マルチスレッド対応型分散共有メモリシステムの設計と実装

緑川 博子　岩井田匡俊　(成蹊大)

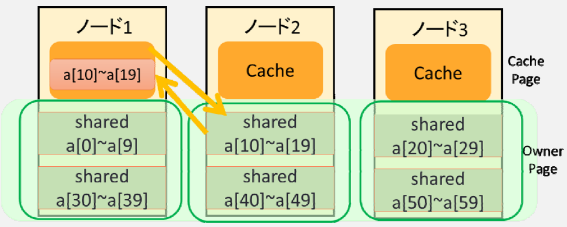
マルチスレッド・マルチノードプログラミングのための分散共有メモリシステム (Multi-SMS)

OpenMP/pthreadマルチスレッドプログラムと同様な，クラスタにおけるPGAS・共有メモリプログラミング環境を提供する基盤システム
従来のMPI＋openMPハイブリッドプログラミング対応する新しいページベースの分散共有メモリシステム

- ユーザプログラム内の動的なスレッド生成にも対応 （pthread_create( )関数のフックによる自動スレッドID管理）
- ユーザプログラム中の複数スレッドによるページアクセス競合を考慮
- 遠隔ノードからのページフェッチ，ノード間通信の効率化を図るマルチスレッドによるシステム管理機構

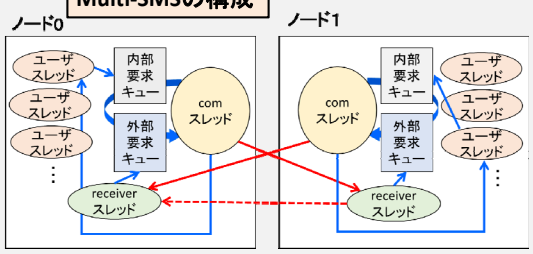
ページベース分散共有メモリ

共有データの仮想共有メモリシステムへの分散配置



ノード間で共有するデータは各ノードに分散して管理
Owner: 各ノードが管理する共有データ
Cache: 他ノードのOwnerページのコピー領域

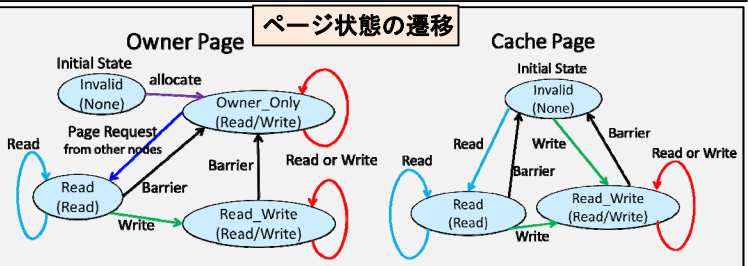
Multi-SMSの構成



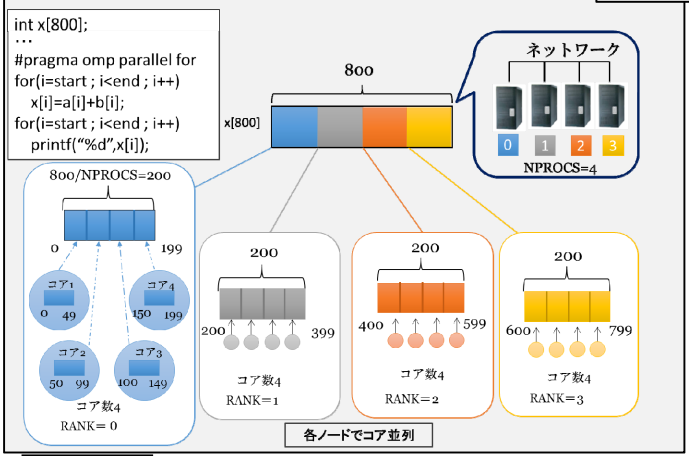
通信スレッド：
ページ管理，ノード内外の非同期・並列要求の処理，外部への要求送信

受信スレッド：
外部ノードからの非同期要求を処理

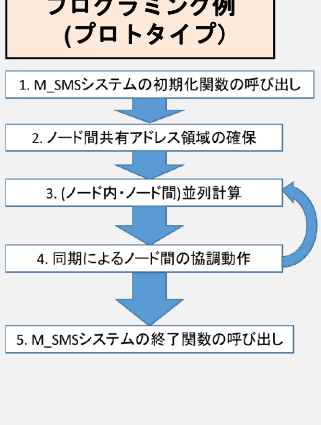
ページ状態の遷移



マルチノード・マルチスレッド 共有メモリプログラミングのイメージ



プログラミング例 (プロトタイプ)




```
int* x;
…
sms_startup();

x=sms_alloc(800,sizeof(int),0);
…

#pragma omp parallel for
for(i=sms_rank*(800/sms_nprocs);
   i<(sms_rank+1)*(800/sms_nprocs); i++)
   x[i]=a[i]＋b[i];
 }
sms_barrier();
if(sms_rank==0)
 for(i=0;i<800;i++)
  printf("x[i]=%d¥n",x[i]);
…

sms_shutdown();
```

稼働実験

2ノードシステム(crestサーバ)

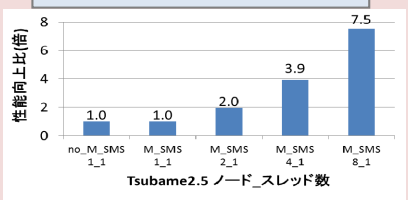
CPU	Intel Xeon E5-2687W(3.1GHz) 2×8core/node = 16core/node
Memory	128GB/node (2 nodes)
Network	Infiniband singleFDR(56Gbps)
OS	CentOS6.3
Compiler	gcc 4.4.7 20120313
MPI Lib	MVAPICH2-2.1a

8ノードシステム(Tsubame2.5)

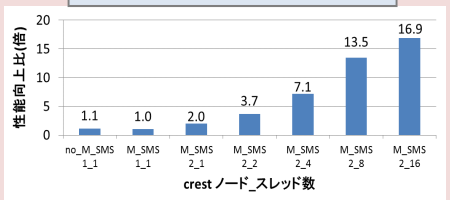
CPU	Intel Westmere-EP 5670(2.93GHz) 2×6core/node = 12core/node
Memory	58GB/node (1367 nodes) 103GB/node (41 nodes)
Network	Infiniband dual QDR(40Gbps x2)
OS	SUSE Linux Enterprise Server 11 SP3
Compiler	gcc 4.3.4
MPI Lib	MVAPICH2-2.1a

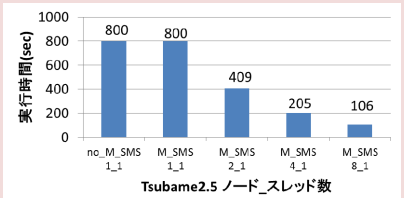
ステンシル計算　8192x8192（1GiB)問題(15x15点計算)

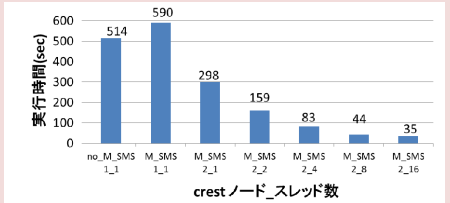
8ノードシステム(Tsubame2.5)



2ノードシステム(crestサーバ)







今後の展開

- 共有データの分散マッピング：分散メモリマップ関数sms_mapallocに組み込み
- プログラムインターフェース、API： 既存開発のSMSシステム、MpC， DLM言語のAPIを踏襲
- 共有メモリー貫性とは別経路による遠隔メモリ・ローカルメモリ間直接コピー操作Get/Put の実装
- メモリー貫性実装方式の検討（現在はall-invalidate方式）
- アクセス局所性を考慮した応用アルゴリズムによる性能評価

分散データマッピングAPI に組み込み予定
shared int x[800]::[sms_nprocs];
shared double data[M][N]::[sms_nprocs][ ];
shared float a[L][K]::[2][2];