

Efficient File Sharing in Coda File System

Mentor:

Jan Harkes

Team:

Preeti Murthy

Arushi Grover

Prathi Shastry

1 Motivation

Coda File System has multiple features for supporting poorly connected environments:

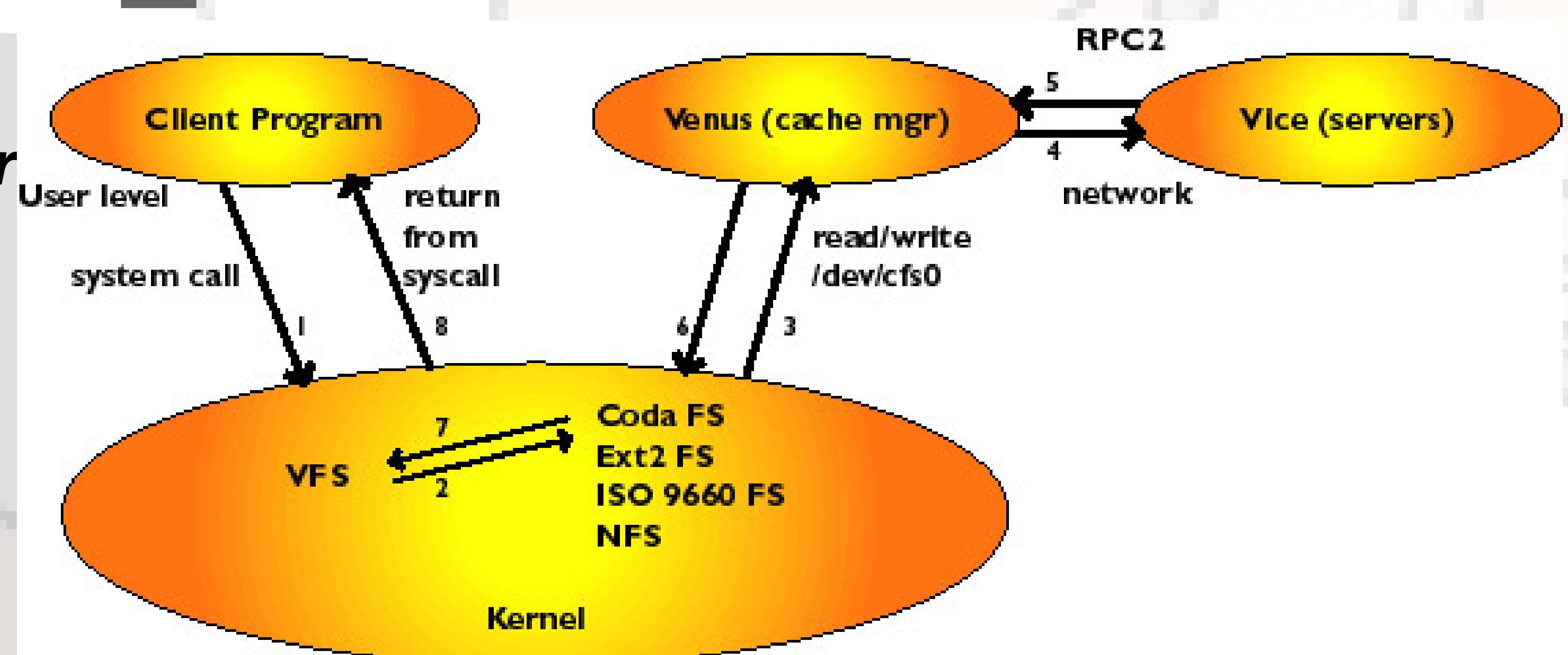
- ▶ *Whole File Caching*
- ▶ *Disconnected Operations*
- ▶ *And much more..*

Suitable for Cloudlets, close to a mobile or user

But, today's workloads involve Large File Sharing (eg. Map Reduce, Video Streaming, etc.)

Whole File Caching becomes inefficient for Large File Sharing.

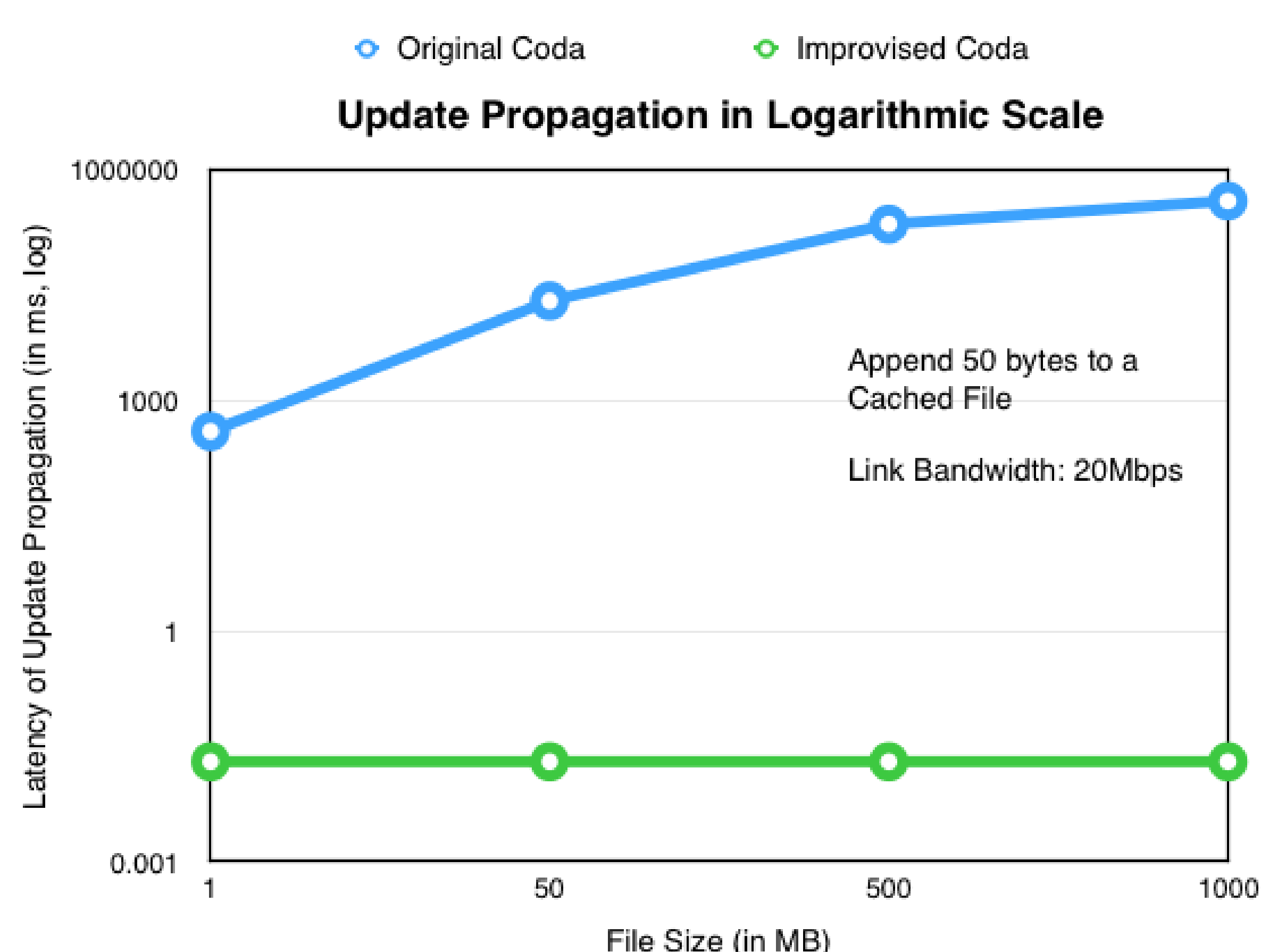
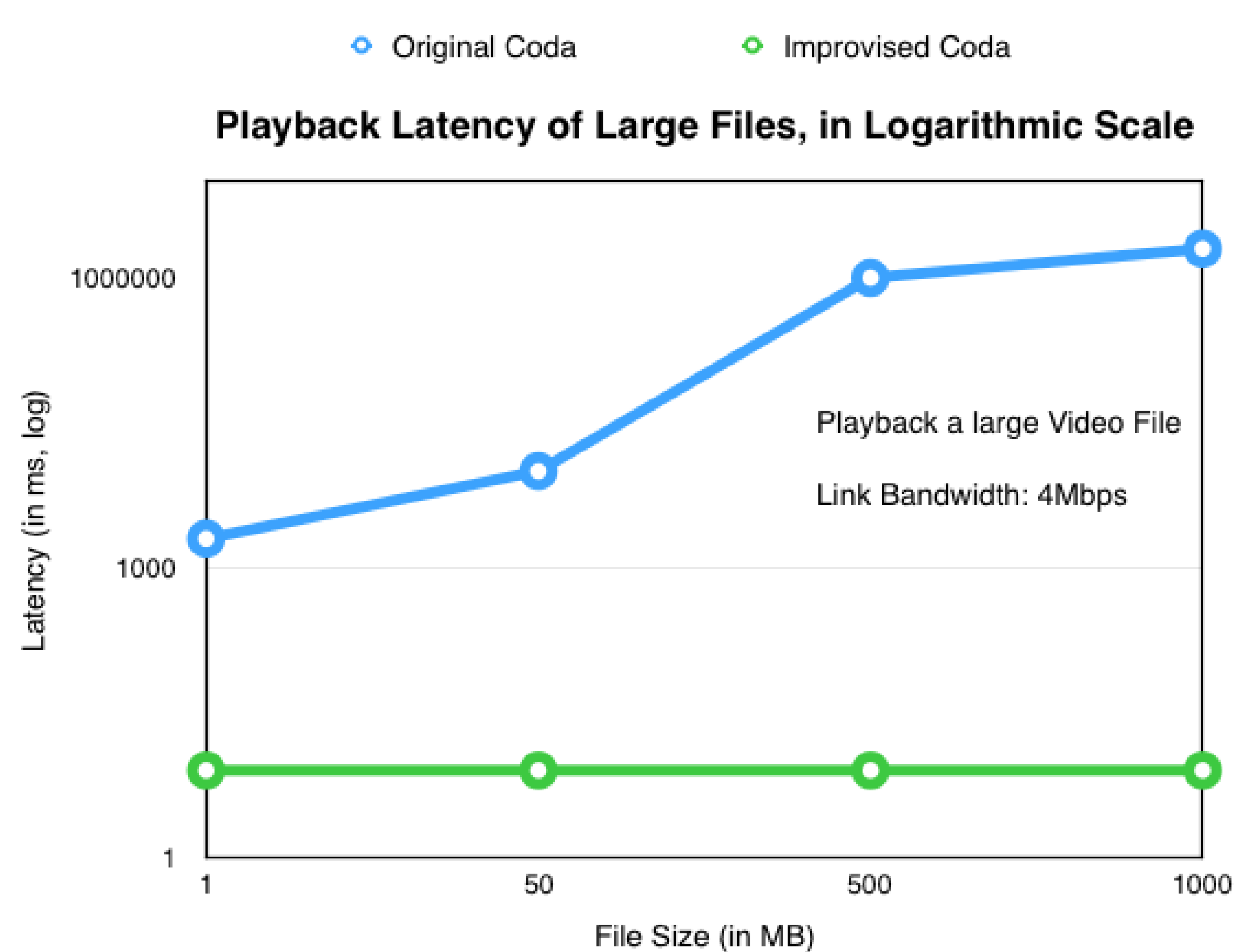
2 Architecture



Our solution for efficient file sharing involved the following implementation steps:

- ▶ *Choose Kernel Modification over FUSE*
- ▶ *Fetch file on Read, instead of Open*
- ▶ *Cache files in Contiguous Chunks*
- ▶ *Write chunks of files to server*

3 Performance Analysis 4 Edge Case Challenges



File Reads:

- ▶ *Reads at the tail end of a file*
- ▶ *Random reads from a file*
- ▶ *Contiguous reads from a file*

File Writes:

- ▶ *Overlapping write log entries*
- ▶ *Creation of new files*

5 Future Work

- ▶ *Backward Compatibility*
- ▶ *Behavior in Disconnected Operation*
- ▶ *Chunking of files for efficient reads*
- ▶ *Prefetching to reduce read latency*
- ▶ *Portability*