Plan9 と分散シェルによるコマンドベース MapReduce

中原 健志1　　並木 美太郎 1

1. 研究背景

大量のデータを処理する分散処理のプログラミングモデルとして MapReduce[1]が広く利用されている．多くの MapReduce の実装ではプログラムを MapReduce 向けに書き換える必要があり，開発コストが高い．本研究では処理させたい内容を粗粒度な単位に分解し，それぞれを一つの入力と出力を持つコマンドとして実装し，それらをパイプで繋ぐことで分散処理を実現することを検討した．この手法では既存のコマンドを応用可能なため，比較的短い開発期間で分散処理の効果を得ることが出来る．関連研究[2][3]では，ローカルマシンとリモートマシン間でのファイルやディレクトリの違いがあるため，利用者は事前に処理に必要なファイルをリモートのマシンからアクセスできるように準備する必要があった．この問題について，リモートマシンの名前空間をローカルマシンの名前空間と重ね合わせ，ローカル・リモートマシン間で透過にファイルへアクセスすることを可能にすることで解決する．ネットワークを介したマシン間での名前空間の重ね合わせは，多くの OS では実現が難しい．そこで本研究では位置透過性とアクセス透過性等の分散透明性の機能を持つ OS「Plan 9[4]」を用いて実現する．

2. システム構成

本研究で開発している分散シェルは利用者からのコマンドの入力受付，入力内容の解釈・変換を行う分散シェル本体，及びリモートマシンの管理を行うリモート管理デーモンから成る（図 1）．利用者は初めに利用可能なリモートマシンを分散シェル本体へ登録する．その後，分散シェル本体上で行いたい処理を記述する．リモートマシンで実行させたい処理に対しては，本シェルで定義した記号（//|, //{, //}）を用いて記述する．分散シェル本体は記述内容を解釈し，必要に応じて登録されているリモートマシンへ接続し，リモートマシン上でリモート管理デーモンの立ち上げを行う．分散シェル本体，及びリモート管理デーモンの役割について，下記に述べる．

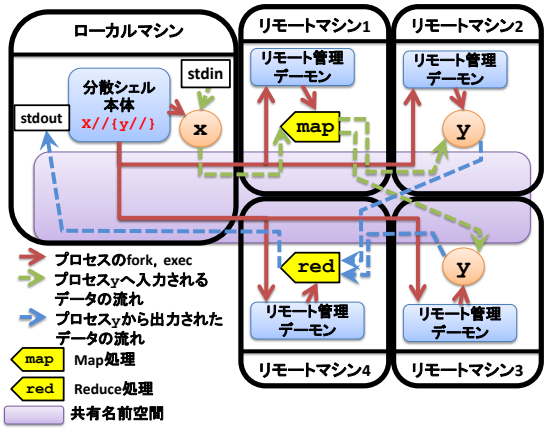

図 1　Map・Reduce 実行時の分散シェル内部の処理構成

- **分散シェル本体**

分散シェル本体はローカルマシン上で動作し，入力されたコマンド列の解釈を行い，必要に応じてコマンドを実行するリモートマシンの選定，及びリモート管理デーモンの立ち上げを行う．リモートマシンの選定では，リモート管理デーモンが提供するリモートマシンの情報を基に，タスクの管理，及び Map･Reduce 処理を行うマシンを選定する．また，コマンド間の標準入出力の管理を行う．

- **リモート管理デーモン**

リモート管理デーモンはリモートマシン上で動作し，Plan9 の透過性の機能を用いてローカル・リモートマシン間で共有名前空間の生成を行う．共有名前空間を生成することで，リモートマシンからローカルマシンのファイルへ透過にアクセス可能である．これにより処理に必要なファイルがリモートマシン上に存在しない場合でも，利用者はファイルの有無を気にせず，処理をリモートマシン上で実行することが出来る．また，リモートマシン上で動作するコマンドの実行・管理を行う．そのほかに，

1 東京農工大学
Tokyo University of Agriculture and Technology

Plan9 の透過性の機能を用いてリモートマシンの情報をファイルインタフェースに仮想化し，共有名前空間へマウントする．これにより，ローカルマシンからリモートマシンの状態を容易に取得することが可能となる．

3. 分散シェルの文法

分散シェルではシェルの記号に追加した 3 つの記号（//|，//{，//}）を用いて，リモートマシン上で処理させたいコマンド列を指定する．追加したそれぞれの記号の処理について説明する．

- **リモート記号： //|コマンド**

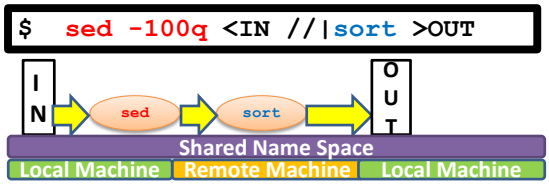

図 2 リモート記号での sort コマンド実行

リモート記号の次に書かれたコマンドは，1 台のリモートマシン上で実行される．この時，どのリモートマシン上で実行するかは分散シェルが自動的に決める．図 2 の例では，コマンド「sed -100q」はローカルマシン上で実行される．次のリモート記号で指定されたコマンド「sort」は，分散シェルが選定した 1 台のリモートマシン上で起動される．この際，それぞれのコマンドの入出力は分散シェルにより，自動的に繋がれる．

- **MapReduce 記号：**
  //[オプション]{ コマンド //[オプション]}

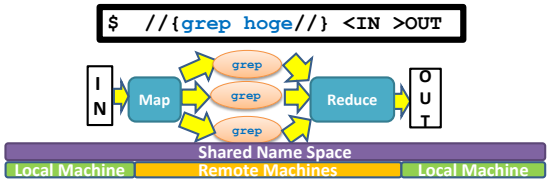

図 3 分散記号での grep コマンド処理

MapReduce 記号で囲まれたコマンドは，利用可能な台数のリモートマシン上で分散実行される．この時，分散シェルは MapReduce 記号の前にあるコマンドからの標準出力を分割する Map 処理を行い，分割されたデータをそれぞれのリモートマシン上で実行されるコマンドの標準入力へ渡す．それぞれのリモートマシンで処理され，標準出力へ出された結果は分散シェルが Reduce 処理を行い一つのデータにまとめ，MapReduce 記号の後ろにあるコマンド・出力先へ渡す．オプションがない場合，多くのフィルターコマンドはテキスト処理を対象として設計されていることが多いため，Map・Reduce 方法はテキストの改行をデータの区切りとして処理する．ただし，処理の内容によって，適切な Map・Reduce 方法は異なるため，利用者で適切な方法をコマンドとして実装し，オプションで指定することも可能である．図 3 の例では，MapReduce 記号で囲まれたコマンド「grep」は複数のリモートマシン上で起動される．入力データは Map 処理により改行単位で分割され，それぞれのリモートマシン上の「grep」へ渡される．処理された結果は分散シェルにより Reduce 処理が行われ，一つのデータに纏められて出力される．

4. 現状と今後

本稿ではコマンドベースの MapReduce を実現する分散シェルについて述べた．本シェルの実現で問題であったローカルマシンとリモートマシン間のファイルやディレクトリの差異について，Plan9 が持つ透過性の機能を用いて解決する．試作した分散シェル上で単語の出現頻度表生成による性能評価を行った．結果，リモートマシン 2 台で 2 倍，3 台で最大の 2.5 倍の性能向上を確認した．現在，Plan9 の標準シェルである「rc」へ分散シェルの機能の実装を行っている．また，リモートマシンへのタスク割り当てについて，ネットワークやマシンの負荷状況を考慮したタスク割り当てアルゴリズムの調査と実装を進める．実装完了後に性能やスケーラビリティなどの評価を行う．

参 考 文 献

[1] Jeffrey Dean and Sanjay Ghemawat. 2004. MapReduce: simplified data processing on large clusters. In Proceedings of the 6th conference on Symposium on Opearting Systems Design & Implementation - Volume 6 (OSDI'04), Vol. 6. USENIX Association, Berkeley, CA, USA, 10-10.

[2] Tange, Ole. "Gnu parallel-the command-line power tool." The USENIX Magazine 36.1 (2011): 42-47.

[3] Noah Evans and Eric Van Hensbergen (IBM Research), PUSH, a dataflow shell, In Proceedings of the 5th ACM SIGOPS/EuroSys European Conference on Computer Systems 2010, pp.14-16, Paris, France, April 2010.

[4] R. Pike, D. Presotto, S. Dorward, B. Flandrena, K. Thompson, H. Trickey, and P. Winterbottom. Plan 9 from Bell Labs. Computing Systems, 8(3):221--254, Summer 1995.