非集中型クラウドストレージのスケーラビリティ評価

氏名 奥寺 昇平学籍番号 07\_0615\_4

卒業年度 2010年度 指導教員 首藤一幸

1.はじめに

Amazon Dynamo[1]、Apache Cassandra[2]をはじめとした単一故障点がなく、負荷が自動的に分散される非集中型のクラウドストレージが普及しつつある。このような非集中なクラウドストレージにおいて、任意のノードからデータを保持する担当ノードにリクエストを到達させるためには、各ノードが他のノードを把握する必要がある。特に、クライアントが接続したノードから担当ノードに直接リクエストを送るクラウドストレージでは、全ノードがシステム全体の最新の状態を保持する必要があり、システムの整合性を保つことが難しい。

そこで、gossipプロトコルをベースとしたメンバーシップ管理を行うプロトコルが取り入れられ、効率よく通信を行うことが可能である。

一方、このようなメンバーシップ管理もスケーラビリティを制約する要因の一つとなりうる。

この管理方法では、すべてのノードで定期的に通信が発生するので、ノード数が増えるにつれ、総通信量が増えるのである。よって、フロントエンドの処理(データの読み書き処理)効率、つまりアベイラビリティを下げると考えられる。

　しかしながら、非集中型クラウドストレージにおいて、この管理を行う処理がどれくらいの通信負荷をもたらすのかといったことは知られていない。

そこで本研究では、gossipプロトコルを用いるCassandraを対象として、ノード数に応じてシステム全体の通信負荷がどのように変化するのかを計測・考察する。

2.gossipプロトコル

gossipプロトコルはソーシャルネットワークで見られる噂(ゴシップ)の伝搬をモデルとしたアルゴリズムである。例えば、以下のような手順で繰り返し行われる。

1. ノードPのデータが更新されたとき、ランダムに他のノードQを選択して更新情報をノードQに反映させる。

2.ノードQがすでに更新済みであったときは、ノードPは他のノードに更新情報を伝えるのをやめる。

The evaluation of the scalability in the Non-Decentrized type Cloud Storage

†Shohei okudera: Tokyo Instriute of Technology

gossipプロトコルに代表される伝染性のアルゴリズムを利用したプロトコルの長所は、プロセスが通信する回数が他の伝搬手法と比べて比較的少ないために、ノード数がスケールしやすいところにある。

3.Cassandra

本節では、実験に用いたCassandraについて説明する。Cassandraは, Facebook社が開発し, Apache Projectとしてオープンソース化した非集中型クラウドストレージである。Cassandraでは、gossip プロトコルをベースとしたメンバーシップ管理アルゴリズム[3]を利用している。このアルゴリズムでは、毎秒各マシンでランダムに他のノードと経路情報を交換しあい、システム全体のノード情報の整合性を保っている。

4.測定手法

実験では、１台あたり複数のCassandraノードを起動し、マシン間で発生するトラフィックを解析した。tcpdumpを使用して、トラフィックを計測した。また、Cassandraノードを多数立ち上げる際に、メモリー使用量を節約するために、Cassandraのデータ保存部分のプログラムを改変し、メモリー使用量を削減した。

以下に実験環境を示す。

・Cassandra 0.6.6

* OS: Linux 2.6.35.10-74.fc14.x86\_64

・Java 仮想マシン: Java SE 6 Update 21

・CPU: 2.40 GHz Xeon E5620×2

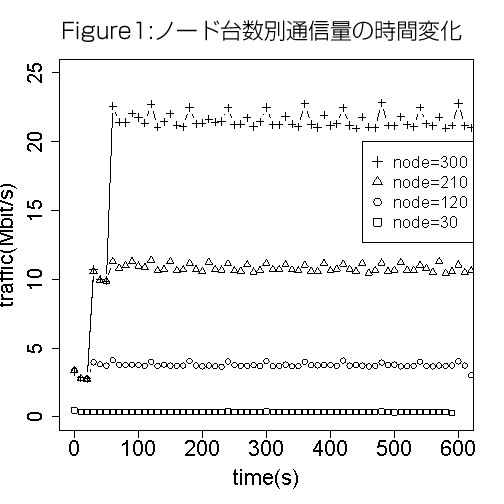
* メモリー: 32GB RAM
* ネットワーク: 1000BASE-T

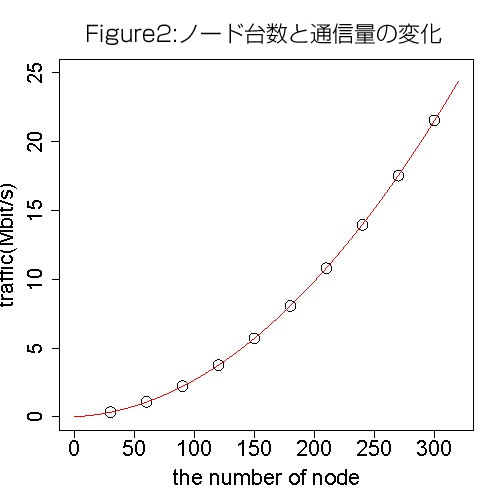
実験にはマシンを10台使用した。

実験シナリオついて説明する。30秒ごとに、１台あたり10ノードのCassandraを一度に起動し、これを目指す台数に達成するまで続ける。最初のCassandraノードを起動した瞬間から10分間の通信量を計測した。

5.実験・評価

Figure 1は、10秒あたりのCassandraノードで発生する総通信量の平均の時間変化をノード数別に表したグラフである。(ただし、1M =10^6,1K=10^3とする。)





ノード数によらず、100秒以降は通信量が安定していることがわかる。

Figure 2は、ノード数と通信量が安定している時の(ここでは、実験開始から200-300秒後とした)1秒あたりの通信量の平均をプロットしたものである。図中の曲線は、プロットした点から二次関数でフィッティングしたものである。nをノード数として得られた関数は、

[1]Avinash Lakchman and Prashant Malik,. *Cassandra – A Decentralized Structured Storage System*,In Proc. LADIS ’09,2009.

[2] Giuseppe de Candia, Deniz Hastorun, Madan Jampani, Gunavardhan Kakulapati, Avinash Lakchman, Alex Pilchin, Swaminathan Sivasubramanian, Peter Vosshall, and Werner Vogels. *Dynamo: Amazon’s Highly Available Key-value Store*. In Proc .SOSP ’07,2007.

[3] Robbert van Renesse, Dan Dumitriu, Valient Gough, Chris Thomas and Amazon.com, Seattle. *Efﬁcient Reconciliation and Flow Control for Anti-Entropy Protocols*.In Proc LADIS ’08, 2008.

for Anti-Entropy Protocols

Dan Dumitriu

y

Valient Gough Chris Thomas

[通信量(bit)] = 224.6×n^2 + 4314.8×n

である。つまり、通信量はO(n^2)でスケールすることがわかった。また、上の式をnで割った１ノードあたり通信量はO(n)で増加することもわかった。

これらの関数から、ノード数をパラメータとしてgossip プロトコルで発生しうる全体の通信量を推測することができる。例えば、n = 1000 のとき、[通信量] = 229Mbpsとなる。このように、この関数を使って総通信量が見積もることができる。また、クラスタの設計時にも活かすことができる。ケースとして、複数のテータセンターをまたぐクラスタを構成する時を考える。この時、データセンター間のリンクはO(n^2)で通信量が増加する。つまり、ノード数が増加したときにこのリンク部分の通信が圧迫される可能性がある。このようなリンク部分を考慮したgossipプロトコルの改善が望まれる。

6.まとめ

非集中型クラウドストレージにおけるメンバーシップ管理の通信負荷を推測するために、gossip プロトコルを用いたCassandraを対象として、ノード数に応じてシステム全体の通信負荷がどのように変化するのかを実験・評価した。

この実験と評価により、定量的にクラウトストレージのgossip ベースのメンハーシップに要する通信量を計測することができ、またそれはO(n^2)で増加することがわかった。さらに複数のデータセンターをまたぐクラスタを構成するとき、データセンターをまたぐリンク部分はO(n^2)で通信量が増加し、通信が圧迫される可能性がある。このようなリンク部分を考慮したgossipプロトコルの改善が望まれる。

7.今後の研究

今後の研究としては、大きく二つを考えている。一つ目は、データセンター間のリンクを意識したgossipプロトコルの提案である。本研究により、データセンター間のリンクはO(n^2)で通信量が増加し、通信が圧迫される可能性があることがわかった。そこでこのようなリンク部分を考慮したgossipプロトコルの改善が望まれる。二つ目としては、gossipプロトコルを通信量以外の切り口でスケーラビリティ評価することである。CPU占有率や故障の伝搬スヒード等、スケーラビリティの制約になる可能性のある指標はまだ残されている。

参考文献