Effect of Data Placement on the Reliability of Data Storage Systems

Invited Talk

Vinodh Venkatesan
IBM Research – Zurich
8803 Rüschlikon, Switzerland
ven@zurich.ibm.com

***Abstract*—Data redundancy, in the form of replication or advanced erasure codes, is used to protect data from storage node failures. It is known that that the placement of this redundant data across storage nodes can have a significant impact on the reliability, especially for large-scale storage systems. In particular, a declustered placement of redundant data is shown to have significantly higher reliability than the traditionally-used clustered placement for many redundancy schemes. This implies that significant gains in reliability can be obtained without losing storage efficiency by choosing the declustered placement scheme. Approximate expressions for the mean time to data loss of the system in terms of the various parameters of the system are obtained by considering the shortest paths to data loss when node failures occur and rebuild processes commence. These expressions are shown to hold true for parameters of practical interest through detailed event driven simulations.**

I. Introduction

Modern data storage systems are extremely large and consist of several tens or hundreds of nodes. In such systems, node failures are daily events, and safeguarding data from them poses a serious design challenge. Data redundancy, in the form of replication or advanced erasure codes, is used to protect data from node failures (see [1] and references therein). By storing redundant data across several nodes, the redundant data on surviving nodes can be used to rebuild the data lost by failure of nodes. As these rebuild processes take some time to complete, there exists a non-zero probability of additional node failures during rebuild, which eventually may lead to a situation in which some of the data have lost so much redundancy that they become irrecoverably lost from the system. The average time taken by the system to suffer an irrecoverable data loss, also known as the mean time to data loss (MTTDL), is a measure of data reliability that is commonly used to compare different redundancy schemes and to study the effect of various design parameters. Traditionally, the MTTDL for simpler systems, e.g. RAID arrays, have been obtained by the use of Markov models, e.g., [2], or through the use of event-driven simulations, e.g., [3]. Whereas the simulations-based approach is time-consuming and does not yield expressions in terms of the various system parameters, the former approach is not applicable for non-exponential real-world failure and rebuild time distributions and becomes challenging for general data placement schemes. To address this issue, a theoretical methodology for reliability analysis is developed in [4] that is based on the probability of direct path to data loss during rebuild. The reliability analysis is detailed in the sense that it accounts for the rebuild times involved, the amounts of partially rebuilt data when additional nodes fail during rebuild, and the fact that modern systems use an intelligent rebuild process that will first rebuild the data having the least amount of redundancy left. Through rigorous arguments and simulations it is established that the methodology developed is well-suited for the reliability analysis of real-world data storage systems. Applying this methodology to data storage systems with different types of redundancy, various data placement schemes, and rebuild constraints, the effect of the design parameters on the system reliability is studied in this article.

TABLE I
Parameters of a storage system

c	amount of data stored on each storage node (bytes)
n	number of storage nodes in the system
$c\mu$	average read-write rebuild bandwidth of a storage node (bytes/s)
$1/\lambda$	mean time to failure of a storage node (s)
$1/\mu$	mean time to read/write c amount of data from/to a node (s)
(l, m)	erasure code parameters

II. System Model

The storage system is modeled as a collection of n independently failing *storage nodes* each of which stores c amount of data. In addition to the space required for the c amount of data that is stored, each node is assumed to have sufficient spare space that may be used for a distributed rebuild process when other nodes fail. The main parameters used in the storage system model are listed in Table I. An (l, m) maximum distance separable (MDS) erasure code is used to protect data from node failures, which maps l user data symbols to m codeword symbols that are stored in the system. Two redundancy placement schemes are considered, namely, clustered and declustered. In clustered placement, the n nodes are divided into n/m sets (or clusters) of m nodes each and each m length codeword is placed in one of these clusters. When a node in a cluster fails, the redundant data from the $m-1$ surviving nodes of that cluster are used to reconstruct the lost data in a new node. In declustered placement, each of the $\binom{n}{m}$ possible ways of storing codewords are equally used. When a node fails in the system, the redundant data corresponding to that node is spread equally across all the remaining $n-1$ surviving nodes. This data is then used to reconstruct the lost

978-1-4799-0838-7/13/$31.00 ©2013 IEEE

data in a *distributed rebuild* process that restores the data in the spare space of these $n-1$ nodes. In large systems, n is much larger than m and therefore, declustered placement can result in extremely fast rebuilds. Furthermore, when additional nodes fail, the amount of data that becomes critical (that is, loses more redundancy) is smaller for declustered than for clustered. Therefore, an *intelligent rebuild* process that rebuilds the most critical data first significantly decreases the chances of data loss in declustered placement. See [4], [5] for a more detailed discussion on the system model.

III. Effect of Data Placement on Reliability

For a general class of failure and rebuild distributions, which include real-world distributions such as Weibull and gamma, the following expressions for MTTDL can be derived using the methodology developed in [4]:

$$\mathrm{MTTDL}^{\text{clus.}} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{1}{\binom{m-1}{l-1}} \frac{M_1^{m-l}(G_\mu)}{M_{m-l}(G_\mu)}. \tag{1}$$

$$\mathrm{MTTDL}^{\text{declus.}} \approx \frac{\mu^{m-l}}{n\lambda^{m-l+1}} \frac{(m-l)!}{(l+1)^{m-l}} \frac{M_1^{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)}{M_{m-l}\left(G_{\frac{n-1}{l+1}\mu}\right)} \times \prod_{e=1}^{m-l-1} \left(\frac{n-e}{m-e}\right)^{m-l-e} \tag{2}$$

The relative errors in the above approximations tend to zero as λ/μ tends to zero. In the above expressions G_x denotes the rebuild distribution with mean rebuild time x, and $M_y(G_x)$ is the yth moment of G_x. It is known that real-world storage nodes are *generally reliable*, that is, the mean time to read all contents of a node (which is typically of the order of tens of hours) is much smaller than the mean time to failure of a node (which is typically at least of the order of thousands of hours). So, it follows that generally reliable nodes satisfy $\lambda/\mu \ll 1$. The validity of these expressions has been verified by detailed event-driven simulations (refer [4] for details). Figure 1 shows the comparison between the theoretically-predicted MTTDL values and the simulation-based MTTDL values for systems using $(3,4)$ and $(6,8)$ MDS codes.

Expressions (1) and (2) reveal that, for large systems using an erasure code with two or more parities, i.e., $m-l \geq 2$ (e.g. 3-way replication, (6,8)-MDS erasure code, etc.), declustered placement of codewords can significantly improve system MTTDL. This is primarily due to the fact that when additional nodes fail during rebuild, the amount of critical data, that is, data with the least amount of redundancy, is significantly lower for declustered placement than clustered placement. This implies that an intelligent rebuild process, which always rebuilds the critical data first, is able to more effectively prevent data loss in a system with declustered placement. Fig. 2 shows how the MTTDL varies as a function of both the codeword length and the spread factor for single parity codes, for a given number of nodes n. Spread factor generalizes the concept of spreading codewords across the nodes of the system for symmetric placement schemes [4]. In Fig. 2, clustered placement corresponds to the cases where the spread factor

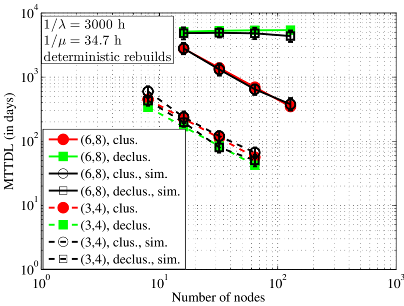

Fig. 1. MTTDL of two different erasure codes with the same storage efficiency for a system with mean time to node failure $1/\lambda = 3000$ h and mean time to read all contents of a node during rebuild $1/\mu = 34.7$ h.

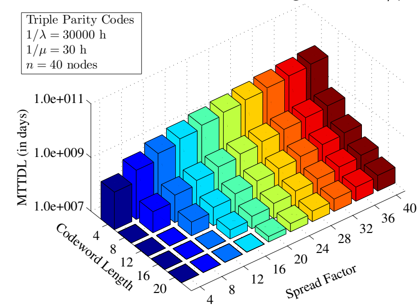

Fig. 2. MTTDL for triple parity codes; $1/\lambda = 30000$ h and $1/\mu = 30$ h

is equal to the codeword length, and declustered placement corresponds to the case where the spread factor is equal to the number of nodes.

IV. Conclusion

Besides the choice of erasure codes, the choice of codeword placement offers another degree of freedom to improve the reliability of data storage systems. In particular, declustered placement of codewords can significantly improve the reliability of systems using erasure codes with two or more parities. However, this result makes the following assumptions: (1) there is sufficient network bandwidth to rebuild from all nodes in parallel, (2) there is no correlation among permanent node failures, and (3) latent sector errors are ignored. A model avoiding assumption (1) has been analyzed in [4]. When assumption (2) is not made, node failure correlations may result in worse MTTDL for declustered placement. If the correlations are known beforehand, the reliability of clustered placement can be improved by choosing the clusters carefully. Building a model that avoids assumption (3) is part of ongoing work.

References

[1] J. S. Plank and C. Huang, "Tutorial: Erasure coding for storage applications," Slides presented at 11th Usenix Conference on File and Storage Technologies (FAST'13), San Jose, February 2013.

[2] S. Ramabhadran and J. Pasquale, "Analysis of long-running replicated systems," in *Proc. 25th IEEE Int'l Conference on Computer Communications (INFOCOM'06)*, 2006, pp. 1–9.

[3] K. Greenan, "Reliability and power-efficiency in erasure coded storage systems," Ph.D. dissertation, University of California, Santa Cruz, 2009.

[4] V. Venkatesan, "Reliability analysis of data storage systems," Ph.D. dissertation, Ecole Polytechnique Fédérale de Lausanne, Switzerland, 2012.

[5] V. Venkatesan and I. Iliadis, "Effect of codeword placement on the reliability of data storage systems," in *Proc. 10th Int'l Conference on Quantitative Evaluation of Systems (QEST'13)*, 2013.