C-003

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

Design and Implementation of Distributed Large Memory System for Multi-client : DLM-M

斉藤和広† 緑川博子† 甲斐宗徳†

Kazuhiro Saito Hiroko Midorikawa Munenori Kai

1. はじめに

筆者らはローカルの物理メモリサイズに制限されることなく，クラスタの各ノードの遠隔メモリを利用し仮想的に大容量のメモリ空間を提供するシステム，DLM(Distributed Large Memory)を構築，評価してきた[1][2].

今回この従来の DLM(以後 DLM-S)とは別にマルチクライアント向けのデーモン型メモリサーバとライブラリ群を新しく構築し，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 (DLM for Multi-client)を実装した．さらに従来の DLM-S にはメモリの解放とそれを行うためのメモリ管理機構が存在しなかったため，DLM-S のメモリ資源の再利用性がなかった．DLM-M ではユーザプログラム実行中の解放と，それに伴う動的メモリ割り当てを効率的に行う大容量メモリの管理機構を設計，実装した．本論文では，主にこの新システム DLM-M の実装とマルチクライアント化における性能評価結果を報告する．

2. DLM-M システムの構成

従来の DLM-S の特徴は

- DLM コンパイラによりソースプログラムを変更しユーザには透過的にメモリサーバを自動起動
- メモリサーバを起動した単一のクライアント専用

という特徴がある．これに対して DLM-M は

- ユーザ又は管理者がメモリサーバを明示的に起動
- 常駐のマルチクライアント向けのメモリサーバ

という特徴があり，図１のような構成が可能となった．

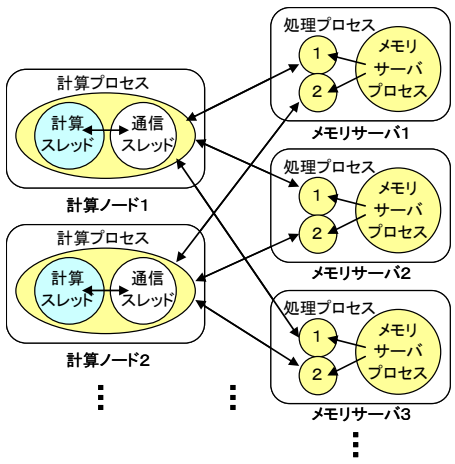

図１　DLM-M システムの構成

2.1 DLM-M システム

図１のように，ユーザプログラムの計算をするノードを計算ノードと呼び，遠隔メモリを提供するノードをメモリサーバと呼ぶ．通信は TCP/IP を用い，基本的に DLM ページ単位でデータの送受信を行う．メモリの管理も同様に DLM ページ単位で行い，DLM-M ページ管理表を用いてメモリ割り当て情報を操作する．このページ管理表は，計算ノードは全ノードのメモリを，メモリサーバは自ノードのみのメモリを管理している．

計算ノードには，ユーザプログラムコードを実行する計算スレッドと，メモリサーバと通信を行う通信スレッドがある．計算スレッドがユーザプログラム中で計算ノードが保持していないデータにアクセスすると SIGSEGV ハンドラ内においてメモリサーバにアクセスページの要求を行いページが受信されるまで待つ．その後メモリサーバからのページを通信スレッドが受信すると計算スレッドにシグナルを用いて知らせ，計算スレッドは計算を再開する．その後通信スレッドは代わりとなるスワップページをメモリサーバに送信する．

メモリサーバには，親プロセスで計算ノード（クライアント）からの接続要求を受け付けるメモリサーバプロセスと，各クライアントが遠隔メモリを利用のための処理を行う処理プロセスからなる．メモリサーバプロセスはそのノードでの全 DLM-M 利用メモリ量を管理している．クライアントからのサーバ接続要求を受けると，クライアントが要求する遠隔メモリ量を受信し，残りの総メモリ量のチェック・更新と初期設定を行う．次に処理プロセスを生成しクライアントの処理を任せ，メモリサーバプロセスは再度新しい接続待ち状態となる．処理プロセスは担当クライアントのメモリ管理表，隙間リスト生成等の固有の初期設定を行い，DLM のサービスを開始する．

2.2 ユーザインタフェース

DLM-M のライブラリ関数は C 言語で利用することが可能である．図２のプログラム例のように記述することで，DLM-M のメモリサーバが提供する遠隔メモリを利用できる．プログラムを実行する手順例を図３に示す．この例では，メモリサーバをサーバマシンで起動し，その後ユーザプログラムを実行する．DLM-M の実行コマンドのオプションは，-n が利用ノード数（計算ノードとメモリサ

```
#include <dlmm.h>
#define N 1<<30 /* 1G (*4B=4GB) */
int main(int argc,char* argv[]){
  int *a;
  dlmm_startup(argc,argv);  /* DLM-Mの初期化 */
  a=(int*)dlmm_alloc(sizeof(int)*N); /* 動的割り当て */
  ・・・  /* 計算 */
  dlmm_free(a);   /* 解放 */
  dlmm_shutdown();  /* DLM-Mの終了 */
  return 0;
}
```

図２　DLM-M の利用例

†成蹊大学工学研究科情報処理専攻, Graduate School of Engineering, Seikei University

ーバ数の和）の指定，-f が設定ファイルの指定である．図４は各ユーザプログラムを実行するために必要な DLM-M の設定ファイル例(dlmm.conf)で，利用するホスト名(又は IP アドレス)と利用可能メモリサイズ(MB)を記述している．

```
サーバでのコマンド> dlmm_server
クライアントでの実行 >
        User_program -- -n 4 -f dlmm.conf
```

図３　DLM-M の実行例

```
calhost     2048   // 2GB  計算ノード
memhost1    2048   // 2GB  メモリサーバ1
memhost2    4096   // 4GB  メモリサーバ2
   :
```

図４　ユーザプログラム実行用設定ファイル(dlmm.conf)

3. 性能評価

3.1 実験環境

今回の性能評価実験に用いた実験環境は，1GbitEthernet で繋がれたサーバマシン１台とクラスタ(７ノード)である．サーバマシンにメモリサーバデーモンを常駐させ，クラスタマシンがクライアントとなり性能評価を行った．それぞれのマシンの性能を表１で示す．

表１　サーバマシンとクラスタマシン

ホスト	サーバマシン	クラスタマシン
マシン	HP ML150 G3	HP ML150 G2 x 7 Nodes
CPU	Xeon E5310 1.6GHz QuadCore x 2CPU	Xeon 2.8GHz x 2CPU HyperThread
メモリ	8GB	1GB
L2キャッシュ	4MB/CPU	1MB/CPU
OS	Linux kernel2.6.23.17-88.fc7 x86-64	Linux kernel2.6.20-1.2320.fc5 x86-64
コンパイラ	gcc version 4.1.1 20070105	gcc version 4.1.1 20070105
NIC	NC7781 OnBoard Gigabit NIC	Broadcom 5721 PCI-Express Gigabit NIC
ネットワーク	1GbitEthernet	

3.2 STREAM ベンチマーク

定常的なメモリアクセス帯域を調べるためのベンチマークである STREAM[3]を用いて，DLM-M のマルチクライアント利用時における遠隔メモリアクセスの帯域を測定した．STREAM は配列アクセス操作を配列全体に複数回繰り返し，この中での最良値（初回を除く）を出力する．また計測にはキャッシュの影響を小さくするために十分に大きな 配列を用いる必要がある．このため事前にローカルメモリアクセス帯域を計測し,キャッシュの影響を受けない 10M 個(240MB)の配列を測定に用いることにした．

次に DLM ページサイズの影響を調べるために上記の配列サイズでサーバマシン１台とクライアントノード１台で DLM ページサイズを変えて測定を行った．測定時にはメモリアクセスが初回を除きすべて遠隔メモリアクセスになるように DLM-M の設定ファイルを調整した．その結果，図５のように，メモリアクセス帯域は最大で 35MB/s，10GbitEthernet 利用時のほぼ 10 分の 1 で[2]，DLM ページサイズが 128KB 以上ではほとんど変化がなかった．

次に，同じサイズの配列に対し，DLM ページサイズを 128KB とし，１台の DLM-M サーバに対して１～７台のクライアントノードから接続した時の STREAM のメモリアクセス帯域をそれぞれ測定した．その結果，図６のように，クライアント数に比例した速度低下を示した．常時遠隔スワップを起こす STREAM では，サーバプロセスの処理ネックというより，ネットワーク帯域の飽和による通信ネックが起きていると考えられ，DLM-M 使用上でも最悪に近い利用例に該当する．一般の応用プログラムでは一定レベルのメモリアクセスローカリティが存在し，STREAM に比べスワップ頻度は低く，より多くのクライアントをサポートできると考えられる．

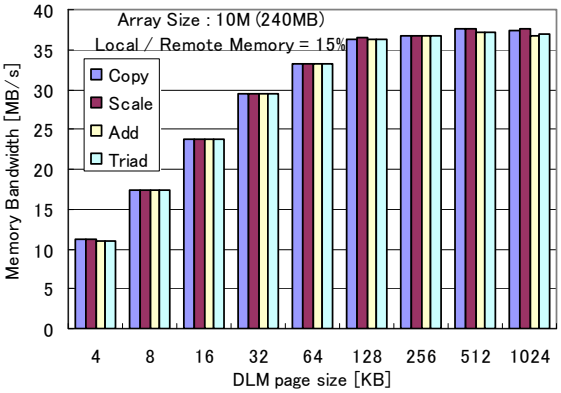

図５　DLM-M のページサイズ毎の STREAM 測定結果

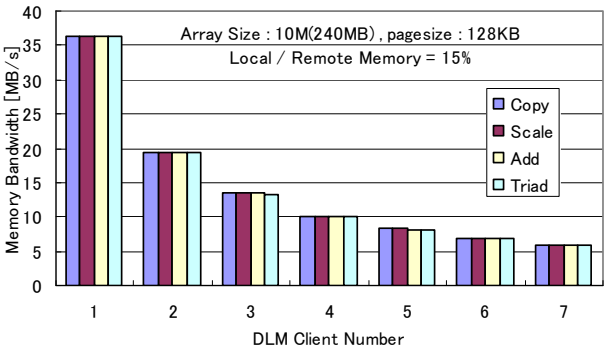

図６　マルチクライアントにおける STREAM 測定結果

4. おわりに

今回，新しくマルチクライアント環境での運用を可能にする DLM-M を設計，実装し，初期性能評価を行った．今後，メモリアクセスローカリティが存在する実際の応用プログラムでの性能，複数メモリサーバにおける遠隔スワップの分散時の性能についても評価していく予定である．また，今回作成したメモリ割当と解放の性能評価，及び dlmm_free の DLM-S への移植も行う予定である．

参考文献

[1]緑川, 小山, 黒川, 姫野, “分散型大容量メモリシステム DLM の設計と DLM コンパイラの構築”, 電子情報通信学会研究報告 CPSY 研究会報告, 信学技法 Vol.102 P.29-34, No.398, Dec.2007
[2]緑川, 黒川, 姫野, “遠隔メモリスワップのユーザレベルソフトウェア DLM と性能評価”,電子情報通信学会研究報告 CPSY 研究会報告, Aug.2008
[3]STREAM Benchmark, http://www.cs.virginia.edu/stream/ref.html