ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Software Performance Tuning

Vijay Datla
Belk Inc, Charlotte, USA

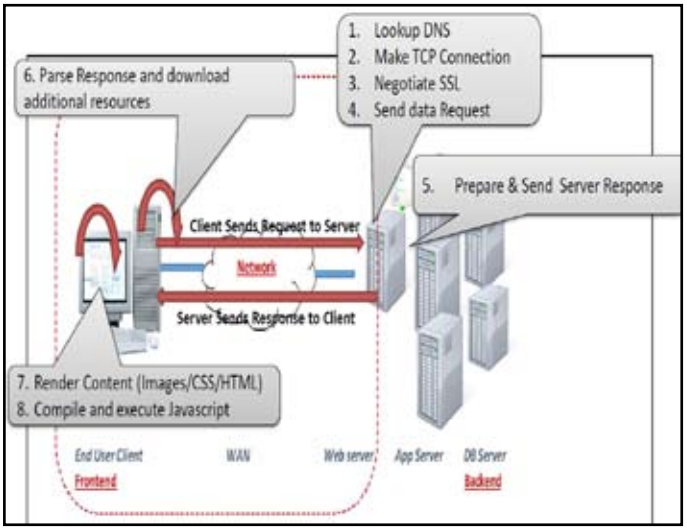
Abstract
Performance tuning across enterprise applications can Involve several processes in analysing slow performance, contention, optimizing and usage. To improve performance of large applications like ecommerce tuning experts make various empirical tuning attempts to change compiler options, program parameters or even the syntactic structure of programs. Those attempts followed by performance evaluation are repeated until satisfactory results are obtained. The task of performance tuning requires a great deal of time and effort. On account of combinatorial explosion of possible attempts, tuning experts have a tendency to make decisions on what to be explored just based on good sense of tuning. This article shows how important is performance tuning and guidelines to achieve optimal values. It explains about multi layered application performance tuning, frontend & backend optimization, application & webserver tuning and recommended performance engineering approach.

Keywords
Performance, tuning, Optimization, engineering, Vijay Datla, Vijay, Datla

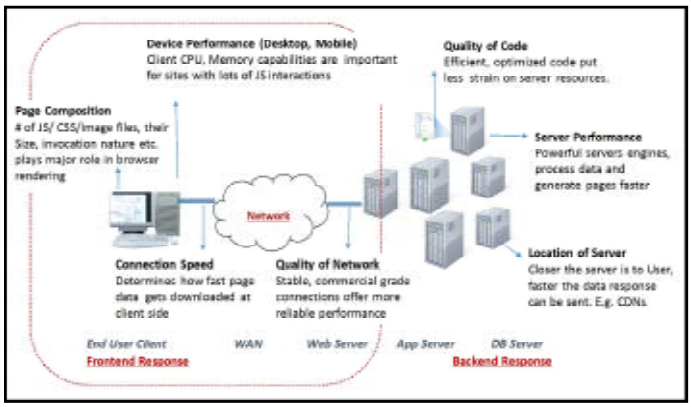
I. Introduction

- Objective
  - Challenges in Tuning Application Performance:
    - Increased Application Complexity
    - Multi-layered architectures
    - Complex deployment topologies across geographies
    - Multiple technology stacks
    - Shorter turnaround time, stricter deadlines
  - Need well-defined tuning approach that discusses set of anti-patterns, performance tuning techniques that cover almost 90% of common issues
- Scope
  - General tuning guide, not an ultimate bible!

1. Typical Multi-tier Application Flow



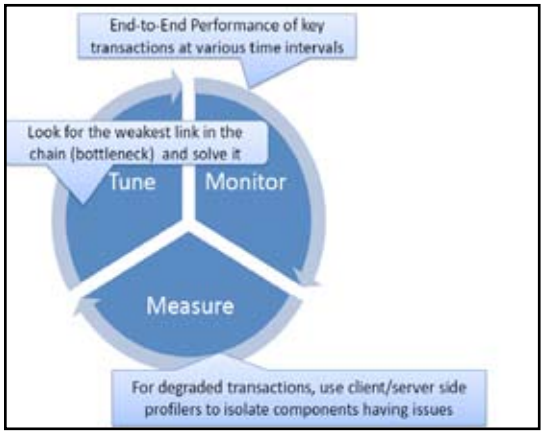
Performance Aspects in Typical multi-tier Web Application Flow



Performance Goals in Typical multi-tier Web Application Flow

- High Speed (Lower response times)
- High throughput (higher load)
- Improved scalability
- Low resource utilization
- Smooth customer experience

General Performance Tuning Methodology

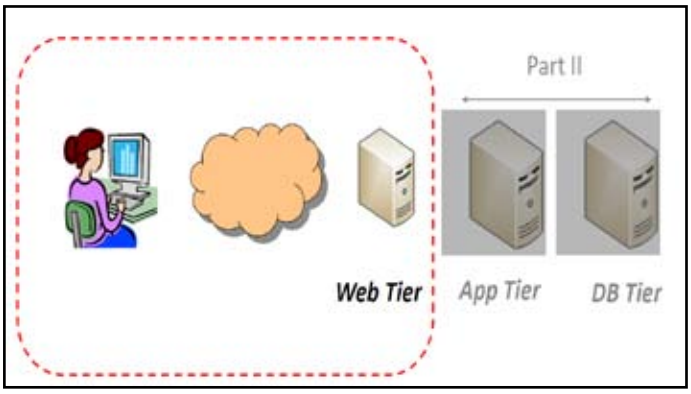


- **Do High Level Analysis**
  - Identifyareasthataretime-consumingandresource-intensiveusingcodeinstrumentation-Loganalysis, client/server side profiling tools
- ***Use 80-20 Principle***
  - Prioritizeonsolvingproblemareasthatarebusinesscriticalan

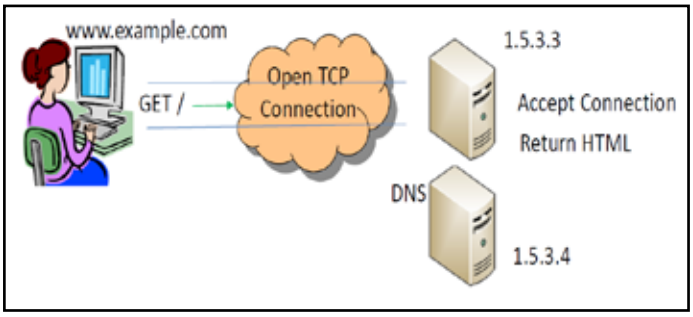
dcontributetolargersection(address 80% of the total problem space)

- ***Tune and Re-iterate***

PART I: FRONTEND & WEB TIER TUNING



Steps in Display of page



1. Connect

- Looks up the DNS server for the IP of the website
- Requests a TCP connection to the server
- Server accepts the connection and the TCP session is established

2. Initial Request and Response

- The browser requests for the page
- The Server looks for the page and returns the HTML contents

3. Parse and Additional Requests

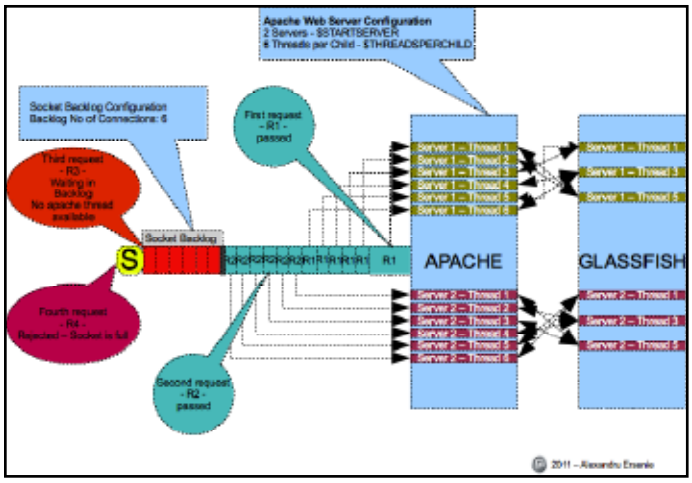
- Parse HTML contents
- Request additional Resources (Images, JavaScript, CSS)

4. Render the Page

Optimizing Initial Steps –Make Connect and Initial Requests Faster

- Minimize DNS Lookup Time
- Register in a geography from where the users access the site the most
- Accept TCP Connection quickly by having sufficient sockets
- At OS level: Set the "number of open files" optimally. Use "ulimit–n" to set this value
- Default is typically around 4K. Use ulimit–n to set the value
- On the Web Server: Set **max clients** and **socket back log**in the Web Server to the right values.
- Send the HTML as quick as possible
- Avoid/Minimize Redirects
- Set **max clients** in the Web Server optimum values
- Use a good bandwidth at the server side
- Cache the HTML in the Web Server using mod cache
- Compress the Text data using mod gzip or mod deflate
- Minimize/Optimize Contents

Add more Concurrency –Leverage Listen Back log and Max Clients



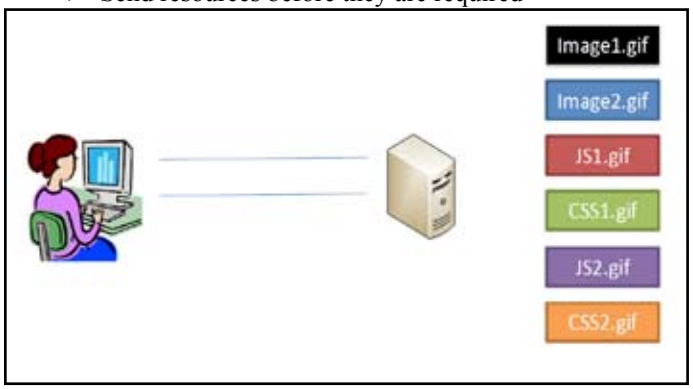
Optimization –TCP

Keep TCP Connection Alive, Set optimum values of:

- Enable KeepAlive
- Keep AliveTimeout
- Max KeepAliveRequests

Optimizing – Request of additional Resources and Render the HTML

- Use same TCP connection for multiple resources
- Reduce Size of Resources
- Reduce Number of Resources
- Don't request and/or send the resources if possible
- Request Resources in Parallel
- Send resources before they are required



Content Optimization by choosing appropriate UI Layout

- Use the right control
  - Radio/Check Box (Up to 5) (Gender)
  - List (Up to 25) (Country, City)
  - Auto Complete Look up (A few hundreds) (Clinics/ Shops in City)
  - Full Search Screen Search (Thousands) (Citizens in a country, Employees in a large organization)
- Use Asynchronous loading of bulky contents

ISSN : 2347 - 8446 (Online)
ISSN : 2347 - 9817 (Print)

Content Optimization –Pagination and Filters

- Pagination
  - Use Pagination for Lists
  - Show relevant list first. E.g. show the latest emails first
- Filters for Online Reports
  - Mandate Filters
  - Constrain period of report
- Will need to come to an optimum value after understanding the business
  - Set defaults for some filters
- Will need to decide on this after understanding the business need of the reports

Other Optimizations –Reduce Resource Size

- Reduce size of resources
  - Optimize images
  - Choose the right image type
  - Minify JavaScript
  - Minify CSS
  - Enable compression for text resources (JavaScript, CSS)
- Caution Notes
  - Minified JavaScript cannot be debugged easily
  - Do not compress Images and other binary resources like PDF which are already compressed

Optimization –Reducing Resource Size

- Tools for Compressing JavaScript
  - Yahoo Minifer, Google Closure
- Tools for Compressing CSS
  - CSS Compressor
- Choose the right image type
  - JPEG for photographic images
  - PNG for transparent images/icon
  - GIF for flat color illustrations/icons
  - JPEG for gradient illustrations/icons
- Tools for Optimizing Images
  - Smush.it, Tiny PNG

Optimization Reduce Number of Resources

- Reduce number of resources
  - Use CSS Sprites
  - Merge Small CSS and JavaScript files into large
- Optimize for above the Fold Content
  - Load the images, JavaScript, CSS required for content not visible in the screen without scroll asynchronously
  - Use Page Speed (https://developers.google.com/speed/pagespeed/insights/) to get tips on optimizing Above the Fold

Optimization –Don't Send/Request Resources at all!

- Don't request and/or send the resources if possible
  - Enable IF-MODIFIED-SINCE by passing Last-Modified Header in Response
  - Disable Etags
  - Set cache-control to public or private as the case may be.
  - Enable Expires
  - Set max-age
  - Use Application cache provided by HTML5
  - Remove all 404 (Resource Not Found) requests
  - Cache possible AJAX Requests. E.g. Auto lookup of Cities

Reduce Network Requests –Persist data at client Side

Why store data at client side?

- Increased performance due to less server requests
- Make application work offline

Available Mechanisms

- JavaScript Variables (Past, Present & Future)
- Cookies (Past, Present &Future)
- Window.name (Past &Present)
- Web SQL Database(Past)–Initial attempt for SQL relational DB & supported by Chrome, Safari, Opera15+
- HTML5WebStorage (Present &Future)
- Indexed Database(Future)
- HTML5FileAPI (Future)

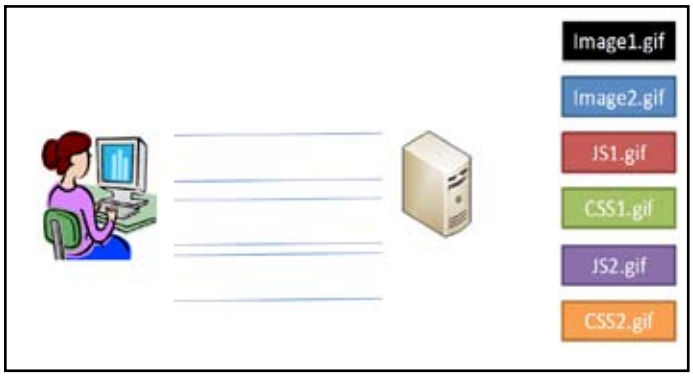
Practical Applications

- Very useful in financial services (stock prices, stock details, news etc.)
- Many large banks with core banking use a local branch DB to reduce server round trips

Optimization -Send Resources before asked for

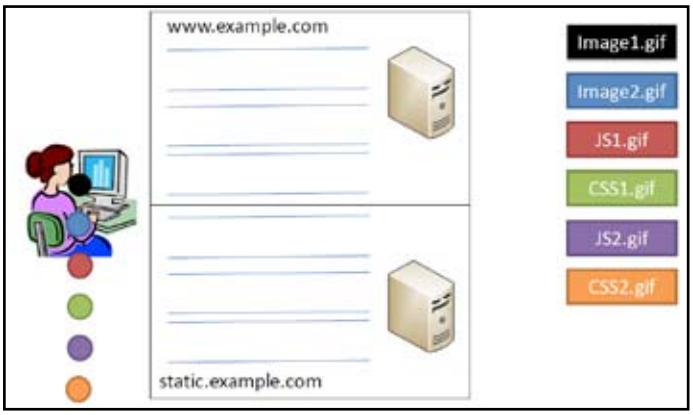
- •Use HTML5 Manifest File
- –<!DOCTYPE HTML><html manifest="demo.appcache"><body>The content of the document.....</body></html>
- –CACHE MANIFEST# 2012-02-21 v1.0.0/theme.css/logo.gif/main.jsNETWORK:login.aspFALL BACK:/html//offline.html
- •Enable HTTP2 on Server Side
- •Not all browsers support Application Cache
- •Poly fills available for some browsers
- •Not all browsers can leverage HTTP2

Make Multiple Connections – Request Resources in Parallel



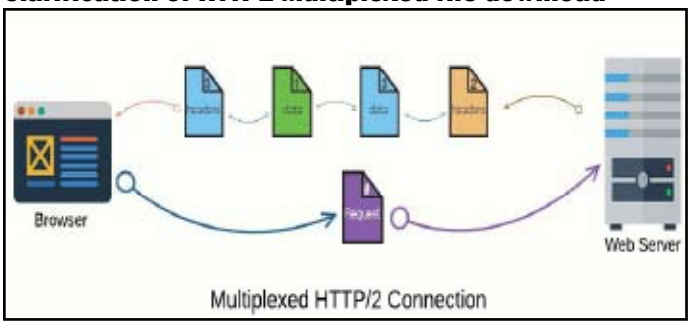
- Limited to maximum of 6 – 8 parallel connections per domain.

Make Multiple Connections to multiple Domains (CDN) – Request Resources in Parallel



- ❖ Deliver from multiple domains
- ❖ Deliver from near geography

Clarification of HTTP2 Multiplexed file download



Web Sockets

- ➢ A long running HTTP connection through which multiple requests and responses can be exchanged.
- ➢ Messages can also be pushed to the client over this connection.
  - ❖ Useful for showing online events like sports score lines, stock market prices, chats etc.
- ➢ Limitations
  - ❖ Not all proxies support Web Sockets
  - ❖ Not all browsers support Web Sockets.
  - ❖ Polyfills available for some browsers.
  - ❖ May have to leverage HTTPS as proxies do not inspect HTTPS traffic

Summary of Request Optimizations

- ➢ Optimize TCP Connection
- ➢ Send Response Quickly
  - ❖ Reduce size of Response, Pagination, Compression
- ➢ Minimize size of Resources
  - ❖ Image Optimization, Minificationof JavaScript and CSS
- ➢ Reduce size of Resources
  - ❖ Merging of smaller JavaScript and CSS, CSS Sprites
  - ❖ Above the Fold Optimization
- ➢ Don't send Resources if possible
  - ❖ Expires, Application Cache
- ➢ Parallel Requests
  - ❖ CDN
- ➢ Send Resources before asked for
  - ❖ Application Cache and HTTP2 Push

References

[1]. High Performance Websites, Steve Souders, O'Reilly2007
[2]. Even Faster Websites, Steve Souders, O'Reilly 2009
[3]. Web Performance Day book Volume2, StoyanStefanov, O'Reilly2012
[4]. http://www.w3schools.com/html/html5_intro.asp
[5]. http://www.html5rocks.com/en/