

# 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 OpenPNE<sup>1</sup> を選択する。動機付けは、X<sup>2</sup> 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 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はあるが、通信関連の外部的な影響は排除できる。

本研究で使用する物理デバイス及び、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: [CPU] 13th Gen Intel(R) Core(TM) i9-13900 3.00GHz, [RAM] 16.00GB, [OS] Windows 11 Pro (23H2)  
メトリクス: CPU 使用率, メモリ使用量 (0.5 秒毎)  
収集方法: Docker stats<sup>3</sup> コマンド

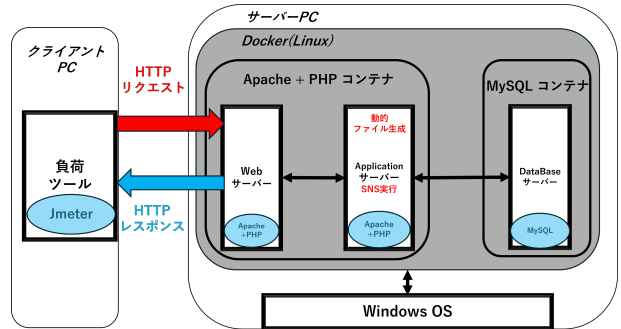


図 1: SNS 運用環境と負荷ツール

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

- クライアント PC: [CPU] Intel(R) Core(TM) i7-6600U 2.60GHz, [RAM] 8.00GB, [OS] Windows 10 Enterprise (22H2)
- メトリクス: エラー率  
収集方法: Jmeter<sup>4</sup>

## 3. 実験

本実験では、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 を負荷期間として設定する。さらに、Torquato ら [2] を参考にし、25h から 27h を監視期間として設定し、RPS= 10 の負荷を与え、メトリクスを監視する。なお、RPS (Requests Per Second) は負荷の大きさを表す単位である。

### 3.1. 本実験環境の性能調査

本節では、RPS= 5, 10, 15, 20 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 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。なお、本節の実験では監視期間を設定していない。

### 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

<sup>1</sup><https://www.openpne.jp/about/>

<sup>2</sup><https://about.x.com/ja>

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

<sup>4</sup><https://jmeter.apache.org/>

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の妥当性が確認された。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。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、メモリ容量を増やす方が効果的であることを示唆する。さらに、3.2節と同期間のメトリクスデータに対して、Mann-Kendall検定とSenの傾き推定を行った結果、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り、各メトリクスの増加傾向が認められた。

また、監視期間におけるエラー率は直前の負荷期間のシナリオによって、異なる振舞を見せていることが確認された。

## 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。また、監視期間のエラー率の振舞が負荷期間のシナリオによって、異なる現象は確認されたが、同期間のCPU使用率やメモリ使用量の振舞に同様の傾向は見られなかったことから、追加のメトリクスの観察が必要であると考えられる。加えて、本実験では意図しないメモリ使用量の制限が確認された。具体的には、

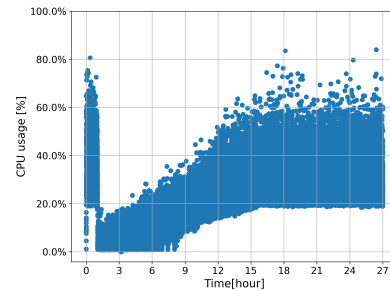


図 2: CPU 使用率

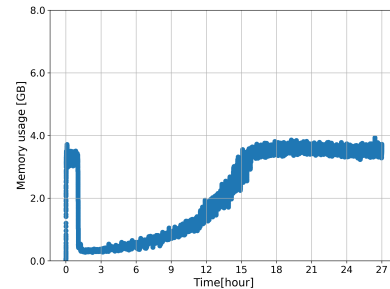


図 3: メモリ使用量

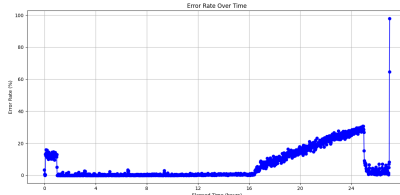


図 4: エラー率

DockerでAPACHE+PHP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4GB(25%)以下で停滞している。この現象の原因究明及び、ソフトウェアエージングとの関連性の解明が課題として挙げられる。

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.