平成 22 年度 RICC 利用報告書

課題名（タイトル）：

大規模クラスタにおけるソフトウエア DSM とその応用

利用者氏名：　緑川 博子
所属　　：　本所　情報基盤センター

1. はじめに

64bitOSの普及により，X86_64の現実装でも256TBという飛躍的に大きなアドレス空間が利用可能となり，大容量データを扱う様々な応用が容易に実行できるようになってきた．従来のOS における仮想メモリ機構では，物理メモリサイズを超えるデータを扱う場合には，ローカルハードディスク上のスワップ領域にページ単位でスワップすることで，物理メモリサイズを超えた仮想メモリを実現している．しかし近年，ローカルハードディスクのデータ転送速度を越える通信性能を持つネットワークが出現し，ローカルマシンと遠隔マシン間でメモリページをスワップし，仮想メモリとして利用する遠隔メモリページングの研究がなされるようになってきた．筆者は，本研究課題のソフトウエア分散共有メモリの応用として，クラスタにおける遠隔マシン上のメモリを利用し，ローカル物理メモリサイズに制限されずに大容量メモリを提供する分散大容量メモリシステムDLM (Distributed Large Memory)の構築，評価を行ってきた[1][2][3]．DLMは大容量データを扱う逐次処理のためのシステムで，並列化の困難なアプリケーションや，並列・分散処理の専門知識を持たないユーザにとって恩恵がある．

通常，物理，化学，宇宙といった様々なサイエンス関連の科学技術シミュレーションでは，まず逐次プログラムを作成し計算のモデル化を行う．このモデルを小規模な問題に適応して有効性を検証した後に，問題の大規模化を行うことが多い．しかし，大規模問題に拡張する際に，ローカルメモリサイズを超える大きなデータを扱うことになると，コンピュータクラスタの複数ノードへデータを分割･分散させ、逐次プログラムで書かれた計算モデルをMPIなどの並列プログラムへ再設計しなおす必要が生じる．しかしすべてのプログラムが並列プログラムにできるとは限らず、また可能であってもそれに伴うプログラム，データの再構築，複雑な並列デバッグなどは，各科学者の本来研究以外に多くの時間と苦労を強いることになる．また他人の設計した逐次コードやライブラリを用いている場合にも並列化することが難しい．

このような時，DLMを用いると，逐次コードのままで，ローカルメモリに入りきらないデータは，ユーザには暗黙のうちに，クラスタの複数ノードに自動的に分散し，ローカルメモリを超える大容量データを扱う逐次プログラムをそのまま実行させることができる．遠隔メモリを利用するため，ローカルメモリのみを使用した従来の逐次実行に比べれば実行時間は長くなるものの，多くのプログラムにはメモリアクセス局所性があり，プログラムの使用する全データの5%をローカルメモリにおき，95%を遠隔メモリで代替えした場合でも（すなわちローカルメモリの19倍程度のメモリが利用できる），ローカルメモリ100%利用時（従来実行）の５倍程度の速度で実行できることも多い[2][3]．並列プログラムへの再設計･デバッグ･並列動作による正当性検証にかかるコストをかけずに，汎用のクラスタをメモリ資源としてすぐに逐次プログラムによる問題大規模化を試すことが可能となる．

DLMは，OSスワップシステムに組み込む他の多くのカーネルレベル実装の遠隔メモリページング手法とは異なり，OSとは独立のユーザレベルソフトウェアとして実装したことで，高速かつ安定した動作が得られることを示してきた[1]．またユーザレベルソフトウエアとして稼動するため，OSやシステムを選ばず高い可搬性と可用性が得られる．

本課題研究では，当面，このDLMシステムの使いやすさと高性能化を進めることを目的とし，次に，このマルチスレッド型遠隔メモリスワップ機構の技術を用いて，ポストMPIとなるような高性能処理用の並列プログラミング環境，言語の構築の基盤となるマルチコア対応の高性能分散共有メモリシステムの構築を目指している．

２．シングルクライアント用システム DLM-S

クラスタノード間の通信として，従来の汎用ソケット通信（TCPなど）を用いたシステム[1][2]に加え，Myrinet，Infiniband，Ethernetといった様々な高速通信媒体で利用可能であるMPIを通信に用いたDLMシステムを構築し[3]，汎用のオープンクラスタなどで一般ユーザが手軽に利用できるシステムを構築した．図１は，計算ノードの計算スレッドでユーザプログラムを実行し，通信スレッドで遠隔メモリサーバノードと通信を行うDLMを示す．このDLMで逐次プログラムを実行できるCプログラム例を図２に示す．図中，dlmと書かれたデータは，ローカルメモリが足りない場合にはクラスタの遠隔メモリサーバのメモリに領域がとられ，必要に応じて計算ノードとの間で，ページ単位でスワップされる．図２のプログラムは，従来の逐次コードのままであることに注目してほしい．これを可能にするDLM関数ライブラリとDLMコンパイラを構築している[9][10]．

図１は計算ノードで実行されるユーザプログラムを唯一のクライアントとする．メモリを提供するメモリサーバプロセスがユーザプログラム開始時に自動的に立ち上がり，ユーザプログラム終了とともに消滅するシングルクライアント用のDLMシステムでDLM-Sと呼ぶ．

３．マルチクライアント用システム DLM-LAN

DLM-S では各ユーザプログラムの起動時にあらかじめユーザが指定したメモリサーバノードにそのクライアント専用のメモリサーバプロセスが起動され，ユーザプログラム終了時に生成したメモリサーバプロセスも消滅するシステムである．一方，常駐型のメモリサーバプロセスを管理者があらかじめ複数立ち上げておき，複数のユーザクライアントが必要に応じて任意のメモリサーバに接続してメモリ提供を受けるマルチクライアント向けシステムが DLM-M である[5]．DLM-M のメモリサーバの内部構成は DLM-S のようなシングルプロセスではなく，クライアントからのメモリ利用要求を受け付ける度に，メモリサーバメインプロセスが，各クライアント専用の処理プロセスを子プロセスとして動的に立ち上げるマルチプロセスサーバとして設計されている．

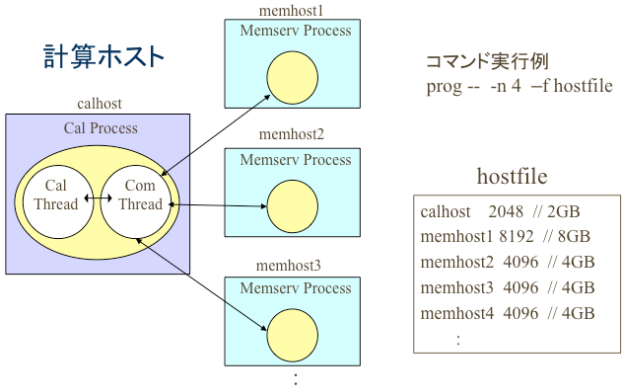

図１　シングルクライアント向けシステム DLM-S

DLMプログラム

```
#include <stdlib.h>
#include <dlmm.h>
#define NUM 10
#define LENGTH (10*(1L<<30))  //10G
dlm int a[NUM][LENGTH]; // 400GB

int median(int num) {
   dlm  int b[LENGTH]; // 40GB
   unsigned int j;
   for ( j = 0; j < LENGTH; j++) b[j] = a[num][j];
   qsort(b, LENGTH, sizeof(int), compare_int);
   return b[LENGTH/2];
}

int main ( int argc, char *argv[])
{
   unsigned int i, j;
   for ( i = 0; i < NUM; i++)
      for ( j = 0; j < LENGTH; j++) a[i][j] = rand();
   for ( i = 0; i < NUM; i++){
      printf("median[%d] = %d\n", i, median(i));
   }
   return 0;
}
```

図２　DLMプログラム例

さらにこの DLM-M を拡張し，管理プロセスを導入した DLM-LAN(図 3)は，計算ノードのユーザプログラム起動時に，クラスタ内で稼働中の複数メモリサーバプロセス群から，メモリ使用量，サービス中のク

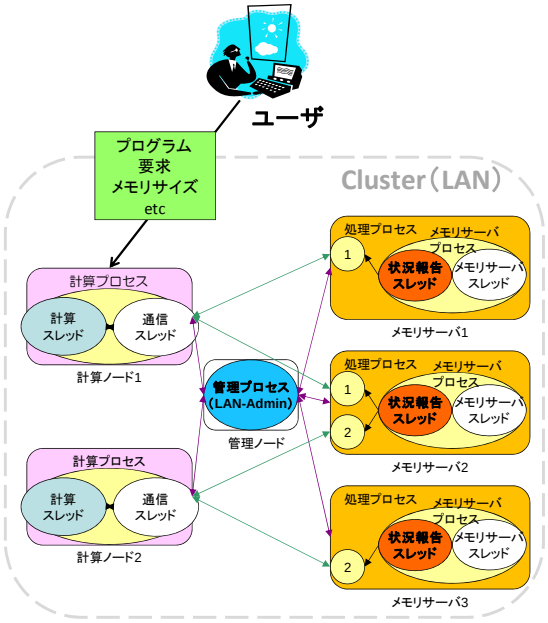

図３　マルチクライアント向けシステムDLM-LAN

ライアント数などを考慮して，適当なメモリサーバを自動選択して実行させる管理プロセス LAN-ADMIN を導入している[6]．各メモリサーバプロセスはこの管理プロセス LAN-ADMIN から起動され，その後も定期的に計算負荷，クライアント数，利用可能メモリ量などの状況を管理プロセスに通知する．これらのステータス情報により，複数のユーザが同時に複数のメモリサーバの利用する場合にも，各ユーザが，他ユーザの状況やメモリサーバの状況を気にすることなく，管理プロセスは負荷を分散させて，全体として効率よく実行できるようにメモリサーバの管理と自動割り付けを行う．

4．WAN接続クラスタ群利用システム　DLM-WAN

今年度は，従来の１クラスタ向けの DLM-LAN システムを拡張し，図４に示すよう WAN 接続された複数のクラスタ群の中から自動で利用クラスタを選択し，その中からメモリサーバノードと計算ノードの自動選択を行うようにした DLM-WAN システムを設計，構築した[7][8][12]．DLM-WAN では，１クラスタ内の各メモリサーバの状態を管理する前述の管理プロセス LAN-ADMIN の上位に，WAN 全体のクラスタ群の状況を管理するプロセス WAN-ADMIN を導入している．

WAN-ADMIN は，ユーザプログラム開始時にユーザの要求メモリ量を指定されると，各クラスタの LAN-ADMIN に問い合わせて，ユーザの要求を満たしできるだけ効率よく実行できるような適切なクラスタと計算ノードを選択する．選択基準はいくつかのモードがあるが，クラスタ全体の利用可能メモリ（メモリサーバと計算ノードのメモリ量の総和），そのクラスタで利用可能な計算ノードのローカルメモリ量と計算負荷などを考慮して，ローカルメモリができるだけ大きくとれ，計算負荷が少ない計算ノードを選ぶようになっている．またメモリ量の比較には，即時利用可能メモリ量(Free)を優先とし，それで優劣がつけられない場合には，未解放メモリ量(Innactive)も加えた量で比較する．適切と思われる計算ノードとクラスタを選択した後，選択したクラスタの管理プロセス LAN-ADMIN に選定クラスタ内でのメモリサーバ割当を委託する．遠隔メモリアクセスは同一クラスタ内（LAN 通信）で行えるように，計算ノードとメモリサーバノードの割り付けは１クラスタ内のノードに置くものの，複数のユーザのメモリ要求を WAN 全体に分散させて，複数のクラスタ群全体をメモリ資源として利用できるようになっている．

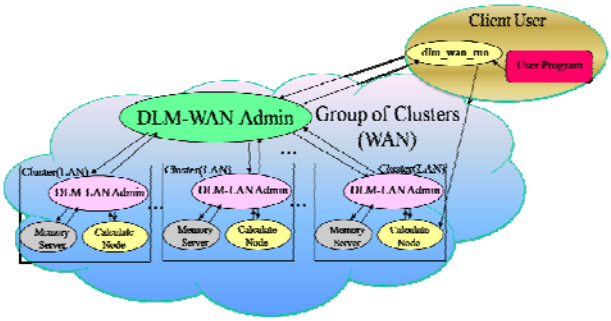

図４　WAN接続クラスタ群におけるDLM-WAN

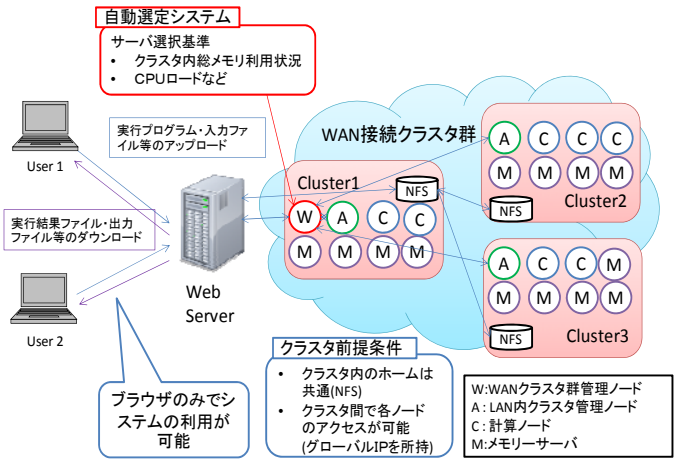

図５　ポータルサイト経由によるDLM-WAN 利用

またユーザプログラムのジョブ投入をどこからでも可能にするため，図５に示すようにポータルサイトとしてwebサーバを立ち上げ，図６のようなwebインターフェースによるアクセスを可能とした．

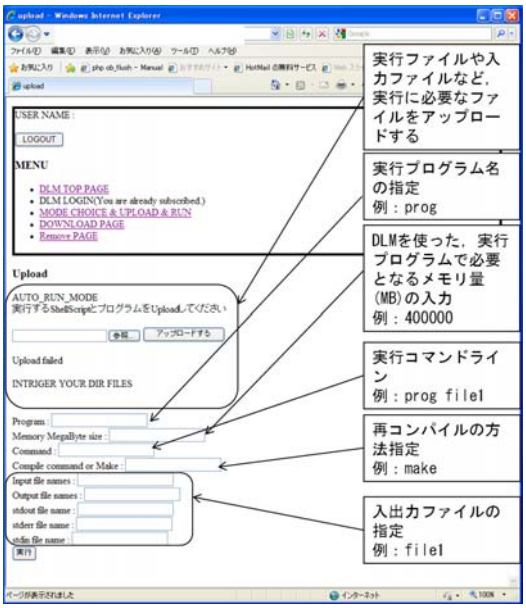

図６　DLM-WAN利用のためのwebインターフェース

５．DLM-WAN　稼働実験

５．１　実験環境

DLM-WANの稼働実験は,インタラクティブなジョブ投入とメモリサーバプロセスと管理プロセスなどの常駐プロセスを起動させておく必要性から，全国21大学のクラスタをWANで接続した分散コンピューティングシステムであるIntrigger[11]を利用した.

現在のDLM-WANの稼働条件は以下の３つである.(1) クラスタ群全体でユーザアカウントは同一,(2)各クラスタでのユーザホームディレクトリは共通,(3)クラスタ内ノードはグローバルIPで遠隔アクセス可能

Intriggerシステムのうち，内部LANに10Gbps Ethernetを持つ３クラスタ（東大，法政，北大）を利用した.各クラスタのノードは,DLM利用可能ローカルメモリサイズを12GBと設定し，メモリサーバノードをそれぞれ９個，７個，５個用意し，計算ノードは各クラスタそれぞれ４個とした.

ここでは１つの計算ノードでは実行できない15GBのメモリを必要とするジョブ（姫野ベンチマークELARGE）を８個連続投入した.

５．２　実験結果

２つの自動割り付けモードによる結果を図７，図８に示す．図中には計算ノードしか示していないが，○印のジョブに付けられた番号が割り付けの順番になっている.

- ローカルメモリサイズ優先モード

  図７に実行結果の例を示す．実験当時，北大のクラスタはDLM以外の他のユーザの使用により空きメモリサイズが小さくなっていたため，東大，法政クラスタに優先的にジョブが割当られ，北大にはジョブが割り当てられなかった.

  このモードは，プログラム実行をする場合，できるだけローカルメモリが豊富な計算ノードを割り当てることにより各ユーザプログラムの実行時間を低減しようとするモードである．この例のように，北大で提供できるメモリに制限があり，他のユーザが利用している環境では，自動的にそのような負荷のかかったクラスタを避けて，メモリが豊富で空いているクラスタに優先的に割り付ける.

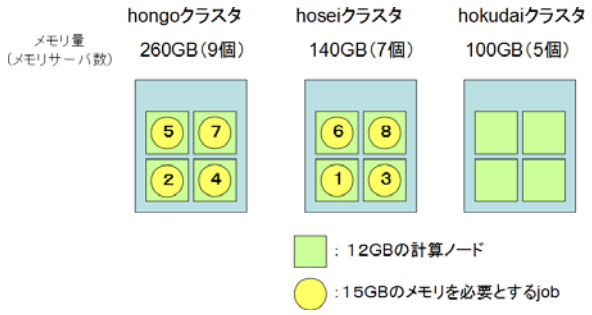

図７　ローカルメモリサイズ優先モードの割当

- クラスタ内クライアント数均一化モード

  割り付けモードを変更し，同一環境で同じジョブを投入した結果が図８である.このモードでは,ローカルメモリサイズを優先しつつも，各クラスタでのクライアント数の均等化を図るモードで割り付けされる．従って，北大はローカルメモリが少ないために,割り付けの優先順位は低いものの,各クラスタになるべく均一にジョブが割り当てられていることがわかる.

  このモードは，１クラスタのクライアント数に大きなばらつきがでないように，クライアントを割り当てる．メモリが多いからといって，１クラスタに多くのクライアントが割り付けられると,クラスタ内のLAN通信が混み合うことになり,性能の低下を引き起こす可能性がある．このため，各クラスタの利用状況がほぼ同等の負荷，メモリ使用量である場合に，負荷を分散させて割り付けられる効果があると考えられる.

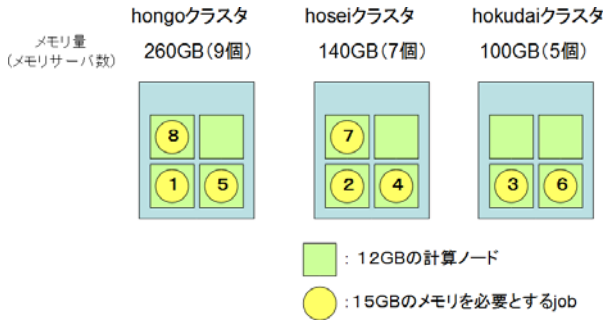

図８　クライアント数均一化モードの割当

６．　今後の計画・展望

ここで紹介したDLM-WANについては,さらに様々な種類のプログラム投入の実験などを行い，サーバ自動割り付け方針などについても評価を行いたいと考えている．この他の進行中の研究としては,現在，遠隔メモリページングの通信プロトコルの

高速化，ページプリフェッチを組み込んだプロトコルの研究を行っている．また，マルチスレッドユーザプログラムの実行を可能にするマルチスレッド対応 DLM システムの試作も行っている．今後，マルチコアに対応した効率のよい通信プロトコルと，マルチスレッドユーザプログラムの利用を進めて行く予定である．

７． 今年度 RICC 利用の成果がなかった理由

今年度は遠隔メモリの利用環境を拡張する DLM-WAN などの研究が主体となり，インタラクティブな環境での実験が必須で，バッチシステムで運用される RICC クラスタを利用する実験を行うことができなかった．

８． RICC の継続利用について

来年度の利用についても現時点では見合わせ，研究の必要性が生じたときに新たに申請を行いたいと考えている．

参考文献

［1］緑川博子,黒川原佳, 姫野龍太郎： "遠隔メモリを利用する分散大容量メモリシステム DLM の設計と 10GbEthernet における初期性能評価", 情報処理学会論文誌コンピューティングシステム，Vol.1, No.3, pp.136-157（2008,12）

［2］ H. Midorikawa, M.Kurokawa, R.Himeno, M.Sato: "DLM: A Distributed Large Memory System using Remote Memory Swapping over Cluster Nodes", Proc. of Cluster Computing, pp.268-273, (2008.9)

[3]緑川博子、齋藤和広、佐藤三久、朴 泰祐："クラスタをメモリ資源として利用するための MPI による高速大容量メモリ"、情報処理学会論文誌，コンピューティングシステム，Vol.2, No.4, pp.15-36, (2009.12)

［4］H. Midorikawa, K.Saito, M.Sato, T.Boku: "Using a Cluster as a Memory Resource: A Fast and Large Virtual Memory on MPI", Proc. of IEEE cluster2009, pp.1-10, (2009.9)

［5］齋藤和広, 緑川博子, 甲斐宗徳："マルチクライアント向け分散型大容量メモリシステム DLM-M の設計と実装",情報科学技術フォーラム FIT2008, FIT 論文集,C－003, pp.199-200, （2008,9）

[6] 三浦望，緑川博子，甲斐宗徳:" クラスタをメモリ資源として利用するための動的メモリ提供システムの提案",情報科学技術フォーラム FIT2009, FIT 論文集,B-029, pp.421-422, （2009,9）

［7］鈴木悠一郎、緑川博子："分散大容量メモリ DLM の WAN 接続クラスタ群への適用 −クラスタ･サーバ自動選定システムの提案 −"、 SACSIS2010, pp.173-174, (2010.5)

[8] 鈴木悠一郎, 緑川博子："WAN 接続クラスタをメモリ資源として利用するためのメモリサーバ自動選定システム ーウェブインターフェースによるユーザビリティの向上ー", ハイパフォーマンスコンピューティングと計算科学シンポジウム HPCS2011, HPCS2011 論文集, p.85, （2011,1）

[9] 吉村 礎, 緑川博子, 甲斐宗徳："ローカルメモリを越える大容量データを扱う逐次処理のための C コンパイラ", 情報科学技術フォーラム FIT2010, FIT 論文集, B-026 , pp.335 -336, （2010,9）

［10］吉村 礎, 緑川博子："遠隔メモリ利用で大容量データ処理を可能にする逐次プログラムための C コンパイラ", ハイパフォーマンスコンピューティングと計算科学シンポジウム HPCS2011, HPCS2011 論文集, p.84, （2011,1）

[11] 田浦: "InTrigger ： オープンな情報処理・システム研究プラットフォーム", 情報処理 Vol.49 No.8,pp939-944,(2009.8)

[12] 鈴木,緑川,市野: ”WAN 接続クラスタ群をメモリ資源として利用するためのメモリサーバ自動選定システムの開発”,情報処理全国大会(2011,03 予定)