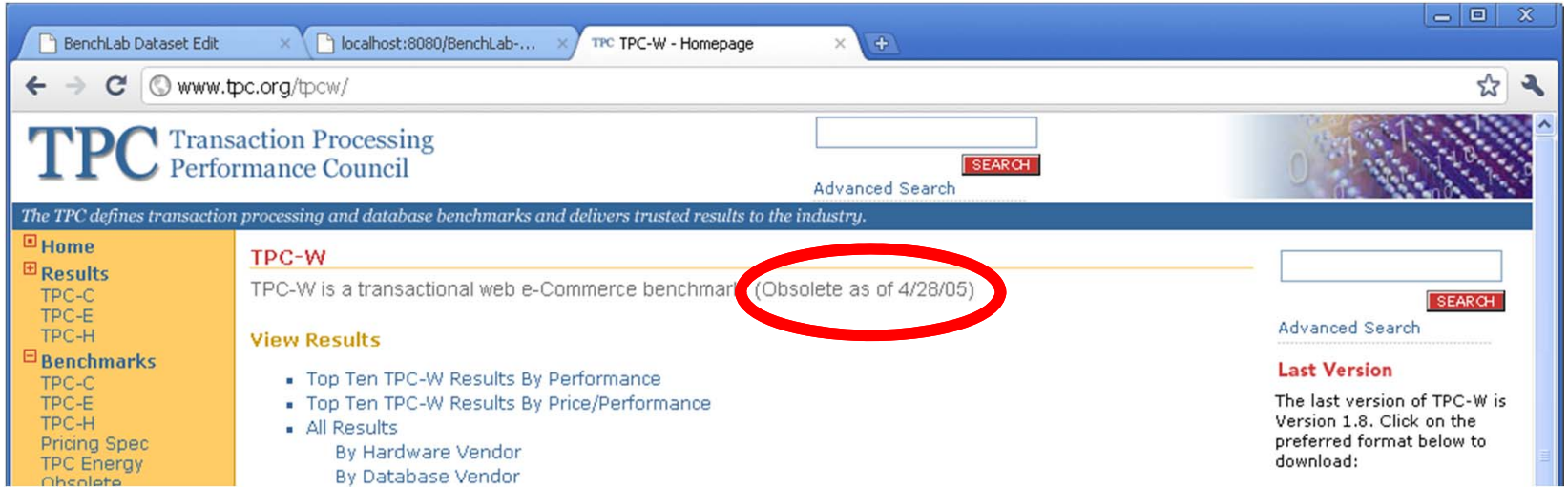




TPC-W BENCHMARK

- 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?

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

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

Dataset size	State size (in GB)	Database rows	Avg cpu load with 25 users
25 users	3.2	173745	8%
100 users	12	655344	10%
200 users	22	1151590	16%
400 users	38	1703262	41%
500 users	44	1891242	45%

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

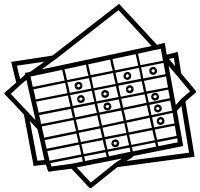
OUTLINE

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

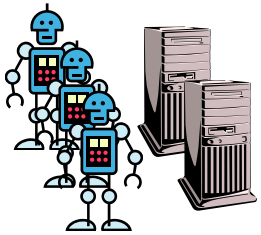
BENCHMARK DESIGN

Traditional approach (TPC-W, RUBiS…)

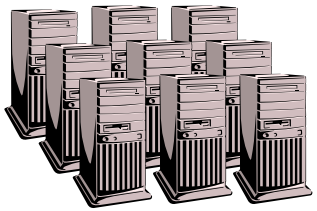
Workload definition



Web Emulator



Application under Test



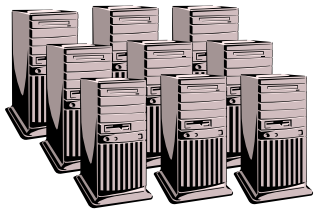
BenchLab approach

HTTP trace

```
http://...
http://...
http://...
http://...
http://...
http://...
```

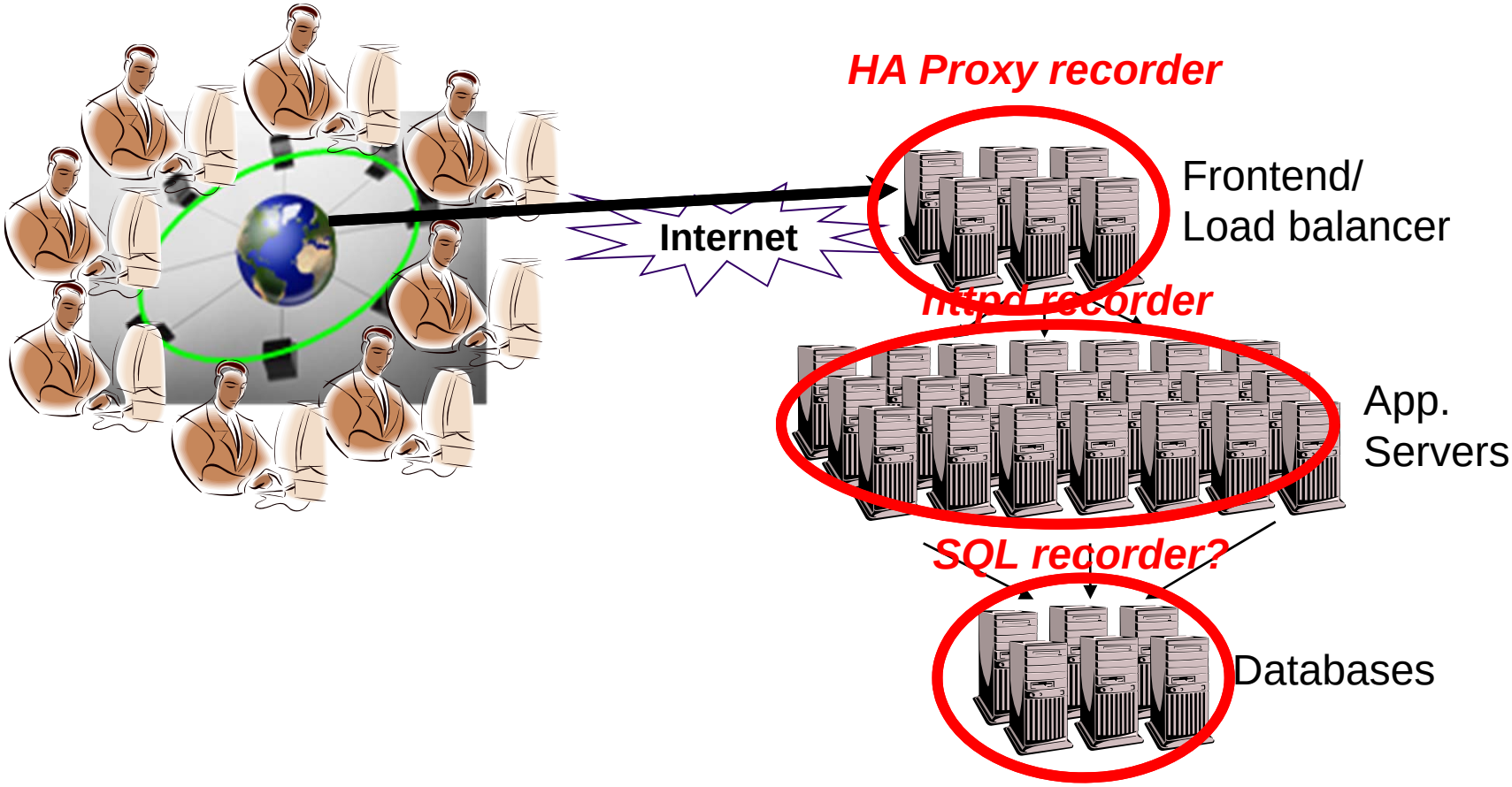
Real Web Browsers

Application under Test



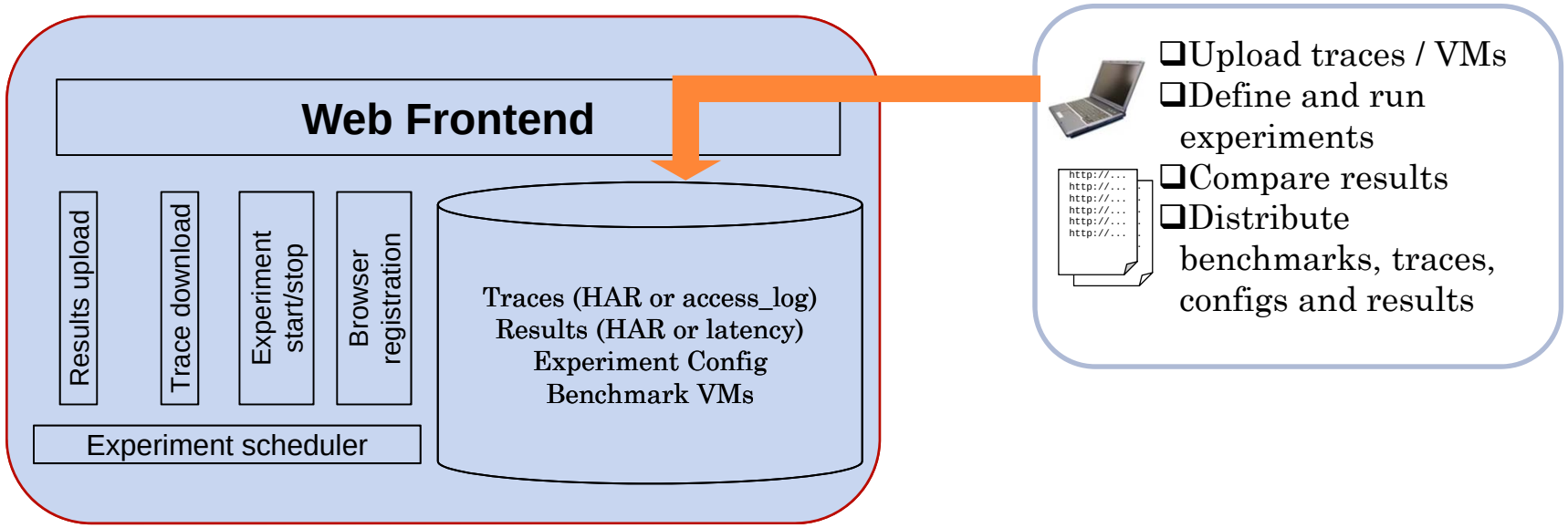
BENCHLAB: TRACE RECORDING

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



BENCHLAB WEBAPP

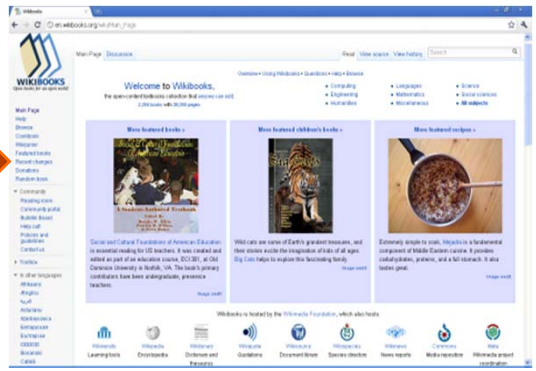
- JEE WebApp with embedded database
- Repository of benchmarks and traces
- Schedule and control experiment execution
- Results repository
- ***Can be used to distribute / reproduce experiments and compare results***



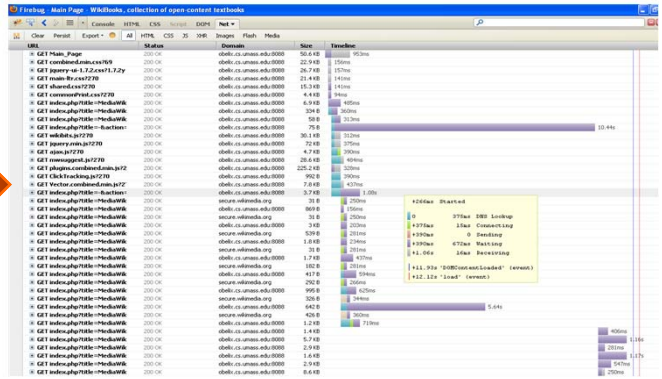
CFSE – cecchet@cs.umass.edu

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



Web page browsing and rendering



HAR results

CFSE – cecchet@cs.umass.edu

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

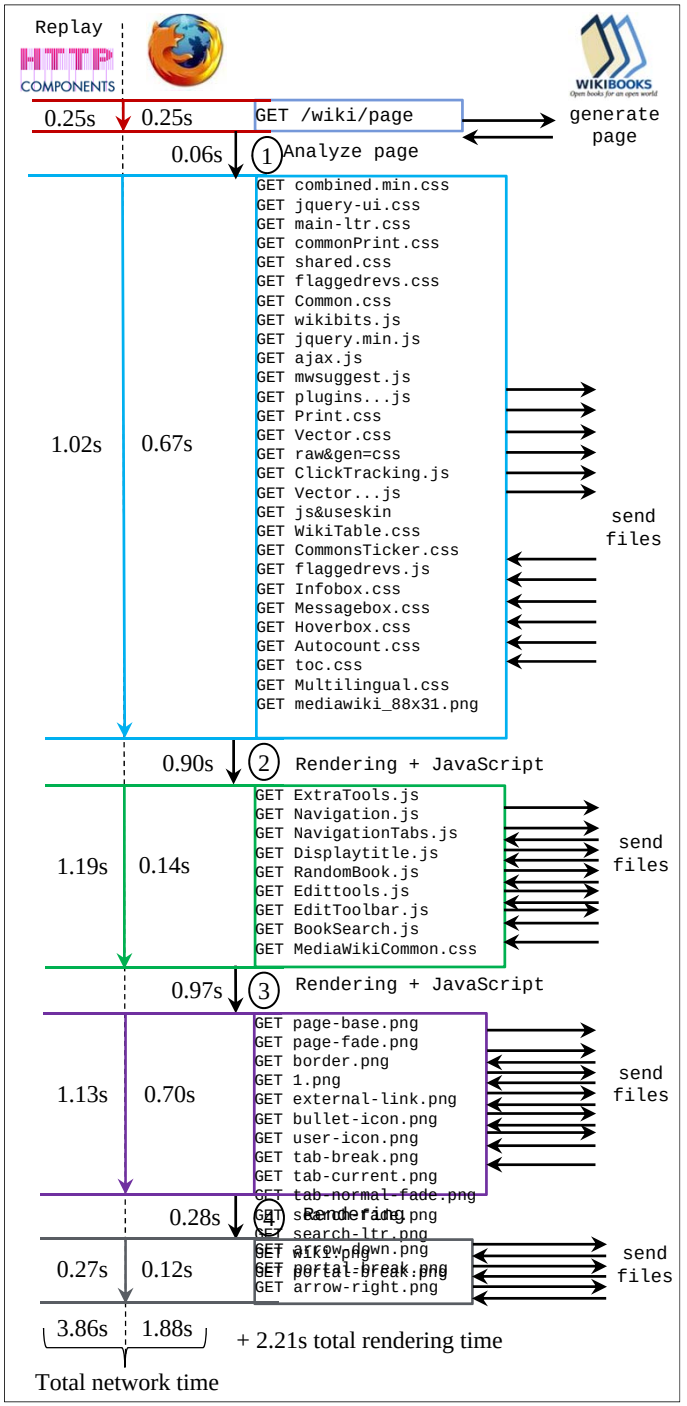
Name	Method	Status	Type	Size	Time	Timeline
site-wide-5674991483.css._V183552863...	GET	200	text/css	30.15KB	141ms	36ms
websiteGridCSS-websiteGridCSS-358...	GET	200	text/css	21.36KB	148	
navPackedSprites-US-22._V183711641_...	GET	200	image/png	7.11KB	200	
transparent-pixel._V192234675_.gif	GET	(from cache)	image/gif	(from cache)	86	
site-wide-13640146130.js._V183531323...	GET	200	application/x-javascript	133.61KB	403	
www.amazon.com	GET	Pending	Pending	0B	10	
51skuPkUrxL._SL500_PIsitb-sticker-ar...	GET	200	image/jpeg	7.10KB	170	
s9-widget-combined-min._V183012506...	GET	200	text/css	8.65KB	127	

7.82s 11.72s 15.63s

DNS Lookup 25ms
Connecting 28ms
SSL 0
Sending 0
Waiting 35ms
Receiving 36ms

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



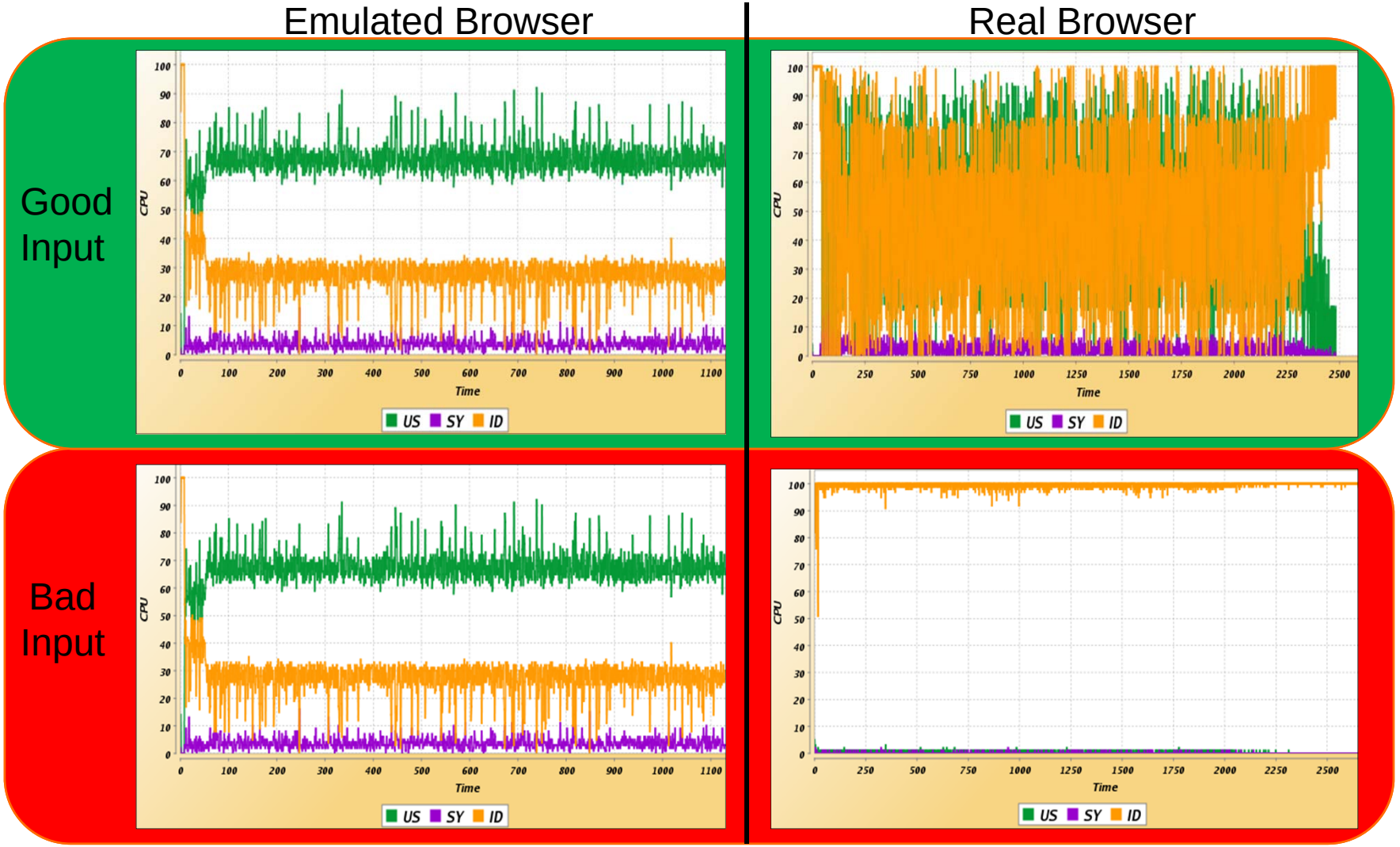
Typing speed matters

- 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.
GET /api.php?action=opensearch&search=Web+2.0
```

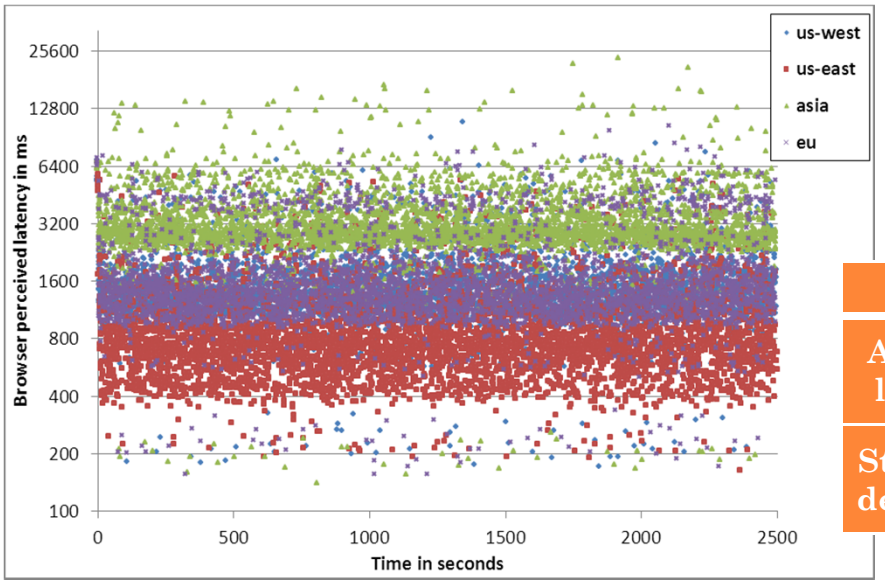
JavaScript effects on Workload

- 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%)



	US East	US West	Europe	Asia
Average latency	920ms	1573ms	1720ms	3425ms
Standard deviation	526	776	906	1670

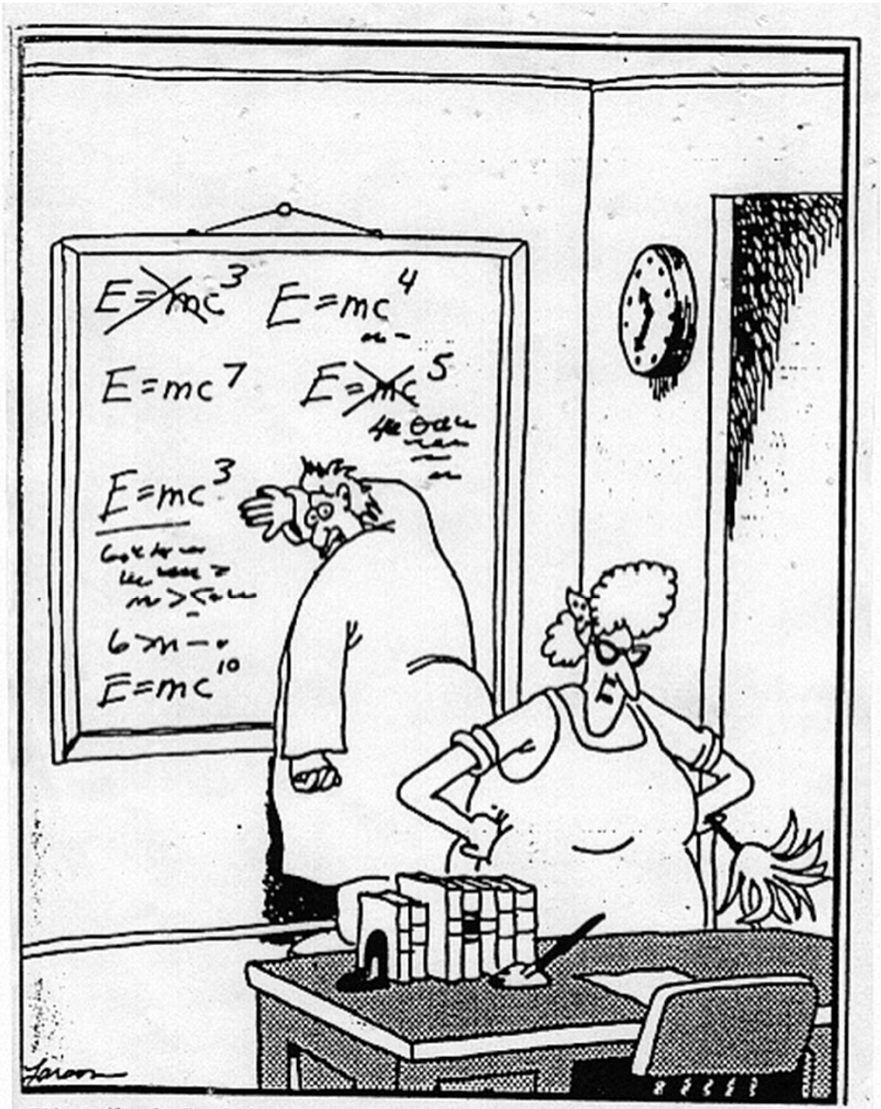
CFSE – cecchet@cs.umass.edu

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



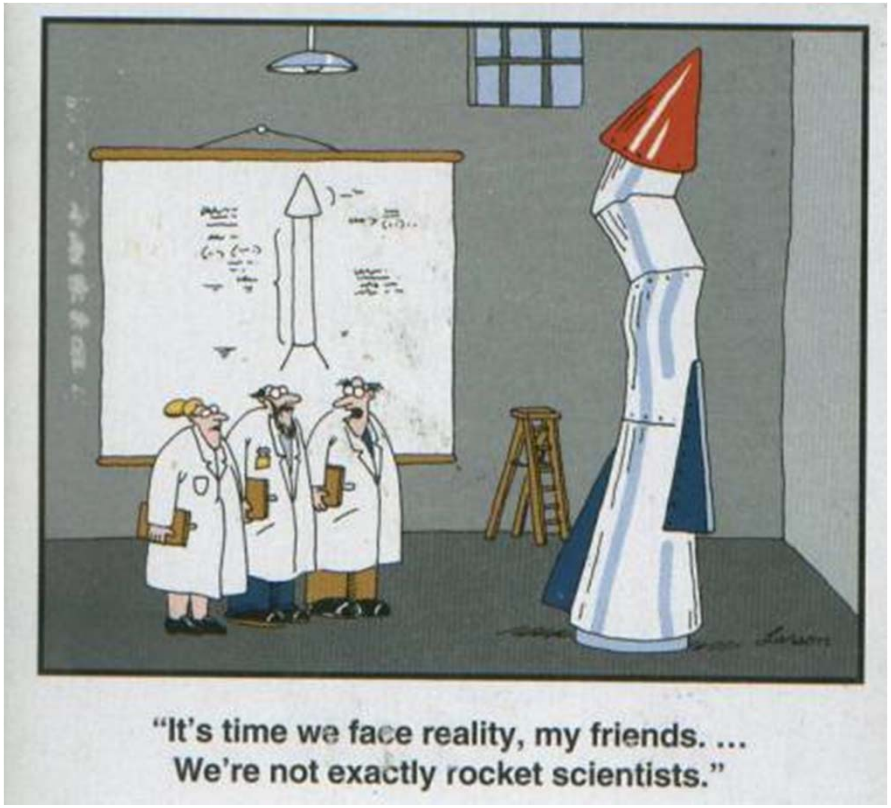
"Now that desk looks better. Everything's squared away, yessir, squaaaaaared away."

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/