

Performance testing Web 2.0

**Stuart Moncrieff, Performance Test
Consultant
JDS**



HP SOFTWARE UNIVERSE 2009



What is Web 2.0? *



Client-side complexity

- Difficulties of scripting for Web 2.0
- Client-side performance

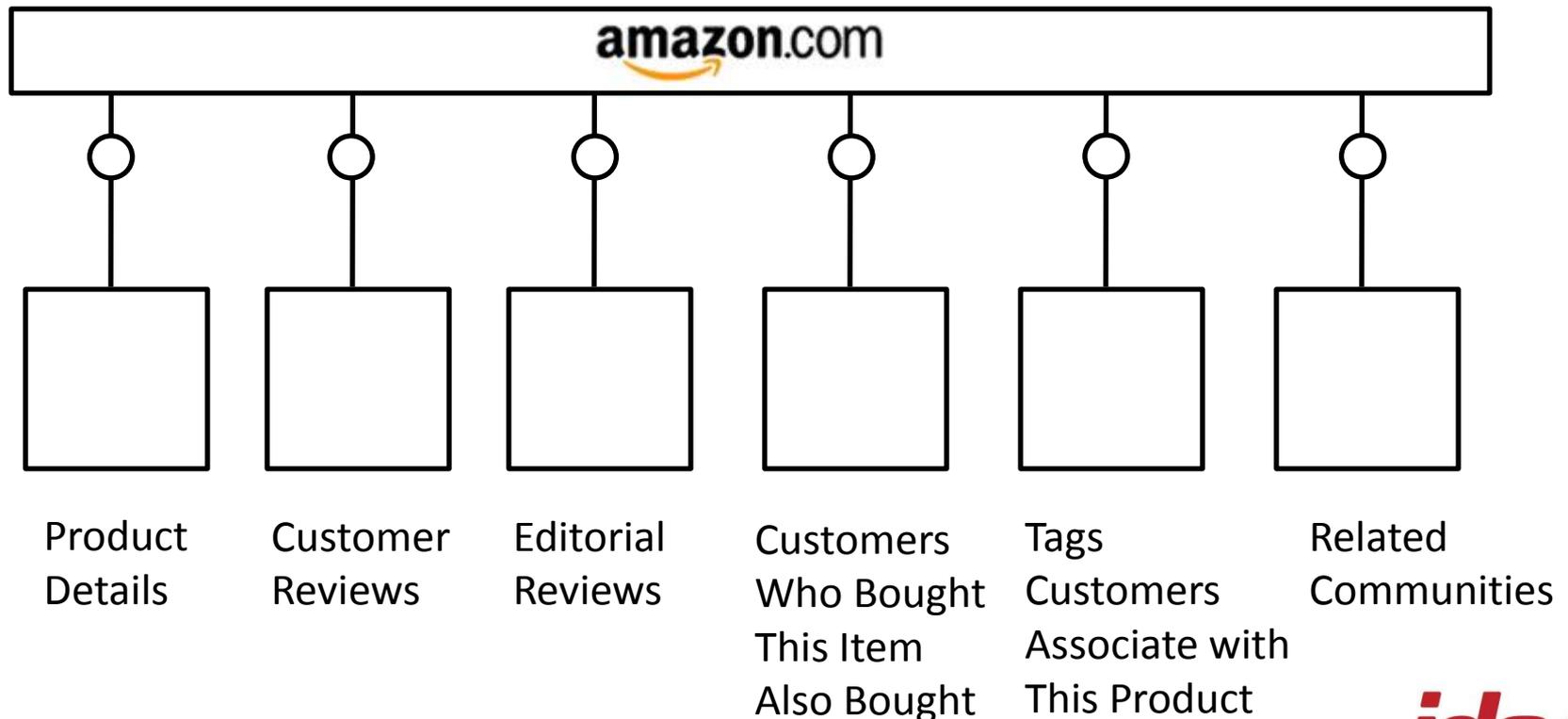
SOA & Web Services

- Testing SOA
- External consumers

* *A buzzword coined by Tim O'Reilly in 2004 :)*

SOA at Amazon.com

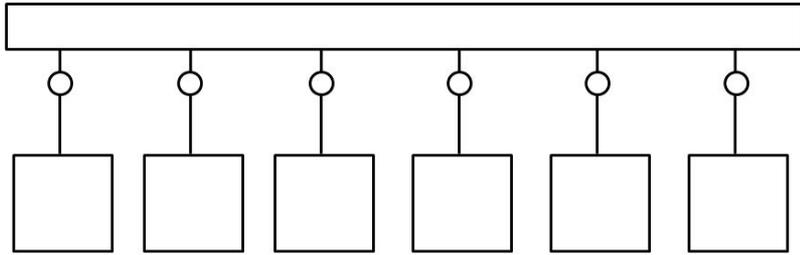
“Every Amazon web page calls on at least 150 web services”
- Werner Vogels, CTO at Amazon.com.



Testing SOA

- Bottom-up approach (test individual services before testing whole system)
 - Suits applications which use new web services
- Top-down approach
 - Suits applications which use existing web services
 - HP Diagnostics “outbound calls” view is good to see where time is being spent
- Stubbing
 - External web services (most do not offer an appropriately sized test environment)
 - Web services which have not been delivered yet
 - Expensive services e.g. Internet Banking project that stubs out service which interacts with the mainframe.

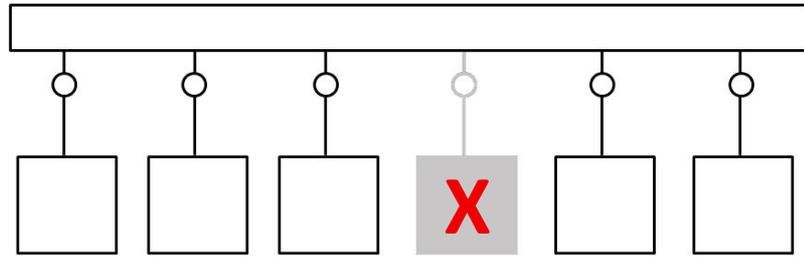
Queue Theory



“Everything looks fine. We could handle almost double the traffic if we needed to”

- Arrival rate = 50 requests/second
- Service time = 1 second
- Threads = 100
- Throughput = 50 requests/second (with ~50 idle threads)
- Max throughput = 100 requests/second
- End-user response time = 1 second + network time

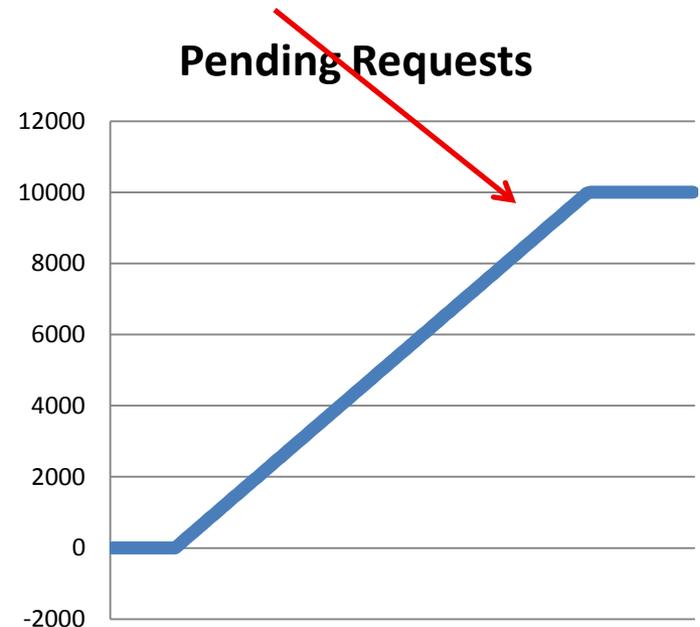
Interface timeouts



Interface has
20 second
timeout

- Arrival rate = 50 requests/second
- Service time = 20 seconds
- Threads = 100
- Throughput = 5 requests/second (0 idle threads)
- Max throughput = 5 requests/second
- End-user response time = see graph

In less than 2 minutes, there are 10,000 pending requests, and all new requests will get a “no response from server” message due to browser timeout.



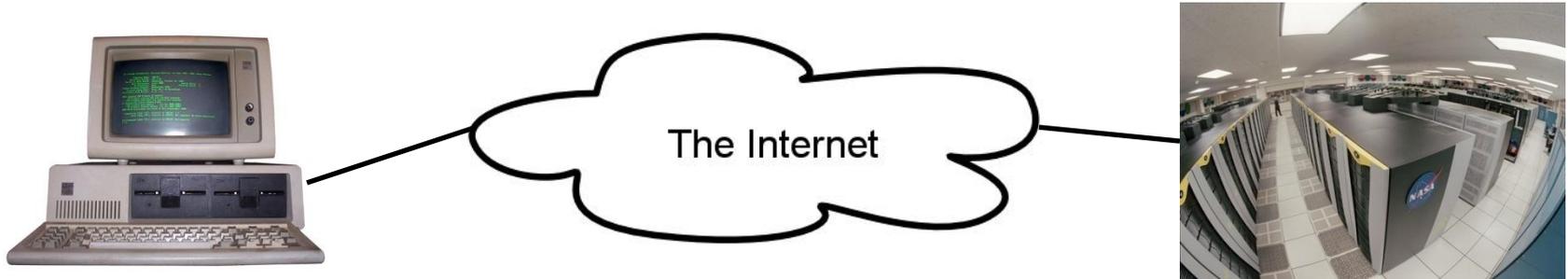
Don't get mashed by Mashups

- Web usage patterns
 - Web 1.0 – the Slashdot Effect (sudden massive increase in visitors)
 - Web 2.0 – badly behaved web service consumers (automated requests)
- Example: popular Australian e-commerce site. High website usage could cause outage/slowdown for call centre application (due to shared CRM server).
- Limit potential impact of web service consumers
 - De-couple public web services from the rest of website, and from other applications (system architecture)
 - Define upper limit through runtime policy enforcement (e.g. by using the “service protection” feature of HP SOA Policy Enforcer)

Testing SOA (summary)

- Top-down vs Bottom-up
- Stubbing
- Test web service failure cases under load
- Limit the impact of web service consumers

End-user response time components



User's PC

- No control over end-user's PC

The Network

- Good control over how much guaranteed bandwidth to Internet
- Some control over latency (by using a CDN or through geographical POP)
- No control over end-user's connection speed

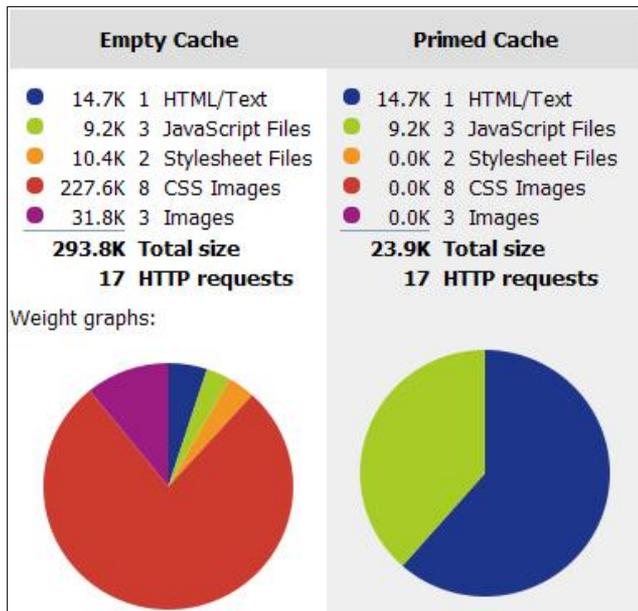
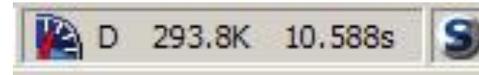
The Server

- High degree of control over code, capacity and configuration

Optimising page load time (YSlow)

Firefox add-on developed by Yahoo!

- Time taken to load the web page
- Component sizes and load times (similar to Web Page Breakdown in LoadRunner Analysis)

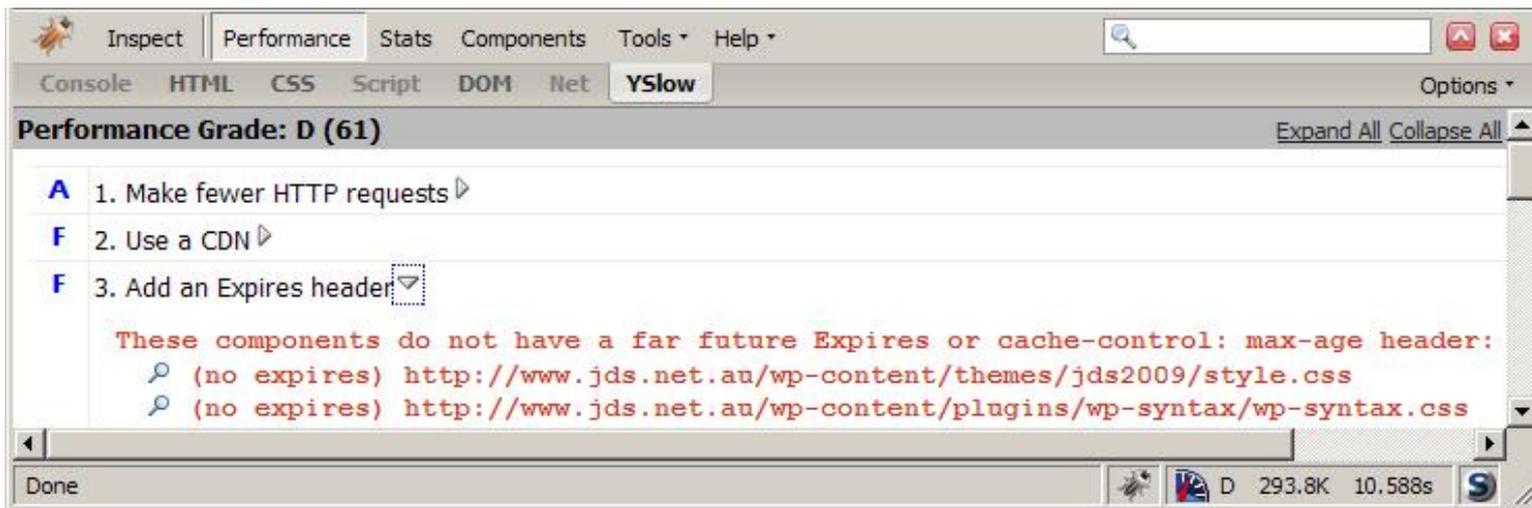


Type	URL	Expires	Gzip	RespTime	Size (Ungzip)
doc	http://www.jds.net.au/			16168	14.7K
cssimage	http://www.jds.net.au/wp-content/themes/jds2009/images/cloud.png			1991	217.4K
css	http://www.jds.net.au/wp-content/themes/jds2009/style.css			933	9.6K

- Beware of small sample sizes. Response times can vary greatly between pageloads.

YSlow continued...

- Gives a performance grade (A – F), and tips to help minimise download time, and the effect of network latency
 - E.g. Gzip enabled, minify JavaScript, Expires headers for static content
- Make sure that you understand Yahoo!'s Best Practices, so you know when they are relevant for the website you are testing.



Client-side performance

- Was not a problem previously, as client-side time is trivially small for traditional web applications.
 - Client-side processing time can now exceed server + network time by a factor of 5 for some applications.
 - But most of the time, it is still trivially small
- Client-side performance is separate from performance under load, and may be tackled separately.
 - Web 1.0 – opening a PDF. Download time small compared to Acrobat start time.
 - Web 2.0 – JavaScript, XSLT, CSS, rendering, DOM modifications
- A protocol-level load testing tool measures server + network time, not client time.

JavaScript profiling with FireBug

Firefox add-on used by web developers (front-end engineers).

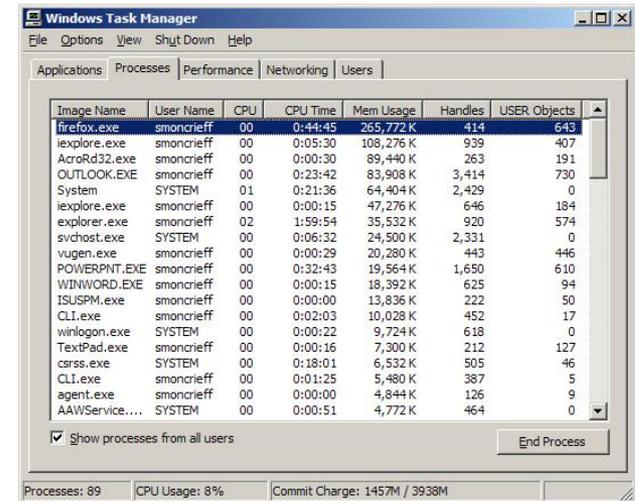
- Profiler component shows where time is being spent in client-side JavaScript code.
- Venkman is an alternative for Firefox. No good profilers for IE.

Profile (360.517ms, 6560 calls)

Function	Calls	Percent	Time	Avg	Min	Max	File
tt_HideSrcTagsRecurs	1696	19.44%	70.101ms	0.041ms	0ms	70.101ms	tooltip.js (line 458)
merge	7	13.89%	50.072ms	7.153ms	0ms	30.043ms	jquery.js (line 753)
tt_GetDivW	1	11.11%	40.058ms	40.058ms	40.058ms	40.058ms	tooltip.js (line 195)
displayAd	2	11.11%	40.057ms	20.029ms	20.028ms	20.029ms	dap.js (line 101)
_D	114	5.56%	30.044ms	0.264ms	0ms	10.015ms	hbxdigg.js (line 35)
classFilter	2	5.56%	20.029ms	10.015ms	10.014ms	10.015ms	jquery.js (line 1243)
_IL	1321	5.56%	20.028ms	0.015ms	0ms	10.014ms	hbxdigg.js (line 38)
_EV	294	2.78%	10.015ms	0.034ms	0ms	10.015ms	hbxdigg.js (line 104)

Client-side Memory Use

- It is possible to write JavaScript code that will use a lot of memory
 - Think of people on low-end machines, and people running other programs.
- It *is* possible to write JavaScript code that will leak memory
 - This is usually browser-specific, as they have implemented their own JavaScript interpreters.
 - A Microsoft development team released a memory leak detector plugin for IE 7
- Check for client-side memory use and growth over time using Windows Task Manager, or Perfmon.



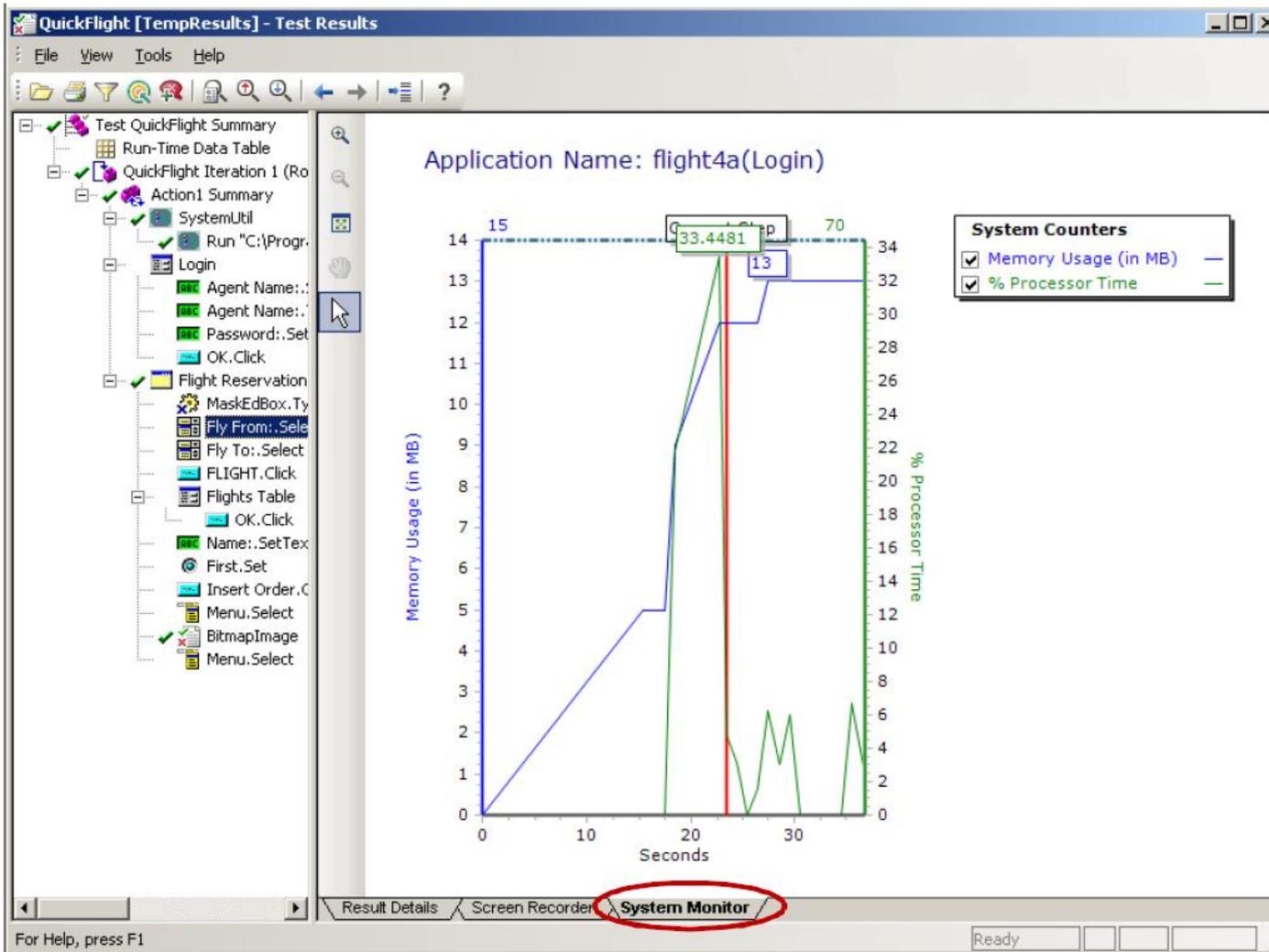
The screenshot shows the Windows Task Manager Performance tab. The 'Performance' tab is selected, and the 'Users' sub-tab is active. A table displays the memory usage for various processes. The 'firefox.exe' process is highlighted, showing a memory usage of 265,772 K. Other processes include iexplore.exe, AcroRd32.exe, OUTLOOK.EXE, System, explorer.exe, svchost.exe, vugen.exe, POWERPNT.EXE, WINWORD.EXE, ISUSPM.exe, CLI.exe, winlogon.exe, TextPad.exe, crss.exe, and AAWService....

Image Name	User Name	CPU	CPU Time	Mem Usage	Handles	USER Objects
firefox.exe	smoncrieff	00	0:44:45	265,772 K	414	643
iexplore.exe	smoncrieff	00	0:05:30	108,276 K	939	407
AcroRd32.exe	smoncrieff	00	0:00:30	89,440 K	263	191
OUTLOOK.EXE	smoncrieff	00	0:23:42	83,908 K	3,414	730
System	SYSTEM	01	0:21:36	64,404 K	2,429	0
iexplore.exe	smoncrieff	00	0:00:15	47,276 K	646	184
explorer.exe	smoncrieff	02	1:59:54	35,532 K	920	574
svchost.exe	SYSTEM	00	0:06:32	24,500 K	2,331	0
vugen.exe	smoncrieff	00	0:00:29	20,280 K	443	446
POWERPNT.EXE	smoncrieff	00	0:32:43	19,564 K	1,650	610
WINWORD.EXE	smoncrieff	00	0:00:15	18,392 K	625	94
ISUSPM.exe	smoncrieff	00	0:00:00	13,836 K	222	50
CLI.exe	smoncrieff	00	0:02:03	10,028 K	452	17
winlogon.exe	SYSTEM	00	0:00:22	9,724 K	618	0
TextPad.exe	smoncrieff	00	0:00:16	7,300 K	212	127
crss.exe	SYSTEM	00	0:18:01	6,532 K	505	46
CLI.exe	smoncrieff	00	0:01:25	5,480 K	387	5
agent.exe	smoncrieff	00	0:00:00	4,844 K	126	9
AAWService....	SYSTEM	00	0:00:51	4,772 K	464	0

Performance testing with QuickTest Pro

- Integrating with LoadRunner
 - Can define steps for StartTransaction/EndTransaction that are picked up when run with LoadRunner or Business Availability Center.
 - Can run a QTP vuser at the same time as a load test (beware problem of small sample sizes!)
- Local System Monitoring (new in QTP 10)
 - Monitor local Windows system counters while the QTP script runs
 - Export system monitor data to file (TXT, CSV) when you export a test results report
 - Useful for reproducing suspected “leak” type problems where it is necessary to perform a business process multiple times (beware problem of developers blaming automation tool for the leak, rather than their code)

QTP Local System Monitoring

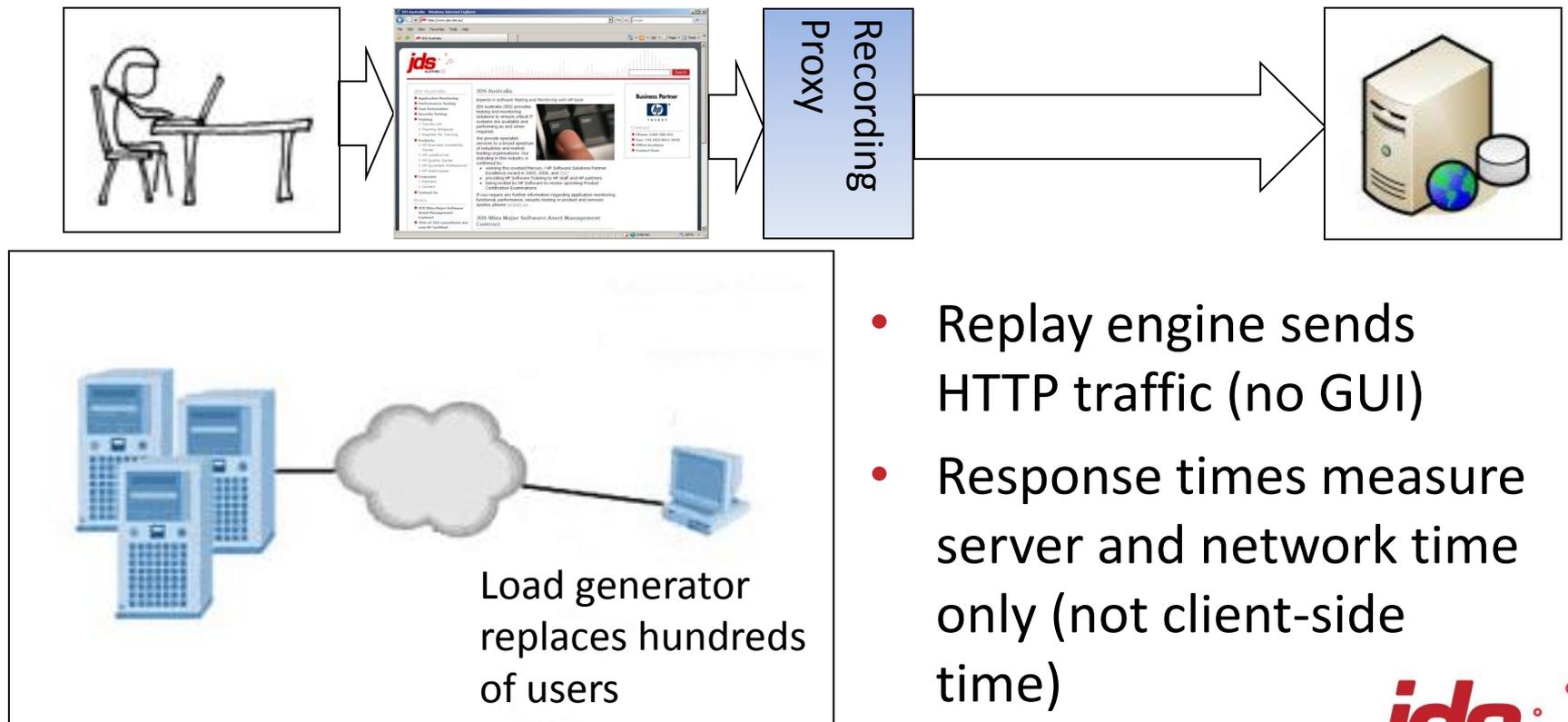


Client-side Performance (summary)

- You don't have much control over a user's PC
- Use YSlow to optimise for download time and network latency.
- Client-side time can now be a big component of overall end-user response time. Measure it!
- Check for client-side resource leaks
- Profile JavaScript code
- Don't forget that QTP 10 has a new Local System Monitoring feature

How current-generation tools work

- Record HTTP traffic as a user steps through a business process



Web load test scoping questions

- Standard questions
 - Number of business processes in scope
 - Expected number of concurrent users and peak hour transaction rate for each business process
 - System architecture (how many servers, what is on each server, what software components)
- New question
 - Does it use Ajax or rich client components like Java applets, or ActiveX objects?
(will require a short Proof-of-Concept before providing an estimate)

VuGen Scripting for Web 2.0

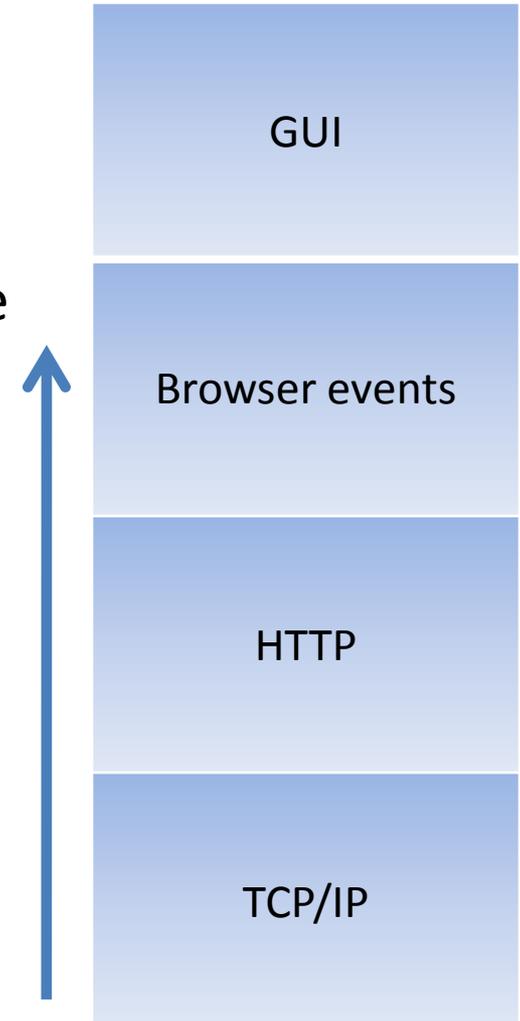
- Complex processing on the client side can make correlation really difficult
 - In *seek.com* example, it is necessary to re-implement client-side processing of JSON objects.
- This increases the script development phase of your performance testing cycle.
- This restricts the number of people who are able to create scripts (due to higher level of technical ability required).

Ajax example: seek.com.au (JSON)

```
{
  "theRequest": {
    "__type": "SEEK.Services.Client.SaveCartRequest",
    "Items": [
      {
        "__type": "SEEK.Services.ResultItem",
        "Key": {
          "__type": "SEEK.Services.ItemCode",
          "Code": "0NHFXCEY-69597554555555-75597554780404-6557-6755-6857-6N2N5"
        },
        "HasAlert": false,
        "PriceBand": 1,
        "Status": 1,
        "Description": "",
        "JobID": 0,
        "InCart": true,
        "FromDate": "2009/04/14",
        "ToDate": "2009/04/20",
        "InventoryCode": null,
        "UserOwnsJobContent": false,
        "ReferenceNumber": "",
        "Price": 220
      }
    ]
  }
}
```

Next-generation load testing tools

- Moving up the software stack.
- From HTTP level to browser event level
- A “high level” script that executes client-side code and triggers browser events e.g. onLoad, onClick
- Does not render GUI, so many virtual users can be run on a single load generator.
- But requires more resources per virtual user than an HTTP-based script.



Scripting Web 2.0 (summary)

- Include Web 2.0 questions during your initial scoping phase.
- Web services and websites with Ajax can still be scripted with regular Web (HTTP/HTML vusers), but you will save time with the Web Services and Click & Script vuser types.
- Click & Script can save you a lot of time, but does not always work. Try before you buy.

Questions...

- If you want a copy of my slides, leave a business card
- If you have a technical question, email me
 - stuart.moncrieff@jds.net.au
- Read my websites :)
 - www.jds.net.au/tech-tips/
 - www.myloadtest.com
 - www.mypentest.com



Breakthrough **Outcomes**

