ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Faster Retrival Of Webpages Using Memory Caching Technique

Miss. Afroz K.. Khan[*1], Miss. Urmila S. Gavali[2], Mr. Rajshekhar Yadav[3] and Prof. S. P. Aware[4]
B.E-I.T (Final Year),Jawaharlal Darda Institute Engineering & Technology,
Yavatmal (MS) INDIA
afrozjkhan25@gmail.com[*1], urmilagavali@gmail.com[2], rajyadav2191@gmail.com[3], Sunita_aware@yahoo.co.in[4]

Abstract: Memory caching technique focuses on performance evaluation of serving nodes responsible for executing web applications requests. A scalability mechanism that will is; memory cached, an object caching system that could speed up throughput of web applications whereby it reduces the burden of database load. In research, it is assessed memory cached behavior on how it affects other nodes in the network and how it differs in single and clustered mode. Researchers have been able to show that performance of web application can be improved by proper integration of memory caching.
This is a fairly new technology aimed at decreasing heavy database loads by adding a scalable object-caching layer to an application As with any valuable new technology, Memory caching evolved to meet an identified problem in the market.

Keywords: Memory caching, DNS Look up , Redirect , CCS , Image compression

I. INTRODUCTION

Memory caching operates on simple data types of key-value pairs, similar to NoSQL databases, but is not persistent like NoSQL. Memory cached stores all the key-value pairs in non-persistent memory, so in the event of failure, all the stored data is lost. Throughout this project we use the term cache item to specify the combination of a key and its associated value. Keys are, by definition, unique values. In a web-serving architecture, the Memory Cahed application sits between the Front-End Web Servers, or Web Tier, and the Back-End Database.

Memory caching's purpose is to intercept data requests and satisfy them out of its logical cache (i.e. system memory) when possible, avoiding trips to disk storage attached to the back-end database. There is also the potential to avoid compute-intensive tasks by retrieving a pre-computed value from the logical cache. In both cases, the time spent retrieving or computing data results is reduced (i.e. transaction latency). A cluster of servers participates in a memory cached logical cache, with each server offering its system memory as a portion of the complete logical cache . For example, a memory cached cluster of 10 nodes, with each node having a 128 GB memory footprint, provides a 1.28 TB logical cache for servicing data requests. The logical cache is populated from data in the back-end database, the persistent data store for web-serving, or from computational servers. Cache items are maintained using Least Recently Used (LRU) policy as well as Time To Live (TTL). When an item is evicted its slot is filled by a more recent item.

II. MEMORY CACHING

A web service request can require multiple trips to memory cached, the database, and other services. If the caching strategy is efficient, the number of trips to memory cached will be an order of magnitude greater than the required trips to the database (i.e. 10:1). Data retrieval from system memory, as provided by memory cached, is an order of magnitude faster than retrieving the same data from the database (microseconds versus milliseconds) and, in most cases, orders of magnitude faster than computing the data on-the-fly. Therefore, avoiding database accesses and computation is necessary to provide acceptable user response times for web service requests.

Let's imagine a scenario. Pretend you are in charge of a large, interactive, database driven web site. It started out small, but has grown dramatically and now has literally thousands of users who hit the site every day. Some users are just reading, some users are posting messages, some users are uploading photos, and many users are doing a combination of all those activities. Each user has different levels of access, which needs to be managed as well. Since you are in charge of this website you are concerned with the user's experience.

You are worried that with a large and growing user base perhaps the speed of the site will become a problem. You're already using several servers to balance the load, but your database clusters in particular are having large trouble keeping up with the demands being placed on it. Memory cached was developed with the specific underlying assumptions that networks are fast, memory is cheap, and memory storage should be spread out across multiple machines to compensate for unreliability in a single server. A global hash table could then manage a cache that multiple web processes can simultaneously access each seeing the changes made by the other and reacting appropriately. A hash is a procedure for turning data into a small integer that serves as an index into an array. The net result is that it speeds up table lookup or data comparison tasks. Memory cached leverages a two-stage hash that acts as like a giant hash table looking up key = value pairs. Memory Cahed can be thought of as having two core components, a server and a client. In the course of a Memory Cahed lookup, the client hashes the key against a list of servers. When the server is identified, the client sends its

© 2010, IJARCS All Rights Reserved

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

request to the server who performs a hash key lookup for the actual data. Because the client performs one stage of the hashing, memory cached naturally lends itself towards the ability to easily add dozens of additional nodes. An added benefit is because there is no interconnect or multicast protocol being employed, the impact to the network is minimized.

This project tends to propose a solution for the heavy database load that many websites suffers due to high number of users using them 24 x7.Due to the increase in number of users the data retrieval and access becomes heavy for the server. This causes into poor performance and turns into very poor user Interface. Using this memory caching tecnique the load on the webserver can be reduced and thus we can optimize the bandwith thus offering a better user Interface to the users.

III. REDUCING PAGE LOAD TIME

The page load time can be reduced if the set of instructions in the best web performance standards are followed during the development process.

The page load time can also be reduced if the basic constrains behind a page loading is improved.such as ,DNS lookup,Reducing Redirect time, image optimization etc.

IV. DNS LOOK UP

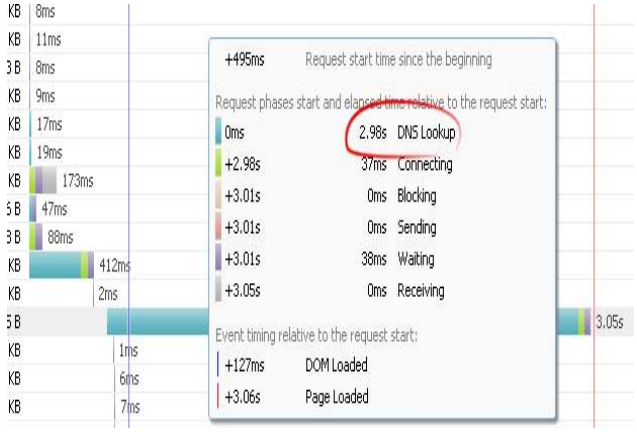

Figure: 1

DNS lookups take a meaningful amount of time to look upthe IP address for a hostname. The browser cannot do anything until the lookup is complete. **Reducing the number of unique hostnames** may **increase response times**. Just look at how a DNS lookup can take about 3 seconds of load time in SEOmoz. You can measure yours, by using Pingdom Tools. I do want to mention that when I re-tested the homepage of SEOmoz.org from a server in Dallas, it showed better results than it did before I started writing this article. Most reasonably technical Internet users have a pretty good idea what DNS is, but what actually happens when you look up a domain name is not always so clear. For those of you who are a bit uncertain of how it works (or just like geeky server charts), we found an excellent picture describing the chain of events of a DNS lookup.

The chain of events to get the IP address for www.abc.com:

First your computer queries the name server (DNS server) it is set up to use. This is the recursive name server shown above. The name server doesn't know the IP address for www.abc.com, so it will start the following chain of queries before it can report back the IP address to your computer (the numbers below correspond to the numbers in the image).

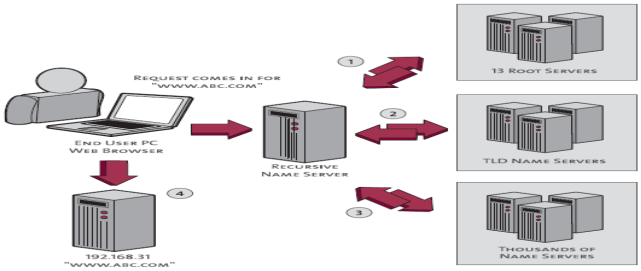

Figure: 2

a. Query the **Internet root servers** to get the name servers for the .com TLD.
b. Query the .com **TLD name servers** to get the authoritative name servers for abc.com.
c. Query the **authoritative name servers for abc.com** to finally get the IP address for the host www.abc.com, then return that IP address to your computer.
d. Done! Now that your computer has the IP address for www.abc.com, it can access that host.

V. REDUCIN REDIRECT

Sometimes to indicate the new location of a URL, track clicks, connect different parts of a site together or reserve multiple domains, you need to redirect the browser from one URL to another. Redirects trigger an extra HTTP request and add latency. Only keep redirects which are technically necessary and you can't find any other solution for it. These are Google's recommendations: Never reference URLs in your pages that are known to redirect to other URLs. Your application needs to have a way of updating URL references whenever resources change their location. Never require more than one redirect to get to a given resource. For instance, if C is the target page, and there are two different start points, A and B, both A and B should redirect directly to C; A should never redirect intermediately to B. Minimize the number of extra domains that issue redirects but don't actually serve content. Sometimes there is a temptation to redirect from multiple domains in order to reserve name space and catch incorrect user input (misspelled/mistyped URLs). However, if you train users into thinking they can reach your site from multiple URLs, you can wind up in a costly cycle of buying up new domains just to stop cybersquatters from taking over every variant of your name.

This image shows what happens when your browser tries to load SEOmoz.org:

© 2010, IJARCS All Rights Reserved

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

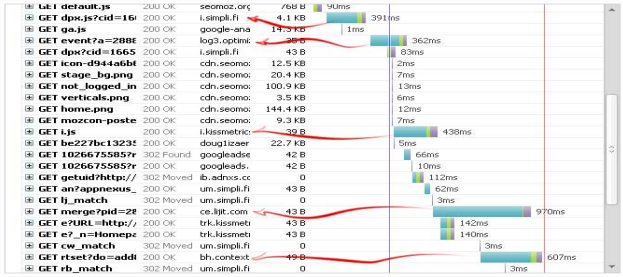

Figure: 3

As you can see, the greatest latency is the result of some external redirect chains. SEOmoz is using about 20 redirect chains that slow down the load time about 3000 milliseconds

VI. MAKE THE LANDING PAGE REDIRECT

Mobile pages redirect users to a different URL, (for example *www.seomoz.org* to *m.seomoz.org*) so making a cacheable redirect **can speed up page load time** for the next time visitors try to load site. **Use a 302 redirect** with a cache lifetime of one day. It should include a *Vary: User-Agent* as well as a *Cache-Control: private*. This way, only those visitors from mobile devices will redirect.

VII. CSS(CASCADE STYLE SHEET

It stands for Cascading Style Sheet. Style sheet refers to the document itself. Style sheets have been used for document design for years. They are the technical specifications for a layout, whether print or online. Print designers use style sheets to insure that their designs are printed exactly to specifications. A style sheet for a Web page serves the same purpose, but with the added functionality of also telling the viewing engine (the Web browser) how to render the document being viewed. Cascade is the special part. A Web style sheet is intended to cascade through a series of style sheets, like a river over a waterfall. The water in the river hits all the rocks in the waterfall, but only the ones at the bottom affect exactly where the water will flow. The same is true of the cascade in Web style sheets. Every Web page is affected by at least one style sheet, even if the Web designer doesn't apply any styles. This style sheet is the user agent style sheet - the default styles that the Web browser will use to display a page if no other instructions are provided. But if the designer provides other instructions, the browser needs to know which instructions have precedence .For example, in my Web browser, the default font is "Times New Roman" size 16. But nearly no pages I visit display in that font family and size. This is because the cascade defines the second style sheets set by the designers to redefine the font size and family and override my Web browser's defaults.

Where is CSS Used? CSS is used to style Web pages. But there is more to it than that. CSS is used to style XHTML and XML markup. This means that anywhere you have XML markup (including XHTML) you can use CSS to define how it will look. CSS is also used to define how Web pages should look when viewed in other media than a Web browser. For example, you can create a print style sheet that will define how the Web page should print out and another style sheet to display the Web page on a projector for a slide show.

Why is CSS Important? CSS is one of the most powerful tools a Web designer can learn because with it you can affect the entire mood and tone of a Web site. Well written style sheets can be updated quickly and allow sites to change what is prioritized or valued without any changes to the underlying XHTML. The challenge of CSS is that there is so much to learn. But it doesn't seem like it. After all, there are only around 60 properties in CSS Level 1 and around 70 in CSS Level 2. Compared with the number of HTML tags and attributes to learn, that can feel like a cake walk. around 70 in CSS Level 2. Compared with the number of HTML tags and attributes to learn, that can feel like a cake walk. But because CSS can cascade, and combine and browsers interpret the directives differently, CSS is more difficult than plain HTML. But once you start using it, you'll see that harnessing the power of CSS will give you more options and allow you to do more and more things with your Web sites. If you want to be a professional Web designer, you need to learn Cascading Style Sheets. But luckily, they are fun to learn.

VIII. USE CDN

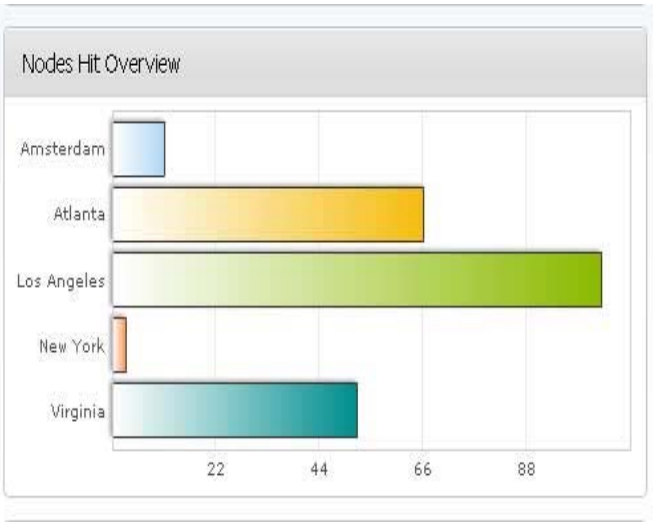

Figure: 3

A content delivery network (CDN) is a collection of web servers distributed across multiple locations to deliver content more efficiently to users. The server selected for delivering content to a specific user is typically based on a measure of network proximity. For example, the server with the fewest network hops or the server with the quickest response time is chosen. As you can see in the above image, it loads from different servers, based on the visitor's region. You can compare CDN hosting with standard web hosting here.It seems that SEOmoz uses Amazon CloudFront for this functionality and I've tried MAXCDN, It's awesome, too. You can manage your caches and lots of other useful tools in one Word Press using W3 Total Cache operating system? Another fact is that publishing your own developed applications is free which not the case for Symbian OS and Windows Mobile.

© 2010, IJARCS All Rights Reserved

This is the reason why Android gets one point and the other operating systems half a point.

IX. IMAGE COMPRESSION

Compression is useful because it helps reduce resources usage, such as data storage space or transmission capacity. Because compressed data must be decompressed to use, this extra processing imposes computational or other costs through decompression, this situation is far from being a free lunch. Data compression is subject to a space-time complexity trade-off. For instance, a compression scheme for video may require expensive hardware for the video to be decompressed fast enough to be viewed as it is being decompressed, and the option to decompress the video in full before watching it may be inconvenient or require additional storage. The design of data compression schemes involve trade-offs among various factors, including the degree of compression, the amount of distortion introduced (*e.g.*, when using lossy data compression), and the computational resources required to compress and uncompress the data. New alternatives to traditional systems, which sample at full resolution then compress, provide efficient resource usage based on principles of compressed sensing. Compressed sensing techniques circumvent the need for data compression by sampling off a cleverly selected basis. Image Compression: It is the Artθ2. Introduction & Science of reducing the It is the most useful and commercially successful technologiesθamount of data required to represent an image. The number of images compressed and decompressed daily isθin the field of Digital Image Processing. innumerable.

To understand the need for compact imageθ3. Introduction representation, consider the amount of data required to represent a 2 hour Standard Definition (SD) using 720 x 480 x A video is a sequence of video frames where each frame is aθ24 bit pixel arrays. Because video player must display the frames sequentially atθfull color still image. rates near 30fps, SD video data must be accessed at30fps x (720x480)ppf x 3bpp = 31,104,000 bpsfps – frames per second,ppf – pixels per frame,bpp – bytes per pixel & bps – bytes per second , 2 hour movie consists of31,104,000 bps x (602) sph x 2 hrs ≈ 2.24 x 1011 bytes. OR To put a 2hr movie on a single DVD, each frame must beθ Twenty seven 8.5GB dual layer DVDs are needed to store it.θ224GB of datasph = second per hour The compression must be even higher for HD, where imageθcompressed by a factor of around 26.3. resolution reach 1920 x 1080 x 24 bits/image.

Web page imagesθ5. Introduction & High-resolution digital camera photos also are also compressed to save storage space & reduce Residential Internet connection delivers data at speedsθtransmission time. ranging from 56kbps (conventional phone line) to more than Time required to transmit a small 128 x 128 x 24 bit full colorθ12mbps (broadband). Compression can reduce the transmission time by a factor ofθimage over this range of speed is from 7.0 to 0.03 sec. Similarly, number of uncompressed full color images that an 8θaround 2 to 10 or more. Megapixel digital camera can store on a 1GB Memory card can be increased, used by the operating system. For a significant classification we need to find the operating system with the lowest "Memory Footprint" which in turn maximizes the performance of the operating system. Symbian OS require 200 Kb. The Windows Mobile platform requires 3 00Kb for a typical installation.

The Android OS which is using Linux kernel will need about 250 kb of memory. All the data above apply to an installation with the basic and minimal functionalities. As a result Symbian OS needs less memory than Android which needs less memory than Windows Mobile. So Symbian gets one point, Android gets half a point and Windows Mobile zero points. Along with these applications , image compression plays anθ6. Introduction important role in many other areas including: Data Compression: It refers to the process of reducing theθ7. Fundamentals amount of data required to represent a given quantity of information. Data Vs Data and Information are not the same thing; data are theθInformation Because various amount of data can be used to represent theθmeans by which information is conveyed. same amount of information, representations that contain irrelevant or repeated information are said to contain redundant data. Let bθ9. Fundamentals & b' denote the number of bits in two representations of the same information, the relative data redundancy R of the representation with b bits is• R = 1 – (1/C); where, C commonly called the compression ratio, is If C = 10 (or 10:1), for larger representation has 10 bits of dataθdefined as• C = b / b' for every 1 bit of data in smaller representation.So, R = 0.9, indicating that 90% of its data is redundant. 2D intensity arraysθ10.

X. ENABLE ZIP COMPRESSION

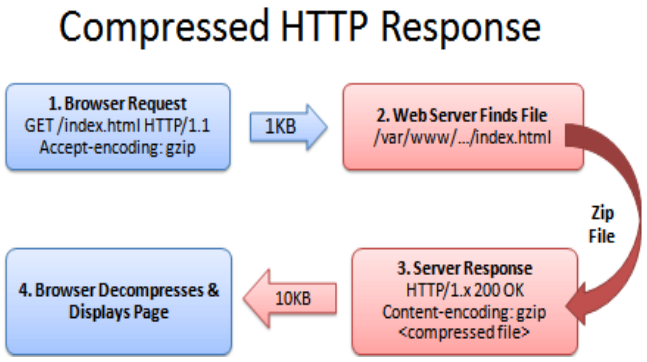

Figure: 4

Gzip is the most popular and effective compression method currently available and generally **reduces the response size by about 70%**. Approximately 90% of today's Internet traffic travels through browsers that claim to support gzip," says Yahoo. Gzipping reduces the size of the HTTP response and helps to reduce response time. It's an easy way to reduce page weight.

Serve resources from a consistent URL: For resources that are shared across multiple pages, **make sure that each reference to the same resource uses an identical URL**. If a resource is shared by multiple pages/sites that link to each other, but are hosted on different domains or hostnames, it's

© 2010, IJARCS All Rights Reserved

better to serve the file from a single hostname than to re-serve it from the hostname of each parent document. In this case, the caching benefits may outweigh the DNS lookup overhead. For example, if both *mysite.example.com* and *yoursite.example.com* use the same JS file, and *mysite.example.com* links to *yoursite.example.com* (which will require a DNS lookup anyway), it makes sense to just serve the JS file from *mysite.example.com*. In this way, the file is likely to **already be in the browser cache** when the user goes to *yoursite.example.com*."

XI. LEVERAGE BROWSER CHACHING

Expires headers tell the browser whether a resource on a website needs to be requested from the source or if it can be fetched from the browser's cache. When you set an expires header for a resource, such as all jpeg images, the browser will store those resources in its cache. **The next time the visitor comes back to the page it will load faster**, as the browser will already have those images available," says *CJ Patrick* in a nice article about **how to use expire headers to set caching**

XII. CONCLUSION

Page speed load time can be reduced with open source external libraries such as data compression and image compression and optimization libraries. With the reduction in dns lookup time the page load time can be decreased. The overall page load time can be decrease using best web performance rules. Using these techiniques an plugin to the web browser will be develop which will rewrite the webpages and optimize them in order to get them load faster. Thus user will get a better browsing experience.

XIII. ACKNOWLEDGEMENT

I would like to avail this opportunity to express my deep sense of graduate and whole hearted thanks to my guide **Prof. S. P. Aware** for giving his valuable guidance, inspiration and affectionate encouragement to study this Paper. I also acknowledge my overwhelming gratitude and immense respect to H.O.D. of I.T. Dept. **Dr. R. M. Tugnayat,** Principal **Dr. A. W. Kolhatkar** and professors who inspired us and encouraged us whole heartedly.

XIV. REFERENCES

[1]. G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W. Vogels, "Dynamo:Amazon's Highly Available Key-Value Store," in Proceedings of 21st ACM SIGOPS Symposium on Operating Systems Principles. New York, NY, USA: ACM, 2007, pp. 205–220.

[2]. B. Fitzpatrick, "Distributed caching with Memory cached," Linux Journal,vol. 2004, no. 124, p. 5, August 2004.

[3]. J. Petrovic, "Using Memory cached for Data Distribution in Industrial Environment," in Proceedings of the 3rd International Conference on Systems. Washington, DC, USA: IEEE Computer Society, 2008, pp. 368–372.

[4]. D. G. Andersen, J. Franklin, M. Kaminsky, A. Phanishayee, L. Tan, and V. Vasudevan, "FAWN: A Fast Array of Wimpy Nodes," in Proceedings of the ACM SIGOPS 22nd Symposium on Operating Systems Principles. New York, NY, USA: ACM, 2009, pp. 1–14.

[5]. K. Lim, P. Ranganathan, C. Jichuan, P. Chandrakant, M. Trevor, and R. Steven, "Understanding and Designing New Server Architectures for Emerging Warehouse-Computing Environments," Proceedings of 35th International Symposium on Computer Architecture, June 2008.

[6]. W. Lang, J. M. Patel, and S. Shankar, "Wimpy Node Clusters: What about non-wimpy workloads?" in Proceedings of the 6th International Workshop on Data Management on New Hardware. New York, NY, USA: ACM, 2010, pp. 47–55.

[7]. A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," SIGCOMM Computuer Communication Review, vol. 39, no. 1, pp. 68–73, December 2008.

[8]. B. Taylor, M., J. Kim, J. Miller, D. Wentzlaff, F. Ghodrat, B. Greenwald, H. Hoffman, J.-W. Lee, P. Johnson, W. Lee, A. Ma, A. Saraf, M. Seneski, N. Shnidman, V. Strumpen, M. Frank, S. Amarasinghe, and A. Agarwal, "

© 2010, IJARCS All Rights Reserved