

分散型 e-Learning システムにおける ノード離脱時のファイル複製数維持手法の検討

新谷 渉^{*†}, 東野 正幸[†], 川戸 聡也^{†‡}, 高橋 健一[†], 川村 尚生[†]
([†]鳥取大学, [‡]米子工業高等専門学校)

A Study of the Method of Keeping the Number of Files' Replicas when a Node Leaving on a Distributed e-Learning System

Wataru Shintani^{*†}, Masayuki Higashino[†], Toshiya Kawato^{†‡}, Kenichi Takahashi[†], Takao Kawamura[†]
([†]Tottori University, [‡]National Institute of Technology, Yonago College)

1 はじめに

我々は複数の一般的なコンピュータを組み合わせることで安価に運用可能な分散型 e-Learning システムを開発している。本システムを構成する各々のコンピュータは、分散ハッシュテーブルに基づく分散データベースのノードとなり、システム内の個々のノードとファイルのメタ情報を分散管理している。ファイルの実体の複製を複数のノードに分散させて保存し、そのファイルの保存先のノードの場所をメタ情報として分散データベースで管理することで、一部のノードが故障しても他のノードが保持しているファイルの複製を利用可能となり、高い可用性を実現できる。組織が保有するパーソナルコンピュータ (PC) には遊休の計算資源が存在する。我々はこの遊休資源を持つ PC (遊休 PC) を分散型 e-Learning システムのノードの一部として利用する手法を検討している [1]。

しかし、他の用途で使用中の PC をノードとして利用する場合、その PC を制御する主体は分散型 e-Learning システムの管理者ではなく PC の使用者となることから、管理者が意図しないタイミングでの PC の使用者によるシステムからの離脱を許容する必要がある。しかし、これを許容した場合、PC の使用者が任意のタイミングでオペレーティングシステム (OS) のシャットダウンやサスペンド及びネットワーク接続の切断を行った際に、冗長性を確保するために保持していたファイルの複製数がノードの離脱とともに減少する問題がある。

そこで本稿では、ファイルの複製の保持数を維持するために、ノードの離脱が発生した際に、離脱理由に応じて適切にファイルの複製数を維持する手法を検討する。

2 ノードの離脱への対応

2.1 ノードの離脱の分類

ノードの離脱理由として OS のシャットダウンやサスペンド、及びネットワーク接続の切断などが挙げられる。PC の利用者がこれらの操作を行う場合、離脱するノードが保持するファイルを他のノードに退避できる場合とできない場合がある。また、離脱するノードが他のノード群に離脱の意志を宣言できる場合とできない場合もある。例えば離

脱するノードの PC の利用者がファイルの転送完了を待つ余裕がない場合や OS の強制シャットダウンのように他のノード群と通信する余地がない場合が考えられる。ファイルの複製数を効率的に維持する場合、これらの可否の組み合わせに応じて適切な処理が必要であると考えられる。

そこで、ノードが離脱する際に他のノードに離脱する意志を通知することを離脱の宣言とよび、ノードが離脱する際にそのノードが保持していたファイルを他のノードへ退避することをファイルの自己複製とよぶこととし、これらの処理の有無の組み合わせに応じたファイルの複製数の維持方法を検討する。

2.2 ノードの離脱の検知

離脱理由が OS のシャットダウンで PC の利用者がシャットダウンの完了を待つことができる場合は、ノードは離脱の宣言とファイルの自己複製を行う余地がある。しかし、PC の強制的な電源の切断や OS の強制シャットダウンの場合はこれらを行えないため、他のノードがこれを検知する必要がある。そこで、システムに参加しているノード群は IP アドレスの辞書順にリング状のネットワークを維持しておき、リング上で通信が行えなくなったノードは宣言を行わずに離脱したものと判断する。参照コストが伴うもののノード群の IP アドレスは分散データベースで管理しているため、それぞれのノードは分散データベースを参照することでノード間のネットワークを修復できる。

2.3 ノードの離脱におけるファイルの複製数の維持

離脱するノードが可能な処理は離脱する理由によって 3 段階に分類できる。1 段階目はノードが離脱の宣言とファイルの自己複製の両方の完了を待てる場合である。2 段階目はノードが離脱の宣言のみの完了を待てる場合である。ファイルの自己複製はノードが保持しているファイルのデータサイズに応じて転送時間が長くなってしまいが離脱の宣言のみであれば短時間で済む。また、他のノード群も離脱の宣言を受けることで他のノード群が素早く協力してファイルの複製数を回復させることができる。3 段階目はノードが離脱の宣言もファイルの自己複製も行わないので、リング状のネットワークの切断からノードの離脱を検知し、検知直後にノード群が協力してファイルの複製数を回復させる (イベント駆動型)。しかし、回復中にさらに離脱

が発生して正常に複製数が回復しない可能性がある。そこで、定期的に各ノードで保持するファイルの複製数を確認し、既定数に満たないファイルの複製数を回復する（ポーリング型）。管理者としては 1 段階目が望ましいが、遊休 PC の利用者が行う操作を強制できないため、2 段階目と 3 段階目の場合が多くなると考えられる。

2.4 離脱したノードの再参加におけるファイルの複製の扱い

遊休 PC は使用者の操作により PC の再起動やサスペンド及びネットワークの一時的な切断などの理由により離脱と参加を繰り返す場合が考えられる。このような一時的な離脱の場合、再参加の際にそのノードが保持していたファイルを再利用することで通信量や処理時間の削減により再参加の処理を効率化できる。ノードが離脱の宣言を行わずに離脱を行う場合、ノードが離脱している期間中に他のノード群がファイルの複製数の回復を行うことで、既定の複製数の維持処理が完了していることが考えられる。この場合、離脱していたノードは再参加してもそのノードが保持していたファイルの複製はどこからも参照されないため、その時点では不要なものとなるが、キャッシュとして活用できる。また、遊休 PC の使用者による短時間の OS のサスペンドやネットワークの瞬断のように、ノードの離脱時間が短時間で済む場合は、他のノード群がファイルの複製数を回復しなくても離脱したノードが単に再接続するだけで再参加の処理が済む場合が考えられる。この場合は、既定の複製数が維持されていることだけを確認すれば良い。

3 実験と考察

実験環境としてノードとして使用する PC にはスペックが CPU: Intel Core i5-6500, RAM: 16 GB, SSD: 240 GB, OS: Debian 9.11 のものを 4 台使用した。また、システムで維持するファイルの複製数は 3 個とした。

はじめに、ノードが離脱する際に自己複製を行う場合と行わない場合とで複製数の維持に要する処理時間にどのような違いが生じるかを調べる実験を行なった。実験では、10 MiB のファイルの個数を 5 個ずつ変化させて複製数の維持に要する処理時間を計測した。図 1 に実験結果を示す。ファイルの個数を変えた場合、自己複製を行う場合は行わない場合に比べて処理時間が短くなり効率的であることを確認した。しかし、処理時間の差は小さいことに加えて、遊休 PC の使用者に自己複製が完了するまでの待ち時間が必要になることを考慮すると遊休 PC の離脱時の自己複製の必要性は低いと考えられる。

次に、ファイルの複製数の維持においてポーリング型とイベント駆動型の処理の特徴を調べるため、1 個のファイルのサイズを 50 MiB 間隔で変化させて経過時間を計測した。実験結果を図 2 に示す。ポーリング型はイベント駆動型に比べてファイルの数が処理時間へ大きな影響を与えることが分かった。

最後に、ノードの再参加におけるファイルの再利用の特徴を調べるため、1 個のファイルのサイズを 50 MiB 間隔で変化させて処理時間を計測した。図 3 よりファイルの再利用はファイルのサイズによる影響を受けないことが分かる。

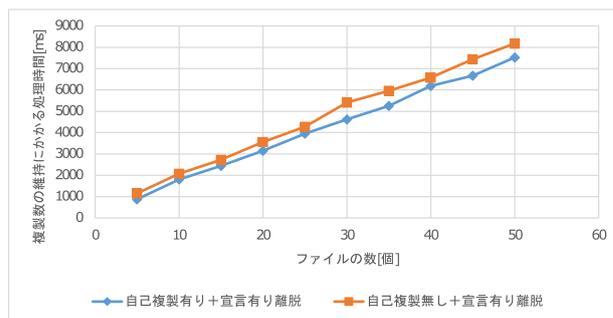


図 1 自己複製の有無による複製数の維持にかかる処理時間の比較

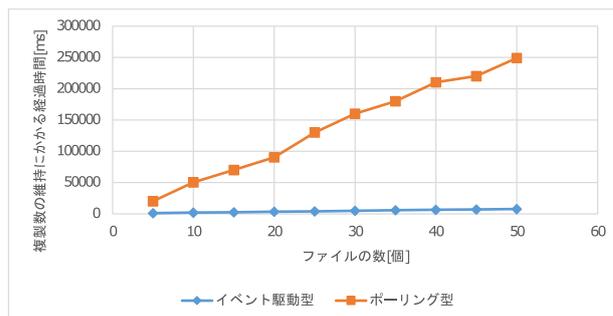


図 2 処理方式による複製数の維持にかかる経過時間の比較

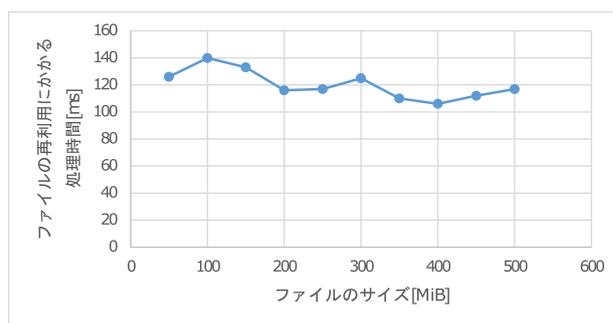


図 3 ファイルの再利用にかかる処理時間

これにより、ファイルのサイズを再利用の判断基準とすることでより効率的に計算資源を利用できると考えられる。

4 おわりに

本稿では、ファイルの複製の保持数を維持するために、ノードの離脱が発生した際に、離脱理由に応じて適切にファイルの複製数を維持する手法を検討した。

謝辞

本研究は JSPS 科研費 JP19K03081 の助成を受けたものである。

参考文献

- [1] Toshiya Kawato, Masayuki Higashino, Kenichi Takahashi, and Takao Kawamura. Proporsal of distributed e-learning system using idle resources. In *Proceedings of the 5th International Conference on Computer and Communication Systems*, pp. 557–561, 2020.