Field:
Mathematics/Applied Mathematics/Informatics

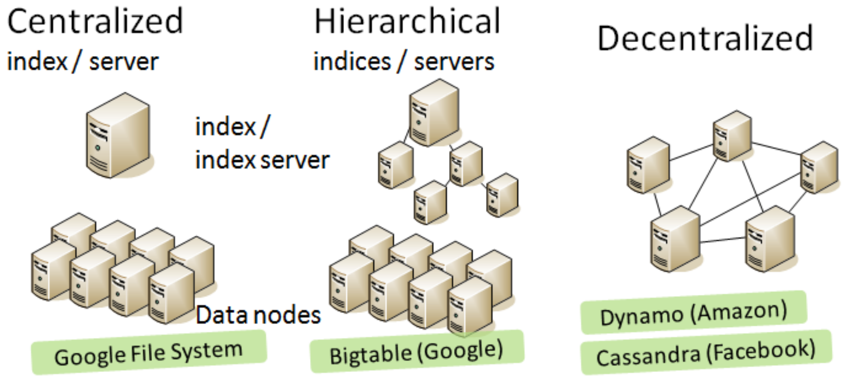
Session Topic:
Web-Scale Computation

Speaker:
Kazuyuki SHUDO, Tokyo Institute of Technology

1. Introduction

Computing resources supporting our activities are moving over from PC and workstations to the Clouds across Internet. Cloud is collective computing resources, which consist from computers, network, data storage, and data themselves. Cloud serves a large number of users and clients and it is pushing the demand for large-scale data store exceeding capacity several computers can provide. Cloud storage [1,2], a kind of distributed data store is today intensively studied to satisfy the requirement. Scalability is a key property of such data stores. A Cloud storage is required to scale from several computers to hundreds of computers in a data center keeping a certain extent of data consistency and service availability.

2. Decentralized distributed data store

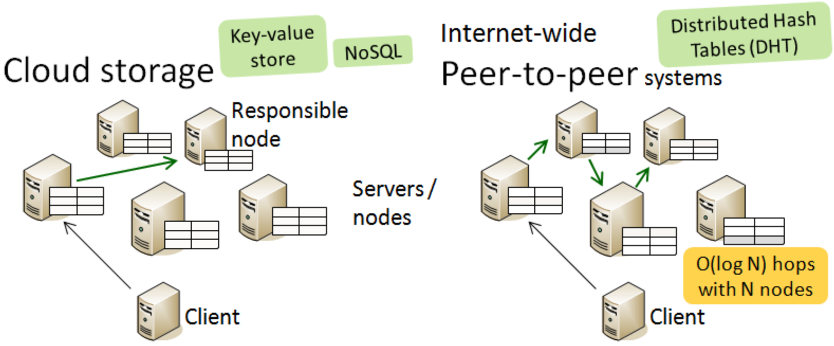


We have choices on which architectural styles to adopt when constructing a distributed system. A style involves centralized servers to orchestrate multiple computers, and in another style all computers are completely decentralized. There are middle styles between them. Naturally supposed, decentralized style scales well, but it introduces many difficulties. "Decentralized" means absence of God's eye, in other words, lack of a summarized directory. For example, it is difficult to balance the amount of data each computer holds when storing. It is also difficult to locate a computer holding a datum efficiently when retrieving.

3. Introduction of peer-to-peer techniques

Current Cloud storages try to share the identical summarized directory among all participating computers. But it is becoming more difficult to share the up-to-date directory consistently as the number of computers increases. It is possible to overlook a datum without the consistent directory. Scalability is limited with the current approach. We have introduced peer-to-peer techniques to Cloud storage to overcome the limitation. For example, the approach allows a Cloud storage to relay an access request to a more appropriate responsible computer. Such a relay relaxes the requirement on consistency.

4. Unified routing algorithm



A Cloud storage today takes “all-knows-all” approach in which all computers knows all the others. All computers have to be able to route a request to the responsible computer. In contrast with it, peer-to-peer systems takes “partial knowledge” approach by which a request approaches to the responsible computer step by step. The two approaches are regarded as different ones. Especially, Cloud storages have not adopted the latter approach because of relaying costs. But the latter approach essentially includes the former, and the former is a special form of the latter. Based on the observation, we designed the unified routing algorithm, FRT-Chord. It works as “all-knows-all” algorithm as long as the capacity of a routing table is enough to hold all other computers. It also works as “partial knowledge” algorithm in case a routing table overflows. “FRT” is a methodology to design a routing algorithm and FRT-Chord is an algorithm based on Chord, a peer-to-peer routing algorithm. FRT can be applied to other routing algorithms. We could design FRT-Kademlia as well.

An implementation of FRT-Chord is included and available in Overlay Weaver [3].

References

[1] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, R. E. Gruber, “Bigtable: A Distributed Storage System for Structured Data”, Proc. OSDI’06, 2006.

[2] A. Lakshman, P. Malik, “Cassandra – A Decentralized Structured Storage System”, Proc. LADIS 2009, 2009.

[3] K. Shudo, Y. Tanaka, S. Sekiguchi, “Overlay Weaver: An Overlay Construction Toolkit”, Computer Communications, Elsevier Science, Volume 31, Issue 2, pp.402-412, 2008.