

C-041

SIP プロトコル処理におけるガベージコレクションの影響評価に関する検討
 A study of an evaluation of effect about a garbage-collection on SIP protocol processing

金子 雅志
 Masashi Kaneko

入江 道生
 Michio Irie

四七 秀貴
 Hideki Shina

飯尾 政美
 Masami Iio

日本電信電話株式会社 NTT ネットワークサービスシステム研究所
 NTT Network Service Systems Laboratories, NTT Corporation

1. まえがき

NGN(Next Generation Network)の商用サービスが開始され、新たなネットワークサービスの登場が望まれている。NGNのアプリケーション開発技術において、JCPで標準化されている SIP Servlet(JSR116,289)が注目されている。SIP Servletは J2EE ベースのサーバーサイドプラットフォームであり、Java/J2EE の開発で培われた高い生産性が期待できる [1]。Java の大きな特徴として、GC(ガベージコレクション)が挙げられる。GC 対応の言語では開発者が明示的なメモリ確保・解放を行う必要がないため、生産性・ソフトウェア品質の向上が期待できる。一方で、GC 処理中の停止時間(pause-time)が与える影響が懸念される。特に、通信事業者が運用する VoIP サーバ等の SIP サーバはソフトウェアリアルタイム処理によって高い応答性を実現してきたため、Java による pause-time が性能(応答時間・呼損率)に与える悪影響が懸念されてきた。近年、Real-time 対応を謳った JavaVM が登場してきており、リアルタイム処理分野への適用拡大が注目されているが、SIP のプロトコル処理に対する pause-time の定量的な影響については明らかにされておらず、VoIP サーバ等への Java 適用の阻害要因となっている。

本稿ではサーバーサイドの SIP プロトコル処理における GC の影響についてシミュレーションと実機評価によって解析した結果について示す。

2. SIP プロトコル処理における Java の適用

2. 1. SIP プロトコル処理の特徴

SIP は UA(User-Agent)間のセッション制御を主な目的としたプロトコルであり、VoIP の通話制御などで利用されている。テキストベースプロトコルであり、HTTP に似た特徴を持つが、トランスポートプロトコルとして UDP をサポートしている点や、非同期イベントを扱うことが可能な点が異なる。UDP-SIP では、メッセージの到達性を保証するために SIP レベルでの再送制御が規定されている。これにより、経路上でのパケットロスやサーバにおける処理バッファ溢れによるパケット破棄が発生した場合でも呼損になることを防いでいる。UDP の再送タイミングは INVITE リクエストで 500ms-1s-2s-4s-8s-16s、非 INVITE リクエストで 500ms-1s-2s-4s-4s...と RFC3261 で規定されており、32 秒以内に応答がない場合はタイムアウト(呼損)となる。SIP サーバの実装では、トランスポート層でのパケット受信遅れによる SIP メッセージのロスを防ぐために、処理前の SIP メッセージをバッファに一時保存した後に、後段の処理スレッドでア

プリケーション処理を行う場合が多い(図 2.1)。

2. 2. Java における stop-the-world について

Java アプリケーションにおいては GC による pause-time は不可避であり、システムの性能設計時にその影響を考慮する必要がある。特に Full GC 時にはメモリの解放だけでなく、断片化したメモリ領域を再配置する処理も行われるが、その際に JavaVM 上の全てのスレッドが停止した状態で行われるため、処理のリアルタイム性は著しく損なわれる。

2. 3. GC が SIP プロトコル処理に与える影響

図 2.1 で示したような処理モデルにおいて GC が発生すると、pause-time 時に一時的にバッファ内に処理待ちのメッセージが溢れ、信号破棄が発生する可能性がある。さらに pause-time が長ければ、信号破棄と応答遅延による再送が大量に発生し、サーバが高負荷状態になる可能性がある。

3. GC 影響のシミュレーション

3. 1. シミュレーションモデルと評価環境

前節のように pause-time が SIP プロトコル処理に与える影響を定性的に推測することはできるが、サーバの性能設計を行う場合には定量的な推測まで可能であることが望ましい。本稿では pause-time の影響を解析するため、待ち行列をベースとしたシミュレーションモデルを構築した(図 3.1)。まず、SIP 端末等のリクエスト発生元(UAC)と、リクエストを処理するサーバ(UAS)の関係を、リクエストの発生率 λ_c 、サーバでのサービス率 μ_T の待ち行列モデルで表現する。また、UDP-SIP の再送を表現するため、個々のリクエスト毎に応答が返信されるまでの間に標準で定められた間隔で再送メッセージをサーバに送信し、その処理時間を μ_r とした ($1/\mu_T > 1/\mu_r$)。また、GC による pause-time を表現するために、GC をサービス率 μ_{GC} の優先処理とした。定期間隔で発生する GC は処理中のメッセージに割り込み、 $1/\mu_{GC}$ の時間処理が持続することとした。本稿におけるシミュレーションの設定においては、一般的な JavaVM(非 Real-time)を対象として非 INVITE リクエストの負荷を再現した。また、シミュレーション結果との比較を目的として Java による SIP プロトコル処理を行うテストプログラム(TP)の環境を構築した。TP は図 2.1 のようなバッファを持つ SIP サーバプログラムであり、MESSAGE リクエストに対して 200OK 応答を送信するものである。

3. 2. シミュレーションの観点

本稿では主に 2 つの観点を確認するためにシミュレーションを行った。最初に前節で示したシミュレーションモデ

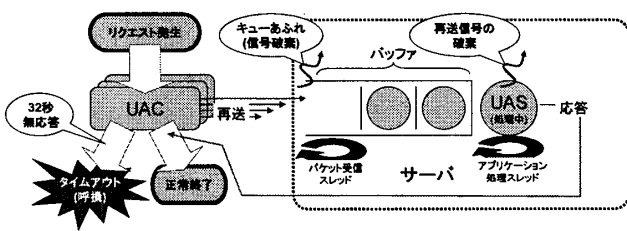


図 2.1 一般的な SIP サーバ処理モデル

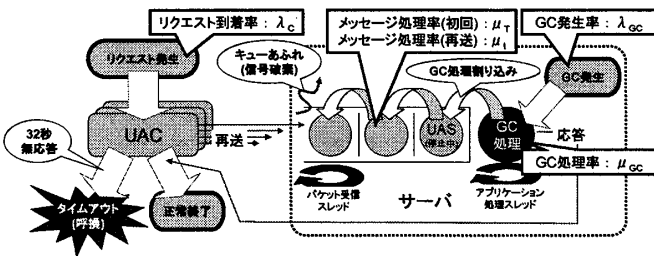


図 3.1 GC による pause-time のシミュレーションモデル化

ルの妥当性を確認するため、GC 発生時の処理待ちメッセージ数(バッファ内の滞留メッセージ数)の推移をシミュレーションと TP 環境で測定し、両者を比較した(観点1)。次に GC による pause-time が性能に与える影響を確認するため、pause-time を 0~5 秒に変化させた場合のリクエストタイムアウトの発生率(呼損率)についてシミュレーションで評価した(観点2)。

3. 3. シミュレーションの条件

(観点1)に対するシミュレーションのパラメータとしては、リクエスト発生率(λ)を 1000、TP でのバッファサイズを 1000,2000,4000 とした場合の実測値を元に、1 リクエスト処理に要する時間($1/\mu_r$)を 550 usec、再送処理時間($1/\mu_s$)を 100 usec、pause-time ($1/\mu_{GC}$)を 2 sec、GC 発生間隔を 10 秒(一定間隔)として評価を行った。(観点2)としてバッファサイズ 2000 の条件下においてシミュレーション上で pause-time を 0 秒~5 秒に変化させた場合の評価を行った。

3. 4. シミュレーション結果

(観点1)シミュレーションモデルの妥当性

図 3.2 は GC 発生時刻[A]を基点として、経過時間に対する処理待ちメッセージ数を表したグラフである。グラフを見ると、バッファサイズを変化させた場合であってもシミュレーション結果と実測値が似た傾向を示しており、モデルの妥当性が確認できる。グラフ上の時間的な変動を見ると、GC 時に処理待ちメッセージが増加している([A]~[D])ことから、これを“一時的な輻輳状態”と定義する。これは、GC によってプロトコル処理が停止することから急激にバッファにメッセージが蓄積し([A]~[B])、GC 完了後も新規信号と再送信号によって SIP メッセージの到着率が上昇することにより一時バッファ溢れの状態が継続し([B]~[C])、再送の発生源となっている応答待ちの UAC の減少に伴い滞留メッセージが減少に転じ([C]~[D])、その後定常状態に戻る([D])、という状態推移によるものである。グラフから読み取れるように、一時的な輻輳状態が収まるまでの時間([A]~[D])はバッファサイズが小さい方が早い。バッファサイズと再送信号処理数の関係の評価するために、1 リクエストあたりの平均再送回数とバッファ溢れによるメッセージ破棄発生率についてのシミュレーション結果を比較した(図 3.3)。グラフより、バッファサイズが小さい方がメッセージの破棄率が大きい、1 リクエストあたりの平均再送回数が少ないことが読み取れる。これは、一時保存するメッセージ数を制限することにより処理しなくてもよい再送メッセージが確率的に破棄され、サーバ側における再送メッセージ処理によるオーバーヘッドが低減されているためである。逆にバッファサイズを大きく取ると、オーバーヘッドが増加し、平常状態に戻るまでの時間が延びてしまう。

(観点2)SIPプロトコル処理性能に対する pause-time の影響

図 3.4 はバッファサイズ 2000 の条件で、シミュレーションによって pause-time を 0~5 秒に変化させた場合の平均再送回数と呼損率を表している。pause-time の長期化に伴い再送回数が上昇し、4.5 秒以上の場合には常にバッファ溢れの状態となり、呼損が発生している。GC の発生頻度と 1 回あたりの pause-time は 1 リクエストあたりのオブジェクト生成数と正の相関があるため、オブジェクトを多数生成するアプリケーションにおいては、輻輳状態が収まる([D])前に次の GC が発生([A])してしまう場合がある。このような状況においては、たとえ平常時のトラフィック密度が低くても GC を契機にシステムが長期間輻輳状態となってしまう。そ

のような状態を防ぐためには、ビジー応答等により端末側に再送の遅延を促す等、一時的な輻輳状態を早期に解消する対策が必要である。

4. まとめ

本稿では SIP アプリケーション開発への Java 適用に向けた GC の影響についてシミュレーションモデルを構築し、実機との比較を行った。評価の結果により、GC による pause-time 及び SIP の再送制御がシステムに与える影響が確認できた。pause-time の最大時間が保証されない一般的な JavaVM では定常時と GC による輻輳状態発生時で負荷に大きな変動が発生することから、CPU リソースの有効利用が困難であることが想定される。

今後の課題としては、シミュレーションモデルの拡張による INVITE~BYE 処理パターンへの適用や、Real-time 対応 JavaVM [2]が処理性能に与える効果について評価を行い、Java による SIP アプリケーション開発の性能設計において汎用的に利用可能な性能予測手法の確立を目指したい。

参考文献

- [1] 金子,入江,平野:”SIP Servlet のポータビリティに関する一考察”, NS 研究会(2008/10)
- [2] Real-time Specification for Java (JSR-1), <http://jcp.org/>

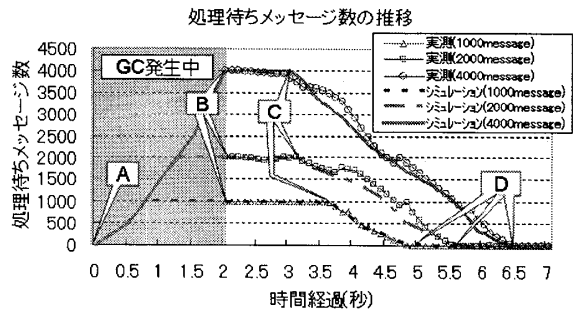


図 3.2 GC 時の一時的な輻輳状態

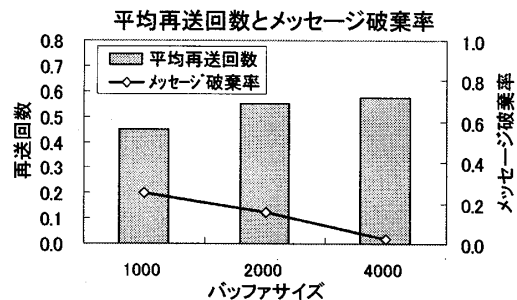


図 3.3 バッファサイズ設定による影響

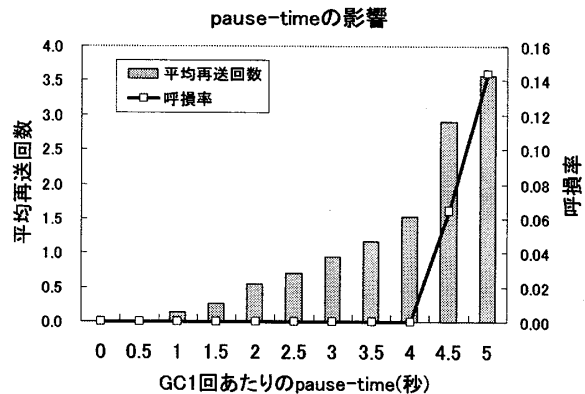


図 3.4 pause-time が再送回数・呼損率に与える影響