WAND: A Meta-data Maintenance System over the Internet

Anubhav Bhatia, Saikat Mukherjee, Saugat Mitra, Srinath Srinivasa
Indian Institute of Information Technology
26/C, Electronics City
Bangalore 560100, INDIA
{anubhav.bhatia,saikat.mukherjee,saugat.mitra,sri}@iiitb.ac.in

ABSTRACT

WAND is a meta-data management system that provides a file-system tree for users of an internet based P2P network. The tree is robust and retains its structure even when nodes (peers) enter and leave the network. The robustness is based on a concept of virtual folders that are automatically created to retain paths to lower level folders whenever a node hosting a higher-level folder moves away. Other contributions of the WAND system include its novel approach towards managing root directory information and handling network partitions.

Categories and Subject Descriptors: E.1 Data Structures [Distributed data structures]

Keywords: Peer-to-Peer, Wide-area distributed file system, Meta-data maintenance

1. INTRODUCTION

Peer-to-peer (P2P) systems are flexible and amorphous means of sharing data over the Internet. A significant challenge in a P2P network is to locate data elements in the network. Distributed hash tables (for example Chord [2]) and data-centric networks try to address this problem.

While hashing is suitable for keyword based searches, it is often desirable to evolve and maintain a global meta-data schema that allows users to browse through the network. Meta-data have taken various forms like ontologies, description logics, etc. In our work, we consider a simple form for meta-data: a file-system tree. The objective of this work is to evolve and maintain a global file system structure over a wide-area P2P network. The file system tree should be resilient to changes happening in the system and retain its structure as much as possible whenever peers leave without warning. The proposed model is called WAND (Wide-Area Network Directory).

WAND is primarily meant for *meta-data management.* It is not a full file-system in that it does not bother about managing data blocks and relies instead on the native file system(s) on the peer hosting WAND. In a WAND network, peers share one or more directories from their local file system. WAND provides a file system tree where peers can mount their shared directories. The global file system structure is implemented in a way that each peer is in charge of one or more mounted directories. The robustness of the directory structure depends upon two novel ideas: virtual directories and a distributed algorithm for detecting handling network partitions.

2. OVERALL ARCHITECTURE

A WAND P2P network comprises of a number of *nodes* or *peers* sharing one or more directories from their local file system. The network appears as a file system with a virtual root directory.A peer shares a directory by `mount()`ing it under some existing directory in the WAND file system. The name by which a shared directory appears in the WAND tree is called its “mount point.”

In any file-system the root directory is of prime importance and hence the root directory structure in WAND is given extra resilience. This is facilitated by distributing root directory information across the network in an asynchronous, need-to-know basis. In addition, all root-level directories are maintained in a closely coupled fashion. Nodes at the root level jointly handle any updates or failures occurring at the root level.

The next section describes the key concepts used in the WAND file-system. A detailed description of WAND alongwith literature survey and performance statistics can be obtained from [1].

3. CONCEPTS IN WAND

The key concepts in the WAND filesystem are Virtual folders, Horizontal and Vertical Caches, root-level chord ring and a novel approach to network partioning.

Virtual Folders: Virtual folders unlike normal folders do not map to any existing directory on a user's system. They are used to maintain the consistency of the directory structure. Virtual folders are created when a node which has one or more mount points gets disconnected from the WAND network. The mount points created by the disconnected node are replaced by virtual folders maintained by some other node in the network. Although the files being shared by the disconnected node are not available anymore, the directory structure remains intact because of the virtual folders. All the root level folders are virtual folders and a user may create a virtual folder at the root level.

Root-level Chord ring The root level folders are of prime importance as users enter the directory structure through the root of the file system. To ensure that root level folders are always available, all the root folders are virtual. One or more nodes share the root level virtual folders by organizing themselves into a Chord [2] ring.

The first node that forms the file system is called the *WAND* node. The WAND node ensures that there are no name clashes at the root level. If the WAND node goes down, the next logical node at the root level (according to the Chord ring) becomes the WAND node. If all root level nodes go down, the file system becomes partitioned. One or more nodes from lower levels independently recreate the file system tree to form independent file systems.

Horizontal and Vertical Caches: The entire WAND directory structure is not maintained on a single node but is distributed across nodes in the entire network. The structure is maintained in two types of caches, a *horizontal cache* and a *vertical cache*. The horizontal cache at any node depicts the node's latest knowledge of the root level directories. The vertical cache at any node contains knowledge of the directory tree structure below all the mount point owned by the node. The depth of the vertical cache should be atleast 2; i.e. including information about its grand children. A set of cache update policies determine how vertical and horizontal caches are updated.

Horizontal Cache Updates The update frequency is determined by two different configurable parameters known as *Horizontal Refresh Rate (HRR)* and *Root-cache Update Interval (RUI)*. The HRR specifies the maximum permitted percentage change introduced by update log before the horizontal cache is refreshed. The RUI specifies the maximum time period that can elapse between any two horizontal cache updates.

Vertical Cache Updates Vertical Cache updates are sent from every child directory to its parent directory. The *Vertical Refresh Rate (VRR)* is a configurable parameter, which determines the percentage of change in the vertical cache that triggers the vertical cache update.

Network Partioning Network partitions are handled by splitting the file system tree into two or more subtrees. The splitting algorithm is such that a node tries its best to remain in its file system tree. But even after trying its best, if it still cannot contact any node, it decides to "walk away" and form its own network containing the same subtree to which it belongs. It does not matter whether the network is really partitioned; the objective is to maintain consistent subtrees whenever the tree splits. Details about the partition detection algorithm and the cache update algorithms can be found in [1].

4. PERFORMANCE STATISTICS

Performance of the WAND network was evaluated using simulation to test scalability against number of nodes and over the PlanetLab[1] network to test scalability against geographical distribution.

The number of messages being passed between nodes is dependent on the number of events occurring in the network (which is correlated to the number of nodes in the network) and the frequency at which these updates are being sent. The frequency of updates is in turn dependent on the HRR and the VRR. A WAND network was simulated using randomized event occurrences and for varying VRR and HRR rates. The graph in Figure 1 shows the test results obtained, showing a linear correlation between cache refresh rates and number of messages. Latency in the network is indicative of propagation delays involved while sending cache updates. It is a function of network topology and geographical distribution of the nodes involved in the cache update. Test results

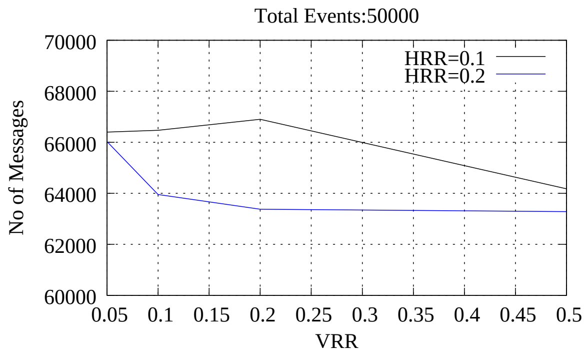

Figure 1: Number of messages plotted against different values of VRR for a given number of events

showed a roughly linear growth in the delays as the height of the worst-case tree increased. A WAND network was simulated using randomized event occurrences and the number of fresh and stale reads occurring with changing VRR were tabulated. Figure 2 shows a graph depicting the fraction of fresh reads in a set of reads for two different simulation runs. As shown in the figure, the percentage of fresh reads remained above 75% even when vertical caches were updated only after 80% change.

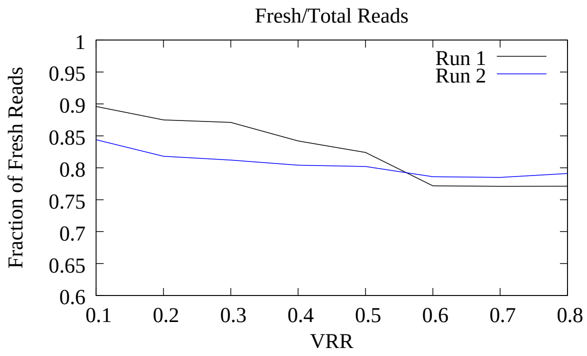

Figure 2: Fresh and Stale Reads

5. CONCLUSIONS

This report presents WAND as a new meta-data management system for large P2P networks. Resiliency of the WAND network depends upon virtual folders and its algorithm to deal with network partitions. There are however, a number of open issues. One of the most challenging ones include merging a split WAND tree after a partitioned network becomes connected.

6. REFERENCES

[1] A. Bhatia, S. Mukherjee, S. Mitra, and S. Srinivasa. Algorithms in wand. Technical Report OSL-IIITB-0501, IIIT Bangalore, Jan. 2005.

[2] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable Peer-To-Peer lookup service for internet applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.

[1]http://www.planet-lab.org/