

# PC クラスタにおけるユーザレベルソフトウェア分散共有メモリSMS

緑川 博子 伊藤 慎\* 大橋 祐介 飯塚 肇 成蹊大学 工学部(\*現在, SUN(株))

User-Level Software Distributed Shared Memory SMS on PC cluster

Midorikawa H., Ito S., Ohashi Y., M., Iizuka H. Seikei University

## 概要

PCクラスタ上にユーザレベルソフトウェアによる分散共有メモリシステムの実装した。新しいメモリコンシステンシモデル(IBECS)を採用し、ページベースの管理を行う。その設計、実装結果について報告する。

### 【システムの特徴】

- ・ユーザレベルソフトウェアによる実装

- ・緩和型メモリコンシステンシ IBECS の採用

Implicit Binding Entry Consistencyという新しいメモリコンシステンシを提案し採用した。lock/unlock、barrierを用い、共有データと同期変数との関連付けはユーザによって暗黙に行う。

### 【システムの構成要素】

- ・ページマネージャ

共有データに関わるページ管理プロセス

```
#include<stdio.h>
#include<sms.h> /* sms 用ヘッダーファイル */
#define NUM 1024
void main(int argc, char *argv[ ])
{
    int *data, *max, n, i;
    FILE *frp;
    frp=fopen(argv[1],"r"); /*データファイル*/
    sms_startup(argc, argv); /* sms 起動関数 */
    n=NUM/sms_nproc;
    /* 共有データ領域確保 */
    data=(int *)sms_alloc((sizeof(int))*NUM,0);
    max =(int *)sms_alloc(sizeof(int),0);
    /*マスタープロセスによるデータ初期化 */
    if(sms_proc_id==0){
        for(i=0;i<NUM;i++) fscanf(frp,"%d",&data[i]);
        *max=0;
    }
    /* 全プロセスによる最大値探索 */
    sms_barrier(0); /* バリア同期 */
    for(i=sms_proc_id*n;i<(sms_proc_id+1)*n;i++)
    { sms_lock(0); /* ロック獲得関数 */
      if(*max < data[i]) *max = data[i];
      sms_release(0); /* ロック解放関数 */
    }
    sms_barrier(0); /* バリア同期関数 */
    /*マスタープロセスによる結果出力 */
    if(sms_proc_id==0) printf("max =%d\n",*max);
    sms_shutdown(); /* sms 終了関数 */
}
```

図1 簡単なプログラム例(最大値を求める)

- ・バリアマネージャ/ロックマネージャ

バリア同期、ロック同期を管理するプロセス

- ・メモリアクセスハンドラ

共有メモリアクセス時に呼び出され、ページ要求や、diff生成のための準備を行うハンドラ。

- ・メッセージI/Oハンドラ

ページ要求、diff転送など、プロセッサ間通信時に呼び出される通信ハンドラ。TCPを採用

- ・分散共有メモリライブラリ(libsms)

図1のプログラム例に使用されている6種のCの関数、2種の定数をAPIを提供している。

### 【評価】

本実験は表1の環境で評価を行った。図2に基本部分のオーバヘッド時間を示す。図3に多体問題プログラムの速度向上比を示す。

PC	CPU:Celeron 400MHz	メモリ:128MB
PC	CPU:MMXPentium 166MHz	メモリ:64MB
OS:	RedHat Linux 6.0 / FreeBSD2.2.2R	
ネットワーク	100Mbpsイーサネット	

表1 システム環境

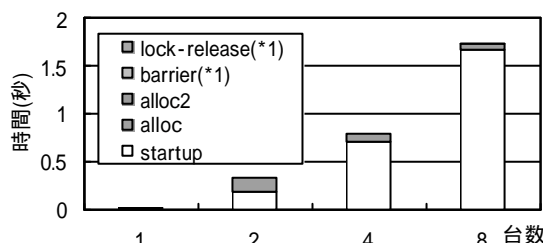


図2 システム開始時のオーバヘッド

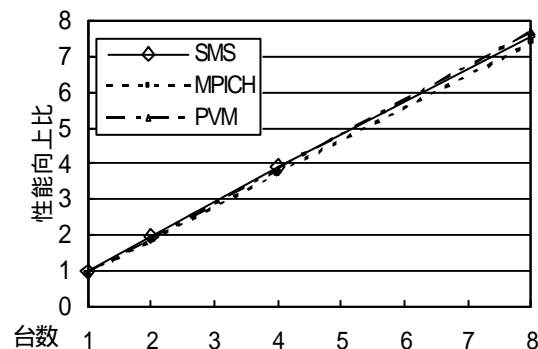


図3 多体問題(サイズ10,000)プログラムの性能向上比