# Methodology applied in the construction of a proxy-cache server

Jose A.Uruburo, Carlos E. Gomez, Christian A. Candela, Luis E. Sepulveda
jose.uruburo@utp.edu.co,{carloseg,christiancandela,lesepulveda}@grid.edu.co
Research Group on Networks, Information and Distribution – GRID[1], "University of Quindio"

September 9th, 2014

*Abstract* - In the study of data networks the proxy cache operation is an issue which is not studied in depth, and it is assumed that the student already knows. A caching proxy server intercepts http requests from customers, looks for the object in its cache and returns to the client if it was found, otherwise the cache proxy server asks the source server and saves a copy in its cache before sending it to the client. This paper presents a methodology applied in the construction of a cache proxy server, which consists of five stages: preliminary studies, problem analysis, solution design, implementation and testing; the five stages were controlled by the managing projects methodology. The developed tool was built using agile software development methodology and best practices of software engineering so that anybody can study it and extend it adding new modules or components.

*keywords – Computer networking, software engineering,* t*echnological research methodology, methodology for project management, caching proxy server, http server.*

## I. INTRODUCTION

The cache proxy server are a widely studied technology in the late 90s, where their field of action has been sufficiently explored and made known to the international academic community technology. However, in the study of data networks, its operation is a matter of little interest, it is assumed that the student knows about them, and involving them in exercises with popular implementations such as Squid, they perceive their functionality based on results generated, without details about how the application operates and its behavior. Squid is a cache proxy server for the Web that reduces bandwidth consumption and improves response times by saving reusable files in the cache, provides access control and gives the user a feeling of greater speed Internet access. Squid runs on most operating systems and is covered by the GNU / GPL license. Squid is used by hundreds of Internet Service Providers to improve web access, improve the flow of data between client and server to increase the performance of the cache to save bandwidth. You can also send requests to the web server and build hierarchical cache that optimizes network performance. These features are very functional but are not very descriptive, a situation that it does not facilitate the learning process [24].

The cache proxy servers are intermediaries in web http transactions among clients and origin servers performing data storage functions, connections and processes [7, 19, 20]; they are classified as income proxy located in the access point that they stores copies of frequently used web objects, improves the download speed, reduces bandwidth costs and controls access to ISP ports; outbound proxy located at the exit of the LAN monitors traffic flow between itself and the Internet, serving as a firewall and improving the performance of data traffic; substitute or reverse proxy located opposite to the web server, adds security features and increases the performance and agility of the network; swap proxy located at the points of interconnection between networks lightens the data flow through the routers; and proxy cache used to store copies of shared documents in order to improve the speed and lower the bandwidth costs [19].

Proxy servers act as an origin server when it receives and process client requests and return responses, and behaves as a client when it forwards to the origin server the requests received from client [19, 20].

Private proxy servers serve only one client, while public share multiple users where the level of performance depends on the number of customers and can become a bottleneck and a single point of failure [7, 19, 20].

The percentage of hits per request is called hit rate, corresponds to objects found in the cache that were requested, which corresponds to a measure of the effectiveness of the cache. In other words, the percentage of bytes served by request is called Byte Hit Rate that measures the percentage of bytes consumed in respect of which were requested [19, 20, 21].

The proxy server allows access control, logging, filtering, translation, virus checking, cache storage, parental control, firewall, content routing, traffic modulation, load balancing, deletion of referenced links to the source server, relaying customer requests to the origin server, receiving replies from the source server which goes directly to the client; must also be fast, robust, transparent, scalable, efficient, adaptable and simple. The time taken for the request for a file is related to the number of bytes accompanying the file, the distance between

[1] Research article derived from the Degree work entitled "DESIGN OF A PROXY SERVER CACHE: ARCHITECTURE, ALGORITHMS AND IMPLEMENTATION" of the Specialization in Data Networks at the Technological University of Pereira - Colombia.

**978-1-4799-6717-9/14/$31.00 ©IEEE**

the web server and browser, the communication capacity between different physical links, the efficiency of intermediate and active devices and traffic [2, 7, 19, 20, 21].

The cache is a component that strengthens operation of some attendees by storing information, setting it on a nearby site despite its usual location, for later use, so it is not necessary to reach the place of origin the next time the object is requested, instead it's obtained from the cache [5, 8, 20, 21].

The cache uses memory that assumes buffer functions and object storage recently requested. It must have enough space without this guaranteeing an increase in the hit rate.

An object is stored in the cache if it can be used in response to a future request. To achieve this, criteria are used souch as the status code of the response indicating whether a request was successful; method of request indicating the process to be performed on the object, the cache control directives in the response indicating the term of the object and any amendments; responses validator pointing exerted on the object changes and the conditions for storing in cache; and request authenticator, which identifies whether the objects coming from the source server are safe or not [20].

The cache systems are evaluated based on metrics such as speed, scalability and reliability [1]. Among the services it offers is to reduce the redundancy of the data transfer, bottlenecks, demand on the source server, the delay to access the information, network traffic, latency, bandwidth on source server, and to analyze the behavior patterns of use in an organization, as well if the server is unavailable, the client can obtain a cached copy of the document [1, 19].

The proxy cache server attempts to improve the network performance aiming to reduce the latency perceived by the user and traffic to external networks [5] generating high availability of content and saving bandwidth [6]. Usually, the cache proxy servers perform registration of events during execution in log files whose analysis can be used to adjust the configuration criteria on the same server.

The proxy server stores files in the cache whose size is limited, and as used it fills up to the point of being unable to accommodate new files, for this reason he must choose one of them to be eliminated and give way to a new one; the file to delete is the result of a policy of replacement (or refund) that is implemented based on any of the following algorithms: LFU, LRU or FIFO [5].

The performance of a web caching system depends on the number of clients where as it grows, so does the probability that a stored document is required. In a complementary way the algorithm used for the location and return of documents affects the effectiveness and performance of the proxy cache server [7].

There are several products implementing the service proxy cache, among them are DeleGate [8, 9], EngageIP Cache Server [8], Proxy Firewall [8, 10, 11], Sun Java System Web Proxy Server [8, 10 ], Apache Web Server [2, 8, 13] FortiNet [8, 15], Microsoft Internet Security and Acceleration Server [8, 16] and Squid [2, 8, 17, 18].

This paper presents a methodology to build a cache proxy server, made of 5 stages [2], which are previous studies where it was identified as the problem under study, the lack of depth study of cache proxy servers at the undergraduate level; analysis of the problem where the problem under study was realized by identifying the basic components of a cache proxy server; solution design which proposes as an alternative the construction of a proxy cache server using good software engineering practices; implementation where design is materialized by building a prototype, and testing stage where the work performed is validated after the execution of unit and acceptance tests; all the above is controlled by a methodology for project management [3] which establishes the divided structure of the work, the work plan and risk analysis. DAR methodology was used [4] in the decision-making to choose the model cache proxy server, which for this work was Squid, and also to select agile methodology for software development in this case was XP [23] .

Making an analogy to the work called Minix (Tanenbaum, Woodhul, 1998) where the development of an operating system based on Unix on a smaller scale for easy assimilation by students is illustrated, this work looks in a similar manner to mediate between a server proxy cache, and Squid as a reference model.

The rest of the paper is structured as follows: Section 2 describes the methodology used in the project; Section 3 describes the work performed and the results obtained; and the final section presents the conclusions and future work.

## II. METHODOLOGY

In the execution of the work a set of good practices were used for different moments. GRID adapted the Technological Research Methodology [25] composed of eleven stages to one of five stages [2] which are previous studies, problem analysis, solution design, implementation and testing. Previous studies identified as problem under study, the lack of in-depth study of the cache proxy servers at the undergraduate level, where usually Squid is used as a testing tool difficult to understand. During the analysis of the problem, the situation under study was understood, through the identification of the basic components of a a proxy cache server. In the design of the solution, was proposed as an alternative a proxy cache server using good software engineering practices. In the implementation phase, the design was implemented through building the prototype using best practices for software development. Finally, in the testing stage, the work done was validated on expert judgment by running unit and acceptance tests.

To observe and monitor the implementation of the project, a project management methodology was used, where the structure of division of labor, called from now on EDT was established, along with the work plan and risk analysis. EDT structured objectives in terms of deliverables divided into five stages. As results of the first stage, the development of the work plan was achieved; in the second stage, we identified and characterized popular implementations of server proxies cache, Squid and was selected as a reference model to which we identified its core functions; in the third stage, was identified and selected an agile methodology for software development and a cache proxy server was designed; in the fourth stage, the prototype was built and in the fifth stage validation was performed.

The work plan allowed defining the schedule of activities and milestones: identification of the basic functions of cache proxy server; selection of agile methodology; construction of the prototype; and validation. In risk analysis, threats and vulnerabilities were identified, the probability and adverse impact if any threat were detected was estimated, and also a contingency plan for risk mitigation was considered, as well as actors and responsibilities.

To select the implementation that was taken as a reference model and agile methodology for software development, DAR was used as a reference for decision-making, which suggests the role of decision-maker, responsible for ensuring that the selection criteria and evaluation methods are according to the solution; the group of participants in the decision making process made up of experts in IT infrastructure, with deep expertise in data networking, educational researchers and consultants for IT industry; and those interested in the decision, represented by the population which affects the decision and do not necessarily participate in the process where the decision was taken. On the other hand, it is recommended to submit for consideration by 2-5 alternative solutions that may arise brainstorming among other mechanisms. The evaluation criteria are defined according to the nature of the project, to which a numerical value hereinafter called coefficient is assigned, the value of which defines its level of importance. Likewise, alternative solutions will be evaluated quantitatively in respect of the acceptance criteria. Score dropped by the evaluation is the sum of the product between the coefficients assigned to the criteria and the rating given to each alternative solution.

Among the agile development methodologies were identified and characterized: Scrum, XP and Kanban; where after applying DAR as a methodology, XP was selected consisting of four phases: planning, design, development and testing.

## III. RESULTS AND DISCUSSION

As a result of the first stage of the research methodology applied to the construction of a proxy cache server, the development of EDT makes it possible to observe and control the project that analyzes the internal architecture of a proxy cache server, identifying their core functions implementing them in the construction of a functional prototype for educational purposes, the work plan defining the schedule of activities and milestones was developed, and the risks analysis was suggested.

In the second stage, the structure and basic concepts of a proxy cache server were analyzed and defined, as a result of the list which characterized them, popular implementations of proxy cache servers were identified, then DAR methodology was used to select the implementation that served as a reference model. The respective roles in the decision were defined, in which the interested as beneficiaries of the work were students in training process in networking and communications; evaluation criteria postulated to select the implementation that will be a model of reference were: cost, performance, documentation, ease of being understood and open source, and its coefficients on a scale of 1 to 5, where 1 is the criterion less relevant for the purposes of the project as shown in TABLE I.

TABLE II illustrates the evaluation of alternative solutions postulated where Squid gets the highest score, followed by Microsoft ISA Server, Apache Web Server, Sun Java System Web Proxy Server and Proxy Firewall; therefore Squid was selected as reference model, its basic functions were identified and specified such as: accept connection, receive http request message, analyze the received message, find file in the cache, create http response message, connect to web server, create http request message, send http request message, and save a copy of the file in the cache.

TABLE I CRITERIA AND COEFFICIENTS TO VALUE THE PROXY CACHE SERVER DEPLOYMENTS.

| **Criteria** | **Coefficient** |
|---|---|
| Cost | 2 |
| Performance | 1 |
| Documentation | 4 |
| Understandable | 3 |
| Open Source | 5 |

TABLE II VALUATION OF PROXY CACHE SERVERS IMPLEMENTATIONS, AS COMPARED TO EVALUATION CRITERIA

| **Evaluation criteria**<br>**Implementations** | **Cost** | **Performance** | **Documentation** | **Understandable** | **Open Source** | **TOTAL** |
|---|---|---|---|---|---|---|
| Squid | 5 | 5 | 5 | 3 | 5 | **75** |
| Microsoft ISA Server | 1 | 5 | 5 | 5 | 1 | **57** |
| Apache web server | 5 | 4 | 3 | 2 | 4 | **56** |
| Sun Java System Web Proxy | 4 | 4 | 2 | 2 | 5 | **55** |
| Proxy Firewall | 5 | 4 | 2 | 1 | 1 | **32** |

In the third stage, the agile software development methodologies including, Scrum [24], XP [23] and Kanban [25], were considered and identified. Scrum is an improvement of the object oriented development cycle , iterative and incremental that combines knowledge, tools and techniques, alongside the good performance of the development team; comprises planning, architecture and sprints; in planning, the list of products, delivery date, functionality, selection of releases, mapping objects, development team, risk assessment, inspection, adjustment, validation of development tool, estimated release cost, training, release and approval of financial aspects are generated. In architecture, the allocation of products, identifying changes, domain analysis, identifying issues or concerns in development and review meetings are considered. The sprints are a set of activities carried out in a period between one and four weeks after which, the team meets and presents progress of work, make adjustments, integrate, document, prepare training material and close the project.

The XP methodology consists of planning, design, development and testing; in the planning phase, the user history, the delivery schedule, the estimated number of iterations, the frequency of meetings between the client and the development team, the roles and adjustment to the methodology are defined. During the design phase the system metaphor was established, the cards class responsibility collaboration (CRC) were developed, the minimum functions were defined and the source code is restructured. In the development phase, the availability of client is considered, tests were defined and coded, integration was performed and the coding standard is established. Finally, in the testing phase the unitary and acceptance tests were executed.

The Kanban methodology consists of a set of so-called Kanban cards representing tasks and contains information such as id, name, estimated time and name of the responsible, in order to control the development process; in addition is governed by six rules. The first specifies that clients defines the deliverables, the second defines that the development team generates only the request made, the third indicates that components are not created or modified without a kanban that determines it, the fourth holds each product must be accompanied by a kanban, the fifth indicates that incorrect or defective products are not to be released or integrated, finally the sixth rule defines that the number of Kanban is reduced as the project culminates.

To select the agile software development methodology, people-related processes and criteria were taken into account, for which coefficients are assigned as shown in the following TABLES III, IV, V and VI.

Considering that the XP methodology scored higher value on expert judgment, it was adopted for the construction of the application that implemented the minimum functions of the proxy server cache.

TABLE III SELECTION CRITERIA AND FACTORS RELATED TO THE PEOPLE TO ASSESS THE AGILE SOFTWARE DEVELOPMENT METHODOLOGIES.

| Criteria related to people | Coefficients |
|---|---|
| Allows a small team | 5 |
| Allows active and continuous communication between developers and customers | 4 |
| Involve the client as part of the development team | 3 |
| Facilitates working team with different skills | 1 |
| Permite retroalimentación | 1 |
| Allows feedback | 1 |
| It allows the development team to participate in making decisions about the product | 1 |
| Allows definition of roles to team working members | 1 |
| Provides customers with visible signs of progress | 4 |
| It offers to the directives visible signs of progress | 2 |
| Requires little sophistication for managers and developers | 5 |

Once development methodology is selected, the proxy cache server, based on the requirements defined by the customer is designed, where also building a web client and a web server is considered; for the web client the user story was defined as "building an application requesting a file to the web server, through the mediation of a proxy cache server and if a file is received, store it in the disk"; as for the web server, user story was defined as "building a web server to send resources when requested and if not available, send an error message"; Finally in relation to the proxy cache server, the user story was defined as "building a caching proxy server that receives requests from the client, and if its available in the cache it will forward it, otherwise it will request it from to the web server and store it in cache".

TABLE IV SELECTION CRITERIA AND FACTORS RELATED TO THE PROCESS TO ASSESS THE AGILE SOFTWARE DEVELOPMENT METHODOLOGIES.

| Criteria related to processes | Coefficients |
|---|---|
| Generates simplicity in development | 5 |
| Analysis process required | 5 |
| Requires design | 5 |
| Requires planning | 4 |
| Requires construction | 5 |
| Requires testing | 4 |
| Facilitates Agility | 3 |
| Generates a documented development process | 5 |
| Allows increases in the function list | 3 |
| Generates an understandable process | 5 |
| Requires an object-oriented approach | 1 |
| Requires iterative and incremental processes | 1 |
| Generates a simple code | 5 |
| Requires unit tests | 5 |
| Requires the development of unit tests | 5 |
| Allows continuous integration | 2 |
| Allows changes once the process started | 2 |
| Work with little identification of the requirements | 1 |
| Work with little understanding of the architecture | 1 |
| Generates a highly reliable system | 1 |
| Creates risks | 1 |
| Requires little execution time | 3 |

TABLE V EVALUATION OF AGILE SOFTWARE DEVELOPMENT METHODOLOGIES AS TO THE CRITERIA RELATING TO PERSONS.

| Agile methodologies | People-oriented assessment |
|---|---|
| Scrum | 118 |
| XP | 108 |
| Kanban | 81 |

TABLE VI EVALUATION OF AGILE SOFTWARE DEVELOPMENT METHODOLOGIES REGARDING THE CRITERIA RELATED PROCESSES.

| Agile methodologies | Process-oriented assessment |
|---|---|
| Scrum | 251 |
| XP | 316 |
| Kanban | 219 |

In the context of agile methodology, test cases to evaluate the performance of the prototype, were specified.

In the fourth stage, the prototype of proxy server cache according to the proposed design is built, where it was considered that the implementation of the prototype will be accompanied by a web server and a web client, which implies the emergence of three independent systems, which articulates at runtime with the communication of process and variables. Thus, a client who has contact with the user, a web server responsible for providing the resources requested by the client; and the proxy cache server as a mediator between the client and the web server in the file sharing process are built.

During the fifth stage, the elements are designed to validate the effectiveness of the prototype cache proxy server, through the application of test cases. Were performed other functional tests consisted in observing records reported by the log files as a result of intervention between the client system, web server and proxy server cache, assuming no interaction between the user and the web server or the proxy cache, since the functionality in these systems is achieved through the mediation of the client system.

For the second iteration, was taken to consideration the case in which the cached file might be out of date compared to the same resource stored on the web server, which is why the conditional GET method that implemented the http request is considered, where the log record indicated the lack of validity of the cached resource from the analysis of the header and the new request of file to the web server for update.

The third iteration, specify new policies to manage the operation of the cache, considering the saturation or overflowing of himself, at which the LRU, LFU and FIFO policies that eliminate cached content in a coherent manner were proposed.

## IV. CONCLUSIONS AND FUTURE WORK

It has been presented the design and construction of a proxy cache server, with minimum functions necessary to illustrate the concept, in which quality attributes and good practices of Software Engineering were considered, generating an extensible and capable of being improved prototype, also the concepts associated with the topic were documented.

In the design and construction of the proxy cache server, analysis and documentation of the structure of the technologies involved in the proxy cache servers were made, in which was assumed Squid tool as the reference model after a selection process performed, which also internal architecture was analyzed by identifying its basic functions implementing them in the construction of a functional prototype for educational purposes, condition to be validated through documentation, applying quality attributes during software development and execution of tests.

The developed prototype is an extensible tool that can be applied as classroom work for those who are being trained as IT professionals; also good software engineering practices applied in the workplace, facilitates the extensibility of the product, as well as in-depth study of the performance of cache proxy server, given the detailed behavioral analysis of the minimum functions.

In this study good practices were used in the research methodology, along with best practices for project management, raised by the PMI for observation and control over the execution of work, also was considered DAR methodology for decision making both in the selection of Squid as a reference model, and the selection of XP as a agile software development methodology.

For validation purposes, were built a client system and a web server to accompany the proxy cache server execution using the Java language as development tool.

As further work, it is considered the construction of a framework for cache in order to decouple this functionality. On the other hand, one can contemplate the possibility of distributing the cache as a strategy to improve performanceadapting it to a P2P DHT infrastructure to enable web browsers on desktop machines to share their local cache.

## V. REFERENCES

[1] Greg Barish, Katia Obraczka. “World Wide Web Caching: Trends and Tecniques”. IEEE Communications Magazine, May 2000, pags.178-185.

[2] Carlos E. Gómez, Luis E. Sepúlveda, Christian A. Candela. “Servidor Proxy Caché: Comprensión y asimilación tecnológica”. Revista INGE CUC, Octubre de 2012, Vol 8 No.1,pags.149-162.

[3] “A Guide to the PROJECT MANAGEMENT BODY OF KNOWLEDGE (PMBOK GUIDE)”. Forth Edition, Project Management Institute, 2008.

[4] “Decision analysis and resolution process”, The process group. Newsletter Post. Vol.12 No.2.October 2005.

[5] Brian. D. Davison. “A Web Caching Primer”, IEEE Internet Computing, Vol.5,No.4, July/August 2001,pags.38-45.

[6] Ramón Cáceres, Fred Douglis, Anja Feldmann, Gideon Glass, Michael Rabinovich. “Web Proxy Caching: The Devil is in the Details”. AT&T Labs.1998.

[7] JiaWang. “A survey of web caching schemes for the Internet”. ACM SIGCOMM Computer Communication Review. Volume 29 Issue 5, October 1999. Pags 36-46.

[8] Carlos E. Gómez, Luis E. Sepúlveda, Christian A. Candela. “Análisis de configuraciones de servidores proxy caché – Informe final proyecto de investigación”. Facultad de Ingeniería, Centro de Investigaciones de la Facultad de Ingeniería. Universidad del Quindío 2010.

[9] Delegate Home Page. [Online] Disponible en http://www.delegate.org/delegate/

[10] [Online] Disponible en http://proxy-firewall.en.lo4d.com

[11] [Online] Disponible en http://proxyfirewall.software.informer.com

[12] Oracle (2014). Sun Java System Web Proxy Server [Online]. Disponible en http://docs.oracle.com/cd/E19528-01/820-1835/gdgk/index.html.

[13] Apache. The Apache Software Foundation. [Online] Disponible en http://www.apache.org/2014

[14] www.nsit.com.co/index.php/pilares/seguridad-2/utm-2/fortinet

[15] Microsoft. Introducción a ISA Server. [Online] Disponible en http://www.microsoft.com/spain/isaserver/2004/info/overview.aspx

[16] “http: The Definitive Guide”. D.Wessels. O´Reilly&Associates,2004.

[17] Rodriguez, M. (2014) SQUID: El proxy-cache GNU [Online] Disponible en http://www.marisabel.com.ar/pdfs/squid proxy-server_next Edicion 15.pdf

[18] “http The Definitive Guide”. David Gourley, Brian Totty, Marjorie Sayer, Sailu Reddy y Anshu Aggarwal. O´Reilly & Associates 2002.

[19] “Web Caching”. Duane Wessels. O´Reilly. First Edition June 2001.

[20] Ari Luotonen, Kevin Altis. “World-Wide web proxies”. CERN. Intel. Abril 1994.

[21] Kent Beck. “Embracing Change with Extreme Programming”. First class software, IEEE 1999.

[22] Jeff Sutherland, Kent Shwaber. “The Scrum Papers: Nut, Bolt, and Origins of an Agile Framework”. Scrum,Inc. 29 Jan 2011, Paris.

[23] Kenji Hiranabe “Kanban Applied to Software Development: from Agile to Lean”. InfoQ, 2008.

[24] [Online] Disponible en http://www.squid-cache.org/

[25] “Metodología de la Investigación Tecnológica”. S. Ríos. Ediciones Hispanoamericanas Ltda. 1998.

**Jose A. Uruburo,** He is a student of MA in Computer and Systems Engineering from the Technological University of Pereira. His areas of interest are data networks, IT infrastructure and P2P DHT systems.
**Email**: jose.uruburo@utp.edu.co

**Carlos E. Gomez,** is a professor and researcher in the Program Systems Engineering and Computer Science, at the "Universidad del Quindío". He has studied MA in Systems Engineering and Computer Science at the "Universidad de los Andes". His areas of interest are data networks, IT infrastructure, P2P systems and cloud computing DHT.
**Email:** carloseg@grid.edu.co

**Christian A. Candela,** is a professor and researcher in the Program Systems Engineering and Computer Science, at the "Universidad del Quindío". He has done Masters Degree in Electronic Commerce at the Polytechnic Institute of Monterrey Mexico. His areas of interest are software architecture and infrastructure.
**Email:** christiancandela@grid.edu.co

**Luis E. Sepulveda, ,** is a professor and researcher in the Program Systems Engineering and Computer Science, at the "Universidad del Quindío". He has done Masters Degree in Free Software at the "Universidad Autónoma de Bucaramanga" in partnership with the Open University of Catalunya-Spain. His areas of interest include IT infrastructure, Data Networks, Project Management, Virtualization and High Performance Computing.
**Email**: lesepulveda@grid.edu.co