

**A Study on High Throughput Large File
Sharing System for a Global Environment
and its Applications**

August 2017

Daisuke Ando

主 論 文 要 旨

No.1

報告番号	○甲 乙 第	号	氏 名	安藤 大佑
主論文題名： A Study on High Throughput Large File Sharing System for a Global Environment and its Applications (広域環境における大容量ファイルの高速共有システムとその応用に関する研究)				
* (内容の要旨) マルチメディアデバイスの発展とネットワーク・クラウド技術の発展に伴い、Full HD や UHD といった高精細映像コンテンツを誰でも簡単に作成できるようになってきた。しかしながら、現在においても、インターネットを利用して大容量ファイルを低ストレージコストかつ高速に世界規模で共有することは難しい。本論文では、高速かつ低コストな大容量ファイル共有を世界規模で実現する Content Espresso の設計・実装および Content Espresso を利用した二つの応用事例を示す。 1 章では本論文の背景として、高精細映像を取り扱うデジタルコンテンツ制作業界の一般的な制作ワークフローを例に示しながら本論文が想定する環境における大容量ファイルの高速共有の難しさについて述べ、さらに本論文の貢献と本論文の構成について言及している。 2 章では Content Espresso の設計について言及している。Content Espresso の基本的なメカニズムは 1) FEC を利用してあらかじめファイルに冗長データを付与する、2) ファイルと冗長データをそれぞれチャンクに分割し、分散設置されている Chunk Server へ保存する、3) ファイル取得の際は、保存されている全てのチャンクを UDP で直接クライアントに送信する、4) Chunk Server の故障やチャンク伝送中のパケットロス等でクライアントが受信できなかったチャンクは、FEC を利用してクライアント側で回復することである。この基本メカニズムを元に Content Espresso は設計されており、Content Espresso が Client、File Manager、Storage Allocator、Chunk Generator、Cluster Head、Chunk Server から構成されていることを述べている。 3 章では Content Espresso の実装について言及している。Content Espresso のそれぞれのモジュールの動作について図を示しながら詳細に言及している。また、Content Espresso の要となる Chunk Generator における FEC 付加の処理及び Client におけるチャンクの受信と FEC による消失パケットの回復処理についてはスレッド構造を示しより詳細に言及している。 4 章では Content Espresso の評価について言及している。Chunk Server 72 台を含む 79 台の実機を利用した計測環境を準備し、ネットワーク遅延やパケットロスがある世界規模のネットワークを想定した環境において、ファイル取得が高速に実現出来ることを示し、また、同様の条件下で生じるファイル書き込みの所要時間変化や、システムの可用性評価、メタデータサーバへの同時アクセス数の評価などを実施し、Content Espresso が広域環境で高速にファイル共有ができることを示している。 5 章では Content Espresso の応用事例の一つ目として非圧縮 UHD 映像受信再生システム Demitasse の設計と実装について言及している。Demitasse を利用する事でフレーム分割され Content Espresso に保存された非圧縮 UHD 映像ファイルをクライアント側で取得しながら 30fps で再生可能であることを示している。 6 章では二つ目の Content Espresso の応用事例として、遠隔コラボレーションシステム SAGE2 におけるファイル共有性能向上に言及している。Content Espresso を SAGE2 のファイル共有用ストレージとして組み込むことで、RTT が 10ms 以上の遠隔コラボレーション環境においてファイル共有性能を向上させることに成功したことに言及している。 7 章では Content Espresso に関する関連研究を提示し、Content Espresso と他研究との関連性について、通信プロトコル、ストレージシステムの両方の点から論じ、Content Espresso の位置づけを明確なものとしている。加えて、Content Espresso が利用している FEC 符号の関連研究とパケット損失モデルの関連研究にも言及している。 最後に 8 章で本論文のまとめを述べ、さらに、本研究の今後の展開について述べている。				

Thesis Abstract

No. _____

Registration Number	<input checked="" type="checkbox"/> "KOU" <input type="checkbox"/> "OTSU" No. _____ *Office use only	Name	Daisuke Ando
Thesis Title A Study on High Throughput Large File Sharing System for a Global Environment and its Applications			
Thesis Summary <p>Large file sharing using the Internet among people and organizations around the world has now become widespread in a variety of industries, with the rapid growth of multimedia devices, networking technologies, and cloud technologies. For example, video production companies usually share source video files with post-production companies and contract out post-production processes like video editing for low cost content creation. They need to share large numbers of large files at low cost and high throughput regardless of network conditions such as network delays and packet loss rates. Although low cost and high throughput file sharing on the Internet is in high demand, it remains challenging to achieve because of the TCP-based protocols usually employed to deliver files, such as HTTP and FTP. TCP performance degrades when the network delay between a server and a client is significant and/or the packet loss rate is high. While creating caches of files using a content distribution network (CDN) improves file sharing performance, it comes with a high storage cost. This thesis proposes Content Espresso, a large file sharing system for a global environment, and its applications.</p> <p>Chapter 2 describes the design of Content Espresso. The fundamental mechanism of Content Espresso is as follows. 1) Redundancy data is appended by FEC to the original file, 2) the original file and redundancy data are split into chunks and stored to the globally dispersed storage servers called Chunk Servers, 3) stored chunks are delivered to the client by using UDP as requested, 4) non-received chunks because of storage failure or network packet loss can be recovered by FEC in the client. Content Espresso is designed using the fundamental mechanism and to satisfy the following requirements: Chunk Servers are built using inexpensive commodity servers, the system can configure required availability level to each file, the system manages metadata and access control information inside the organization of the file owner, and the system can securely share a large number of files among organizations.</p> <p>Chapter 3 describes the implementation of Content Espresso. Content Espresso is composed of File Manger, Storage Allocator, Chunk Generator, Cluster Head, and Chunk Server. This chapter explains the detail of thread structure of each module and the procedure of handling the messages.</p> <p>Chapter 4 evaluates Content Espresso by installing on 79 physical machines including 72 Chunk Servers. The result of the evaluation confirms that Content Espresso can achieve high throughput large file sharing regardless of the network condition.</p> <p>Chapter 5 mentions Demitasse, which is a network-oriented uncompressed UHD video playback system. Demitasse succeeded to retrieve uncompressed UHD video frame files from Content Espresso and playback it with 30fps.</p> <p>Chapter 6 improves the file sharing performance of SAGE2, which is a web-based remote collaboration system by using Content Espresso. Content Espresso improves the file sharing performance of SAGE2 in case that the RTT to the remote site is larger than 10ms.</p> <p>Chapter 7 clearly shows the differences between Content Espresso and existing systems. It compares them from the viewpoint of the transport layer protocol and as a distributed storage system.</p> <p>Finally, chapter 8 summarizes the thesis, outlines the limitations of Content Espresso, and explores Content Espresso's implications and impact.</p>			