Reliable Partial Replication of Contents in Web Clusters: Getting Storage without losing Reliability

Jose Daniel Garcia, Jesus Carretero, Felix Garcia,
Javier Fernandez, David E. Singh and Alejandro Calderon
Department of Computer Science, University Carlos III of Madrid, Colmenarejo, Spain
Email: {josedaniel.garcia,jesus.carretero,felix.garcia,javier.fernandez,david.exposito,alejandro.calderon}@uc3m.es

***Abstract*— Traditionally, distributed Web servers have used two strategies for allocating files on server nodes: full replication and full distribution. While full replication provides a highly reliable solution, it limits storage capacity to the capacity of the smallest node. On the other hand, full distribution provides higher storage capacity at the cost of lower reliability. A hybrid solution is partial replication where every file is allocated to a small number of nodes. The most promising architecture for a partial replication strategy is the Web cluster architecture. However, Web clusters present a big flaw from reliability perspective as they contain a single point of failure. To correct this flaw, in this paper we present a modified architecture: the Web cluster with distributed Web switch. Reliability of Web clusters is evaluated for different replication strategies. System evaluations show that our proposal leads to a highly reliable solution with high scalability.**

***Index Terms*— Web cluster, Web content replication, reliable Web servers, Web content allocation, storage systems.**

I. Introduction

Due to increasing development of Web based applications a huge interest has arised in building hardware and software for this kind of systems in such a way that solutions are scalable both in performance and storage capacity. Traditional solutions to this problem include software and hardware scale up [2]–[4]. A recent architectural alternative is the distributed Web server [5] where a set of server nodes are used to host a Web site. In these systems, performance scalability is achieved by adding new server nodes. In a distributed Web server, received requests are dispatched among the server nodes.

Usually, solutions are based either on full content replication (every file is replicated in every server node) or full distribution (every file is stored in one and only one server node). With full replication the system is highly reliable and request distribution is easy to implement, as each request may be served by any server node. On the other hand, storage scalability is minimal, as the full storage capacity is limited by the server node with the lowest capacity. Furthermore, adding new server nodes to the system does not increase storage capacity. On the other hand, with full distribution the system gives a lower reliability (a node failure makes some content unavailable) and request distribution needs to use some sort of directory service [6] to determine the node storing a file. However, storage scalability is maximum, as adding a new node means increasing the total amount of available storage.

A third alternative is partial replication, where each file is replicated in a possibly different subset of the set of server nodes. With partial replication reliability is higher than in the case of full distribution and lower than in the case of full replication. In terms of storage capacity, partially replicated solutions provide less capacity than fully distributed solutions but more than fully replicated solutions. However, an important difference with full replication is that partial replication provides storage capacity scalability (i.e. adding new server nodes increases the system storage capacity).

This paper studies the reliability of partially replicated distributed Web servers and compares the offered reliability with the reliability of classical fully replicated and fully distributed solutions. The paper is organized as follows: section II discusses related work; section III presents the architectural alternatives for a Web cluster with partial replication of contents; section IV studies the reliability of a Web cluster when contents are partially replicated; section V shows evaluation; in section VI we summarize our conclussions; finally, in VII we outline future work.

II. Related Work

Several distributed Web server architectures have been proposed. Those architectures may be classified in three major families:

1) **Cluster based Web systems** Cluster nodes own IP addresses are not visible from the clients. Instead, clients use an virtual IP address which corresponds to that of some request distribution device (Web switch). The Web switch [7] receives requests from clients and sends each request to some server node.

This paper is based on [1] "On the Reliability of Web Clusters with Partial Replication of Contents," by J.D. Garcia, J. Carretero, J. Fernandez, F. Garcia, D.E. Singh and A. Calderon, which appeared in the Proceedings of the First International Conference on Availability, Reliability and Security, ARES 2006 , Vienna, Austria, April, 2006. © 2006 IEEE.

This work was supported in part by Spanish Ministry of Science and Education under the TIN2004-02156 contract and Madrid Regional Government under contract UC3M-INF-05-003.

2) **Virtual Web clusters** All the server nodes share a single common public IP address [8]. Each node receives all messages in a way that every request is discarded by all nodes except one.
3) **Distributed Web systems** Each node has its own different publicly visible IP address [9], [10]. Request distribution is made by a combination of dynamic DNS [11] and request redirection.

The idea of partial replication of files was previously used in distributed file systems [12], where placement algorithms were proposed. The usage of partial replication of Web contents has been proposed by several authors, studying architectural issues [13] or placement algorithms [14], [15]. The effect of partial replication on the scalability of storage capacity has been shown in [16]. However, the effect of partial replication on reliability needs to be addressed.

Service reliability and availability of Web clusters has been studied [17], although such studies assume full replication of contents and simple dispatching policies as round-robin policies. In this paper we present a study of reliability where contents are partially replicated.

III. ARCHITECTURE

Several request distribution policies have been used in the last years to decide which server node must be in charge of a request [5]. When contents of a Web server are partially replicated in a set of server nodes, standard request distribution algorithms cannot be directly used. This is because the first needed step is the determination of the subset of server nodes where the requested file is effectively stored.

In a cluster based Web system (or Web cluster), the request distribution algorithm is implemented in the Web switch, where every request is analyzed and assigned to a server node. When contents are partially replicated, the Web switch needs to determine the subset of server nodes effectively containing the requested file and, after that, selecting a server node from that subset.

In a virtual Web cluster, request distribution is typically performed by computing a hash function on the client IP address (and perhaps the client port). That may be done assuming that every file is stored in every server node. When partial replication is used, a general hash function cannot be used as it could yield the selection of a node not containing the requested file.

In a distributed Web system any request may reach to any server node (due to the usage of dynamic DNS) which may redirect the request to another node if the requested file is not present or the node's load is too high. When a partial replication is used, the number of redirections may increase dramatically if replication degree is low because there is a high probability of sending a request to a node which does not contain the requested file.

Although distributed Web systems are not well suited for partial replication of contents, some of the ideas behind their design may be used to improve Web clusters. We use those ideas to propose a new Web server distributed architecture: the Web cluster with distributed switch (see Fig. 1).

A Web cluster with distributed switch [13] uses several switches to increase reliability and to eliminate the single point of failure of a standard Web cluster. Request distribution among switches is performed by means of dynamic DNS [11] and by allowing that a highly loaded Web switch redirects a request to another switch. This request redirection may borrow some routing mechanism for distributed Web systems, as triangulation [9] or HTTP redirect [18]. Request redirection among Web switches is only performed when a Web switch is highly loaded and there is another switch with less load. If Web switches are not highly loaded there is no performance penalty over a standard Web cluster architecture. On the other hand, if a Web switch is highly loaded the redirection cost will be lower than the cost associated to keep the request waiting in the queue of the highly loaded Web switch.

IV. RELIABILITY OF WEB CLUSTERS

In a Web cluster, the system may be viewed as the composition of two series system: the Web switch and the set of server nodes. Thus, system reliability may be expressed as the product of Web switch reliability R_s and server nodes reliability R_n as expressed in (1).

$$R_{sys} = R_s \cdot R_n \tag{1}$$

The reliability of server nodes set depends not only on the system topology, but also on replication strategy. Three alternatives must be considered: full replication, full distribution and partial replication, giving different reliability expressions.

The reliability of a set of server nodes may be expressed in terms of the probability of failure upon a request F_n as expressed in (2).

$$R_n = 1 - F_n \tag{2}$$

If every file is replicated r times in a Web cluster with M server nodes, a failure happens when the r server nodes where the requested file resides fail simultaneously.

We make the following assumptions to determine reliability:

1) Replica distribution among server nodes is uniform.
2) Individual reliability of server nodes and the Web switch is identical.

For full distribution and full replication, uniform distribution is always true. For partial replication, uniform distribution must be a goal of the replica allocation algorithm to perform load balancing. Second assumption is usually satisfied when the Web cluster is implemented using standard workstations (making it cheaper).

Table I shows system reliabilities for different replication strategies as shown in appendix (see corollaries) as an application of theorem 1.

In a Web cluster with distributed switch there are several Web switches acting as front-end. When a request

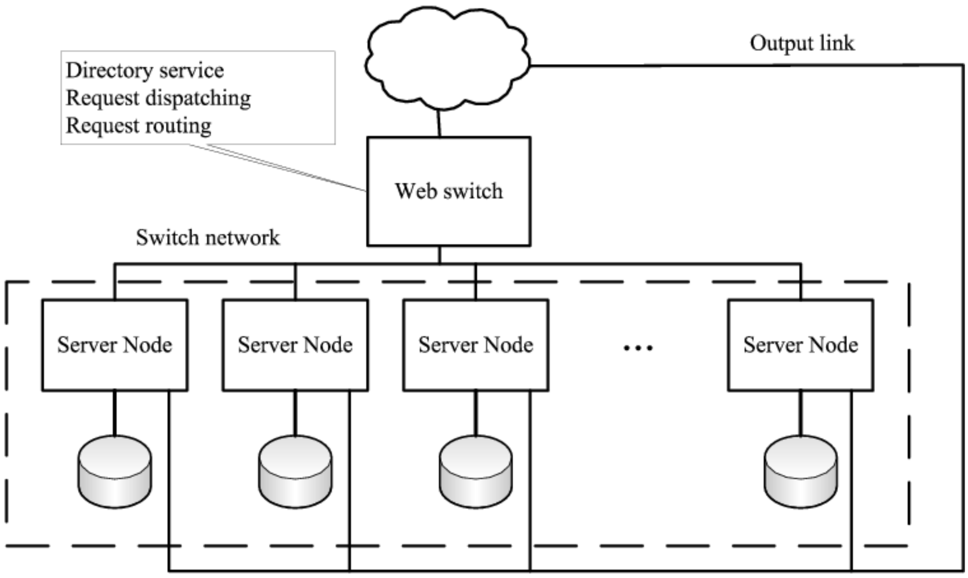

Figure 1. Architecture for a Web cluster with distributed switch.

TABLE I.
WEB CLUSTER SYSTEM RELIABILITIES

Strategy	System Reliability
Full replication	$R(1-(1-R)^M)$
Full distribution	R^2
Partial replication	$R(1-(1-R)^r)$

arrives to the system, it is processed by a Web switch, which transfers the request to a server node. The Web switch subsystem may be viewed as a parallel system and its reliability may be computed by using the general parallel system reliability equation. Thus the system reliability R_{sys} is given by the reliability of the Web switch subsystem (where each of the P switches has an individual reliability R_{s_i}) and the server nodes subsystem R_n) as expressed in (3)

$$R_{sys} = \left(1 - \prod_{i=1}^{P} (1 - R_{s_i})\right) R_n \qquad (3)$$

V. EVALUATION

For evaluation purposes we have used a cluster with 32 nodes where individual reliabilities of nodes and switches have been set to different levels (ranging from 70% to 99%). First, evaluations for a Web cluster are presented. Then, evaluations for a Web cluster with distributed Web switch are presented. Finally, we present an analysis of the reliability improvement obtained when new switches are added to a cluster.

A. Web cluster reliability

Maximum reliability for a Web cluster is reached when full replication is used. If the number of server nodes is high enough, reliability for the full system is given by the reliability of the Web switch.

$$\lim_{M\to\infty} R_{sys} = \lim_{M\to\infty} R\left(1-(1-R)^M\right) = R \qquad (4)$$

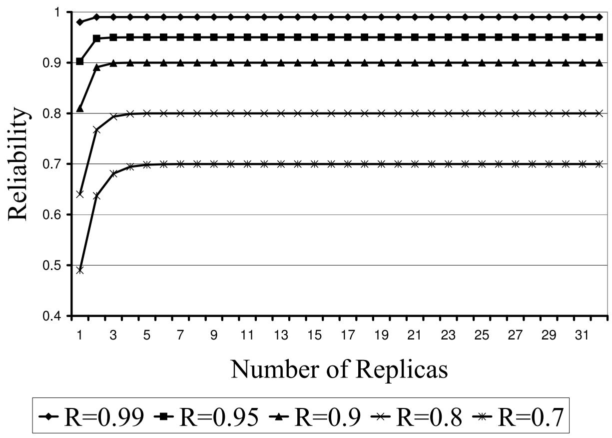

Figure 2. Reliability for a Web cluster with partial replication.

Minimum reliability for a Web cluster is reached when full distribution is used ($R_{sys} = R^2$). For partial replication, system reliability goes from R to R^2. To evaluate the evolution of system reliability for different replication levels, reliability has been computed (see Fig. 2) for different reliability levels (ranging from 0.7 to 0.99) from 1 to 32 replicas per file.

For each reliability level, maximum reliability is reached with few replicas. The number of replicas to achieve reliability similar to that of a fully replicated system is low compared with the number of server nodes in the cluster. This number of replicas depends on the reliability level and the required precision. When the target reliability is tR (where t is the fraction of reliability R to achieve and it is very close to 1), the following restriction must be satisfied:

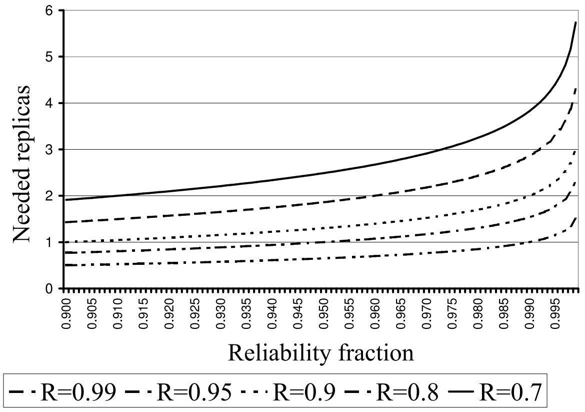

Figure 3. Number of replicas needed for different levels of reliability.

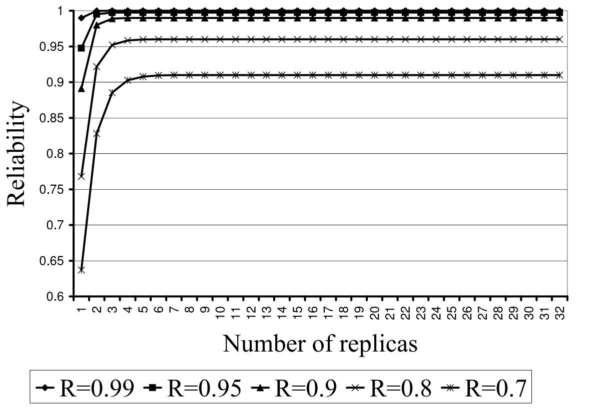

Figure 4. Reliability for a Web cluster with 2 switches and partial replication.

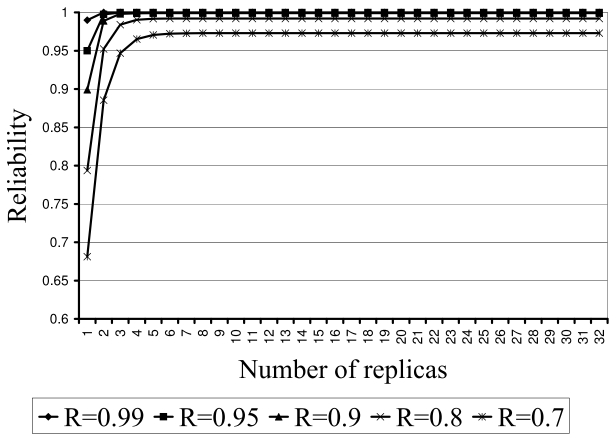

Figure 5. Reliability for a Web cluster with 3 switches and partial replication.

$$R_{sys} > tR \tag{5}$$

and consequently

$$\begin{aligned} &R\left(1-(1-R)^r\right) > tR \\ &(1-R)^r < 1-t \\ &r\log(1-R) < \log(1-t) \\ &r < \frac{\log(1-t)}{\log(1-R)} \end{aligned} \tag{6}$$

Fig. 3 represents the number of replicas per file needed to get a fraction t of maximum reliability R. To get 99% of the reliability of a full replicated Web cluster it is enough with full distribution if components have an individual reliability of 99%. If components are 90% reliable then 2 replicas per file are needed. If components are 70% reliable, then 4 replicas per file are needed.

An important general consequence is that, for a Web cluster, partial replication with a relatively low number of replicas per file, gives the same reliability than full replication. However, partial replication allows to store high volume Web sites with less resources, as it allows size scalability.

B. Web cluster with distributed switch reliability

In a Web cluster with multiple Web switches, maximum reliability is achieved when full replication is used. With a high number of server nodes, cluster reliability is determined by the reliability of the Web switch subsystem.

$$\begin{aligned} \lim_{M\to\infty} R_{sys} &= \lim_{M\to\infty}\left(1-(1-R)^P\right)\left(1-(1-R)^M\right) \\ &= \left(1-(1-R)^P\right) \end{aligned} \tag{7}$$

Minimum reliability for a Web cluster with multiple Web switches is reached when full distribution is used ($R_{sys} = \left(1-(1-R)^P\right)R$). For partial replication intermediate values, from $\left(1-(1-R)^P\right)R$ to $1-(1-R)^P$, are obtained. To evaluate the evolution of system reliability for different number of replicas per file, reliability has been computed from 1 to 32 replicas per file for different reliability levels. Fig. 4 shows results for a Web cluster with 2 switches, Fig. 5 shows results for a Web cluster with 3 switches, and Fig. 6 shows results for a Web cluster with 4 switches.

Evaluations performed on each configuration show that a system with a relatively small number of replicas (from 3 to 5 replicas) offers a reliability equivalent to the reliability of a fully replicated system. However, when the number of switches increases, dependence on individual reliability is less important as system reliability approaches to one.

To compare reliability of a Web cluster configuration when the number of switches increases, individual reliability has been fixed to 0.9 and system reliability has been evaluated for configurations from 1 to 4 switches (see Fig. 7).

For a fixed individual reliability level maximum system reliability is always reached with the same number of replicas per file. However, the incorporation of new Web switches increases reliability. In the example configuration evaluated in Fig. 7, usage of three web switches increase

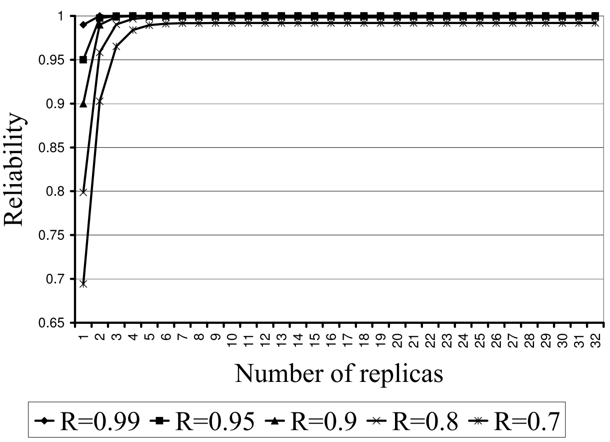

Figure 6. Reliability for a Web cluster with 4 switches and partial replication.

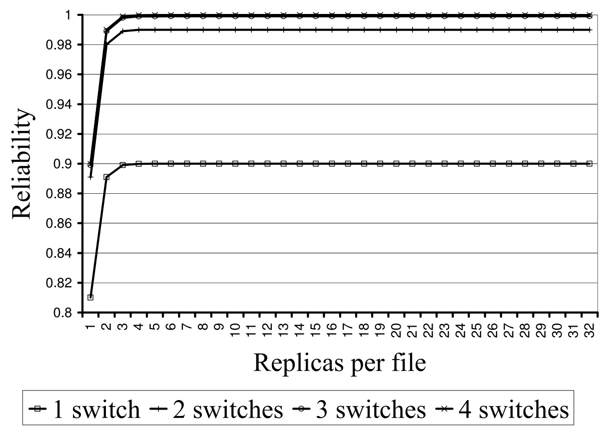

Figure 7. Reliability for clusters with different number of switches.

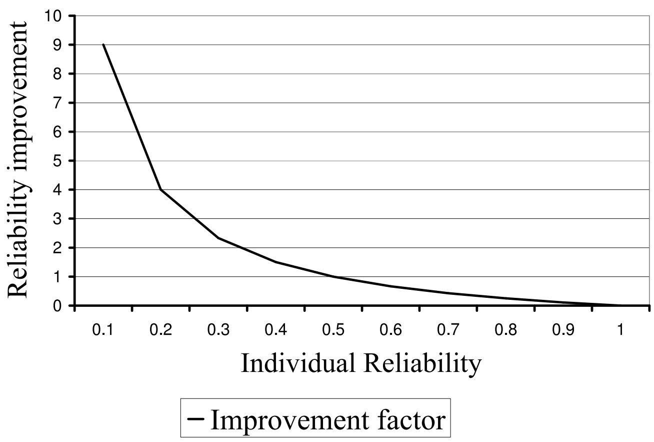

Figure 8. Maximum reliability improvement to be obtained by adding new Web switches.

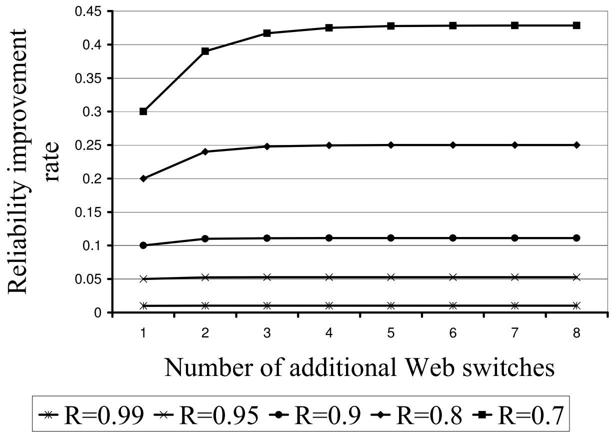

Figure 9. Reliability improvement obtained by adding new Web switches.

reliability in such a way that adding more Web switches will not have any non marginal effect. Thus, we conclude that, for this configuration, optimal solution is based in a set of 2-3 Web switches (single point of failure elimination is achieved).

C. Reliability improvement adding Web switches

An important issue is the reliability improvement obtained when a new Web switch is added to a Web cluster. When ΔP new switches are added to a Web cluster having one Web switch, the system changes its reliability R^1_{sys} to $R^{1+\Delta P}_{sys}$. Reliability improvement $I_{\Delta P}$ may be measured as the rate of reliability increment over the original single switch based solution.

$$I_p = \frac{R^{1+\Delta P}_{sys} - R^1_{sys}}{R^1_{sys}} = \frac{\left(1-(1-R)^{1+\Delta P}\right) - R}{R} = \frac{1-R}{R}\left(1-(1-R)^{\Delta P}\right) \quad (8)$$

The maximum reliability improvement that may be obtained by adding Web switches is only determined by the individual Web switches reliabilities.

$$\lim_{\Delta P \to \infty} I_p = \frac{1-R}{R} \quad (9)$$

Fig. 8 shows maximum reliability improvement which may be obtained by adding new Web switches to a Web cluster.

To evaluate the effect of adding new Web switches to a Web cluster with one Web switch, we have plotted (see Fig. 9) the reliability improvement rate for different reliability levels. When individual reliability is very high (e.g. 0.99), system reliability improvement is very low and independent of the number of Web switches. For lower individual reliabilities (e.g. 0.8 or 0.7), system reliability improvement increases until the third or fourth switch is added. After that, improvement stabilizes.

From reliability point of view, an improvement may be obtained by incorporating additional Web switches to a Web cluster. However, no significant improvement is obtained when a system has more of 4 or 5 Web switches.

VI. CONCLUSION

Web content partial replication emerges as an hybrid alternative between full replication and full distribution when using a distributed Web server. Among the existing architectures for distributed Web servers, Web clusters present less disadvantages for partially replicated contents. However, a Web cluster has a major reliability flaw as it contains a single point of failure: the Web switch. To solve this reliability flaw, a modified architecture is presented in this paper (the Web cluster with distributed Web switch) which improves the global reliability of the system.

In a standard Web cluster the usage of a small number of replicas per file gives the same reliability of a fully replicated Web cluster. Typically, the number of replicas per file to get an equivalent reliability to a fully replicated system is about 3 or 4 replicas per file. It must not be forgotten that a Web cluster with partial replication and a low number of replicas per file offers higher storage capacity than a fully replicated Web cluster.

In a Web cluster with distributed Web switch it is also true that the usage of a small number of replicas per file gives the same reliability of a fully replicated Web cluster. Also in this case, the number of replicas per file to get an equivalent reliability level to a fully replicated system is about 3 or 4 replicas per file. An additional advantage of this architecture is that maximum reliability (which is achieved with few replicas per file) increases as the number of Web switches is increased. Reliability increase reaches its maximum for a cluster of 4 switches (i.e. adding more switches does not produce a significant increase of reliability).

Evaluations show that, from a reliability perspective, an effective architecture is a Web cluster with distributed Web switch in which 2 or three Web switches are used and contents are partially replicated using 4 replicas per file. Configurations with less switches and less replicas per file may be also acceptable when the target reliability is lower.

VII. Future Work

This paper has examined reliability offered by partially replicated Web clusters with the same number of replicas for each file. However another possibility is the unbalanced replication where not all files have the same number of replicas. Work is going on evaluating reliability of unbalanced partially replicated Web clusters.

Appendix

Theorem 1: Reliability of a single switched Web cluster with partial replication, where individual nodes failures are independent and replica distribution is uniform, does not depend on number of server nodes and it is only determined by individual reliabilities of nodes, number or replicas per file and reliability of the Web switch as expressed in (10).

$$R_{sys} = R_s \left(1 - \prod_{j=i_1}^{i_r} F_i\right) \tag{10}$$

Proof: Let $P_{i_1}^{i_r}$ be the probability that the r replicas of the requested file are stored in server set $\{S_{i_1}, S_{i_2}, \ldots, S_{i_r}\}$. For each possible assignment of the replicas of a file to a set of server nodes, its contribution to the failure probability is given by the product of probability of the assignment $P_{i_1}^{i_r}$ and the probability of the individual failures F_j for each server s_j as expressed in (11).

$$F_n = \sum_{\substack{i_1,i_2,\ldots,i_r=1 \\ i_1<i_2<\ldots<i_r}}^{M} P_{i_1}^{i_r} \prod_{j=i_1}^{i_r} F_j \tag{11}$$

As replica distribution is uniform, probability $P_{i_1}^{i_r}$ that the r replicas of the requested file are stored in server set $\{S_{i_1}, S_{i_2}, \ldots, S_{i_r}\}$ is given by (12).

$$P_{i_1}^{i_r} = \frac{1}{C_{M,r}} = \frac{1}{\binom{M}{r}} \tag{12}$$

Using (12) in (11) allows to express probability of failure F_n exclusively in terms of individual nodes failure probability, as expressed in (13).

$$\begin{aligned} F_n &= \sum_{\substack{i_1,i_2,\ldots,i_r=1 \\ i_1<i_2<\ldots<i_r}}^{M} \frac{1}{\binom{M}{r}} \prod_{j=i_1}^{i_r} F_j \\ &= \frac{1}{\binom{M}{r}} \sum_{\substack{i_1,i_2,\ldots,i_r=1 \\ i_1<i_2<\ldots<i_r}}^{M} \prod_{j=i_1}^{i_r} F_j = \frac{1}{\binom{M}{r}} \binom{M}{r} \prod_{j=i_1}^{i_r} F_j \\ &= \prod_{j=i_1}^{i_r} F_j \end{aligned} \tag{13}$$

■

Corollary 1: A partially replicated Web cluster where all server nodes and the Web switch are equally reliable, offers a system reliability given by (14).

$$R_{sys} = R\left(1 - (1-R)^r\right) \tag{14}$$

Proof: The fact that all server nodes and the Web switch are equally reliable is expressed by (15) and (16).

$$F_j = 1 - R \forall j \in [1, M] \tag{15}$$

$$R_s = R \tag{16}$$

As system reliability is expressed in terms of Web switch reliability and server nodes reliabilities ($R_{sys} = R_s(1 - F_n)$), its expression may be derived by substitution of ((13)), (15) and (16) as it is shown in (17).

$$R_{sys} = R\left(1 - \prod_{j=i_1}^{i_r} F_j\right) = R\left(1 - (1-R)^r\right) \tag{17}$$

■

Corollary 2: Reliability of a fully replicated Web cluster where all server nodes and the Web switch are equally reliable is given by (18).

$$R_{sys} = R\left(1 - (1-R)^M\right) \tag{18}$$

Proof: In a fully replicated Web cluster, the number of replicas per file is M, as every file is replicated in every server node. Substitution on (14) leads to (18). ■

Corollary 3: Reliability of a fully distributed Web cluster where all server nodes and he Web switch are equally reliable is given by (19).

$$R_{sys} = R^2 \tag{19}$$

Proof: In a fully distributed Web cluster, the number of replicas per file is 1, as every file is allocated in a single node. Substitution on (14)leads to (20).

$$R_{sys} = R\left(1 - (1 - R)\right) = R^2 \tag{20}$$

■

REFERENCES

[1] J. D. Garcia, J. Carretero, F. Garcia, A. Calderon, J. Fernandez, and D. E. Singh, "On the reliability of web clusters with partial replication of contents," in *First International Conference on Availability, Reliability and Security, 2006. ARES 2006.* IEEE, April 2006, pp. 617– 624.

[2] V. S. Pai, P. Druschel, and W. Zwaenepoel, "Flash: An efficient and portable Web server," in *Proceedings of the USENIX 1999 Annual Technical Conference*. Monterey, CA, USA: USENIX Association, June 1999, pp. 199–212.

[3] ——, "IO-Lite: A unified I/O buffering and caching system," *ACM Transactions on Computer Systems*, vol. 18, no. 1, pp. 37–66, Feb. 2000.

[4] G. Banga, P. Druschel, and J. C. Mogul, "Resource containers: A new facility for resource management in server systems," *Operating Systems Review*, vol. Special Issue, Winter 1998, pp. 45–58, 1999. [Online]. Available: http://www.cs.rice.edu/~druschel/osdi99rc.ps.gz

[5] V. Cardellini, E. Casalicchio, M. Colajanni, and P. S. Yu, "The state of the art in locally distributed Web-server systems," *ACM Computing Surveys*, vol. 34, no. 2, pp. 263–311, June 2002.

[6] G. Apostolopoulos, D. Aubespin, V. Peris, P. Pradham, and D. Saha, "Design, implementation and performance of a content-based switch," in *Proceedings of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000)*, vol. 3. Tel Aviv, Israel: IEEE, Apr. 2000, pp. 1117–1126. [Online]. Available: http://www.ecs.umass.edu/ece/wolf/courses/ECE697J/papers/web_switch.PDF

[7] T. Schroeder, S. Goddard, and B. Ramamurthy, "Scalable web server clustering technologies," *IEEE Network*, vol. 14, no. 3, pp. 38–45, May 2000. [Online]. Available: http://www.comsoc.org/ni/public/2000/may/index.html

[8] S. Vaidya and K. J. Christensen, "A single system image server cluster using duplicated MAC and IP addresses," in *Proceedings of the 26th Annual IEEE Conference on Local Computer Networks (LCN 2001)*. Tampa, FL, USA: IEEE, Nov. 2001, pp. 206–214.

[9] L. Aversa and A. Bestavros, "Load balancing a cluster of web servers: using distributed packet rewriting," in *Conference Proceedings of the 2000 IEEE International Performance, Computing, and Communications Conference (IPCCC 2000)*. Phoenix, AZ, USA: IEEE, Feb. 2000, pp. 24–29.

[10] V. Cardellini, "Request redirection algorithms for distributed web systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 14, no. 4, pp. 355–368, Apr. 2003.

[11] T. Brisco, *DNS Support for Load Balancing. RFC 1794*, Internet Engineering Task Force, Apr. 1995. [Online]. Available: ftp://ftp.is.co.za/rfc/rfc1794.txt

[12] L. W. Dowdy and D. V. Foster, "Comparative models of the file assignment problem," *ACM Computing Surveys*, vol. 14, no. 2, pp. 287–313, June 1982.

[13] J. D. Garcia, J. Carretero, J. M. Prez, F. Garcia, and J. Fernandez, "A distributed web switch for partially replicated contents," in *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics (SCI 2003)*, vol. VIII, Orlando, FL, USA, July 2003, pp. 1–6.

[14] L. Zhuo, C.-L. Wang, and F. C. M. Lau, "Document replication and distribution in extensible geographically distributed web servers," *Journal of Parallel and Distributed Computing*, vol. 63, no. 10, pp. 927–944, Oct. 2003.

[15] S. S. H. Tse, "Approximate algorithms for document placement in distributed web servers," *Transactions on Parallel and Distributed Systems*, vol. 16, no. 6, pp. 489–496, June 2005.

[16] J. D. Garcia, J. Carretero, F. Garcia, J. Fernandez, A. Calderon, and D. E. Singh., "A quantitative justification to partial replication of web contents," in *International Conference on Computational Science and its Applications*, ser. Lecture Notes in Computer Science, vol. 3983. Springer Verlag, May 2006, pp. 1136–1145.

[17] M. Martinello, M. Kaaniche, and K. Kanoun, "Web service availability–impact of error recovery and traffic model," *Reliability Engineering and System Safety*, vol. 89, no. 1, pp. 6–16, July 2005.

[18] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee, *Hypertext Transfer Protocol - HTTP/1.1. RFC 2616*, Internet Engineering Task Force, June 1999. [Online]. Available: http://www.w3.org/Protocols/rfc2616/rfc2616.html

Jose Daniel Garcia received his Advanced Studies Diploma (MS) and PhD in Computer Science from University Carlos III of Madrid in 2003 and 2005 respectively and his degree in Computer Science from Madrid Technical University in 2001.

He is an Associate Professor at University Carlos III of Madrid. Previously, he was an Assistant Professor at the same University and worked as Systems Engineer in projects for several companies including DMR Consulting, Siemens, British Telecom and Telefonica. His research interests include distributed systems and high performance computing.

Prof. Garcia is member of the IEEE, IEEE Computer Society and the ACM.

Jesus Carretero received his degree and PhD in Computer Science from Madrid Technical University in 1989 and 1995 respectively.

He is a Full Professor at University Carlos III of Madrid. Previously he was an Associate Professor at University Carlos III of Madrid and Madrid Technical University. In 1997 and 1998 he was a visiting scholar at Northwestern Unversity in Chicago (USA). His research interest is focused on parallel and distributed systems, especially data storage systems, real-time systems, and multimedia techniques. He is the author of several educational books and he has published papers in several major journals of its area as, for example, Parallel Computing and the Journal of Parallel and Distributed Computing.

Prof. Carretero is member of the IEEE Computer Society and he is actually in charge of the Scalable Storage and File Systems of the Technical Committe on Scalable Computing.

Felix Garcia received his degree and PhD in Computer Science from Madrid Technical University in 1993 and 1996 respectively.

He is an Associate Professor at University Carlos III of Madrid. Previously, he was an Associate Professor ant Madrid

Technical University. His research interests include high performance computing and parallel file systems. He is coauthor of nine books and he has published 70 articles in journals and conferences.

Prof. Garcia is member of the ACM.

Javier Fernandez received his Advanced Studies Diploma (MS) and PhD in Computer Science from University Carlos III of Madrid in 2002 and 2004 respectively and his degree in Computer Science from Madrid Technical University in 1999.

He is an Associate Professor at University Carlos III of Madrid. Previously, he was an Assistant Professor at the same University. His research interests include high performance computing on cluster and grid, QoS and virtualization in storage systems, and parallel and distributed storage systems.

David E. Singh received the BS and MS degrees in Physics from the University of Santiago de Compostela (Spain) in 1997. He received the PhD degree in 2003 from the University of Santiago de Compostela.

He is currently an Associate Professor of Computer Architecture at University Carlos III of Madrid. His research interests include parallelizing compilers, code optimization for distributed memory multiprocessor and high performance I/O.

Alejandro Calderon received his Advanced Studies Diploma (MS) and PhD in Computer Science from University Carlos III of Madrid in 2003 and 2005 respectively and his degree in Computer Science from Madrid Technical University in 2000.

He is an Assistant Professor at University Carlos III of Madrid. His research interests include high performance computing on cluster and grid, fault tolerant storage systems, and virtualization in high performance computing.