Analysis for REPERA: A Hybrid Data Protection Mechanism in Distributed Environment

Longbin Lai, Shanghai Jiao Tong University, China
Linfeng Shen, IBM China Development Lab, China
Yanfei Zheng, Shanghai Jiao Tong University, China
Kefei Chen, Shanghai Jiao Tong University, China
Jing Zhang, Shanghai Jiao Tong University, China

ABSTRACT

Distributed systems, especially those providing cloud services, endeavor to construct sufficiently reliable storage in order to attract more customers. Generally, pure replication and erasure code are widely adopted in distributed systems to guarantee reliable data storage, yet both of them contain some deficiencies. Pure replication consumes too much extra storage and bandwidth, while erasure code seems not so high-efficiency and only suitable for read-only context. The authors proposed REPERA as a hybrid mechanism combining pure replication and erasure code to leverage their advantages and mitigate their shortages. This paper qualitatively compares fault-resilient distributed architectures built with pure replication, erasure code and REPERA. The authors show that systems employing REPERA share with erasure-resilient systems a higher availability and more durable storage with similar space and bandwidth consumption when compared with replicated systems. The authors show that systems employing REPERA, on one hand, obtain higher availability while comparing to erasure-resilient systems, on the other hand, benefit from more durable storage while comparing to replicated systems. Furthermore, since REPERA was developed under the open platform, REPERA, the authors prepare an experiment to evaluate the performance of REPERA by comparing with the original system.

Keywords: Cloud, Computer Science, Distributed System, Erasure Code, HDFS, Pure Replication, Reliable Storage, REPERA

1. INTRODUCTION

Cloud has now attracted both commercial and academic focus. Cloud sophisticatedly combines the economic, commercial, cultural and technological conditions, causing a disruptive shift in IT industry towards a service-based economy. Usually, large distributed systems, especially those providing services as cloud, are built upon cheap hardware to reduce cost and are exposed to the Internet to accumulate customers accompanied with their massive

DOI: 10.4018/ijcac.2012010105
information. However, cheap components follow by potential hardware failures, while public exposure leads to the facilities of hack attacks and viruses. Furthermore more, disk drives are statistically the most commonly replaced hardware components in large storage systems, accounting for almost 30% to 50% of all hardware replacements (Schroeder & Gibson, 2007). Thereby, mechanism providing reliable storage turns out as one of the most essential components in design of distributed systems.

Generally, pure replication and erasure code are introduced to guarantee highly-available storage, and they obtain widespread adoption in distributed systems. Google File System (Ghemawat, Gobioff, & Shun-Tak, 2003), Pastry (Rowstron & Druschel, 2007), and Tapestry (Zhao, Kubiatowicz, & Joseph, 2001) are typical replicated systems, in which data is mirroring not only to tolerate failures, but also to support concurrent data access in consideration of latency reduction. However, pure replication consumes too much extra storage and bandwidth, hence boosting the cost for deploying additional devices and managements. Erasure code has then been introduced into storage systems to increase storage durability (also known as the expected mean time to failure, or MTTF, of losing any data unit) and to reduce the investment of preparing massive storage as well as the extra transferring and maintaining overhead for the replicas. The key idea behind erasure code is that m blocks of source data are encoded to produce n blocks of encoded data, in such a way that any subset of m encoded blocks suffices to reconstruct the source data. Such a code is called an $(n, m)$ erasure code and allows up to $(n - m)$ losses in a group of $n$ encoded blocks (Rizzo, 1997).

Considering a data sized $x$, 3-way replication consumes $3 \times x$ storage space as well as transferring bandwidth, while $(7, 4)$ erasure code only takes up nearly $7/4 \xi$, to suffer up to 3-disk failure. Chen, Edler, Goldberg, Gottlieb, Sobti, and Yianilos (1999) initially implemented erasure code in the distributed context to build a high-efficient archival intermemory. Aguilera and Janakiraman (2005) presented methods of efficiently using erasure code in distributed systems. Weatherspoon and Kubiatowicz (2002) pointed out that systems employing erasure code have MTTF many orders of magnitude higher than replicated systems with similar storage and bandwidth requirements, and reversely, erasure-resilient systems take up much less bandwidth and storage to provide similar system durability as replicated systems. However, besides an extra computation for coding procedure, erasure coding mechanism disappoints system designers for a longer access delay as multiple servers containing the erasure stripes have to be contacted to read a single block and it encounters bottlenecks when either modifications or failures happen frequently, which may introduce excessive coding processes as to possibly drain the system.

From the aforementioned discuss, we notice that on one hand, pure replicated systems benefit from high access performance yet consume intolerable extra resources, on the other hand, pure erasure-resilient systems win more durable storage yet may not be so high-efficiency in reducing access delay and only suitable for read-only contexts. A nature mind is to combine them in a single system. Kubiatowicz et al. (2000) introduced the hybrid method coordinated with version mechanism in OceanStore. They classified objects into active form representing the latest version of its data, which is replicated for performance consideration, and archival form indicating a permanent, read-only version of the object, which is thereby, encoded with erasure code to achieve more durable storage (Kubiatowicz et al., 2000). However, version itself brings in overhead and for some large storage systems, data in older version is not often useful, which as a matter of fact, leaves OceanStore a pure mirroring system.

We propose REPERA to truly combine replication and erasure code as an objective to leverage their advantages and mitigate their deficiencies. The remainder of this paper is organized as follows. Section 2 presents the main idea of designing REPERA. A qualitative analysis of REPERA will be specifically...
10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage: 
www.igi-global.com/article/analysis-repera-hybrid-data-protection/64636?camid=4v1

This title is available in InfoSci-Journals, InfoSci-Journal Disciplines Computer Science, Security, and Information Technology. Recommend this product to your librarian: 
www.igi-global.com/e-resources/library-recommendation/?id=2

Related Content

A Service Science Perspective on Human-Computer Interface Issues of Online Service Applications
www.igi-global.com/article/service-science-perspective-human-computer/2526?camid=4v1a

MoBiFlow: Principles and Design of a Workflow System for Molecular Biology
www.igi-global.com/article/mobiflow-principles-design-workflow-system/60736?camid=4v1a

Design of Quality Aspects in Service Oriented Architecture through Service Level Agreements: The Streaming Case Study
www.igi-global.com/chapter/design-quality-aspects-service-oriented/52188?camid=4v1a
E-Personalization and Its Tactical and Beneficial Relationships to E-Tailing


www.igi-global.com/article/personalization-its-tactical-beneficial-relationships/65747?camid=4v1a