HA8000クラスタシステムでMPI性能を測定する場合の注意点

システム運用係、(株)日立製作所

HA8000クラスタシステムの利用者からの問い合わせから、ノード内でのMPI通信の性能測定において、注意を促したほうがいいと思われる案件がありましたので、実際に性能測定した結果とあわせて報告します。

１．ノード内MPI通信性能を測定した場合の性能で誤解が生じる可能性がある

HA8000クラスタシステムで、ノード内に複数のプロセスを立ち上げて、それらプロセス間でMPI通信を行うジョブでの利用があると思います。その利用に先立ち、ノード内MPI通信の基本性能を測定しようと、同じ通信を繰り返してその実行時間を測り、性能を算出する手段が考えられますが、測定プログラムの作り方や実行の仕方によって、実際のアプリケーションでは出せない性能が出る場合があります。

MPI通信の性能測定プログラムとして、Intel MPI Benchmarkがありますが、Intel MPI BenchmarkのSendRecvは、各転送データーサイズにおいて繰り返しSend、およびRecvを行い性能測定しますが、その送受信の繰り返しでは、同一の送信バッファ、同一の受信バッファを使用するという特徴があります。

一方、HA8000クラスタシステムでは、ノード内MPI通信はノード内の共有メモリーを用いた通信（SHMEM通信）を標準値としています。SHMEM通信は、CPUを利用してソフトウェア処理でプロセス間のデーターコピーを行うため、小さなデーターサイズの場合、同一の送信バッファ、同一の受信バッファを繰り返し使用する通信性能測定ではキャッシュアクセスとなり、実際のアプリケーションでは出すことができない性能が出る場合があります。正しく性能を測定する場合は、キャッシュの影響を受けないように、送信バッファと受信バッファを繰り返して使わないようにする（例えば、ずらしながら使用する）必要があります。

HA8000クラスタシステムの計算ノードに搭載しているCPUであるAMD Opteronプロセッサ8356（開発コード名 Barcelona）のキャッシュ構成を表1に示します。

表1. Opteronプロセッサのキャッシュ仕様

項目	仕様
コア数	4コア/プロセッサ
L1キャッシュ容量	64kByte/コア
L2キャッシュ容量	512kByte/コア
L3キャッシュ容量	2MByte/プロセッサ（コア間共有）

本資料では、Intel MPI Benchmarkを用いて、以下の3パターンの性能測定を行いました。

a. ノード内通信：2プロセス/1ノード、2プロセスを同一CPU内に配置
b. ノード内通信：2プロセス/1ノード、2プロセスを異なるCPUに配置
c. ノード間通信：1プロセス/1ノード×2ノード、どちらのノードもプロセスをCore0に配置

２．測定結果

ノード内通信性能、ノード間通信性能の結果を図1に示します。

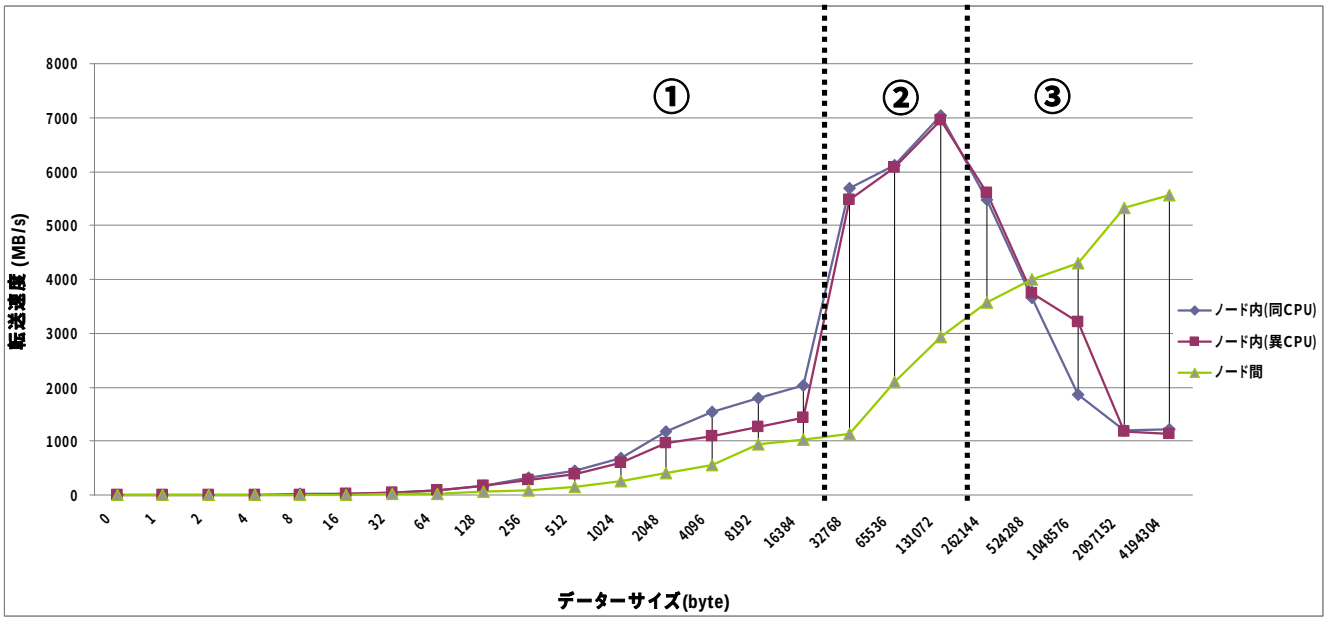

図1　ノード内、ノード間通信性能の測定結果

図1について、転送データーサイズによりノード内通信での結果を3つの領域に分け、各領域での結果について説明します。

① 転送データーサイズが16kByte以下の時
② 転送データーサイズが32kByte以上から128kByteまでの時
③ 転送データーサイズが256kByte以上の時

図1の②の領域で、極端に高い性能が得られていますが、実際のアプリケーションでは出すことが困難な性能です。（注意点）

① 転送データーサイズが16kByte以下の時

MPICH-MXのノード内通信では、転送データーサイズによって通信方法を切り替えており、16kByte以下の転送では共有mmap領域を中間バッファとして利用し、プロセス間でデーターの受け渡しを行います。

mmap領域を使ったプロセス間のデーターの受け渡しとなるため、実メモリー（主記憶）への書き込み、読み出しが発生します。そのためメモリーコピー性能が通信性能として現れます。

この範囲での通信性能は、実際のアプリケーションで出すことができます。

② 転送データーサイズが32kByte以上から128kByteまでの時

転送データーサイズが32kByte以上のMPICH-MXのノード内通信は、MXドライバーが送信側と受信側相互のメモリー空間にアクセスしてデーターコピーを行います。

このとき、Intel MPI BenchmarkのSendRecvでは、同一の送信バッファ、同一の受信バッファを繰り返し使用しているため、それらバッファがCPUのキャッシュに載る場合があります。例えば、転送データーサイズが32kByteの場合は、送受信合わせて64kByteのバッファがアクセスされますが、コアあたりのL2キャッシュが512kByteあるため、すべてL2キャッシュに載ることになります（キャッシュはwriteバック動作します）。つまり、32kByte以上のノード内通信においては、バッファへのアクセスがキャッシュへのアクセスで済むため、急激に性能が高くなっているように見えてしまいます。

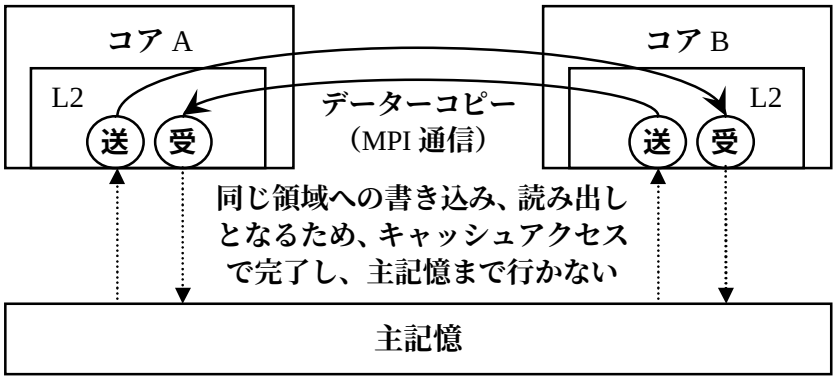

図2　②領域でのデーターコピー（MPI通信）の動作

③ 転送データーサイズが256kByte以上の時

転送データーサイズが128kByteの場合は、送受信合わせて256kByteのバッファでL2キャッシュに全て載りますが、転送データーサイズが256kByteになると、送受信合わせて512kByteのバッファが必要となりL2キャッシュでは足らず、L3キャッシュを使い始めるため、転送データーサイズが128kByteの時がグラフでのピーク性能となります。

Intel MPI Benchmarkプロセスを同一CPU内に配置した場合、異なるCPUに配置した場合のグラフは、転送データーサイズが②の範囲から512kBまではほぼ一致した変化をしていますが、1MByteのところで差（同一CPU内に配置した場合の性能が落ちる）が出ています。これは、転送データーサイズが1MByteの場合は、送受信合わせて2MByteのバッファが必要となり、異なるCPUに配置した場合は各プロセスで2MByteのL3キャッシュをほとんど占有できるのに対して、同一CPU内に配置した場合は2MByteのL3キャッシュでは足りず、実メモリーへのアクセスが発生するためです。

転送データーサイズが2MByteを超えると、プロセス配置によらずL3キャッシュでは足りなくなり、実メモリーへのアクセスとなるため、実メモリーのアクセス性能が現れてくることになります。

３．共有メモリー（SHMEM）を利用しない通信

Myrinetのノード間通信においては、ノード内通信時のように共有メモリーを利用するのではなく、Myrinetデバイスを利用したDMA（Direct Memory Access）通信を行います。

また、ノード内通信においても、環境変数MX_DISABLE_SHMEMに1を設定すると、MPI通信時に共有メモリーを利用しないようにすることが可能です。共有メモリーを利用しない場合に、同じ測定方法でノード内、ノード間通信の性能を測定すると、図3の結果となります。

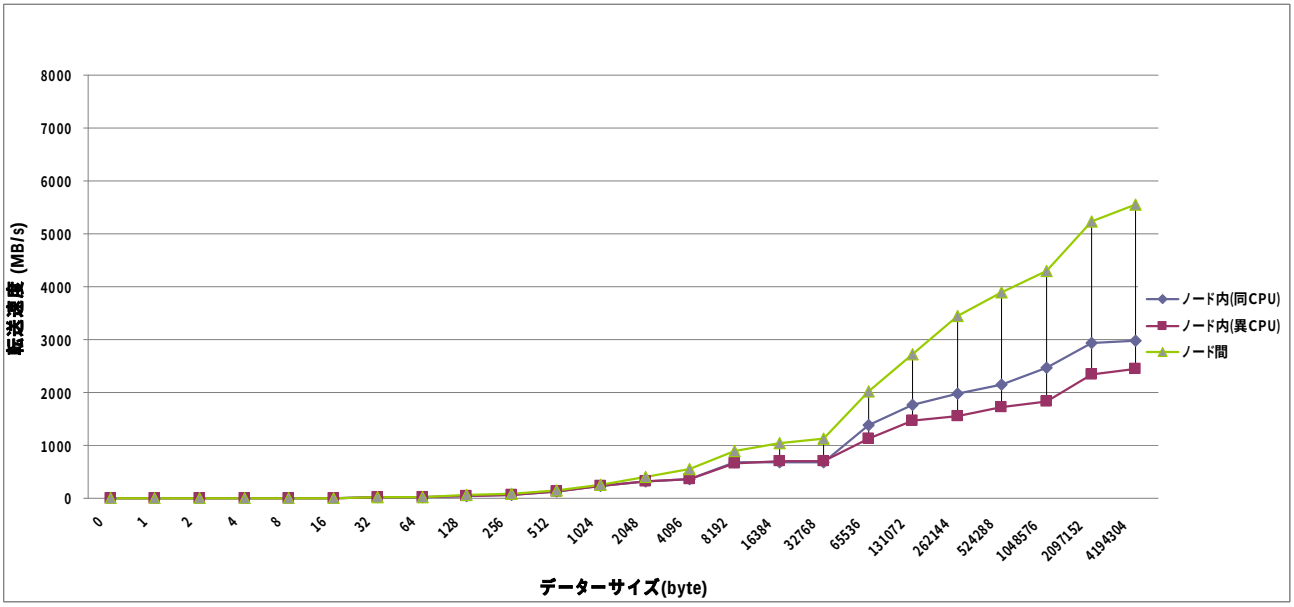

図3　ノード内、ノード間通信性能の測定結果（共有メモリーを利用しない時）

ノード内通信の性能は、図1と比較して、極端な変化がなく、データーサイズが大きくなるにつれて、性能が上がっていく安定したグラフになっています（図2のグラフは増加傾向のままですが、頭打ちになります）。

４．まとめ

Intel MPI Benchmarkは、図2の測定方法のように、1回の測定でも、複数回の繰り返し測定でも同じ性能が再現される通信を前提とした性能測定プログラムであると考えられ、図1のように、繰り返しによりキャッシュが効いてしまうような通信の性能測定には適していません。共有メモリー利用時の性能測定を行う場合は、キャッシュの影響があることを考慮して、実際に使用するアプリケーションのメモリーへのアクセスパターンや通信パターンを再現するようにしたプログラムでの性能測定をお願い致します。

以上