TELKOMNIKA Indonesian Journal of Electrical Engineering
Vol.12, No.3, March 2014, pp. 2042 ~ 2050
DOI: http://dx.doi.org/10.11591/telkomnika.v12i3.4660

Study of the Principles and Models in Web Performance Optimization

Xin Wang
Computer engineering college, Weifang University, Shandong 261061, China
e-mail: www268@126.com

Abstract

With the development of Web, the optimized question is becoming more and more prominent, so the Web performance optimization is inevitable. The important principle of Web Performance Optimization is understanding, and recognizing that gain must lose, the repayment is decreasing progressively, and return is diminishing at the same time, the optimized goal is the overall performance, and start from the highest level to optimize will obtain the biggest. Models of improving Web performance are as follows: sharing costs, high-speed caching, parallel processing, profiles, and using known information. To optimize the performance of Web database is also critical, such as to analyze and research from the cache, statements, tables, connection pooling, query, index, and several other aspects. Based on this study, given the crucial Web performance optimization recommendations, which will improve the performance of Web usage, accelerate the efficient use of Internet has an important significance.

Keywords*: web, principle, model, performance optimization*

1. Introduction

Nowadays Web has grown into an essential tool to disseminate information from a novelty. With the continuous development of Internet, the problems of Web performance are becoming more and more seriously because of massive information and modern affairs. Therefore, to optimize the performance of the Web is necessary. There are some general principles suitable for the general and patterns unify a variety of specific solutions. Some optimization principles and methods are discussed in several literatures [1-5], and some database performance optimization strategy are introduced in papers [6-10]. The following are the analyses and summaries of these principles and patterns in detail.

2. The Principles of Web Performance Optimization

2.1. Understanding is the Basic Factor

You may hurt yourself when you enter a house in the dark. Things are sure to be easier if you turn on the light. The light can give you the information and enable you to optimize the path. So does the Web Performance Optimization. Problems are easier to be solved if you consider them in the mind more clearly. By now, the light to guide you includes the workstation, the software, and the router reference manual. Therefore the first thing you must do is to read the reference manual and understand it.

When the reference manual has the key to reduce your pains, they seem rare. It is feasible to measure performance by changing the setting, but it is more important to figure out why there are differences between different settings, and their relationship with other settings and the subsystems. This knowledge can be found in the reference manual. The increase in performance here may cause degradation in performance there. If you do not know what to pay, then you will not know whether it is worthwhile to modify the system [1].

2.2. When There is Gain, There is Loss

Only after doing the research can you know whether it can improve a system's performance. But you also brave the risk that you waste a lot of time on research only to find you can not do anything to improve the performance. The risk is to be greater especially when the funds are short and the time is urgent. You must carefully analyze the difficulty of the

Received August 25, 2013; Revised September 29, 2013; Accepted October 14, 2013

problem and possibility of the harvest. If the gains can be large, then it should be determined to find a solution to improve performance, otherwise it is not worthy to spend the time [1].

Although the key of optimal performance is to do more with less loss, but it is inevitable to pay for it as long as you make effort to understand and solve this problem for any increased performance. In some cases, you have to buy new hardware, re-planning system architecture, and you may also reduce system portability, maintainability, security, reliability, or increase development time. You can let the system bus running day and night to improve server performance, but such systems are also more likely to collapse. To improve performance, you can remove the firewall and any encryption device, but you will also be exposed, more vulnerable to external attack. In fact, it will inevitably hurt some other different usage patterns when optimizing the known usage patterns.

2.3. The Repayment is Decreasing Progressively

You can know when you complete the task of the hardware optimization the size of the investment. When it comes to optimizing the system, it is usually easy to find the simplest problem and fix it. With the optimization of the system, the increase of the performance becomes more and more difficult, and it is more dependent on the particular configuration and usage patterns [2]. When the investment does not match with the return, the optimization is over. Some clues to explain the end of the optimization process, at least in the following cases:

a. When users are not aware of any performance improvement no longer.
b. When changed the good programming style for the performance to make the code can not be transplanted and difficult to maintain.
c. When you consider programs written in assembly language.
d. When the total cost per page view is bigger than the person employed in accordance with customer's request to call them and fax them.
e. When you're bored.

The optimization goal is changing, because the configuration, usage patterns and available components is constantly changing, so it is impossible to achieve and maintain optimal performance. In the long run, using standard protocols and API to optimize are better than the use of proprietary or solution of their own invention. Then the efforts of the performance optimization can be rewarded, while the system between the different generations have portability, and will have a more long-term investment returns.

2.4. Portability Will Reduce the Performance

There is a conflict between the performance and the probability. The best performance only can be get after the optimization of a special environment, while portability was defined as definite function in many different environments. Because they can not optimize all of the circumstances, it is sometimes necessary to exchange the performance for portability.

Fully optimized software is limited to a specific platform, because it must take full advantage of the platform’s available features, such as the use of dedicated CPU registers and systems. On the other hand, portable software does not take full advantage of the characteristics of a specific platform, because if so there is no portability. The optimization method can be used in a particular mode at another level, when the environment changes, it is likely to reduce the performance rather than boost it [3].

If the portability of software is not particularly valuable, then the loss of portability of the software on the optimization will not have much impact. But that is not the case, the portability of the source code level indicates that when it is running on other platforms it simply need re-compile the code instead of rewriting it. This saves development costs and provides a larger market. Portability of the object code level, such as Java and Smalltalk, is the ideal goal for developers and users. Developers can focus on writing code instead of busy porting the code, and users can choose their favorite platform. There is no good for users to be limited in a platform.

Following the public network standard can obtain another kind of probability, so the software cannot be transplanted it may at least correspond with other computers. The emergence of Web HTTP protocol is the direct cause of portability. HTTP may not bring the optimum performance to all computers, but since realized in so many machines, it is very important to provide the value of information sharing. Any Web server can be visited by any browser because they both use the same language. You can use performance in exchange for

the improvement of portability, but it is only an ideal of the transaction, eventually you will pay a very high price.

2.5. The Safety Protection Conflicts with the Performance

System security is another limitation, and all restrictions will reduce the freedom of optimizing performance: SSL connection establishment takes time, the firewall will slow down the transmission speed of data packets, entering the password will slow down the user's speed. Security is necessary, but its effect on performance is often very significant.

2.6. Abstracts Conflict with the Performance

In the "higher" level of abstraction for programming, more details need to be considered, it will only get the performance in general, and will not be "the best performance." Whether using the high-level programming language or automatically generating SQL, program deals with the details of the system optimization procedures will lose the simplicity on understand and control. Sometimes this is good, but sometimes not.

2.7. Memory Has a Certain Structure and Mode

Web can be seen as simply the slowest and most expensive kind of memory. Although the Web is not really in your computer, and in most cases it is just read-only, but it actually matches with the other parts of memory structures [4].

Memory structures are different at every level of cost performance, and price is often associated directly with the access speed. Recently used data are cached normally by the next faster level, the purpose of caching is to obtain the best performance using the fastest memory, or to occupy the least area of the high-speed cache. Moreover, the lost of the high-speed cache costs great price because the high-speed cache can not only improve the performance but also decrease the cost. The data exists in high-speed cache is cheaper than retransmit it if you want to use the data for many times.

If memory access is completely random, then the cache does not help the performance, and the high-speed cache memory will continue to be covered in the random part of the data. Therefore you are less likely to access the same data twice within a short time. However, there is a pattern of memory access. The recent memory address and the adjacent memory address may be accessed immediately. This model is called the relevance of addresses. This is the reason for the actually effective performance in the calculation of high-speed cache of recently visited memory address and the adjacent memory address [5]. For example: It is effective to use this mode in the Unix file system buffer and cache Web browser.

2.8. The Optimized Goal is the Overall Performance

It is considered to be in optimal condition when there is not a bottleneck in Web system. However, this did not mention the total throughput of the system-a very low system throughput can be in a technically optimized state. The optimization goal is not wasting any capacity, in other words, to achieve the ultimate state of each component at the same time. Some parts of the system are more durable than some other parts of little significance.

There is also a fact that the least effective optimized system is the most likely to be optimized. When a component used up all the time, it is a clear need to focus on solving this problem. After handling the problems one by one, round of this cycle begin again till all components are equal when the issue so far. The optimization comes to the end at this point.

Because the Web system is usually dynamic, ensuring to identify the weakest link in a timely manner at all times is almost impossible. Optimal performance is not to spend all the time to track and trace volatile bottleneck. To determine the appropriate performance is more meaningful, even if some components have not been fully utilized. However, when most of the components are not fully utilized, there can be a problem.

In the Web, the smaller the packet is, the arrival time will be faster. No matter how to optimize your system, too much content will crash the performance of the system. Therefore, to maintain the content concise, not to receive arbitrarily large data from the user is reasonable [1].

2.9. The Gains of the Optimization are Biggest when Start from the Highest Level

It is difficult to make performance of the system optimization through modifying the parameters repeatedly. Only through a careful analysis of system architecture can you remove

every part of the processing steps that can be canceled in order to optimize the performance of the system. To gain the most, you have to analyze system's architecture in the highest level at first. This can also reduce the risk of the work. It may just be futile if only optimizing the small details of the level because you may find later that the entire processing steps can be canceled.

3. The Model of Web Performance Improvement

The performance improvement may divide into groups according to the model. This way is more advantageous than each concrete proposal to the work. The following is about the analysis on the technical model of performance improvement.

3.1. Sharing Costs

For the purpose of economy, performance improvements usually involve how to share the cost between multiple transaction processing:

a) HTTP allows a single TCP connection for downloading multiple files. This feature is known as the "persistent connection". Therefore, establishing and canceling a TCP connection involves more than one file instead of only connecting with a single file.

b) Design image map is another example. It is necessary to sent a large image instead of multiple small images to users. If the original image is clickable, the same functionality can be get by way of changing this big picture into a clickable image map.

c) Java.jar document is similar to the case. Packing the Java.class document can download them through a TCP connection, not through establishing an independent TCP connection to each kind of document. There is a negative influence under this kind of situation, that is, the .jar document will possibly contain some classes that you will never use them.

3.2. Caching Technology

The idea of caching technology is simple: setting those frequently accessed data at hand. The high speed cache technology will come to effect only when some data is more frequent accessed than other data in practice.

a) Obtain better performance by the following approach to the storage space: running the most popular input program to input offline data to the CGI program, and high-speed cache all the results together. Then users can quickly access to static HTML without having to generate dynamic HTML.

b) Increasing the memory can reduce the necessity of the server to find contents on the disk, thereby reducing the access time.

c) Web proxy server caches some of the most popular Web pages, so that organizations can reduce the load on Internet access, and at the same time reduce the time to access these pages.

3.3. Parallel Processing

Many problems profit from solving the same problem through multiple entities at the same time in Web services:

a) Netscape and some other browser can open multiple connections to the server, to issue multiple requests in parallel, and want the server to determine the order of one of the most effective services to these requests, rather than a random order the customers requested.

b) The Java procedure benefits from the multithreading. Multithreading allows other threads to continue to carry out when some threads are blocked. For example, a user of Java application procedure needs to fill in some things on the screen when registering, and then another different thread may use this opportunity to download other kind of documents. It does not carry on the serial process easily unless the order is indeed very important.

c) Symmetric multiprocessing hardware can map multiple threads to multiple CPU, and can execute code in parallel.

3.4. Configuration File

Configuration file can be used to discover the reality of some patterns in use. You can find bottlenecks in your code by using it, and can also optimize the usage patterns. The following "Amdahl's recommendations" to quickly solve commonly encountered problems can be followed.

a) Discover the most frequently visited code from configuration files' code in order to optimize these codes to the largest extent.

b) Configure the users, and put the web site closest to them based on the information.

c) Write down the user download time, and assume that what kind of access throughput they may have, then adjust the content to make them more suitable for the user's access type.

3.5. Using Known Information

Do not underestimate the value of information, even the most trivial information:

a) You know that the next visit is possibly be an image on a HTML page, then, theoretically speaking, the Web server may analyze the HTML page and prefetch the image.

b) You will have the possibility to use to connection once having used it. Therefore HTTP has a lasting connection.

c) If the Web server can identify a particular user's usage patterns; you can optimize these models and prepare its contents, in advance, or the content needs to be dynamically generated.

3.6. Simplified Processing

Sometimes simplifying the process of the matter and reducing its scope can bring lots of gains:

a) It is fast and cheap when there is no connection between built-in modem and system bus. You can not buy the wrong connections just because there are no connections between them.

b) You can shorten the download time by keeping the HTML content small and simple, getting rid of frames and tables, retaining the image as little as possible. Yahoo content is like that.

c) Only use static content instead of CGI to greatly reduce service response time on the cost of decrease flexibility.

4. The Selection Strategy in Web Database

People’s great interest on the Web lies in the relatively cheap and easy access to the global database on the Internet. Most of the information is on the host or in a relational database management system.

At present there are three kinds of standard database classes which have the different request:

a) The inquiry way of read-only database (such as AltaVista) is the individual query.

b) It is usually the need of marketing to inquire complicatedly in the mass data. This is called the data mining. There is a very famous data mining example: it is discovered that the beer and the handkerchief sell together frequently after the grocery store has collected all goods sales situation. Nobody notice this phenomena previously. But both of these two things have sold out and needs to be purchased, which becomes meaningful. As the result of the discovery, the grocery store is sure to put the beer and the handkerchief on the same place routinely. The data mining is read-only, and it is very usually complex, and it need spend very long time, therefore carrying on the public Web visit is not be suggested.

c) Business processing, business will become an important and valuable application fields of Web.

The three kinds of database accesses are different in scalability and ability. Read-only access class can easily be expanded by copying the database. Data-collection database is usually unnecessary to extend the database because few users will make such inquiries. Transaction database is the most difficult to expand because it can only copy the data to a host at any time that can cause a very significant bottleneck.

Planning and optimizing the database is a big area, which is much larger than all the work done together for optimizing the Web services.

Choosing the traditional SQL database has low performance sometimes, but the programming is easier. A low-capacity site can consider using them.

For those small data set that only need a simple way to check, the best option is to download all of the data in HTML form to the client-side, to allow users to use the browser's

search function to obtain the corresponding row. You can consider writing a Java applet and downloaded with the data together when inquiring complicatedly on small data sets. The program represents a user's query interface. You can simplify the query.

One procedure to the access speed on the client side, if the data set is too big, is to carry on the inquiry by applications of conventional CGI, the server API model or Java servlet in the server end. The Unix grep order is quite effective and very easy to be used in CGI. Sometimes, inquiring a simple ASCII data file is able to gain the higher investment rate of return than any forms of database, because programs are very easy. Perl has the Hash table which is very easy to use. The ndbm document in Unix also has a similar function to the Hash table for the people who like to compile CGI with C. C programmers can read a map to the memory and the memory structure directly as a binary file. If the cost of starting CGI can be reduced through running or writing a server API module as a background program, then the performance of this approach will be great.

Finally, if you need to use SQL for complex queries, but having relatively small data set, you can consider using MiniSQL. It is also known as mSQL. The mSQL has a very good performance, moreover, it supports a large subset of ANSI SQL. MySQL is another good choice in small database and it is free.

5. Database Performance Optimization Strategy

In Web, the database may do greater work than the Web server. For example, when accessing to dynamic content, the content of the access is generated by the Web server application if the user click the same connection each time, such as common ASP, JSP, etc. This requires calling frequently the data of the database server and sending to the HTTP request user by converting it into HTML format. This may make the database become the bottleneck of the entire site. Thus, it is necessary to carry on some planning and optimization based on the database.

5.1. Database Connection Caching

It is known that the user will access the database through the Web server, and Web application and database server need to establish a connection first, then deposit and withdraw the data. This kind of connection is closed after the processing had ended. Each time the user repeat the step each time he or she visits. The database connection process dissipates the system resources, moreover, the time expenses are also quite large, and the efficiency is especially low particularly when using public gateway connection CGI (Common Gateway Interface). Under the usual situation, the influence to the system response time produced by connection process is great [6].

The database connection cache refers to the regular connection between the Web server and the database server. The user directly use the already existed connection when need visit the database. The connection still maintains and does not close after the operation had ended. Therefore the user’s procedure of visiting the database is simplified, and then the system's efficiency is improved.

5.2. Preprocessing Statement and Binding Variable

The pre-analytical statement of database can be stored in a specific location together with variables. These variables are called "binding variable." Preprocessing statements have much higher performance compared to the statement which are analyzed and optimized before the implementation. But there are certain expenses at the beginning of establishing the statement [6].

It is the best way to use the preprocessing statement when the carrying-on inquiry is almost exactly the same and only the value of the query is not the same, that is, the structure of the table do not change. However, the storage cost of preprocessing statement is relatively high. Therefore it is only suitable for one-time use, not suitable for circulation.

5.3. Make the Form Non-normalized

Sometimes, the purpose of improving performance can be achieved by simply putting the data with common feature into the same table, which will avoid the high cost of joint operation and make the work of writing queries simple. Because it eliminates the query trouble

for joint multiple table and only need to query a single table. On the other hand, making the form non-normalized increases the possibility of inconsistent data, so as not to mistakenly think that the different data in the table are the same. However, making the form non-normalized will add the administrator's difficulty [7].

5.4. Good Query Mechanism

A good mechanism can reduce the workload of the database query. Now the database contains a lot of optimization procedures, which are divided into two categories: rule-based optimizer and cost-based optimizer. Rule-based optimizer is to optimize under a specific set of rules, while the cost-based optimizer pays attention to the cost of the actual time for a particular query. Some comments can be added for the optimization program in SQL statement on the following principles [8]:

a) The most common query results will be cached up.

b) Query and update firstly for some more restrictive operations, and the remaining part of the data to be processed will be less, thus speeding up the speed.

c) It is best to maintain a large time interval between mutual database operations, which is to execute a small number of larger queries rather than a large number of smaller queries.

d) Limit the lock in the data bits that you really want to lock. If all of the query are locked to the same table, then these queries can only make serial execution, then the performance will be decreased.

e) A bad SQL query will lead to the collapse of the entire database. So do not let the public access to the database unlimitedly. It should be limited even on the intranet.

5.5. The Rational Allocation of Connection Pool

For a large-capacity site, the usage of connection pool is necessary. Cache and connection pool are important techniques in data access and have a huge increase to the performance of accessing the database in some cases, and have been generally supportive to the database field. You can conceive this kind of situation: when you need to drink a glass of water, of course the sooner the better. Usually, the production of a cup of water includes extracting from the water source, purifying through the pipeline transmission and the equipment before reaching to your drinking water container. The above process is integrant, but it is not the repetition of the above process for the production of each cup of water [9]. You can use a larger container to contain lots of water and distribute the water to cups, and your cost is only transfer the water from the large container to cups. You may also turn on the water valve without laying down the pipeline to water source temporarily and purchasing water-purify equipment when massive water is needed Therefore, the Government usually lays pipelines and constructs water treatment station and complete the more difficult work to achieve the purpose of sharing resources. You can have the particular purpose water with containers for your own need. There are many similarities between caching and connection pool to the above specific vessel and the transmission pipeline. They share the same goal, that is, to meet the desires of the user under the premise of shared resources as much as possible in order to improve overall system performance.

Because the establishment of database connection is very time-consuming, it is impossible to establish a database connection to the Web site for each visit. If you are using a single application server (such as Weblogic), you should to configure the connection pool initial capacity as the maximum. This is because creating a new connection takes a long time when needed to increase the capacity of the connection pool. If set the connection pool to maximum at the very beginning, you need not wait for the growing time the connection pool required. However, the shortcoming of doing so is the demand on more database resources, and also may use other resources of application program.

5.6. Do Not Create a Cursor in a loop

It is known that the cursor is a memory region which stores the query results. The cost of creating cursors is large, so to create a cursor inside the loop should be careful.

5.7. Multi-tier System

The setup of browser / Web server / database is a three-tier structure. In a two-tier structure, the database is also the server that can get better performance for a small amount of users, but the scalability is not good. For three-tier systems, they can not take full advantage of the benefits in three-layer protocol if you do not make a plan [10].

When there are lots of users, the three-tier system can reuse the business objects in Web server or application server, and do the operations of reading and writing without immediately access to the database again. This can greatly improve the performance. Multiple databases can be connected into one database by a middle layer, which will distribute the application database. And the middle tier processing monitor can dramatically improve the performance through access to the database, so it is not needed to open and close the connection for each query.

5.8. Integrated Web Server/Database

Some of the database itself is the HTTP server that cancels the layer between the client and the database. Moreover, this may also construct HTML dynamically, such as CGI, and it maintains the status of the transaction too. They can be set to use the same database connection for all requests. Compared to opening a connection for each request, this approach can greatly improve the performance. The side effect is that they are dedicated, not easy to expand. The application program written for a certain server in these mixed servers can not run on other servers. Database allows network access to other databases, but you do not have the performance advantages of dealing with only a process any longer. Here are some Web server / database:

- The IBM Merchant server uses DB2.
- Web Datalade server uses the Informix database.
- NS LiveWire server uses the Informix database, and now also uses the Oracle database.
- Oracle WebServer server uses the Oracle database.
- Sybase Web SQL server uses the Sybase database.

6. Conclusion

Of course, to make the Web the fastest is to do nothing. That is, if you can remove a part of the system, then this part should be removed. One way is to observe whether there is redundancy. If any, remove it. There is a better way, that is, do not use certain equipment as possible as you can. Your users may run their own Web server. You may not need to use Web for a specific business. Thus there is no Web performance issue.

Acknowledgement

This work was supported by Science and technology development projects of Shandong Province(2012YD01031).

References

[1] Patrick Killelea. *Web Performance Tuning*. Beijing: Tsinghua University Press. 2003: 265-285.
[2] Andrew B King. *Web Optimization*. Mechanical Industry Press. 2009: 156-163.
[3] VA Narayana, P Premchand, A Govardhan. Performance and Comparative Analysis of the Two Contrary Approaches for Detecting Near Duplicate Web Documents in Web Crawling. *International Journal of Electrical and Computer Engineering.* 2012; 2(6): 819-830.
[4] Kohavi R Mining. *Mining E-commerce data: the good, the bad, and the ugly.* Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining. 2011: 8-13.
[5] Wang Xin. *Research on Software Optimization Solutions of E-commerce Site*. International Applied Mechanics, Mechatronics Automation & System Simulation Meeting. 2012: 485-488.
[6] Spiliopoulou M, Mobasher B, Berendt B. A framework for the evaluation of session reconstruction heuristics in Web-usage analysis. *INFORMS Journal on Computing*. 2003; (2): 86-88.
[7] Srivastava J.Cooley B.Deshpande M. *Web usage mining: discovery and applications of usage patterns from Web dat*a. Computer Science Bibliography. 2010; (2): 36-37.

[8] Vimal Shukla, Jyoti Sarup. Applied Open Web GIS Server based solution to develop the WebBased Mapping Application using OpenSource Software Server OSS. *International Journal of Informatics and Communication Technology (IJ-ICT)*. 2012; 1(1): 33-42.
[9] Jyoti Sarup, Vimal Shukla. *WebBased solution for Mapping Application using OpenSource Software Server.* International Journal of Informatics and Communication Technology (IJ-ICT). 2012; 1(2): 91-99.
[10] Wang Xin. *Analysis of Database Optimization Principles and methods in Network. Applied Mechanics*, Mechatronics Automation & System Simulation. 2012: 612-615.