2Q-04

分散オブジェクトストレージの性能の解析に関する一考察

早川峻平† 山口実靖†

†工学院大学大学院 工学研究科 電気・電子工学専攻

1. はじめに

ストレージに保存するデータ量の増加とデータの性質の変化に伴い，ディレクトリ構造では膨大なデータの維持や管理が難しくなってきている. この問題を解決するために，データをファイルやブロックではなくオブジェクトとして扱うオブジェクトストレージが提案された. オブジェクトストレージはディレクトリ構造を持たず，オブジェクトはURIが付与されデータとそのメタデータで構成される.

オブジェクトストレージでは，HTTP/REST API を用いてファイルのアップロードおよびダウンロードを行う. 多くの場合，オブジェクトストレージはストレージノードや認証ノードなどの複数のノードで構成される分散システムであり，その動作の把握や性能劣化原因の特定は困難である. 本論文では，OpenStack Swift[1]を用いてオブジェクトストレージシステムを構築し，通信の観察によるシステムの振舞の解析手法[2]を用いて Swift API を用いるオブジェクトストレージシステムの性能について考察する.

2. 関連研究

複数の計算機で構成される分散システムの動作観察システムとして，我々は過去にネットワークストレージシステムの統合トレースシステム[3][4]や，OpenFlow ネットワークにおける動的な転送制御情報の修正の観察システム[5]を提案している．ただし，これらの既存研究ではオブジェクトストレージシステムの様な多数の計算機で構成される複雑なシステムへの適用はされていない.

この様な複雑なシステムの観察手法としてパケットトレースによる解析手法[2]がある．同手法では各計算機におけるパケットの送信と受信を記録し，各パケット転送を送信時刻(横軸)と送信計算機(縦軸)の点と受信時刻と受信計算機の点を結ぶ線分で可視化し，動作の把握を容易としている．しかし，同文献では Swift API を用いるオブジェクトストレージシステムの性能解析はなされていない.

A Study of Performance Analysis in Distributed Object Storage
†1 Shunpei Hayakawa, Saneyasu Yamaguchi
†1 Electrical Engineering and Electronics, Kogakuin University Graduate School

3. 性能評価

本章にて，Swift におけるオブジェクトアップロード性能の評価を行う．実験では，ストレージノード，プロキシノード，クライアントの 3 台の物理マシンを使用し，ストレージノード上では認証サーバープロセス，アカウントサーバープロセス，コンテナサーバープロセス，オブジェクトサーバープロセスを，プロキシノード上ではプロキシサーバープロセスを実行する．オブジェクトを格納するストレージデバイスとしては，HDD あるいは SSD を使用する.

性能評価はクライアントノードにて Swift API の PUT を呼び出しアップロードすることを繰り返すことにより行った．アップロードするオブジェクトのサイズは1MB とし，100 個のオブジェクトが入ったディレクトリを複数用意し，アップロードの並列数は 1 から 32 とした．実験は，ディレクトリ内のオブジェクト 1 つにつき 1 回の API 呼び出しを行うオブジェクト単位と，1 回の API 呼び出しでディレクトリ内の全てのオブジェクトのアップロードを行うディレクトリ単位の 2 種類で行った.

図 1 に HDD および SSD における並列数とオブジェクトアップロードのスループットおよび失敗数の関係を示す．図 1 より，オブジェクト単位のアップロードはスループットが低く，HDD と SSD のどちらの場合も並列数 1 の時は 1 オブジェクトのアップロードに平均で約 4 秒要していることがわかる．また，オブジェクト単位のアップロードでは HDD と SSD の場合でほとんどスループットの差がなく，オブジェクト単位のアップロードはストレージノードの I/O 処理能力に関わらない箇所がボトルネック処理であると考えられる．ディレクトリ単位のアップロードではオブジェクト単位のアップロードに比べスループットが高く，さらに HDD を用いた場合よりも SSD を用いた場合の方がスループットが高いことがわかる.

並列数が増加するにつれてスループットも高くなるが，並列数がある程度高くなるとスループットが低下することがわかる．これは，オブジェクトアップロードの失敗によるものであると考えられる.

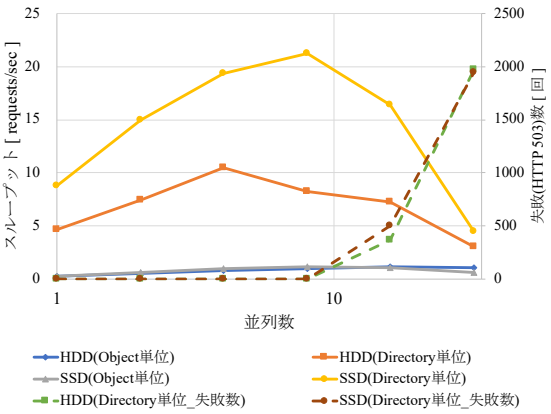

図 1 HDD および SSD におけるスループットと失敗数

4. 解析

前章の評価より，1 並列のときはアップロードに長い時間(約 4 秒)を要していることがわかったが，オブジェクトストレージは複雑なシステムのためどの処理に時間を要しているのかの判断が困難である．この課題に対して既存の解析手法[2]を適用し考察を行う．

HDD における並列数 1 のときのアップロードの振舞を図 2 に，SSD における並列数 1 のときのアップロードの振舞を図 3 に示す．

図より，最も時間を要している処理はプロキシノード上での処理あることがわかる．また，クライアント機において Swift プロセスが生成されてから最初のパケットがクライアント機から送出されるまでにも長い時間を要していることが分かる．これには Python インタプリタの起動などが含まれている．以上の様に，提案した解析手法[2]によりオブジェクトストレージシステム全体の振舞を俯瞰でき，ボトルネック処理を発見できることが確認できた．具体的には，オブジェクトノード以外のノードが性能劣化原因となっていることを確認できた．

5. おわりに

本論文では，パケット転送の記録によるオブジェクトストレージシステムの振舞の観察手法を Swift API に適用し考察を行った．

今後は，高い並列度におけるオブジェクトストレージの振舞の解析方法についての考察を行う予定である．

謝辞 本研究はJSPS科研費15H02696, 17K00109, 18K11277 の助成を受けたものである。

本研究は、JST、CREST JPMJCR1503 の支援を受けたものである。

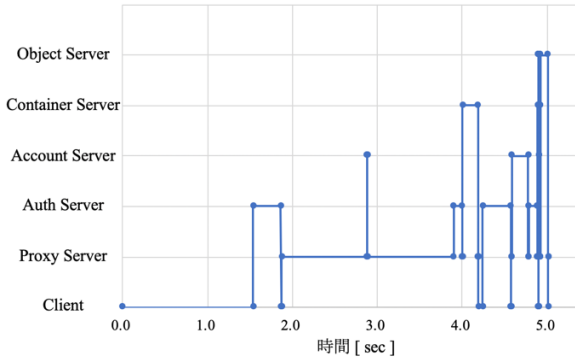

図 2 HDD における並列数 1 の時の可視化例

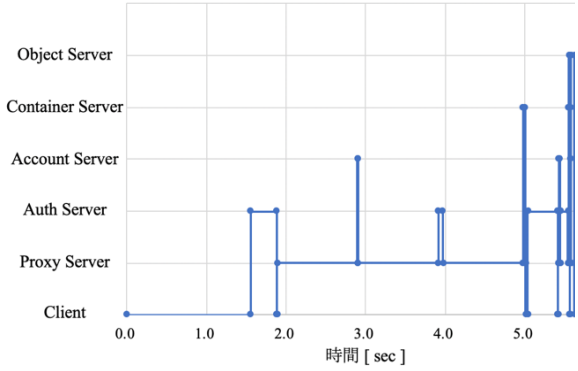

図 3 SSD における並列数 1 の時の可視化例

参考文献

[1] Sridevi Bonthu, Y S S R Murthy, M. Srilakshmi “Building an Object Cloud Storage Service System using OpenStack Swift”, International Journal of Computer Applications (IJCA’14), 2014.

[2] 早川峻平・山口実靖, “オブジェクトストレージの性能の解析に関する一考察”, 研究報告データベースシステム(DBS),2018-DBS-168(9),1-6 (2018-12-14), 2188-871X

[3] S. Yamaguchi, M. Oguchi and M. Kitsuregawa, "Trace system of iSCSI storage access," The 2005 Symposium on Applications and the Internet, 2005, pp. 392-398. doi: 10.1109/SAINT.2005.65

[4] S. Yamaguchi, M. Oguchi, M. Kitsuregawa, “Trace System of iSCSI Storage Access and Performance Improvement,” Database Systems for Advanced Applications (DASFAA), pp. 487-497, Springer Berlin Heidelberg, 2005. DOI: 10.1007/11408079_43

[5] Saneyasu Yamaguchi, Akihiro Nakao, Masato Oguchi, Atsuhiro Goto, and Shu Yamamoto. 2016. Monitoring Dynamic Modification of Routing Information in OpenFlow Networks. In *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication* (IMCOM '16). ACM, New York, NY, USA, Article 27 , 7 pages. DOI: http://dx.doi.org/10.1145/2857546.2857574