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分散ストレージでは、各ノードが適切なノードにホップするために各自が持つ経路情報を参照することが多い。特にクライアントからのリクエストを１ホップで対象ノードに転送するシステムでは、最新のシステム全体のノード情報整合性を保つことが必要になる。そこで、Gossip Protocolをベースとした信頼性の高く、スケーラブルな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。

ところで、クラウドストレージにおいて、このようなメンバーシップ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。

(スケーラビリティを制約する要因。メンバーシップ管理もそのひとつ、ノード内の経路表管理の仕組み。)

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。

しかし、Gossip Protocol通信量がどれくらいになっているのか、ノードが増えたときに通信量がどのようにスケールしていくのかなどは明らかではない。

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、毎秒各マシンがランダムに他のノードと自身の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。

(非集中型のクラウドストレージで使われているgossipの通信量を調べる。あきらかでない。Cassandraについても予測が立てられない。)

(Cassandra Gossip Protocol通信量について実装しだいでよくわからん。)

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。

(１台あたりノードを立ちあげるために、改善したポイント、それによって何ノードまで立ち上げられるのか？をかいた。軽量化する理由)->新しいセクションもあり。はじめ、まとめにももりこみたい。

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

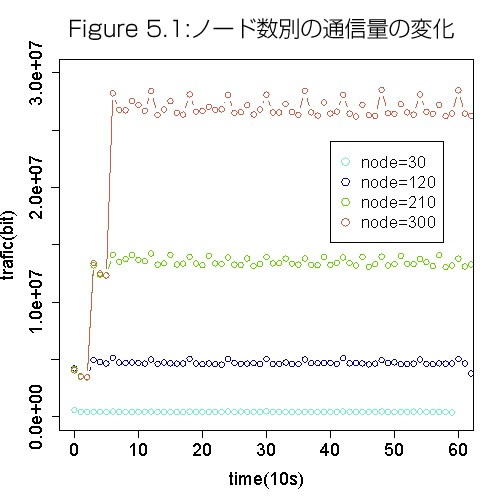
・CPU: 2.40 GHz Xeon E5620×2

* 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 5.1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*

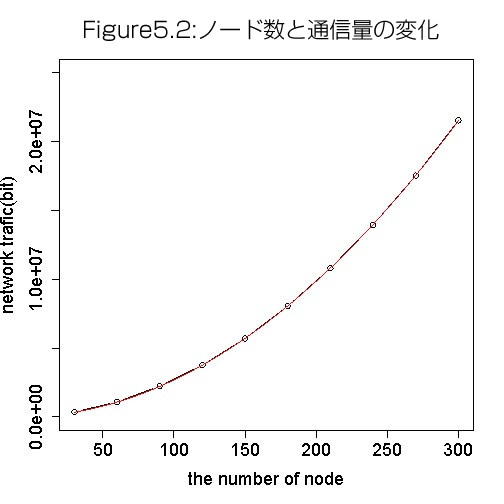
[3] Robbert van Renesse, Dan Dumitriu, Valient Gough, Chris Thomas and Amazon.com, Seattle. Efﬁcient Reconciliation and Flow Control for Anti-Entropy Protocols.In *Proc LADIS ’08 2008.*

for Anti-Entropy Protocols

Dan Dumitriu

y

Valient Gough Chris Thomas



(あまりばらつきがなく安定したと。図1,図2.○が大きほうが良い。Network traffic,文字ができるだけ大きく、Mbitにしたほう良い。10^6=1Mでよい。[一言書くとよい。],fittingした式のみで良い。両端まで伸ばす。)

Figure 5.2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。ここで得られた関数は、

y = 224.6 \* x^2 + 4314.8 \* x

である。(x,yより、m=　or ノード数)

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、x = 1000 のとき、全体の通信量y = 229Mbpsとなる。逆に、一般的なスイッチの転送速度(適切に)を1Gbpsとすれば、そのスイッチの限界値を超えるには、x = 2100台の時であると推測できる。(クラスタの設計時にも生かされる。重要である。)

6.まとめ

(ざっくり、何をやったかもかく。書かなくていいかも。バランスをみて。)

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。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