3. WG 活動で出された、課題・回答・コメントのまとめ

本章では、2 年間の WG 活動を通し、大規模 SMP（PRIMEPOWER HPC2500）に対して出された課題・要望とその対応等について整理した。
これら課題・要望には富士通からの回答（或いはコメント）だけでなく、他の機関の参考になるよう、各委員それぞれの評価・分析結果や回避策も記載した。
また、すべての情報には、それを挙げた委員の機関名と日付を記載し、５つのカテゴリー別に分類している（①CPU 時間、②資源管理、③使い勝手と性能、④I／O、⑤その他）。
なお、スペース上ここで説明しきれないものには添付資料をつけた。添付資料はカラーでないと分かりづらいものもあるため、SS 研サイトでの参照を推奨する。

SS 研サイト　http://www.ssken.gr.jp/

CPU 時間

(1) CPU 時間の見せ方

【指摘】

- 複数 CPU を使用する並列ジョブの課金を、最長の CPU 使用時間（複数 CPU の中で最も CPU 使用時間の長いもの）にすることにより、アプリケーションの並列化チューニングが促進できるのではないか。（各 CPU を満遍なく使用できれば得だと思う）
- またこの場合、ジョブの打ち切り時間も課金に合わせて、プロセス単位でなく、最長の CPU 使用時間で指定したい。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

- NQS ジョブだけでなく、会話型処理の場合も同様にしてほしい（今後、ラージページを使えるような会話型の重要性が増大すると予想）

〈京大学情：2005/10/07 第 4 回会合〉

【回答】

- Parallelnavi 2.4.1（2005 年 8 月出荷済）で提供済です。バッチジョブおよび、会話型ジョブの両方において、ジョブ打ち切り時間として CPU 単位、プロセス単位の何れかを選択可能としました。この選択は、ノード単位に設定可能ですが、全ノード共通とすることを推奨します。
  課金・統計情報については、Parallelnavi 初版から CPU 単位をサポート済です。

〈富士通：2006/2/17 第 5 回会合〉

(2) 資源競合によりジョブの実行終了時間が予測できない。

【指摘】

- 実行時間で制限する場合は致命的な問題だ。
  実際にどの程度のバラツキなのかデータを公表してほしい。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- 宇宙航空研究開発機構様、日本原子力研究開発機構様での実測・分析結果を添付資料に示します。

添付資料 2-1, 2-2 を参照

〈富士通：2006/2/17 第 5 回会合〉

資源管理

(3) CPU 数のみでの並列度指定

【指摘 1】

- 現在の Parallelnavi では、スレッド数とプロセス数の両方をエンドユーザが指定しなければならない（ハイブリッド並列前提）。また、アプリケーションの特性（スレッド並列かプロセス並列か、並列数は幾つかなど）に応じてバッチキューを複数用意することを推奨している。しかし、プログラミングモデルに依存せず、ジョブ全体の CPU 数を並列度(?)として指定することが望ましい。また、バッチキューも同様に一つのキューで各プログラムモデルを実行可能とすべきだ。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

- NQS ではプロセス数とスレッド数を別々にしか制限できない。
  運用側がやりたいのはトータル CPU 数での制限だ。OS 側で実現して欲しい。

〈京大学情：2005/6/29 第 3 回会合〉

【指摘 1 への回答】

- ハイブリッド並列を基本プログラムモデルと想定しました。
  → スレッド数とプロセス数の両方を指定することが自然と考えました。
  → ジョブ全体の CPU 数のみを指定したいという要件は理解できますが、スレッド並列・プロセス並列・ハイブリッド並列に共通した資源量指定の良い方式が見出せていません。
- なお、NQS-JM の標準値・最大値設定機能により、通常はプロセス数（スレッド数は 1）のみ指定しスレッド並列ジョブとハイブリッド並列ジョブはプロセス数とスレッド数の両方を指定してもらうなど、エンドユーザの利便性を向上させています。

- 現状では、確かに使いにくいと考えていますが、例えばハイブリッド並列ジョブを複数ノードに跨って実行する場合、各ノードから何 CPU(=何スレッド)分の CPU を割当てないといけないのかが、ジョブ全体の CPU 数のみの指定では分からないという問題等があり、スレッド数・プロセス数を指定する仕様となっています。良い方式が見出せていない状況であり、ご指導いただきたい。

〈富士通：2004/11/24 第 1 回会合〉

【指摘 1 への回答へのコメント】

- JAXA では NSJS によりトータル要求 CPU 数による投入制限を実現している。
- ジョブ全体の CPU 数の制限であれば、NQS 出口により実現可能と思われる。

〈JAXA：2004/11/24 第 1 回会合〉

【指摘 2】

- 東大のバルク制運用が簡単に実施できるか。(ユーザ毎に使用 CPU 数を定める)

〈京大学情：2005/10/07 第 4 回会合〉

【指摘 2 への回答】

- Parallelnavi は、JOB が実行し始めてから資源不足で待ちに入ることを避けるため、資源を事前予約した上で JOB を走行させる仕組みを採用しています。このため、ノード間に跨いだ CPU 割り当てが可能か否かを判断するために、プロスセ数とスレッド数を指定させています。
  ただし、1JOB が利用できる最大 CPU 数を新しく設けることで、最大プロセス数×最大スレッド数分の CPU 数を用意しないでよくなるが、有効でしょうか？
  有効ならば、NQS 出口でチェックする機構を提供します。
- 東大のバルク制運用では、以下の指定で簡単な指定を実現していると思います。
  ■プログラミングモデル：プロセス並列かスレッド並列（ハイブリッド並列含む）か
  ■使用する CPU 数の合計（ただし、スレッド並列は 8CPU 固定）
  NQS-JM の標準値設定機能によりスレッド数の標準値を 8 とし、通常はプロセス数のみを指定させることにより、東大センター殿のバルク運用と同様な運用が可能と考えます。

〈富士通：2006/2/17 第 5 回会合〉

(4) ジョブの資源量指定が複雑で分かりにくい

【指摘】

- CPU 数/メモリ容量はジョブ種別を意識しないで指定したい。
- 誤った指定ができてしまう。(32 スレッド×32 プロセス=1024CPU)

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

項番 3 に同じ。

なお、日本原子力研究開発機構様においては、qsub 実行時、-lp×-lP の値が指定 CPU 数未満であるかをチェックするシェルスクリプトを作成し、qsub コマンドを置き換えることで投入制限しております。

〈富士通：2006/2/17 第 5 回会合〉

(5) 1 ジョブをできる限り同じノードへ割り当てる

【指摘】

- ノード間並列ジョブを想定すると、跨るノード数が少ないほど、ノード異常によるジョブへの影響が確率的に少なくなる。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- Parallelnavi の NQS-JS 機能の「PACK 割り当て方式」により、ノードの跨りが少なくなるノード割り当てをサポート済です。詳細は、「添付資料 5-1 1 ジョブを同一ノードに割り当てる方法」を参照願います。

添付資料 5-1 を参照

〈富士通：2004/11/24 第 1 回会合〉

(6) CPU 利用台数によるジョブの優先度制御

【指摘】

- デバッグは少 CPU で通常のバッチジョブよりも優先度を高くして実施したい。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- 以下の二つの運用方法にて実現可能です。ご利用をご検討ください。
  (1) 少 CPU 使用ジョブ用に専用キュー(会話型キューや少 CPU キュー)を設け、当該キューの優先度を他のキューよりも高く設定する。
  (2) NQS-JS の既存機能である、資源の重み評価値機能を使用する。

  「添付資料 6-1 CPU 利用台数によるジョブの優先度制御」に、上記二方式の具体例を示しています。

添付資料 6-1 を参照

〈富士通：2006/2/17 第 5 回会合〉

(7) JOB スワップ時の異常 CPU の切離し

【指摘】

- JAEA では、ジョブスワップ機能をジョブ間の優先度制御の他に、実行中ジョブを異常 CPU（ジョブ実行は可能であるが性能低下するような異常）から、正常な CPU へ移動させる目的にも使えると考えていた。

しかし、スワップアウト中に異常 CPU を割当抑止（DISABLE）状態としても、スワップイン時に異常 CPU がそのまま割当てられ実行された。

〈JAEA：2004/11/24 第 1 回会合〉

【回答】

- Parallelnavi の割当抑止機能は、新規に走行を開始するジョブへの資源割当を抑止することを目的に開発していたため、ご指摘のように割当抑止設定した異常 CPU へのスワップインが行われました。しかし、改善が望ましいため、ハード故障全体への対応を踏まえた機能強化を検討させていただきます。
  なお、当面は、日本原子力研究開発機構様にて実施いただいている方法により、代替をお願いします。
  《代替手段》
  日本原子力研究開発機構様では、CPU 異常発生時、当該 CPU がジョブ用 CPU であれば disable に設定し、CPU のグループ定義(sg.conf ファイル)も書き換えることで、新規ジョブやスワップジョブを割り当て対象外に設定するシェルツールを作成して対応。

〈富士通：2006/2/17 第 5 回会合〉

(8) CPU･メモリの使用量を考慮した資源割り当て＝資源の効率的利用

(例：多 CPU 少メモリジョブと少 CPU 多メモリジョブを同一ノードへ割当)

【指摘】

- 計算資源を効率的に利用し、システム全体の稼動率を向上させるためのジョブスケジューリングが必要。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- PRIMEPOWER HPC2500 ではジョブへ割当てる資源種別と割当上の条件が複雑であり、最適資源探索処理のオーバヘッドが非常に大きくなります。このため、オーバヘッドの削減を最優先に考慮し、資源間のバランスを考慮した高度割当はサポートせず、比較的単純な資源探索ロジック（ノード→CPU→ノード内バリア→ノード間バリア→DTU の順に空き資源を探索し割当て）を採用しています。
- オーバヘッドの削減を優先するか、資源間のバランスの考慮を優先するかはセンターによって異なる筈なので、今後はカスタマイズ性に力をいれたいと考えています。

〈富士通：2004/11/24 第 1 回会合〉

(9) ジョブスケジューリング

【指摘】

- ジョブ投入の際、Parallelnavi ではプロセス単位に空き資源をチェックし割当てているが、京大のように電気代を節約したいサイトでは、逆に空きノードを作りたい。ユーザ側で、そのような定義ができると良い。

〈京大学情：2005/2/16 第 2 回会合〉

- JAEA はノード数が少ないので、大きなジョブが投入されることを想定し、通常は京大と同様に、可能な限り空きノードを作っておきたい。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- 「ノード集中割り当て機能」を 2006 年 12E 目標でご提供します。
  詳細な仕様は、「添付資料 9-1 ジョブスケジューリングの定義(空きノードを作るスケジューリング)」をご参照ください。

添付資料 9-1 を参照

〈富士通：2006/2/17 第 5 回会合〉

(10) ラージページとノーマルページの二重構造

【指摘】

- ラージページをエンドユーザが意識しなければならない。
  コンパイル時にラージページ指定忘れなどによりノーマルページを枯渇させるとシステムハングに陥る。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- ラージページの利用を促進するために、以下の機能をサポート済みです。
  - ・ センター毎にラージページをコンパイル時のデフォルトとする機能
  - ・ ノーマルページとラージページの制限を別々に定義できる機能

〈富士通：2006/2/17 第 5 回会合〉

(11) ジョブ実行に必要なメモリ量を如何に見積もるか

【指摘】

- エンドユーザがメモリ（ラージページ）使用量を指定しなければならないが、面倒なので大きく指定してしまう。このため、実際には使用していないメモリもリザーブしてしまうため、後続ジョブがメモリ資源不足で実行できず、システムの稼動率が低下する。正しいメモリ使用量を指定させるための機能の充実が必要。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- 自らプログラムする場合、ある程度想定可能であるが、スタック域など見積りが難しい、プログラムしないエンドユーザには見積もりができないなどの課題があります。
- 現在は、エンドユーザがプログラムを一度実行して使用量を把握し、次回からの実行で使用量を指定していただくよう願いたい。

〈富士通：2004/11/24 第 1 回会合〉

- なお、宇宙航空研究開発機構様では、ラージページメモリ使用量（text 域、data 域、bss 域）を計算する概算するコマンド「lsize」を作成し、ユーザ指定を簡便化しています。
  仕様については、第 2 回会合での運用ツールで説明されていますので、参照ください。

添付資料 11-1(P1 No1)を参照

〈富士通：2005/2/16 第 2 回会合〉

【回答へのコメント】

- エンドユーザのプログラムは多種多様なので、ジョブスケジュールの導入が必要不可欠と思う。

〈JAXA：2004/11/24 第 1 回会合〉

(12) 動的なメモリ解放機構の実現

【指摘(第一次)】

- 要求メモリと実際にジョブで使用されるメモリ量との乖離が激しい。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

- ユーザはアプリを開発途上であり、固定的な指定ができないため、大きく指定する傾向がある。動的なメモリ開放機構をぜひ実現して欲しい。期待する仕様は、凡そ以下の通りである。
  ○センタデフォルトとして以下の選択
  1)動的メモリ解放機構を ON/OFF するか選択
  2)ジョブ開始から開放するまでの Interval 時間
  ○ユーザ指定(qsub オプション)で変更可能
  ○マルチブロックジョブを意識したヘテロなメモリ資源開放？

〈京大学情：2005/6/29 第 3 回会合〉

- 誤って必要以上に確保された資源を解放して他ジョブで利用したい。

〈JAEA：2005/2/16 第 2 回会合〉

【回答(第一次)】

- 非ラージページ：ページング（後続ジョブ実行の阻害要因にはならない)、ラージページ：事前割当てし、ジョブ実行中の動的変更なし。
  指摘のようなジョブ動作途中でのメモリ開放を行った場合、メモリ開放以降に

発生したメモリ獲得要求を如何に処理すべきかという課題があります。ジョブアベンドでよいのか？他ジョブをスワップアウトして空きメモリを作るのか？（本当に稼働率は向上するのか？）
この機能を追加した場合、メモリ開放以降に発生したメモリ獲得要求を如何に処理すべきか、ジョブアベンドでよいのか、他ジョブをスワップアウトして空きメモリを作るのか。どのように考えればよいかご指導願いたい。

〈富士通：2004/11/24 第 1 回会合〉

【回答(第一次)へのコメント】

- オプションにより、エンドユーザに選択させてもよい。ただし、一定時間後に開放することを選択したエンドユーザがなんらかの形で有利となるような運用上の工夫が必要である。

〈JAXA：2004/11/24 第 1 回会合〉

- なお、JAXA では、ラージページを使用しないユーザが多い。それらのジョブはセンター管理者が強制終了し、ロードモジュールを正しく作成(コンパイル)するよう指導している。エンドユーザをこのような形で指導(教育)していくことも重要。

〈JAXA：2004/11/24 第 1 回会合〉

【回答(第二次)】

- ご要望に対する検討結果を回答します。現在、 以下のような機能仕様で実現を考えています。
  - 動的にメモリ解放機構を使用するか否かを qsub オプション/キューで設定
  - メモリ解放するタイミングをジョブ実行開始から自動解放までのインターバルで指定
  - インターバル経過後、未使用ラージページをセグメント単位（ディフォルト 64MB）に解放。以降のメモリ要求はエラー終了（エラー終了した場合、自動解放したジョブである旨、実行結果にレポート）

《確認事項》
本機能によりメモリ資源を有効活用する運用を実現するには、エンドユーザが自動解放を積極的に利用するための推進が必要と考えます。例えば、メモリ課金により、メモリチューニングを考慮したエンドユーザは、安価にシステムが利用できるような運用ルール等（デフォルトで動的メモリ解放を運用するならば別ですが)。
このような運用ルールの変更も踏まえ、本機能の必要性について再度のご検討をお願いいたします。

〈富士通：2006/2/17 第 5 回会合〉

《必要性の検討結果》
各センターのメモリ利用状況を確認した結果、本機能の緊急性は高くないとの結論を得た。ただし、センター運用上は有効な機能と考えられるので、今後のエンハンス項目の参考とする。

(13) ノード別に分散された空き資源を集約することができない

【指摘】

- 各ノードに分散された空き資源をまとめて、新しいジョブを実行したい（ジョブ実行状況により空き資源の分散が必ず発生、実行中ジョブを他ノードへ移動できれば対応可能)。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- ジョブマイグレーションにより実現可能だと認識しています。
  現在、次期システムでのジョブマイグレーション/ジョブ凍結/定期的チェックポイントの実現に向け、技術的な目処がつき、プロトタイプ版の開発に着手しました。

〈富士通：2006/2/17 第 5 回会合〉

(14) 実行中ジョブの制限値が変更できない

【指摘】

- 実行時間などの制限値を一時的に変更したい。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- 実行中ジョブの制限値（CPU 時間や経過時間制限値）を変更できるように改善します。
  変更機能として、ジョブ投入者、システム管理者向けに qmodify コマンドを新規に提供予定（2006 年 12E 目標）です。詳細は、「添付資料 14-1 実行中ジョブの制限値変更について」をご参照願います。

添付資料 14-1 を参照

〈富士通：2006/2/17 第 5 回会合〉

(15) UDTU 活用が難しい

【指摘】

- ノード内が UDTU 未使用ジョブのみとなった場合に UDTU が無駄となる。
- UDTU の使用状況が把握しにくい。Shared-UDTU は遅い。(プログラム依存)
- 測定結果を例示してほしい。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- 宇宙航空研究開発機構様及び日本原子力研究機構様での評価結果を添付資料 15-1 及び 15-2 に示します。

添付資料 15-1, 15-2 を参照

〈富士通：2006/2/17 第 5 回会合〉

(16) NQS のバッチ share 実行

【指摘】

- 期待通りの share 実行とはならない。

〈JAXA：2005/2/16 第 2 回会合〉

【回答】

- Share モードに対する見解については、第 3 回会合（2005/6/29）で説明した添付資料 16-1「Share 運用での課題に対する回答」を参照されたい。

添付資料 16-1 を参照

〈JAXA：2005/6/29 第 3 回会合＞

(17) Checkpoint/Restart（長時間ジョブ向け対策）

【指摘】

- CPU 異常でプロセスが KILL されてしまった場合、長時間ジョブを先頭から再実行では非常に無駄。異常発生直前のチェックポイントから再開できると無駄が少なくて済む。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

- Checkpoint/Restart（長時間ジョブ向け対策）は、将来的にもぜひ要望したい。

〈JAXA：2004/11/24 第 1 回会合〉

【回答】

- 項番 13 の回答を参照ください。

〈富士通：2006/2/17 第 5 回会合〉

(18) キャッシュ縮退時に性能劣化の考慮

【指摘】

- キャッシュが縮退すると性能にブレが発生し、ジョブ実行性能の再現性が保証できなくなる。当該 CPU を新規ジョブには割当てない考慮が必要。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- キャッシュ縮退が発生した場合、syslog メッセージ(注)が出力されますので、以下の対応をお願いします。
  - swatch(フリーソフト)や Systemwalker などのメッセージ監視機能を利用し、キャッシュ縮退メッセージを監視します。
  - メッセージ出力時に当該 CPU を Parallelnavi の割当抑止機能(pwadmin コメンド)により無効化し、新規ジョブへの割当を抑止します。

注）syslog メッセージ

WARNING: [AFT1] WAY Degradation Event detected by CPUX, errID XX.XX AFSR XX.XX AFAR D1 XX.XX AFAR U2 XX.XX

・"WAY Degradation"：キャッシュ縮退が発生したことを示します。

・"CPUX"　　：キャッシュ縮退が発生した CPU を示します。

X の個所に当該 CPU の CPU-ID が出力されます。

〈富士通：2006/2/17 第 5 回会合〉

(19-1) 一つの CPU 故障だけでノードを停止させない（ﾕｰｻﾞﾌﾟﾛｾｽ走行時）

【指摘】

- CPU 故障が発生した場合、現状、プロセス KILL 後リブートであるが、リブートまでしなくても良いのでは。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- CPU 故障発生時、プロセスの動作を保証することができないため、当該プロセスを kill しています。さらに、kill されたプロセスが運用上必須なプロセスである可能性を考慮し、システムをリブートしています。
- Solaris10 で以下のようなノードダウンしない仕組みを実現しました。

  ・ユーザプロセス走行時の CPU 異常　　：実行中のプロセスを kill した後、重要なプロセスが kill された場合のリスタート方法を定義できる機能(Solaris Fault Manager：SMF）により、CPU 縮退した状態で運用継続が可能になります。

〈富士通：2006/2/17 第 5 回会合〉

(19-2) 一つの CPU 故障だけでノードを停止させない（OS 走行時）

【指摘】

- OS 走行時に異常発生しても簡単にパニックさせないでほしい。

  現状、OS での CPU 異常はパニック。一つの CPU 異常で、ノードダウンしないような考慮が必要。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

- 一部の障害でノード全体がダウンしてしまう。

  ジョブ用システムボードで障害が発生してノード全体がダウンしてしまう。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- CPU 異常発生時は、データの保証ができないためパニックとせざるを得ません。

〈富士通：2006/2/17 第 5 回会合〉

(20) ジョブ凍結

【指摘】

- 保守作業にノード停止が必要な場合、当該ノード上で実行中の長時間ジョブの実行継続性を保証する必要がある。その手段として、実行中ジョブの凍結→システム停止→保守→システム起動→ジョブ解凍が考えられる。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- 定期的チェックポイント(ダウン対策)とジョブ凍結（システム保守向け）の実現については、項番 13 の回答を参照ください。
  現システムでは、システム凍結機能をご利用いただきたい。システム凍結を利用できる範囲は以下のとおりです。
  - システムメモリ全体を凍結・解凍
  - システム凍結中の保守：CPU, メモリ, SB, DDCON のハード交換可
  - SYSVOL ミラー：Solstice Disk Suite を利用（GDS による SYSVOL ミラーは不可）
  - テープ装置がなど DR(Dynamic Reconfiguration)不可なハードが未接続であるシステムのみ

〈富士通：2006/2/17 第 5 回会合〉

(21,22) ジョブマイグレーション

【指摘】

- 上記に加え、他ノードへマイグレーション可能であれば、ジョブの一時中断が短時間で済み、運用性が向上する。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

- ジョブが走っていると、ノードの電源が切れない。

〈名大基盤：2005/6/29 第 3 回会合〉

【回答(第一次)】

- 重要性は認識しているものの、Solaris OS での実現には多くの時間が必要なため、非サポートとしています。

〈富士通：2004/11/24 第 1 回会合〉

【回答(第二次)】

- 次期システムでのジョブマイグレーションの実現に向け、技術的な目処がつき、プロトタイプ版の開発に着手しました。

〈富士通：2006/2/17 第 5 回会合〉

(23) 障害ノードが運用中に活性保守できない

【指摘】

- 障害復旧にはシステム停止による部品交換が必要である。
  NQS ジョブが実行中で停止できない。(実行中ジョブを他ノードへ移動できれば対応可能であるが)

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- PRIMEPOWER HPC2500 では、拡張インターリーブをサポートしているため以下の範囲のみ活性保守が可能となっています（SB の活性保守が制限)。
  ・SCF(システム制御ボード)（ノード/HS)
  ・フロントエンド電源(AC-DC 変換ユニット)（ノード/HS)
  ・クロスバ用電源ユニット(DC-DC コンバータ)
  ・HS 用電源ユニット
  ・冷却ファン（ノード/HS)
  ・内蔵ハードディスク(ミラーなど要)
  ・HS 側光モジュール(但し該当ポートに接続されるノードは切り離し)

〈富士通：2006/2/17 第 5 回会合〉

(24) ホットスタンバイ時の省電力

【指摘】

- VPP に比べ消費電力が多く、エンドユーザへサービスしていない間の消費電力を極力抑えたい。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- SMP の UNIX サーバは他社も同様に、CPU・メモリ・ディスクともアイドル時を含め稼動状態のままです。CPU はアーキテクチャ上動作したままです。

〈富士通：2004/11/24 第 1 回会合〉

(25) 起動・停止と縮退運転：高速 IPL

【指摘】

- エンドユーザへサービスしていない間の消費電力削減。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- 電源投入時に多くの時間が必要であったハードウェア診断(POST)については、メモリチェックを全 CPU で行うことや CPU の性能向上により、概ね 40 分(1.3GHz CPU)を 25 分(2.08GHz CPU)に改善しております。なお、複数のお客様から POST

の簡易化を要請されておりますが、リブート時に構成変更されている可能性があり、診断を削除することはできません。

〈富士通：2006/2/17 第 5 回会合〉

(26) 電気使用量

【指摘】

- VPP5000 と PRIMEPOWER HPC2500 の電気使用量を比較した際、HPC2500 が高い。(2 倍弱)

〈名大基盤：2005/6/29 第 3 回会合〉

- 電気代節約、障害時保守の効率化のため、JOB マイグレーションを可能にしてほしい。たとえ実現に時間がかかっても構わない。

〈京大学情：2005/6/29 第 3 回会合〉

【回答】

- 項番 9 に示した「ノード集中割り当て機能」によるノード資源の効率利用をご検討ください。

〈富士通：2006/2/17 第 5 回会合〉

(27) 起動・停止と縮退運転：ノード起動・停止の空調機連動

【指摘】

- ジョブ混雑具合により、ノードを起動・停止し省電力を図る → ノード起動・停止と連動し空調機も起動・停止

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- 空調機のインタフェースなどセンター毎に対応内容が異なるため、個別に対応させていただいております。
  宇宙航空研究開発機構様では、各ノードと空調機の高度な自動連動機構を開発されています。

〈富士通：2006/2/17 第 5 回会合〉

(28) 起動・停止と縮退運転：ノード起動・停止の NQS 連携

【指摘】

- ジョブの混雑状況に応じて、ノード起動・停止を自動化。停止時は、当該ノードにジョブが新規に実行されないなどのジョブスケジューリング上の工夫が必要。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- ジョブ混雑の考え方、起動・停止タイミングなどお客様毎に要件が異なるため、富士通の製品では制御するための部品(電源制御コマンド、ジョブ状態監視コマンドなど)を提供しています。京都大学様、宇宙航空研究開発機構様での利用方法は以下のとおり。

□京都大学様

ノード起動：ジョブのキューイング状況をチェックし、待ちジョブの alloc CPU 数に応じて起動ノード数を決定。128CPU 以内で 1 ノード、129～256CPU なら 2 ノードといったルールを決めており、なるべく空調機と連動しているノードのペアで起動させている。現在は 20 分単位で cron で実行。

ノード停止：単位時間(例えば 1 時間)内でノードにジョブがいなければノード停止。
ただし、猶予時間を与えそれ以内にジョブが投入された場合は停止処理を中断する。現在は 30 分単位で cron で実行。

□宇宙航空研究開発機構殿様

NSJS(独自のスケジューラ)と設備の連動機構を開発しています。
例えば、NSJS はカレンダを持っており、停止対象ノードのジョブ割当を抑止し、安全に停止する機構がある。

〈富士通：2004/11/24 第 1 回会合〉

(29) 起動・停止と縮退運転：ジョブ凍結・マイグレーション

【指摘】

- ノード停止時に実行中ジョブを他ノードへマイグレーション（もしくはジョブ凍結）できれば、より一層効率的な省電力運用が可能。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答(第一次)】

- 重要性は認識しているものの、Solaris OS での実現には多くの時間が必要なため、非サポートとしています。

〈富士通：2004/11/24 第 1 回会合〉

【回答(第二次)】

- 次期システムでのジョブマイグレーションの実現に向け、技術的な目処がつき、プロトタイプ版の開発に着手しました。

〈富士通：2006/2/17 第 5 回会合〉

使い勝手と性能

(30) qsub のシェルでは融通性がない

【指摘】

- エンドユーザからみた使い勝手の向上を図りたい。
  例えば以前の MSP のように、キーパラメータだけでプログラム名を指定し変更できるとよい。また、やがては WEB インタフェース化したい。

〈京大学情：2004/11/24 第 1 回会合、10/27(科学技術計算分科会)〉

【回答】

- エンドユーザの使い勝手向上は、Parallelnavi Workbench にてカバーする考えですが、現状の使い易さをどう把握しているのかや WEB インタフェースのサポート等について、開発部門に状況と方針等を確認し、検討させていただきたい。

〈富士通：2004/11/24 第 1 回会合〉

【コメント】

- メーカから提供されている Workbench を使用することは、エンドユーザにとっては難しい。

〈京大学情：2004/11/24 第 1 回会合〉

- JAXA では、プラットフォームが変わっても共通のエンドユーザ向けインタフェースを提供するために NS コマンドというスクリプトレベルの簡易インタフェースを提供している。Workbench は高機能である反面、簡単に使用したいエンドユーザには向いていない。

〈JAXA：2004/11/24 第 1 回会合〉

- メーカが Workbench を製品化するとエンドユーザにとって難しすぎるものになるとの印象がある。もっと軽く簡単なものでよい。

〈京大学情：2004/11/24 第 1 回会合〉

JAXA 開発のツール、HPC ポータル等を導入し検討する。〈WG 総意〉

(31) 特定のジョブの実行が他のジョブの経過時間に影響を与える問題

【指摘】

- 特定のジョブの実行により、転送回数が多い他のジョブの経過時間に大きく影響を与えてしまう問題が発生していた。この原因は、DTU であった。転送要求の多いジョブはプライオリティを高めるなど、ユーザごとに DTU を調節する機能が欲しい。
- また、この原因を解明するのに 2 年半もかかった。このような問題点は、開発者側から予め教えてほしかった。

〈JAXA：2005/2/16 第 2 回会合〉

【回答】

- DTU 競合による他のジョブの経過時間に影響を与える原因の説明を添付資料 31-1 に示します。
  また、宇宙航空研究開発機構様での経過時間のブレ分析結果を添付資料 31-2 に示します。

添付資料 31-1,31-2 を参照

〈富士通：2006/2/17 第 5 回会合〉

(32) ジョブ実行状況の確認が遅い

【指摘】

- rscsetstat コマンドのレスポンスが遅く、負荷も高い。
  ※多重実行でメッセージ

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- Parallelnavi 2.4.1 にて、ノード内の情報収集処理の性能改善を実施済み。
  例：128 プロセス並列/1 ノードの情報収集処理：20 秒→1 秒に減少

  《回避策》（日本原子力研究開発機構様、宇宙航空研究開発機構様での回避策）
  cron で定期的に rscsetstat コマンドを実行しファイルへ保存、エンドユーザはこのファイル内容を表示するスクリプトを利用している。

〈富士通：2006/2/17 第 5 回会合〉

(33) メモリアクセス性能改善

【指摘】

- キャッシュミス/TLB ミスのプログラム改善対策はユーザ負担が大きい。
  既存プログラムの高速化チューニングが容易ではない。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- 宇宙航空研究開発機構様では、性能チューニングのために、性能チェックシートを作成し、ユーザ支援する仕組みを実施しています。
  宇宙航空研究開発機構様で利用しています「性能阻害箇所の検出ツール」及び「性能チェックシート」を添付資料 33-1 及び 33-2 に示します。

添付資料 33-1,33-2 を参照

〈富士通：2006/2/17 第 5 回会合〉

(34) ポートスキャンで NQS トラブル

【指摘】

- ポートスキャンで NQS バッチリクエスト起動デーモン(nqsexecd)異常終了
- 新規ジョブが実行できなくなったため、改善要望し、該当デーモンの異常終了を監視するようにした。
  トラブルシューティングとして運用に必要なサービス(デーモン)が正常に起動されているか確認する手段が必要と思われる。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- NQS デーモンの異常終了は、Parallelnavi 2.4.1 で対応済です。
  Parallelnavi で提供するそれ以外のデーモンで同様な問題がないことは確認済みです。
  《回避策》(日本原子力研究開発機構様での回避策)
  ps コマンドを定期的に実行し、nqsexecd が起動されていない場合にメール通知するような運用ツールを作成しで対処しています。

〈富士通：2006/2/17 第 5 回会合〉

(35) 適切な稼働率の計算方法

【指摘】

- 稼働率が低いノードからジョブを詰めていくため、ジョブが分散し稼働率が低下してしまう。(稼働率の定義を含め、評価方法を検討中である)

〈名大基盤：2005/6/29 第 3 回会合〉

【回答】

- 項番 9 に示した「ノード集中割り当て機能」により、稼働率の高いノードからジョブをつめていく機能を予定しています。

〈富士通：2006/2/17 第 5 回会合〉

(36) 不足資源調査コマンド

【指摘】

- 現在の仕様では、Reason 欄に、どの資源が不足しているかが文字列で表示される。(例：CPU,MEM,UDTU)
  より分かりやすくなるように、グラフィカル表示してもらいたい。

〈名大基盤：2005/6/29 第 3 回会合〉

【回答】

- 名古屋大学様：JAXA 殿で稼動中の CMS(システムの利用状況を Web ブラウザによりグラフィカル表示するツール)を流用することにより対応予定である。

〈富士通：2006/2/17 第 5 回会合〉

I／O

(37) PRIMECLUSTER　GFS によるディスク共用は遅い

【指摘】

- 小規模ファイルを多数アクセスすると遅い。

〈JAEA：2005/2/16 第 2 回会合〉

【回答】

- PRIMECLUSTER GFS において以下のような操作におけるレスポンスが UFS と比較して悪いとのご指摘と認識しております。
  - ・ 多数（千ファイル程度）をアーカイブした tar ファイルの展開
    (= 多数・小容量ファイルの一斉新規書き込み)

  PRIMECLUSTER GFS は、複数ノード間での共用ファイルシステムであり、ノード間での整合性維持のためファイルのメタ情報（ディレクトリ情報など）を MDS (Meta Data Server) というサーバ・プロセスにて管理します。tar ファイルの展開のように一度に多数のファイルを新規に作成するような操作では、MDS との通信が多発するため UFS と比較すると高いレスポンスが得られません。ノード間共用ファイルという機能の性格上、短期的に大きなレスポンス改善につながる対応は技術的に困難な状況であり、以下の対応につきご検討いただきますようお願いいたします。

  tar ファイルのような多数かつ小容量のファイルを扱うのは、プログラムのコンパイル（make）　時など一時的な操作と想定しております。

  make のような操作は、各ノードのローカルなファイルシステム（UFS）上で実施し、make したバイナリを GFS 上へ複写し複数ノードで共用するような運用をご検討願いたい。

〈富士通：2006/2/17 第 5 回会合〉

その他

(38) Parallelnavi のエンハンス状況を知りたい

【要望】

- 本 WG にて出された課題が契機となってエンハンスする機能を含めた Parallelnavi 全体のエンハンス内容 (実施済み、計画) について知りたい。

〈京大学情：2006/2/17 第 5 回会合〉

【回答】

- Parallelnavi エンハンス一覧を、添付資料 38 に示します。

添付資料 38-1 を参照

〈富士通：2006/2/17 第 5 回会合〉

(39) 大規模 SMP クラスタ運用のユーザ事例を、普及すべき

【要望】

- 大規模 SMP クラスタにおける運用に工夫を重ねるユーザの事例は、既に大規模 SMP クラスタを導入されている機関は勿論のこと、これから導入予定の機関にも参考になるのではないか。ユーザ事例を広げるべきではないか。

〈WG 総意：2006/5/30 第 6 回会合〉

【回答】

- JAXA の大規模 SMP クラスタにおける運用の課題と工夫を、添付資料 39 に示します。

添付資料 39-1 を参照

〈JAXA：2006/9/15 第 7 回会合〉

(40) HPC Portal への要望

【要望】

- SSL に対応した際、ファイルのアップロード遅延が発生。OS の違い（Windows2000 と WindowsXP）によりアップロード時間が大幅に違った。（10 倍程度）
- パスワード変更機能を追加してほしい。

〈以上 2 点、JAEA：2005/10/07 第 4 回会合〉

- 例えばジョブ投入の際にマニュアルを参照したい場合、一旦メインに戻らないとならないのは不便。マニュアルのトップページを、別ウィンドウで表示する等の手段は?
- 「ファイル操作」で日本語のディレクトリ名が作成できてしまうのはよろしくない。
- 自分の好きなエディタを使うことができれば便利。（例：Windows 版 emacs）

〈以上 3 点、名大基盤：2005/10/07 第 4 回会合〉

【回答】

- 添付資料 40-1 に示します。

添付資料 40-1 を参照

〈富士通：2006/10/27〉

(41) 富士通の WWSite 構築サービスの商品体系

【要望】

- HPC Portal を含む、富士通製 R&D サーバ WWSite 構築サービスの商品体系を教えてほしい。

〈京大メディア：2006/5/30 第 6 回会合〉

【回答】

- 添付資料 41-1 に示します。

  添付資料 41-1 を参照

〈富士通：2006/10/27〉

以上