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ソフトウェアにおいて、ソフトウェアエージングの潜在可能性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[1] を選択する．SNS ならではの変動する負荷期間を想定し，監視期間における 3 種類のメトリクスに着目する．その結果，比較的低い一定負荷においても，少なくともソフトウェアエージングの潜在可能性を確認できた (暫定)．

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．現代の SNS システムは拡張性も高くユーザー数も莫大であることに伴い，X 社[2]のような運営企業にとって，サーバーの運用リスクは高まっているといえる．本研究では，Measurement-based Approach[4] に則って，ソフトウェアエージングの潜在可能性を調査する．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 [5] 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，Docker により，ホスト PC の 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．一方で，負荷ツールとは LAN ケーブルでつなぐため，物理的な接続の影響を受けるリスクがある．

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

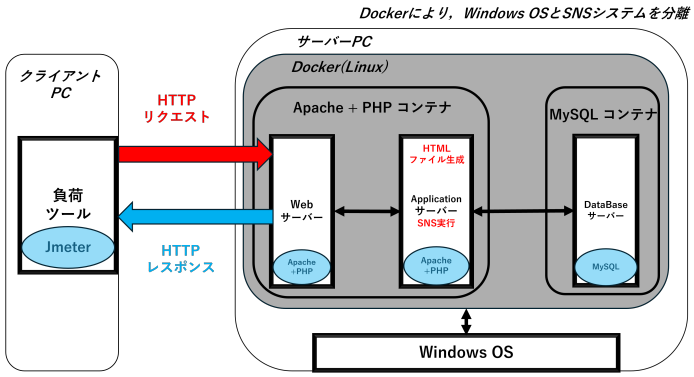

図 1: SNS 運用環境と負荷ツール

- クライアント:Desktop-7CL98lG，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 stats[3] コマンドで取得.
  計測対象:CPU 使用率，メモリ使用量 (0.5 秒毎)
- クライアント側:Jmeter[4] で全リクエストを取得.
  計測対象: リクエストエラー率 (※エラー率と表記)

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 では，シナリオ毎の負荷を与える．これらに加え，本研究では，Torquato らが 2018 年に，ストレス期間，待期期間，若化期間のそれぞれでリソースの監視を行った研究を参考にし，本研究では監視期間というものを考える．そのため，4.2，4.3 では 25h〜27h の期間を追加し，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，複数の RPS で負荷を与え，本環境のサーバーの性能限界を推定する．

表 1 は，複数の一定負荷シナリオのエラー率，およびそれらの値から推定できる性能限界を示したものである．

[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/

表 1: エラー率と性能限界

RPS	error rate[%]	estimated RPS
10	0.87	-
30	56.51	13.5
50	72.75	13.63
70	80.20	15.33
Avg	-	14.15

この結果から，本環境の性能限界は 14.15RPS であると推定する．また，メモリ使用量について，Mann-Kendall 検定と Sen の傾き推定の結果から，10RPS と 50RPS でソフトウェアエージングのリスクが認められた．

4.2. シナリオ 1: 一定負荷

5，10，15RPS の図，検定結果，ページ制限のためテキストのみでも可

4.3. シナリオ 2: 増加負荷 (監視期間 2H)

4.1 の結果より，1〜15RPS の増加変動を行い，アクセスが増加するという現象を想定する．対照実験として，**??**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し，ソフトウェアエージングの影響の差異に着目する．

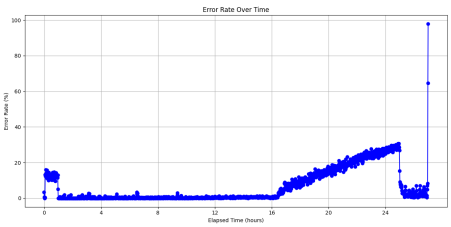

図 2: エラー率

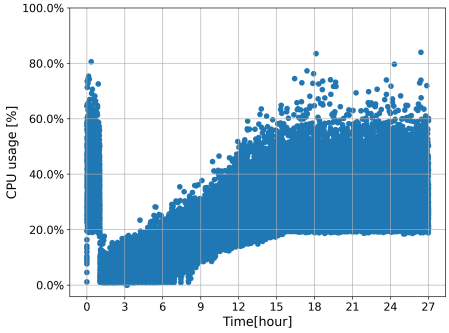

図 3: CPU 使用率

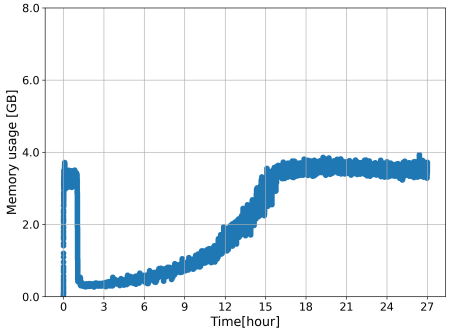

図 4: メモリ使用量

図 2，3，4 は，1〜15RPS の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，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，RPS が 10RPS 程度に至る 15 時間ごろから極端に増加しており，推定 RPS 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，その時間帯までは増加傾向にあったが，それ以降は極端な増加は少なくとも見られない．また，負荷後 2 時間のメトリクスについて，

考察としては，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 50%以下，後者は 20%〜80%の使用率であることを考えると，メモリ使用の設定に問題があると考えられる．この主な原因は，php の mpm_event モジュールが並列処理に不適な環境であることだと考えられる．以降については，追加実験の結果が出てから．

5. 結論と今後の展望

結論．

本研究では，比較的シンプルな SNS の設定で行い，〇〇と分かったが，より高度な拡張機能を有したものに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 SNS 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] Adams, N.Edward, “Optimizing Preventive Service of Software Products,” IBM Journal of Research and Development, vol. 28, no. 1, pp. 2-14, 1984.

[2] D.L.Parnas, “Proceedings of the 16th International Conference on Software Engineering (ICSE1994),” Software aging, pp. 279-287, 1994.

[3] Y.Huang, C.Kintala, N.Kolettis and N.D.Fulton, “Software Rejuvenation: Analysis, Module and Applications,” 25th International Symposium on Fault-Tolerant Computing(FTCS1995), pp. 381-390, 1995.

[4] T.Dohi, K.Trivedi and A.Avritzer, “Handbook of Software Aging and Rejuvenation,” WORLD SCIENTIFIC, pp. 73-90, 2020.

[5] 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.