2.2.3. 書式付 I/O 性能

上智大学　南部伸孝
富士通株式会社　内藤俊也、杉崎由典

1. はじめに

I/O 処理に要する時間の内訳は、システムの実 I/O 時間＋ランタイムの書式処理時間となっている。

real*8 write プログラムを用いた書式付 I/O 時間を富士通コンパイラで評価した所、システム時間が約 12%であり、ランタイム時間が約 88%ということが確認された。

即ち、書式付 I/O 性能は、ランタイムの性能に大きく影響される。ここでは、ランタイムの書式付 I/O 処理性能という観点で評価を行う。

2. 測定プログラム

以下のプログラムを用いた。

【real*8 プログラム】
r8(7500000) ⇒５７．２Ｍバイト

●real*8 write

```
open(11,file='iof_r8.dat',form='formatted',status='new',err=999)
call gettod(t(1,1))
write(11,'(4d24.15)') r8
call flush(11)
call gettod(t(2,1))
close(11)
```

●real*8 read

```
open(11,file='iof_r8.dat',form='formatted',status='old')
call gettod(t(1,2))
read(11,'(4d24.15)') r8
call flush(11)
call gettod(t(2,2))
close(11)
```

上記は real*8 の例であるが、型別にそれぞれ write と read を測定した。

また、各型毎の I/O 対象のファイルのサイズを以下に挙げる。ただし、ファイルのサイズは、配列サイズではなく、実際に入出力処理する対象のファイルのサイズを示している。

表 1.　入出力対象のファイルサイズ

	INTEGER*4	REAL*4	REAL*8	COMPLEX*8	COMPLEX*16
要素数 7.5M 個の I/O 対象 Filesize(byte)	75750000	181875000	181875000	363750000	363750000

3. 評価環境

評価したマシン環境及びコンパイラを以下に挙げる。

表 2.　評価対象コンパイラ

コンパイラ	バージョン
富士通	3.2／3.0
Intel	10.1
PGI	7.1－3
PathScale	3.3.1
日立	－

表 3.　評価対象マシン環境

	PRIMERGY RX200 S3	SR11000	SR11000	SR16000	FX1	HX600	PRIMEQUEST 580
ﾌﾟﾛｾｯｻ	Woodcrest	POWER5	POWER5+	POWER6	SPARC 64 VII	Opteron	Itanium2
周波数	3.0GHz	1.9GHz	2.3GHz	4.7GHz	2.52GHz	2.3GHz	1.6GHz
ﾌｧｲﾙｼｽﾃﾑ	ext3/srfs	mmfs	mmfs	nfs3,mmfs	srfs/nfs/tmpfs	nfs/ext2	srfs/tmpfs

4.　測定結果及び評価

評価の観点として以下の点が挙げられる。
- 変数の型
- total write/read 性能
- マシン
- ファイルシステム

それぞれの観点で評価を行った。

4.1　型別の性能評価

評価する I/O の write/read 種別と変数型として以下を対象とした。
- integer*4　write/read
- real*4　write/read
- real*8　write/read
- complex*8　write/read
- complex*16　write/read

各社のコンパイラを用いて、PRIMERGY RX200S3 上で測定を行った。
結果は以下の通り。

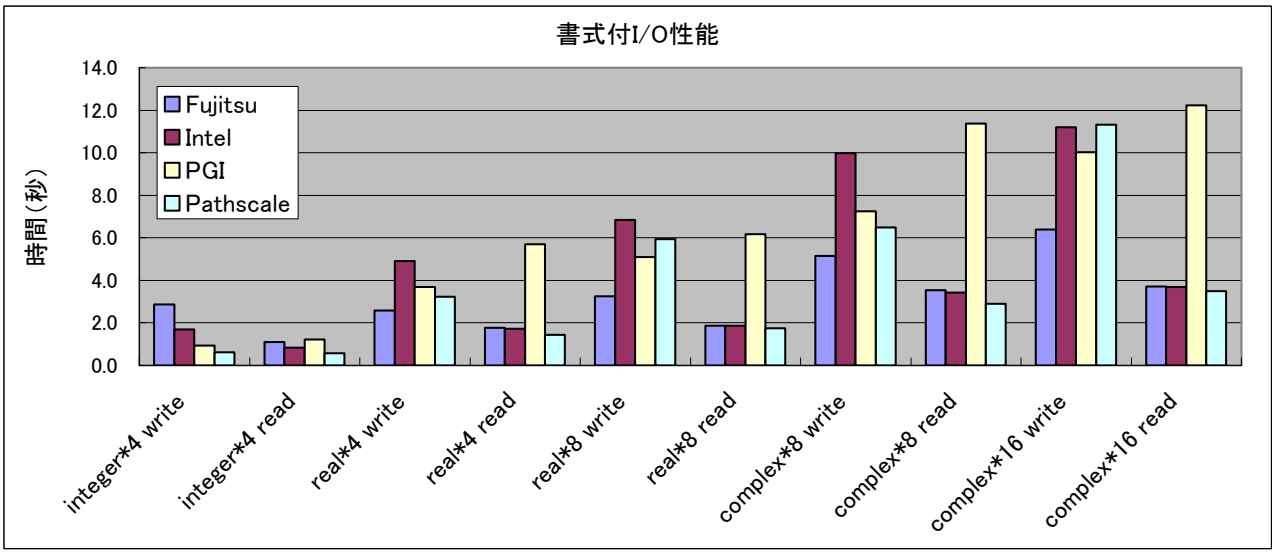

図 1.　各社のコンパイラで型別の測定結果

富士通コンパイラは、real,complex 型で良い結果となっている。

4.2　Total,write/read 性能評価

前節で測定した値に対して、以下のように結果時間を合計して評価した。
- 各型の write 合計時間
- 各型の read 合計時間
- 各型の write/read 合計時間

各社のコンパイラを用いて、PRIMERGY RX200S3 上で測定を行った。
結果は以下の通り。

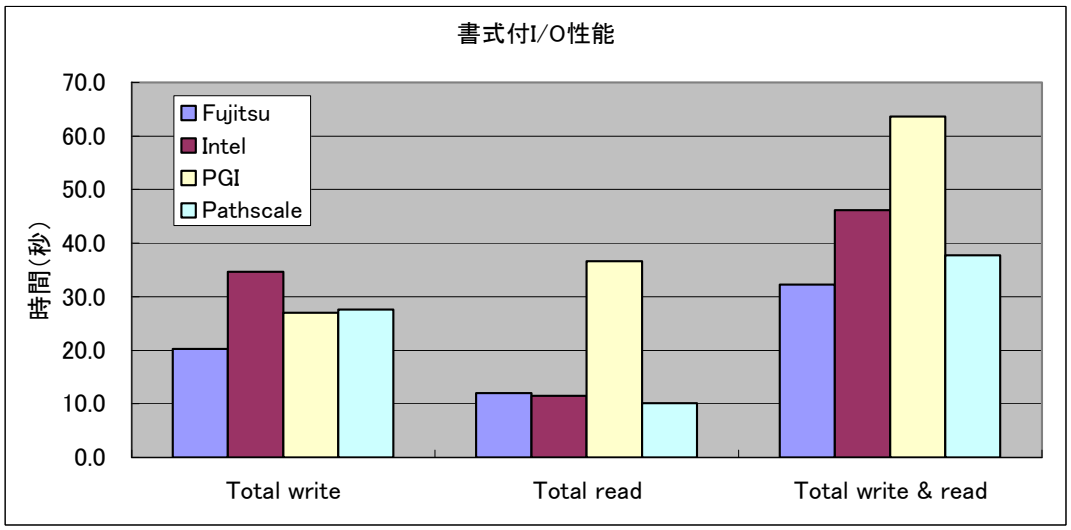

図 2. 各社のコンパイラで write/read 合計の結果

富士通コンパイラは、合計時間でも良い結果となっている。

4.3 マシン毎の型別の性能評価

各社のマシンで変数型それぞれに対して評価を行った。また、複数のファイルシステムで測定した。対象マシン及びファイルシステムは以下の通り。

- PRIMERGY RX200S3 ext2
- PRIMERGY RX200S3 srfs
- SR11000 mmfs
- SR16000 nfs3

各社のコンパイラを用いて、測定を行った。
結果は以下の通り。

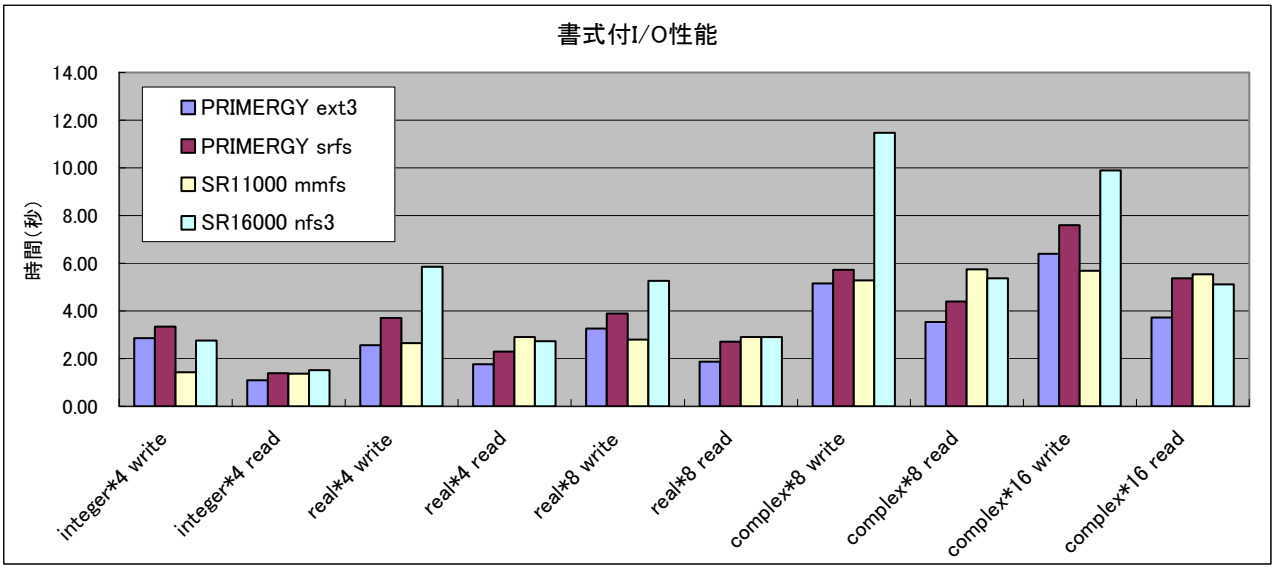

図 3. 各社のマシン及びファイルシステムで型別の測定結果

PRIMERGY は read で良い性能、SR11000,SR16000 は nfs3 よりも mmfs の方が良い性能となっている。

九州大学のシステムである PRIMERGY RX200S3(srfs)と SR11000(mmfs)で比較すると、write 性能では SR11000 が有利、read では PRIMERGY RX200S3 が有利となったが、性能差は大きくなく、ほぼ同等の性能といえる。

4.4 マシン毎の total,write,read 性能評価

前節で測定した値に対して、以下のように結果時間を合計して評価した。

・各型の write 合計時間
・各型の read 合計時間
・各型の write/read 合計時間

各社のコンパイラを用いて測定を行った。
結果は以下の通り。

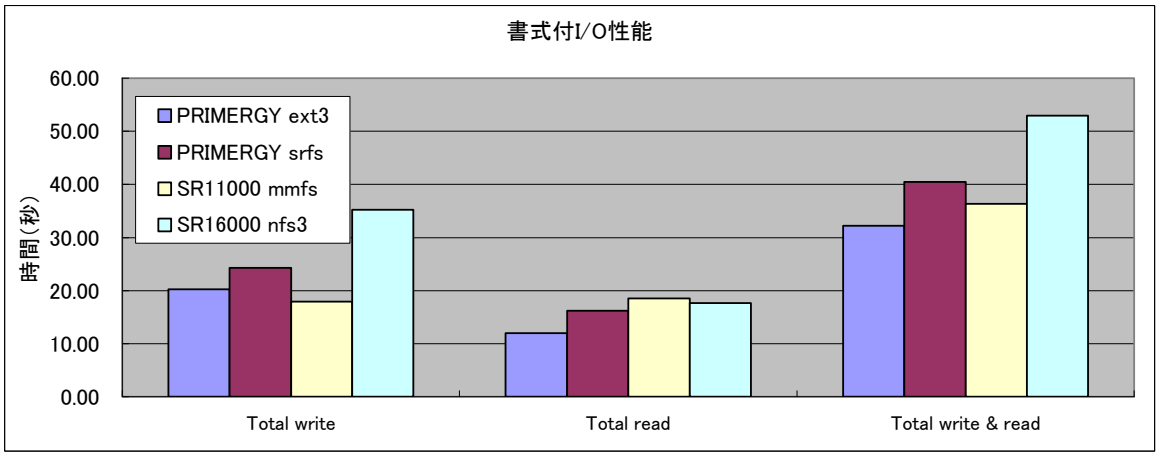

図 4. 各社のマシン及びファイルシステムで write/read 合計の結果

total で見ると、nfs3 以外はほぼ同等の性能である。

複数のマシンを利用して、I/O 性能の測定を行ったが、I/O 性能はファイルシステムやメモリの性能に大きく依存するため、同じシステムで評価する必要がある。

4.5 ファイルシステム毎の型別評価

ファイルシステムを変化させての測定を実施した。
各社のマシン及びファイルシステムにおいて、変数型それぞれに対して評価を行った。
結果は以下の通り。

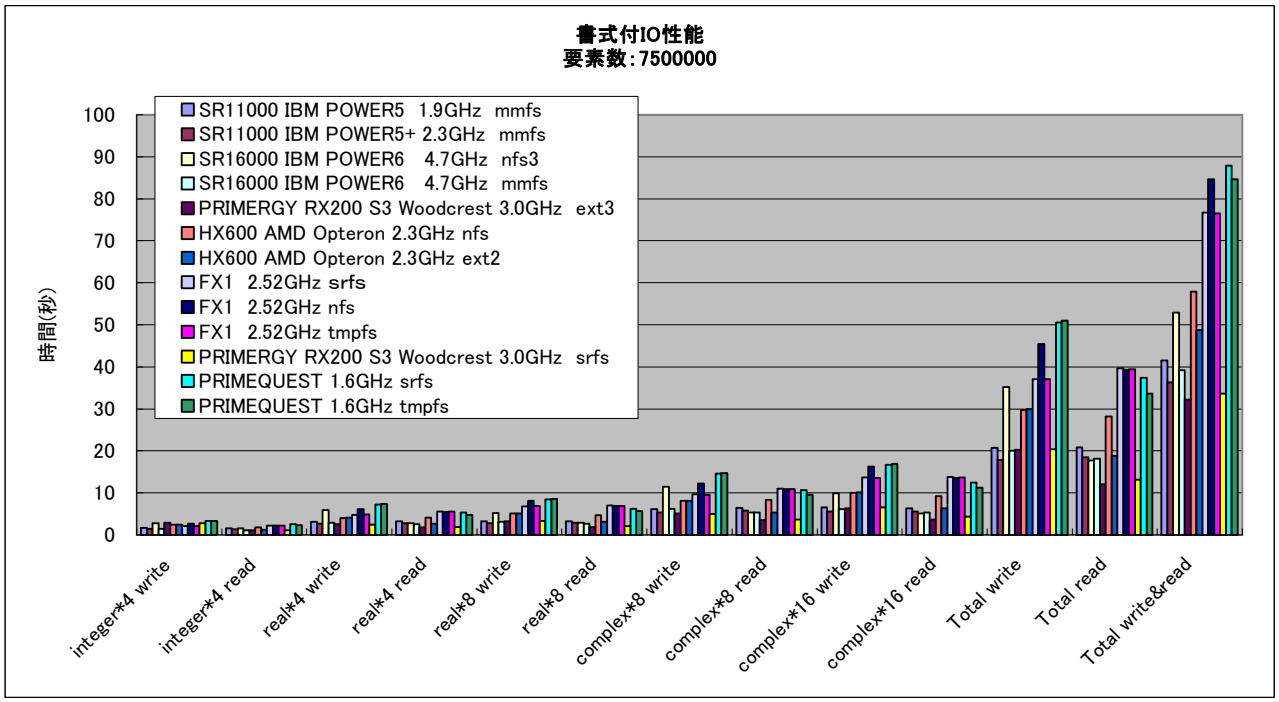

図 5. ファイルシステムで型別の測定結果

同一マシンでは、tmpfs,srfs が良好であり、nfs は遅いという結果となった。

次に、上記の結果から、各社のマシン毎に結果を抽出して、同一マシン上でのファイルシステム毎の性能を評価する。

(1)SR11000,SR16000 抽出

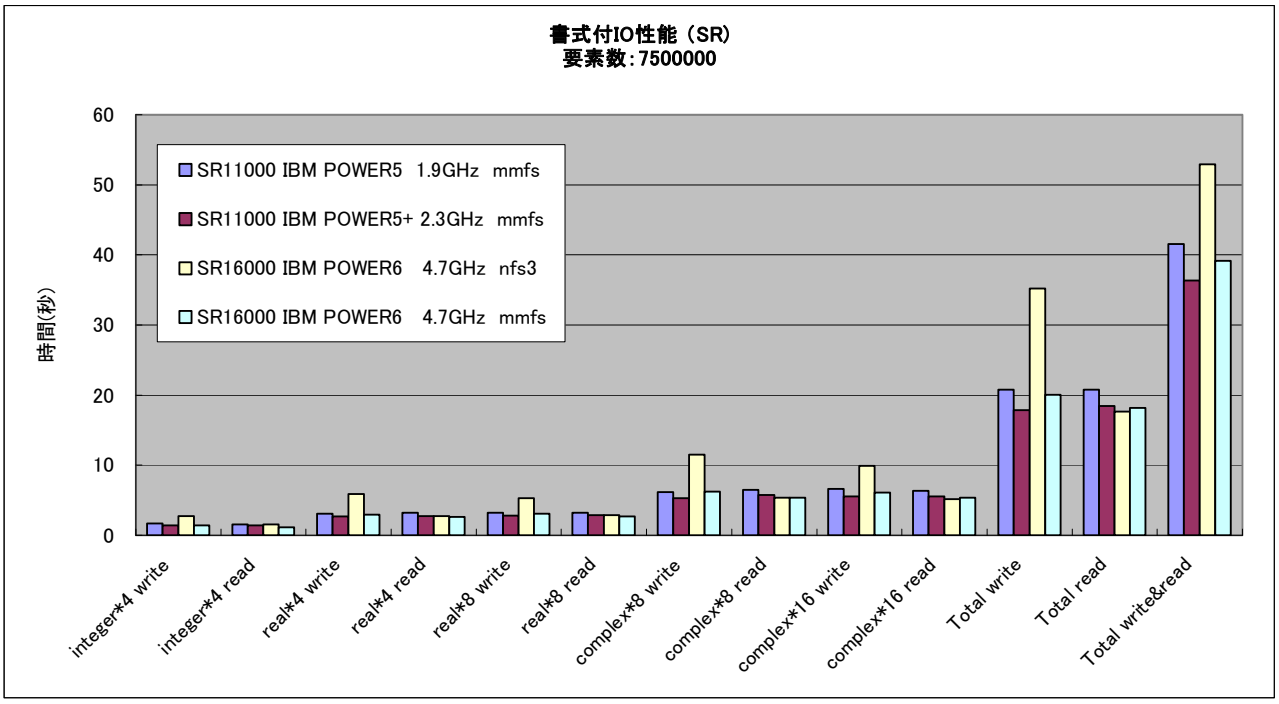

図 6. ファイルシステムで型別の測定結果（SR 抽出）

mmfs と nfs を比較すると、mmfs の方が write 性能では約 1.72 倍速い結果となった。

(2) FX1 抽出

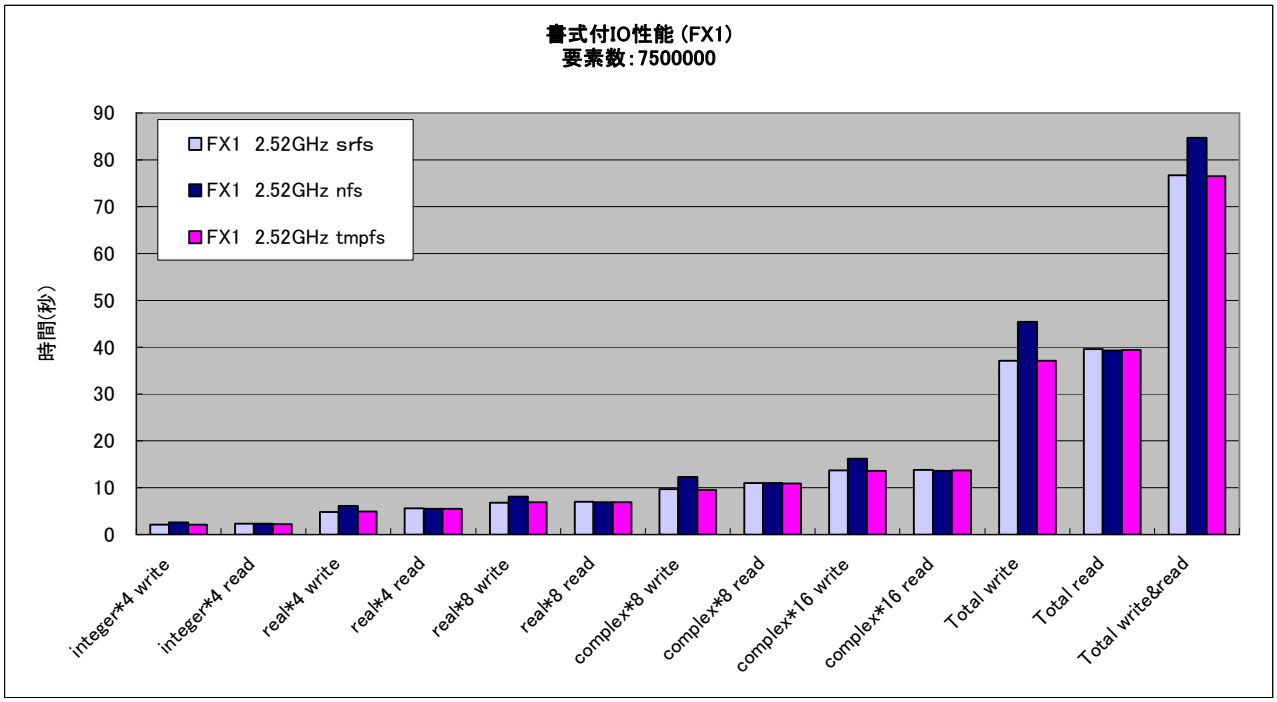

図 7. ファイルシステムで型別の測定結果（FX1 抽出）

srfs と nfs を比較すると、srfs の方が write では約 1.22 倍速い結果となった。

(3) PRIMERGY,HX600,PRIMEQUEST 抽出

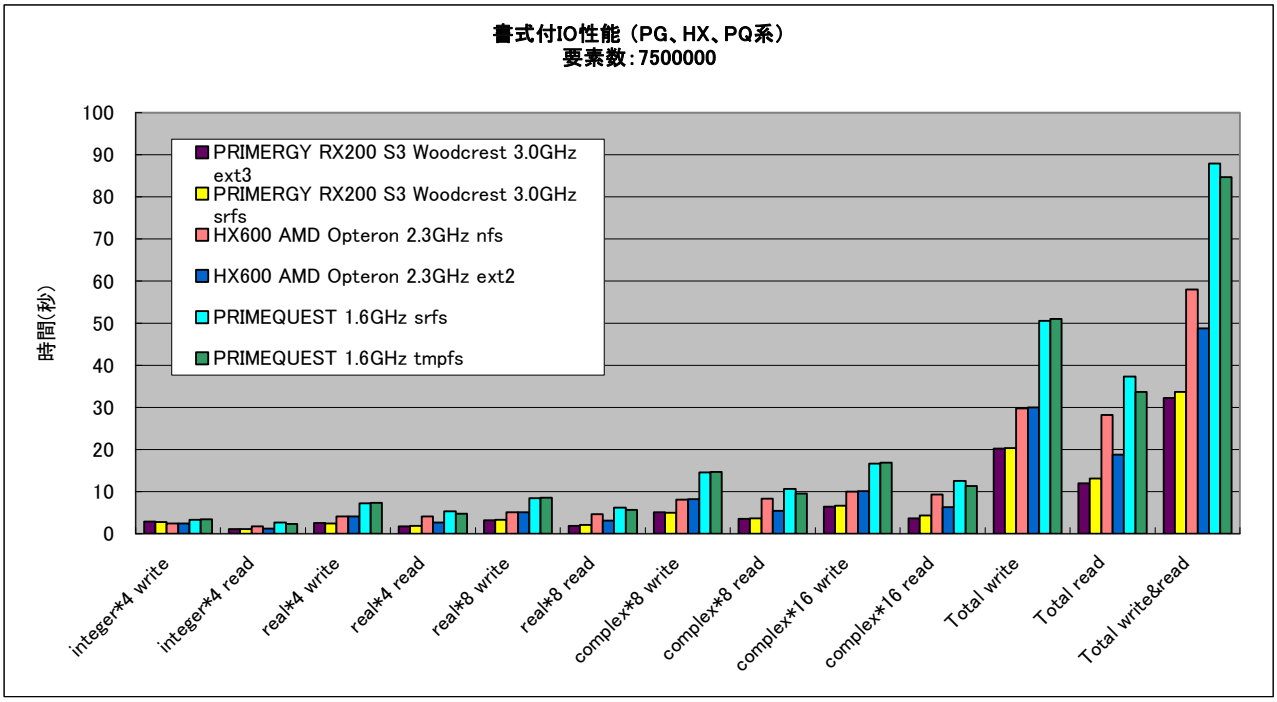

図 8. ファイルシステムで型別の測定結果（PRIMERGY,HX600,PRIMEQUEST 抽出）

4.6 ファイルシステムを統一した際のマシン機種毎の性能評価

これまでの測定結果から、ファイルシステム種別を同一（高速ファイルシステム：srfs,mmfs）にすることにより、ファイルシステムの性能差の影響を排除した性能評価を行った。

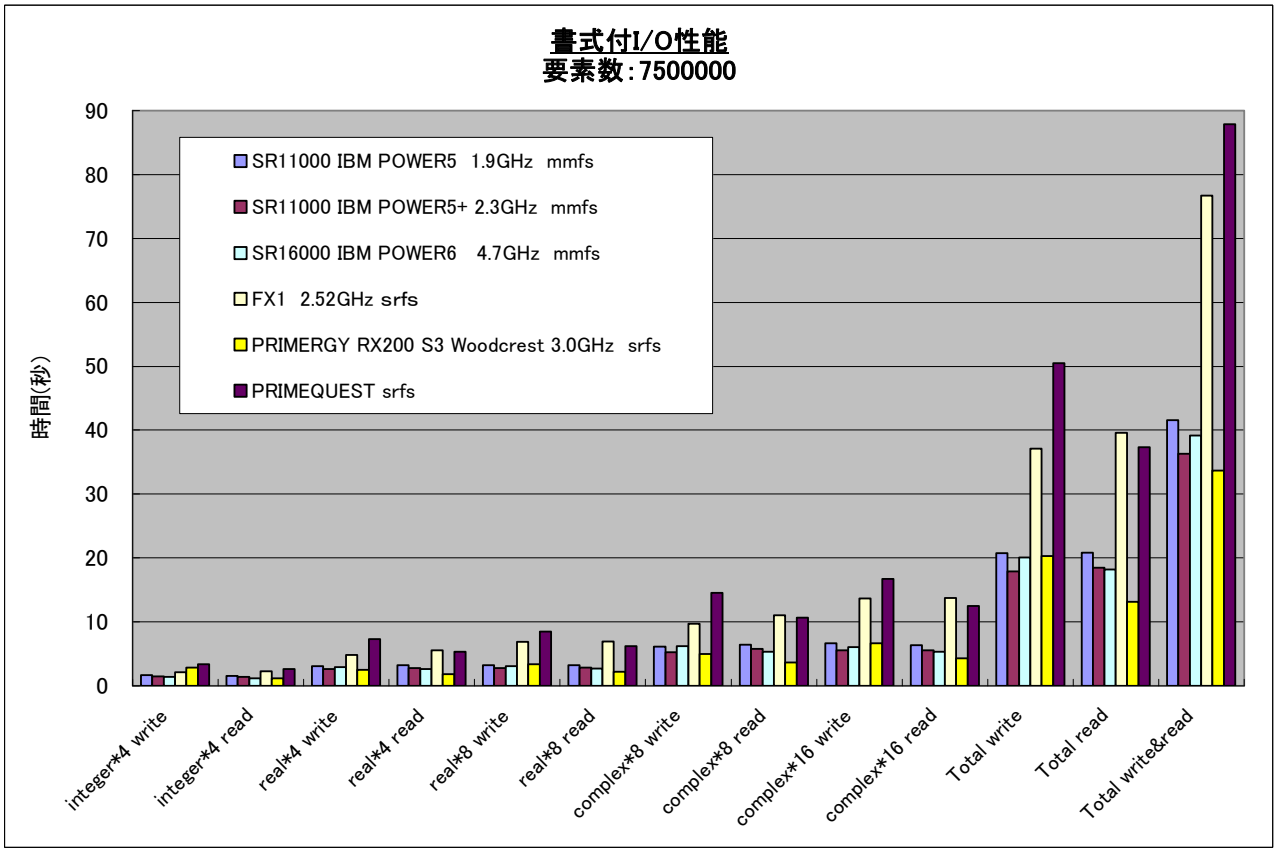

図 9. ファイルシステムを同一としたマシン別の性能比較

PRIMERGY は read で若干良く、SR11000 は write で若干良い。ただし、両者の性能はほぼ同等といえる。

上記の評価では、CPU 周波数の高いマシンが有利となってしまう。書式付 I/O 処理がランタイムの書式処理の性能差であることと、ランタイムの処理は CPU 周波数の影響が大きいことを考慮する必要がある。

そこで、CPU 周波数の低いマシンの性能を補正し、CPU 周波数の違いによる影響を排除した。PRIMERGY RX200S3 の CPU 周波数比を乗算することで補正し、PRIMERGY RX200S3 の性能を 1 とした相対的な性能を比較した。

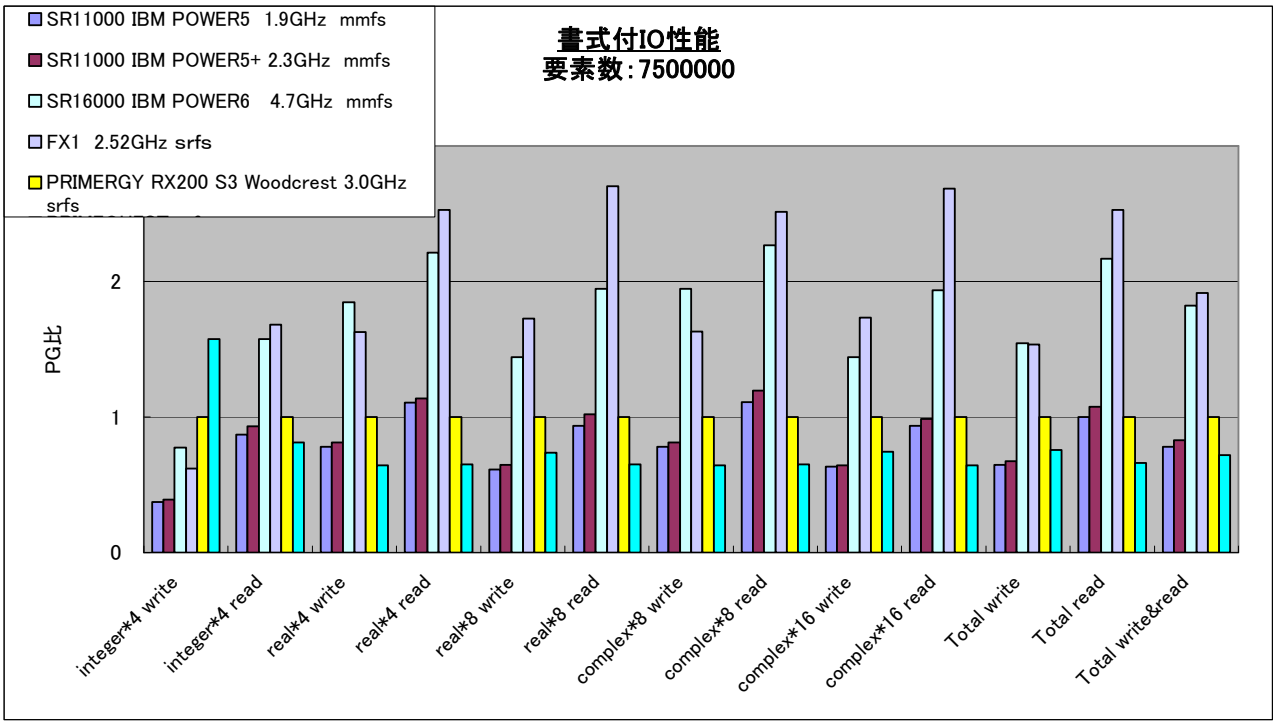

図 10. ファイルシステムを同一としたマシン別の性能比較（周波数比補正）

補正した結果から、PRIMERGY は周波数比で見ると妥当な性能といえる。（ただし、integer*4 は除く）

SR16000 は CPU 周波数の割には性能が出ていない。また、FX1 は浮動小数点の処理が遅いという結果となった。

4.7 測定範囲について

書式付 I/O 処理の測定が、システムの I/O 処理及びランタイム処理のどの範囲まで含まれるのかについて述べる。

(1)書式付 write 性能

RTS 性能（書式変換処理＋バッファへのコピー）＋I/O キャッシュ性能

注）fflush は I/O キャッシュへの flush であり、ディスクへの書込でない。（富士通のシステムではこの様な処理となっている）

(2)書式付 read 性能

ディスクからの読込処理＋RTS 性能（書式変換処理）

注）open 文の処理は、最初の read 文で実行される。そのため、read には open 文のコストも含まれてしまう。（富士通のシステムではこの様な処理となっている）

表 4. 書式付 I/O 処理のコスト内訳

		RTS 処理	RTS(%)	Fflush	Fflush(%)
FX1	Write(秒)	2.044661	99.94 %	0.001131	0.06 %
	Read (秒)	2.285298	100.00 %	0.000001	0.00 %
SR11000 (POWER5)	Write(秒)	1.789408	99.58 %	0.007491	0.42 %
	Read (秒)	1.566522	100.00 %	0.000001	0.00 %

4.8　Sun コンパイラとの比較

FX1 の性能が遅い点に関して、コンパイラの違いによる性能差を確認するため、SUN コンパイラと富士通コンパイラの結果と比較を行った。また、富士通コンパイラの次版は、FX1 のハード機能(prefetch)を意識した性能向上を行ったライブラリであり、こちらも合わせて比較した。

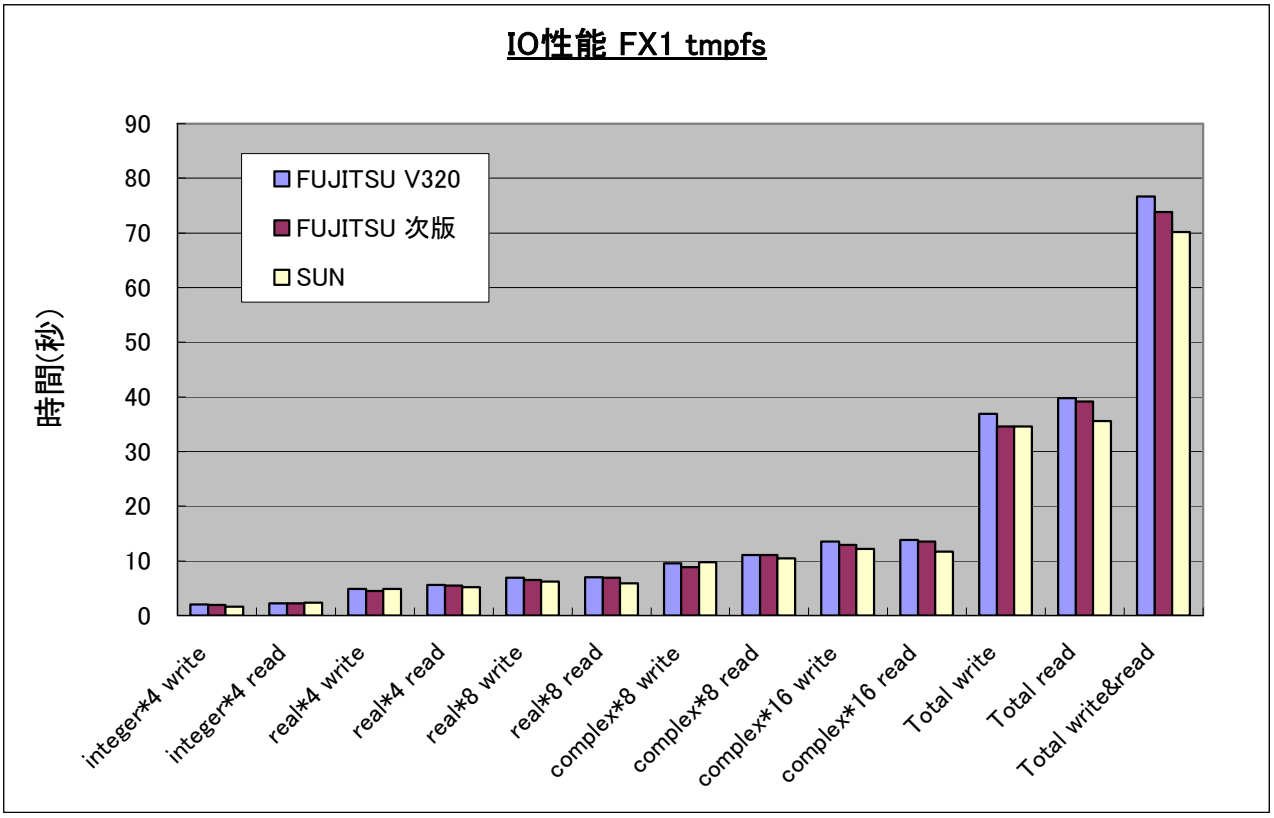

図 11. SUN コンパイラとの性能比較

結果から、多少の性能の違いはあったが、大きな性能差は確認出来なかった。

4.9　FX1 と PRIMERGY との性能差

まず、書式付入出力文で動作するランタイムの機能とコストの割合を調査した。結果を以下に示す。

書式付入出力文で動作する機能とその割合：

・書式付入出力部分制御、文の組み合わせチェック、データ転送　(5%)
・書式解析とチェック　(10%)
・データ編集　(15%)
・データ変換　(70%)

データ変換処理のコストが高く、この処理の差が大きいと推測する。

次に、FX1 と PRIMERGY の性能差を real*8 write と real*8 read でそれぞれ比較した。

real*8 の write 部分で約 2 倍、read 部分で約 3 倍の性能差がある。(クロック比を加味した値では、write 部分で約 1.6 倍、read 部分で約 2.65 倍の性能差がある)

ランタイムシステムライブラリの中で性能差が顕著な部位について調査すべきである。

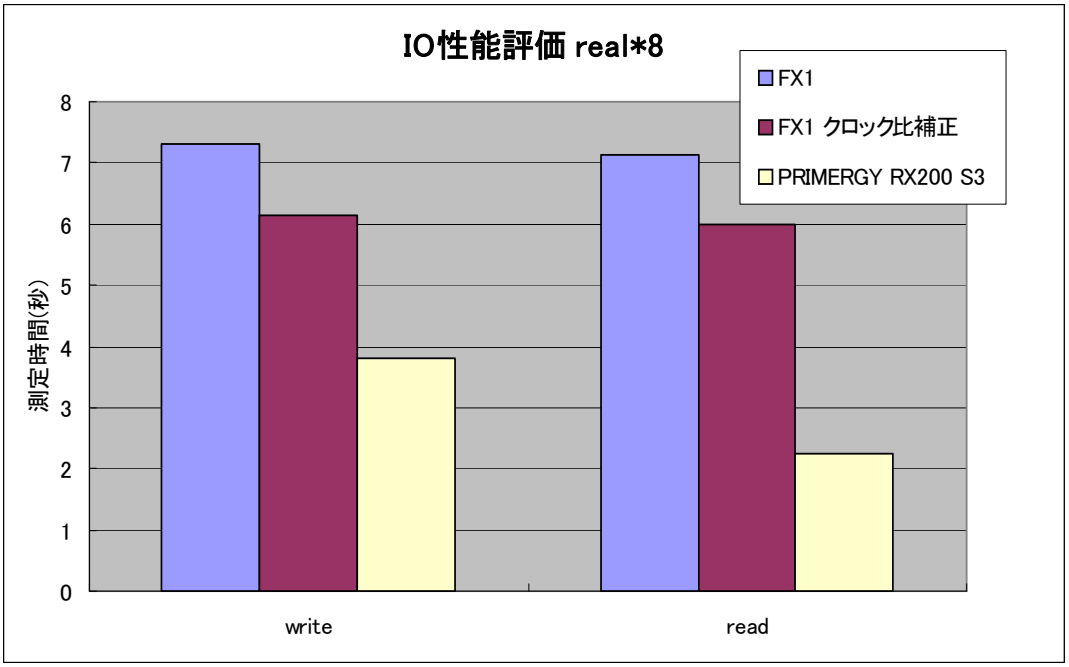

図 12. real*8 型 write/read 性能の FX1 と PRIMERGY との性能比較

(1) real*8 write

real*8 write 処理の中で、ライブラリコストが大きいのは、E 型変換処理手続き。E 型変換処理手続き以外の部分では、性能差はない。

表 5. write ライブラリコスト比較 1

	FX1(クロック比補正)	PRIMERGY RX200S3	性能差
E 型変換処理	4.84 秒	2.27 秒	1.91 倍
これ以外の処理	1.31 秒	1.37 秒	0.96 倍

E 型変換処理手続きの高コスト手続き内に、高コストループが 2 つ含まれる。これらのループ以外では、大きな性能差は見られなかった。(1.3 倍)

表 6. write ライブラリコスト比較 2

	FX1(クロック比補正)	PRIMERGY RX200S3	性能差
ループ 1	1.55 秒	0.73 秒	2.53 倍
ループ 2	1.29 秒	0.58 秒	2.65 倍
これ以外のループ	1.11 秒	0.85 秒	1.31 倍

性能差原因：

高コストループ 1，ループ 2 部分には、FX1 と PRIMERGY RX200S3 で、型変換の命令セットの違いによる以下の影響がある。

- FX1 では、r8 から i4 への型変換の際に、メモリへの store と load が発生する。
- PRIMERGY RX200S3 では､r8 から i4 への型変換の際に､メモリへの store と load が発生しない。

■FX1 r8→i4 型変換アセンブラ(例)

```
fdtoi   %f26,%f28
st      %f28,[%fp+1991]
ldsw    [%fp+1991],%g4
```

■PRIMERGY RX200S3 r8→i4 型変換アセンブラ(例)

```
cvttsd2si   %xmm3,%edx
```

図 13. real*8 write アセンブラ比較

注）FX1 では、「型変換→メモリへの store→メモリからの load」という処理になっている。

(2) real*8 read

real*8 read 処理の中で、ライブラリコストが大きいのは、10 進数部処理手続き、指数部処理手続きの 2 つ。その他の部分は、小さなコストの集まりであるため、調査対象から除外した。

表 7. read ライブラリコスト比較

	FX1(クロック比補正)	PRIMERGY RX200S3	性能差
10 進数部処理	1.49 秒	0.52 秒	2.83 倍
指数部処理	1.50 秒	0.50 秒	3.03 倍
これ以外の処理	3.00 秒	1.24 秒	2.42 倍

10 進数部処理手続きと指数部処理手続きについては、FX1 と PRIMERGY RX200S3 では switch 文の処理に違いがある。

- FX1 のライブラリは富士通 C コンパイラで翻訳されており、pic オプションを指定した時に、switch 文がジャンプテーブルを使用せず、case 文を一つずつ判定(if)し処理される。
- PRIMERGY RX200S3 の場合は、intel C コンパイラで翻訳されており、pic オプションを指定しても switch 文はジャンプテーブルを使用して処理されている。

※ pic オプション：位置独立コード(PIC) を生成することを指示するオプション

性能差原因：

アセンブラで確認すると、ジャンプテーブルを使用しない FX1 では、case の数だけ cmp 命令が生成されるため、命令数が増加している。

■switch文ソース（例）

```
for (i=0;   i < n; i++) {
    switch(値) {
    case   '0' : case   '1' : case   '2' : case   '3' : case   '4' :
    case   '5' : case   '6' : case   '7' : case   '8' : case   '9' :
            break ;
    case   ' ' :
            処理1
            break ;
    case   '-' :
            処理2
            return   4   ;
    case   '+' :
            処理3
            return   4   ;
    case   'e' : case   'd' : case   'q' :
    case   'E' : case   'D' : case   'Q' :
            処理4
            return   3   ;
    default :
            return   6   ;
    }
}
```

■FX1 アセンブラ（例）

```
.SSN153:
! 36355

        ldsb    [%o4],%o4
        cmp     %o4,48
        be,pt   %icc, .L1494
        nop

(省略：caseの数だけcmp命令が生成される。)

! 36355
        cmp     %o4,69
        be,pt   %icc, .L1492
        nop
! 36355
        cmp     %o4,68
        be,pt   %icc, .L1492
        nop
! 36355
        cmp     %o4,81
        be,pt   %icc, .L1492
        nop
```

■PRIMERGY RX200S3 アセンブラ(例)

```
# 24691
        jmp *   %rdx

以下のようなジャンプテーブルを生成し、使用している。

.LBT2:
        .quad   .LL1552
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        (省略)
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1563
        .quad   .LL1580
```

図 14. real*8 read アセンブラ比較

(3) 性能差まとめ

性能差の原因として、FX1 では型変換処理（命令セットの違い）と switch 文の処理が原因であることが判明した。

型変換の改善策として、型変換をすることは処理上必要な部分であるため、ランタイムシステムライブラリのプログラム修正（型変換しない）による改善は難しい。しかし、命令密度を向上することにより、改善できないか検討している。

switch 文の処理については、pic オプション指定時でもジャンプテーブルを使用した処理をするように、コンパイラの改善を検討している。

以上