非集中型クラウドストレージのスケーラビリティ評価

奥寺 昇平† 長尾 洋也† 中村 俊介† 首藤 一幸†

東京工業大学†

1. はじめに

Amazon Dynamo、Cassandra 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、整合性を保つことが難しい。

そこで、Gossip Protocol 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、ノード台数が増えるにつれ、総通信量が増えるのである。よって、フロントエンド(例えばストレージであれば、データの読み書き処理。) の処理効率つまり、アベイラビリティを下げると考えられる。

しかしながら、非集中型の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 Protocol を用いる Cassandra を対象として、ノード台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 Protocol

Gossip Protocol 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。

Gossip Protocol 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、ノード台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のクラウドストレージである。Cassandra では、Gossip Protocol をベースとしたメンバーシップ管理アルゴリズムを利用している。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 ノードを複数台立ち上げる際に、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、１台あたりの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。

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

†Shohei okudera: Tokyo Instriute of Technology

5. 実験・評価

Figure 1 は、10 秒あたりのマシン間の総通信量の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。)

Figure1:ノード台数別通信量の時間変化

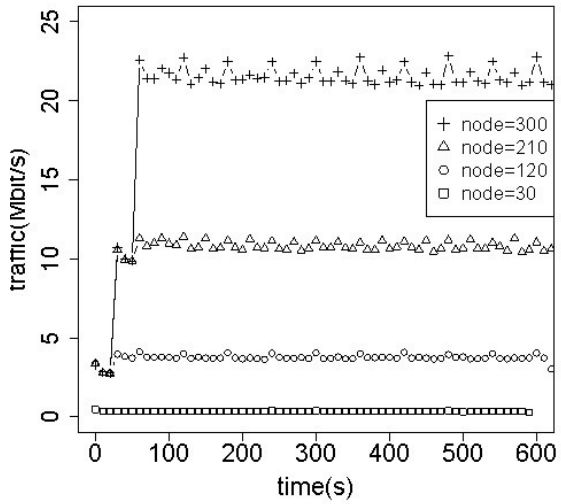
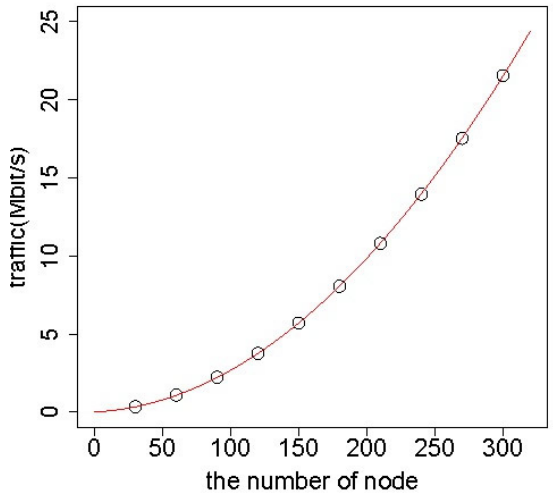


Figure2:ノード台数と通信量の変化



ノードの台数によらず、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 をノードの台数として得られた関数は、

[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。

この関数から、ノード台数をパラメータとして Cassandra の Gossip Protocol 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。クラスタ設計時にその構成するノード数に応じて、どれくらいの通信量が発生するかを見積もることができるため、ネットワークのセットアップや Gossip Protocol がそのシステムにおいて有効か否かを判断することが可能である。

6. まとめ

非集中型の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 Protocol を用いた Cassandra を対象として、ノード台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。

この実験とその評価により、二つのことがわかった。一つ目は、Gossip Protocol をベースとしたメンバーシップ管理を行うプロトコルに必要な通信量が把握できたことである。二つ目は、ノード台数に対して Gossip Protocol の通信量の見積もりが可能になったことである。

7. 今後の研究

今後の研究としては、二つを考えている。一つ目は、Gossip Protocol 以外のメンバーシップ管理を行うプロトコルの通信量を調べることである。Gossip Protocol 以外のプロトコルの通信量を知ることで、プロトコルの比較、良し悪しを通信量という観点から測ることができるようになる。二つ目としては、チャーン状態の時の通信量を計測することである。チャーン状態とは、クラスタを構成するマシンのノードの出入りが激しい状態である。チャーン状態の時の通信量の変化に加え、各ノードがもつ経路情報と現実のノード状態を比較することで、分散協調プロトコルの情報伝達度合いの性能を計測することができる。

参考文献

[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.