高速並列I/Oモデル「コンテナモデル」の汎用化に関する考察

1P−11

平野 聡 田沼 均 須崎 有康

電子技術総合研究所

1 はじめに

高速な並列計算機を実現する際に高いI/O性能は欠かすことができない要点である。特に大規模な科学技術計算やデータベース処理においては、主記憶の容量を超えるデータを扱う必要があるため、マスストレージからデータをローディングする事、ステージング／デステージングする事、結果を格納する事が高速にできなければならない。本論文では高速並列I/Oモデルである「コンテナモデル」を電総研で開発中の汎用並列マシンEM-X[2]のオペレーティングシステムに適用する際に必要な汎用化について基本的な考察を行なう。また、EM-XのI/Oの構成についても触れる。

2 コンテナモデルについて

コンテナモデル[4, 5, 6, 7]は東京大学において開発中の高並列関係データベースサーバであるスーパーデータベースコンピュータSDC[1, 3]のために開発された高速並列I/Oモデルである。

CPUの性能に対しI/Oの性能が低い「I/Oボトルネック」の存在が問題となっている。その解消を目的として並列I/Oの研究開発が行なわれているが、実際に作成されたマシンを検討すると、ある程度の性能向上は果たしているものの理論的な上限値には遠く及んでいない。その原因は、各々が高い性能を有するハードウェア構成要素（プロセッサ、ディスク、ネットワーク）を多数並置しても、その構成要素間のデータ転送を司るシステムソフトウェアが提供するI/Oモデルやデータ転送メカニズムが従来のオペレーティングシステムのモデルを踏襲しており、データインテンシブな応用を対象とする並列環境に適合しないためであると考えられる。

そこでこのソフトウェアのI/Oボトルネックを解消するために、ディスク、ネットワーク、プロセッサの関係として、I/Oデバイスからのデータがタスクとなってプロセッサを駆動する独自の高速並列入出力モデルである「コンテナモデル」を提案し、それに基づいたシステムソフトウェアを構築し有効性の実証を試みた。

コンテナモデルでは共通規格の容器「コンテナ」を用いて、意味階層の最も上位のプログラムが使用するデータの意味「最上位意味」のデータをそのままカプセル化し運搬する。システム中の全ての構成要素はコンテナを単位としてデータの交換を行なうため、システム中をデータが移動しても意味変換や無用なコピーは発生しない。データが格納されたコンテナをタスクと呼び、タスクを処理する実体をプロセスと定義する。タスクはI/Oによって生成され、プロセスに渡される。プロセスはタスクを処理し、演算結果から新たなタスクを生成してI/Oに送る。従来のOSのI/Oモデルで用いている read() システムコールのセマンティクスではプログラムがデータを必要とする時点で要求を発行するためデータ待ちが発生するが、コンテナモデルではデータをタスクとして連続的に受け取るため、データ待ちでプロセスが停止することはない。プロセスは、ディスクアクセスであるかネットワークによるプロセス間通信であるかにかかわらず、SDCのOSが提供する4つの基本関数を用いてコンテナをアクセスする。コンテナは並列度の高いN面バッファリングを用いて管理されており、コンテナをデバイスとプロセス間で高速に回転させ、少ないメモリ量でI/Oとプロセスによる処理を完全に並行して行うことが可能である。また、バッファリング、ディスクストライピング、ハッシュファイル、プロセス間通信、ガベージコレクション、負荷分散、フローコントロール、デッドロック防止及び回避[5]等を統一的に実現することが可能である。

コンテナモデルに基づくバッファ管理手法として、デバイスへの負荷に応じてバッファの容量を適応的に制御することによりバッファを通過するデータ流を調整し、より少ないバッファ容量で実行時間の短縮を図る適応化アルゴリズムを提案し、有効性を示した。[7]。

3 汎用並列マシンEM-Xへの適用

このようなコンテナモデルを電総研にて開発中のデータフローマシンEM-X[2]に適用することを考える。EM-Xは汎用の並列マシンであるため、専ら関係演算モデルのために考案されたコンテナモデルを拡張する必要がある。

EM-Xではハードウェアの制約からディスクをPEの集積されたPEノードとは別のI/O専用のI/Oノードに配置する。

3.1 任意の意味をサポートする機構

SDCでは最上位意味を関係演算モデルに固定したため、デバイスとコンテナ間の意味変換は比較的容易であった。汎用並列マシンにおいては任意の意味、即ち型の入出力を行なう必要がある。そこで、意味変換部をユーザプログラムの一部として記述し、実行時にPE上のオペレーティングシステム内部に登録、実行する機構を用意する。

ディスクからのデータの読み込みの場合、意味変換部はディスクからセクタイメージで送られてくるデータをコンテナの容量に収まる量に区切りつつ、プログラムが必要とする型に変換しながらコンテナに格納する。処理内容としては不要なデータの切捨て、データ中に含まれるポインタがコンテナを処理するPE上で意味をなすようにアドレスを変換する処理等がある。また、システムデフォルトとしてバイト、ワード、浮動小数点のストリーム用も用意する。I/Oノード上のオペレーティングシステムは、多数登録されている意味変換部のうちから、読み込んでいる最中のディスクブロックに対応する意味変

Consideration of generalization of a high performance parallel I/O model, Container Model

S.Hirano,H.Tanuma,K.Suzaki, ElectroTechnical Laboratory, Tsukuba

換部を活性化する。生成されたコンテナはネットワーク出力バッファに移され、ネットワークの負荷を一定に保ちながら送出される。

3.2 I/O ノード内でのコンテナモデル

I/O ノードの内部では、ディスクから読み込んだデータをディスクのデータ転送の終了を待たずにネットワーク負荷を一定以下に保ちつつネットワークに送出するためコンテナモデルを使用する。

I/O ノードがサポート可能なディスクの総転送速度はネットワークの転送速度により制限される。EM-X のネットワーク転送速度は 40MB/s が見込まれるため、その 9 割の 36MB/s をデータ転送用に、残り 4MB/s をコントロール用に使用すると仮定すると、3MB/s の転送速度を有するディスクドライブをネットワーク 1 ポート当たり 12 台使用するのが適当である。

EM-X の要素プロセッサでネットワークとのインターフェースを司る機能を有するプロセッサ EMC-Y はデータのネットワークへの送出性能が約 11MB/sec であるため、意味変換部のオーバーヘッドを考慮すると 1 PE あたり 3MB/s のディスク 2 台が適当である。従って、1 台の EMC-Y、2 台のディスクを 1 ユニットとし、3 ユニットを集積するボードを 2 枚一組として使用すると 36MB/s の実効転送速度を得ることが可能となる。

ディスクの単位時間のデータ転送量を 1 本とすると、メモリのバンド幅はディスクからの入力 2 本、ネットワークへの転送 2 本、ネットワークからの転送 2 本を収容するだけのバンド幅がなければならない。EMC-Y はネットワークへの転送 2 本、ネットワークからの転送 2 本、をプログラムで転送し、さらにディスクからの入力 2 本の DMA の切替えを同時に行なう必要がある。

3.3 PE 内部でのコンテナモデル

I/O ノードからネットワークを介してパケット (メッセージ) として送られてくるデータを受け取る方法として以下の 2 方法が考えられる。

1. データをトークンとして受け取る。
2. データをトークンとして扱わず、単にメモリ上に格納する。(コンテナを使用)

データをトークンとして受け取る方式 I/O ノードに read オペレーションを発行した後、指定したデータの個数に達するまでトークンを受け取りながら処理を施す。本方式は到着したデータに対し直ちに演算を施すため、オーバーヘッドは少ない。しかし、一般に読み込んだデータ全てに対し演算を施す必要がないことが多いため、プロセッサが浪費されてしまうことがある。また、PE の処理がデータ流に追い付けない場合、I/O ノードからのパケットが FIFO(ネットワークからのパケットを保持するハードウェア) に詰まって他 PE からのパケットを処理できなくなる状態を起きるため、デッドロック解消の機構が必要となる。

データをトークンとして扱わず、コンテナを使用する方式 複数個のコンテナをフリーリストで管理し、I/O ノードからのデータを順次コンテナに格納してゆく。PE がアイドル状態であったらトークンを発生し、処理を起動する。

本方式はメモリ上に格納されたデータのうち必要なデータにのみ演算を施すことが可能である。また、非同期 I/O であるため、PE がデータを必要とした時点で遅れなしでアクセスすることが可能である。しかし、ネットワークからのデータを一旦メモリに格納するコストが必要となる。このコストは EMC-Y の場合最短 2 クロック、最長 6 クロック (FIFO があふれた場合) であり、どちらも実行ユニットの CPU タイムを消費する。

このコストは PE が関与するメモリ書き込みルーチンを介さずに、ネットワークインターフェース部 (IBU) からメモリに直接アクセスするパスを設けることにより回避できる。ネットワークからのデータの読み込みと演算の同時実行も可能となる。

4 おわりに

科学技術計算や大規模データ処理では大容量ファイルの長いシーケンシャルアクセスが多いが、汎用の並列マシンとしては小さなデータの多頻度のアクセスも効率良くサポートする事が必要である。その他、プロセスのマイグレーションに追随する I/O、I/O ノード方式以外の構成のハードウェアへの対応等についても考察を進めて行きたい。

参考文献

[1] M. Kitsuregawa, S. Hirano, M. Harada, M. Nakamura, and M. Takagi. The super database computer (SDC): Architecture, algorithm and preliminary evaluation. *Proc. of HICSS-25*, 1992.

[2] 児玉, 甲村, 佐藤, 坂井, 山口. 高並列処理向け要素プロセッサ EMC-Y の設計. 並列処理シンポジウム *JSPP'92*, 1992.

[3] 平野, 原田, 中村, 小川, 楊, 喜連川, 高木. スーパーデータベースコンピュータ SDC のアーキテクチャ. 並列処理シンポジウム *JSPP'90*, 1990.

[4] 平野, 原田, 中村, 小川, 楊, 喜連川, 高木. スーパーデータベースコンピュータ SDC のソフトウエア. *SWoPP'90*, 1990.

[5] 平野, 原田, 中村, 相場, 鈴木, 喜連川, 高木. スーパーデータベースコンピュータ SDC に於けるデータ流制御方式. 情報処理学会第 *43* 回全国大会, 1991.

[6] 平野, 原田, 中村, 相場, 鈴木, 楊, 喜連川, 高木. SDC におけるモジュール群制御方式と 2 モジュール SDC の試作・評価. 並列処理シンポジウム *JSPP'91*, 1991.

[7] 平野, 原田, 中村, 鈴木, 喜連川, 高木. スーパーデータベースコンピュータ (SDC) におけるデータ流制御方式の評価. 情報処理学会第 *44* 回全国大会, 1992.