

ベクトル並列型スーパーコンピューター SR8000 利用ガイド

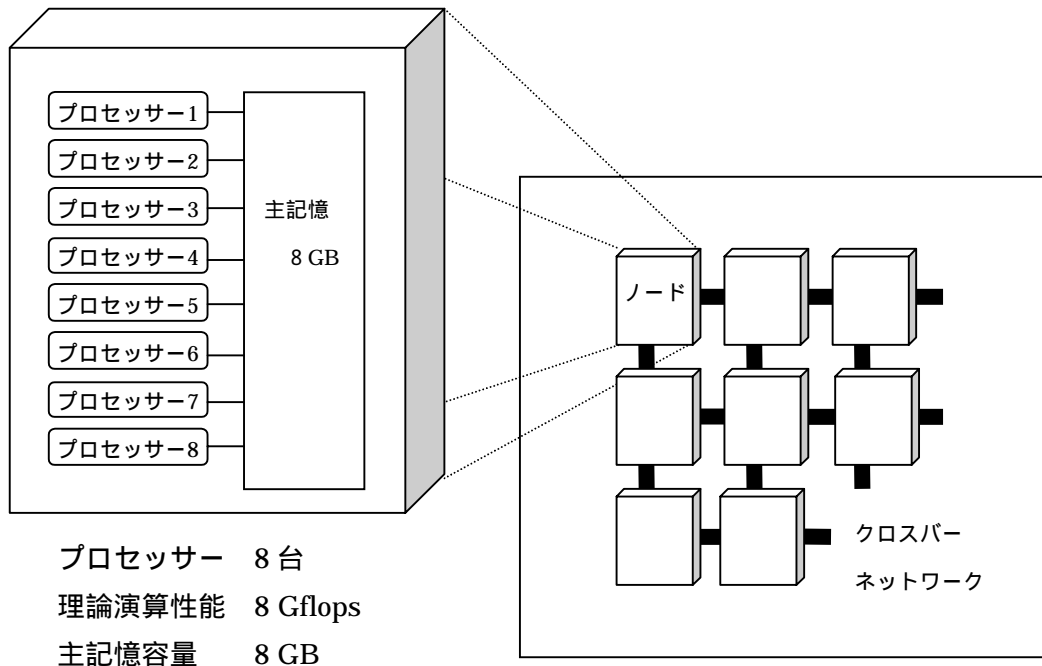
システム運用掛 宮崎 洋
佐々木 一孝

本センターでは 1999 年 5 月 6 日より「ベクトル並列型スーパーコンピューター SR8000 (日立製作所)」の正式運用を行っております。SR8000 は当初よりベテランの利用者の方々によって、高度に活用されていますが、UNIX や並列計算の経験の少ない方にとっては機能、構成の複雑さから戸惑いも多いかと思えます。そのため、これから御利用頂く方々の道標となるよう、SR8000 特有の機能やセンター固有の設定について記述しましたので御参照下さい。なお、内容は 2000 年 7 月 1 日現在の設定を反映しておりますが、今後変更となる場合があります。最新の情報は必ずスーパーコンピューティングニュースや Web ページ (<http://www.cc.u-tokyo.ac.jp/>) にてご確認頂きますようお願い致します。

1. システム構成

本センターのベクトル並列型スーパーコンピューター SR8000 は 1 台の理論演算性能が 8GFlops、主記憶容量が 8GB の演算ノードを 128 台搭載する並列型計算機です。各演算ノードは 8 つのマイクロプロセッサで構成 (SMP) されており、単体ノードによるスカラージョブの多重実行や並列処理が可能です。また、これらのノードを複数使用した並列処理も可能なため、多様なジョブ形態に対応できます。

- ノード間並列処理
複数のノードを使用した分散メモリー型の並列処理ができます。MPI や PVM、PARALLELWARE などの通信ライブラリーを使用してクロスパーネットワーク経由で通信します。
- ノード内並列処理
ノード内の複数プロセッサを使用した共有メモリー型の並列処理ができます。プログラムをスレッド分割して各プロセッサで並列実行する要素並列処理や、MPI が生成するプロセスを各プロセッサに配置して実行するノード内 MPI 等が可能です。

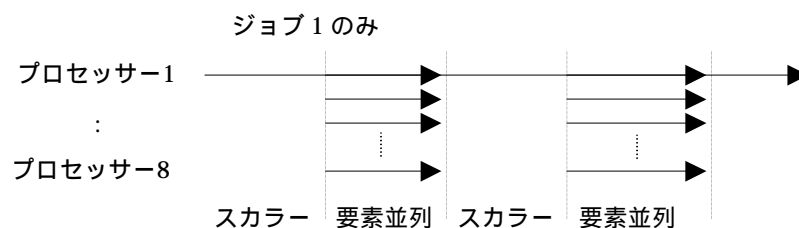


(要素並列実行)

各演算ノードは 8 つのマイクロプロセッサで構成され、演算処理をベクトル要素ごとに並列実行する機能をもっています。コンパイラによりスレッドと呼ばれる並列処理単位に分割 (要素並列化) されたプログラムは、複数のプロセッサに別々に割り当てられ、並列処理により高速に実行されます。一方、要素並列化されていないプログラム (スカラープログラム) は単体ノード内で他の利用者ジョブと混在して多重実行します。

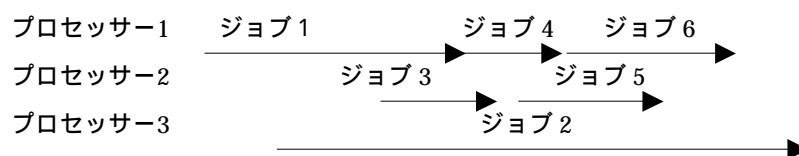
- 要素並列実行の例

要素並列化プログラムを含んだジョブがノードを占有して実行します。



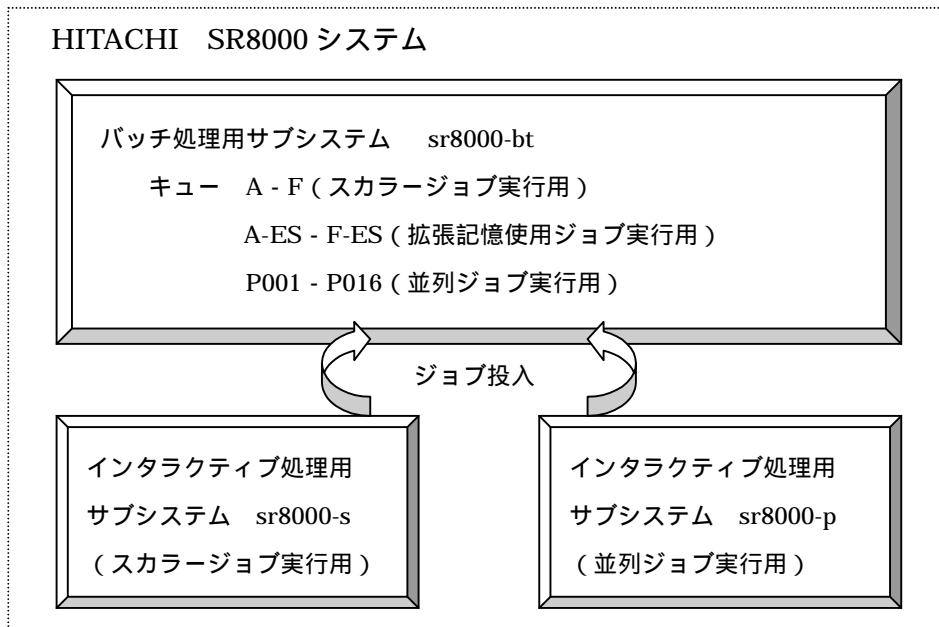
- スカラー多重実行の例

複数利用者のスカラージョブがノードを共有して実行します。



(サブシステム分割)

SR8000 の機能を活かした多様なジョブの実行形態に対応するために、本センターでは 128 台のノードを以下に示すようなサブシステムに分割して運用しています。



● インタラクティブ処理用サブシステム

インタラクティブ処理用サブシステムはログインしてコマンド実行するときに使用します。 以下に示す 2 つのサブシステムを用意しています。

- スカラー処理サブシステム (sr8000-s)
主にスカラージョブ実行、プログラムの編集、コンパイル、バッチジョブの投入をインタラクティブに行ないます。
- 並列処理サブシステム (sr8000-p)
主に要素並列ジョブ実行をインタラクティブに行ないます。また 2 ノードまでのノード並列実行を可能とします。

● バッチ処理用サブシステム (sr8000-bt)

バッチ処理用サブシステムはバッチジョブ実行専用のサブシステムです。 このサブシステムにログインすることはできません。ジョブのタイプに合わせて各種ジョブクラス(キュー)を用意しており、ジョブはインタラクティブ処理用サブシステムから NQS (Network Queuing System) により投入します。

2. 利用登録

SR8000 利用登録は大型計算機システム利用登録後に別途「newuser」オンライン手続きにて行なう必要があります。sr8000-s.cc.u-tokyo.ac.jp にログイン名「newuser」でログインしてその後の指示に従って下さい。

- (1) 「newuser」手続きは以下の手順で行います。VOS3 のアカウントとパスワードが必要になります。

```
% telnet sr8000-s.cc.u-tokyo.ac.jp
```

```
HI-UX/MPP (sr8000-s.cc.u-tokyo.ac.jp) (pts/5)
```

```
login: newuser
```

System Registration Command

```
Enter Your (VOS3) Login name : a30000 (利用者番号を小文字で)
```

```
Enter Your (VOS3) Password : _____ (VOS3 パスワードを小文字で)
```

VOS3 パスワードが不明の場合には共同利用掛 (03-5841-2717) まで御連絡下さい。

- (2) ホスト名の一覧が表示されますので sr8000-s.cc.u-tokyo.ac.jp を選択します。

- 1 * m-unix.cc.u-tokyo.ac.jp
- 2 sr8000-s.cc.u-tokyo.ac.jp
- 3 frontend.cc.u-tokyo.ac.jp
- 4 x-controller1.cc.u-tokyo.ac.jp
- 5 io-controller.cc.u-tokyo.ac.jp
- 6 * scalar-mpp.cc.u-tokyo.ac.jp

メニュー項目には sr8000-s.cc.u-tokyo.ac.jp のみ表示されますが、これを選択すると sr8000-p.cc.u-tokyo.ac.jp、sr8000-bt.cc.u-tokyo.ac.jp (バッチシステム) にも同時に登録されます。(* 印は既に登録されているホストです。)

- (3) 利用登録、取り消し、課金番号 (支払コード) 変更、ファイル使用量増減の各手続きはメニューから項目を選択します。

1. User Account Registration
2. User Account Deletion
3. Account Code Change
4. Increase File Size Limit
5. Decrease File Size Limit

本 newuser 手続きで取り消し、課金番号変更、ファイル使用量増減が可能です。(利用登録以外を選択した場合、sr8000-s.cc.u-tokyo.ac.jp のパスワードの入力が必要となります。)

- (4) 利用登録時には /home ファイル使用量の上限値の設定が必要です。
メッセージに従って間違いの無いように設定して下さい。

Specify your /home size(MB)[default 50MB, max 4000MB]:

ファイル上限値と課金額が表示されますので確認して下さい。確認メッセージに回答した時点でファイル課金が発生しますので慎重に入力して下さい。なお、/short ファイル使用量はこの手続きとは別に実際に作成したファイル量から課金します。

なお、本システムへの登録関連の反映の契機は以下の様になっています。

- 登録手続きが完了すると即時登録(5分程かかることがあります)されます。
- 登録の抹消は取り消し手続きを行った月の月末となります。
- ファイル使用量上限値の増加は即時変更、削減は翌月から変更となります。
- 課金番号(支払コード)の変更は即時反映されます。

「newuser」手続きを行った場合の注意事項

- 利用者自身で取り消し手続きを行なわない限り本システムの登録は翌月に継続されます。本システムの取り消しを希望する場合には、必ず取り消し手続きを行って下さい。
- 大型計算機システム利用登録を継続している場合には年度を越える場合でも本システムは継続扱いとなります。
- 利用負担金見込み額を超えた場合でも本システムの登録は継続されます。従って、利用負担金見込み額を超えていても継続で必要とする利用負担金は課金されます。
- 大型計算機システム利用登録上の変更(課金番号の削除、基本負担金額の変更等)があった場合には本システム利用の課金番号(支払コード)課金額が自動的に変更となることがあります。

以下の手続きは「newuser」ではできません。「大型計算機システム届出書」により手続きして下さい。

- センター利用（大型計算機システム）の取り消し
- 課金番号（支払コード）の削除
- 基本負担金額（月額 1,000 円/2,000 円）の変更
- VOS3 システムのファイル使用量上限値の変更
- 利用負担金見込み額の変更

3. ログインとバッチジョブ

(1) インタラクティブ処理用サブシステム

プログラムの編集やジョブの投入には「スカラー処理サブシステム」、要素並列プログラムの実行テストやデバッグには「並列処理サブシステム」といった様に 2 つのサブシステムを使い分けます。

- スカラー処理サブシステム

ホスト名 sr8000-s.cc.u-tokyo.ac.jp

- コマンドの実行、スカラープログラムの実行ができます。
(要素並列プログラムの実行はできません。)
- プログラムの編集、ジョブの投入、実行結果の参照等に使用します。
- ノードは他の利用者と共有します。
- メモリーの使用は 512MB までに制限されています。
- 使用したプロセッサ時間について課金します。

- 並列処理サブシステム

ホスト名 sr8000-p.cc.u-tokyo.ac.jp

- コマンドの実行、要素並列プログラムの実行ができます。
(2 ノードまでのノード間並列実行も可能です。)
- ログインは 1 ノードを占有します。
- メモリーは 7GB まで使用できます。
- ログインは 1 セッションしかできません。
- ノード占有時間の 8 倍について課金します。
(ログインのみの場合はログイン経過時間の 8 倍、別途ノードを確保した場合には、確保したノードの占有時間合計の 8 倍を加算)

(ログイン)

通常の UNIX システムと同様に telnet (または電話回線) でログインして利用できます。(2000年7月より SSH による接続をサポートしています。詳細は本誌別記事「UNIX システムへの SSH 接続について」を御覧ください。)

```
% telnet sr8000-s.cc.u-tokyo.ac.jp
```

```
Trying...
```

```
Connected to sr8000-s.cc.u-tokyo.ac.jp.
```

```
Escape character is '^['.
```

```
HI-UX/MPP (sr8000-s.cc.u-tokyo.ac.jp) (pts/18)
```

```
login:
```

ログインパスワードは変更すると sr8000-s と sr8000-p とで別々のパスワードになります。(「newuser」登録後の初期パスワードは同一です。) また、パスワードの入力を 5 回連続して間違えるとアカウントがロックされます。この場合には共同利用掛(03-5841-2717)までご連絡ください。

(2) バッチ処理用サブシステム

NQS (Network Queuing System) によりインタラクティブ処理用サブシステム (sr8000-s、sr8000-p) からジョブをサブミットすることにより利用します。(このサブシステムへのログインはできません。) ジョブの形態によって次の 3 つのジョブクラスを用意しています。

- スカラージョブクラス (A ~ F)
 - スカラープログラムの実行ができます。
(要素並列プログラムの実行はできません。)
 - ノードを他の利用者と共有します。
 - メモリーは 7GB まで使用できます。
 - 使用したプロセッサ時間について課金します。

- 拡張記憶 ES 使用ジョブクラス (A-ES ~ F-ES)
 - 要素並列プログラムを実行できます。
 - ジョブは 1 ノードを占有します。
 - メモリーは 3GB まで使用できます。
 - 拡張記憶ファイルは 16GB まで使用できます。
 - ノード占有時間の 8 倍について課金します。

● 並列実行ジョブクラス (P001 ~ P016)

- 要素並列プログラムを実行できます。
- ノード間並列ジョブ (最大 16 ノード) を実行できます。
- ジョブは 1~16 ノードを占有します。
- メモリーは各ノード 7GB まで使用できます。
- ノード占有時間×8 倍×占有ノード数について課金します。

(ジョブスクリプトファイル)

バッチジョブはインタラクティブ処理用サブシステム(sr8000-s、sr8000-p)から qsub コマンドでバッチシステム (sr8000-bt) に投入します。バッチジョブ用のスクリプトファイルは #@\$ ではじまる計算機資源設定オプションとシェルスクリプトの記述からなります。ジョブクラス (バッチキュー) はスクリプトファイル内に以下の記述をすることで自動的に選択されます。

```
#@$-q    キュー名 (パイプキュー名)
#@$-N    ノード数
#@$-IM   メモリーサイズ制限値
#@$-IV   拡張記憶 (ES) サイズ制限値
#@$-IT   経過時間制限値 (センター設定により -IE と同意)
```

バッチキューはこれらのオプションの指定で決定します。スカラージョブの場合は -q single (無指定時は single) を指定、-IT でバッチキューが選択されます。拡張記憶 (ES) 使用ジョブは -q single、-IT の指定の他に -IV が必要です。また、並列ジョブの場合は -q parallel の指定の他に -N を指定します。以下の記述を参考にして下さい。

記述例

スカラージョブの例

```
#@$-q single
#@$-IM 2GB
#@$-IT 00:30:00
```

要素並列ジョブの例

```
#@$-q parallel
#@$-N 1
#@$-IT 01:00:00
#@$-IM 4GB
```

拡張記憶 ES 使用ジョブの例

```
#@$-q single
#@$-IM 2GB
#@$-IV 10GB
#@$-IT 00:30:00
```

並列ジョブ (複数ノード) の例

```
#@$-q parallel
#@$-N 4
#@$-IT 01:00:00
#@$-IM 1GB
```


ジョブスクリプトファイル (job.sh) の例

```
#@$-q parallel
#@$-N 2
#@$-IT 01:00:00
cd sample
prun -n 2 program
```

(計画停止時刻)

保守作業、停電等により予めシステムの停止時刻がわかっている場合にはバッチシステムに計画停止時刻を設定します。この計画停止時刻が設定されているとき、この時刻を超える経過時間制限値 (#@\$-IT) を記述しているジョブは実行されませんのでご注意ください。(経過時間制限値の記述がない場合には各キューで設定している最大の経過時間制限値が仮定されます。) なお、計画停止時刻は qstat コマンドで確認できます。

(ジョブ関連コマンド)

qsub	ジョブのサブミット (投入)
% qsub job.sh Request 6128.sr8000-s.cc.u-tokyo.ac.jp submitted to queue: parallel	

qstat	ジョブの状態表示																																																																																										
% qstat -h sr8000-bt 2000/06/15 (Thu) 15:51:56: BATCHPIPE REQUESTS on sr8000-bt.cc.u-tokyo.ac.jp NQS schedule stop time : 2000/06/30 (Fri) 08:00:00 (Remain: 352h 13m 52s)																																																																																											
	<table border="1"> <thead> <tr> <th>REQUEST</th> <th>NAME</th> <th>OWNER</th> <th>QUEUE</th> <th>PRI</th> <th>NICE</th> <th>CPU</th> <th>MEM</th> <th>STATE</th> </tr> </thead> <tbody> <tr> <td>6128.sr8000-s</td> <td>STDIN</td> <td>a30000</td> <td>A</td> <td>0</td> <td>20</td> <td>960</td> <td>7168</td> <td>RUNNING</td> </tr> </tbody> </table>	REQUEST	NAME	OWNER	QUEUE	PRI	NICE	CPU	MEM	STATE	6128.sr8000-s	STDIN	a30000	A	0	20	960	7168	RUNNING																																																																								
REQUEST	NAME	OWNER	QUEUE	PRI	NICE	CPU	MEM	STATE																																																																																			
6128.sr8000-s	STDIN	a30000	A	0	20	960	7168	RUNNING																																																																																			
% qstat -b -h sr8000-bt 2000/06/15 (Thu) 15:46:08: BATCH QUEUES on sr8000-bt.cc.u-tokyo.ac.jp NQS schedule stop time : 2000/06/30 (Fri) 08:00:00 (Remain: 352h 8m 4s)																																																																																											
	<table border="1"> <thead> <tr> <th>QUEUE</th> <th>NAME</th> <th>STATUS</th> <th>TOTAL</th> <th>RUNNING</th> <th>RUNLIMIT</th> <th>QUEUED</th> <th>HELD</th> <th>IN-TRANSIT</th> </tr> </thead> <tbody> <tr><td>A</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>B</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>C</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>D</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>E</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>F</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>3</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>A-ES</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>B-ES</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>C-ES</td><td></td><td>AVAILBL</td><td>0</td><td>0</td><td>4</td><td>0</td><td>0</td><td>0</td></tr> </tbody> </table>	QUEUE	NAME	STATUS	TOTAL	RUNNING	RUNLIMIT	QUEUED	HELD	IN-TRANSIT	A		AVAILBL	0	0	4	0	0	0	B		AVAILBL	0	0	4	0	0	0	C		AVAILBL	0	0	4	0	0	0	D		AVAILBL	0	0	4	0	0	0	E		AVAILBL	0	0	3	0	0	0	F		AVAILBL	0	0	3	0	0	0	A-ES		AVAILBL	0	0	4	0	0	0	B-ES		AVAILBL	0	0	4	0	0	0	C-ES		AVAILBL	0	0	4	0	0	0
QUEUE	NAME	STATUS	TOTAL	RUNNING	RUNLIMIT	QUEUED	HELD	IN-TRANSIT																																																																																			
A		AVAILBL	0	0	4	0	0	0																																																																																			
B		AVAILBL	0	0	4	0	0	0																																																																																			
C		AVAILBL	0	0	4	0	0	0																																																																																			
D		AVAILBL	0	0	4	0	0	0																																																																																			
E		AVAILBL	0	0	3	0	0	0																																																																																			
F		AVAILBL	0	0	3	0	0	0																																																																																			
A-ES		AVAILBL	0	0	4	0	0	0																																																																																			
B-ES		AVAILBL	0	0	4	0	0	0																																																																																			
C-ES		AVAILBL	0	0	4	0	0	0																																																																																			

qdel	ジョブのキャンセル
% qdel -k 6128.sr8000-s Request 33319.sr8000-s is running, and has been signalled.	

(ジョブクラス制限値)

SR8000 のジョブクラス制限値

(2000年7月1日現在)

キュー名	制限時間		メモリーの大きさ (MB)		
	CTIME	ETIME	仮想メモリー	区分 ES	ノード数
single					
A	16 分	10 分	512(7168)	0(0)	1
B	80 "	50 "	"	"	"
C	480 "	300 "	"	"	"
D	1440 "	900 "	"	"	"
E	4800 "	3000 "	"	"	"
F	21600 "	13500 "	"	"	"
A-ES	80 分	10 分	512(3072)	0(16384)	1
B-ES	400 "	50 "	"	"	"
C-ES	2400 "	300 "	"	"	"
D-ES	7200 "	900 "	"	"	"
E-ES	24000 "	3000 "	"	"	"
F-ES	108000 "	135000 "	"	"	"
parallel					
P001	無制限	64 時間	7168(7168)	0(0)	1
P002	"	32 "	"	"	2
P004	"	16 "	"	"	4
P008	"	8 "	"	"	8
P016	"	4 "	"	"	16
TSS					
sr8000-s	無制限	18 時間	128 (512)	-	-
sr8000-p	"	"	" (7168)	-	1 (2)

- キュー名の single, parallel はパイプキューの名称です。
- バッチキューは各制限値の指定によりの自動的に選択されます。
- メモリーの大きさは標準値 (括弧内は最大値) です。
- メモリーの大きさは 1 ノード当たりの容量です。
- TSS のメモリー制限値は limit コマンドで変更できます。
- TSS の最大連続接続時間は通常で 18 時間、無入力の場合は 2 時間です。

ジョブクラス制限値は運用状況に応じて制限値の変更を行なう場合があります。また、バッチシステムにシステムダウン等の障害が発生した場合には、実行中のジョブは異常終了となりますが、システム起動後、自動的にキューへ再投入するよう設定しています。

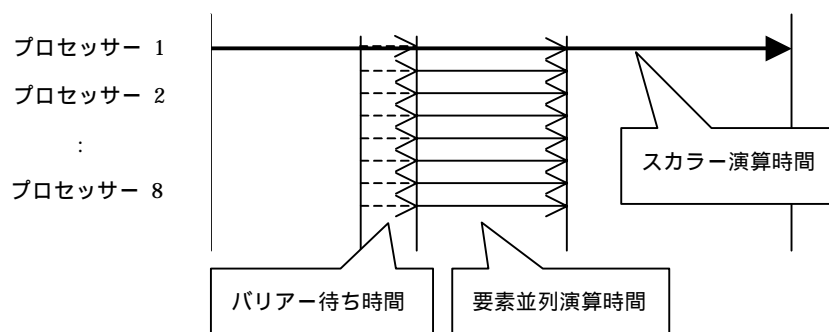
(ジョブ終了時メール)

ジョブの実行開始時および終了時にはメールが送信されます。特に終了時に送信されるメールにはそのジョブの実行に関する以下のような統計情報が出力されます。

submit user	ジョブをサブミットしたユーザー名
account number	ジョブの課金番号
request id	リクエスト ID (ジョブ ID)
submit host	ジョブをサブミットしたホスト名
execute host	サブミットしたジョブを実行したホスト名
submit queue	ジョブをサブミットしたキュー名
submit time	ジョブのサブミット時間
request start time	ジョブ開始時刻
request finish time	ジョブ終了時刻
request end status	ジョブの終了状態
request exit code	ジョブの返却コード
request cpu time (user + system)	ユーザー cpu 時間とシステム cpu 時間の合計
request exist time	ジョブ経過時間
used ES size	拡張記憶 (ES) の最大使用量
dynamic allocated memory size	実メモリの平均使用量
number of forked process	フォークしたプロセス数
number of nodes	確保したノード数
number of threads	生成したスレッド数
number of EP processes	要素並列プログラムを実行したプロセス数
total computing time of EP processes	要素並列プログラムの総演算時間
scalar computing time of EP processes	要素並列プログラムのスカラー演算時間
parallel computing time of EP processes	要素並列プログラムの要素並列演算時間
barrier waiting time of EP processes	要素並列プログラムのバリアー待ち時間
number of total instructions	全命令実行回数
number of floating point operations	浮動小数点演算実行回数
floating point operations per second	浮動小数点演算実行回数 / ユーザー cpu 時間

要素並列プログラムの各時間には以下の関係があります。

$$\text{要素並列プログラムの総演算時間} = (\text{スカラー演算時間} - \text{要素並列演算時間}) + \text{要素並列演算時間} \times 8$$



4. ファイルシステム

利用者が使用できるファイルシステムとして /home、 /short、 /para-io が用意されています。

(2000年7月1日現在)

ファイルシステム名	ファイル上限値	保存期間	ディスク容量
/home	利用者の宣言値	-	128GB
/short/sr8000-s	制限なし	15 日間	192GB
/short/sr8000-p	"	"	"
/short/sr8000-bt	"	"	"
/para-io	"	5 日間	160GB

- /home、 /short は sr8000-s、 sr8000-p、 バッチシステム (sr8000-bt) 間で互いにネットワーク (NFS) 経由でマウントしており、どのサブシステムからでも読み書きできます。
- /home はバッチシステム、 /short/{ ホスト名 } は各ホスト (サブシステム) にローカル接続しています。入出力性能上、ネットワーク (NFS) 経由を避けたローカル利用 (例えば /short/sr8000-s は sr8000-s からの使用) がお勧めです。
- /para-io は バッチジョブからのみ読み書きすることが出来ます。(NFS マウントしておりませんのでバッチジョブ以外からの入出力は出来ません。)
- 万一の事故に備えて大切なファイルは各自でバックアップをとっておかれませうお願い致します。

5. プログラムの作成と実行

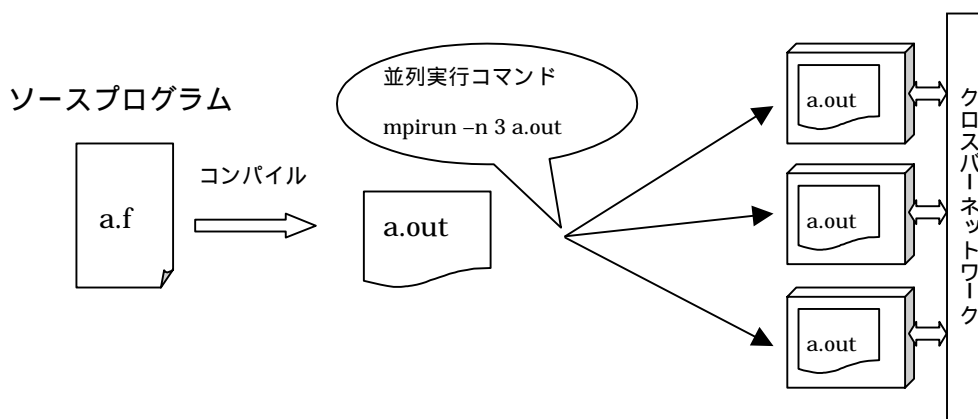
SR8000 には十分なハードウェア性能を引き出すための各種アプリケーションソフトウェアが提供されています。

- 並列化支援ソフトウェア
 - MPI
 - PVM
 - リモート DMA 転送
 - PARALLELWARE(Express)
 - 並列化支援コマンド
- 言語コンパイラ
 - 最適化 FORTRAN77
 - 最適化 FORTRAN90
 - 最適化 C
 - 最適化 C++
 - Parallel FORTRAN (HPF)

- 数値計算ライブラリー
 - MATRIX/MPP
 - MATRIX/MPP/SSS
 - MSL2
 - BLAS、LAPACK、ScaLAPACK

並列化プログラムの作成、実行の手順には以下のような例があります。

- ノード間並列処理
MPI、PVM、リモート DMA 関数等を用いてクロスバーネットワーク経由でノード間通信を行ないます。



(コンパイル)

f77	最適化 FORTRAN 7 7	
% f77 -parallel	a.f	(a.fを要素並列化してコンパイルする)
% f77 -parallel	a.f -lmatmpp	(a.fを要素並列化してコンパイルする.MATRIX/MPP 使用)
% f77 -i,U -W0,'vos3(nosymnchk)'	a.f	(a.fをコンパイルする、8文字以上の変数、関数名使用)

オプションの詳細は「最適化 FORTRAN77 使用の手引」参照

f90	最適化 FORTRAN 9 0	
% f90 -parallel	a.f	(a.fを要素並列化してコンパイルする)

オプションの詳細は「最適化 FORTRAN90 使用の手引」参照

mpif77	(MPI) 最適化 FORTRAN 7 7	
% mpif77	a.f	(a.fをコンパイルして実行ファイル a.out を作成)
% mpif77 -parallel	a.f	(a.fを要素並列化してコンパイルする)

オプションの詳細は「MPI・PVM 使用の手引」参照 (コマンドパスは /usr/mpi/bin)

mpif90	(MPI) 最適化 FORTRAN 9 0	
% mpif90	a.f	(a.fをコンパイルして実行ファイル a.out を作成)

オプションの詳細は「MPI・PVM 使用の手引」参照 (コマンドパスは /usr/mpi/bin)

その他、コンパイラーには最適化 C、PARALLELWARE、Parallel FORTRAN (トランスレーター) 等があります。各アプリケーションの利用方法はマニュアルを参照して下さい。

(並列実行コマンド)

プログラムを並列実行するためにはプログラムを複数のノードに割り当て、それぞれ実行させる必要があります。このため SR8000 では各アプリケーションに並列実行コマンドが用意されています。(ホスト及びノードプログラムをそれぞれ作成するホスト-ノードモデルでプログラムする場合には、ホストプログラムが起動コマンドとなりますので特定の並列実行コマンドを使用しないこともあります。)

prun	並列実行コマンド
% prun -n 4 a.out	(a.out を 4 ノードで実行)
% prun -f defilne.file	(定義ファイルを使用する場合)

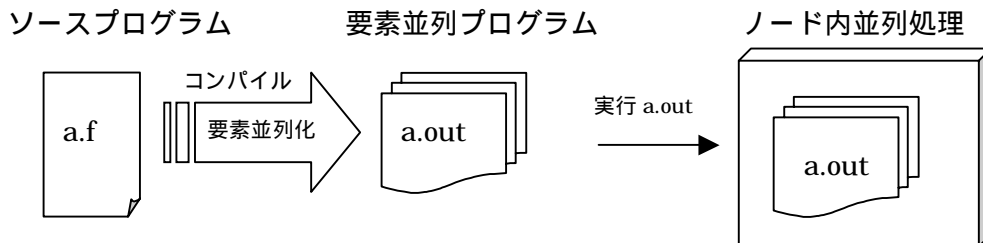
オプション、定義ファイルの詳細はオンラインマニュアル % man prun 参照

mpirun	MPI 並列実行コマンド
% mpirun -n 4 a.out	(a.out を 4 ノード、 4 プロセスで実行)
% mpirun -n 4 -np 8 a.out	(a.out を 4 ノード、 8 プロセスで実行)

オプションの詳細は「MPI・PVM 使用の手引」参照 (コマンドパスは /usr/mpi/bin)

- ノード内並列処理

要素並列化されたプログラムはノード内の複数プロセッサで並列処理することができます。(単体ノードの演算性能をフルに発揮するためには、要素並列処理をする必要があります。) また MPI でもノード内並列処理が可能です。



(要素並列化)

SR8000の特長である要素並列処理を行うためにはプログラムを要素並列化する必要があります。コンパイル時にオプションを指定するか、ソースプログラム中に指示文を記述することによって要素並列化をコンパイラに指示します。例えば以下のように要素並列化オプション (-parallel) を指定してコンパイルします。

f77	最適化 FORTRAN 7 7
% f77 -parallel a.f	(a.f を要素並列化してコンパイルする)
% f77 -parallel=4 a.f	(要素並列化のレベルを 4 にする)

オプションの詳細は「最適化 FORTRAN77 使用の手引」参照

f90	最適化 FORTRAN 9 0
% f90 -parallel a.f	(a.f を要素並列化してコンパイルする)

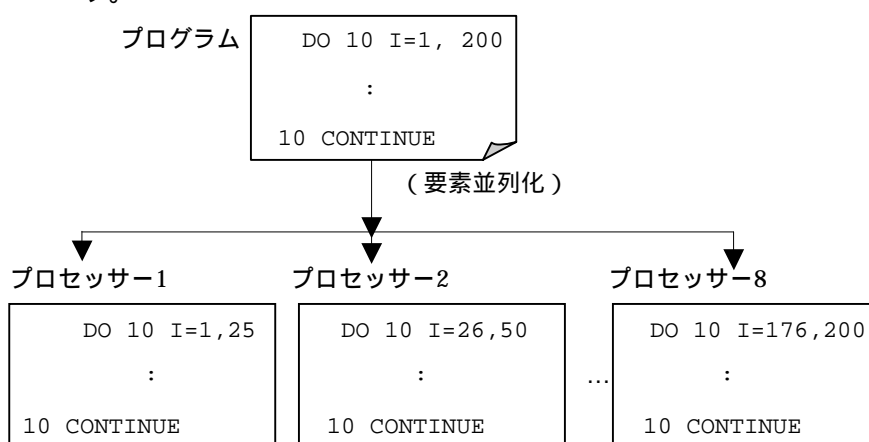
オプションの詳細は「最適化 FORTRAN90 使用の手引」参照

要素並列化プログラムの実行方法は通常の実行ファイル (a.out) と同様ですが、ノードを占有して動作しますので、sr8000-p か、バッチジョブとしてジョブクラス P001~P016、A-ES~F-ES に投入して実行する必要があります。(ジョブクラス A~F や sr8000-s で実行するとエラーとなります。)

要素並列化には次の二通りがあります。

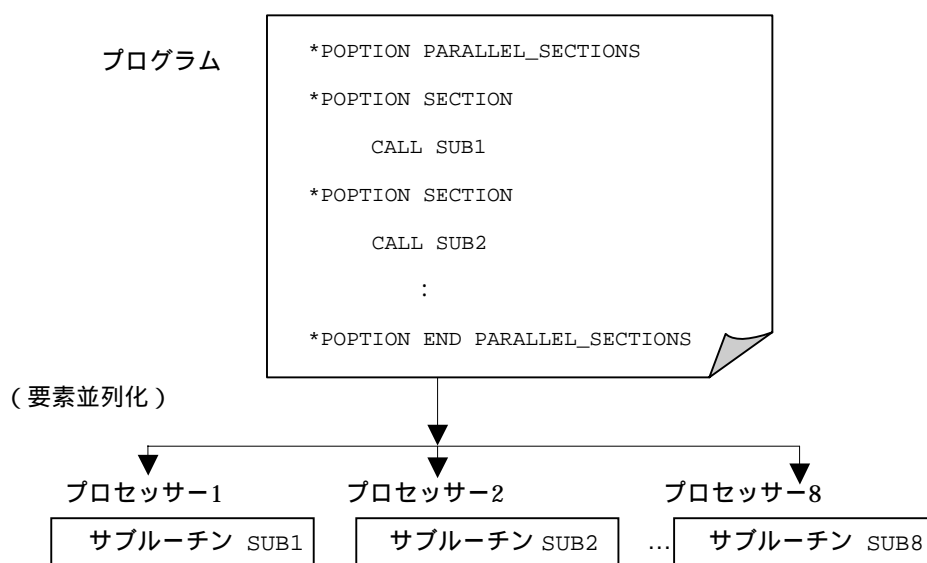
- DO 型要素並列化

DO ループをスレッドに分割し、複数プロセッサで並列に実行します。コンパイルオプションの指定によりコンパイラが判断して自動的に変換します。



- SECTION 型要素並列化

手続き呼出しなどの文の集まりをスレッドに分割し、複数プロセッサで並列に実行します。ソースプログラム中に指示文*POPTION を記述することで利用者が指示して変換します。(指示文の詳細はマニュアルを御覧ください。)



6. 利用負担金

● CPU 課金

利用者が使用する CPU 時間 / 月（または CPU 時間 / 月に相当する値）について以下の単価を適用します。

月額 1,000 円定額	月額 2,000 円定額	単 価
~ 150 時間 / 月	~ 450 時間 / 月	0 円 / 秒
150 時間超 ~ 6,000 時間 / 月	450 時間超 ~ 5,000 時間 / 月	0.02 円 / 秒
6,000 時間超 / 月 ~	5,000 時間超 / 月 ~	0 円 / 秒

計算方法（1,000 円定額コースで 500 時間使用した場合）

$$150 \text{ 時間} \times 3600 \times 0 \text{ 円} + (500 \text{ 時間} - 150 \text{ 時間}) \times 3600 \times 0.02 \text{ 円} = 25,200 \text{ 円}$$

（CPU 時間の課金方法）

各ノードにおいてジョブは以下に示す「共有」と「占有」のいずれかのタイプで実行されます。ログインセッションやバッチジョブでどちらを使用したかによって課金方法が異なります。（例えば要素並列ジョブの実行はノードを占有します。またスカラージョブクラスでの実行のときにはノードは複数のスカラージョブで共有します。）

- ノード共有の場合

複数のスカラージョブで 1 ノードを共有します。他のジョブも同じノードで動作しますので、課金は実際に使用した CPU 時間に対して行ないます。（ログインしているノードには他の利用者も同時にログインすることがあります。）

ジョブクラス A ~ F でバッチジョブを実行する場合と sr8000-s にログインしたときが対象です。

例：sr8000-s に 1 時間ログインしたとき、プログラムやコマンドの実行による実際使用した CPU 時間が 30 秒だった場合、課金は、

$$30 \text{ 秒} \times 0.02 \text{ 円} = 0.6 \text{ 円 (切り上げで 1 円)}$$

- ノード占有の場合

1 利用者がノードを占有します。ジョブが動作している間、他の利用者はそのノードを使用することができません。このため、課金はノード当たりの占有時間（経過時間）の 8 倍を CPU 時間とみなして単価を適用します。（ログインしているノードに他の利用者は同時にログインすることができません。）

ジョブクラス P001 ~ P016 、 A-ES ~ F-ES でバッチジョブを実行す

る場合と sr8000-p にログインしたときが対象です。

なお、要素並列ジョブはノードを占有して動作します。

例： sr8000-p に1時間ログインした場合には、プログラムやコマンドの実行で使用した CPU 時間の多少に係わらず

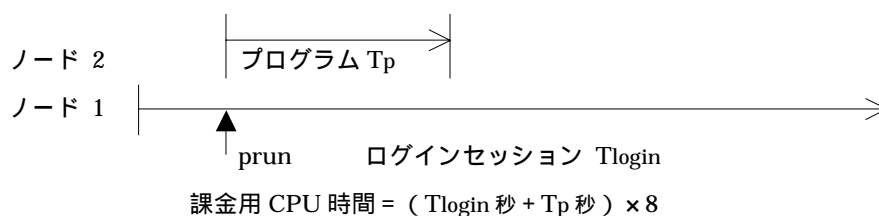
$$3600 \text{ 秒} \times 8 \times 0.02 \text{ 円} = 576 \text{ 円}$$

(並列ジョブの課金方法)

複数ノードの並列ジョブを実行する場合は必ずノード占有となります。

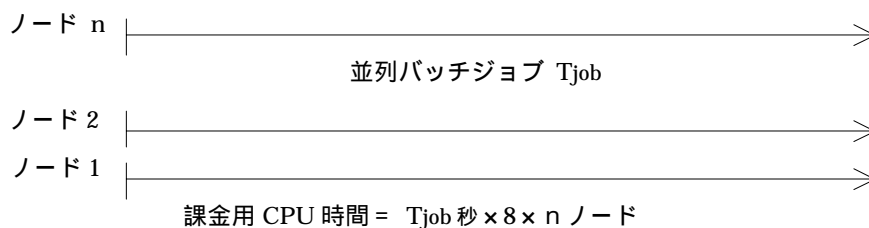
- インタラクティブジョブ (sr8000-p)

ログインノードについてはセッション時間 (= ノードの占有時間) の 8 倍を CPU 時間とみなして課金します。ここから新たなノードを確保し、プログラムを実行 (prun, mpirun 等の並列実行コマンド) した場合は確保したノードの占有時間合計の 8 倍を CPU 時間とみなして別途追加金を課金します。



- バッチジョブ

バッチジョブの実行中は利用者が指定した数のノードを確保しますのでジョブの実行時間 (= ノード当たりの占有時間) の 8 倍をさらにノード数倍したものを CPU 時間とみなして課金します。



● ファイル課金

- 長期保存ファイル /home

利用者が指定したファイル量の上限値まで使用しているものとして以下の単価を適用します。なお、上限値変更の手続きは「newuser」により行います。

月額 1,000 円定額	月額 2,000 円定額	単 価
~ 50MB	~ 100MB	0 円 / (MB ・ 月)
50MB 超 ~ 1,250MB	100MB 超 ~ 1,000MB	20 円 / (MB ・ 月)
1,250MB 超 ~	1,000MB 超 ~	0 円 / (MB ・ 月)

計算方法 (1,000 円定額コースで 100MB を上限値とした場合)

$$50\text{MB} \times 0 \text{円} + (100\text{MB} - 50\text{MB}) \times 20 \text{円} = 1,000 \text{円} / \text{月}$$

- 短期保存ファイル /short

利用者が使用しているファイル量に対して以下の単価を適用します。

5 円 / (月 ・ MB)

(課金の契機)

ファイル課金のうち長期保存ファイル /home は毎月の月始めに 1 ヶ月分を課金します。ただし、newuser による利用登録または上限値増加の手続きをした場合はその時点で日割り分を課金します。短期保存ファイルは実際のファイル保存量から 1 日分を算出 (切り上げ) し、毎日課金します。

7. マニュアル

以下のマニュアルがメーカーから提供されていますのでご参照下さい。なお、これらのマニュアルはセンターユーザーフロア、プログラム相談室に置いてあります。マニュアル購入についてはアカデミービジネスサービス 03-3813-3881 または (株) 日和出版センター 03-5281-5054 までお問い合わせ下さい

マニュアル名称	マニュアル番号
HI-UX/MPP 概説	6A30-3-001
HI-UX/MPP ユーザーズガイド	6A30-3-010
HI-UX/MPP Command Reference	6A30-3-011(E)
HI-UX/MPP アプリケーションプログラマーズガイド	6A30-3-020
HI-UX/MPP リモートDMA転送使用の手引 -C-	6A30-3-021
HI-UX/MPP 並列アプリケーションプログラマーズガイド -C-	6A30-3-022
HI-UX/MPP Network Applications Programmer's Guide	6A30-3-023(E)
HI-UX/MPP Programmer's Reference	6A30-3-024(E)
HI-UX/MPP 並列アプリケーションプログラマーズガイド -FORTRAN-	6A30-3-025
HI-UX/MPP MPI,PVM 使用の手引	6A30-3-026
HI-UX/MPP OSCNQS NQS ユーザーズガイド	6A30-3-230
HI-UX/MPP 最適化C言語	6A30-3-301
HI-UX/MPP 最適化C使用の手引	6A30-3-302
HI-UX/MPP 最適化C++使用の手引	6A30-3-303
HI-UX/MPP 最適化 FORTRAN90 言語	6A30-3-310
HI-UX/MPP 最適化 FORTRAN90 使用の手引	6A30-3-311
HI-UX/MPP リモートDMA転送使用の手引 -FORTRAN-	6A30-3-312

HI-UX/MPP 最適化 FORTRAN77 言語	6A30-3-313
HI-UX/MPP リモートDMA転送使用の手引 -FORTRAN77-	6A30-3-315
HI-UX/MPP Parallel FORTRAN 言語	6A30-3-320
HI-UX/MPP Parallel FORTRAN 使用の手引	6A30-3-321
HI-UX/MPP PARALLELWARE ユーザーズガイド -FORTRAN-	6A30-3-400
HI-UX/MPP PARALLELWARE リファレンス -FORTRAN-	6A30-3-401
HI-UX/MPP PARALLELWARE ユーザーズガイド -C-	6A30-3-402
HI-UX/MPP PARALLELWARE リファレンス -C-	6A30-3-403
HI-UX/MPP 並列デバッグ使用の手引	6A30-3-404
HI-UX/MPP アプリケーションプログラム開発環境 使用の手引	6A30-7-400
HI-UX/MPP 行列計算副プログラムライブラリ MATRIX/MPP	6A30-7-600
HI-UX/MPP 行列計算副プログラムライブラリ スカイライン法 MATRIX/MPP/SSS	6A30-7-601
HI-UX/MPP 数値計算副プログラムライブラリ MSL2 機能編 第1 分冊 行列計算	6A30-7-610
HI-UX/MPP 数値計算副プログラムライブラリ MSL2 機能編 第 2分冊 関数計算	6A30-7-611
HI-UX/MPP 数値計算副プログラムライブラリ MSL2 機能編 第 3分冊 統計計算	6A30-7-612
HI-UX/MPP 数値計算副プログラムライブラリ MSL2 操作編	6A30-7-613
HI-UX/MPP 統合構造解析システム ISAS2 機能編 第1分冊 機能	6A30-7-650
HI-UX/MPP 統合構造解析システム ISAS2 機能編 第2分冊 入力 データ	6A30-7-651
HI-UX/MPP 統合構造解析システム ISAS2 操作編	6A30-7-652
HI-UX/MPP 統合構造解析システム ISAS2 例題編	6A30-7-653

(英文マニュアル)

HI-UX/MPP General Information	6A30-3-001(E)
HI-UX/MPP User's Guide	6A30-3-010(E)
HI-UX/MPP Command Reference	6A30-3-011(E)
HI-UX/MPP Application Programmer's Guide	6A30-3-020(E)
HI-UX/MPP Remote DMA -C- User's Guide	6A30-3-021(E)
HI-UX/MPP Parallel Application Programmer's Guide -C-	6A30-3-022(E)
HI-UX/MPP Network Applications Programmer's Guide	6A30-3-023(E)
HI-UX/MPP Programmer's Reference	6A30-3-024(E)

HI-UX/MPP Parallel Application Programmer's Guide -FORTRAN-	6A30-3-025(E)
HI-UX/MPP MPI/PVM User's Guide	6A30-3-026(E)
HI-UX/MPP OSCNQS User's Guide	6A30-3-230(E)
HI-UX/MPP Optimizing C Reference	6A30-3-301(E)
HI-UX/MPP Optimizing C User's Guide	6A30-3-302(E)
HI-UX/MPP Optimizing C++ User's Guide	6A30-3-303(E)
HI-UX/MPP Optimizing FORTRAN90 Reference	6A30-3-310(E)
HI-UX/MPP Optimizing FORTRAN90 User's Guide	6A30-3-311(E)
HI-UX/MPP Remote DMA -FORTRAN- User's Guide	6A30-3-312(E)
HI-UX/MPP Optimizing FORTRAN77 Reference	6A30-3-313(E)
HI-UX/MPP Optimizing FORTRAN77 User's Guide	6A30-3-314(E)
HI-UX/MPP Parallel FORTRAN Reference	6A30-3-320(E)
HI-UX/MPP Parallel FORTRAN User's Guide	6A30-3-321(E)
HI-UX/MPP PARALLELWARE User's Guide -FORTRAN-	6A30-3-400(E)
HI-UX/MPP PARALLELWARE Reference -FORTRAN-	6A30-3-401(E)
HI-UX/MPP PARALLELWARE Reference -C-	6A30-3-402(E)
HI-UX/MPP PARALLELWARE User's Guide -C-	6A30-3-403(E)
HI-UX/MPP Parallel Debugger User's Guide	6A30-3-404(E)
HI-UX/MPP Application Program Development Environment User's Guide	6A30-7-400(E)
HI-UX/MPP MATRIX/MPP User's Guide and Reference	6A30-7-600(E)
HI-UX/MPP MATRIX/MPP/SSS User's Guide and Reference	6A30-7-601(E)
HI-UX/MPP MSL2 Matrix Calculation	6A30-7-610(E)
HI-UX/MPP MSL2 Function Calculation	6A30-7-611(E)
HI-UX/MPP MSL2 Statistical Calculation	6A30-7-612(E)
HI-UX/MPP MSL2 User's Guide	6A30-7-613(E)

8. 参考資料

スーパーコンピューティングニュース

SR8000 チューニング支援機能を利用したプログラムチューニング	Vol.2 No.4(2000.7)
SR8000 性能モニター機能の仕様改善の報告	Vol.2 No.2(2000.3)
SR8000 性能モニターとログメッセージを用いたチューニング	Vol.2 No.1(2000.1)
SR8000 性能モニター機能の利用法	Vol.1 No.4(1999.11)

UNIX システム利用入門（初級編）	Vol.1 No.3(1999.9)
UNIX システム利用入門（中級編）	Vol.1 No.3(1999.9)
SR8000 の有効利用法	Vol.1 No.3(1999.9)
スーパーテクニカルサーバ HITACHI SR8000 LINPACK 性能のご紹介	Vol.1 No.2(1999.7)
システム変更等のお知らせ	Vol.1 No.1(1999.5)

センターニュース

ベクトル並列型スーパーコンピュータ-SR8000 運用開始のお知らせ	Vol.31 No.2(1999.3)
SR8000 システム上の FORTRAN 処理系と従来システム上の FORTRAN 処理系との互換性	Vol.30 No.6(1999.11)