CS443 Paper Review
Lei Yang
2005-5-02

**********

Title

Glacier: Highly durable, decentralized storage despite massive correlated failures

Author

Andreas Haeberien, Alan Mislove, Peter Druschel

Summary

This paper presents Glacier, a distributed storage system that relies on massive redundancy to mask the effect of large-scale correlated failures.

Most important ideas

Due to virus and worm attacks, large-scale correlated failures are observed in the internet. Many P2P storage systems assume failure independence, which is unrealistic. In addition, worms can cause large-scale correlated Byzantine failures. The authors pointed out that it is essential for a distributed storage system to maintain the durability for important and otherwise unrecoverable data, such as business data, personal records, calendars and user emails, without any assumption or introspection of independent node failures.

The proposed approach, Glacier, is designed to be robust to large-scale correlated failures. It assumes the exact nature of the correlation among failures to be unpredictable. The key idea behind Glacier is to trade efficiency in storage utilization for durability, thus turning abundance into reliability. Hence, Glacier must create extra redundancy to ensure that data survives any correlated failure with high probability. The main challenges of this work are: (1) minimizing storage requirement, (2) minimizing bandwidth requirement, and (3) defense against attacks and Byzantine failures.

Glacier uses erasure codes and garbage collection to mitigate the storage cost of redundancy and relies on aggregation and a loosely coupled fragment maintenance protocol to reduce the message costs. Their experiments show that message overhead is low; however, the storage overheads can be substantial when the availability requirements are high and a large fraction of nodes is assumed to suffer correlated failures.

Flaws/Questions

Firstoff, I don't like the argument the authors made to justify the storage cost of their approach, "Fortunately, disk space on desktop PCs is a vastly underutilized resource...Glacier leverages this abundant but unreliable storage space to provide durable storage for critical data". Well, if disks are so cheap and underutilized, why don't we just use an extremely large disk drive for basic storage and another for backups? Why would people bother to explore the distributed storage to backup their data? In addition, the fact that there is still large space left in the hard disk does not necessarily indicates that this space is left casually -- it might be reserved by the user deliberately!

Second, the authors attacked other approach saying that they do not provide hard durability guarantees. This approach, to me, doesn't either. It could provide a higher probability that important data survives with correlated failure, but it cannot guarantee that the data will never be lost.

Third, a minor detailed question. In Section 4, Glacier operates with a primary store, which maintains a small number of full replicas of each data object. What if there is a failure on this primary store?

Relevance/Potential Future Research

Glacier was evaluated on ePOST, a decentralized email system. It would be interesting to see it in use on other systems, especially those which have experienced severe correlated Byzantine failures. More importantly, it should be useful to compare Glacier and other approaches, such as TotalRecall, OceanStore, and PAST, on exactly the same system, with more or less the same workload. Then the tradeoff between storage and durability can thus be viewed more clearly. Basically, my doubt is that even though disks are cheap, I'd prefer not to waste disk spaces unless super necessary. I'd say I am not a super fan of this approach, unless they could convince me with the results of such comparisons on real-life systems: Glaciers rescues most of the important data due to correlated failures, while other approaches cannot. Otherwise the motivation is not strong enough.