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

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クラウドストレージが重要性を増している。このような非集中なクラウドストレージにおいて、任意のノードが受け取ったリクエストは、データの保持を担当するノードまで転送される。そのために、各ノードが転送先候補となるノードを把握しておく必要がある。特に、応答性能を向上させるために直接担当ノードにリクエストを転送する場合、各ノードが全ノードの最新の状態を管理する必要があり、システムの整合性を保つことが難しい。

この場合、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ことで、効率よく通信を行うことが可能である。

しかし、このような管理方法では、全ノードで定期的に通信を行うため、ノード数が増えるにつれ総通信量が増加し、本来の目的であるデータの読み書き性能の低下や、メンバーシップの整合性が損なわれるなどの問題が発生する。その結果、システムのスケーラビリティを制約する要因となりうる。

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の程度の通信負荷をもたらすのかは知られていない。

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測定・評価する。

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。

gossip プロトコルでは、ランダムでノードを選択して通信を行うために、通信回数が他の伝搬手法と比べて抑えられる。よって、 各ノードが全ノードと毎回通信を行う他の伝搬手法と比べて、ノード数がスケールしやすいメリットがある。

3. Cassandra

Cassandra は、Facebook 社が開発し、Apache プロジェクト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 [3]を行っている。この管理では、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。)

Figure1:ノード数別の通信量の時間変化

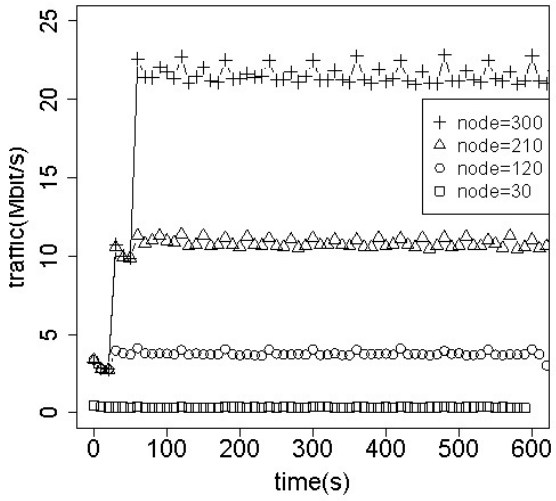
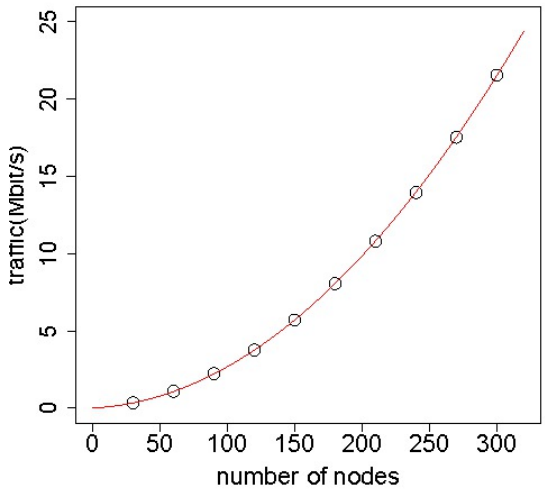


Figure2:ノード数と通信量の変化



このグラフから、ノード数によらず 100 秒以降の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。また、図中の曲線は、プロットした点から２次関数でフィッティングしたものである。ノード数を n としたとき、通信量は、

[通信量(bit)] = 224.6×n^2 + 4314.8×n

と近似でき、O(n^2)である。また、上の式を n で割った１ノードあたり通信量は O(n)である。

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] = 229 Mbps となる。これは、クラスタの設計時に活かすことができる。ケースとして、二つ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部分の通信量は O(n^2)である。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、通信のボトルネックとなる可能性がある。よって、このようなリンク部分を考慮した gossip プロトコルの応用が望まれる。

6. まとめ

非集中型クラウドストレージにおいて、通信量の観点からスケーラビリティ評価をするために、gossip プロトコルを用いる Cassandra を用いて、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 gossip ベースのメンハーシップ管理に要する通信量を計測し、またそれは O(n^2)であった。このことから、２つ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の通信量が O(n^2)であり、通信のボトルネックとなる可能性があるとわかった。よって、このようなリンク部分を考慮した gossip プロトコルの応用が望まれる。

7. 今後の研究

今後の研究としては、大きく三つを考えている。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。データセンター間をまたぐ通信のうち重複する情報を省略することで、データセンター間の通信を抑えることができるのではないかと考えている。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。

三つ目として、アベイラビリティやリライアビリティを直接到達させる方式より向上するためにノード数を非常に多くした時に、マルチホップ方式への移行を考慮したメンバーシップ管理方式を考えている。例えば、FRT [4]の導入である。

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.

[4]長尾 洋也, 首藤 一幸. "柔軟な経路表によるオーバレイネットワークのルーティング方式," *DICOMO2010*, 2010.