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†　 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。特にこれらのストレージに共通する傾向は、強い一貫性が求められていないことである。

このような非集中なクラウドストレージでは、任意のノードがデータを保持する担当ノードにリクエストを送るまでに複数のノードを経由する。そのためには各自が周囲のノードの経路情報を管理する必要がある。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をベースとした信頼性の高い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になった。

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。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一般的に論じられていない。

そこで本研究では、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計測・考察する。

2.Gossip Protocol

一般的なGossip Protocolは以下のような手順で行われる。この手順を繰り返し行う。

0. ノードPのデータXが更新されたとき、自らが更新されたことを伝える。

1. ノードPがランダムに他のノードQを選択して更新情報を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はやめる。

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

†Shohei okudera: Tokyo Instriute of Technology

Gossip Protocolに代表される伝染性のアルゴリズムを利用したProtocolの長所は、プロセス間で同期する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ノードを起動させ、マシン間で発生するトラフィックを解析した。tcpdumpを利用して、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、メモリー使用領域が増えるために、Cassandraのデータ保存部分のソースを改変し、１台あたりのメモリー使用領域を削減した。

以下に実験環境を示す。

・Cassandra 0.6.6

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

・JVM: Java SE 6 Update 21

・CPU: 2.40 GHz Xeon E5620×2

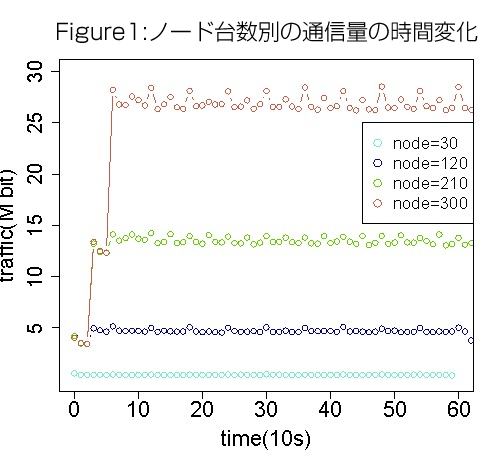
* メモリ: 32GB RAM
* Network adapter: 1000BASE-T

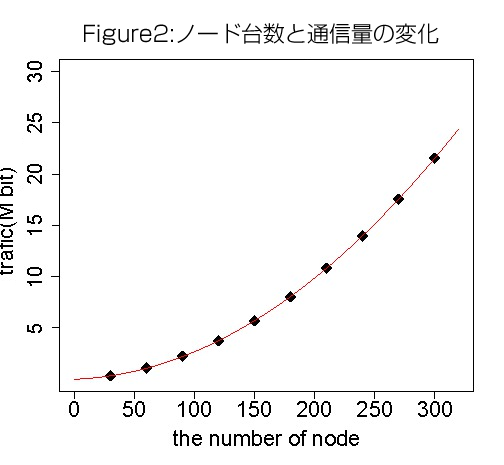
実験にはマシンを１０台使用した。

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、これを指定した台数に達成するまで続ける。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。)





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。

[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

Figure 2は、ノード数と通信量が安定した時の(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] = 224.6 \* n^2 + 4314.8 \* n

である。

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。この関数によって総通信量が見積もれることができるため、 クラスタの設計時にも生かすことができる。

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・計測した。

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語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ときである。チュ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