

処理性能の低下に着目したソフトウェア若化モデル

池淵 広貴[†] 蓑原 隆[†]
[†] 拓殖大学工学部情報工学科

1. はじめに

近年、ソフトウェアの運用時間の経過にしたがって性能が低下したり突然ハングアップしたりするエイジングと呼ばれる現象が問題になっている。この問題に対して、ソフトウェア障害発生前に再起動や初期化を行うことで障害発生を回避しようとするソフトウェア若化[1]という対策が考えられている。

ソフトウェア若化では若化処理中のサービスが停止することから、若化を行うタイミングが重要な問題である。これまでに確率的なモデルを用いたタイミングの決定や、実測に基づくタイミングの決定の研究が行われている[2]。本研究では、ソフトウェアエイジングによって低下するスループットに注目し、若化処理を考慮したスループットを最適化するためのソフトウェア若化モデルの提案を行う。

2. 処理性能を考慮したソフトウェア若化モデル

ソフトウェアエイジングによる性能低下が時間によって進んでいく状況として、時刻 t の処理性能を $f(t)$ で表し、ソフトウェア若化(再起動)に必要な時間を τ とすると、時刻 t で若化を行ったときの時刻0から若化終了時刻 $t + \tau$ までの平均スループット $g(t)$ は次の式で与えられる。

$$g(t) = \frac{\int_0^t f(t) dt}{t + \tau}$$

この式において $f(t)$ が単調に減少すると、 $g(t)$ は一度増加した後、減少に転ずる。したがって、 $g(t)$ が減少に転ずるタイミングでソフトウェア若化を行えば、平均スループットを最大化できる。 $g(t)$ を t で微分すると次のようになり、 $f(t) - g(t) = 0$ となる t で極大になる。

$$\frac{dg(t)}{dt} = \frac{f(t)}{t + \tau} - \frac{\int_0^t f(t) dt}{(t + \tau)^2} = \frac{1}{(t + \tau)} (f(t) - g(t))$$

よって、 τ が既知であれば、ソフトウェア起動からのスループットの変化 $f(t)$ を実測しながら、 $g(t)$ を算出し、両者が一致したタイミングでソフトウェア若化を開始すればよいと考えられる。

3. キャッシュミスの挿入による性能低下の推定

実際のソフトウェアの性能低下の挙動を調べるために、長時間ソフトウェアを動作させて試験をすることは現実的には困難である。そこで本研究では、Webサ

ーソフトウェア apache を対象として、メモリ資源の欠乏をエミュレートすることでソフトウェアエイジングの挙動を調べる実験を行った。具体的には、Valgrind[3]というソフトウェアの実行をエミュレートするツールを用いてキャッシュサイズ等のパラメータを変更し、意図的にキャッシュミスが発生させることで、性能低下を誘発する。

キャッシュサイズの変更に伴うミスの誘発による性能低下と実際の運用時間の関係は明らかではないが、キャッシュサイズの減少がソフトウェアエイジングの時間経過に比例すると仮定し、若化処理時間を換算し平均スループットの変化をプロットしたものを図1に示す。図のように平均スループットが時間の経過にしたがって上昇したのち緩やかに減少し、ピークとなる時間は若化時間が大きくなるほど遅くなるという結果が得られた。

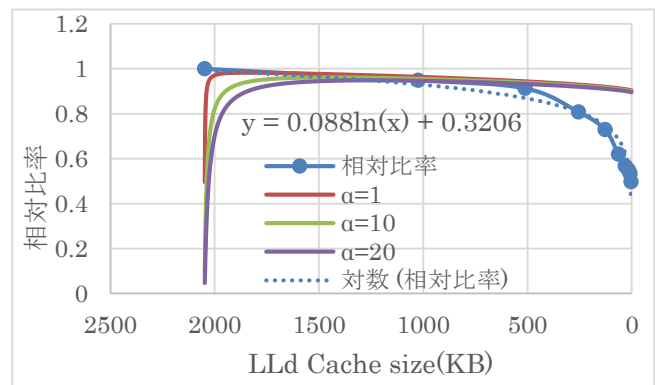


図1 平均スループットの変化

4. まとめ

本稿ではソフトウェアエイジングによる性能低下の観点から若化タイミングを求めるモデルの提案を行った。また、提案したモデルについてキャッシュエミュレーションによる実験を行い平均性能の変化を調べた。

参考文献

- [1] Y. Huang, et al. "Software rejuvenation: analysis, module and applications", in FTCS-25, 1995, pp. 381-390
- [2] K. Vaidyanathan, K. Trivedi, "A comprehensive model for software rejuvenation", IEEE Trans. on Dependable & Secure Computing, vol. 2, no. 2, 2005
- [3] Valgrind Home <http://valgrind.org>, (2015年6月8日アクセス)