

**Semester Project: The Coda Distributed File System**

**Student Name: Liliya Kaplan**

**Course: CS522**

## **1. Introduction**

Distributed file systems have grown in importance in recent years. A distributed file system stores files on one or more computers called servers and makes them accessible to other computers called clients. Advantages of this kind of system are: files are more widely available because many computers can access the server, backup and safety of the information are easier because only the servers need to be backed up, system administration becomes easier when everybody share the same file. But there are can be many problems when designing distributed file system: network bottleneck and server overload can occur, security of data can also be an issue: how can we be sure that a client is authorized to access the information. Server failure can be a big problem because it disables all the clients from accessing the information. The Coda project paid attention to many of these issues and implemented them as a research prototype. Coda is designed to operate in an environment, where many hundreds or thousands of workstations span a complex local area network. It aspires to provide the highest degree of availability possible in such an environment. An important goal is to provide this functionality without significant loss of performance.

The Coda distributed file system was developed by the research group of Professor M. Satyanarayanan in the School of Computer Science at Carnegie Mellon University. It has a lot of features not existing in other systems. In his article "The Coda Distributed File System" Mr. Braam lists the following features:

### **1. Mobile Computing:**

- Disconnected operation for mobile clients
- Reintegration of data from disconnected clients
- Bandwidth adaptation

### **2. Failure Resilience:**

- Read/write replication servers

- Resolution of server /server conflicts
  - Handles network failures which partition the servers
  - Handles disconnection of client's client
- 3. Performance and scalability:**
- Client-side persistent caching of files, directories and attributes for high performance
  - Write-back caching
- 4. Security:**
- Kerberos-like authentication
  - Access control lists (ACLs)
- 5. Well defined semantics of sharing**
- 6. Freely available source code.**

Coda implements application-transparent adaptation in the context of a distributed file system.

Coda is the collective name for the programs and the kernel modules, which make up the Coda file servers and clients. Coda is implemented as a collection of substantial user level programs together with a small kernel module on the client, which provide the necessary Coda file system interface to the operating system. The user level programs comprise Vice, the server, and Venus, the client cache manager. The file server Vice is implemented entirely as a user-level program servicing network requests from a variety of clients.

On the client the kernel module does some caching of names and attributes, but it is mostly there to redirect system calls to Venus. The experience of the Coda project has been that placing complicated code in user-level processes offers tremendous development advantages without incurring unacceptable performance compromises. This is shown in Fig.1