CS5673 – Distributed Systems – Spring 2012
Instructor – Professor Mishra
Semester Project Proposal - Distributed File Storage
Project Members – Bryan Dixon, Brendan Kelly, Mark Lewis-Prazen, Andrew Sayler

Problem Statement

The new generation of applications requires the processing of terabytes and petabytes of data. This processing has largely been achieved by distributed processing architectures and methodologies, driving the rise of companies such as Google, Amazon and Yahoo, which have embraced such technologies over the past ten years. The increased importance of large scale data processing and a corresponding explosion in alternative storage architectures and service strategies requires computer science researchers and professionals to become well versed in the available alternatives and trade-offs in these areas. Yet, while the technology has evolved significantly over the past five years, a deep understanding of both technical and business storage issues has lagged behind as many in these fields still view data storage as a mundane activity confined to some back office function. The current trend in distributed cloud storage and the focus of delivering IT infrastructure and applications bundled with data storage has led to an "as a service" model being used to promote such products. This bundling has in turn lead to the proliferation of industry and technical buzzwords which confuse the issue and discount importance of the storage component decision for businesses.

The intent of our project is to examine the current state of both core storage and related service offerings on both a technical and business level to understand (1) what is technically available, (2) what are current research and developmental directions in storage technology, and (3) how core storage architectures are being coupled with associated services to produce bundled storage product offerings that purport to add value above and beyond core offerings.

We believe that doing this analysis and a subsequent implementation of a distributed proof of concept storage model, we will gain a better understanding of evolving storage technologies as well as the tradeoffs that need to be made when implementing bundled storage services in a production environment.

Project Approach

Our project approach will consist of three phases. In the first we will review the current research and technical literature with respect to distributed storage in order to gain an understanding of the current state of the technology offerings as well as potential future directions in distributed storage architecture and product offerings. In the next phase we propose to define an appropriate architecture for a proof-of-concept distributed storage system that can be implemented in some detail and justify our choices in that design. In the final phase we propose to implement the design noted above to demonstrate proof of concept. If additional time is available we propose to add additional services and features not part of the core design in order to explore newer services currently available in existing storage products.

The exploratory phase can be further summarized as follows:

- Investigating various distributed file systems (DFS) as a prelude to design
- Determining DFS best use cases
- Identifying DFS security features
- Identifying DFS fault tolerance provisions
- Implementing selected features in a proof-of-concept model
- Evaluating how the system operates against selected industry metrics
- Comparing our proof-of-concept model to other models proposed in the literature
- Investigating improvements that could be made on our model

In conjunction with our project we have requested EEF funding for associated hardware in order to build a rudimentary test platform for implementing a proof of concept of our design. The intent is that these machines will serve as a both proof of concept and subsequent testing platform for our implemented distributed file storage design. Then, at the conclusion of the project these hardware resources can be used to enhance available computing resources for the graduate computer science program.

Evaluation Methodology

We will evaluate our implemented design by comparing its performance against popular storage processing and storage services metrics available from prior literature and vendor papers as they are reviewed. We expect criteria for evaluation to include, but not necessarily be limited to the following:

- Failure handling and automatic recovery
- ACID (Atomicity, Consistency, Isolation, Durability) properties
- Efficiency
- Performance
- Cost efficiency
- Availability
- Durability
- Security/Authentication/Authorization

- Incremental Scalability - Ability to add one host at a time
- Simplicity - With all this complexity, the developer interface should still be simple to use

A complete list of criteria for evaluation and actual parameter bounds will be finalized at the conclusion of the literature review and then subsequently updated at each project update step as new information on industry standards and our model's performance (or lack thereof) become more apparent through the project implementation and our corresponding research.

Selected References

F. Chang et al. 2006. Bigtable: A Distributed Storage System for Structured Data. In OSDI: Symposium on Operating Systems Design and Implementation.

GHEMAWAT, S., GOBIOFF, H., AND LEUNG, S.-T. 2003. The google file system. In Proceedings of the nineteenth ACM symposium on Operating systems principles. ACM Press, 29–43.

Giuseppe DeCandia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakshman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall and Werner Vogels. 2007. Dynamo: Amazon's Highly Available Key-value Store. SOSP'07, October 14–17, 2007, Stevenson, Washington, USA. Copyright 2007 ACM