

# 実データ統計を用いた動的な Memcached 評価用ロードジェネレータ

Dynamic Load Generator for Memcached

定久紀基 福田エリック 浅井哲也 本村真人  
Tsunaki Sadahisa Eric S. Fukuda Tetsuya Asai Masato Motomura

北海道大学大学院情報科学研究科  
Graduate School of Information Science and Technology, Hokkaido University

## 1 はじめに

Memcached とは Key-Value ストアの一つで YouTube や Wikipedia、Facebook 等の Web サイトで使われている。また高速化の研究が盛んであるため、より正当な評価を行えるロードジェネレータが必要とされている。現在、Memcached には Facebook の統計データが論文 [1] として出されており、それを考慮した評価用ロードジェネレータ [2] があるが、Memcached の反応 (Reply) を考慮していないため実際の動作の再現性が低い。本研究では先行研究 [2] 同様に論文のデータを参考にしたうえで、さらに Reply を考慮することで先行研究 [2] よりも実際の Memcached の動作に近い状態でテストを行うことが目的である。

## 2 実装

### a. 提案手法の性質

提案手法における長所は先ほど述べたように Reply の考慮 (Memcached 上に欲しいデータが存在しなかった場合に、そのデータを Memcached に格納するという動作) である。Reply を考慮することによってデータの時間的局所性を再現することができるため、より正当な評価が行えると期待できる。また先行研究同様、文献 [1] に基づいた Request を送信する。提案手法で取り入れるのは以下の三点である。

1. データを Memcached から引き出すことに成功した割合 (Hit 率、表 1)
2. Key、Value のサイズ (図 1 の実線)
3. リクエストの比率 (Command:データを格納させる Set、データを参照する Get、データを消す Delete、表 1)

### b. ロードジェネレータの動作

本研究で既存のロードジェネレータ [2] よりも実際の動作に近づけるため、以下の点の工夫をした。

1. Memcached が容量不足によるデータの追い出しを起こすまでデータを格納する
2. a-1,2,3 を満たすリクエストを作成し送信
3. Reply を見て、Hit しなければ Value を作って Set を行う
4. 3 と 4 を所定の回数まで繰り返す

### c. 統計データの再現手法

Key は一般極値分布、Value は一般パレート分布に

従った乱数によりサイズを決定する。これらの手法は、新たな統計データが出た場合も統計的性質が明らかでない場合において、分布およびパラメータを変更することで対応することができる。また、a-1,2,3 の条件には内部パラメータの調整を行うことで柔軟に対応することができる。

## 3 評価

ロードジェネレータが生成した Request の統計を取り、文献 [1] に記されている統計モデル、Hit 率や Command の比率等と比較し評価する。評価の結果、参考文献のデータと近い統計データが取れた (表 1、図 1)。

表 1 ロードジェネレータの評価結果

	Facebook[1]	提案手法
Hit 率	81.4 %	82.7 %
Command 比 (Get:Set:Del)	30:1:14	29.7:1:14.7

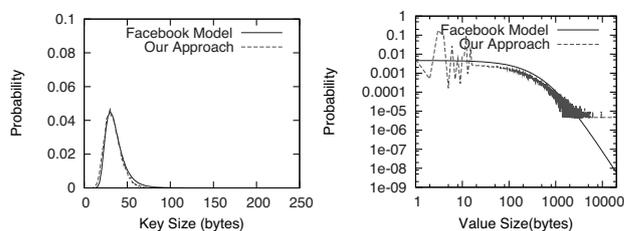


図 1 Key, Value のサイズ

## 4 まとめ

本研究では、先行研究 [2] よりも Reply を考慮することで実際の Memcached の動作に近いロードジェネレータを作成し、有効性を確認できた。しかし、論文のデータすべてを反映させることができず、またプログラムの実行速度が遅いためリアルタイムシミュレーションができなかったため、今後それらの点を改善していきたい。

## 参考文献

- [1] B. Atikoglu, Y. Xu, E. Frachtenberg, S. Jiang and M. Paleczny. Workload Analysis of a Large-Scale Key-Value Store, *2012 IEEE International Symposium on Workload Characterization (IISWC)*, November 2012.
- [2] J. Leverich. Mutilate: high-performance memcached load generator. [github.com/leverich/mutilate](https://github.com/leverich/mutilate).