RESEARCH ARTICLE OPEN ACCESS

Towards an Improved Understanding of Data Availability in P2P Storage Systems

Rama Alkhayer, Ahmad Saker Ahmad
Department of System and Computer Networks Engineering
Tishreen University, Latakia, Syria.

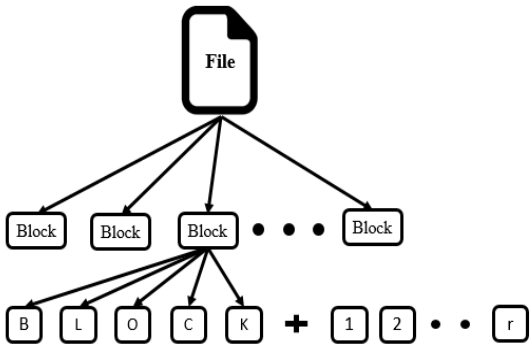
ABSTRACT
P2P storage systems provide a decent alternative to cloud storage. The core designing issue of these systems is data availability. Data availability is jeopardized in P2P systems due to the random and uncontrollable arrival and departure of nodes, aka churn. To improve availability, data is made redundant by distributing copies of each file or replicas of its pieces on multiple nodes on the network. This redundancy is achieved either by replication or by erasure coding. Previous work in this field addressed node and data availability as if they were equated. In this paper, we examine the difference between data availability and node availability and prove that node availability is an inaccurate indicator of data availability and the two are not equivalent. Furthermore, we show that excessive replication threatens system performance and leads to degraded quality of service. Finally, we compare the methods by which redundancy is achieved i.e. erasure coding and replication.
Keywords **:-** P2P, storage systems, data availability, replication, erasure coding.

I. INTRODUCTION

P2P systems utilize the peripheral idle resources on the edge of the Internet. These resources include, but are not limited to, storage space, idle CPU cycles and bandwidth. P2P storage systems focus on unexploited storage space on network hosts and attempt to exploit this empty space as best and as much as possible. While P2P storage systems surpass cloud storage in terms of scalability, privacy, anonymity and the lack of a single point of failure. However, one major shortcoming of these systems is churn; that is the random arrival and departure of peers. Churn threatens the availability and reliability of the service. In case of storage systems, the temporary departure of a peer renders the files hosted on this node temporarily unavailable during its offline time. To compensate for such events, data is made redundant. Redundancy is achieved by means of one of two methods: replication and erasure coding. With replication, the system breaks the file into fixed-size chunks. Every chunk is replicated and these replicas are stored on various nodes in a way that each of these nodes has no more than only one replica of a said chuck. The more replicas are there, the more available the file is and the faster its retrieval is. With erasure coding, the file is broken into b fixed-size fragments. Each fragment is in turn divided into s fixed-size blocks. Using one of the erasure coding schemes, r redundant fragments are then added as shown in Fig. 1.

One common mistake made when studying P2P storage systems is addressing node availability and data availability as if they were one and the same. However, this is not credible. Assume that data D (be it a replica or an erasure code block) is hosted on a node N, data availability $D(t)$ is the probability that D is available at a moment t in time. Node availability $N(t)$ is the probability that a node N is available, this is strictly dependent upon node online behaviour pattern. Thus, $D(t)$ can be represented as follows:

Fig. 1 Data segmentation in Erasure coding



$$D(t) = \begin{cases} P_{D|N}(t)N(t) & if\ N(t) > 0 \\ 0 & if\ N(t) = 0 \end{cases} \quad (1)$$

Where $P_{D/N}$ is a conditional probability that depicts that data D is only available should node N is available at a given moment in time. So, it is true that when node is not available, data hosted on that node is unavailable as well. Nevertheless, there are cases when a node is online, however it doesn't contribute to the system, and thus data hosted on it is still unavailable. These nodes are referred to as free riders. Table 1 illustrates the possibilities of node and data availability combination.

TABLE 1.
NODE AND DATA AVAILABILITY COMBINATION

	High node availability	Low node availability
High data availability	Nodes who are constantly online and effectively participating	Nodes that participate effectively in the system when they're online which is rare.
Low data availability	Nodes that are continuously online but barely contribute to the system, i.e. free riders.	Nodes that are scarcely online, with less than unacceptable contribution to the system.

The unrestrainable nature of the arrival and departure of peers along with the freedom with which a node decides whether or not to respond to a request represent the two challenging factors that compromise data availability. In this paper, we present a study of the impact of the number of replicas on data availability in the system and the system's overall respond time. We also compare between achieving redundancy using replication and erasure coding in terms of availability and system performance.

II. RELATED WORK

The authors in [1] calculated the probability that a set of nodes are connected to the system during a given amount of time. Nodes with similar that together constitute a robust storage system that is both churn-aware and super-node oriented. [2] listed several data placement policies for the ideal placement of data on system nodes. Results showed that in terms of ease of administration and latency, local date placement on narrow geographical area is the best approach. However, geographical data placement has no effect whatsoever on the probability of the loss of a piece of data. The variance of bandwidth consumption required to maintain redundancy percentage is negligible. In [3], a novel replication model was presented to achieve redundancy through the calculation of node failure and then replicating files on nodes with the least probability of failure. Regarding Erasure coding, [4] offered a detailed analysis of the application of Erasure coding in P2P systems. On the other hand, [5] presented a comparative analysis among multiple Erasure coding patterns and the ideal applications of each of these patterns in terms of repair bandwidth, storage requirement, and the probability of unavailability. The authors of [6] studied data availability in P2P systems in terms of redundancy and fault tolerance. Both [7] and [8] illustrated the direct effect of data redundancy on data availability. Research on P2P storage systems was more focused on testing existing systems that there are very few work on modelling or simulating a P2P storage systems. The authors in [9] offered the requirements for an effective design of a P2P storage system. WHILE [10] PRESENTED A THOROUGH SIMULATION OF THE DOWNLOAD AND RECOVERY SCHEMES IN P2P STORAGE SYSTEMS USING NS2.

III. EXPERIMNET AND RESULTS

Our work in this paper is divided into three parts. First, we distinguish between data and node availability. Second, we inspect the impact of the number of replicas on data availability. Third, we evaluate data availability in terms of fault tolerance in both replication and erasure coding approaches. For these scenarios, we employed the simulation that the authors in [10] presented.

A. Distinguishing data availability and node availability

The model in [10] offers an HTTP-like request model for a file in the storage system. When a hosting node is unavailable, it reasonably follows that all files hosted on this node is unavailable. On the other hand, when a hosting node is available, then data hosted on this node is available with a probability that is related to the node's honesty and the extent of its participation in the system. The model offers two methods: the *request()* method only queries whether the node has a given file but does not request retrieval. The *download()* method demands that the queried node sends the requested file. For this to work, we modified the *request()* function and made it mandatory that a node must answer whether or not it hosts a specific file. Considering the above mentioned, we end up with the following possibilities:

TABLE 2.
NODE AND DATA AVAILABILITY RESPONSE IN SIMULATION

Response	Node	Date
No response for either request() or download()	Unavailable	Unavailable
A response for request() but no response to download()	Available	Unavailable
Response for both request() and download()	Available	Available

We ran the simulation for a period of 24 hours, after which results were illustrated using MATLAB as shown in Fig. 2 below.

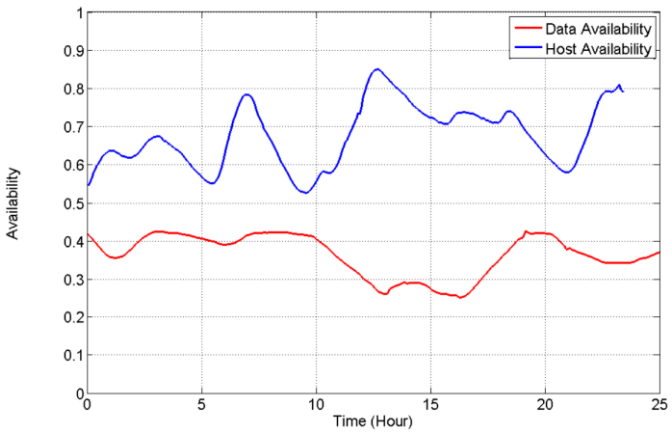


Fig. 2 The difference between data and node availability

Fig. 2 makes obvious the difference between peer node availability and data availability. Data availability is always less than peer availability and that is due to free riders, nodes that are indeed online but nevertheless choose not contribute to the system. These nodes are obliged to answer the *request()* method but they choose not to answer the *download()* method, which means that they host the requested file but refuse to serve it. We can clearly see that it is inaccurate to equate node and data availability. For example, in the figure around the 12th hour of simulation, we witness the vastest difference between data and node availability at values of 0.3 and 0.85 respectively. Thus, node availability is an imprecise indicator of data availability and will deviate any decision that might be made to improve availability, for instance the number of required replicas. In other words, should we deploy replicas based on the node availability value, then the number deployed is 3 times less than what should have been deployed.

B. *The impact of the number of replicas on data availability and response time*

The very first stage of achieving redundancy is replication, i.e. copying replicas of the whole file or its pieces and scattering them over network nodes. For this experiment, we gradually increased the number of replicas of each file and monitored the change in data availability with the increment of replicas, results are illustrated in Fig. 3. As the number of hosted replicas increase, data availability noticeably increases. However, after a certain threshold, 102 replicas, data availability falls off swiftly to reside at zero. This decrease in availability was accompanied by slow simulation response. To illustrate the rationale behind this, we experimented with the mean response time of the nodes in the system. We calculated the mean response time after every increase in replica number, results are depicted in Fig. 4. At the beginning, we notice that as the number of replicas increased, the mean response time decreased. That is due to having many nodes that have the same replica and can serve it in parallel. However, after a certain threshold, the mean response time increased significantly, until the system stopped responding and mean response time went to infinity.

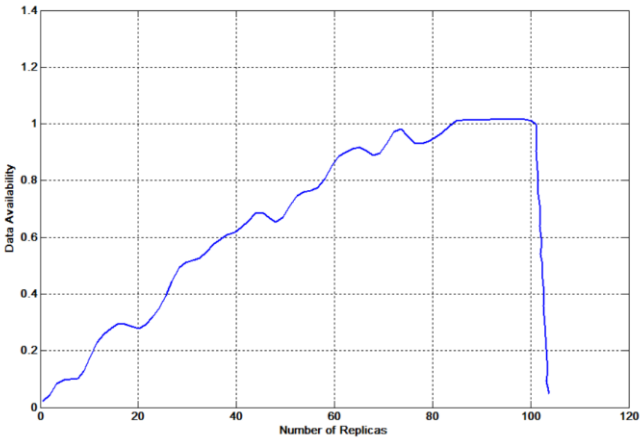


Fig. 3 Data availability vs the number of replicas per file

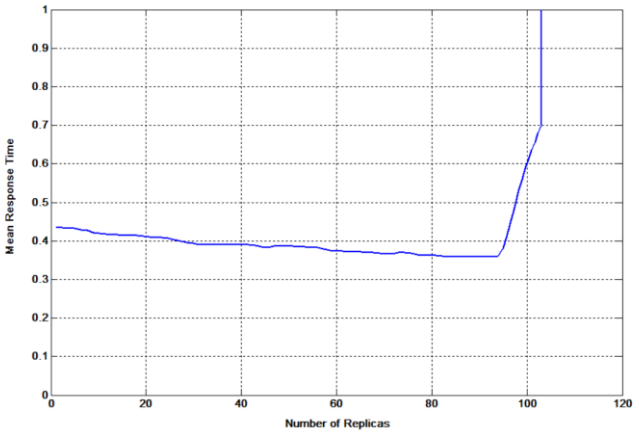


Fig. 4 Mean response time vs the number of replicas per file

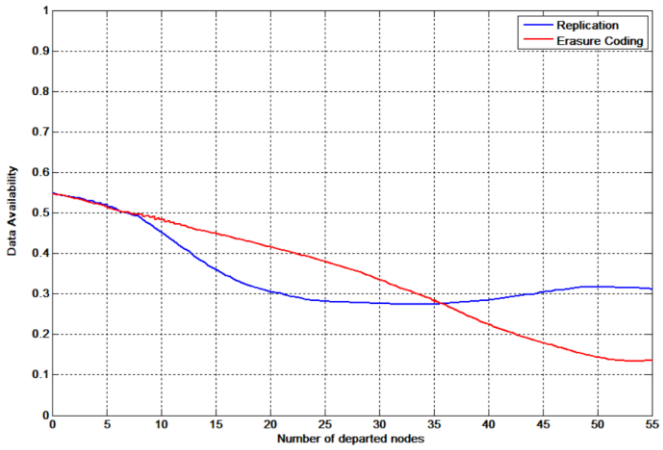
The increase in node mean response time corresponds with the decrease in availability towards zero at the same time that the nodes stop responding even to *request()*. This is due to storage capacity overload, and simulation failure soon after. This demonstrate that although increasing the number of replicas seems intuitive to improve data availability in the system, there is a certain system-dependent threshold that must not be exceeded for two reasons: first, to avoid the exhaustion of available storage capacity in the systems and second, to minimize maintenance, update, and consistency monitoring of replicas.

C. *The impact of replication vs erasure coding on data availability*

The final experiment in this work is comparing the two available redundancy approaches which are replication and erasure coding. Erasure coding has an advantage over replication which is its ability to recover a file with missing pieces. For this experiment, we compared the two approaches

in term of fault tolerance when the node failure ratio increases and their resource consumption.

Fig. 5 The effect of node departure on data availability in case of replication and erasure coding



When the nodes depart the network, the number of replicas available for immediate download in the system decreases significantly possibly leading to the permamnt loss of some file pieces that jeopardizes the file availability. Data availability declines immediately with the departure of nodes in both replication and erasure coding. However, the declinison is less severe in case of erasaure coding. This is due to the fact that although some pieces are missing, the file is still recoverrable due to parity chunks. However, as more nodes leave the system, replication proves to be more effective as data avilability using erasure coding declines significantly. This is due to the fact there are still pieces that can be recovered of the file in case of replication. On the other hand, losing enough parity chucks renders the file permanently unrecoverable.

For the final experiment, we compared replication vs erasure coding in terms of resources consumed on the peripheral node, Fig. 6.

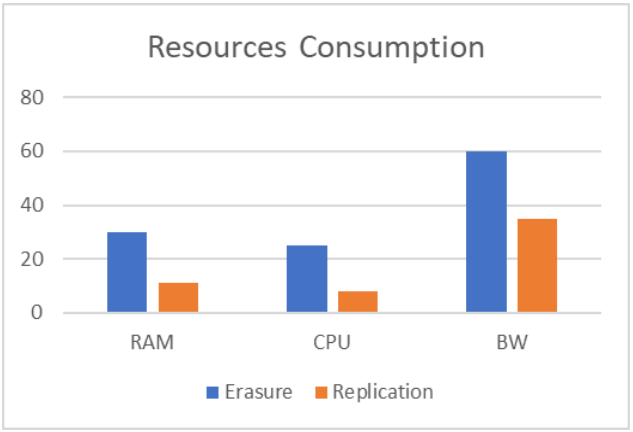


Fig. 6 Erasure coding vs replication in terms of resources consumed on nodes

Erasure coding consumes more bandwidth, RAM and CPU than replication. This is due to maintenance and updates costs along with the calculation of parities whenever a file or a part of it are modified. Thus, erasure coding is preferable of super nodes or in systems where file updates are not frequent, such as back-up systems. Replication is recommended for regular robust storage systems where more value is placed on speed and resources.

IV. CONCLUSION

Although P2P storage systems offer a decent alternative for cloud storage, these systems are yet to tackle the obstacles that threaten their popularity among potential users. One major obstacle is the impact of system churn on data availability. This paper compared both data redundancy approaches, erasure coding and replication, on data availability. Results showed that replication constitutes the better approach for redundancy in systems where files are frequently updated due to the fact that these updates are immediate and consume minimum processing power unlike erasure coding that needs to re-calculate parity. However, erasure coding performs better than mere replication in case of back-up systems where data files are relatively large and demand is at minimum.

ACKNOWLEDGMENT

The authors wish to acknowledge the Faculty of Information Engineering in Tishreen university for their support of this research.

REFERENCES

[1] X. Meng, “A churn-aware durable data storage scheme in hybrid P2P networks,” J. Supercomput., vol. 74, no. 1, pp. 183–204, Jan. 2018.

[2] S. Caron et al., “P2P Storage Systems: Study of Different Placement Policies,” 2013.

[3] T. Zhang, “A novel replication model with enhanced data availability in P2P platforms,” Int. J. Grid Distrib. Comput., vol. 9, no. 4, pp. 151–160, 2016.

[4] G. Nychis, A. Andreou, D. C.-I, “Analysis of erasure coding in a peer to peer backup system,” pdfs.semanticscholar.org.

[5] S. M. Singal, N. Rakesh, and R. Matam, “Storage vs repair bandwidth for network erasure coding in distributed storage systems,” Int. Conf. Soft Comput. Tech. Implementations, ICSCTI 2015, pp. 27–32, 2016.

[6] N. U. I. Galway, “Data availability analysis in P2P networks,” 2012.

[7] R. Gracia-Tinedo, M. S. Artigas, and P. García López, “Analysis of data availability in F2F storage systems: When correlations matter,” 2012 IEEE 12th Int. Conf. Peer-to-Peer Comput. P2P 2012, pp. 225–236, 2012.

[8] A. Dandoush, S. Alouf, and P. Nain, “Lifetime and

availability of data stored on a P2P system: Evaluation of redundancy and recovery schemes," Comput. Networks, vol. 64, pp. 243–260, 2014.

[9] Dandoush, Abdulhalim, Sara Alouf, and Philippe Nain. "Simulation analysis of download and recovery processes in P2P storage systems." 2009 21st International Teletraffic Congress. IEEE, 2009.

[10] Dalle, Olivier, and Emilio P. Mancini. "Integrated tools for the simulation analysis of peer-to-peer backup systems." Proceedings of the 5th International ICST Conference on Simulation Tools and Techniques. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2012.

[11] S. Nazir and M. Hauswirth, "Time series analysis of P2P networks to improve data availability," Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA, pp. 283–290, 2011.

[12] A. Dandoush, "Analysis and optimization of peer-to-peer storage/backup systems," p. 201, 2010.

[13] Y. Hassanzadeh-Nazarabadi, A. Küpçü, and Ö. Özkasap, "Decentralized and locality aware replication method for DHT-based P2P storage systems," Futur. Gener. Comput. Syst., vol. 84, no. March, pp. 32–46, 2018.