



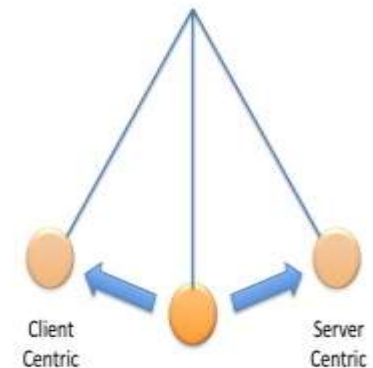
Testing on the other side of the pendulum

Issues and considerations

"Those who cannot remember the past are condemned to repeat it." as poet and philosopher George Santayana said in 1905.

The Pendulum analogy

- If you have some time in IT you should have noticed an slow pendulum swing between centralized and distributed computing paradigms
- First only few people had their own computers
- Then, there were mainframes and everyone had to timeshare
- Later, everyone had their own desktop computers... and they connected to a network
- Then, client/server computing was born and there were “thin” clients (does someone recognise the Web?)
- Then, it came Java (and the Applets) and JavaScript and “fat” clients again



Why to test Front-End performance?

- Emphasis on the look and feel of web applications puts more and more complexity on the front-end.
- Web applications that do not perform well drive people away.
- Another factor which makes the front-end performance more important is (now) Google rank websites in search results based on their web page speed.
- If usability/user experience is bad for a single user it will n-times bad for n users.

Why FrontEnd performance? (technical perspective)

- Web applications work on a client-server architecture.
- Servers are powerful machines, why do we take the risk to transfer some processing to clients?
 - We transfer processing that cannot afford network latency (therefore, needs to render results rapidly)
 - We take advantage that there is an “idle” processor on the clients machine (is that right???)
- Client side resources contribute on response time more than back-end.

Web-Apps - Testing on the other side of the pendulum

- Transferring JS code to the client's browser is actually transferring processing to the client's environment
- There are implicit requirements. Some non-functional reqs that apply to the server environment should also apply to the client's environment; e.g. maximum response (rendering) times
- Client's environment most of the time will have greater restrictions (in terms of processing power, memory, power – electrical, and others)
 - I read a week ago an article on the news stating that Facebook app was chewing out battery from mobiles
- One cannot control what one cannot measure

■ Requirements

- Some times developers comply with requirements by accident (e.g. by using libraries)
- When that happens, developers do not really understand the requirements they happen to fulfil. Further, or later, changes can make them to stop fulfilling the requirement without noticing.

Issues and considerations

- Business (customers) are not completely aware of what are the (technical) trade-offs of deploying feature-rich web applications
- Mobile vs. desktop
 - Battery charge duration
 - Memory
 - Processing power
 - Bandwidth/data-allocation usage
 - Different platforms (browsers)
- Different browsers/phones have different performance. This is much more complicated and engaged than simply server performance testing

Issues and considerations

- Implementation – inclusion of third party provided JavaScripts (especially when they are sourced from an external server)
 - Sometimes need to stub out the required script and investigate what happens if it is not available
- Application implemented using unnecessary round-trips to the server
- Not taking advantage of parallelism in browsers
 - Downloading scripts disables parallel calls
- Applications reduce load page time by lazy-loading
 - Risk – page loads fast but is consistently slow on its use
- JavaScript garbage collection uses up to 3 times the memory the application requires

Front-End Optimisation Utilities



Fiddler
Web Debugger



Page Speed Service



- Dynatrace
- Dynatrace AJAX Edition
- BrowserStack.com
 - Provides JavaScript testing and Selenium over a number of browser and devices
- WebDriver (GTAC 2013: Web Performance Testing with WebDriver)
 - Activate profiler logs in capabilities
 - Works seamlessly in Android (param --android)
 - Results in webpagetest format
 - Uses functional tests and captures performance stats from them
 - Only available in Chrome Driver
- WebPageTest
 - Open source. Can be used online. Can be installed in site.