

## 「FX10 スーパーコンピュータシステム試験運転期間中の運用について」記載内容の訂正

スーパーコンピューティングニュース Vol.14, No.2 (2013.3) でお知らせしておりました「FX10 スーパーコンピュータシステム試験運転期間中の運用について」ですが、記述内容に誤りがありましたので訂正します。

## p.12 表 5.コア数について記述誤り

(誤)

表 5. インタラクティブジョブサービス ジョブクラス制限値 (試験運転期間中)

キュー名	最大ノード数	制限 (経過) 時間	メモリ容量
interactive			
interactive_n1	1 (12 コア)	2 時間	28 GB
interactive_n8	8 (96 コア)	10 分	28 GB

(正)

表 5. インタラクティブジョブサービス ジョブクラス制限値 (試験運転期間中)

キュー名	最大ノード数	制限 (経過) 時間	メモリ容量
interactive			
interactive_n1	1 (16 コア)	2 時間	28 GB
interactive_n8	8 (128 コア)	10 分	28 GB

## p.13 表 6.コア数について記述誤り

(誤)

表 6. バッチジョブサービス ジョブクラス制限値 (試験運転期間中)

キュー名	最大ノード数	制限 (経過) 時間	メモリ容量
debug	1 ~ 240 (2,880 コア)	30 分	28 GB
short	1 ~ 72 (864 コア)	2 時間	28 GB
regular			
small	12 ~ 216 (2,592 コア)	12 時間	28 GB
medium	217 ~ 372 (4,464 コア)	12 時間	28 GB
large	373 ~ 480 (5,760 コア)	12 時間	28 GB
x-large	481 ~ 1,440 (17,280 コア)	6 時間	28 GB

(正)

表 6. バッチジョブサービス ジョブクラス制限値 (試験運転期間中)

キュー名	最大ノード数	制限 (経過) 時間	メモリ容量
debug	1 ~ 240 (3,840 コア)	30 分	28 GB
short	1 ~ 72 (1,152 コア)	2 時間	28 GB
regular			
small	12 ~ 216 (3,456 コア)	12 時間	28 GB
medium	217 ~ 372 (5,952 コア)	12 時間	28 GB
large	373 ~ 480 (7,680 コア)	12 時間	28 GB
x-large	481 ~ 1,440 (23,040 コア)	6 時間	28 GB

## 「FX10 スーパーコンピュータシステムの利用方法について」記載内容の訂正

スーパーコンピューティングニュース Vol.14 No.2(2012.3)で、お知らせしておりました「FX10 スーパーコンピュータシステムの利用方法について」ですが、試験運転開始時点で一部変更および訂正がありましたのでお知らせいたします。なお、試験運転期間中は、システムの設定変更等のため、予告なく運転の停止、運用仕様の変更を行う場合がありますので、予めご了承ください。

### p.20 表 5. Fortran コンパイルコマンド

誤字訂正

誤) スレッド並列化(OpenMP) `frptx -Kfast, openmp`  
正) スレッド並列化(OpenMP) `frtpx -Kfast, openmp`

### p.21 表 7. C コンパイルコマンド

誤字訂正

誤) スレッド並列化(OpenMP) `fcctx -Kfast, openmp`  
正) スレッド並列化(OpenMP) `fccpx -Kfast, openmp`

### p.22 図 2. インタラクティブの実行例 (2)

ノード数の修正 (最大 8 ノード)

変更前)

`[z30000@oakleaf-fx-1 ~]$ pjsub --interact -L "node=12"` インタラクティブジョブ (12 Node) を起動

変更後)

`[z30000@oakleaf-fx-1 ~]$ pjsub --interact -L "node=8"` インタラクティブジョブ (8Node) を起動

誤字訂正

誤) `[z30000@e10-087 ~]$ mpiexec ./a.out` (カンマ)  
正) `[z30000@e10-087 ~]$ mpiexec ./a.out` (ピリオド)

### p.23 1.2

誤字訂正

誤) 3.1 にある、主な利用形態ごとのバッチジョブスクリプト例を示します。  
正) 3.2 にある、主な利用形態ごとのバッチジョブスクリプト例を示します。

### p.23 図 4. バッチジョブの実行例 (2)(スレッド並列ジョブ)、 図 7. バッチジョブの実行例 (5)(ハイブリッド並列ジョブ)

環境変数の指定条件の変更

変更前)

`export OMP_NUM_THREADS=16` スレッド数の設定 (`OMP_NUM_THREADS`,  
`export PARALLEL=16` `PARALLEL` の指定は必須)

変更後)

`export OMP_NUM_THREADS=16` スレッド数の指定 (OpenMP のとき `OMP_NUM_THREADS`,  
`export PARALLEL=16` 自動並列化のとき `PARALLEL` の指定が必須)

p.25 図 9. バッチジョブの状態確認 (イメージ)

pjstat コマンドの表示内容の変更 (試験運転開始時点における変更後のイメージを掲載します。)

```
[z30000@oakleaf-fx-1 ~]$ pjsub a.sh                                     バッチジョブの投入
[z30000@oakleaf-fx-1 ~]$ pjstat                                       状態確認
Oakleaf-FX scheduled stop time: 2012/04/23 (Mon) 09:00:00 (Remain: 21days 14:13:06)
```

JOB_ID	JOB_NAME	STATUS	PROJECT	RSCGROUP	START_DATE	ELAPSE	TOKEN	NODE
1234	job.sh	RUN	pz0000	small	04/01 18:45:47	00:01:06	3.1	96
1235	job.sh	QUE	pz0000	small	(04/01 19:20:00)	00:00:00	6.0	12

```
[z30000@oakleaf-fx-1 ~]$
```

**【各項目の説明】**  
 pjstat コマンドで表示される内容の主な項目は以下の通りです。ヘッダ情報として、定期保守予定日を表示します。

JOB_ID	ジョブID
JOB_NAME	ジョブ名
STATUS	ジョブ状態 (QUE: 実行待ち、RUN: 実行中)
USER	利用者番号
PROJECT	プロジェクトコード
RSCGROUP	リソースグループ名 (キュー名)
START_DATE	ジョブ実行開始時刻。ジョブが実行前の場合には、ジョブ実行開始予測時刻 (時刻にカッコが付きま)
ELAPSE	経過時間 (ジョブが開始されてからの時間)
TOKEN	トークン使用量 (状態表示時の使用量)。ジョブが実行前の場合には、トークン予測使用量 (使用量にカッコが付きま)
NODE	ジョブ投入時の指定ノード数

グループ管理者は、pjstat コマンドに -A オプションを付与すると、グループ内利用者のジョブ情報を参照できます。なお、グループ内利用者は自ジョブ情報のみ参照可能です。

```
[z00000@oakleaf-fx-1 ~]$ pjstat -A
Oakleaf-FX scheduled stop time: 2012/04/23 (Mon) 09:00:00 (Remain: 21days 14:12:44)
```

JOB_ID	JOB_NAME	STATUS	USER	PROJECT	RSCGROUP	START_DATE	ELAPSE	TOKEN	NODE
92752	job.sh	RUN	z00001	gz00	small	04/01 18:45:47	00:01:29	4.2	96
92753	job.sh	RUN	z00002	gz00	small	04/01 18:45:48	00:01:28	4.1	96
92755	job.sh	RUN	z00002	gz00	small	04/01 18:46:01	00:01:15	0.3	12
92756	job.sh	QUE	z00001	gz00	small	(04/01 19:20:00)	00:00:00	6.0	12

```
[z00001@oakleaf-fx-1 ~]$
```