

遊休計算機を用いた分散システムのための負荷テスト環境の構築

永島 優也[†] 東野 正幸[‡] 高橋 健一[§] 川村 尚生[¶]
 鳥取大学[†] 鳥取大学[‡] 鳥取大学[§] 鳥取大学[¶]

1 はじめに

近年, DX が推進される中, 多くの教育機関が e-Learning 等の情報システムを導入している. しかし, 専用サーバの導入や運用にかかる費用が高額になりやすい. そこで我々は, 組織内に有る計算資源を余らせている既存計算機(遊休計算機)を組み合わせることで運用可能な分散型 e-Learning システム [1] を開発している. 遊休計算機の特徴として, 本来の計算機の使用者の操作により任意のタイミングで通信回線の接続や切断が発生することが挙げられる. 分散システムでは一般にこれらの不安定な動作は障害として扱われることが多いが, 我々が提案しているシステムでは, これらを正常な動作として許容する必要がある. そのための負荷テスト環境が必要となる. そこで本研究では, 遊休計算機の不安定な挙動をシナリオとして定義して実施可能な負荷テスト環境を提案する.

2 遊休計算機を用いた分散システム

我々は遊休計算機を用いた分散システムとして分散型 e-Learning システムを提案している [1]. 本システムは, システム専用のノードとして使用される計算機(以後, 常駐ノード)と遊休計算機(以後, 遊休ノード)を複数用いて構築される. 本システムにおいて, ファイルは各ノードに分散して保存され, ファイルの保存場所等のメタ情報は常駐ノードを使って構成した分散データベースの Apache Cassandra^{*1}で分散管理される.

各ノードはシステムに対して, **参加**, **宣言有り離脱**, **宣言無し離脱**を行える. 参加とは新規のノードが新しくシステムの一部となることである. 宣言有り離脱とは参加済みのノードが他のノードに離脱することを伝えてからシステムから外れることである. 宣言無し離脱とは他のノードに何も伝えることなく離脱することである. 宣言無しで離脱したノードの検知は常駐ノードが他のノードとの定期的な通信により行っている. 常駐ノードの場合は, システム専用のためシステムの管理者が参加及び宣言有り離脱を行うことを基本とするが, 遊休ノードの場合は, 参加, 宣言有り離脱, 及び宣言無し離脱が任意のタイミングで行われる場合がある.

3 負荷テスト環境の構築

3.1 ノードの参加・離脱制御

遊休計算機を用いた分散システムの負荷テスト環境において, 遊休ノードの振る舞いである任意のタイミングでの参加・離脱を再現するには, 遠隔の計算機からノードの参加・離脱を実行できる必要がある. また, これらの処理をテストコードから制御できる必要がある. そこで, 各ノードの制御用計算機からノードを遠隔で参加・離脱させる API をシステムに実装する.

ノードの参加については, 制御用計算機から既に参加しているノード経由で SSH (Secure Shell) を用いて参加させたいノードに接続して参加処理を実行する.

ノードの宣言有り離脱については, 提案する分散型 e-Learning システムが JVM (Java Virtual Machine) で動作していることから, Java の API である `Runtime.exit()` を実行する API を追加する. この API を用いて終了する場合は, 離脱時にファイナライザが実行され, 常駐ノードに離脱することを伝えるメッセージを送ることで宣言有り離脱処理が実行される.

宣言無し離脱については, Java の API である `Runtime.halt()` を実行する API を追加する. この API を用いて終了する場合は, 離脱時にファイナライザが実行されないため, 常駐ノードにメッセージを送らず宣言無し離脱が実行される.

3.2 シナリオの作成

負荷テストを行う際に利用者の実際の操作手順(シナリオ)に沿ってシステムを操作する必要がある. しかし, アクセス回数を増やす際に, 手動で行うのは限界がある. そのため, 指定したシナリオを自動で指定した回数実行可能とする. Web ベースのシステムにおいてユーザ操作のシナリオを作成する際, ユーザのアカウント情報, アクセス先の URL, HTTP Headers, HTTP Cookie が必要になるが, これらをすべて手動で記録し, テストごとに作成するのは手間を要する. そのため, テストを行うシナリオを実際に 1 回実行し, その際に通信内容を記録するツールを用いて再利用可能なシナリオデータ (JMX ファイル) を作成する.

シナリオデータの作成には Apache JMeter^{*2}を用いる. Apache JMeter の機能である HTTP プロキシサーバを用いてテスト用のユーザの操作のリクエスト内容を記録し, Apache JMeter で利用できるシナリオ形式で保存する.

Construction of a load-testing environment for distributed systems including idle computers

[†] Yuya Nagashima, Tottori University

[‡] Masayuki Higashino, Tottori University

[§] Kenichi Takahashi, Tottori University

[¶] Takao Kawamura, Tottori University

^{*1} Apache Cassandra: <https://cassandra.apache.org/>

^{*2} Apache JMeter: <https://jmeter.apache.org/>

3.3 アクセス負荷の制御

システムに負荷を与える際に、負荷を与える時間とシナリオの実行回数を指定できる必要がある。そのため、保存した JMX ファイルを実行する際に、時間と実行回数を引数として実行する。また、実行回数を増やす際に、負荷を掛ける計算機（以後、アクセス用計算機）がメモリ不足になる可能性がある。そのため、アクセス用計算機を複数台用いて、制御用計算機から並列実行するために parallel-ssh^{*3}を用いる。

3.4 負荷テストの自動化

負荷テストを簡単に行うために、ノードの参加・離脱とアクセス負荷の制御を自動化する必要がある。負荷テストを行うのに必要な情報として、参加・離脱処理を行うテスト開始からの時間とノードの IP アドレス、アクセス先の IP アドレス、シナリオの実行回数と実行時間がある。この情報を設定ファイルとして保存し、アクセス負荷を与え、その際にノードを参加・離脱させる操作を記述したシェルスクリプトを実行する方式で自動化する。

4 実験と考察

構築した負荷テスト環境を用いて実験を行った。実験は、常駐ノード 1 台、遊休ノード 4 台、クライアント制御用計算機 1 台、クライアント用計算機 3 台で行った。常駐ノードには、OS: Debian 11, CPU: Intel Celeron N3350, RAM: DDR4 6GB, SSD: 64GB を用いた。遊休ノードには、OS: Debian 11, CPU: AMD Ryzen 9 4900H, RAM: DDR4 32GB, SSD: 512GB を用いた。

実験するシナリオとしては、分散型 e-Learning システムの利用者が、システムへログインし、教科を選択し、当該教科の授業資料の PDF ファイル (688 KB) をダウンロードする動作を 1 分あたり 3000 セッション分行うものとした。組織内の講習会の資料をメーリングリストなどで一斉通知した場合等を想定している。1 つ目の実験は負荷テスト開始から 30 秒後に 4 台の遊休ノードが参加する場合としない場合を比較して参加が与える影響を調べた。2 つ目の実験は負荷テスト開始から 30 秒後に 4 台の遊休ノードが宣言有り離脱、宣言無し離脱、離脱しない場合を比較して、離脱が与える影響を調べた。

実験結果を図 1 と図 2 に示す。横軸の単位は分: 秒. ミリ秒、縦軸の単位は秒である。図 1 より、遊休ノードが 4 台参加することにより、全てのリクエストの処理が完了するまでの時間（横軸）が約 50 秒減少していることから、全体としては負荷を分散できていることが分かる。しかし、システムの応答時間（縦軸）がノードの参加直後から急激に増加していることから、新しく参加したノードへの処理の割当がオーバーヘッドとして存在することが推察され、性能改善が必要であると考えられる。このことから、提案システムの改善点を本負荷テスト環境により容易に見つけ出すことができた。また、図 2 より、宣言有り離脱と宣

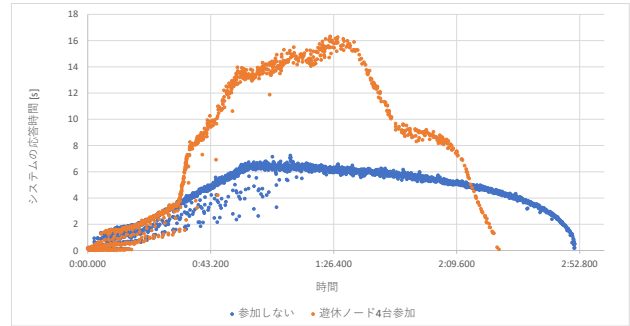


図 1 参加処理の有無の比較

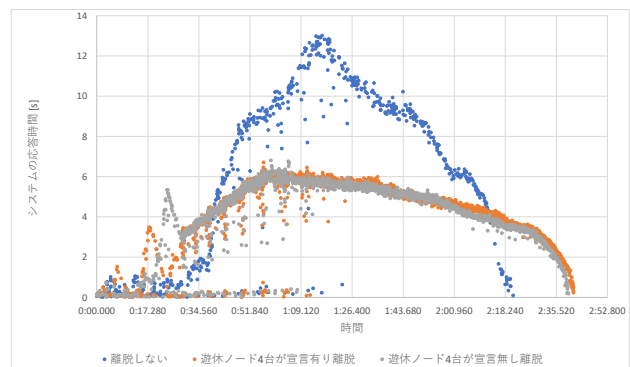


図 2 離脱処理の有無の比較

言無し離脱に大きな違いは無いことから、本実験のシナリオではどちらの離脱方法でも性能に大きな違いが無いことが確認できた。

5 おわりに

本研究では遊休計算機を用いた分散システムにおいて遊休ノードの参加・離脱が任意のタイミングで行われる場合の負荷テストを容易に行うための環境を提案した。この負荷テスト環境を用いることで、遊休計算機の不安定な振る舞いをシナリオとして定義して再現するとともにその際の負荷テストを行うことが容易になる。今後、我々が提案している遊休計算機を用いた分散型 e-Learning システムにおいて、不安定なノードの振る舞いを許容する負荷分散アルゴリズムの改良における有用性を評価する予定である。

謝辞

本研究は JSPS 科研費 19K03081 の助成を受けたものである。

参考文献

- [1] Kawato, T., Higashino, M., Takahashi, K. and Kawamura, T.: Proposal of distributed e-learning system using idle resources, *In Proceedings of The 5th International Conference on Computer and Communication Systems*, pp. 557–561 (2020).

^{*3} parallel-ssh: <https://parallel-ssh.org/>