

OpenPNEのソフトウェアエージングに関する研究

東京都立大学 *近藤和希 KONDO Kazuki
会員 東京都立大学 肖霄 XIAO Xiao

1. 序論

現代では、ソフトウェアは私たちの身の回りから宇宙規模のシステムに至るまで、非常に長時間の連続稼働が必要なものが多く存在する。そのような連続稼働によって予期しない性能劣化が生じる現象は「ソフトウェアエージング」と呼ばれ、特に高可用性が要求されるソフトウェアにおいては、ソフトウェアエージングのリスクを定量的に評価する必要性が高まっている。

本研究では、ソフトウェアエージングの研究対象として、SNSシステムの一つであるOpenPNE¹を選択する。動機付けは、X²のようなSNSシステムは、高い拡張性を持ち、膨大なユーザー数を抱えており、企業にとってサーバー運用のリスクが高まっているためである。また、これまでのソフトウェアエージングに関する研究の中で、SNSシステムを対象にしたものは、調査の限らない。本研究では、Measurement-based Approach[1]に基づき、SNSシステムの特徴を踏まえた負荷シナリオを設定し、各シナリオに対して、負荷期間と監視期間[2]を設けることで、SNSシステムにおけるソフトウェアエージングのリスクを調査する。

2. 実験環境

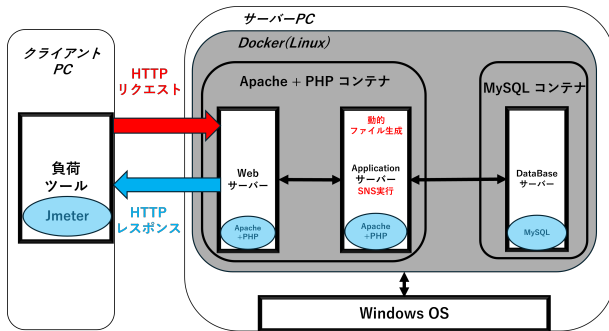


図 1: SNS 運用環境と負荷ツール

図1に示す実験環境は、2台のデバイスで構成されている。サーバーPCでは、Dockerを用いて、Windows OSとOpenPNEの運用環境を隔離している。この設定により、OpenPNE環境を独立させ、移植性が向上し、クリーンなメトリクスの取得が可能となる。クライアントPCにはJMeterをインストールし、LANケーブル

ルでサーバーPCと接続する。物理的な接続による影響のリスクがあるが、通信関連の外的な影響は排除できる。

本研究で使用する物理デバイスは以下の通りである。

- サーバー PC:[CPU] 13th Gen Intel(R) Core(TM) i9-13900 3.00GHz, [RAM] 16.00GB, [OS] Windows 11 pro(23H2)
- クライアント PC:[CPU] Intel(R) Core(TM) i7-6600U 2.60GHz, [RAM] 8.00GB, [OS] Windows 10 Enterprise(22H2)

また、メトリクスとその収集方法は以下の通りである。

- サーバー PC: Docker stats³コマンド
メトリクス: CPU使用率, メモリ使用量 (0.5秒毎)
- クライアント PC: Jmeter⁴
メトリクス: エラー率

3. 実験

本実験では、負荷の大きさをRPS (Requests Per Second) で表す。実験を開始して、0hから1hの間に10,000スレッドを立ち上げ、その後1hから25hをシナリオごとの負荷期間として設定する。さらに、Torquatoら[2]を参考にし、監視期間を設定する。具体的には、3.2節と3.3節で、25hから27hを監視期間として追加し、10RPSの負荷を与えメトリクスを監視する。

3.1. 本実験環境の性能調査

本節では、 $RPS = 10, 30, 50, 70$ で負荷を与え、本実験環境におけるサーバーの性能限界を推測する。

表 1: エラー率とサーバーの性能限界

負荷 [RPS]	5	10	15	20
エラー率 [%]	0.36	3.08	28.51	追加中
性能限界 [RPS]	4.98	9.69	10.72	追加中

表1に、エラー率とそれに基づいて推測されたサーバーの性能限界を示す。性能限界は、1秒間に正しく処理できるリクエスト数とし、性能限界 = 負荷 × (100% - エラー率) により計算する。表1から、本実験環境の性能限界は9RPSから11RPS程度であると推測する。

¹<https://www.openpne.jp/about/>

²<https://about.x.com/ja>

³<https://docs.docker.com/reference/cli/docker/container/stats/>

⁴<https://jmeter.apache.org/>

3.2. シナリオ 1: 一定負荷 (監視期間 2h)

本節では現実の定常的な状況を想定した一定負荷のシナリオを考える. 3.1 節の結果を踏まえ, $RPS = 5$, 10, 15 と設定して, 3 種類のメトリクスデータを得た. スレッド立ち上げの影響が残る負荷期間の最初の 15 分を除いたデータに対して, Mann-Kendall 検定 [3] と Sen の傾き推定 [4] を行った結果, 前者は帰無仮説が棄却され, 後者は傾きの推定値が正であることから, OpenPNE のサーバーにソフトウェアエージングのリスクが確定された.

3.3. シナリオ 2: 増加負荷 (監視期間 2h)

本節では, 3.1 節より推測した性能限界の範囲内で負荷を段階的に増加させ, 現実における注目度が高まる状況を想定する. 具体的には, 1h から 25h の間に, 1RPS から 15RPS まで線形に負荷を増加させる.

図 2, 図 3, 図 4 は, 3 種類のメトリクスの時系列変化を示している. スレッド立ち上げ期間中は負荷の大きさにより, 各メトリクスが比較的高い値を示している. 増加負荷の期間では, 負荷の増加に伴い, 全メトリクスが増加していることが確認できる. 特にエラー率は, 負荷が約 10RPS 程度に達する 16 時間ごろから急激に増加しており, 推測された性能限界を超えた分の負荷に対してエラーが生じた. また, CPU 使用率とメモリ使用量については, 同時間帯までは増加傾向にあったが, それ以降は急激な増加は見られなかった. 3.1 節と 3.2 節の結果と合わせて考えると, メモリ使用量が CPU 使用率よりも先に限界に達したことが, エラー率上昇の原因と考察する.

また, 3.2 節の監視期間のデータと比較すると, と CPU 使用率メモリ使用量は, 負荷期間の差異による監視期間のメトリクスの変動への影響は見られなかった. 一方で, エラー率は, 負荷の程度によって, 増減の範囲が大きくなるという意味で不安定になる可能性があることを確認できた.

4. 結論と今後の展望

本研究では, 比較的シンプルな SNS 設定を用いて, サーバーの性能限界を推測し比較的低い一定負荷でもソフトウェアエージングのリスクが存在することを確認した. 一方で, 一定負荷と増加負荷の場合との間で, 監視期間の各メトリクスデータへの影響は確認できなかった. また, 本実験環境では意図しないメモリ使用量の限界が確認された. 具体的には, OpenPNE のサーバーコンテナに割り当てたメモリは 16GB であるが, 実際のメモリ使用量は 25%以下で停滞したことに伴い, CPU

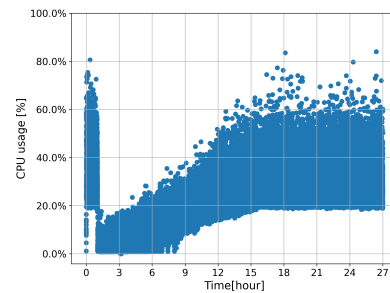


図 2: CPU 使用率

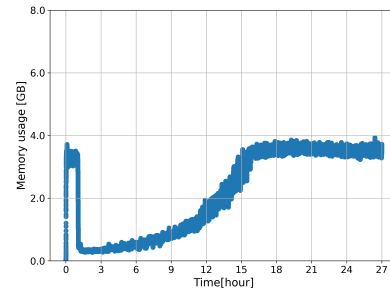


図 3: メモリ使用量

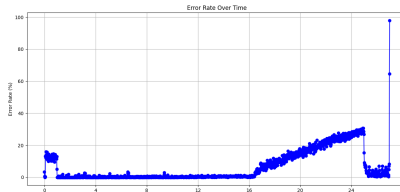


図 4: エラー率

使用率の上昇も停滞した結果, エラーが生じた. これは, サーバー運用において予算が限られている場合, 高性能な CPU を用意するよりも, メモリ使用量を増やす方が効果的であることを示唆する. 今後は, より高度な拡張機能を追加し適切な設定を整えた上で, ここで得られた知見の検証を行うことが課題である. また, 近年では, SNS だけでなく, 多くの Web サーバーはクラウド環境で運用されている. 従って, 今後は OpenPNE をクラウド環境にデプロイし, より現実的な負荷テストを実施することが望まれる.

参考文献

- [1] T. Dohi, K. Trivedi and A. Avritzer, *Handbook of Software Aging and Rejuvenation*, WORLD SCIENTIFIC, pp. 73–90, 2020.
- [2] M. Torquato, J. Araujo, I. M. Umesh, and P. R. M. Maciel, “SWARE: A Methodology for Software Aging and Rejuvenation Experiments,” *Journal of Information Systems Engineering & Management*, vol. 3, pp. 15, 2018.
- [3] H. B. Mann, “Nonparametric Tests Against Trend,” *Econometrica*, vol. 13, no. 3, pp. 245–259, 1945.
- [4] P. K. Sen, “Estimates of the Regression Coefficient Based on Kendall’s Tau,” *Journal of the American Statistical Association*, vol. 63, no. 324, pp. 1379–1389, 1968.