

# 分散データベース Jungle の評価

145762E 氏名 仲松栞 指導教員：河野 真治

## 1 研究背景

スマートフォンやタブレット端末の普及にともない、年々 Web サービスの利用者は増加した一方で、データ量が増大し、サーバ側への負荷も増加している。これを解決するため、Web サービスには、よりシステムの処理能力を拡張する性質である、スケーラビリティが求められてきている。

スケーラビリティとは、高性能のマシンを用意したり、複数のマシンに処理を分散させたりすることで、システムの処理能力を向上させる性能を指す。本実験で指すスケーラビリティとは、後者の方である。Web サービスにスケーラビリティを付与する方法の1つに、データベースにスケーラビリティを持たせる事が考えられる。

そこで、当研究室ではスケーラビリティを持つデータベースとして木構造を持つ分散データベース Jungle を開発している。方法としては、分散環境上で複数のデータベース Jungle を起動することで、処理を分散させる。

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 [2] では、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 が使用されており、純粋な Jungle の性能は測定できていなかった。今回は、新たに改良された Jungle の性能を、Web サーバーを取り除いた Test プログラムを用いて測定することを目的とする。

## 2 分散データベース Jungle

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。

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 [3] 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っており、編集ごとのデータをバージョンとして TreeOperationLog [2] 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。

## 3 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ドに順番に、接続先のサーバードの 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 台分のサーバードを、木構造を形成するように采配する (図 1)。

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。

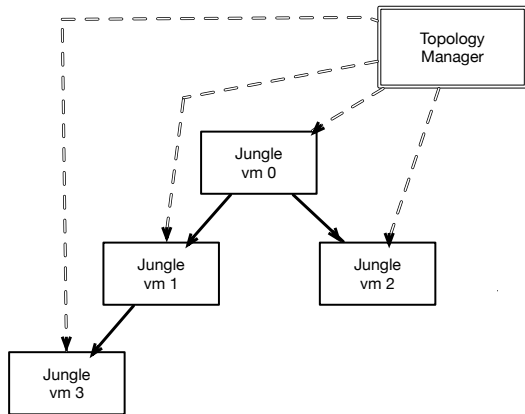


図 1: Alice による Jungle の木構造トポロジーの形成

## 4 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[1] 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 2)。

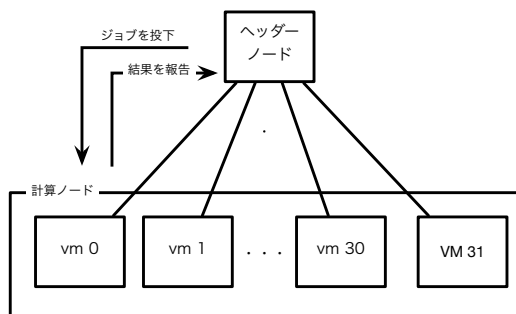


図 2: TORQUE の構成

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。

## 5 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う、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)。

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。

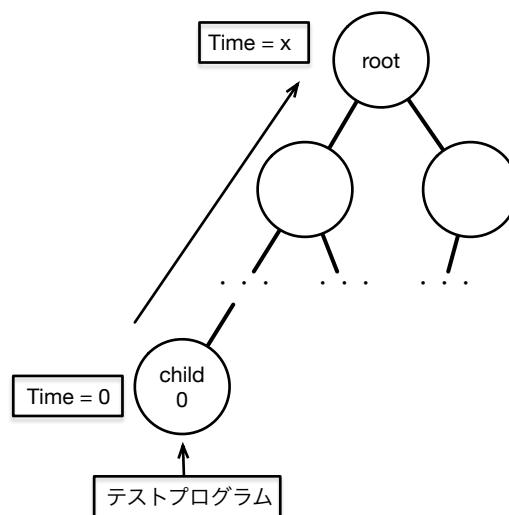


図 3: Test プログラムによる Jungle の性能測定

## 6 今後の作業

今後の方針として、本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[1] と TORQUE Resource Manager[1] を用いて、分散環境上での Jungle の性能を測定する。測定後は、結果をふまえ、Jungle の性能を向上させる為の merge アルゴリズム等を考案・実装し、スケーラビリティを持つ実用的な分散データベースの開発を目指す。

## 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).