DCAST: 自己組織型 PC クラスタ構築システム
-資源追加が容易に行える初心者向け PC クラスタ構築システム-

1. 背景

近年，科学技術分野における大規模な計算シミュレーションやデータ解析，ビジネス分野におけるデータマイニングといった極めて大きな計算量を要求する問題が増加している．このような大規模計算を行うためにはスーパーコンピュータや専用計算機が必要とされてきたが，導入コストが極めて高いことから誰にでも容易に利用できるものではない．

一方で，コモディティなハードウェアの高性能化と低価格化の流れから，PC をネットワークで接続した並列計算機であるPCクラスタが注目されている．PC クラスタは同程度の性能を持つスーパーコンピュータと比較して，コストパフォーマンスに優れているという利点を持つ．現在では企業や研究所に2000ノードといった大規模なシステムが導入された例もあるが，こういった大規模化にしたがって構築，管理のためのコストが無視できない問題となってきた．

そこで，本プロジェクトでは PC クラスタの構築，管理を容易にするソフトウェア Dynamic Cluster Auto Setup Tool（以下 DCAST）を提案する．

2. 目的

DCAST は適切に接続された PC ノードに対して，OS およびクラスタリング用ミドルウェアの自動インストール機能を提供する．また PC クラスタの構築に際しては各ノードのスペックに応じたアーキテクチャ（論理な接続関係）を決定する必要があるが，DCAST では起動時に各ノードを調査して最適な構造を自動的に判断する．アップデートの際にはノード間の一貫性を保証するためにシステム全体の再インストール（ユーザデータ領域を除く）を行い，その際にノードの追加・変更等がなされていればアーキテクチャは再び変化する．これにより，管理コストの大幅な削減や，遊休資源の容易な追加が可能になる．

3. 開発の内容

DCAST システムは Service Server，Master ソフトウェアおよび Client ソフトウェアの 3 つの要素から構成されている(図 1)．インターネット上の Service

Server は Debian ミラーを兼ね，ノードの自動インストーラである Master および Client からの要求に従って必要なパッケージを送信する．ユーザは最初に Master を用いてマスタノードを構築する．次に Client を用いてスレーブノードを起動すると，自ノードのパフォーマンスを計測し，その情報をマスタノードに送信する．Master はこれらの情報から一定のルールに基づいてアーキテクチャを決定し，それぞれのスレーブノードに必要なディスクイメージを作成，ノードごとに必要な設定を行ったうえでハードディスクを持つスレーブノードにコピーする．これにより，対話的な作業を必要とせず，しかもバタフライ通信を用いたコピーを行うなどの高速化手法によって 60 ノード規模の大規模 PC クラスタの構築時間を約 25 分という短時間で構築することが可能であった．

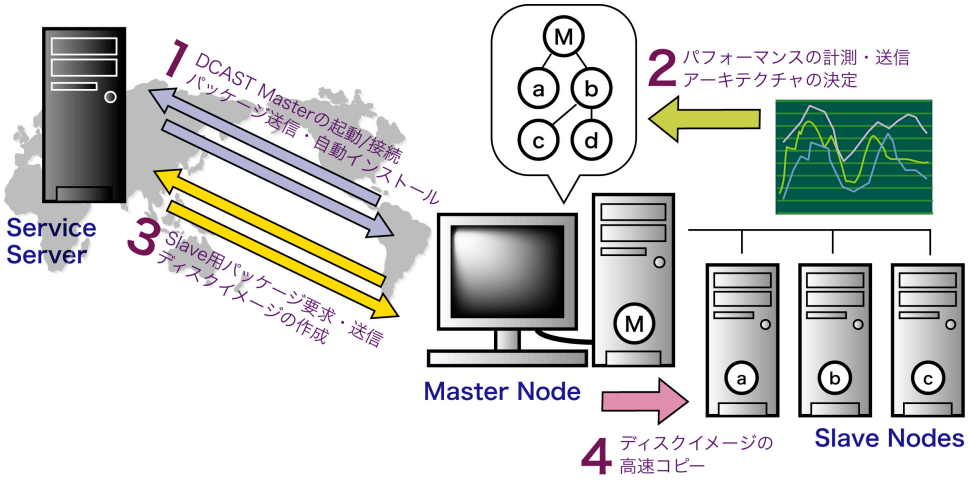

図 1：DCAST の概念図

4. 従来の技術（または機能）との相違

PC クラスタ構築ツールは商用，フリーソフトウェアを含めいくつか存在するが，以下の点において DCAST に優位性があると考えられる．

- 自己組織化機能：この機能により遊休資源の容易な活用が可能である．
- 設定ファイル自動生成：既存の PC クラスタ構築ツールには，その設定ファイルの記述方法が難解なものが多く，PC クラスタ初心者にとって大きな壁となっているケースが多い．DCAST は update-cluster の機能を利用して，設定ファイルを自動生成することができる．

・ 機能拡張性がある：既存の PC クラスタツールはソフトウェア構成がすでに決定されているものがほとんどであり，ユーザが機能拡張することが難しい．そこで DCAST はユーザがモジュールスクリプトを作成することにより，機能拡張できる構成になっている．

5. 期待される効果

一般に，PC クラスタの構築には 1 ノード数 40～50 分の時間を要し（ディスクフルノードの場合），また操作が対話的であることから，構築を行うユーザはその間常に何らかの作業を要求されることになる．この場合，8 ノードの PC クラスタではのべ 6 時間以上，100 ノードを超える PC クラスタでは 3 日程度の時間が必要となる．スレーブノードの構築を HDD のディスクコピーによって行う場合には対話的な処理は減るものの，コピーに要する時間が大幅に増大し，また最終的な設定は人間の手で行う必要があるために全体的なコストは低下しない．また，システムの安定性・一貫性を高めるためには，アップデートの際にも OS やソフトウェアを全システムに再インストールすることが望ましいが，これらの作業を何度も繰り返すことは現実的でない．これらの理由から，大規模なクラスタでは管理を納入業者に任せることが多いが，この方法では金銭的コストの負担が大きくなるばかりでなく，ユーザの要求を迅速に満たすことが難しくなる．

DCAST は初期の導入時にもアップデート時にもシステム全体の（再）インストールを行うために総作業時間はあまり変化がないが，スレーブノードの構築ではバタフライ通信による同時コピーを行うことから，実質的な作業時間は 100 ノードの場合で 1/10 程度に短縮されると思われる．しかも，その間には対話的な作業は極めて少ないため，頻繁にメンテナンスや新しいソフトウェアをインストールすることができる．

また各ノードを構成する PC がそれぞれ異なったスペックである場合には，クラスタの論理的構造（アーキテクチャ）がシステム全体のパフォーマンスを大きく左右する．DCAST ではこの論理的構造をハードウェア構成（CPU の速度，ディスクフルノード/ディスクレスノード）やネットワークの距離，ノードにかかる負荷（ディスクフルノードが何台のディスクレスノードを受け持っている）等から自動的に決定するため，設計ミスによるボトルネックの発生を最小限に抑えることが可能である．また，新しいノードを追加して PC クラスタを

再構築するとそれにあわせて構造が変化する．この機能により，遊休資源をクラスタに接続し，有効活用することができる．

6. 普及（または活用）の見通し

DCAST のプロトタイプ版を DCAST ホームページ(http://www.dcast.org)で公開を行っている．今回作成した自己組織化機能を持つ DCAST を 3 月中に DCAST ホームページに更新し，Debian-Beowulf プロジェクトに通知を行う予定である．なお，DCAST のライセンスは GPL である．

今後の DCAST の拡張予定として，機能拡張が行えるモジュールを作成する予定である．それはフォールトレランス機能を持ったジョブスケジューラや，各マシンの負荷状況がわかるモニタリングツール等である．

7. 開発者名（所属、e-mail アドレス）

開発者：中尾昌広，同志社大学大学院工学研究科知識工学専攻，mnakao@mikilab.doshisha.ac.jp

共同開発者：澤田淳二，（同上），junjis@mikilab.doshisha.ac.jp