

プロジェクトの納期と進捗にもとづく優先ユーザの決定によるファイル送信時間の増加の抑制

三上 智徳¹ 高橋 風太² 串田 高幸¹

概要: 映像制作会社の社員は、ファイルの共有やバックアップのためにファイルサーバへファイルを送信する。映像制作会社には、複数の社員が居るため各社員はいつファイルを送信するかが分からない。その際に、複数の社員のファイル送信が重なるケースがある。課題は、複数の社員のファイル送信が重なることによる各社員のファイル送信時間の増加である。本稿の提案手法では、プロジェクトの納期から作業日を引くことで残り日数を算出する。その後、残り日数と作業日数を用いた計算を行うことで現時点での理想のプロジェクトの進捗度を算出する。終了したタスク数と全タスク数の割り算で算出する現時点でのプロジェクトの進捗と算出された理想のプロジェクト進捗度の差を求めることで優先度を決定する。それをもとに、優先度が低いファイルの送信を中断することで優先度が高いファイルの送信時間の増加を抑制する。課題を立証する実験結果として、14GBの動画ファイルにおいて1人でファイルを送信した際が121秒、課題のケースでは195秒となり、ファイル送信時間が約61%増加した。

1. はじめに

背景

データ量の増加に伴い、データ損失の被害もこれまで以上に大きくなっている [1]。損失の被害の例として、操作エラー、セキュリティ攻撃、ハードウェア障害がある [2]。そのため、データのバックアップは自然災害、人為的事故によるデータ損失を防ぐために必要不可欠である [3]。また、災害時にメインのデータが失われた場合でもバックアップ側からデータを復元することができる [4]。

例として、人気アニメーションや映像作品の制作を手掛ける東映アニメーションを挙げる*¹。ここでは、コンテンツ数が映画とテレビそれぞれ200本を超えており、コンテンツをベースとしたキャラクター商品の開発、著作権事業も展開している。これらの制作のために使用する画像や動画ファイルは、業務に用いるデータとして社員間で共有する。その際に、ファイルサーバが使用される。

ファイルサーバとは、主にLANの中でそれぞれのメンバーが自由にファイルを保存して、共有できる仕組みに使用されるサーバコンピュータである。使用用途の中で、最も使用される用途はファイルの共有とバックアップであ

る*²。各社員が使用している重要なファイルを各々のデバイスに保存しているだけでは、そのデバイスに障害が起きた場合や間違えて削除、上書きしてしまった際にデータを喪失する恐れがある。その際に、ファイルサーバにバックアップをしておくことで万が一の事態になった場合でもデータを保護することができる [5]。また、マルチメディアが普及している中で映像制作会社は、より多くのコンテンツを迅速に制作する必要がある [6]。その中で、ファイルサーバを使用することでファイルをメンバー間で共有することができるため業務の分担や共同作業をすることができる。

映像制作会社には、制作進行管理職というディレクション業務のうちスケジュール管理に従事する職種が存在する*³。プロジェクトでは、作業の遅れが発生する。その際に、対策を講じなければ人件費がかかったり、納期に間に合わないという問題が発生する。そのため、制作進行管理職の人は、プロジェクトの進行管理、プロジェクトの計画と進捗のズレを是正する。この作業を行う際に、管理ツールや管理ファイルを使用する。

課題

社員は、ファイルの共有、バックアップのためにファイルサーバにファイルを送信する。社員が複数人居る際は、各社員がいつファイルを送信するかが分からない。その際

¹ 東京工科大学コンピュータサイエンス学部

〒192-0982 東京都八王子市片倉町1404-1

² 東京工科大学大学院バイオ・情報メディア研究科コンピュータサイエンス専攻

〒192-0982 東京都八王子市片倉町1404-1

*¹ <https://techtarget.itmedia.co.jp/tt/news/1704/12/news07.html>

*² <https://navi.dropbox.jp/file-server>

*³ <https://www.stock-app.info/media/production-management>

に、複数の社員のファイル送信が重なるケースがある。これにより、各社員のファイル送信時間が増加することが本稿での課題である。図 1 に課題となるケースを示す。

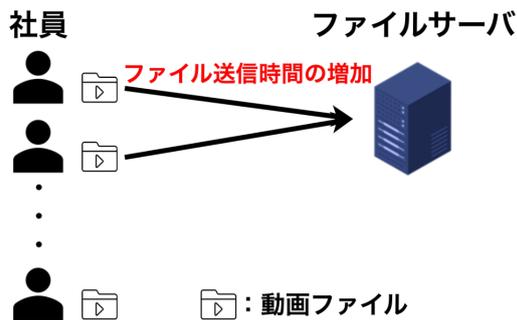


図 1 複数社員のファイル送信が重なるケース

各章の概要

第 2 章では、関連研究について述べる。第 3 章では、本稿の課題を解決するための提案方式について述べる。第 4 章では、提案方式をもとに作成したソフトウェアの実装について述べる。第 5 章では、提案方式の評価と分析、評価実験をするための実験環境について述べる。第 6 章では、本稿の提案方式についての議論を述べる。第 7 章では、本稿のまとめについて述べる。

2. 関連研究

マルチ社員光グリッドネットワークでの大容量ファイル転送の問題に対処するために、4 つの動的スケジューリングアルゴリズム (LWTF, STF, LWATF, SATF) を提案している研究がある [7]。スケジューラは、タスクにネットワークリソースを割り当て、タスクは光ネットワーク上の選択されたルートに沿って宛先ノードに一度にスケジュールされるようにしている。結果として、SATF が平均待ち時間が最も短く、最も効率性が高い結果となり、LWATF が待ち時間の標準偏差が最も低く、最も公平性が高い結果となった。つまり、それぞれのスケジューリングアルゴリズムで待ち時間の軽減と公平性の確保を可能としている。しかし、スーパーコンピュータを使用することを前提としているため課題を解決することができない。

QoS 分類に従って医療データを分類して、有線及び無線ネットワーク上で QoS 対応の医療データを送信するためのテストヘッドを提案している研究がある [8]。バイタルサイン、画像、ビデオの複数の医療データに対して QoS 分類を使用した優先付けを行う。その後、優先付けをもとに帯域幅を制御している。具体的には、データの優先度が同じである場合は帯域幅を均等に振り分けて、データの優先度が異なる場合は優先度の高いデータに対して帯域を割り当てている。結果として、優先度によって帯域幅が変化

するため、その帯域幅の割り当て方によってファイル送信時間が増加した。この研究では、複数の種類のファイルがあることで優先付けをしているため複数の種類のファイルがある環境を前提としている。本稿では、ファイル形式を動画ファイルに絞っているため課題を解決することができない。

待ち時間を短縮してリソース使用率を最大化するために、タスクのグループ化、優先付け、SJF(最短ジョブ優先)を統合したスケジューリングアルゴリズムを提案している研究がある [9]。既存のアルゴリズムのタスクスケジューリングアルゴリズムを改善、処理能力を意識した優先順位付けを行っている。結果として、既存のアルゴリズムと比較して、待ち時間と処理時間が 30%以上減少した。この研究は、クラウドプラットフォーム向けのアルゴリズムを提案している。実際に実験環境にも CloudSim という新しいクラウドコンピューティングインフラストラクチャと管理サービスのモデリング、シミュレーション、実験を実現できる拡張可能なプラットフォームを使用している。本稿では、クラウドプラットフォームを対象としているわけではないため課題を解決することができない。

ホストとネットワークの両方のリソースを考慮した優先順位を意識したリソース配置アルゴリズムである PAVA を提案している研究がある [10]。VM の配置に関してアプリケーションの優先度が分析されて、優先度が高いアプリケーションにリソースが提供されて通常のアプリケーションよりも優先される。また、ホストとリンクの選択アルゴリズムが仮想マシン (VM) とフローを配置する場所を決定する。結果として、ネットワーク送信時間が 52.1%削減されて、応答時間も 38.4%削減された。この研究では、複数のアプリケーションが存在する環境を前提としているため課題を解決することができない。

3. 提案

提案方式

本提案方式では、最初に社員が送るファイルがどのプロジェクトのファイルかという選択からそのプロジェクトの納期と作業日を用いて納期までの残り日数を算出する。その後、算出した残り日数と作業日数を用いて現時点での理想のプロジェクトの進捗度 (%) を算出する。最後に、算出した理想のプロジェクトの進捗度と現時点でのプロジェクトの進捗度を用いて優先度を決定して、優先度が低いファイルの送信を中断するように制御する。こうすることで、社員のファイル送信が同時に行われることがなくなり、優先度が高いファイルの送信時間の増加を抑制する。優先度の決定方法とファイル送信の制御方法の 2 つに分けて説明する。

優先度の決定方法

初めに、制作進行管理職の社員がプロジェクトを管理するための管理ファイルを用意する。図2に管理ファイルの内容を示す。プロジェクト名と開始日、納期、現時点でのプロジェクトの進捗をまとめた内容とする。現時点でのプロジェクトの進捗の値は、以下の式で算出する。以下の式では、終了したタスク数をF、プロジェクトの全タスク数をA、プロジェクトの進捗をDで示す。算出されたDの値は、小数であるため100を掛けることで%の値に変換する。

$$\frac{F}{A} = D (\%) \quad (1)$$

プロジェクトごとの納期と進捗をまとめたファイル

プロジェクト名	開始日	納期	進捗
プロジェクトA	05月01日	05月13日	20%
プロジェクトB	05月04日	05月16日	40%
プロジェクトC	05月02日	05月25日	60%
プロジェクトD	05月03日	05月30日	30%
⋮			
プロジェクトN	NN月NN日	NN月NN日	N%

図2 設定ファイル

図2の設定ファイルの納期の部分と作業日（出勤日）を使用して、納期までの残り日数を算出する。図3に残り日数の算出方法の例を示す。算出するための式は、各プロジェクトの納期 - 作業日（出勤日） = 残り日数である。各プロジェクトをこの式に当てはめると、図3の赤字の部分である残り日数の数字が算出される。

各プロジェクトの納期 - 作業日（出勤日） = 残り日数

始まり 作業日（出勤日） 残り日数 各プロジェクトの納期

05月01日	05月10日	3日	A 05月13日
05月04日	05月10日	6日	B 05月16日
05月02日	05月10日	15日	C 05月25日
05月03日	05月10日	20日	D 05月30日

図3 残り日数の算出方法

残り日数と作業日数を使用して、現時点での理想のプロ

ジェクトの進捗度（%）を算出する。本稿では、日にちごとに均等に進捗が出る環境での算出方法である。残り日数は、今までの工程で算出されているため作業日数を算出する。作業日数は、図2の設定ファイルの納期の部分と開始日を使用して算出する。算出するための式は、納期 - 開始日 = 作業日数である。この段階で残り日数と作業日数が算出されるため、現時点での理想のプロジェクトの進捗度（%）を算出する。算出するための式を以下に示す。以下の式では、作業日数をT、残り日数をL、出力される現時点での理想のプロジェクトの進捗度をRで示す。

$$\frac{T - L}{T} = R (\%) \quad (2)$$

例として、(2)の式にプロジェクトA～Dを代入した式を(3)、(4)、(5)、(6)とする。算出された分数の値を%に直す。%の値に直すと(3)が77%、(4)が50%、(5)が35%、(6)が26%となる。

$$\frac{13 - 3}{13} = \frac{10}{13} \quad (3)$$

$$\frac{12 - 6}{12} = \frac{1}{2} \quad (4)$$

$$\frac{23 - 15}{23} = \frac{8}{23} \quad (5)$$

$$\frac{27 - 20}{27} = \frac{7}{27} \quad (6)$$

理想のプロジェクトの進捗度（%）を算出したため、図2の設定ファイルの現時点でのプロジェクトの進捗を使用して、算出した理想のプロジェクトの進捗度（%）との差を求める。図4に算出された値を示す。ここで算出された値で優先度を決定する。算出された値での優先度の決定は、図4を指標とする。+の値が大きいほど、理想のプロジェクトの進捗から実際の進捗が遅れていることになるため優先度が高くなる。-の値が大きいほど、理想のプロジェクトの進捗よりも実際の進捗が進んでいることになるため優先度が低くなる。今回の一連の例では、優先度が高い順にプロジェクトA、プロジェクトB、プロジェクトD、プロジェクトCとなる。

優先度	高 ↑ +	算出された差の値			優先度	
		差	理想値	優先度		
		プロジェクトA	20%	+57%	77%	①
		プロジェクトB	40%	+10%	50%	②
		プロジェクトD	30%	-4%	26%	③
	低 ↓ -	プロジェクトC	60%	-25%	35%	④

図4 算出された差の値

ファイル送信の制御方法

優先度の決定方法で決まった優先度を使用して、ファイル送信の制御を行う。今回は、優先度が一番高いファイルのみを送信許可として、それ以外のファイルは送信を中断する。この制御を行うことによって、優先すべきファイルを持つ社員のファイル送信時間の増加を抑制する。このファイル送信の制御をする処理の一連の流れを図5に示す。

この図では、社員が3人とファイルサーバが1台でのケースを例としている。まず、社員1がファイル送信を開始する。この時、ファイルサーバ側でファイル送信の検知を行う。社員1がファイル送信している際に、社員2がファイル送信を開始する。ここでも、ファイルサーバ側でファイル送信の検知を行う。この段階で2人の社員が同時にファイル送信をしている状態になるため、ファイルサーバ側で優先度の比較を行う。今回は、社員2の方が優先度が高いため社員2のファイル送信は継続して、優先度が低い社員1のファイル送信は中断される。こうなることで、ファイル送信の競合が無くなり、優先度の高い社員2のファイル送信時間の増加を抑制することができる。社員2のファイル送信が終了したタイミングで、中断していた社員1のファイル送信を再開する。再開した社員1のファイル送信中に社員3がファイル送信を開始する。この際も、2人の社員が同時にファイル送信をしている状態になるため、ファイルサーバ側で優先度の比較を行う。今回は、社員1の方が優先度が高いため社員1のファイル送信は継続して、優先度が低い社員3のファイル送信は中断される。最後に、中断していた社員3のファイル送信を再開する。

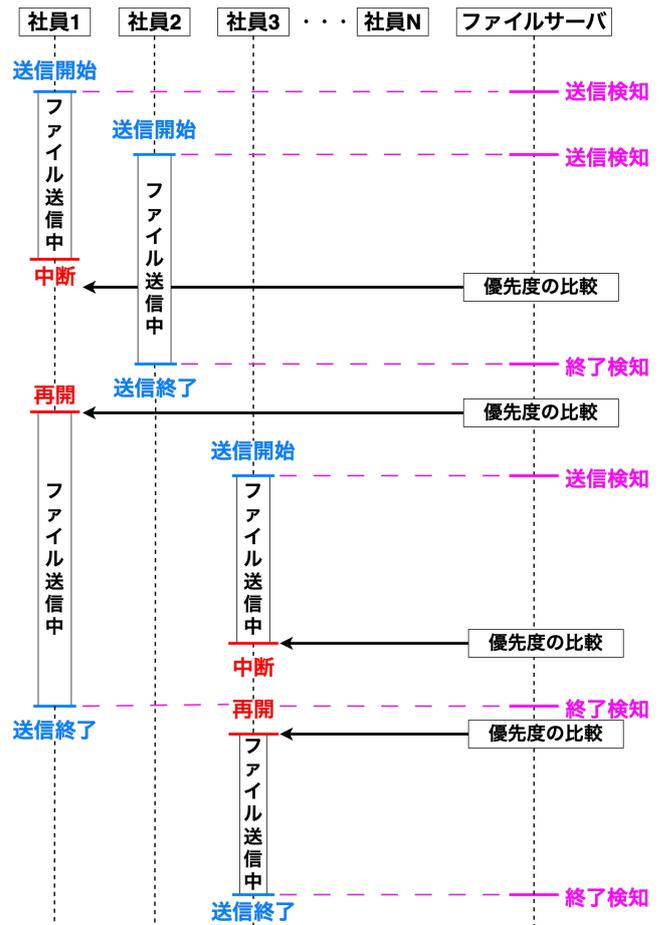


図5 処理の流れ

ユースケース・シナリオ

動画ファイルを扱う映像制作会社をユースケースとして想定する。ユースケースシナリオを図6に示す。動画ファイルの漏洩を防ぐため、映像制作会社は社内サーバを使用する。社員は、ファイルの収容と共有、バックアップのためにファイルサーバに動画ファイルを送信する。動画ファイルのサイズは、10GB以上としている*4。

映像制作会社の社員は複数人居て、各社員がいつファイルを送信するかが分からない。その際に、複数の社員のファイル送信が重なり各社員のファイル送信時間が増加するケースがある。そのため、本稿で提案するソフトウェアを使用することで優先度を決定する。その後、優先度が低いファイルの送信を中断するように制御することで社員のファイル送信が同時に行われることが無くなる。そのため、優先度の高いファイルの送信時間の増加を抑制することができる。

：動画ファイル

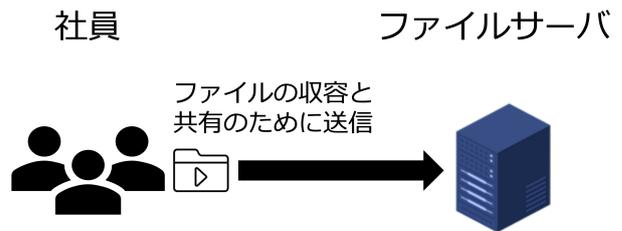


図6 ユースケースシナリオ

4. 実装

提案手法をもとにしたソフトウェアを新たに作成する。プログラム言語のPythonを用いてソフトウェアを作成した。実装の全体構成図を図7に示す。作成するソフトウェアは、優先度を比較するソフトウェアである you-send とファイル送信を制御するソフトウェアである receiver の2つである。また、優先度を比較するために必要なプロジェクトの管理ファイルを作成する。それぞれを分けて説明する。

*4 <https://skeed.jp/case/>

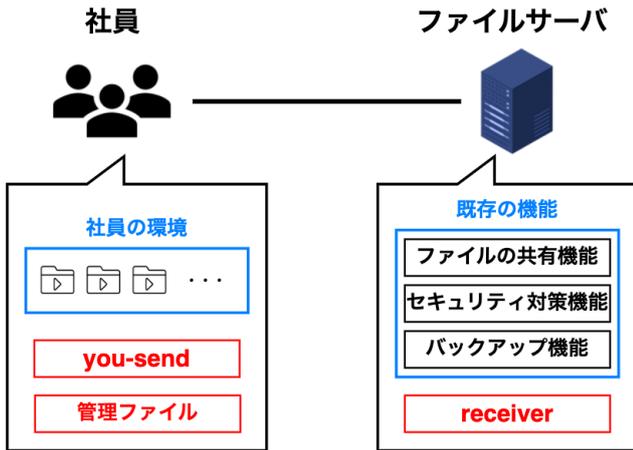


図 7 全体構成図

プロジェクトの管理ファイル

プロジェクトの管理ファイルは、Excel 上で作成した。記載する情報は、各プロジェクト名、各プロジェクトの開始日、各プロジェクトの納期、各プロジェクトの進捗度、作業日の 5 つである。管理ファイルの内容を図 8 に示す。管理ファイルは、各ユーザ側に配置する。

プロジェクト名	開始日	納期	進捗度	作業日
プロジェクトA	5月1日	5月13日	20	5月10日
プロジェクトB	5月4日	5月16日	40	
プロジェクトC	5月2日	5月25日	60	
プロジェクトD	5月3日	5月30日	30	

図 8 管理ファイルの内容

優先度を比較するソフトウェア

図 7 の you-send が優先度を比較するソフトウェアである。図 7 の通り、優先度を比較するソフトウェアは社員側に配置する。処理の流れを以下の①から⑧に示す。処理の流れは、図 9 の赤い数字と関連付いている。

- ① 社員がファイルを送信するためにソフトウェアを実行する
- ② 選択肢を表示 1：ディレクトリに存在するファイルの一覧を選択肢として表示する。その際に、数字と各ファイルを紐づけている。（例：1.test.mov, 2.test.mp4）
- ③ 社員が選択肢を入力 1：送りたいファイルの数字を入力する。（例：test.mov を送りたい場合は 1 を入力）
- ④ 選択肢を表示 2：現在進行しているプロジェクトを選択肢として表示する。その際に、数字とプロジェクト

を紐づけている。（例：1. プロジェクト A, 2. プロジェクト B）

- ⑤ 社員が選択肢を入力 2：表示された選択肢から自分の所属しているプロジェクトの数字を入力する。（例：プロジェクト A に所属している社員は 1 を入力）
- ⑥ 作業日数と残り日数の算出：プロジェクトの管理ファイルのセル情報を使用して算出する。
- ⑦ 優先度の算出：プロジェクトの管理ファイルのセル情報と作業日数、残り日数を使用して算出する。
- ⑧ ファイルの送信開始：scp コマンドを使用して、ファイルサーバに動画ファイルを送信する。

ファイル送信を制御するソフトウェア

図 7 の receiver がファイル送信を制御するソフトウェアである。図 7 の通り、ファイル送信を制御するソフトウェアはファイルサーバ側に配置する。処理の流れを以下の①から⑤に示す。処理の流れは、図 9 の青い数字と関連付いている。

- ① 社員からのファイル送信を検知：2 回以上検知した場合、優先度の比較が必要になるため次の処理に進む。
- ② 優先度の比較：社員からファイルが送信された際に、社員が選択肢したプロジェクトの優先度の数値が送られてくるためその数値を比較する。
- ③ ファイル送信の中断：優先度が低かったファイルの送信を中断する。
- ④ ファイル送信終了の検知：ファイルサイズが変わらないことを条件にして、ファイル送信終了を検知する。
- ⑤ ファイル送信の再開：中断されたファイル送信を再開する。

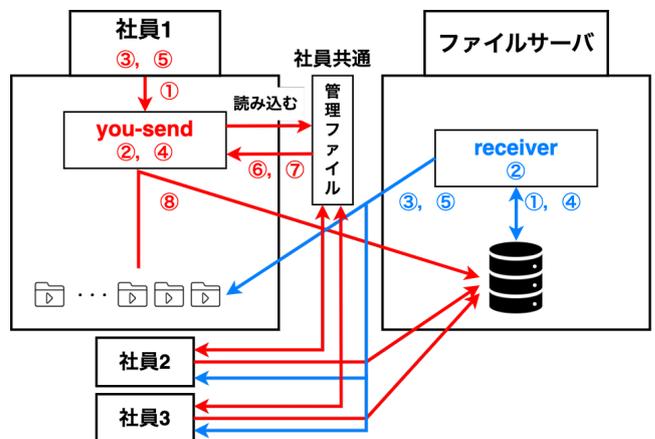


図 9 一連の処理の流れ

5. 実験

本稿の課題となる際（複数社員のファイル送信が同時にあったケース）のファイル送信と社員が1人でファイル送信をした際の2つのファイル送信時間を比較する。

実験環境

実験環境は、各社員とファイルサーバ共に仮想マシン（VM）を用いる。今回、社員の人数は2人の場合と3人の場合を実験する。各社員は、10GB以上の動画ファイルをファイルサーバへ送信する。ファイルを送信するために scp コマンドを使用する。また、時間を計測するために time コマンドを使用する。以下に、VMの構成要素と図10に実験環境を示す。

- VM 構成情報（社員1人目）
OS : Ubuntu-22.04
vCPU : 2 コア
RAM : 8GB
HDD : 100GB
- VM 構成情報（社員2人目）
OS : Ubuntu-22.04
vCPU : 2 コア
RAM : 8GB
HDD : 100GB
- VM 構成情報（社員3人目）
OS : Ubuntu-22.04
vCPU : 2 コア
RAM : 8GB
HDD : 100GB
- VM 構成情報（ファイルサーバ）
OS : Ubuntu-22.04
vCPU : 2 コア
RAM : 8GB
HDD : 200GB

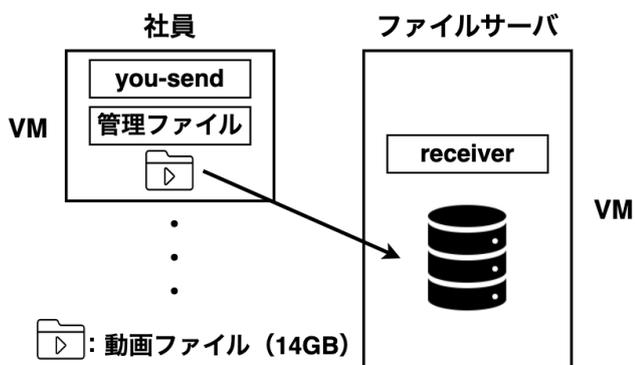


図 10 実験環境

実験結果と分析

課題のケース（複数の社員のファイル送信が重なる場合）の社員数が2人の場合と3人の場合の結果を分けて説明する。社員が1人でファイル送信をした際の時間と課題のケースである社員数が2人の場合と3人の場合のファイル送信時間の3つを比較した結果を図11に示す。横軸がファイルを送信する社員数を示している。実験回数は社員が1人でファイル送信をした際と課題のケースをそれぞれ10回行い、平均を取った。

社員数が2人の場合

社員数が2人の場合は、社員数が1人の場合と同一のユーザ（VM）のファイル送信時間を対象としている。実験の結果、社員が1人でファイル送信をした際が2分1秒（121秒）、社員が2人でファイルを送信した際が3分15秒（195秒）であった。

図11より、社員がファイルを送信する際に社員数が2人の場合は社員数が1人の場合と比較してファイル送信時間が1分14秒（約61%）増加している。したがって、この結果から複数社員のファイル送信が重なる場合、社員のファイル送信時間が増加することが示されている。

社員数が3人の場合

社員数が3人の場合も社員数が1人の場合と同一のユーザ（VM）のファイル送信時間を対象としている。実験の結果、社員が3人でファイル送信をした際は5分10秒（260秒）であった。

図11より、社員がファイルを送信する際に社員数が3人の場合は社員数が1人の場合と比較してファイル送信時間が3分9秒（約156%）増加している。また、社員数が2人の場合と比較してもファイル送信時間が59秒（約23%）増加している。したがって、この結果から複数社員のファイル送信が重なる場合の中でも社員数が増えたと社員のファイル送信時間が増加することが示されている。

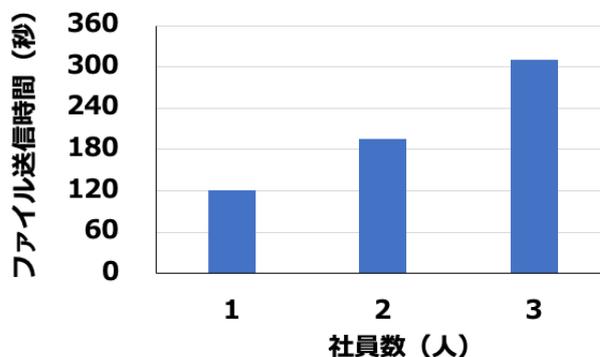


図 11 社員数が3人の場合の基礎実験結果

6. 議論

本稿の理想のプロジェクトの進捗度の算出方法は、日にちごとに均等に進捗があることを前提にして算出されている。実際は、プロジェクトの進捗は均等ではなく工程があり、各工程によって時間がかかる（日にちがかかる）場合やすぐに作業が終わる（日にちがかからない）場合がある。そのため、各工程ごとに今の提案方式を使用することで工程ごとの進捗で優先度を決定することができるため工程ごとに要する日付の長さの違いによるプロジェクトの進捗の誤差を軽減することができる。また、プロジェクト管理ツールやサービスと紐付けることでも改善可能である。

また、プロジェクトによっては最初に指定した納期に間に合わずに納期が延びる場合がある。その際に、本稿の提案方式では納期までの残り日数も優先度の値を算出する際に使用しているため最初の納期に間に合わずに納期が延びているプロジェクトの優先度が下がるケースが考えられる。しかし、最初に指定した納期に間に合わずに納期が延びている時点で優先度を下げたはけけない。そのため、このような状況に該当したプロジェクトは優先度を下げないためにラベル付けを行うことにより、ラベルが付いているプロジェクトの優先度を下げずにファイルを送信できるように制御することによって改善可能である。

本稿のファイル送信の制御方法だと優先度が低くなったファイルの送信は中断される。そのため、優先度が低いファイルは複数社員が同時にファイルを送信した際に断続的に中断されて、ファイル送信時間がより増加してしまう。これは、優先度が低いファイルの送信を中断するのではなく、優先度が低いユーザがイライラしない送信時間でファイルを送信できる帯域幅を確保して、それ以外の帯域幅を優先度が高いユーザに振り分けることによって改善可能である。しかし、優先度が高いファイルの送信時間は本稿よりも増加する。

7. おわりに

社員が複数人居る際は、各社員がいつファイルを送信するかが分からない。そのため、複数の社員のファイル送信が重なるケースがある。ここでの課題は、複数の社員のファイル送信が重なることによる各社員のファイル送信時間が増加することである。提案では、プロジェクトの納期と進捗度を用いて優先度を決定して、優先度が低いファイルの送信を中断するように制御することで優先度が高いファイルの送信時間の増加を抑制する。課題を立証する実験の結果として、14GBの動画ファイルにおいて1人目のファイル送信時間が約61%増加した。

参考文献

- [1] Xia, R., Machida, F. and Trivedi, K.: A Markov Decision Process Approach for Optimal Data Backup Scheduling, *2014 44th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 660–665 (online), DOI: 10.1109/DSN.2014.100 (2014).
- [2] Maher, R. and Nasr, O. A.: DropStore: A Secure Backup System Using Multi-Cloud and Fog Computing, *IEEE Access*, Vol. 9, pp. 71318–71327 (online), DOI: 10.1109/ACCESS.2021.3078887 (2021).
- [3] Xia, R., Yin, X., Alonso Lopez, J., Machida, F. and Trivedi, K. S.: Performance and Availability Modeling of IT Systems with Data Backup and Restore, *IEEE Transactions on Dependable and Secure Computing*, Vol. 11, No. 4, pp. 375–389 (online), DOI: 10.1109/TDSC.2013.50 (2014).
- [4] Tamimi, A. A., Dawood, R. and Sadaqa, L.: Disaster Recovery Techniques in Cloud Computing, *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pp. 845–850 (online), DOI: 10.1109/JEEIT.2019.8717450 (2019).
- [5] Zhang, J. and Li, H.: Research and Implementation of a Data Backup and Recovery System for Important Business Areas, *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Vol. 2, pp. 432–437 (online), DOI: 10.1109/IHMSC.2017.209 (2017).
- [6] Mateus, J., Malheiro, F., Cavaco, S., Correia, N. and Jesus, R.: Video annotation of TV content using audiovisual information, *2012 International Conference on Multimedia Computing and Systems*, pp. 113–118 (online), DOI: 10.1109/ICMCS.2012.6320188 (2012).
- [7] Hu, M., Guo, W. and Hu, W.: Dynamic Scheduling Algorithms for Large File Transfer on Multi-user Optical Grid Network Based on Efficiency and Fairness, *2009 Fifth International Conference on Networking and Services*, pp. 493–498 (online), DOI: 10.1109/ICNS.2009.37 (2009).
- [8] Lee, J.-M., Lee, J.-H. and Chung, T.-M.: Experimental QoS test of Medical Data over Wired and Wireless Networks, *2008 10th International Conference on Advanced Communication Technology*, Vol. 1, pp. 142–146 (online), DOI: 10.1109/ICACT.2008.4493730 (2008).
- [9] Ru, J. and Keung, J.: An Empirical Investigation on the Simulation of Priority and Shortest-Job-First Scheduling for Cloud-Based Software Systems, *2013 22nd Australian Software Engineering Conference*, pp. 78–87 (online), DOI: 10.1109/ASWEC.2013.19 (2013).
- [10] Son, J. and Buyya, R.: Priority-Aware VM Allocation and Network Bandwidth Provisioning in Software-Defined Networking (SDN)-Enabled Clouds, *IEEE Transactions on Sustainable Computing*, Vol. 4, No. 1, pp. 17–28 (online), DOI: 10.1109/TSUSC.2018.2842074 (2019).