

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ものまでに至り、非常に長時間の連続稼働が必要なものも数多く存在する。そのような連続稼働により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最新のソフトウェアにおいて、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。SNS ならではの変動する負荷期間を想定し、共通の監視期間における三種類のメトリクスに着目し、ソフトウェアエージングのリスク調査を目指す。

2. 研究目的

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直接的に SNS システムを対象にしたものは、私たちの知る限りない。X²のような SNS システムは、拡張性も高く莫大なユーザー数を抱えるため、企業にとってサーバー運用のリスクは高まっているといえる。本研究では、Measurement-based Approach[1] に則って、ソフトウェアエージングのリスクを調査する。特に、SNS システムならではの負荷シナリオを検討し、それぞれにおいて、負荷期間と監視期間 [2] を設けることで、シナリオ毎のソフトウェアエージングの調査を行う。

3. 実験環境

図 1 が実験環境であり、二つのデバイスで構成される。サーバー PC では、Docker により、Windows OS と OpenPNE の運用環境を隔離している。また、OpenPNE の環境を Docker 上に独立させることで移植性が高く、メトリクスをクリーンに取得できる。Jmeter をインストールしたクライアント PC とは、LAN ケーブルでつなぐ。そのため、物理的な接続の影響を受けるリスクが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は以下のとおりである：

- サーバー PC: HP Z2 mini G9 workstation desktop PC, 13th Gen Intel(R)Core(TM) i9-13900 3.00GHz, RAM:16.00GB, Win11 pro, 23H2

¹<https://www.openpne.jp/about/>

²<https://about.x.com/ja>

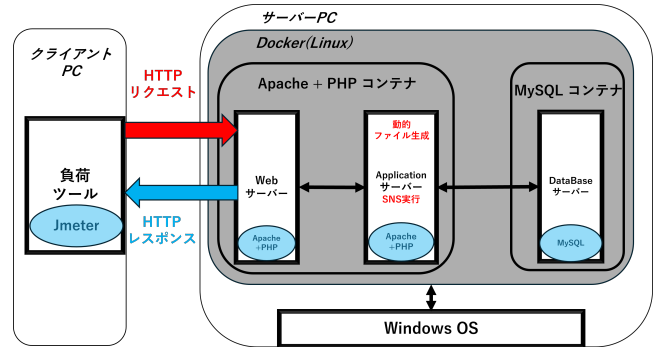


図 1: SNS 運用環境と負荷ツール

- クライアント PC: Desktop-7CL981G, Intel(R) Core(TM) i7-6600U CPU@2.60GHz, RAM:8.00GB, Win10 Enterprise, 22H2

また、データ収集は以下のように行う：

- サーバー側: Docker stats³ コマンドで取得。
メトリクス: CPU 使用率, メモリ使用量 (0.5 秒毎)
- クライアント側: Jmeter⁴ で全リクエストを取得。
メトリクス: エラー率

4. 実験

以下の実験において、負荷の大きさは、RPS (Requests Per Second) で表し、Jmeter で設定する。0h～1h では 10,000 スレッドを立ち上げ、1h～25h をシナリオ毎の負荷期間とする。これらに加え、本研究では、Torquato らの 2018 年の研究を参考にし監視期間を考え、4.2 節、4.3 節では 25h～27h に監視期間を追加し、10RPS の負荷を与えメトリクスを監視する。

4.1. 本研究の性能調査

本節では、RPS=10,30,50,70 で負荷を与え、本環境の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]	10	30	50	70
エラー率 [%]	0.87	56.51	72.75	80.20
性能限界 [RPS]	9.91	13.5	13.63	15.33

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

⁴<https://jmeter.apache.org/>

表1は、エラー率および結果から推定できるサーバーの性能限界を示したものである。性能限界は、1秒間に正しく処理できるリクエスト数だと仮定し、性能限界＝与えた負荷×リクエスト成功率(100－エラー率)/100の計算により得られたものである。この結果から、本環境の性能限界は12～15RPS程度であると推測する。

4.2. シナリオ 1: 一定負荷

4.1節の結果より、RPS=5, 10, 15で試行する。現実において、定常的な状況と対応する。三つの負荷シナリオについて、1h～25hの負荷期間における全メトリクスは、Mann-Kendall検定[3]により帰無仮説が否定された。また、Senの傾き推定[4]により傾き E の推定値は、0.00005～0.0035の範囲と非常に小さいものの、正の値を示し、ソフトウェアエージングのリスクが発見された。ただし、検定の対象からは、スレッド立ち上げの影響が残る負荷期間の最初の15分を除く。

4.3. シナリオ 2: 増加負荷 (監視期間 2h)

4.1節の結果より、推測した性能限界の範囲内で増加変動を行い、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～25hにかけて、1RPSから15RPSまで線形に負荷を増加させる。対照実験である4.2節の結果と比較し、監視期間における各メトリクスの変動に着目する。

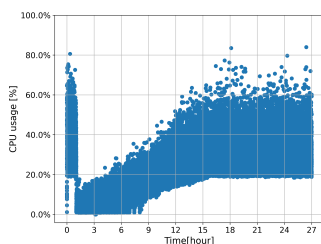


図 2: CPU 使用率

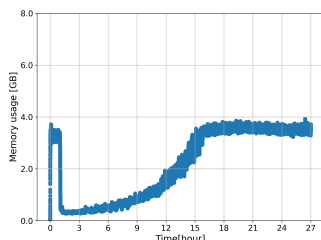


図 3: メモリ使用量

図2, 3, 4は、増加負荷における各メトリクスの時系列変化である。スレッド立ち上げ期間は負荷の大きさから各メトリクスが比較的高い値を示している。増加負荷の期間については、負荷の増加に伴い、全メトリクスが増加していることが分かる。特に、エラー率については、負荷が10RPS程度に至る16時間ごろから極端

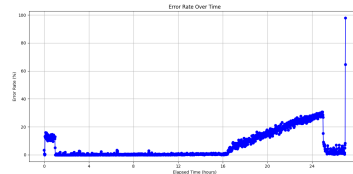


図 4: エラー率

に増加しており、推測した性能限界よりも低い負荷の段階でエラーを生じた。一方で、他二つのメトリクスについては、同時間帯までは増加傾向にあったが、それ以降極端な増加は少なくとも見られない。これまでの4.1や4.2節の結果と併せて考えると、メモリ使用量がCPU使用率よりも先に制限に達したことで、エラー率が上昇し始めたと考えられる。ただし、この制限は本研究における環境設定の不備だと考えられる。具体的には、コンテナに割り当てたメモリは16GB、CPUは8コアであるが、前者は25%以下、後者は20%～80%程度の使用率であるにも関わらず、エラーを生じている。

また、4.2節の結果と比較すると、メモリ使用率、CPU使用率に関しては、負荷期間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。

5. 結論と今後の展望

本研究では、比較的シンプルなSNSの設定で行い、サーバーの性能限界の推測および比較的低い一定負荷においてもソフトウェアエージングのリスクがあることを発見した。一方で、増加負荷による各メトリクスの影響は発見できなかった。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うえで、ここで得られた知見の検定を行うことが課題である。近年では、SNSだけでなく、多くのWebサーバーは、クラウド環境に用意されている。そのため、今後OpenPNEをクラウド環境にデプロイするなどして、大規模でより現実的な負荷テストを行うことが望ましいと言える。

参考文献

- [1] T.Dohi, K.Trivedi and A.Avrizter, “Handbook of Software Aging and Rejuvenation,” WORLD SCIENTIFIC, pp. 73-90, 2020.
- [2] M.Torquato, Araujo, Matheus, Jean, I.M.Umesh, and P.Maciel, “SWARE: A Methodology for Software Aging and Rejuvenation Experiments,” Journal of Information Systems Engineering & Management, vol. 3, 2018.
- [3] H.B.Mann, “Nonparametric Tests Against Trend,” Econometrica, vol. 13, no. 3, pp. 245-259, 1945.
- [4] P.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.