

分散フレームワーク Alice の PC 画面配信システムへの応用

125769C 氏名 照屋のぞみ 指導教員：河野真治

1 研究目的

当研究室ではデータを Data Segment、タスクを Code Segment という単位で分割して記述する分散フレームワーク Alice の開発を行っている。本研究では、Alice 上に実用的なアプリケーションである画面共有システム TreeVNC[1] を構築する。構築するにあたり必要となった転送・圧縮などの機能を Alice の Meta Computation として実装する。そして Alice を使用していない TreeVNC との比較を行うことで MetaComputation の役割と有効性を示す。

2 分散フレームワーク Alice の概要

Alice はデータを Data Segment (以下 DS)、タスクをと Code Segment (以下 CS) という単位に分割してプログラミングを行う。DS は Alice が内部にもつデータベースによって管理されている。DS に対応する一意の key が設定されており、その key を用いてデータベースを操作する。

CS は実行に必要な DS が揃うと実行されるという性質を持ち、入力された DS に応じた結果が出力される。CS を実行するために必要な入力 DS は InputDS、CS が計算を行った後出力される DS は Output DS と呼ばれる。データの依存関係にない CS は並列実行が可能である。(図 1) 実際には Alice は Java で実装されており、DS は JavaObject で CS は RunnableThread である。

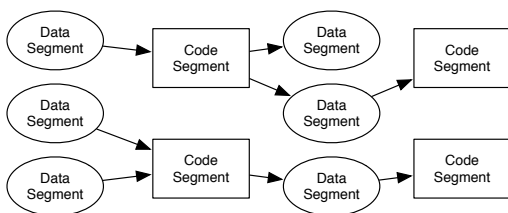


図 1: CodeSegment の依存関係

[Data Segment]

Alice はデータを分割して記述する。その分割されたデータを DS と呼ぶ。実際には整数や文字列などの基本的なデータの集まりのことを指し、Alice の場合は Java オブジェクトに対応する。CS の実行において DS は占有されるため、

Alice ではデータが他から変更され整合性がとれなくなることはない。

[Data Segment Manager]

DS は実際には queue に保存される。queue には対になる key が存在し、key の数だけ queue が存在する。この key を指定して DS の保存、取得を行う。queue の集合体はデータベースとして捉えられる。このデータベースを Alice では DS Manager (以下 DSM) と呼ぶ。DSM には Local DSM と Remote DSM が存在する。Local DSM は各ノード固有のデータベースである。Remote DSM は他のノードの Local DSM の proxy であり、接続しているノードの数だけ存在する。(図 2) Remote DSM に対して書き込むと対応するノードの Local DSM に書き込まれる。

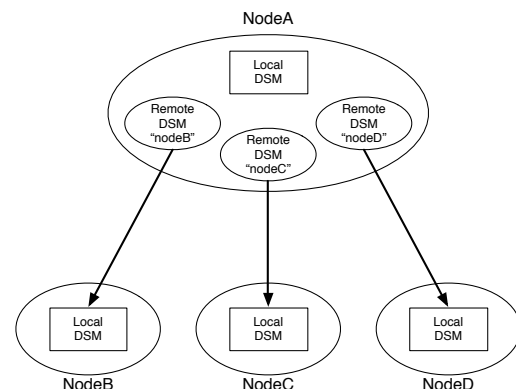


図 2: Remote DSM は他のノードの Local DSM の proxy

[Computation と Meta Computation]

Alice では、計算の本質的な処理を Computation、Computation とは直接関係ないが別のレベルでそれを支える処理を Meta Computation として分けて考える。Alice の Computation は、key により DS を待ち合わせ、DS が揃った CS を並列に実行する処理と捉えられる。それに対して、Alice の Meta Computation は、Remote ノードとの通信時のトポロジーの構成や切断・再接続の処理と言える。つまりトポロジーの構成は Alice の Computation を支えている Computation とみなすことができる。

プログラマーはCSを記述する際にトポロジーや切断、再接続という状況を予め想定した処理にする必要はない。プログラマーは目的の処理だけ記述する。そして、切断や再接続が起こった場合の処理を記述しMeta Computationで指定する。このようにプログラムすることで、通常処理と例外処理を分離することができるため、シンプルなプログラムを記述できる。Aliceの機能を追加するということはMeta Computationを追加すると言い換えられる。

3 AliceVNC

当研究室では授業向け画面共有システムTreeVNCの開発を行っている。授業でVNCを使う場合、1つのコンピュータに多人数が同時につながるため、性能が大幅に落ちてしまう。この問題をノード同士を接続させ、木構造を構成することで負荷分散を行い解決したものがTreeVNCである。Aliceが実用的なアプリケーションを記述する能力をもつことを確認するために、TreeVNCをAliceを用いて実装したAliceVNCの作成を行った。

4 Aliceの新機能

実用的なアプリケーションであるTreeVNCをAlice上で実装することで、Aliceに必要な機能を洗い出した。

[flip機能]

Data Segment APIを用いると、Output DSが毎回新しく作成され、出力するデータのコピーが行われる。しかし、Input DSとして取得したデータをそのまま子ノードにOutput Data Segmentとして出力する場合、コピーを行なうのは無駄である。そこで、Input DSとOutput DSにそのまま転送する機能をflip機能として実装することで、コピーのオーバーヘッドを減らした。TreeVNCでは親ノードから受け取った画面データをそのまま子ノードに配信するため、Meta Computationとしてflip機能が有用である。

[Data Segmentの表現の追加(圧縮機能)]

TreeVNCでは画面配信の際、データを圧縮してノード間通信を行っている。そのため、AliceVNCにも圧縮されたデータ形式を扱える機能が必要だと考えた。しかし、ただデータを圧縮する機構を追加すればいいわけではない。

AliceVNCでは、ノードは受け取った画面データを描画すると同時に、子ノードのRemote DS Managerに送信する。ノードはDSを受信するとそれを一度解凍して画面を表示し、再圧縮して子ノードに送信する。しかし、受け取ったデータを自分の子ノードに対して送信する際には、解凍す

る必要はない。圧縮状態のまま子ノードに送信ができれば、解凍・再圧縮するオーバーヘッドを無くすることができる。

そこで、1つのData Segmentに対し複数の表現を持たせることで、必要に応じた形式でDSを扱うことを可能にした。DSを扱うクラスに、次の3種類の表現を同時に持つことができる。

1. 一般的なJavaのクラスオブジェクト
2. MessagePack for Java[2]でシリアライズ化されたバイナリオブジェクト
3. 2を圧縮したバイナリオブジェクト

1はLocalDSMにDSを保存する場合に用いる。また、圧縮せずにRemoteDSMにDSを保存する場合は2のMessagePack形式で送られる。そしてLocal, Remoteに対しDSを圧縮して保存する場合は3を用いる。

さらに、圧縮状態を持つDSを扱うDSMとしてLocalとRemoteそれぞれにCompressed Data Segment Managerを追加した。Compressed DSMの内部では、DSが呼ばれた際に圧縮表現を持っていればそれを使用し、持っていないならばその時点で圧縮表現を作って操作を行う。これによりユーザは指定するDSMを変えるだけで、他の計算部分を変えずに圧縮状態を持つDSを扱うことができる。ノードは圧縮されたDSを受け取った後、そのまま子ノードにflipすれば圧縮状態のまま送信されるので、送信の際の再圧縮がなくなる。

しかし、データの表現に圧縮したbyteArrayを追加したため、Remoteから受け取ったbyteArrayのDSが圧縮されているのかそうでないのかを判断する必要がある。そこで、Aliceの通信におけるヘッダにあたるCommandMessage.classにシリアライズ状態表すフラグと、圧縮状態を表すフラグを追加した。これによってputされたDSMはフラグに応じた適切な形式でReceiveData.class内にDSを格納できる。

5 今後の課題

AliceVNCとTreeVNCを比較することにより実装したMeta Computationの有効性を示す必要がある。また、TreeVNCでは複雑な記述になってしまったNAT越えの機能を提供できると期待される。

参考文献

- [1] MIWA OSHIRO, and Shinji KONO:授業やゼミ向け画面配信システムTreeVNCの拡張機能, 琉球大学工学部情報工学科平成26年度学位論文(学士)(2014).
- [2] <http://msgpack.org/>

- [3] Yu SUGIMOTO and Shinji KONO: 分散フレームワーク Alice 上の Meta Computation と応用, 琉球大学工学部情報工学科平成 26 年度学位論文 (修士) (2014).