平成30年度 卒業論文

分散版jungleデータベースの性能測定方法

琉球大学工学部情報工学科

145762E 仲松　栞

指導教員 河野 真治

目 次

図 目 次

リスト目次

第1章　はじめに

1.1　研究背景

天気予報やニュース、エンタメや生活に必要なありとあらゆる情報は、インターネット上で手軽に閲覧できるようになり、また、SNS により情報の発信も気軽にできるようになった。その利便性から、パソコンやスマートフォン、タブレット端末等のメディアがますます普及し、Web サービスの利用者は増大する一方で、サーバ側への負荷は増加している。Web サービスの負荷を減らすために、データベースには処理能力の高さ、すなわちスケーラビリティがますます求められてきている。データベースの処理能力を向上させるために、スケールアウトとスケールアップの方法が考えられる。スケールアップとは、ハードウェア的に高性能なマシンを用意することでシステムの処理能力を上げることを指す。スケールアウトとは、汎用的なマシンを複数用意し、処理を分散させることでシステムの処理能力を上げることを指す。単純に処理能力を上げる方法として、スケールアップは有効であるが、高性能のマシンには限界がある。コストはもちろん、そのマシン単体では処理できない程負荷がかかる可能性がある。それに対して、スケールアウトは、処理が重くなるたびに汎用的なマシンを順次追加していくことで性能を上げるため、ハードウェア的に高性能なマシンを用意せずにすみ、また柔軟な対応を取ることができる。よって、データベースの性能を向上させる方法として、このスケールアウトが求められている。本研究で扱うスケーラビリティとはスケールアウトのことを指す。

分散データシステムは、データの整合性 (一貫性)、常にアクセスが可能であること (可用性)、データを分散させやすいかどうか (分割耐性)、この 3 つを同時に保証するこ とは出来ない。これは CAP 定理と呼ばれる。一貫性と可用性を重視しているのが、現在最も使われているデータベースである Relational Database(RDB) である。そのため、データを分割し、複数のノードにデータを分散させることが難しく、結果スケールアウトが困難になってしまうという問題がある。分断耐性を必要とする場合は、NoSQL データベースを選択する。

当研究室では、これらの問題を解決した、煩雑なデータ設計が必要ないスケーラビリティのあるデータベースを目指して、非破壊的木構造データベース Jungle を開発している。Jungle は NoSQL を元に開発されているため、分断耐性を持っている。また、Jungle は、全体の整合性ではなく、木ごとに閉じた局所的な整合性を保証している。 整合性のある木同士をマージすることで新しい整合性のある木をす繰り出すことも 可能であるため、データの伝搬も容易である。Jungle は、これまでの開発によって木構造を格納する機能をもっている。

1.2 研究目的

Jungle は現在、Java と Haskell によりそれぞれの言語で開発されている。本研究で扱うのは Java 版である。

これまでに行われた分散環境上での Jungle の性能を検証する実験では、huskell で書かれた jungle の方が、java で書かれた jungle よりも、読み込みで 3.25 倍, 書き込みで 3.78 倍の性能が確認できた。

huskell は、モダンな型システムを持ち, 型推論と 型安全により、信頼性に重きを置いてプログラミングを行う関数型言語である。対して、Java はコンパイラ型言語であり、構文に関しては C や C#の影響を受けており、プログラムの処理に関しては Huskell よりもパフォーマンスが高い言語であるといえる。よって、java で書かれた jungle が、huskell で書かれた jungle より性能が遅くなってしまった原因として、性能測定時に使用するテストプログラムのフロントエンドに Web サーバー Jetty が使用されていたことが考えられる。

本研究では、新たに Web サーバーを取り除いた測定用プログラムを用いて、純粋な java 版 Jungle の性能を測定する方法を提案する。

第2章　分散版jungleデータベース

Jungle は、スケーラビリティのあるデータベースの開発を目指して当研究室で開発されている分散データベースである。現在 Java と Haskell によりそれぞれの言語で開発されており、本研究で扱うのは Java 版である。本章では、分散データベース Jungle の構造と分散部分について触れる。

2.1　Jungleデータベースの構造

Jungle は、当研究室で開発を行っている木構造の分散データベースで、Java を用いて実装されている。一般的なウェブサイトの構造は大体が木構造であるため、データ構造として木構造を採用している。

Jungle は名前付きの複数の木の集合からなり、木は複数のノードの集合でできている。ノードは自身の子のリストと属性名、属性値を持ち、データベースのレコードに相応する。通常のレコードと異なるのは、ノードに子供となる複数のノードが付くところである。

通常の RDB と異なり、Jungle は木構造をそのまま読み込むことができる。例えば、XML や Json で記述された構造を、データベースを設計することなく読み込むことが可能である。

また、この木を、そのままデータベースとして使用することも可能である。しかし、木の変更の手間は木の構造に依存する。特に非破壊木構造を採用している Jungle では、木構造の変更の手間は O(1) から O(n) となりえる。つまり、アプリケーションに合わせて木を設計しない限り、十分な性能を出すことはできない。逆に、正しい木の設計を行えば高速な処理が可能である。

Jungle はデータの変更を非破壊で行なっており、編集ごとのデータをバージョンとして TreeOperationLog に残している。Jungle の分散ノード間の通信は木の変更の TreeOperationLog を交換することによって、分散データベースを構成するよう設計されている。

2.2　分散機構

Jungle の分散機構は、木構造、すなわちツリー型を想定したネットワークトポロジーを形成し、サーバー同士を接続することで通信を行なっている。リング型 (図 2.1 ）やメッシュ型 (図 2.2 ）のトポロジーでは、データの編集結果を他のサーバーノードに流す際に、流したデータが自分自身に返ってくることでループが発生してしまう可能性がある。ツリー型であれば、閉路がない状態でサーバーノード同士を繋げることができる為、編集

履歴の結果を他のサーバーノードに流すだけですみ、結果ループを防ぐことができる (図 2.3)。

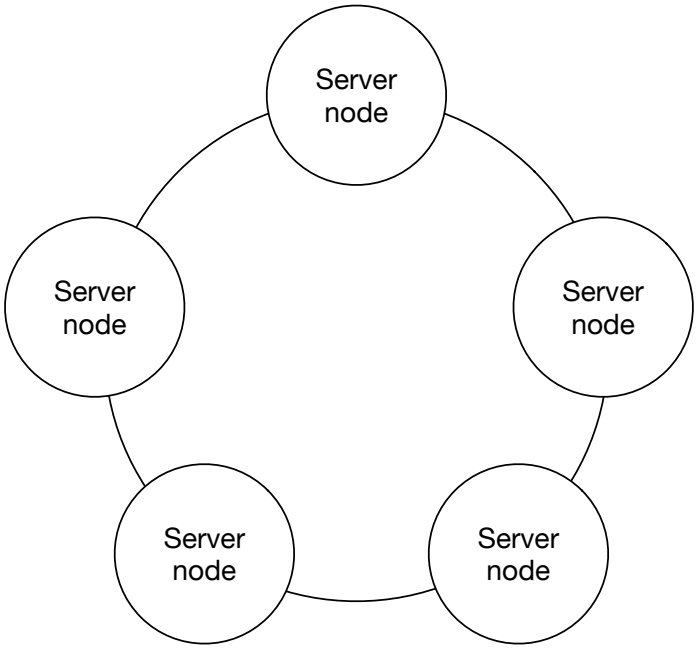

図 2.1: ring 型のトポロジー

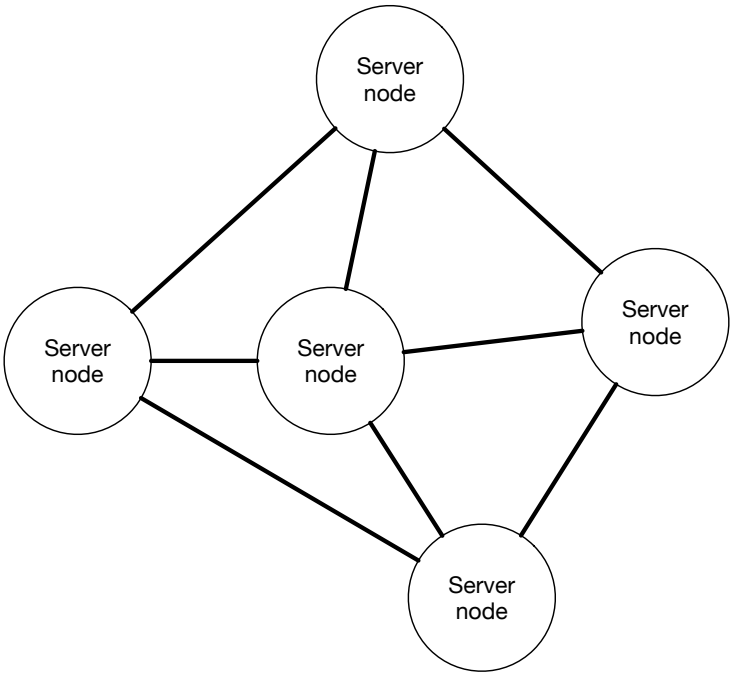

図 2.2: メッシュ型のトポロジー

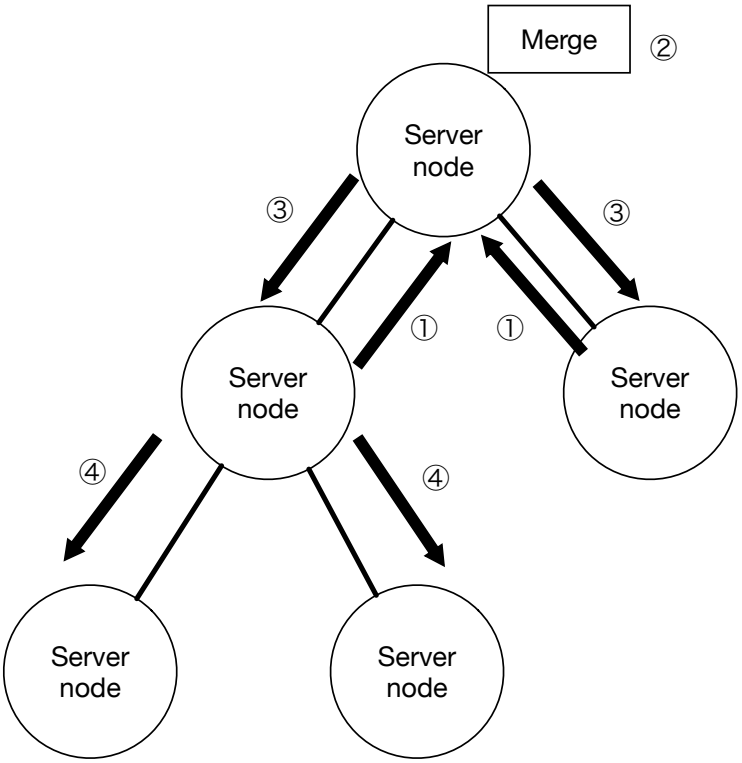

図 2.3: ツリー型のトポロジー

ネットワークトポロジーは、当研究室で開発している並列分散フレームワークである Alice が提供する、TopologyManager という機能を用いて構成されている。

また、データ分散の為には、どのデータをネットワークに流すのか決めなければならない。そこで、TreeOperationLog を使用する。前セクションでも述べたが、TreeOperationLog には、どの Node にどのような操作をしたのかという、データ編集の履歴情報が入っている。この TreeOperationLog を Alice を用いて他のサーバーノードに送り、データの編集をしてもらうことで、同じデータをもつことが可能になる。

第3章　評価実験

本研究は、Jungle の分散環境上での性能を正しく評価するための実験を行う。本章では実験の概要について述べる。まず、本研究の目的について述べ、次に、分散フレームワーク Alice による、本研究の分散機構を構成する方法について述べる。次に、木構造上に立ち上げた Jungle を制御するジョブスケジューラーである TORQUE について述べる。最後に、本実験の測定用プログラムについて述べる。

3.1　実験目的

これまでの分散環境上での Jungle の性能を測定する実験で使われたテストプログラムは、フロントエンドに Jetty という Web サーバーが使われていた。しかし、この測定方法では、Web サーバーが仲介した測定結果となってしまい、純粋な Jungle の性能を測定できないという問題がある。そこで、Web サーバーを取り除き、純粋な Jungle の性能を測定するテストプログラムを作成する。

テストプログラムは、木構造における子ノードに、データを複数書き込む機能を提供する。末端の複数の子ノードにデータをそれぞれ書き込み、最終的に root ノードへデータを merge していく (図 3.4)。

測定範囲は、

- 末端ノードから root ノードへデータが到達する時間
- 末端 Jungle から root ノードを介して別の末端ノードへデータが到達する時間

の 2 点を計測する予定である。

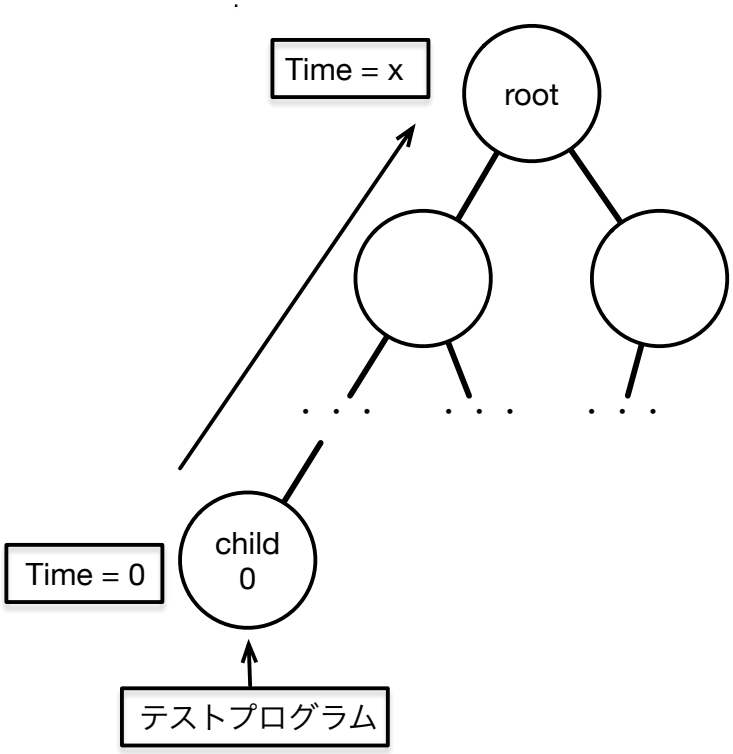

図 3.1: テストプログラムによる Jungle の性能測定

3.2 分散フレームワーク Alice による分散環境の構築

本研究では、分散環境上での Jungle の性能を確認する為、VM32 台分のサーバーノードを用意し、それぞれで Jungle を起動することで、Jungle 間で通信をする環境をつくる。Jungle を起動したサーバーノード間の通信部分を、当研究室で開発している並列分散フレームワーク Alice[1] にて再現する。

Alice には、ネットワークのトポロジーを構成する TopologyManager[2] という機能が備わっている。TopologyManager に参加表明をしたサーバーノードに順番に、接続先のサーバーノードの IP アドレス、ポート番号、接続名を送り、受け取ったサーバーノードはそれらに従って接続する。今回、TopologyManager は Jungle をのせた VM32 台分のサーバーノードを、木構造を形成するように采配する（図 3.2 ）。

トポロジー構成後、Jungle 間の通信でのデータ形式には TreeOperationLog を利用する。TreeOperationLog には、ノードの編集の履歴などの情報が入っている。TreeOperationLog を Alice によって他の Jungle へ送ることで、送信元の Jungle と同じ編集を行う。こうして、Jungle 間でのデータの同期を可能にしている。

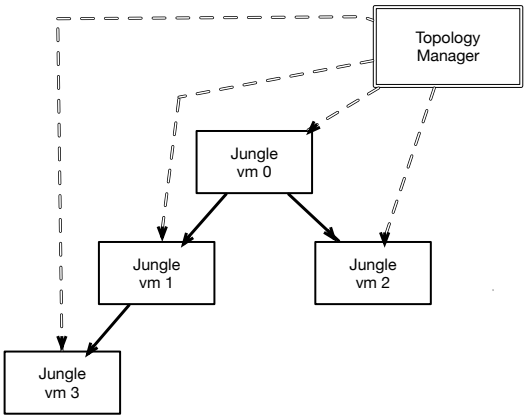

図 3.2: Alice による Jungle の木構造トポロジーの形成

3.3 TORQUE Resource Manager

分散環境上での Jungle の性能を測定するにあたり、VM32 台に Jungle を起動させた後、それぞれでデータを書き込むプログラムを動作させる。プログラムを起動する順番やタイミングは、TORQUE Resource Manager というジョブスケジューラーによって管理する。

TORQUE Resource Manager は、ジョブを管理・投下・実行する 3 つのデーモンで構成されており、ジョブの管理・投下を担うデーモンが稼働しているヘッダーノードから、ジョブの実行を担うデーモンが稼働している計算ノードへジョブが投下される（図 3.3 ）。

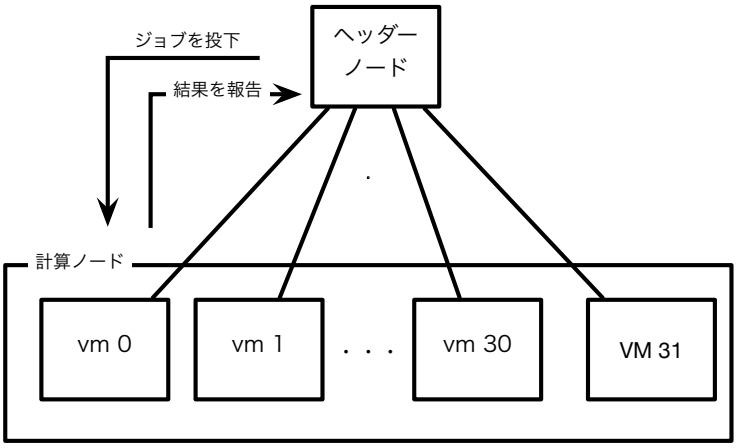

図 3.3: TORQUE の構成

ユーザーはジョブを記述したシェルスクリプトを用意し、スケジューラーに投入する。その際に、利用したいマシン数や CPU コア数を指定する。TORQUE は、ジョブに必要なマシンが揃い次第、受け取ったジョブを実行する。

3.4 Jungleの分散性能測定用テストプログラム

これまでの分散環境上での Jungle の性能を測定する実験で使われたテストプログラムは、フロントエンドに Jetty という Web サーバーが使われていた。しかし、Web サーバーが仲介した測定結果となってしまい、純粋な Jungle の性能を測定できないという問題がある。そこで、Web サーバーを取り除き、これまでの研究により純粋に Jungle の性能を測定するテストプログラムを作成する。

テストプログラムは、木構造における子ノードに、データを複数書き込む機能を提供する。末端の複数の子ノードにデータをそれぞれ書き込み、最終的に root ノードへデータを merge していく (図 3.4)。

測定範囲は、

- 末端ノードから root ノードへデータが到達する時間
- 末端 Jungle から root ノードを介して別の末端ノードへデータが到達する時間

の 2 点を計測する予定である。

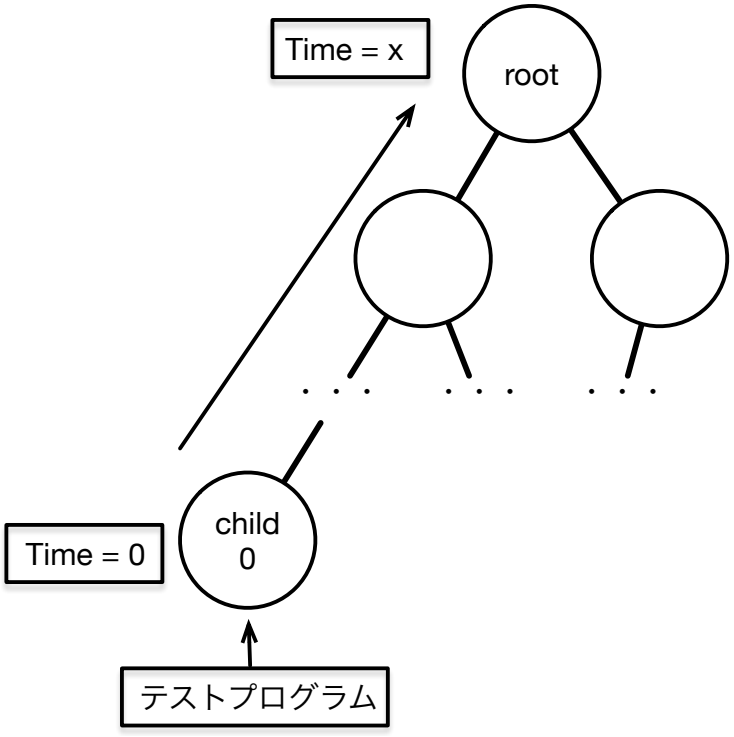

図 3.4: Test プログラムによる Jungle の性能測定

3.5 LogupdateTree.sh

LogupdateTree.sh は、Alice のトポロジーマネージャー起動後、引数で渡した数の分だけ node を立ち上げる。複数の node のうち、1 つをルートノードとして立ち上げ、残りを子ノードとして、ルートノードの下にツリー上に接続されていく。(図 3.5)

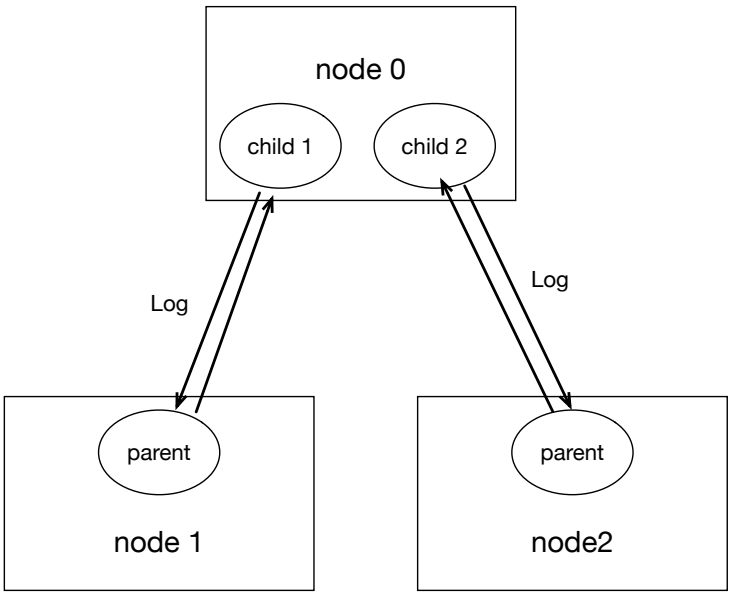

図 3.5: ルートノードと子ノードによって構成されるツリー構造

3.6 killLogupdate.sh

第4章　性能評価

4.1　java版jungleとhuskell版jungleの比較

4.2　java版jungleの分散性能の評価

4.3　性能測定方法の評価

第5章　結論

5.1　まとめ

5.2　今後の課題

参考文献

[1] 杉本 優 : 分散フレームワーク Alice 上の Meta Computation と応用,

[2] 大城 信康 : 分散 Database Jungle に関する研究,

[3] 金川 竜己 : 非破壊的木構造データベース Jungle とその評価

[4] 大城 信康, 杉本 優, 河野真治 : Data Segment の分散データベースへの応用, 日本ソフトウェア科学会 (2013).

謝辞

謝辞だよ。