OPTIMIZATION OF BLAST COMPUTE SITE THROUGH DESIGNING LITESPEED CACHE USING PPDIOO METHOD

Imam Arief Rahman [1], Iskandar Ikbal [2]

[1,2] Informatics Engineering Study Program, Indonesia Computer University

Jl. Dipatiukur No. 222-226, Coblong, Bandung. West Java 40232

E-mail: imam@blast.co.id[1] , Iskandar.ikbal@email.unikom.ac.id[2]

ABSTRACT

PT Blastech Digital is a company that was established in 2013 and is engaged in IT Infrastructure and Software Development. The company became PT Blastech Digital of CV Blastech Digital, in 2018 and began working onbusinesses Cloud Hostingthat are managed.In marketing services, the company uses the Blast Compute website www.blast.co.id. Based on the audit results using Lighthouse Engine from Google Pagespeed Insight the site has performance metrics very poorwith a total score of 0 and also still uses Web server Apachebased on aarchitecture process-based whoseusage resourceis not efficient inhandling request. That is because the Blast Compute site has not optimized through technology cache effective so that the site is perceived by visitors as still slow so that it adversely affects the user experience, sales, retention, engagement of visitors. Based on the existing problems at PT. Blastech Digital, it requires the design of cache infrastructure and web servers based event driven such as Litespeed Web server which are proven to have better performance than Apache and have advantages in the technology cache used.optimization Web server on the web www.blast.co.idusing the PPDIOO method starts with conducting a system analysis in the phase prepare, and the plan is distributed by PPDIOO. Then design the infrastructure cache through the phase of design PPDIOO. After analyzing and designing the web server cache , the implementation and testing stages are then carried out in the phase implementation of the PPDIOO channel by installing and optimizing and testing NRFU 8 times in mobilemode and desktop.

***Keywords** : Web server, Litespeed web server, Litespeed cache, PPDIOO, Optimization, Google Pagespeed Insight, Lighthouse engine.*

1. INTRODUCTION

PT. Blastech Digital is a company engaged in cloud services, based in the city of Bandung. This company is aservice cloud hosting provider named Blast Compute. The Blast Compute service can be deployed in more than 8 datacenter globally. Blast Compute has a website with the domain www.blast.co.id to market its products in Indonesia.

From the results of an interview with one of the IT Division of PT Blastech Digital, the Blast Compute site uses a cloud server. Blast Compute sites run their sites using Apache web server. In terms of performance the architecture model Event-Based outperforms Process-Based and Process-Based itself has the potential to create performance bottlenecks for traffic requests very large[3]. Because Apache is designed over thearchitecture Process-Based Model, at high volumes traffic request apache will run many worker processes to serve web resources to the browser and will consume a lot of RAM to handle high requests . In contrast to Litespeed which uses architecture event-based because it does not use many threads / processes so that it can serve thousands of requests without burdening server resources[3].

Tests based on Google Pagespeed Insight on the Blast Compute site show performance metrics very low. Intesting desktop mode web Blast Compute site gets a total score of 5 out of 100, and for mobile mode it gets a total score of 0. Scores in the range 0-49 (slow) by Lighthouse Engine from Google Pagespeed Insight indicate the slow loading speed of pages fromperception user[9]. The audit results show that optimization has not been carried out effectively using technology cache on the Blast Compute site.

Scores performance metrics lower that have impact a great in the business which is to reduce userexperience,level of engagement,retention, and conversions / sales and SEO rankings of a siteweb[7]. Reducing site loading time by just 100ms can increase conversions by 1.55% [7].

Based on the problem, it is necessary to optimize and design the infrastructure cache on web server in the Blast Compute website order to produce scores performance metrics which have implications for improving user experience, engagement, retention, conversion, and SEO ranking in search engines. [7]

2. RESEARCH CONTENTS

2.1 The cornerstone ofTheory

2.1.1 Optimization

Optimization an attempt to achieve a minimum or maximum value in order to get the output / output in accordance with expectations [6].

2.1.2 Cache

Cache is hardware or software that is used to store something, usually data, for a while in a computing environment [5].

A smaller amount of faster and more expensive memory is used to improve the performance of recently accessed or frequently accessed data that is temporarily stored on quickly-accessible storage media that is local to the client cache cache and separate from mass storage. Cache is often used by clients cache, such as CPU, application, web browser or operating system (OS).

1. Browser / HTTP cache

Browser cache helps simplify and make loading faster [7]. With browser cache, the web server no longer needs to make requests and data transmissions to display the sites that you want to visit in the browser [8]. With cache, the data needed to display the site you want to visit is already on the computer. That way, the site will have a time faster loading and data from the site can be accessed as quickly as possible. All web browsers from Google Chrome to Firefox have cached browsers.

2. OpCode Cache

OpCode Cache is a cache in the script PHPby storing the results of a compilation of PHP. So that the process to execute PHP is shorter without having to go through the stages of parsing and compilation and stored in memory [0].

3. Object Cache

With object cache, object data can be stored locally so that it does not need to be fetched constantly for additional requests. Thus, object cache helps increase the speed and performance ofapplications web[5]. An object is a data set that includes a word, video, or image document. If the object is cached when the user requests for information, it can be transferred directly from the cache local, rather than requesting it from the server.

This is the main benefit of the object cache. If users request data that hasn't changed, they can access it without using a server, making things much easier, and faster. Because the user is not left waiting for the content to load, and bandwidth is not wasted [5].

4. Page Cache

Page cache has similarities with caches other. One of the benefits of doing a page cache is to increase the speed of time loading a page webto provide user experience a better[4]. A cache page stores the page web complete to be displayed later to visitors. This data is stored in an unused portion of RAM and thus has no real impact on memory [5]. Even if it is used to store this information, the computer may still register this portion of the computer's memory as available or even empty. If the data is read again later, it will be read from cache this that is already in memory.

5. Content Delivery Network Cache

CDN cache is a form of wider data storage. With the cache CDN,content websiteis static added to proxy servers that are distributed globally. This allows visitors from all over the world to download your site's content faster, thus speeding up site load time.

Cache CDN also helps owners to website reduce costs, eliminate the pressure of the original server, and put the focus on smaller local servers located globally where visitors can access data locally [5].

2.1.3 Litespeed

LiteSpeed Web server (LSWS), is asoftware web server. This is web server the 4th most popular, estimated to be used by 3.5% of websites by October 2018. [3] LSWS was developed by LiteSpeed Technologies privately owned. This software is an Apache drop-in replacement, which means using the same configuration format as Apache. [10]

LiteSpeed web server improves the performance & scalability of a web hosting platform through infrastructure cache an efficient and effective using LSCache. LSCache can cache this page in full. LSCache Plugin can also be used to configure HTTP Cache, Object Cache, CDN Cache, and OpCode Cache. A infrastructure cache good gives Litespeed the ability to serve thousands of clients simultaneously with minimal server resource usage such as memory and CPU. The unique code developed and optimized from the web server LiteSpeedimproves PHP performance and also serves websites stati cfaster than Apache. Litespeed has the ability to handle sudden spikes in traffic and help manage DDOS attacks without DDOS mitigation hardware. [9]

2.1.4 Google Pagespeed Insight

PageSpeed Insights (PSI) is a tool for measuring page performance on mobile and desktop devices, and provides suggestions on how these pages can be improved using a Lighhouse Engine. [9]

At the top of the report, PSI provides a score that summarizes page performance. This score is determined by running the Lighthouse to collect and analyze lab data about the page. Scores of 90 or more are considered fast, and 50 to 90 are considered average. Below 50 is considered slow.

PSI presents this metric distribution so developers can understand the range of FCP and FID values for the page or origin. [9]

Performance metrics are scoring systems from the Lighhouse Engine of five parameters namely

First Contentful Paint, First Meaningful Paint, Speed Index, First CPU Idle, Time to Interactive, and Estimated Input Latency. Lighthouse gives a performance score between 0 and 100 which is calculated overall from the five parameters [9]. The score indicates the lowest score. A score of 0 usually indicates an error with Lighthouse. A score of 100 is the best score that represents the 98th percentile which indicates the best performance of a website. A score of 50 on the other hand represents the 75th percentile [9]. There are 6 metrics used to produce the total score from the Performance Score including:

1. First Contentful Paint

First Contentful Paint (FCP) measures the time from navigation to the time when the browser renders the first bit of content from the DOM (Document Object Model). This is an important milestone for FCP users provide feedback for how thepage web in load the actual[9].

2. First Meaningful Paint

First Meaningful Paint (FMP) measures how long the main content of a web will be displayed. This audit section identifies how users perceive when the main content is actually displayed [9].

3. Speed Index

Speed Index shows how quickly a page is clearly visible. The smaller the score, the better [9].

4. First CPU Idle

This metric measures when a page at a minimum starts interactively. First CPU Idle marks the start time when the main thread is ready to handle input. [9].

5. Time to Interactive

Time to Interactive is the amount of time needed for a page to be fully interactive [9]. This can be indicated from the following:

1. The page displays useful content that is measured from First Contentful Paint.

2. All event handlers are available for every visual element on the page.

3. The page responds tointeraction user in at least 50ms.

6. **Estimated Input Latency**

Estimated Input Latency is an estimate of how long it takes a website to respond to user input in milliseconds, while loading a page. If the latency is higher than 50ms, then the user will perceive the website as laggy. [9].

2.2 Methodology

System Analysis and Design will be carried out by following the flow of the PPDIOO method, covering several processes of activities that can be seen in Figure 3.1. Starting from Prepare, Plan, Design, Implement, and Optimize [1]. But thephases Implement and Optimize will not be passed because they are only limited to the Staging Environment [2].

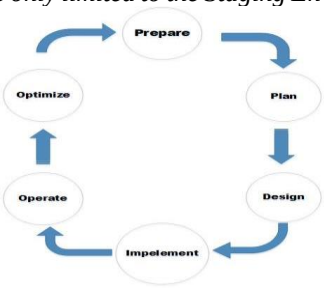

Figure 1. *PPDIOOMethod*

2.2.1 Prepare

Prepare is the initial stage in system design. At this stage be carried Gathering Requirements and an overview of the infrastructure to be designed will out, namely High Level Design.

1. Gathering Requirements

From the results of the interview with the IT Division of PT. Blastech Digital can be summarized into 4 points namely

1. *Blastech Digital IT Division wants efficiency of resources on their servers.*
2. *The IT division wants to implementtechnology cache on their web server .*
3. *IT Division Blastech want to increase the score performance metrics of LighthouseEngine Blast Computeon the site.*
4. *The IT division wants to improve the user experience, engangement, retention, conversion and SEO ranking of their sites throughscores high performance metrics .*

2. High Level Design The

High Level Deisgn is an overview of the overall infrastructure cache that will be designed.

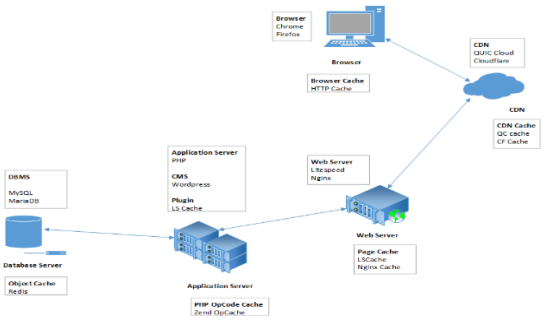

Figure 2. *High Level Design*

2.2.2 Plan

Phase plan consists of two stages, namely Site Requirements Specifications, namely the stage in analyzing hardware and software requirements. And the Solution Test Plan stage where the researcher audits the current system, the Blast Compute site using the Lighthouse Engine.

1. Site Requirements Specifications The

Following are the steps to

1. *Hardware Requirements Requirements.*

Table 1. *Hardware Requirements Analysis*

No.	Function	Hardware	Total	Cost
1	Web server Litesped	of Cloud Server	1	Rp750000 / month
2	CDN	Object Storage	1	Rp750000 / month
3	Client PC	or Laptop PC	1	$ 0 / month

2. *Analysis Software Requirements.*

Table 2. *Hardware Requirements Analysis*

No	Function	Software	Version	Cost
1	Web server	Litespeed Starter	5.3.7	Rp0 / month
2	Operating System	Ubuntu	18.04	Rp0 / month
3	Control Panel	OnyxPlesk Web Admin	17.8.11	Rp0 / month
4	MySQL	Database Server	5.7 .25	Rp0 / month
5	CMS	Wordpress	5.1.1	Rp0 / month
6	Plugin Cache Nginx	W3Total Cache	0.9.7.3	Rp0 / month
7	LS Cache	Plugin LS Cache	2.9.7.1	Rp0 / month

2. Solution Test Plan

Following are the results of audits performance metrics from the Lighhouse Engine at the ongoing stage of the Blast Compute site. Here are scores from Blast Compute sites that have not been optimized and have not yet implemented Litespeed cache.

Table 3. *Performance metrics*

Metrics	Time	Information
First Contentful Paint	12.3s (slow)	FCP marks the time when the text or image begins to be painted by the browser.
First Meaningful Paint	14,5.0 s (slow)	FMP measures when the main content of the page is displayed.
Speed Index	19.3s (slow)	SI measures how quickly the page content is loaded.
First CPU Idle	26s (slow)	FCI measures how fast minimal interactive pages are and is ready to accept user input.
Time to Interactive	26.3s (slow)	TTI measures how fast the full interactive page is.
Max Potential First Input Delay	1030ms (slow)	The maximum potential time the user feels when entering input. Recommendations <50ms
Performance metrics Score	0 (slow)	Total scores in the range (0-49) are categorized as slow.

2.2.3 Design

In this phase, theinfrastructure will be designed LS Cache to optimize the Blast Compute site. In this phase there will only be one stage, namely Low Level Design.

1. Low Level Design

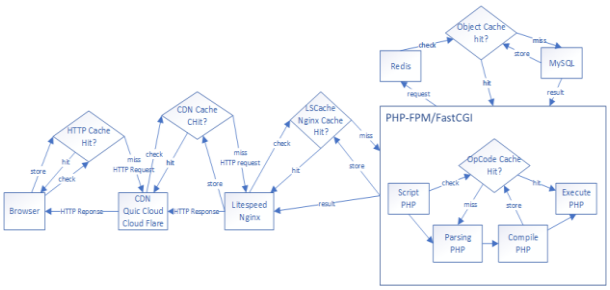

Figure 3. *Low Level Design*

Low level Design provides the Litespeed infrastructure design Cache which will be applied to the Blast Compute site where the scheme can be seen in Figure 3.3.

1. *Browser / HTTP Cache*

HTTP Cache or Browser cache is a strategy cache client-side by storing static assets on user devices. So that users no longer need to make requests to the server but directly accessed from local storage from the user's device. cache TTL on HTTP Cache can be configured at the CDN or web server level. [7] Following is the scheme of how it works

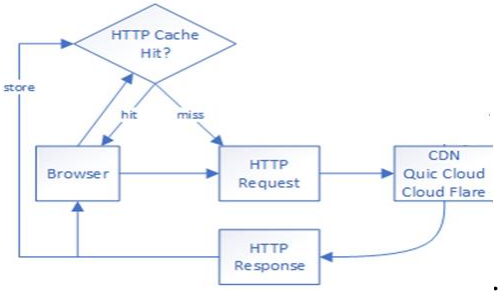

Figure 4. *Browser / HTTP Cache*

a. *HTTP Cache hit.*

1. *The browser checks whether HTTP cache is available in the browser.*
2. *If cache is available or hit, then only make requests via HTTP / 2 for resources dynamicto Cloudflare, and download all static resources from the local storage browser.*
3. *The browser receives responses via HTTP / 2 over QUIC in the form of dynamic content from QUIC Cloud.*

b. *Cache miss HTT.*

1. *The browser checks whether HTTP cache is available in the browser.*
2. *If not available or miss available, request static and dynamic content to the Cloudflare CDN.*
3. *Quic Cloud CDN sends responses in the form of dynamic content via HTTP / 2 Over QUIC and responses in the form of static content via HTTP / 2 push are then displayed to the Browser and the Browser also stores cache static content from Cloudflare CDN to HTTP Cache with Cache TTLthat has been defined from the web server Litespeed for 1 year or 31536000s.*

2. *CacheCDN*

Content Delivery Network caches to be a Reverse Proxy in front of the webserver.In CDN Cache, two CDNs will be used, namely Cloud flare to cache static content and QUIC Cloud to cache dynamic content. The following scheme works:

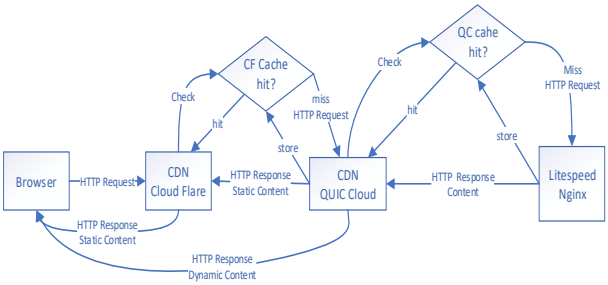

Figure 5. *CDN Cache*

1. *Cloudflare*

a. *CF cache hit*

1. *CDN Cloudflare checks whether there is a cache available.*
2. *If the CF is cached hit, download a cache static from Cloudflare with "content encoding": br (brotli compression) and continue to only request dynamic content to the Quic Cloud CDN.*
3. *Cloudflare continues requests via HTTP / 2 for dynamic content to the Quic Cloud CDN Quic Cloud*
4. *CDN sends responses in the form of dynamic content via HTTP / 2 over QUIC to the browser with "content encoding": br (brotli compression).*

b. *CF cache miss*

1. *Cloudflare CDN checks if there is cache available.*
2. *If the cache is miss, Cloudflare makes requests via HTTP / 2 static and dynamic content to the Quic Cloud CDN.*
3. *Quic Cloud provides responses via HTTP / 2 over QUIC in the form of static and dynamic content. Static content will be sent to Cloudflare and cache static content to CF Cache.*
4. *Then the CloudFlare server will forward the response in the form of static content to the browser via HTTP / 2 with "content encoding": br (brotli compression)*
5. *Quic Cloud provides responses via HTTP / 2 over QUIC for dynamic content to browsers with "content encoding": br (brotli compression).*

2. *Quic Cloud*

a. *QC Cache hit*

1. *CDN Quic Cloud checks QC Cache, is cache available?*
2. *If the QC cache is hit, download cache the static and dynamic contentfrom Quic Cloud with "content encoding": br (brotli compression) and images in format Webp or SVG.*

b. *QC Cached miss*

1. *CDN Quic Cloud checks if there is cache available.*
2. *If the cache is miss, theserver Quic Cloudmakes requests via HTTP / 2 over QUIC for static and dynamic content to the web server Litespeed.*
3. *Web server Litespeed responds via HTTP / 2 over QUIC for static and dynamic content. Static and dynamic content will be sent to theserver QUIC Cloudand cache static and dynamic content to the QC Cache.*

3. *Page Cache*

Page Cache is a strategy cache to reduce the burden on application servers and databases because the page cache goes through the PHP execution process (parsing, compiling, executing) and querying the database to MySQL by caching dynamic pages in the form of static pages. cache Thisis very effective for speed up requests to a dynamic site page repeatedly without the need to execute PHP and query to the database. Following is the scheme of its work.

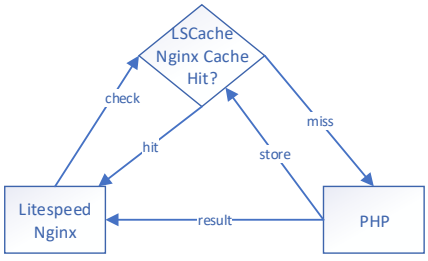

***Figure 6. Page** Cache*

a. *Litespeed Cache hit*

1. *Web server Litespeed checks whether LSCache is available?*
2. *If LSCache hit, load it cache of pages from the Web server Litespeed Optimized.*

b. *Litespeed Cache*

1. *Web server Litespeed checks whether LSCache is available?*
2. *If LSCache miss, do a request to the backend application server PHP.*
3. *The Application Server PHP will generate the page requested and send it to the web server then store the page generated to Litespeed Cache.*

At thelevel Page Cache, it can also be configured to process static HTML, CSS, JS assets and combine CSS and JS files. Page Cache in Litespeed Cache also plays a role in minimizing the property of async and defering to static files can be downloaded asynchronously or defered so as not to become rendering blocking.

The process of compressing text and images using brotli and gzip can also be done using Litespeed cache. And so that images can be presented in the next generation format such as WebP must be configured on the web server. So the process Page Cache by the web server gives a impact big besides the CDN so that it can pass an audit from the Lighhouse Engine.

4. *OpCode Cache*

OpCode Cache shortens the execution time by executing the cache of compiled scripts PHP that have been parsed and compiled. The working scheme of OpCode Cache can be seen in Figure 7.

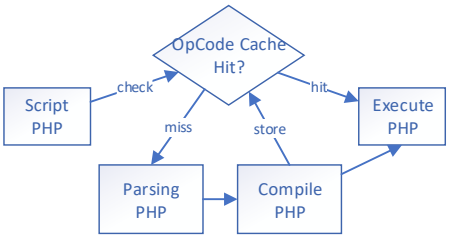

***Figure 7.** OpCode Cache*

a. *OpCode Cache hit*

1. *Backend PHP to check whether OpCode Cache is available?*
2. *If the OpCode cache hits, do the PHP execution process immediately.*

b. *OpCode Cache Miss*

1. *Backend PHP checks whether the OpCode cache is available in memory?*
2. *If the OpCode cache is miss, do the process script parsing PHP.*
3. *After parsing thescript successfully PHP, proceed to compile the PHP from the results parsing.*

4. *After theprocess is compiling PHP successful, execute the results PHP compiling, and save the results PHP compil ein the OpCode Cache in memory.*

5. *Object Cache*

Object Cache or also known as Database Cache is a strategy cache server-sideby reducing requests query to the database server and storing them into objects in memory so that they are easy to retrieve, and eliminates the need for repeated access to the database. Object Cache greatly reduces the time needed to retrieve query results from the database

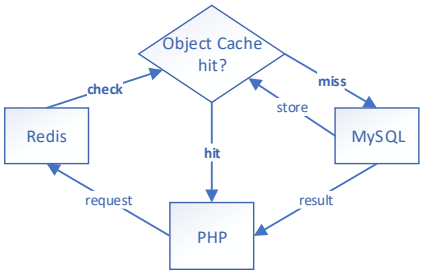

Figure 8. *Object Cache*

a. *Object Cache hit*

1. *Application Server PHP making requests to Redis via PORT 6379?*
2. *Redis checks whether the Object Cache is available in memory?*
3. *If the Object Cache hit, send the results to the Application Server PHP*

b. *Object Cache miss*

1. *The Application Server PHP made a request to Redis via PORT 6379?*
2. *Redis checks whether the Object Cache is available in memory?*
3. *If Object Cache miss,do request a query to the database MySQLserver.*
4. *MySQL sends the results to the Application server PHP and stores the results as Object cache in memory.*

2.2.4 Implementation

In this phase, the installation and implementation of the software configuration and Litespeed performed cache are according to the specifications in the phase Design PPDIOO of the Litespeed web server.

There are three stages in this phase, namely the Implementation Log, Network Ready For use (NRFU) Test, and NRFU Test.

1. *Implementation Log*

At this stage, the steps and description of Installation, configuration are carried out. and the optimization process Litespeed Cacheon thesite Blast Compute.

2. *NRFU Testing*

At this stage testing after the implementation and optimization of Litespeed web server after the implementation of Litespeed Cache.

1. *Mobile Mode, 4G Slow Throtling, Clear Storages*

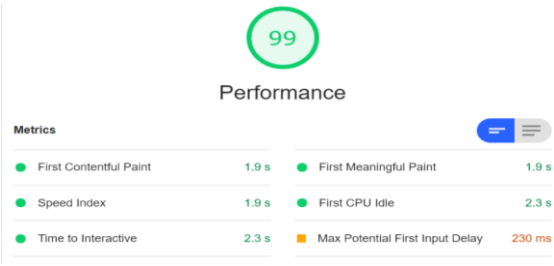

Figure 9. *Testing 1*

2. *Mobile Mode, 4G Slow Throttling*

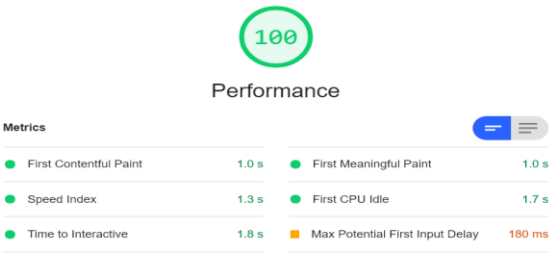

Figure 10. *Testing 2*

3. *Mobile Mode, No Throttling, Clear Storages*

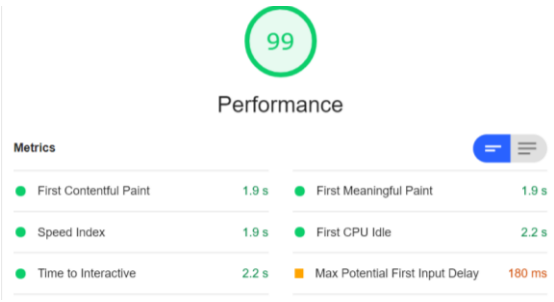

Figure 11. *Testing 3*

4. ***Mobile Modes, No Throttling***

Figure 12. *Testing 4*

5. ***Desktop Modes , Throtling Slow 4G, Clear Storages***

Figure 13. *Testing 5*

6. ***Desktop Modes, 4G Slow Throttling***

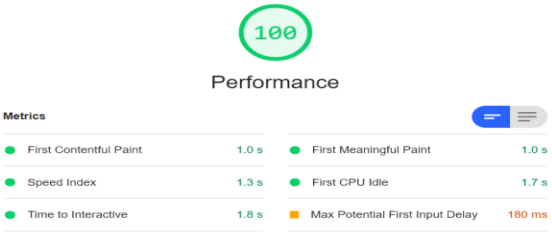

Figure 14. Testing 6

7. ***Desktop Modes, No Throttling, Clear Storages***

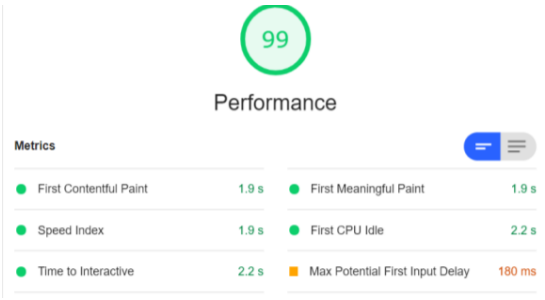

Figure 3.11 *Testing 7*

8. ***Desktop Modes, No Throttling***

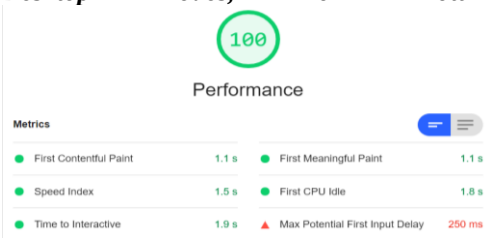

Figure 3.12 *Testing 8*

2.3 Results and Discussion

2.3.1 NRFU Testing Report

NRFU Testing Report is a data report from NRFU Test whose results can be used to provide interpretation of test results.

1. ***Test Results 1***

Table 4. Performance metrics Report 1

Metrics	***Time***
First Contentful Paint	*1.9s (fast)*
First Meaningful Paint	*1.9s (fast)*
Speed Index	*1.9s (fast)*
First CPU Idle	*2,3s (fast)*
Time to Interactive	*2 , 5s (fast)*
Estimated Input Latency	*230ms (medium)*
Performance metrics Score	***99 (fast)***

2. ***Test Results 2***

Table 5. Performance metrics Report 2

Metrics	***Time***
First Contentful Paint	*1.0s (fast)*
First Meaningful Paint	*1.0s (fast)*
Speed Index	*1.0s (fast)*
First CPU Idle	*1.0s (fast)*
Time to Interactive	*1.3s (fast)*
Estimated Input Latency	*180ms (medium)*
Performance metrics Score	***100 (fast)***

Test Results 3

Table 6. Performance metrics Report 3

Metrics	***Time***
First Contentful Paint	*1.9s (fast)*
First Meaningful Paint	*1.9s (fast)*
Speed Index	*2.2s (fast)*
First CPU Idle	*2.2s (fast)*
Time to Interactive	*2.5s (fast)*
Estimated Input Latency	*180 ms (medium))*
Performance metrics Score	***99 (fast)***

3. ***Test Results 4***

Table 7. Performance metrics Report 4

Metrics	***Time***
First Contentful Paint	*1.1s (fast)*
First Meaningful Paint	*1.1s (fast)*
Speed Index	*1.9s (fast)*
First CPU Idle	*2.3s (fast)*
Time to Interactive	*1.9s (fast)*
Estimated Input Latency	*250 ms (moderate)*
Performance metrics Score	***100 (fast)***

4. Test Results 5

Table 8. Performance metrics Report 5

Metrics	*Time*
First Contentful Paint	*1.7s (fast)*
First Meaningful Paint	*1.7s (fast)*
Speed Index	*2.4s (fast)*
First CPU Idle	*2.2s (fast)*
Time to Interactive	*2.2s (fast)*
Estimated Input Latency	*420ms (medium)*
Performance metrics Score	***98 (fast)***

5. Test Results 6

Table 9. Performance metrics Report 6

Metrics	*Time*
First Contentful Paint	*1.1s (fast)*
First Meaningful Paint	*1.1s (fast)*
Speed Index	*1.6s (fast)*
First CPU Idle	*2.8s (fast)*
Time to Interactive	*1.9s (fast)*
Estimated Input Latency	*420 ms (moderate)*
Performance metrics Score	***100 (fast)***

2.4 Test Results 7.

Table 10. Performance metrics Report 7

Metrics	*Time*
First Contentful Paint	*1.1s (fast)*
First Meaningful Paint	*1.1s (fast)*
Speed Index	*1.3s (fast)*
First CPU Idle	*2,2s (fast)*
Time to Interactive	*2,3s (fast)*
Estimated Input Latency	*450 ms (medium)*
Performance metrics Score	***100 (fast)***

2.5 Test Results 8.

Table 11. Performance metrics Report 8

Metrics	*Time*
First Contentful Paint	*1.0s (fast)*
First Meaningful Paint	*1.0s (fast)*
Speed Index	*1.5s (fast)*
First CPU Idle	*1.6 seconds (fast)*
Time to Interactive	*1.7s (fast)*
Estimated Input Latency	*330 ms (medium)*
Performance metrics Score	***100 (fast)***

From the results of the above tests the results of each difference in desktop and mode mobile mode, as well as throttling and clear storages with performance metrics scores ranging between 99-100. Here are the average scores:

Total Score = (99 + 100 + 99 + 100 + 98 + 100 + 100 + 100)

Average Score = 796/8 = 99.5

The above experiment proves that there have been many aspects and parameters of the Lighouse Engines that have passed audits with an average score of 99.5 and are in the range ***fast(90-100).*** *Ascore high performance metrics indicates a site with a relatively fast loading of user perception..*

3. CLOSING

3.1 Conclusions

After the design, optimization, and implementation Litesped Cache on sites Blast Compute the mobile mode and the desktop, with various variable throttling and clear storages, we can conclude:

1. *The use of server web-based event-driven and application cache proved capable of providing efficiency in the use of resources seen fromscore performance metrics the highon the web server.*
2. *Optimization of the web server and Litespeed Cache succeeded in increasing the score performance metrics of the Google Pagespeed Insight API on the Blast Compute site and the average score of 99.5 High*
3. *score performance metrics of 99.5 on the Blast Compute site will have implications for increasing user experience, engangement, retention , conversions / sales from visitors and SEO rankings.*

3.2 Suggestions

This research still has shortcomings in exploring the possibility of other variables and parameters such as operating system differences, differences in web servers, application servers, etc. Suggestions for the development of this research, by giving several points, among others, are as follows:

1. *For research optimization and design of the cache is expected to further expand the problem by adding parameters and other variables such as differences in operating systems, web servers, and application servers.*
2. *In designing Litespeed Cache infrastructure can add other caching technologies such as Varnish and Reverse Proxy Cache.*

ACKNOWLEDGMENTS

In this study we would like to thank PT. Blastech Digital because for helping us carry out this research to the finish we hoped for and the researcher thanked Mr. Iskandar Ikbal, ST, M.Kom as a supervisor who helped and gave me enthusiasm to complete the research.

References

[1] Solikin, Imam, "*Penerapan Metode PPDIOO dalam Pengembangan LAN dan WLAN*," vol. 07, no. 01, hal. 65–73, 2017.

[2] Wilkin, Seans, (2011, 5 April). Cisco's PPDIOO Network Cycle 2019:http://www.ciscopress.com/articles/article.asp?p=169788&seqNum=2

[3] Nguyen, VanNam, *Comparative Performance Evaluation of Web servers* , (6), 28–36.

[4] Luthfi Muhammad, Data Mahendra, Widhi Yahya. (2018). *Perbandingan performa reverse proxy cache nginx dan varnish pada web server apache. Pengembangan Teknologi Informasi Dan Ilmu Komputer*, *2*(4), 1457

[5] Mahad, Fairus Safwan., & Wan Kadir, Wahn Mohd. Nasir (2013). *Improving web server performance using two-tiered web cache*. *Journal of Theoretical and Applied Information Technology*, *52*(3), 243–251.

[6] Syahrul Mauluddin, Iskandar Ikbal, Agus Nursikuwagus. (2018). *Optimasi Aplikasi Penjadwalan Kuliah Menggunakan Algoritma Genetik 2*(3), 792-779

[7] Manhas, Jatindes (2013). *A Study of Factors Affecting Websites Page Loading Speed for Efficient Web Performanc*e, (3), 32–35

[8] Havala, Olavi. (2018). *Analyzing and Improving the Loading Performance of Large Scale Websites on Mobile Devices* , (6), 27–36

[9] Pagespeed, Google (2018, 15 Mei). *Why Performance Matter* Dikutip 15 Mei2019: https://developers.google.com/*web*/fundamentals/performance/

[10] *Web server*, Litespeed (2017, 16 Mei). *Litespeed Wiki* Dikutip 1 4 April 2019:https://www.litespeedtech.com/support/wiki/doku.php/litespeed_wiki