超並列計算機 SR2201 のファイル I/O 高速化方式

4Z－8

清水 正明† 鵜飼 敏之† 鍵政 豊彦† 藤田 不二男‡
†(株)日立製作所 中央研究所 ‡(株)日立製作所 ソフトウェア開発本部

1.はじめに

多数の演算プロセッサを持つ超並列計算機においては、少数の I/O 装置が複数の演算プロセッサからの I/O 要求を処理しなければならない。このとき単体の I/O 装置には、単一ファイルへのシーケンシャルアクセス時の性能ばかりではなく、多数のファイルへの同時 I/O 時の性能も求められる。

本稿では、超並列計算機 SR2201 においてのファイル I/O 高速化方式について報告する。

2.SR2201 のファイル I/O の概要

超並列計算機 SR2201 は演算を行う ProcessingUnit (PU)と入出力を行う I/O Unit(IOU)で構成し、これら Unit 間を高速ネットワークで接続している。通常 PU と IOU は 16:1 程度の比率で構成するので、平均すると一つの IOU が 16 個の PU からの I/O 要求を処理することになる。利用者は複数 PU や IOU を意識することなく、複数プロセスを複数の PU で動作させ、ファイルストライプ機能を利用して複数の IOU に I/O 要求を分散させることができる。

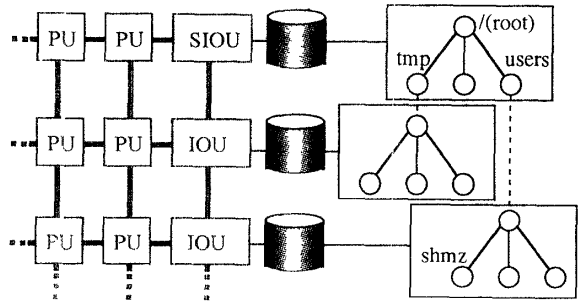

図1 SR2201 の構成

3.シーケンシャルファイル I/O 高速化方式

科学技術計算では、データの一括読み出し→演算→一括書き込みといった、フェーズの別れた処理をすることが多く、この場合単一ファイルに対する高いシーケンシャルファイル I/O 性能が求められる。

3.1 まとめ書き

SR2201 の OS である HI-UX/MPP はマイクロカーネル構成を採っており、PU ではファイルシステム機能等を持たなくて済む利点がある。しかし、統合カーネル構成の OS と比較すると一回あたりのシステムコールのオーバヘッドが大きく、小さな単位の I/O システムコールが多発するような使い方では性能を出しにくい。そこで、

・I/O 要求一括化(System Call,DeviceI/O 回数の削減)
・データコピー回数削減
・デバイスと OS のパイプライン動作

を行うことで OS のオーバヘッドを減らし、デバイスを効率的に動かして I/O 性能を高めることを考えた。

当初は OS の入出力キューに溜まった要求を 8KB 単位でデバイス I/O していたが、今回は OS の負荷に応じて 8KB～256KB までの可変量で一括して I/O する。一括化の最大量を 256KB に制限したのは、一括量を大きくしすぎるとデバイス I/O 時間が長くなり、終了を待つ OS 時間が無駄になるからである。デバイス I/O 中に、OS が直前の I/O の終了処理と直後の I/O の準備処理がちょうど終了するサイズが SR2201 では 256KB であった。

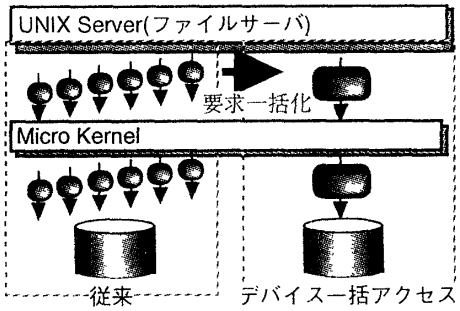

図2 一括化方式

3.2 先読み

書込みにおいては、入出力要求を OS の入出力キューに入れた時点で見かけ上の I/O を終了する非同期的な処理

High-speed File I/O Method on SR2201
Masaaki SHIMIZU,Toshiyuki UKAI, Toyohiko KAGIMASA,Fujio FUJITA, HITACHI,Ltd.

が可能である。したがって利用者プログラムはデバイス動作の終了を待たずに次の要求を発行でき、要求の一括化のみを行っていればOSとデバイスがパイプライン動作し、十分な性能を得ることができる。しかし読出しにおいては、一度来た要求をすべて完了して処理を戻さないと次の要求が来ない(同期的に処理が進む)のでデバイスが動いていない時間が多く、理想的なパイプライン動作ができない。そこで、シーケンシャルファイルアクセスの場合には、

・常に大単位先読み

を行うことで、利用者プログラムがデバイスI/Oの終了を待つことなく、常に先読みされているデータを得るようにした。

4.3BSD等のファイルシステム[1]では、先読みは一ブロック(8KB)だけ行うようになっていたが、今回は読出し単位拡大をあわせて大単位で先読みすることにした。先読みは一つのファイルあたり最大1MB程度行うが、書込みと同様にデバイスとOSのパイプライン動作をさせるために、256KBずつに一回の要求サイズを制限している。この方式により、シーケンシャル読出しに関しても、書込みと同程度の性能を出すことが可能になった。

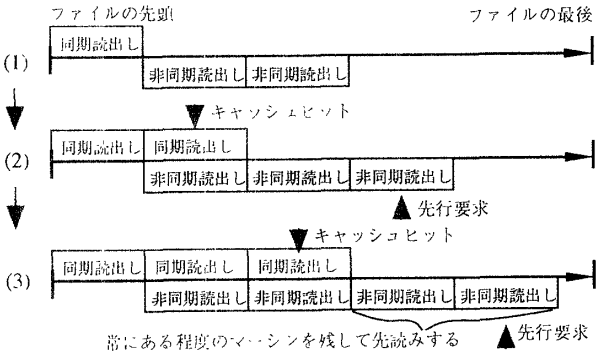

図3 先読み方式

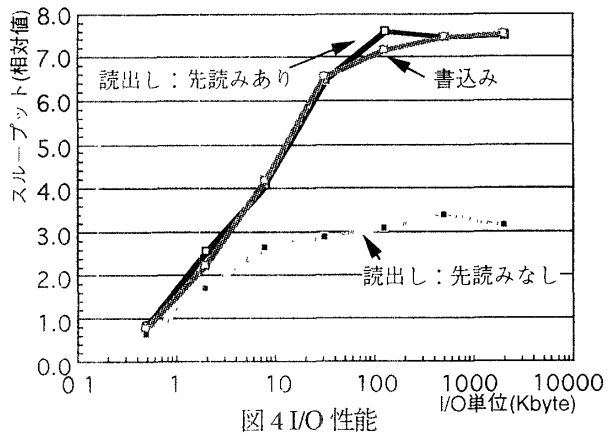

図4 I/O性能

4.並列I/O時の性能劣化防止方式

3章に述べた高速化方式で、単一のシーケンシャルファイルアクセスに関しては、デバイスの限界までの性能を出すことができた。しかし冒頭に述べたように、SR2201では一つのIOUで複数のPUからの要求を受けることがある。同時にI/Oしているファイル数が増えた場合、先読み領域(ファイルキャッシュ領域)が枯渇して先読みデータが捨てられ、先読みをした分の処理が無駄になる(性能が悪化する)。そこで、並列I/O時に先読み領域の取り合いにならない方式を考えた。これは、

・あるファイルの先読みデータが利用される前に捨てられたら、そのファイルに関する先読みは、一度Closeするまで停止する

という方式である。全部のファイルの先読みを同時に停止するのではなく、ファイル単位で停止する(先読みバッファを使用しなくなる)ことで、停止しなかったファイルに関しては、先読みが機能して高速にI/O可能である。この方式で、ファイル読出しの並列度が高いときでも、先読みなしの場合より性能が劣化することを防止した。

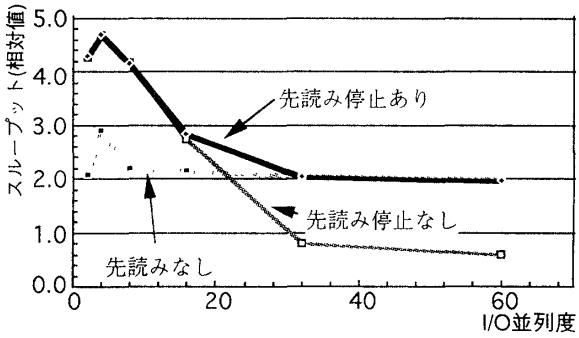

図5 先読み停止の効果

5.おわりに

超並列計算機SR2201において、シーケンシャルファイルI/O高速化方式として、要求一括化、OSとデバイスのパイプライン動作、そして常にファイルの先読みを行なう機構を実現した。また、並列読み出しの際に先読みバッファの不足で性能が劣化することを防止した。これらの方式により、従来の二倍以上の性能が得られた。

6.参考文献

[1] S.J.Leffer,et al：The Design and Implementation of the 4.3BSD UNIX Operating System,Addison-Wesley(1989)