

Windows クラスタの利用を支援する計算環境構築の取り組み

辻 田 祐 一^{†1} 大 西 悠 平^{†1,*1} 丸 山 拓 哉^{†1,*2}

近年, Windows Compute Cluster Server 2003 (以下, Windows CCS) を用いた PC クラスタ利用が盛んになってきている。Windows に慣れた利用者にとって, 同じ GUI インタフェースでクラスタの管理・運用が行えるため大変有用である。しかしながら, 様々な機能が利用できる一方で, Windows CCS を知らない利用者にとって, 必要とする機能を習得する負担も発生する。そこで, 数値計算を行う利用者, 特に MPI による並列処理を行う利用者を対象に, 計算処理を支援する環境を提供することが有用であると考えた。我々は, 利用する際の複雑さを隠蔽することにより, Windows CCS クラスタに対する操作に利便性を提供することを検討してきた。検討の結果, 我々は, 計算処理に関わる操作を視覚的にサポートするコンパクトで利用し易い計算支援環境を提案し, システムの構築を進めている。本計算環境が提供する GUI ベースのユーザ・インタフェースにより, ジョブ実行に関わる諸パラメタの設定並びにジョブ実行支援・ジョブ監視などが容易に行えることを確認した。

Building a Compact Computing Environment for a Windows PC Cluster

YUICHI TSUJITA,^{†1} YUHEI ONISHI^{†1,*1} and TAKUYA MARUYAMA^{†1,*2}

Recently, Windows Compute Cluster Server 2003 (hereafter Windows CCS) has been focused for its fruitful functionalities in scientific computation. It is very useful for users who are familiar with a Windows system because they can use the same GUI which is available in a Windows PC during management and operations. However, it is time consuming to learn functionalities of the Windows CCS for non-expert users. We have focused into providing a compact and user friendly computing environment for parallel computing using MPI. We have considered to provide usability in manipulations of the Windows CCS cluster by hiding complexity in its control operations. After the consideration, we have built a compact and user-friendly GUI-based computing environment for computation on the Windows CCS. Its GUI interface gives hints for job submission, job monitoring, and so forth associated with computation. With the help of this system, users can easily carry out their applications on the Windows CCS.

1. はじめに

近年, Windows CCS¹⁾ を用いた PC クラスタが注目され, 盛んに利用されている。MPICH2²⁾ をベースに開発された MPI ライブラリ (MS-MPI¹⁾ などによる並列処理もサポートされており, 慣れ親しんでいる Windows 環境と同じ GUI でクラスタの管理・運用が出来ることなども, 有用な点として挙げられる。Windows CCS では, PC クラスタの管理・スケジューリングを行うジョブスケジューラにより, ジョブ実行支援が行われている。このジョブスケジューラは GUI ベースのインタフェースを持ち, ユーザへの利便性を高めているが, 多くの機能を提供していることもあり, それらの機能の中から利用者が必要とする機能を選び,

利用できるまでには, ある程度の労力及び時間を要する。我々は, 様々な利用形態がある中で, MPI による並列処理に関して, 利用者のジョブ実行を支援するコンパクトな計算支援環境の構築を試みた。

この計算支援環境により, GUI ベースのユーザ・インタフェースを用いてジョブ及びタスクの作成・実行に加え, ジョブの実行状態の監視等が容易に行える。以下, 2 章で Windows CCS について概説した後, 3 章で我々が行った実装の動機・実装メカニズム・動作例について述べる。次に 4 章で関連研究について述べた後, 最後に 5 章で本論文のまとめを行う。

2. Windows CCS の概要

PC クラスタの OS として, これまでは UNIX, 特に Linux が殆どを占めていたが, 近年 Windows OS を用いた PC クラスタが注目されている。特に Windows CCS が提供されたことにより, Windows 環境に慣れ親しんだ利用者が, 容易に PC クラスタ環境に触れる

^{†1} 近畿大学 工学部 電子情報工学科
Department of Electronic Engineering and Computer
Science, School of Engineering, Kinki University

*1 現在, 日立ビジネスソリューション

*2 現在, 富士通ビジネスシステムズ

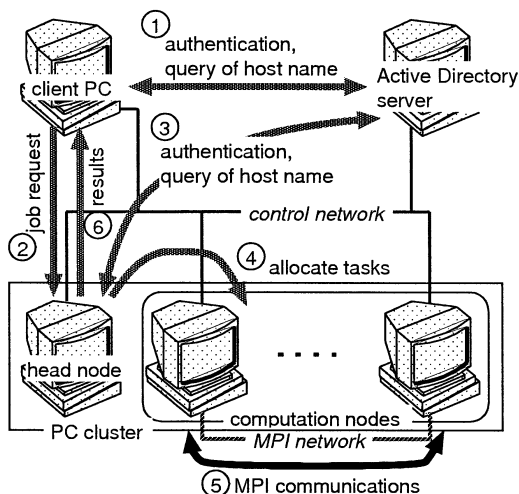


図 1 Windows CCS による PC クラスタ環境構築例

ことが出来るようになった。Windows CCS は、OS となる Windows Server 2003 Compute Cluster Edition と、クラスタ利用の為にツールキットを提供している Compute Cluster Pack から成る。前者は、サーバ OS として利用されている Windows Server 2003 に対し、クラスタ運用に必要な機能を中心とした構成の OS で、後者は、クラスタ運用の為にジョブスケジューラや、並列処理に用いる MPI ライブラリである MS-MPI ライブラリなどが用意されているパッケージ群となっている。クラスタ環境の構築例を図 1 に示す。PC クラスタに加えて、利用者の認証や、ホスト名の問い合わせなどのために Active Directory サーバを構築する必要がある。このためにサーバを立ち上げるか、あるいはクラスタのノードの一つに兼用させることで対応できる。クラスタは 1 台のヘッドノードと複数の計算ノードで構成され、ヘッドノードは、クラスタの管理・運用に関わる処理を、計算ノードは与えられた計算処理を行う。ヘッドノードに計算ノードの機能を兼用させることも可能である。

3. 計算支援環境

3.1 開発の動機

Windows CCS は、利用者が必要とする様々な機能を提供しているが、Windows CCS のツールの中から、必要な機能を選び出し、使い慣れるまでにある程度の時間を要することなど、利用者にある程度の負担が生じる可能性がある。利用される分野により、使用される機能は異なると思われるが、我々は、主に MPI による並列計算を行う利用者向けに Windows CCS の利用を支援する環境構築が必要と考えた。また、我々の知る限り、Windows CCS におけるジョブスケジューラ

機能のジョブ実行支援メニュー画面である Compute Cluster Job Manager において、標準出力および標準エラー出力をその都度メニュー画面で参照する機能は無く、出力結果を確認したい場合、利用者が別途エディタ等により繰り返し参照する必要がある。このような利用方法では、随時出力される結果を定期的に確認するには、その都度エディタによるファイル参照を行わねばならず、不便である。また、クラスタで利用可能な CPU 数や稼働中の CPU 数などは、ジョブスケジューラ機能の管理ツールである Compute Cluster Administrator により参照可能であるが、このツールを利用するには、管理者権限が必要であり、一般利用者では、情報を得ることが出来ない。

しかしながら、標準出力及び標準エラー出力を画面上で定期的に参照出来る機能を備えることが、ジョブが正常に実行されているかどうかを確認する上で、大変有用と考えている。また、クラスタの状態を適宜知ることが出来るようにすることもジョブ実行の際に必要なであると考えている。

以上の検討を踏まえ、我々は Windows CCS のための計算支援環境の実装を試験的に行った。以下、提供する機能とその実装方法について説明する。

3.2 提供機能及び実装方法

本実装を実施するにあたり、大きく分けて以下の機能を提供することを検討した。

- ジョブ及びタスク作成支援機能
- ジョブ投入・制御機能
- クラスタ・ジョブ・タスク監視機能
- 標準出力・標準エラー出力参照機能
- 共通パラメタ設定機能

Compute Cluster Administrator の GUI メニュー画面では、管理者でないクラスタの状態を参照できないなど、いくつか制限があったが、別途用意されているコマンドラインインタフェース (CLI) では、稼働中の CPU や空き CPU の数、ジョブコンテナ並びにタスクの作成・実行・監視機能などが一般利用者でも利用可能である。ここに着目し、以上の機能を提供する為に、表 1 に示すような Java クラスを作成し、図 2 に示すような動作をする支援環境を構築した。本支援環境は、クライアント側 (利用者端末) で利用者により Java アプリケーション (ParJobMenu クラス) として起動される。一方、サーバ側 (Windows CCS クラスタのヘッドノード) では、サーバ機能を有する Java アプリケーション (CCSManageServer クラス) があらかじめ起動されており、両者間で Java RMI によるデータ通信を行う。

クライアント側で起動された Java アプリケーションは、メニュー画面を担当する JobMenuFrame クラスのインスタンスを生成することで、支援環境メニューを画面上に起動する。ここで各種の操作に対応するクラス (図 2 のクライアント側の lower layer にあるク

表 1 作成した Java のクラスと提供機能

	Java クラス名	提供機能
サーバ側 (ヘッドノードで動作)	CCSManageServer	Windows CCS への要求処理サーバアプリケーション
	CCSManageImpl	Windows CCS へのインタフェース
クライアント側 (クライアント PC で動作)	ParJobMenu	クライアント側 Java アプリケーション上位レイヤ
	JobMenuFrame	支援環境画面生成
	SubmitJob	ジョブ及びタスク作成・制御
	MatchJob	ジョブ監視 (別スレッドで起動)
	WatchProcess	クラスタ監視 (別スレッドで起動)
	OpenFile	リモートファイル操作
	PrefFile	初期設定ファイル編集機能

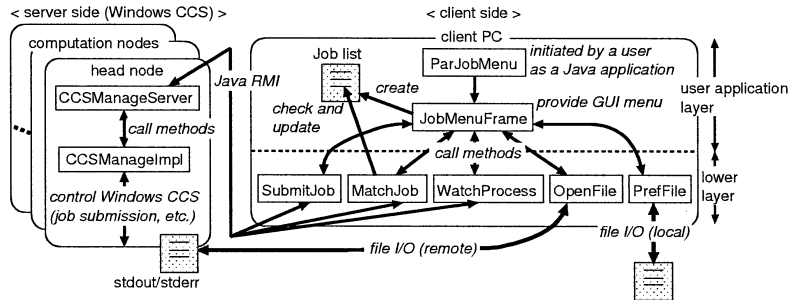


図 2 作成した Java のクラスとその動作

ラス) のメソッドを呼び出すことにより、当該操作が行われる。例えば、SubmitJob クラスはジョブ及びタスクの作成・制御を、MatchJob クラスは実行ジョブの監視を担当している。これらの操作を担当するクラスは、Java RMI を用い、Windows CCS クラスターのヘッドノード上のサーバアプリケーションに操作要求を送信する。要求を受けたサーバアプリケーションは、CCSManageImpl クラスにある Windows CCS への操作を行うメソッドを呼び出す。このメソッド内部では、CLI のコマンドを使用することにより、Windows CCS への操作が行われる。

なお、本実装において、Java は Java SE Development Kit 6³⁾ を利用した。

3.2.1 ジョブ及びタスク作成支援機能

Windows CCS では、ジョブ実行のために、ジョブコンテナを作成する必要がある。そのため、ジョブ名や利用可能最大 CPU 数を指定した後に当該コンテナを作成できる機能を提供する。次に、並列処理を行うために、プログラム実行を担当するタスクを作成し、ジョブコンテナに登録する機能を提供する。タスク作成には、実行プログラム名、CPU 数、作業ディレクトリ、標準入出力、標準エラー出力などのパラメタ設定を行う。

ParJobMenu クラスを Java アプリケーションとして起動すると、JobMenuFrame クラスにより、図 3 に示すような画面が現れる。更に、作成したジョブコンテナをジョブスケジューラに投入した際のジョブ及び

タスク管理用リスト (以下、ジョブリスト: 図 2 のクライアント側の Job list に相当) が、JobMenuFrame クラスにより作成される。

この画面には複数のタブメニューが用意されており、一番左側にジョブ及びタスク作成用画面 (タブメニュー "submit") がある。まずこの図の (1) のジョブコンテナ作成の欄において、ジョブ名や使用する最大 CPU 数を設定し、"create Job container" ボタンを押すと、ジョブコンテナが生成され、画面中央より下に、作成したコンテナに対応するアイコンがツリー表示される。次に、実行する計算処理に関するパラメタを図 3 の (2) タスク作成の欄で設定し、"add" ボタンを押すことにより、ジョブコンテナに作成したタスクが登録され、この図の真中にあるように、ツリー表示されたジョブコンテナのアイコン (この例では "MXD, 950" と表示) の下に登録したタスクのアイコン (この例では "MXD-1,950.1..." と表示) が表示される。

3.2.2 ジョブ投入・制御機能

ジョブ及びタスク作成支援機能により作成されたジョブコンテナとその中に登録されたタスク群は、図 3 の (3) ジョブ投入の欄において、クラスタを利用する利用者 ID とパスワードを入力し、"submit" ボタンを押すと、SubmitJob クラスにより、ヘッドノードのサーバアプリケーションを介して、Windows CCS が稼働する PC クラスターのヘッドノードのジョブスケジューラにジョブが引き渡される。ジョブを受け取ったジョブスケジューラは、利用可能資源が取得でき次第、ジョ



図 3 ジョブ作成機能画面の例

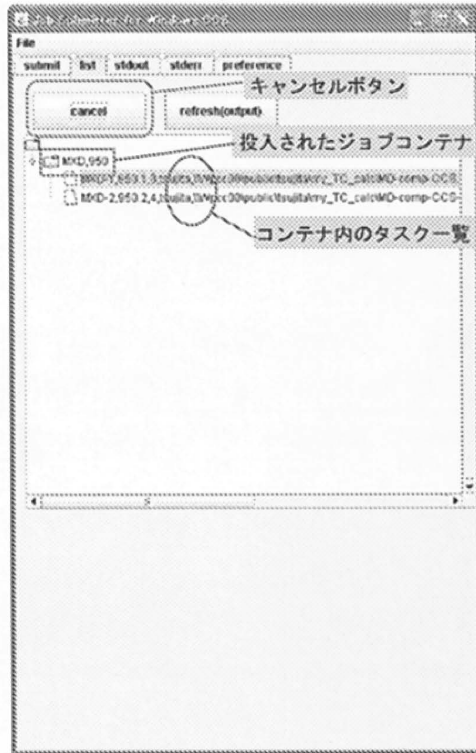


図 4 投入ジョブ一覧表示画面の例

ブの実行が開始される。又、ジョブの引き渡し完了後、上位レイヤの JobMenuFrame クラスによって、ジョブリストに当該ジョブの情報が登録される。

3.2.3 クラスタ・ジョブ・タスク監視機能

クラスタや投入したジョブ及びタスクの状態を知る為に、定期的に図 3 の (4) 状態監視の欄に、ジョブの状態・総 CPU 数・利用可能な CPU 数が表示される。

PC クラスタの総 CPU 数及び利用可能な CPU 数は、計算支援環境の起動時に別スレッドとして起動されている WatchProcess クラスにより、定期的に PC クラスタ側に情報を参照し、上記の表示欄に表示を行う。

一方、図 3 の (3) ジョブ投入欄で投入されたジョブコンテナ及びタスク群は、3.2.2 で述べた通り、ジョブリストに登録されており、これらの情報が図 4 に示すジョブ一覧表示画面にツリー表示される。ツリー表示のリストから目的のタスクを選択すると、計算支援環境の開始時に既に別スレッドとして起動されているジョブ監視スレッド (MatchJob クラス) により、当該タスクに対する監視要求がサーバ側に送られる。サーバ側は要求に応じ、状態情報を取得した後、クライアント側に返送する。状態情報を受け取ったクライアント側は、ジョブの状態 (例えば、実行中は "Running"

と表示) を図 3 の (4) 状態監視の欄にある "status =" の欄に表示する。

また、図 4 のツリー表示から目的のタスクを選択し、"Cancel" ボタンを押すことにより、当該タスクの実行をキャンセルすることが可能になっている。この時、クライアント側の SubmitJob クラスのメソッドにより、サーバ側にキャンセル要求が送られて、サーバ側でタスクのキャンセルコマンドが発行される。これにより、Windows CCS のジョブスケジューラに投入されたタスクがキャンセルされ、その時の結果 (成功又は失敗) がクライアント側に返される。失敗の場合、ポップアップ表示により、失敗したことを利用者に知らせる。

3.2.4 標準出力・標準エラー出力参照機能

標準出力並びに標準エラー出力は、ジョブの実行状況を判断する際に、良く利用される。このような利用形態も想定し、両出力を実行中に確認できる機能も提供している。それぞれ、ジョブリストから目的のタスクを選択した後、"stdout" 並びに "stderr" タブメニューを選択することで、参照可能である。それぞれの出力の参照は、OpenFile クラスによって、出力先のファイルに対し読み出しを行い、計算支援環境の画面上に表示させている。標準出力を表示している例を図 5 に

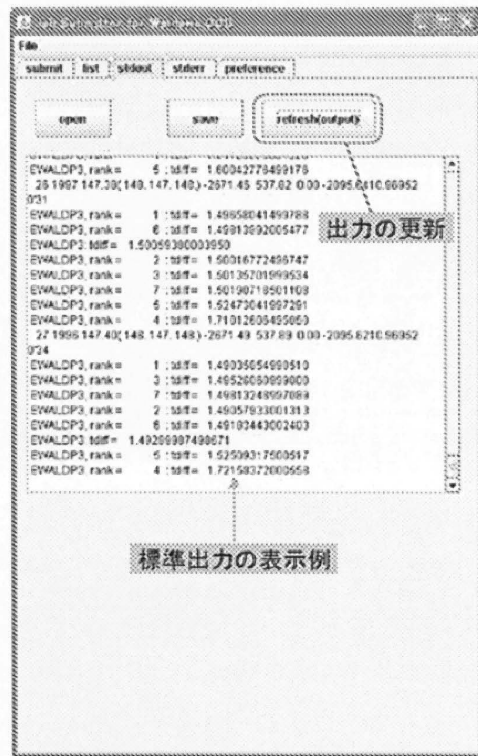


図 5 標準出力表示の例 (アプリケーション例として MD プログラムの一つである MXDORTOP⁴⁾ に手を加えたものを利用)

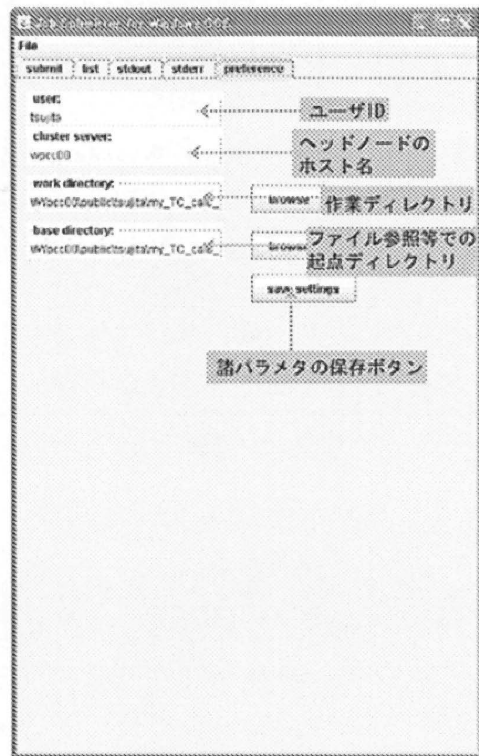


図 6 設定情報保存画面

示す。この画面の上部にある“refresh(output)”ボタンを押すことにより、現在の出力結果を確認することが可能である。さらに、“save”ボタンにより、その段階での出力結果を自計算機上にファイルとして保存することも出来る。

3.2.5 共通パラメタ設定機能

本計算支援環境を繰り返し利用する際に、毎回ユーザー ID や PC クラスタのヘッドノード名など、同じものを使い続ける可能性があるものについては、あらかじめ設定ファイルに記述しておき、本計算支援環境の起動時にそれらを参照し、設定するようになっている。さらに、図 6 に示すように、これらの設定値を変更させる支援画面 (タブメニュー “preference”) を用意している。入力したパラメタ群は、画面下にある “save settings” ボタンにより、設定ファイルに保存ができる。

4. 関連研究

PC クラスタ等を用いた計算におけるクラスタ資源管理及びジョブスケジューリングを行う実装が多数開発されている。その一つとして、Condor⁵⁾ が挙げられ

る。Condor は、分散された計算資源を集めた計算環境を効率良く利用する環境構築を支援している。ジョブスケジューラである Condor Central Manager は、遊休状態の計算資源にジョブを割り当てることにより、資源の有効利用を実現している。Sun Grid Engine⁶⁾ も同様な機能を提供するものとして広く利用されている。あらかじめ定義した資源利用ポリシーに基づき、クラスタ環境の管理・運用を行うことができる。

PC クラスタの OS としては、UNIX、特に Linux が多数を占めてきたが、Windows CCS を用いた PC クラスタも近年注目を集めており、MPI による並列計算だけに限らず、Windows 上で動作するアプリケーションのバックエンド計算機としても利用されている。グリッド環境でも、計算資源として利用する試みがなされており、例えば、英国の National Grid Service⁷⁾ において、英国のサウサンプトン大学では、Windows CCS 向けに改造した Globus⁸⁾ ベースのグリッド通信基盤を用い、計算資源の一つとして Windows CCS クラスタを提供する試みを行っている⁹⁾。

一方、我々が構築した計算支援環境は、LAN 等の小規模な環境で、手軽に Windows CCS を利用することを目的としている。利用に際しては、Windows CCS と Java が利用可能であれば、すぐに利用できる。

5. ま と め

我々は、Windows CCS クラスタにおける計算処理を支援する環境の構築を行い、GUI メニューによるジョブ作成及び実行の支援機能とクラスタ並びにジョブの監視機能を実現した。この中で我々は、Windows CCS において並列処理を行う利用者に対し、(1) 必要なパラメタ（プログラム名、作業ディレクトリ、標準入出力、標準エラー出力等）の設定支援、(2) ジョブコンテナの作成・投入などの操作支援、(3) クラスタや投入ジョブの監視情報の提供、(4) 標準出力・標準エラー出力の参照機能を提供する計算支援環境構築を行った。

本実装では Java を用い、Windows CCS のヘッドノードをサーバ側、利用者端末をクライアント側とし、サーバ・クライアント型の環境を構築した。両者間のデータ通信は Java RMI により実現している。Windows CCS へのジョブ作成やジョブ投入・監視などは、一般ユーザでも CLI によるコマンドラインでの実行が可能で、サーバ側にあらかじめサーバアプリケーションを起動しておき、クライアント側から送られてくる要求に応じて、サーバアプリケーションが必要なコマンドを生成し、Windows CCS 側への操作を行うように設計した。これにより、クライアント側のユーザレベルでは、Windows CCS に依存しないインタフェースを提供している。ジョブコンテナやタスクの作成やジョブ実行の状態参照において、それらをアイコンとツリー表示により画面表示を行い、利用者の利便性を提供した。また、標準出力と標準エラー出力を容易に参照できるようにしたことで、利用者に対し、計算プログラムの状況を容易に把握できる機能を提供した。

今後、クライアントとして Linux などへの対応を検討するつもりである。また、計算アプリケーションの利用者との意見交換により、ユーザ・インタフェースや Java クラスの機能・構成の見直しを行う予定である。

謝辞

本計算支援環境構築にあたり、計算アプリケーションとして、東京工業大学 理学部 地球惑星科学科 河村雄行 教授が作成された MXDORTOP を利用させて頂いた。ここに深く感謝致します。

参 考 文 献

- 1) Microsoft Compute Cluster Server 2003:
<http://www.microsoft.com/windowsserver2003/ccs/>.
- 2) MPICH2: <http://www.mcs.anl.gov/research/projects/mpich2/>.
- 3) Java Technology: <http://java.sun.com/>.
- 4) Kawamura Laboratory, Tokyo Institute of

Technology: <http://www.geo.titech.ac.jp/kawamuralab/kawamuralab.e.html>.

- 5) Condor Project Homepage:
<http://www.cs.wisc.edu/condor/>.
- 6) Sun Grid Engine: <http://www.sun.com/software/gridware/index.xml>.
- 7) NGS: <http://www.grid-support.ac.uk/>.
- 8) Foster, I., Kesselman, C. and Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations, *The International Journal of High Performance Computing Applications*, Vol.15, No.3, pp.200-222 (2001).
- 9) Microsoft High Performance Computing Institute, School of Engineering Sciences, University of Southampton: <http://www.soton.ac.uk/ses/research/mshpci/index.html>.