The Science and Engineering Doshisha University, Vol.43, No.4 January 2003

# DCAST: A Simple Installation and Administration Tool for the Large-scaled PC Cluster System

Kenzo KODAMA* Tomoyuki HIROYASU** Mitsunori MIKI** Yusuke TANIMURA* and Junichi UEKAWA*

(Received October 18, 2002)

In this paper, a new setup/administration tool for PC cluster systems is proposed. Recently, in a high performance computing field, PC cluster systems have been focused in research. PC cluster systems consist of PCs connected via a network and are used for parallel/distributed computing. The PC cluster yields low priced performance due to using commodity hardware to set up the cluster. However, it is very hard to install and configure a PC cluster because many nodes existe and abundant knowledge is required for the installation and configuration of the cluster. In this study, to solve this problem, a simple installation and administration tool for PC cluster called "Doshisha Cluster Auto Setup Tool: DCAST" is developed. DCAST has the following features; DCAST can be used for both diskless and diskful clusters. DCAST is targeted for Linux but is not dependent on a certain distribution of Linux. There is no interaction procedure during the installation. Slave nodes are booted over the network. When the system is configured, the whole system is reinstalled. The differences of functions and behaviors between existing PC cluster systems setup software and DCAST are compared and examined.

**Key words** : PC cluster, parallel computing

**キーワード** : PC クラスタ，並列計算

# DCAST: 大規模クラスタにおける簡易セットアップ・管理ツールの提案

児玉憲造・廣安知之・三木光範・谷村勇輔・上川純一

## 1. はじめに

近年，科学技術計算分野において大規模計算の要求が高まっており，できるだけ安価で，高性能な計算機が望まれている．現在，PC クラスタは安価に導入できる高性能計算機の一つであり，Beowulf project[1, 2] 等に代表されるようにさまざまなシステムを構築する取り組みも行われている．クラスタとは「群れ，房」をさす言葉であり，クラスタシステムとは単一で稼動するコンピュータの集まりで，一つの計算資源として使用可能な並列もしくは分散システムのことを指す．

数年前までは，大きな計算パワーを得るためにはスーパーコンピュータと呼ばれる特別なハードウェア構成を持った専用計算機が必要であった．しかし近年，並列計算の研究の進歩によりクラスタという考えが生まれ，汎用の PC をクラスタ化することにより，安価に並列計算環境を導入することが可能となった．このことにより個人レベルでの並列計算機を導入すること

* Graduate Student, Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6921, Fax:+81-774-65-6921, E-mail:kenzo@mikilab.doshisha.ac.jp
** Department of Knowledge Engineering and Computer Sciences, Doshisha University, Kyoto
Telephone:+81-774-65-6930, Fax:+81-774-65-6780, E-mail:mmiki@mail.doshisha.ac.jp , tomo@is.doshisha.ac.jp

ができ，従来では解明できなかった複雑な問題の解明に多くの人が取り組むことが可能となり，大きな期待を集めている．

このように，比較的安価で導入可能な PC クラスタだが，システムの構築や使いやすい環境設定，高性能計算能力を引き出すこと等は容易ではなく，多くの場合大変な労力を必要とする作業となる．また，システム構築後の管理においても台数が多い場合には，管理労力は増大する．

一方，PC クラスタでは多くの場合ソフトウェアの設定ファイルは全ノードで同じであり，またその設定もホスト名や IP アドレスなど決まった項目に限定されている．我々はこういった点に着目しクラスタリングソフトウェアの開発を進めている．

本研究では代表的なクラスタリングソフトウェアに必要な機能の一つである「システム構築」「システムアップデート」を行うソフトウェア DCAST(Doshisha Cluster Auto Setup Tool) を開発した．DCAST では LINUX を OS とするクラスタシステムの構築を対象とし，極力ユーザーとの対話をインストールの際に必要としないアプリケーションの構築を目指す．また，アップグレードの際にはシステムの部分，部分のバージョンアップを行うのではなく，システム全体の再インストールを行う方式をとる．

本論文では，まず DCAST の必要性と目的について述べ，提案する DCAST のユーザ側の操作手順，システム全体の動作について述べる．そして既存のクラスタセットアップツールとの比較検討を行う．

## 2. PC クラスタの特徴と問題点

各ノードが PC で構成されているクラスタシステムを本論文では PC クラスタと呼ぶ．PC クラスタの基本的な構成図を Fig. 1 に示す．この図にあるように，PC クラスタでは Master となるノードのみが外部と

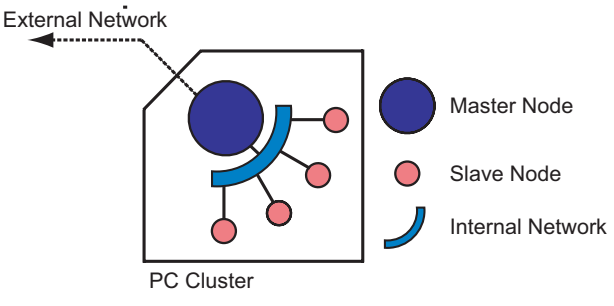


Fig. 1. 基本的な PC クラスタの構成.

の通信が行えるようになっており，Slave となるノードについては内部ネットワークハブによって外部からはログインできないようになっている．

PC クラスタは，近年の計算機技術のなかではもっとも注目されている技術の一つであり，その特徴の一つに低価格で導入可能であることがあげられる．従来，並列計算を行うためには高価格で特別なアーキテクチャを持っているスーパーコンピュータが必要であった．これに対し，PC クラスタは市場製品を取り入れ，一般的なアーキテクチャで構成する．このことから市場の最新技術を取り入れることができる．また一般的な製品で構成されることから低価格での導入が実現できる．OS にも市販製品やオープンソースプロジェクトなど，一般の PC で利用できるものを採用することが可能であるため，スーパーコンピュータよりも使い勝手がよい．

しかし，従来まではスーパーコンピュータと比べ性能面で大きく差をつけられていた．しかし近年のコンピュータ部品の高性能化に伴い PC クラスタはスーパーコンピュータに劣らない計算能力を有することとなった．これにより価格対性能比が非常に優れたシステムを構築可能となったため PC クラスタに対する注目が高まった．

このようにさまざまな利点を持っている PC クラスタだがいくつかの問題点もある．問題点の一部を以下に示す．

- 導入労力
  PC クラスタを構成する各ノードにはそれぞれ OS が必要となる．数百台規模の大規模な PC クラスタになると，OS インストールやシステム設定だけで膨大な時間が必要となる．

- 保守管理コスト
  クラスタを構成する各ノードはそれぞれが一つのコンピュータとして動作しているため，一つ一つに通常の PC と同様の管理 (ソフトウェアのインストール，ハードウェアのチェック，システムのアップグレード，故障時の対応等) が必要となる．また並列計算機としての管理 (ネットワークの設定，並列計算ライブラリのインストール，ネットワークソフトウェアやミドルウェアの導入) も必要となる．

- ノードの追加
  PC クラスタは市販 PC をそのままネットワーク

で結合するというシンプルな構成であるため，プロセッサ数やネットワークの構成などにおいて非常に自由度が高い．しかし，ノードを追加する際もOSからインストールしなおさなければならないため構築に時間がかかる．

- システムアップデートの手間
  新たなハードウェアの追加や，ソフトウェアのアップデートなどでクラスタシステム全体をアップデートする機会は非常に多い．特に安価な計算環境を構築するためにOSやソフトウェアにオープンソースプロジェクトの無料の製品を使用することが多い．このようなオープンソースプロジェクトでは頻繁なアップデートがなされる．またPCの市場ではより高性能の製品が次々とリリースされる．これらのソフトウェアや製品は性能に直接関係するため，長期間高性能を出すためには頻繁にアップデートをする必要がある．そのためノード数が多くなるとアップデート作業は大きな労力となる．

これらの問題点を解決するためにさまざまなソフトウェアが存在するが，それぞれが独自性を持っているため統合的なシステムはない．すなわちそれらのソフトウェアでは想定した環境でのクラスタセットアップしかできず，カスタマイズができない．しかしクラスタシステムを持つ人の要求はさまざまであるため，カスタマイズができるシステムが必要となる．

## 3. クラスタリングソフトウェア

一般に，クラスタを構築するには，OSやネットワークに関する高い技術力と豊富な知識が必要となる．しかし，並列計算環境のみを求めるユーザーにとって，クラスタ構築に関する知識は不要であり，構築を自動で行えるソフトウェアの要求が高まってきた．また，Linuxベースのクラスタの普及，一般化にともない，構築や管理にかかるコストを下げたいという要求も増大している．

そこで，多くの企業や研究所がクラスタリングソフトウェアと呼ばれるツールの提供を行っている[3, 4, 5, 6, 7, 8]．クラスタリングソフトウェアとは，クラスタを構築して利用するための各種ソフトウェアのことをさす．これらは企業が提供しているものは商用ベース，大学や研究機関が提供しているものはフリーで提供されていることが多い．また，多くの研究機関において自前のクラスタの構築，管理には独自のツールを用いて管理しており，標準的に用いられているツールは現在のところ存在しない．既存のソフトウェアについて，Table 1に示す．

Table 1. 主なクラスタリングソフトウェア．

| | |
|---|---|
| フリー | Rocks(NPACI)[3]<br>OSCAR(Open Cluster Group)[4]<br>SCore(PC Cluster Consortium)[5]<br>LUCIE(東工大松岡研究室)[8] |
| 商用ベース | Scyld(Scyld)[6]<br>Scali(Scali)[7] |

### 3.1 既存ソフトウェアの問題点

既存のクラスタリングソフトウェアにはいくつかの問題点が存在する．

- Linuxディストリビューションの問題
  既存のクラスタリングソフトウエアはRedHatベースのものが多い．RedHatではネットワークを利用したシステムのスムーズな更新ができない．すなわちセキュリティ関係のアップデートや新たなソフトウェアの導入など頻繁にアップデート作業を行うことが必要なものに関しても困難である．また多くのクラスタリングソフトウェアはRedHat以外のディストリビューションに対応していないためディストリビューション選択の自由がない．

- クラスタ構成が固定されている問題
  クラスタはネットワーク構成やノードごとの使用方法など自由にシステム構成が可能である．しかし，既存ソフトウェアではある想定環境下のクラスタしか構築することができない．すなわち，ノードの追加，削減が容易ではない．

### 3.2 クラスタリングソフトウエアに必要な機能

上記の問題を解決するために，クラスタリングソフトウェアには次のような機能が必要である．

#### 3.2.1 簡易なクラスタ構築システム

システム構築が容易 (自動) であることが必要となる．これは並列計算のみを行いたいユーザにとってクラスタの構築は不必要であるためである．素早くシステムの構築を行ったり，ユーザの入力ミスによるシステムの構築ミスを防ぐためにも自動で行えることが望ましい．Myrinetなどの専用ハードウェアやMPI等

の通信ライブラリ等の導入も全ノードに自動でインストールする機能が求められる．

#### 3.2.2 簡易なクラスタアップデート構築システム

Linuxをベースとするシステムでは各々のモジュールのバージョンアップが頻繁に行われる．特に通信ライブラリは頻繁なアップデートが行われることが多いため何度も利用する重要な機能であるといえる．そのため，容易にクラスタをアップデートする機能が必要となる．

#### 3.2.3 ノードジョブ管理の一元化

全ノードの管理やユーザのジョブ管理なども一元管理できることが必要となる．例えば，カーネルのバージョン管理や組み込まれているモジュールの管理，インストールされているパッケージソフトウェアの管理，ノードや通信の状態の管理，故障からの迅速な復旧，ユーザの投げるジョブのスケジューリング等が必要となる．これらはGUIによるモニタリング等，管理者やユーザが視認できるツールによって常に監視できることが望ましい．

#### 3.2.4 並列処理のためのツールの提供

クラスタは計算サーバとして稼働するため，リモートで操作できるようなRPCソフトウェアを導入することが多い．これによって悪意のあるユーザがシステム破壊するようなファイルの実行や読み書きを行う可能性があるため，セキュリティに関する機能も求められる．

またクラスタシステムで並列処理を行うためには，各ノードにおいてファイルコピーや実行，プロセスの起動，監視，停止などの処理を頻繁に行う．これらの処理を迅速に行うためのツールが必要となる．

このように，クラスタリングソフトウェアにはさまざまな機能が求められる．今回はこれらの求められる機能の中からシステム構築，システムアップデートを容易に行えるツールの開発を目的としたツール，DCASTの構築を行った．

### 3.3 DCASTの設計指針

本研究で提案するDCASTは以下の設計指針で構築している．

1. OS，ハードウェア要件
   OSはサーバ用途で使われることの多いLinuxを対象とし，ハードウェアは一般的なコンピュータのアーキテクチャであるx86アーキテクチャを対象とする．DCASTを用いたPCクラスタ構築では上記の要件を満たすことで可能となる．

2. ディストリビューション非依存
   OSをLinux対象としたが，そのディストリビューションは，特定のものに依存しないこととする．現在，Linuxは一般に普及しつつあるが，そのディストリビューションはさまざまである．先に述べたように，現在のクラスタリングソフトウェアはRedHatベースのものが多く，他のディストリビューションで使用できるものが少ない．DCASTは特定のディストリビューションに依存しないよう設定ファイルのパス設定を変えることによってどのディストリビューションでも動作することとする．

3. インストール時の対話的操作の排除
   Linuxに限らず，OSのインストールには多くの対話的操作が必要となる．一台のコンピュータとして動作させるのであれば問題ないが，PCクラスタのように大規模なシステムを作成する場合，対話操作は大きな労力となる．DCASTは対話操作を一切行わないことを目標とする．

4. バージョンアップ時はクラスタを再インストール
   PCクラスタは，最新の市場技術をそのまま取り入れることが可能であるため，頻繁なアップデート，アップグレードを行うことがある．このようなバージョンアップを行った場合に様々なファイルの書き換えが行われることが多いため，大きなバージョンアップを行う際には全てを再インストールすることとする．

5. ディスクレス，ディスクフルとも作成可能
   クラスタの形態に，基本的にハードディスクを持たず，より低価格で導入できる「ディスクレスクラスタ」がある．また通常のコンピュータを複数台接続した形態のクラスタを，本論文では「ディスクフルクラスタ」と呼ぶ．DCASTではそのいずれも構築可能とする．

## 4. クラスタインストーラの提案

### 4.1 クラスタシステム構築手順

PCベースのクラスタを作成するには，少なくとも以下のような手順を踏まなければならない．

```
#choice diskless or diskful
SET     diskless
#Enter SWAP size.
SWAP    256
#       NETWORK         NETMASK         BROADCAST
NET     192.168.0.0     255.255.255.0   192.168.0.255
#MASTER bootserver's name       bootserver's IP
MASTER  host01                  192.168.0.1
#slave's name   slave's IP      slave's MACaddress
host02          192.168.0.2     009027D0A6FB
host03          192.168.0.3     009027D0A7BE
host04          192.168.0.4     009027D0A7B9
#host05         192.168.0.2     00D0B7D5491A
#host06         192.168.0.3     0002B317086B
#host07         192.168.0.4     00D0B7D54878
```

Fig. 2. slave.lst.

1. 各ノードへの OS インストール
   ハードディスクパーティショニング，基本システムのインストール，ネットワークの設定

2. カーネルの再構築
   クラスタとして動作するための設定

3. ソフトウェアインストール，設定
   rsh，MPI などクラスタに必要なソフトウェアのインストール，設定

これらに加えて特別なハードウェアが存在する場合には全てを認識，動作させる作業が必要となる．クラスタの構築はこれら一連の作業を繰り返しとなる．

### 4.2 操作方法

我々が開発しているクラスタインストールツール「DCAST」は，ユーザにかかる負荷を極力軽減したシステムとなっている．DCAST のユーザの操作方法について述べる．

1. DCAST インストール
   DCAST は基本的にスクリプト言語を用いて書かれているため，特別なインストール処理は必要ない．DCAST は同志社大学工学部知識工学科知的システムデザイン研究室クラスタグループソフトウェアページ [9] で公開，配布されているため，適切なディレクトリに展開するだけでよい．

2. ソフトウェアインストール
   DCAST が動作するための最低限必要なソフトウェアをインストールする．必要なものは，BOOTP サーバが動作するための bootp パッケージ，TFTP サーバが動作するための tftp パッケージ，NFS サーバが動作するための NFS パッケージ，一般ユーザにルート実行権限を与える sudo パッケージ，リモートログインを行うための rsh サーバ，rsh クライアントパッケージである．これらのパッケージに関しては，setserver というコマンドで DCAST が動作するためのすべての設定を行われるため，設定は不要である．

3. カーネルの再構築
   子ノードが起動するカーネルを構築する．クラスタとして動作させるための設定と DCAST が動作するための設定 (BOOTP，NFS，NIC のドライバ) を行い，サーバにインストールする．また，子ノード用のカーネルを bzImage として用意する．

4. 設定ファイルの記述
   DCAST の唯一の設定ファイルである slave.lst を記述する．その例を Fig. 2 に示す．このファイルには主にネットワークの設定を記述する．以下でその詳細について述べる．

   - SET
     ディスクレスかディスクフルを指定する．構築するクラスタの形態によって書き換える．ディスクレスクラスタの場合は diskless，ディスクフルクラスタの場合には diskful を指定する．
   - SWAP
     SWAP サイズを決定する．SWAP 以外の残りのハードディスク領域は全て同じパーティション構成となる．ディスクレスクラスタを作成する場合には記述しなくてもよい．
   - NET
     クラスタ内部のネットワーク設定 (ネットワーク，ネットマスク，ブロードキャスト) を記述する．
   - MASTER
     MASTER のホスト名と IP アドレスを記述する．
   - その他
     子ノードのホスト名，IP アドレス，MAC アドレスを全て記述する．

5. DCAST の動作
   残りの操作は DCAST が用意しているコマンドで全ての処理が行われる．ユーザ側の操作はここで終了となる．ディスクレスクラスタを作成する場合には，この処理の終了後，子ノードの電源を入れればクラスタとして動作する．ディスクフルクラスタのセットアップを行うためにはこの処理後子ノードの電源を入れ，起動を確認した後，Master で Enter キーを一度押すことによって，子ノードのハードディスクをフォーマットし，ルートディレクトリを作成する処理が始まる．これらの処理が終了し，自動で再起動が行われればクラスタとして動作する．

### 4.3 システムアップデート

DCAST を利用して，システムのアップデートを行う際には，システム全体を再構築する．すなわち，新たなソフトウェアの導入やセキュリティ関連のアップグレードを行う場合には，一度システム全体をシャットダウンし，子ノードのルートディレクトリをもう一度作成しなおす．再度起動した時にはアップグレードが行われている．またカーネルのアップグレード時には，子ノードの再起動を行い，カーネルをネットワーク上の Master から取得し起動を行う．再度子ノードを起動させた時には，新たなカーネルの読み込みが行われており，システム全体のアップデートが行われる．

すなわち DCAST を利用したクラスタ運用において，システムアップデートを行う際には Master のディレクトリ情報から全てを入れ替える方針となる．Master の情報を基に子ノードのシステムを起動しなおすため，ノードごとの再設定が不必要となる．クラスタ全体のアップデートは Master をアップデートし，子ノードを再起動するだけで行える．

上記のようなアップデートを行うために，DCAST ではすべてのノードをシャットダウン再起動を行うためのコマンドを提供している．さらにすべての子ノードのルートディレクトリを消去するコマンドも提供している．これらの機能を利用してノード全体のアップデートを容易に行うことが可能である．

## 5. DCAST の動作

DCAST は，いくつかの動作を経てクラスタの構築を行っている．本章ではその動作について述べる．

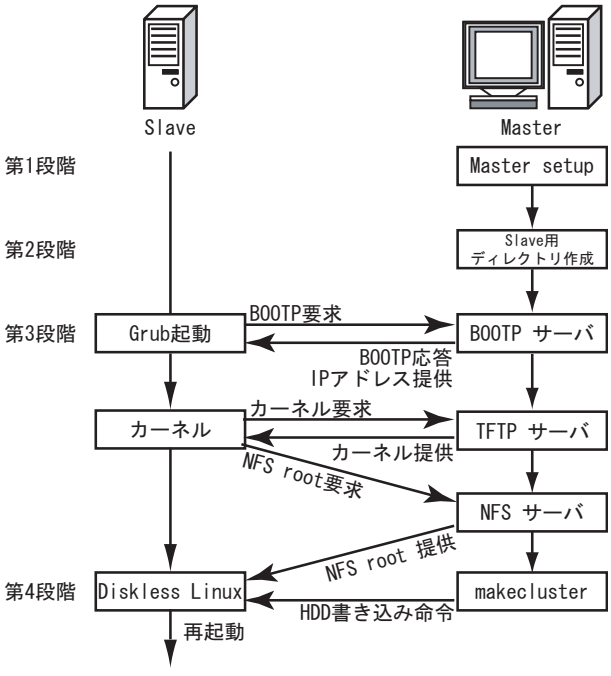


Fig. 3. DCAST の動作.

### 5.1 DCAST が行うインストール処理

DCAST は，現在 Master ノードにおいてすべての処理を行っている．Master ノードでにおける DCAST の処理は，子ノードをディスクレスで起動させるための BOOTP サーバ，カーネルを子ノードに提供する TFTP サーバ，ディスクレスマシンのルートファイルシステムを提供する NFS サーバ，子ノードからのパスワード認証なしでアクセスできる RSH サーバから成り立っている．これらの DCAST 用の設定は，DCAST によって一度だけ実行される．

DCAST の動作は大きく 4 つの段階に分けることができる．DCAST の動作を Fig. 3 に表す．

第 1 段階では Master ノードにおいて適切な設定を行う．これは DCAST を構成するファイルの中で setserver というファイルで提供している．この設定は一度だけ行い，それ以後は実行されることはない．setserver では，slave.lst の記述を基に，以下の項目の設定，動作を行う．

- NFS の設定
  NFS サーバの設定を行う．子ノードのルートファイルシステムとなるディレクトリを特定の IP アドレス，ホストからアクセスできる設定にする．

- inetd の設定
  BOOTP サーバ，TFTP サーバを動作させるため

に inetd.conf を適切な設定に書き換え動作させる．

- ディスクフルクラスタ用の root.tgz 準備
  ディスクフルクラスタ用のルートディレクトリとなる圧縮ファイル root.tgz を用意する．このファイルは消せば自動で新規作成される．消さない限りはずっと同じものを使用する．

第 2 段階では子ノードのルートディレクトリを準備する．このディレクトリを利用して子ノードは起動する．子ノードのディレクトリは DCAST によって各ノード用にディスクレスマシンの設定に書き換えられる．ここで準備する子ノードのルートディレクトリは基本的には Master ノードと同じものを用意する．したがって，認識しているハードウェア等は再設定する必要がない．ここで作成したルートディレクトリを使用して，子ノードはディスクレスクラスタとして起動する．

第 3 段階では，子ノードの電源を入れる．これは手動で入れるか Wake On LAN などを用いて起動させるかなどは自由に行える．この際に，grub を用いて Master ノードに BOOTP 要求を出す．それを受けて Master は子ノードに IP アドレスを提供する．さらに IP アドレスをもらった子ノードは Master ノードにカーネルを要求する．Master ノード上で動作している TFTP サーバはこれを受けて子ノードにカーネルを提供する．カーネルが提供された子ノードは準備されているルートディレクトリに NFS マウントを実行し，ディスクレスクラスタとして動作を行う．ディスクレスクラスタとして動作を行う場合にはこの段階で処理は終了する．

第 4 段階として，ディスクレスマシンとして起動した子ノードのハードディスクをパーティショニングし，ルートファイルシステム情報をコピーする．このコピーは NFS 経由で行われる．最後に */etc/fstab* の書き換えを行ってルートディレクトリのマウント先などを変更する．また grub の menu.lst の書き換えを行ってローカルハードディスクからの起動ができるようにする．

### 5.2 DCAST によって構築されたクラスタの動作

#### 5.2.1 Master ノードの動作

DCAST によって構築された Master ノードは，構築後も基本的にはそれぞれのサーバ (BOOTP サーバ，NFS サーバ，TFTP サーバ) が動作している．これは，ディスクレスクラスタを構築した場合の子ノードの再起動時に Master のサーバを利用して起動するためである．

#### 5.2.2 子ノードの動作

子ノードの動作は，DCAST が提供する grub の設定ファイルにより再起動時の動作が異なる．ディスクレスノードの場合，DCAST が提供するディスクレス用の grub 設定ファイル (menu.diskless) で立ち上がる．ディスクレスノードの場合，構築時と再起動時において大きな動作の違いはない．

ディスクフルノードの場合，DCAST サーバから再インストールを行う動作をする grub 設定ファイル (menu.diskful) とローカルハードディスクから起動する grub 設定ファイル (menu.local) を提供している．menu.diskful を利用して起動する場合には，ローカルハードディスクの初期化，パーティション，ファイルシステム作成を行い，Master から再コピーが始まる．そのため，システムアップデート時に使用する．menu.local で起動した場合には，DCAST の Master に関係なくローカルのハードディスクにある情報で起動する．単に再起動を行う場合にはこちらの設定ファイルで起動する．

DCAST はこれらの起動/再起動に関するコマンドを提供している．たとえば，ディスクフルクラスタとして動作しているクラスタのシステムアップデートを行いたい場合には，以下のようなコマンドを利用して再起動を行い，DCAST による再インストールを行う．

```
# allexchangereboot diskfull
```

このコマンドにより，動作している子ノードの grub 設定ファイルを全て menu.diskful に変更して再起動が行われる．そのためこのコマンドを Master で実行すればあとは待っているだけでシステムアップデートを行うことができる．引数の部分は”diskful”の他に”diskless”，”local”を与えることができる．それぞれ menu.diskless，menu.local で起動するための引数である．

## 6. 既存ソフトウェアとの比較

クラスタのセットアップや管理ツールに関する研究は多数行われているが，その大部分は単一ノード用自動インストーラをクラスタへ適用したものが多い．そのためネットワークからインストールするといったようなクラスタに特化した機能を持ったものは少ない．ここでは，既存インストーラの LUCIE，Oscar，およ

び NPACI Rocks との比較を行う．

### 6.1 LUCIE との比較

LUCIE[8] とは，東京工業大学松岡研究室で開発が行われている，大規模クラスタセットアップツールである．LUCIE でも DCAST と同様に子ノードをディスクレスブートさせ，ハードディスクのパーティショニング，各種ソフトウェアのインストール，設定等を行う．LUCIE と DCAST は類似点が多いが，異なる点について検討する．

#### 6.1.1 NFS ルート

LUCIE では，子ノードをディスクレスブートさせるために使用する Master ノード上の NFS ルートを一つしか生成しない．そのため，複数の NFS ルートが必要なディスクレスクラスタの構築は行えない．これは，LUCIE は基本的にはディスクレスクラスタとして稼働させることを前提としていないからだと考えられる．

一方，DCAST ではディスクレス，ディスクフルともに作成できるようになっている．そのため，子ノードの NFS ルートはそれぞれに作成している．

#### 6.1.2 ファイルコピー

LUCIE では，ファイルコピーという操作は OS のベースとなる最小基本システムを含む圧縮ファイルを子ノードのローカルハードディスクに NFS 経由で取得し展開する．さらに各種ソフトウェアは debootstrap[10] と呼ばれるパッケージを用いて http/ftp サーバ (Debian /GNU Linux のミラーサーバ) から取得し対話的操作無しにインストール/ 設定を行う．LUCIE サーバの NFS ルートを使用してブートした子ノードは，LUCIE 独自の */etc/init.d/rcS* を用いて，インストーラを起動する．LUCIE インストーラは以下の順序で起動する．

1. パーティショニング
2. ローカルハードディスクマウント
3. Linux の最小基本システムを含む圧縮ファイルをローカルハードディスクに展開
4. ユーザが定義したソフトウェアをインストール
5. */etc* 以下の書き換え

DCAST では，Master と全く同じファイルを子ノードのローカルハードディスク上に展開する．これにより新たなハードウェアの認識など Master ノードと同じ設定で動作するものは，迅速に設定が可能となる．展開後，LUCIE と同様に */etc* 以下の書き換えを行う．

```
nisdomain = "blossum"
domainname = "test.doshisha.ac.jp"
ftp_proxy = "http://192.168.0.1:8021/"
http_proxy = "http://192.168.0.1:8080/"
subnet = "255.255.255.0"
host01 {
        luciesrv = "192.168.0.10"
        gateway = "192.168.0.10"
        dns1 = "192.168.0.5"
        kernelimage = "kernel-image-2.2.19_BOOTP1_i386.deb"

        host02 {
                ip = "192.168.0.11"
                mac = "009027D0A6FB"
        }
        host03 {
                ip = "192.168.0.12"
                mac = "009027D0A7BE"
        }
        host04 {
                ip = "192.168.0.13"
                mac = "009027D0A7B9"
        }
}
```

Fig. 4. LUCIE のクラスタ構成定義ファイル．

#### 6.1.3 設定ファイル

LUCIE では少なくとも以下の設定ファイルを用意しなければならない．

- クラスタ構成の定義ファイル (Fig. 4)
- パッケージサーバの設定ファイル
- パーティション構成の定義ファイル

これらはそれぞれ個別のファイルであり，適切に用意しなければならない．またこれだけでは，クラスタとして動作しない．LUCIE は OS を対話的操作なく基本からインストールするインストーラであるため，RSH の設定や MPI の設定，ユーザ管理などを行うファイルを全て準備しなくてはならない．これらの準備したファイルは LUCIE によって適切に置き換え処理が行われインストールが終わった後クラスタとして動作する．作業量は多くなるが，置き換え設定をすることでどんなパッケージでも使用可能となる．

DCAST では Fig. 2 にあるように，設定は唯一の設定ファイルのみである．DCAST では子ノードが外部にネットワークで接続することがないので，プロキシなどの設定はなく，シンプルに構成できる．また基本的な設定は Master と同じになるためファイル置き換えを行う必要はない．また，Master と子ノードで設

定の異なるパッケージ (NIS など) の設定ファイルはスクリプトの処理で書き換えを行っている．これにより対話的操作を減らし，またさまざまなパッケージに対応できるソフトウェアレベルでの拡張性を持たせている．

#### 6.1.4 OS 依存

LUCIE では現在のところ Debian GNU/Linux でのみ動作する．これは，Debian GNU/Linux のパッケージ管理ツール (apt-get) を利用して子ノードのインストールを行うためであり，他のディストリビューションでは違った方法をとらなければならない．例えば RedHat Linux を用いる場合には，RedHat Linux のパッケージ管理ツール rpm 用にカスタマイズをする必要があるため作成者の労力が増大する．

DCAST では，基本的には Master のディレクトリ構成，ファイル構成をそのまま利用するので，特定のディストリビューションに依存していない．*/etc* 以下などクラスタを構成するために必要なファイル書き換えのパスのみを書き換えればどのディストリビューションでも利用可能である．

#### 6.1.5 ノード間コピー機構

LUCIE では，基本システムを含む圧縮ファイルなどの大きなデータのノード間転送に dolly+[11] と呼ばれるディスククローニングソフトウェアを用いている．dolly+は複数台のホスト間で，ネットワークを介してファイルやディスクイメージを複製する．

一般にこのようなクローニングソフトウェアはサーバクライアント方式で実現されているが，対象ホスト数が大規模で，転送するファイルが大きなサイズである場合，サーバボトルネックが発生しコピー速度が著しく低下する．

こうした状況を回避するために，dolly+では，各ノードを Master を先頭としたリング上に接続し，直列転送を行う．このようにして一箇所にボトルネックが発生することを回避し，全二重ネットワークスイッチの性能を最大限に引き出している．

一方，DCAST ではこうしたコピー機構に対しては未対応であり，台数の増加にしたがってコピー時間は線形に増加する．LUCIE よりも大きなファイル転送を行うため何らかのソフトウェアを組み込んでファイル転送の高速化を行う必要がある．

### 6.2 Oscar

Oscar[4] は Open Cluster Group によって開発が進められているクラスタ構築，管理ツールである．Open Cluster Group ではグリッド [12] のエンドポイントとしてのクラスタに取り組んでおり，クラスタソフトウェアの標準化に取り組んでいる．Oscar を用いることによって，Open Cluster Group 標準のクラスタシステムを構築することが可能となる．

Oscar のインストーラは RedHat Linux にインストールを行う．LUCIE や DCAST と同様に Master に Oscar をインストールする．Master でインストールする際に，いくつかの設定を行い，子ノードの MAC アドレスを自動収集する．さらに Network Boot を利用して子ノードを起動し，クラスタノードを構築する．Network Boot にはブートフロッピーを用いるか PXE boot を利用する．構築終了後に再起動を行う場合に対話操作が必要となる．PXE boot の場合は全て BIOS 起動を行って再設定しなくてはならないため非常に大きな労力が必要となる．また Oscar では想定された構成のクラスタを構築することはできるが，柔軟な設定に対応していない．

DCAST では，起動時に grub を用いて起動する．この際，ディスクレスブートとディスクフルによるブートの設定ファイルの交換を行うため，再起動時の対話操作が省略されている．

### 6.3 NPACI Rocks

NPACI Rocks[3] は LUCIE や DCAST と同様に，再インストールによってクラスタノード上のソフトウェアアップグレードや復旧を行う．基本的には XML ファイルにクラスタの論理構造を記述し，RedHat KickStart の CD-ROM を作成することによってクラスタのインストールを行う．

NPACI Rocks を用いたクラスタ構築の手順は次のようになる．まず Master ノードで CD-ROM を入れて起動を行い，DHCP と SQL サーバを動かす．次に CD-ROM を子ノードに入れ換える．この作業を繰り返すことによってクラスタセットアップを行う．

CD-ROM の抜き差しといった作業を台数分繰り返さなくてはならないため，手間が多く大規模クラスタ向きではない．

## 7. まとめ

本論文では，PC クラスタ用セットアップ，管理ツールとして DCAST の提案を行った．その動作および特徴について，既存のクラスタセットアップソフトウェアとの比較を行った．

DCAST では，grub と呼ばれるブートローダを使

用し，システムの動作から再起動をしてクラスタとして動作するまでを完全に自動構築が可能である．またシンプルな構成によって動作するソフトウェアであるため特定のディストリビューションに依存することが少なく，他のどのセットアップツールよりも柔軟性がある．

また，管理に関して，現在はノードの稼動状態を目で確認し，正常に動作していないノードを DCAST によって再インストールするようにしているが，そのようなノードを自動的に検出し，復旧させる機構についての開発を進めて行く予定である．

## 参 考 文 献


1) T. Sterling, D. Savarese, D. J. Beeker, J. E. Dorband, U. A. Renawake, and C. V. Packer. BEOWULF: A parallel workstation for scientific computation. *In Proceedings of the 24th International Conference on Parallel Processing*, pp. 11–14, 1995.

2) T. L. Sterling, J. Salmon, D. J. Beeker, Savarese, and D. F. Savarese. How to Build a Beowulf: A Guide to the Implementation and Application of PC Clusters. *MIT Press*, 1999.

3) Philip M.Papadopoulos, Mason J.Katz, and Greg Bruno. NPACI Rocks: Tools and Techniques for Easily Deploying Manageable Linux Clusters. *IEEE Cluster 2001*, pp. 258–267, 2001.

4) Open Cluster Group. OSCAR: A packaged cluster software stack for high performance computing. *http://oscar.sourceforge.net/*.

5) PC Cluster Consortium. *http://pdswww.rwcp.or.jp/*.

6) SCYLD COMPUTING CORPORATION. *http://www.scyld.com*.

7) SCALI: Scalable Linux Systems. *http://www.scali.com/*.

8) 高宮安仁, 真鍋篤, 白砂哲, 松岡聡. Lucie: 大規模クラスタに適した高速セットアップ 管理ツール. SWoPP 湯布院 2002, pp. 131–136, 2002.

9) 同志社大学工学部知的システムデザイン研究室クラスタグループ. *http://mikilab.doshisha.ac.jp/dia/research/cluster/*.

10) Debian GNU/Linux debootstrap. *http://packages.debian.org/stable/admin/debootstrap.html*.

11) Dolly+ home page. *http://corvus.kek.jp/~manabe/pcf/dolly/*.

12) Carl Kesselman Ian Foster. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Pub, 1998.

**出典：**
同志社大学 理工学研究報告 Vol.43, No.4 January 2003
pp. 7-16

**問い合わせ先：**
同志社大学工学部/ 同志社大学大学院工学研究科
知的システムデザイン研究室
(http://mikilab.doshisha.ac.jp)