HA8000 クラスタシステムの利用に関するアンケートの集計結果

スーパーコンピューティング部門

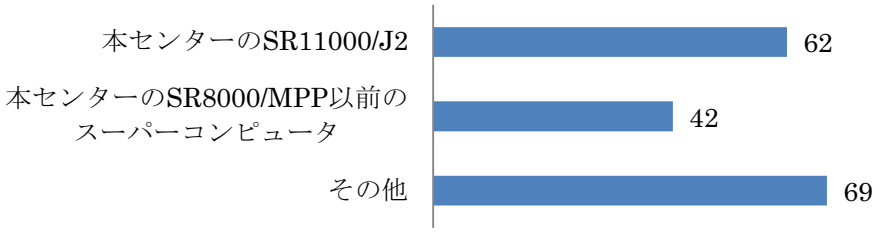
本年 6 月から 9 月末まで、HA8000 クラスタシステムを無料で利用できる試行サービスを行いました。利用者の方にアンケートを実施させて頂きましたので、その結果をご報告致します。

アンケート期間：平成 20 年 9 月 29 日(月) ～ 10 月 14 日(火)
アンケート依頼者数：353 人
アンケート回答者数：152 人
アンケート回答率　：43.1%

アンケート依頼者数は、同一人物がパーソナルコースとグループコースなどで複数のアカウントを持っている場合も 1 人として数えた場合の人数です。

1．一般

(1-1) これまでに利用したことのあるスーパーコンピュータを教えてください（回答者 117 人）

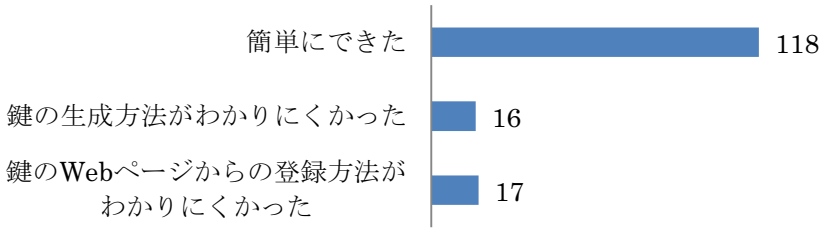


本センターのスーパーコンピュータを利用したことがある、と回答した人は 80 人でした。逆に 72 人がこれまで本センターのスーパーコンピュータを利用したことのない新規ユーザーということになります。

「その他」に挙げられていたスーパーコンピュータには、以下のものがありました。

海洋研究開発機構の地球シミュレータ(12 人)、東京大学物性研究所のスパコン(12 人)、東工大の TSUBAME(10 人)、自然科学研究機構計算科学研究センターのスパコン(5 人)、京都大学学術情報メディアセンターのスパコン(5 人)、理化学研究所情報基盤センターのスパコン(4 人)、筑波大学計算科学研究センターのスパコン(4 人)など。

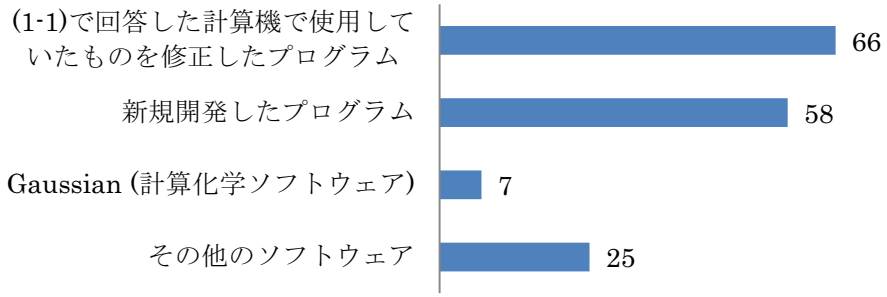
(1-2) ログインは簡単にできましたか？（回答者 151 人）



他に「ログインを試みたがうまくいかなかった」という人が 1 人いました。

本システムへのログインに関して、これまで利用されていたパスワード認証ではなく、最初のログインから鍵認証方式のみを利用する方式としました。すでに、鍵認証方式が一般的になってきたためか、大きな混乱はなかったようです。

(1-3) 試用期間中に実行したプログラムについて教えてください （回答者 142 人）

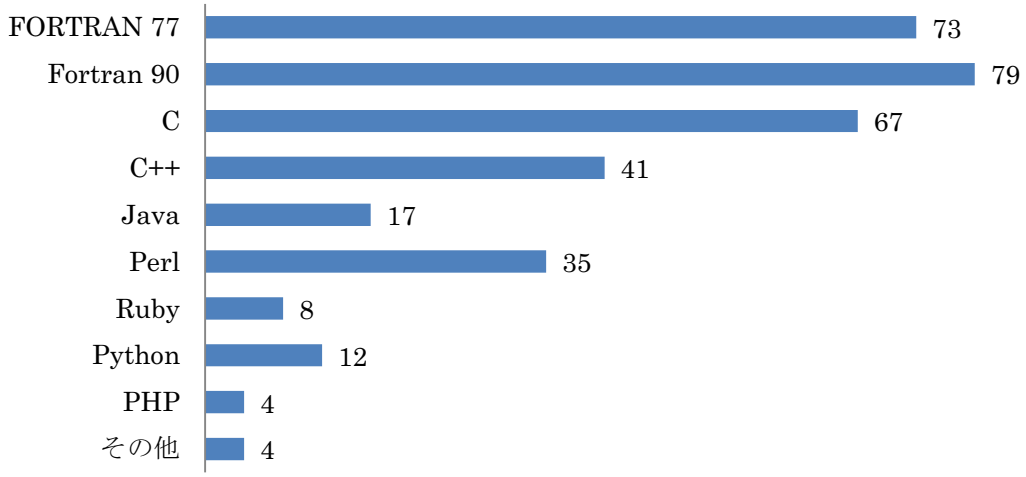


「その他のソフトウェア」として挙げられていたものは、下記のとおりでした。

SIESTA(2 人)、NAMD(2 人)、Amber(2 人)、PHASE(2 人)、LBLRTM、STREAM、姫野ベンチ、ADVENTURECluster、ポアソン方程式の反復解法ベンチマークコード、画像重畳コード、固有値計算関数、Frontflow/red(RSS21)、meep、OpenFOAM(各 1 人)。

2. プログラム開発環境

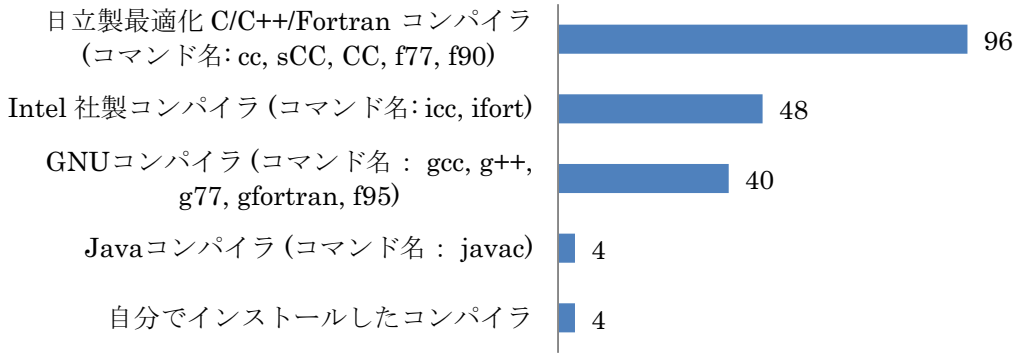
(2-1) あなたが普段利用しているプログラミング言語を教えてください（HA8000 上に限りません）(回答者 149 人)



「その他」として挙げられていたものは、下記のとおりでした。

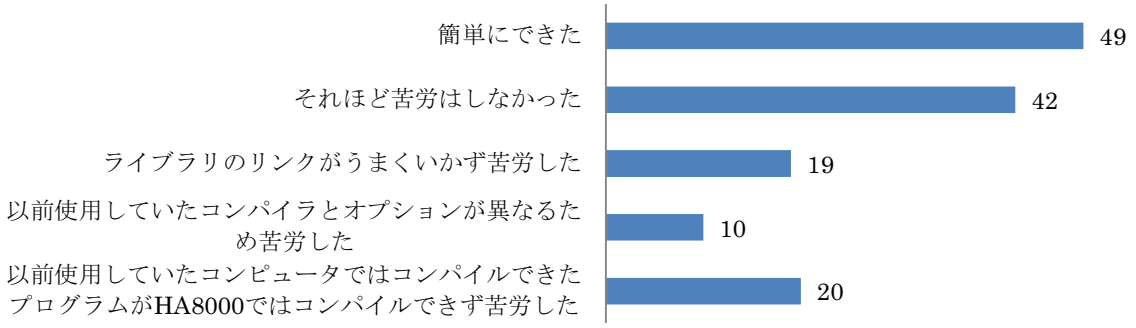
Objective Caml、bash、Ox、JavaScript(各 1 人)

(2-2) HA8000 上で主に使用したコンパイラを教えてください （回答者 139 人）

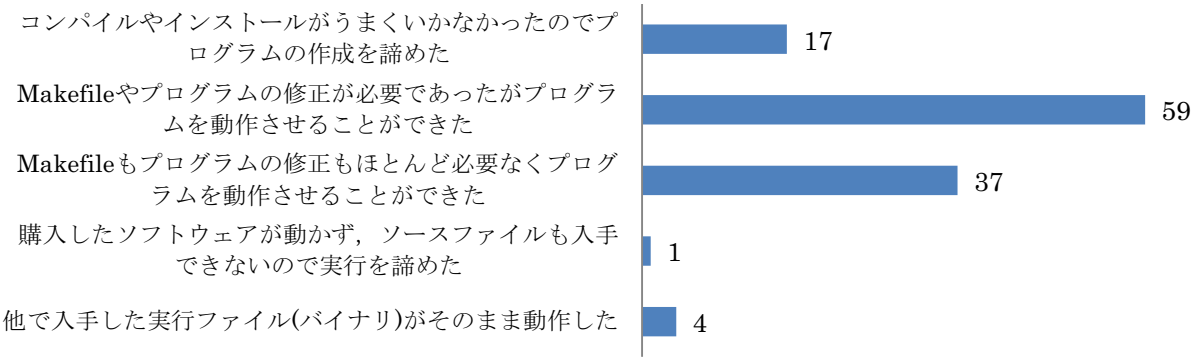


「自分でインストールしたコンパイラ」として挙げられていたものは、下記のとおりでした。
Fujitsu Compiler、Objective Caml、g++の最新版、gfortran(各 1 人)

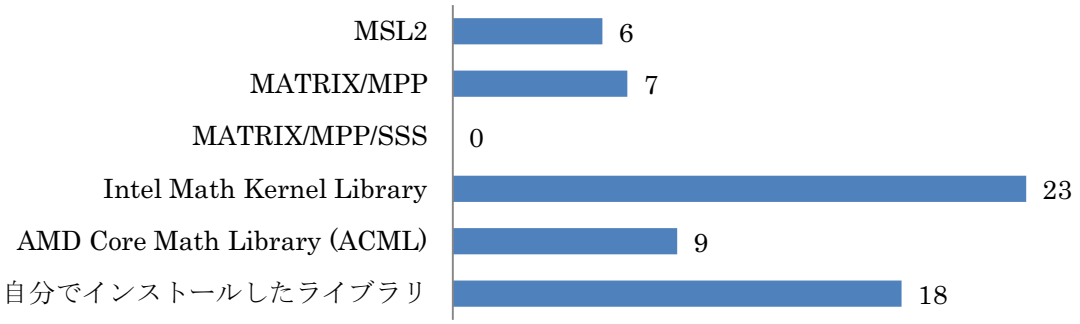
(2-3) コンパイルはすぐにできましたか？ (回答者 140 人)



(2-4) 試用期間中に次のようなことがありましたか？(回答者 108 人)



(2-5) 試用期間中に利用したライブラリを教えてください (回答者 47 人)

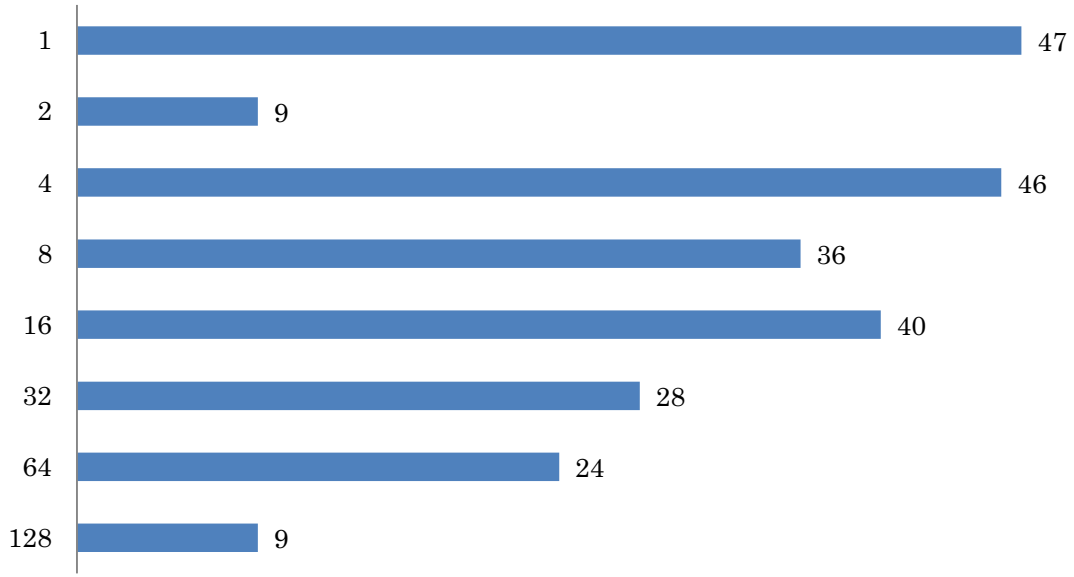


「自分でインストールしたライブラリ」として挙げられていたものは、下記のとおりでした。
GotoBLAS(5 人)、LAPACK(3 人)、Xerces C++(3 人)、Boost(3 人)、ACML(2 人)、FFTW(2 人)、MPICH、cppunit、cpplapack、clapack、LAM/MPI、OpenMPI、MPICH2、libXML2、zlib、ScaLAPACK、FMLIB、GSL、LAPACK95(各 1 人)

3. キューについて

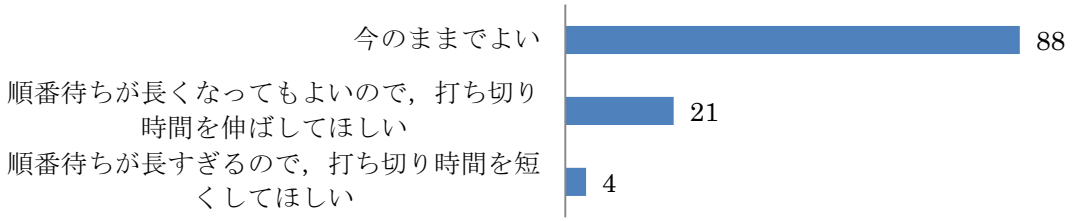
(3-1) バッチジョブの投入は簡単にできましたか？（回答者 138 人）

(3-2) 比較的よく利用したノード数（NQS スクリプトの「#@$-N」で指定した値）を教えてください（最大 3 つまで） （回答者 133 人）



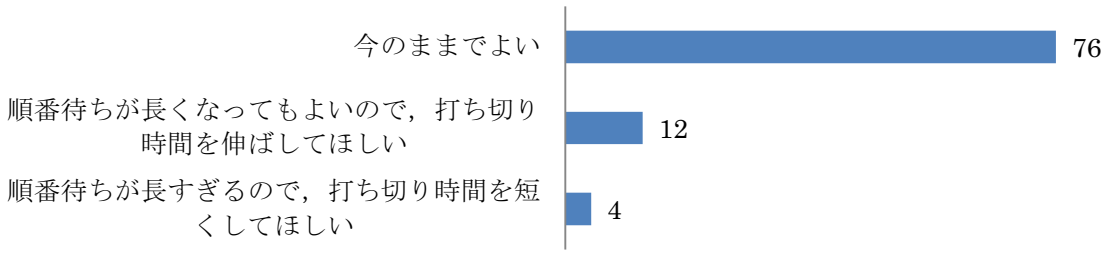
HA8000 クラスタシステムでは、1 ノード専用のキュー（SR11000 の P001 キューに相当するもの）を用意しておりませんが、1 ノードだけの利用も多かったようです。

(3-3) デバッグキューを使用された方：ジョブの打ち切り時間(5 分)は適切ですか？（回答者 113 人）

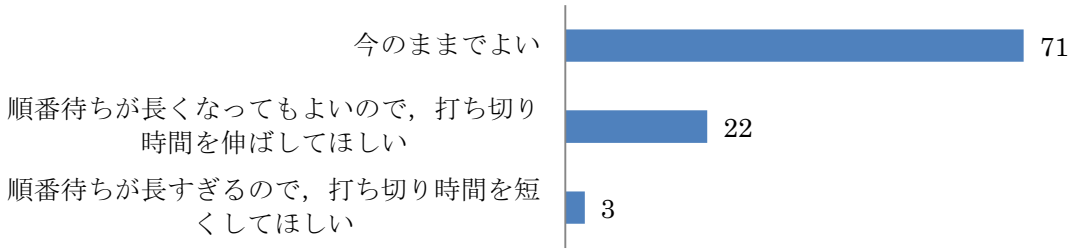


デバッグキューの打ち切り時間の決定は悩ましい問題です。本センターとしては、5 分以上かかるプログラムの実行は、制限時間が 1 時間の short キューを利用して頂きたいと考えています。

(3-4) パーソナルコースの方:short キューのジョブの打ち切り時間(1 時間)は適切ですか？（回答者 92 人）



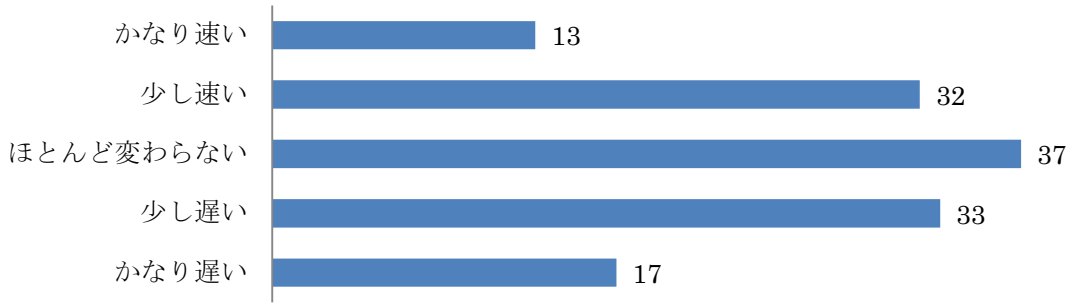
(3-5) パーソナルコースの方：parallel キューのジョブの打ち切り時間(P016 までは 24 時間、P032 以上は 12 時間)は適切ですか？（回答者 96 人）



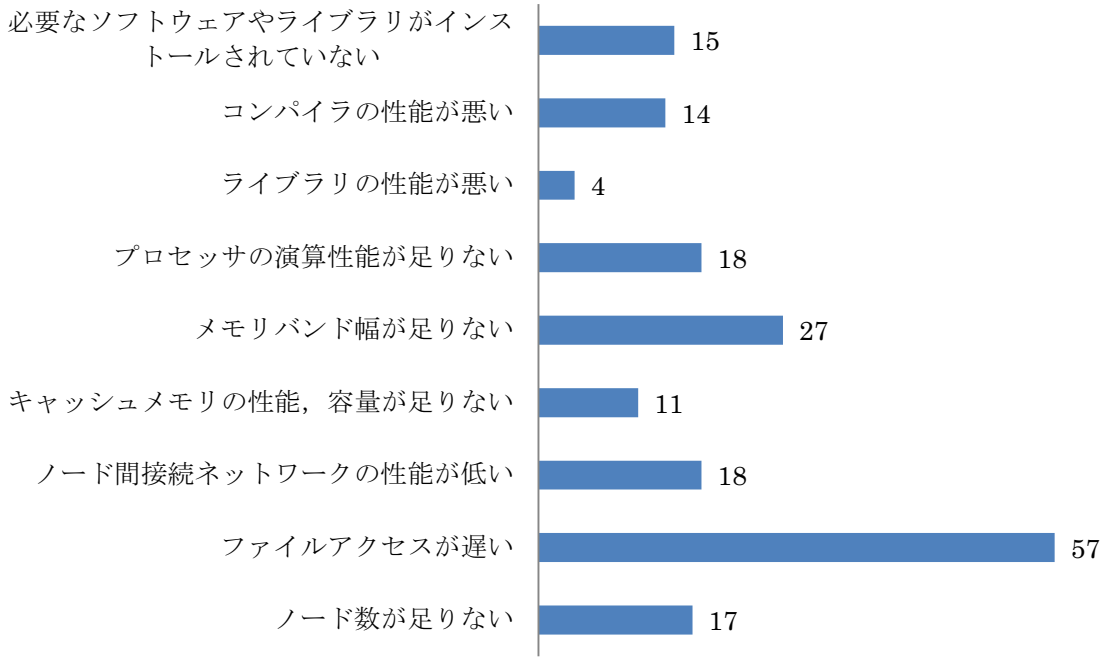
(3-6) 9 月 5 日から 128GB メモリの計算ノードが利用できる large キューのサービスを始めましたが、ご存じでしたか？ （回答者 142 人）

4. 性能について

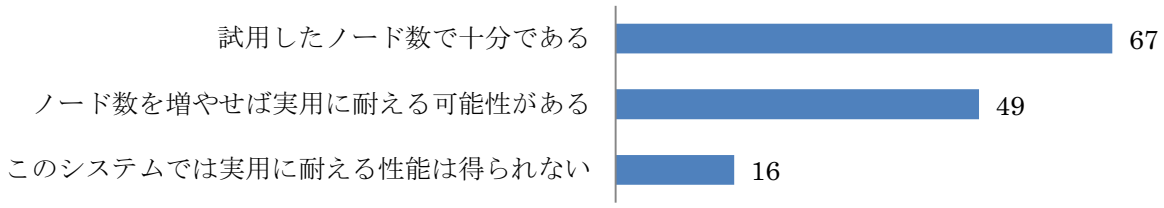
(4-1) 全体として、期待していた性能（あるいはこれまで利用していた計算機）と比較してどうでしたか？（回答者 132 人）



(4-2) 下記の点に関して、感じたものについてチェックを入れてください（回答者 106 人）



(4-3) 今後研究を進める上で十分な性能が得られましたか？（回答者 132 人）



5. その他

自由記入項目欄では、たくさんのご意見やご要望を頂きました。ここで、ご記入頂いた内容を全てご紹介致します。なお、読み易くするために、意味を変えない程度に少し文章の修正を行った部分があります。

(5-1) HA8000 上にインストールして欲しいソフトウェア、ライブラリ、ツールがありましたら、下記にご記入ください

- 一般的な UNIX システムにあるような/usr/bin 下のツールは自由に使えるようにして欲しい。たとえば top が一般ユーザーの権限がなく、実行出来ないので、インタラクティブノードが現在どのように利用されているのかが分からず、不便でした。
- IMSL
- GCC など、大きな更新があった際には、新バージョンを使えるようにして欲しい。
- FUSE(http://fuse.sourceforge.net/)
- gcc や intel の c コンパイラで-M オプションに相当するものが日立 C にもあると、他で動いていたプログラムパッケージの makefile の edit の際、楽なのですが、ご検討くだされば幸いです。
- 最適化された Gaussian03、GAMESS
- FOX-GUI ライブラリ：このフリーのライブラリを使用して開発していますが、日立コンパイラではコンパイルできず、諦めて GNU コンパイラを使用しました。

- 速い SCALAPACK
- OpenMPI をインストールしていただきまして、大変感謝しております。もし可能でしたら、Gigabit Ethernet だけでなく、Myrinet も使用できるようにしていただけましたら幸いです。
- XML 関連、PGI コンパイラ、ACML
- FFTw3
- python
- 既にインストールされていなければ、Xvfb
- ノード内並列＋ノード間並列混合にもハングしない ScaLAPACK
- Frontflow/red
- 分子動力学ソフトの charmm をインストールしてほしい。
- インストールして欲しいソフトウェア
  ① meep （フリーFDTD 電磁界解析ツール）
  ② mpb （フリー平面波解析ツール）
  以下は、上記のソフトをインストールするために必要なライブラリ
  libtool、guile、libctl、harminv、fftw2、fftw3、zlib、szip、hdf5、libpng、vis5d、libmatheval、h5utils、gsl
- プログラムの性能を評価するツールが必要。日立製コンパイラのノード内並列化機能の状態を評価できること。プログラムの各部の FLOPS 値、キャッシュや TLB の利用効率の評価も行なえること。
- GNU C++コンパイラをこれからもインストールしてもらいたいです。
- ACML ライブラリのインストール
- 第一原理計算プログラム（VASP など）。センターとして HA8000 に最適化したものを提供できると特によいと思います。
- たとえば、MPI に比べて通常は性能が出ないであろうが、試験的にインテル社の ClusterOpenMP を利用可能に出来ないだろうか？
  私は使ったことはないのだが、AMD のシステムでは PGI のコンパイラがよく取り上げられているように思うが、これはたとえばインテルコンパイラに対して Opteron にとって性能的にはどうなのであろうか？
  （たとえばインテルコンパイラは Core2Quad　CPU 等では-xT や-fast というコンパイラオプションを許すが、Opteron 上では-xT や-fast を指定すると、コンパイルは出来ても、実行時に CPU がインテル製でないことを検出して、実行を拒絶するバイナリを作る。Core2Quad 上は-xT -O3　-fast にくらべて-axT -O3 は若干性能が落ちることからみて、AMD の CPU に対しては多少なりともわざと不利な扱いをしていると思われる。
- インテルのコンパイラ
- job をチェーン化できるよう何とかしてほしい。
- Intel Thread Building Block
  CPPUNIT
- FFTW
- mpif90(ifort の)から利用できる MKL。なぜか HA8000 上では、mpif90(ifort の)+MKL の組み合わせでうまく使えなかったので。
- mpich-mx を GCC でコンパイルしたもの(shared も)
  NUMA 関係のスクリプト
- JavaParty、その他 Java 用分散処理系
- ①現在、PHASE ver.701 がインストールされているが、使用していてチューニングが不十分な気がします。開発元に使用していただいて改良を継続して欲しい。プログラムを陳腐化させないためにもメンテは続けて欲しい。

②CHASE-3PT のような PHESE のプリポストプロセッサーを HA8000 上にインストールして欲しい。原子数が多くなるとローカルの PC で入力データを作ったり、計算結果をグラフィック表示するのは限界を感じており、PC を X 端末として使用するような形態は取れませんか。

- よりよいファイルシステム

要望の多いフリーソフトウェアにつきましては、センター側でインストールを行ったり、インストール手順についてドキュメントを公開したりしていく予定です。

(5-2) 利用の手引に関して、誤っている点、わかりづらい点、内容を充実して欲しい点がありましたら教えてください。

- はっきりとは覚えていないが、C コンパイラで説明の通り並列化と最適化のオプションを組み合わせてコンパイルしたら、うまくコンパイルできなかったような覚えがある。
- OpenMP と MPI を使い分けることのメリットやその具体例が知りたいと思いました。
- NUMA の利用例を増やした方がよいと思われる。
- ①ps など、一部のコマンドの仕様や管理の方針
  ②キューの混雑度などを調べる方法があれば解説してほしい
- Intel 系のコンパイラ付随の lib*.so ファイルが実行時に呼び出せず、-static でのコンパイルを余儀なくされました。こちらのミスかもしれないのですが、Intel 系のコンパイラ・ライブラリの使用方法に対する記述を、手引きに簡単にでも加えて頂ければと思います。
- pdf の目次からリンクで参照できるようにしてほしい。
- 上記の設問で ACML の名称があったので、ACML がインストールされていたのかもしれませんが、そのことを知らず MKL を使っていました。試したことがないのですが、たぶん Opteron では、MKL よりも ACML の方が高速に処理できるのだと思いますが、実際どのくらいの性能差がでるのか示していただけたら幸いです。
- Intel コンパイラが利用可能でしたが、その説明が無かったと思います。
- NUMA 最適化についてもう少し初心者向けの説明がほしかった。
- ステージインや numa コントロールに関して細かい使用方法を記載して欲しい。
- バッチスクリプトの実行が自分のホームディレクトリで始まることを知らずに混乱した。
- ハイブリッド実行
- マニュアルに従って try しても TTSSH での login ができなかった。
- クラスタとノードがわかっていないので、システム概要を読んでも理解できなかった。ジョブの実行でも“ノード当たりのプロセス数”といわれても、良くわからなかった。
- numactl についての言及もあってもよいと思います。
- SR11000 での性能評価オプションを HA8000 においても引き継いで欲しかった。
  また自動並列化等のコンパイルオプションについて、より詳細な説明が欲しかった。
- p15、p43、p44 で、英字のエルが数字の 1 に似ている。その違いが分かる様にした方がよいと思います。
- GotoBLAS、ACML などを個人でインストールする方法について記述が必要(日立製コンパイラの元で正しく利用できるように。ノード内並列化との共存関係についても記述すべき。)
- 非常にわかりやすく、操作に当たって大変参考になりました。
- MPI などを利用しているプログラムを日立製コンパイラ以外のコンパイラを利用してコンパイルする場合、いくつかのアーカイブを手でリンクする必要がありますが、手引きに明示的に書いていただけるとありがたいです。
- 日立コンパイラで設定すべき推奨最適化オプションが、-Os -noparallel であることがわかりにくかったです。

- ジョブスクリプトオプションの説明がもう少し詳しいと助かります。例えば、「プロセス毎の CPU 時間」（#@$-lt）は実行時間とどのように違うのでしょうか。また、メモリ制限の#@$-lM はノードあたりでしょうか、ジョブあたりでしょうか。
- MPI コンパイラの切替え方(/opt/itc/mpi/mpiswitch.sh)を載せて欲しい。
- マルチスレッドによるプログラミング（MPI、OpenMP、Pthread 等）で、プログラムが走行中に各スレッドをそれぞれノード単位で指定して割り当てる方法、さらに、その各スレッド毎に、ノード内の CPU ごとにサブスレッドを指定して割り当てる方法、さらに、その各サブスレッド毎に、CPU 内のコアごとにサブサブスレッドを指定して割り当てる方法の説明。通常の OpenMP や MPI は、共有メモリであるか分散メモリであるかは異なるが、計算エレメントとしては均質なものがずらっと一階層で横に並んで居るイメージモデルをもともと採用しているが、たとえば HA8000 では、それがノード、CPU、コアと三階層となっており、現実とモデルの乖離が激しく、現実に沿ったスレッドの配置や、メモリ上やキャッシュ上のデータを引き継ぐ形でスレッドが割り当てられないと、データの移動ばかりが頻発して性能が落ちるであろう。

  MPI や OpenMP で計算機システム内のデータネットワーク階層を十分に生かすようなプログラミングを行うための制御法（コンパイラ、ライブラリ、OS へのシステムコール）が可能ならば解説されたい。
- バッチジョブ投入用のスクリプトファイルの書き方が分かりにくいので参考例を増やした方がいいと思います。
- ファイルへの書き出し、読み込みに関してどうすれば性能が向上するかについてサンプルプログラムを含め、解説を載せてほしい。
- pthread library を使った独自の並列化プログラムを動かした際、スレッドを多数作成する(しかし、ほとんどは wait して同時に動くのはコア数以下の)プログラムがログインノードでは動いても、ジョブ中では pthread_create に失敗して動かなかった。こういった制限に関する情報が利用の手引き中には見つからなかった。

本件については、10 月からの本運用で仮想記憶メモリの上限設定を行ったことが原因でした。
pthread_create 実行時に標準で 2GB のスタック領域を確保するためメモリ不足となっていました。

- makefile に関する記述を充実して欲しい(サンプルなど)。また、コンパイルオプションによって計算結果が変わることがあるならば、目に付くように注記してほしい。
- NUMA 周りの最適化、numactl の使い方について内容を充実して欲しい。
  Intel Compiler のオプション、特に使用している CPU で使用可能な最適化オプションについて内容を充実して欲しい。
- バッチジョブの指定の例だけでは、具体的に使える単位(メモリ量なら MB、GB、時間なら hh:mm:ss とか)がわからない。また、性能情報の採取方法も簡単に説明してほしい。
- 7.3 ジョブスクリプトの書き方　①主要オプション #@$-T では動作しませんでした。T を lT に変更することで動作しました。

ここで、ご指摘頂いた内容に付きましては「利用の手引」に反映させたり、本センターの Web ページ上にて情報を掲載したりするように致します。

(5-3) 各質問で、選択肢の中に適切な項目がなく他の理由によるものなどがありましたら、下記にご記入ください。試用期間終了後に継続利用されない方は理由を書ける範囲で構いませんので、教えてください。最後に、情報基盤センターへのご意見やご要望などがありましたら、ご自由にお書きください。

- ディスクアクセスが非常に遅く、満足に使えませんでした。

使用しているオリジナルプログラムが頻繁かつ膨大なディスク I/O を発生させるのですが、バッチで流している間、インタラクティブノードの負荷が異常に高くなる現象が見られ、実行出来ませんでした。

- 埼玉大学における説明会で MPI を使わなければ十分な性能はでないとの説明を受けていたので、並列化していない私のプログラムでは予想通りの結果でした。
  グループコースの場合どのキューに投入すればよいのか周知してほしかったと思います。問い合わせたときも回答になっていないような回答でした。
- ジョブ管理システムへ登録できるキューの数を増やしてほしい。
  キューの判定時間を短縮してほしい。
  ファイルアクセスのケース分けをして、それぞれに最適化したファイルシステムを用意していただきたい。
- 申し訳ありませんが、ログインはさせていただきましたが、まだ未使用なので、上の問い(1-3)、(2-3)-(4-2)は回答できませんでした。
- 巨大なライブラリを ./configure && make しようとすると恐ろしい時間がかかる。ファイルアクセスが遅く、10 万円の PC と比べて体感で 10 倍近くかかる。
- ファイル生成の遅さが致命的であることから、頻繁にファイルにログを出力するプログラムや、ローカルディスクへのアクセスを要求するプログラムの使用に耐えない。
  　たとえば、ファイルを通してデータを通信するプログラムの場合、本システムの特性を活かして計算させようとすると、ファイル経由の通信から、MPI を用いてインフィニバンド経由通信のプログラムに書き直す必要があり、これが本システムを使う際の障壁になっている。
  　from scratch で MPI を用いて書く小規模のプログラムならまだしも、生物情報の扱いのように、他人が作成したソフトウェアを間に通して処理する必要がある場合、すべてを MPI で書き直すこと自体が土台不可能（時間、開発コストが膨大）であり、システム構成（主にファイル処理の遅さ、IP 通信ができない）による不自由が大きい。
- 固定ノードあるいは専用キューを使用する際にもジョブ投入から実行が始まるまで少々時間がかかるのが、ややストレス。
  ファイルシステムの最適化の都合上仕方ないのかもしれないが、ls コマンドのレスポンスが悪いのもストレス。
- SR8000 等にあった日立製コンパイラのプログラム実行時の性能評価の機能が無くなっているのが残念。
- 現在、所属研究室で 8 ノードを使用していますが、同研究室の他の研究者が使用している状況や、自分のジョブの実行が開始されるまでの最長待ち時間などの情報を表示したりできる機能があれば、さらに便利になると感じました。
- ①専用キューでジョブが思ったほど流れなかった。
  ②鍵の Web ページのパスワードについての説明がなかった。（問い合わせないとログインできなかった。）
  ③同じグループのアカウントのジョブの状況が、qstat で出ない。また、他グループを含めた全体のジョブの状況（混雑具合）を調べる術がなかった。
- 16nodes 8cores 128 並列で実行しています。約 4GB の結果の出力に平均して 4,400 s
  24nodes 16cores 384 並列で、約 4GB の結果の出力に平均して 3,600 s 掛かっています。
  T2K から弊社に約 4GB の結果を scp するのに約 3,600 s です。
  16nodes 8cores 128 並列で約 260MB*128 のリスタートファイルを読み込むのに約 13,870 s 掛かります。
  24nodes 占有で利用しています。48 時間で打ち切られますと約 4 時間無駄な時間を必要とします。
  nodes 占有ジョブは出来る限り長期に連続して利用できるようご配慮下さいますようお願いします。

- ifort でコンパイルは出来ましたが、以下のエラーメッセージにより実行は出来ませんでした。
  Use of uninitialized value in subroutine entry at
  /opt/itc/mpi/mpich-mx-intel/bin/mpirun.ch_mx.pl line 872.
  Bad arg length for Socket::inet_ntoa, length is 0, should be 4 at
  /opt/itc/mpi/mpich-mx-intel/bin/mpirun.ch_mx.pl line 872.
  intel ではなく hitachi の mpirun を使う様、指示を受けましたが、それも動かず実行に至っていません。
- 私どもの無理な要求にも丁寧に対応してくださり、大変感謝しております。
  引き続きシステムを利用させていただきます。よろしくお願いいたします。
  ログインして作業するとき、ファイルアクセスがかなり遅いようでしたので、
  特にこの点を改善していただきたいと思っております。
- C++でテンプレート機能を多用したライブラリを日立コンパイラでコンパイルしようとしたところ、最適化オプション-O3 などを指定すると 12 時間以内にコンパイルが終了しませんでした。ということで、日立コンパイラの利用を諦めました。
- 東京大学情報基盤センタースパコン HA8000 で直面している問題は以下の 2 点です。
  ① 128core(8 ノード)のジョブを走らせようとしても実行中(running)の状態のままフリーズします（9 月中は走りました)。
  また、64core（4 ノード）のジョブを二本同時に走らせることはできました。
  ② 計算速度が遅い。
- ディスク性能をもっと上げて欲しい。
  プログラムのチューニング方法に関するドキュメントや講習会などをもっと増やしてほしい。
- 安定稼働を維持するためには、様々なテストや変更を繰り返していると思いますが、使用期間中にシステムの設定が現在どのようになっているのかを知るのが難しかった。ベンチマークを行っていたので、システムのバンド幅などの情報を簡単に確認できるとよかった。
- (1-2) 他のクライアントマシンの鍵追加に関して
  linux マシンは簡単にできたが、windows マシンはできなかった。あまり必要ではなかったのでそのままにしてしまった。
- 今回、HA8000 システムを利用させて頂く機会を与えて頂きまして誠にありがとうございました。
  予想していたよりシステムが安定していました。これだけの規模のシステムを安定して運用するのは本当に大変なことだと思います。関係者の皆様のご苦労に敬意を表します。
- 柏キャンパスに在籍しているのですが、ファイル転送が速くても 10MB/s くらい、遅い時には 1MB/s もでません。 まるで自宅の ADSL のようです。
  どのレベルに問題があるのかはわかりませんが、少なくとも SR11000 で srft を使った際には 30MB/s 近くでていたので、柏<->基盤センター間の回線の問題だけではなさそうです。
  本郷キャンパス内の人がどの程度速度がでているのかわからないですが、現状では柏キャンパスからテラバイト規模のデータを転送するにはちょっと不足していると感じます。
  改善が容易でないことは承知していますので、要望というよりは現状報告と受け取ってください。
  これからもお世話になると思いますのでよろしくお願いいたします。
  (10/10 追記) 本利用が始まった今月からは 10~20MB/s くらいでるようになりました。
  ハードウェア的に改善がなされたのか、単にユーザー数が減って帯域に余裕がでたのかわかりませんが、このくらいの速度がでると実用上問題なさそうです。
- コンパイル、プログラム実行時にエラー等のトラブル及びそれに対する解決策（解決しているのか？していないのか？）を集めた Q&A があればよいと思いました。私は以前、学生の時、北大に所属していたのですが、北大の大型計算機センターの HP に"利用者の皆さんからのご質問（Q&A 集）"（URL: http://www.hucc.hokudai.ac.jp/~a10020/www/qanda.cgi）というのがあり、便利でした。

他のユーザーがどのようなトラブルにあって、どうやって解決しているかを知るよい方法の一つだと思います。これに対応するような、なんらかのシステムがあるとありがたいです。

- Compiler option '-parallel=n' の n の値により、n が大きいと正しい結果を出さない（収束判定がうまくいかない）。n が小さいと正しい結果が得られるが、それでは SR11000 に比べて早くないのでメリットがない。また、ゆっくり症状や原因を追求する時間もなかったため、まだあまり使っていません。
- メモリバンド幅が不足しており SR11000 に比べ計算時間が 3～5 倍になってしまったため現時点での移行は見送る事にしました。現在一部計算では MPI による並列化を行なっていますが、今後更なる並列数の増加、並列化効率の改善が進み HA8000 での計算時間が SR11000 と同等になれば移行する事も視野に入れております。やはり HA8000 の 100GB のディスク容量は非常に魅了であるので出来るだけ早い段階で移行できればと思っております。
- HA8000 の場合、並列化しないと使用するメリットがないので、MPI、OMP などの並列化手法について、少し詳細なコンパイル方法（インライン展開、OMP でのネスト解除方法）、実行性能向上方法をマニュアル化して欲しい。
- 若手利用者推薦制度に採択されたため、継続利用を取り止めました。
  講習会で細かい点も分かりやすく教えていただき、利用する際にたいへん参考になりました。
  長期休暇中の方が雑務がなく研究に専念できるため、お盆の時期に長期に渡りサービス休止になっているのが不便でした。
- ①Job が終了する前に中身を見たい file が Job が終了するまで見えないという問題があった。
  ②HA8000 から地震研究所までの通信速度が 1MB/s 程度（地震研内部だと 30MB/s）と遅いので、計算結果を sftp で移すのに時間がかかる。学内（せめて本郷・弥生地区内だけでも）の通信速度の向上を図っていただけると大変助かる。
- ①C++ で書いたプログラムが非常に簡単なものであっても、日立製コンパイラでコンパイルすることができません。改善されることを要望します。
  ②ファイルシステムに関連するコマンド(ls、cp など)のレスポンスが非常に遅いです。改善を強く要望します。
  ③システムが落ちる回数を減らしていただけると大変うれしいです。（これは上記 2 と同様、共有ファイルシステムまわりが原因だと推測しておりますが。。。）
- 1 コアあたりにひとつのプログラムを動かすことがメインだったが、SR11000 に比べて劇的に遅かった。メモリバンド幅が原因だったのかもよくわからないまま試用期間を終えてしまった。
- short キュー、parallel キューの打ち切り時間はそれぞれ 2 倍程度が良いのではないかと思います。また、割り当てノード数的には short キューを少し減らして、parallel キューをその分増やす方がプロダクション・ランの効率が上がるような気がします。
  このような形でユーザーの意見を反映する姿勢は素晴らしいと思います。
- 現在使用しているプログラム中に ACML ライブラリを利用しています。
  そのため、本システムについても利用できるかどうかの問い合わせをしたところ、このシステムから利用できるとのことでした。ただ、後ほどの問い合わせの際、各自でインストールしなければならないとのことで、指示通りに行いましたが、コンパイルの時点で上手くいかなかったため本システムの使用を断念してしまいました。
  上記(1-1)で記したシステムと使い勝手が違うからという理由からかもしれません。
  知識不足の感も否めず誠に恐縮ではございますが、導入を御検討ください。
- センターへの意見：大規模計算を促進するという HA8000 の趣旨はわかりますが、それにしても、キュー構成が超並列計算向けに偏っている気がします。「多くの人に手軽に使ってもらう」というのもオープンスパコンの意義ですので、1 - 2 ノードで RUNLIMIT が多いキュー（または別システム）を設けてもいいのではないでしょうか。

- 質問(2-3) 日立製コンパイラでは NAMD をコンパイルできなかった。
- ノードの指定ができないのでどう対応してよいのかわからなかった。
  あと待ち時間が割と長いというのが気になった。待ち時間が長いがジョブは 1 時間しか走らせることができないのでちょっと不公平に感じた。
- 特に昼間によく高確率で発生する事象として、ユーザーシェルで、たとえば ls コマンドを叩いて改行キーをうってからファイルのリストが表示されるまでに 10 秒はザラであり、酷いときには 60 秒近くも応答が帰らない状態がずっと続くなどが発生していた。（% time ls <CR> などとして確認）また、vi エディターを抜けてファイルを書き出しが終るまでにも、かなり待たされることが良く起きる。その前後ではエディター自身の応答が止まるということは起きていないが、ls コマンドの応答が極めて遅れていたりする。つまり、CPU やネットワーク経由による応答の遅延ではなくて、個人のホームディレクトリのファイルやディレクトリー情報、つまりファイルシステムの情報のファイルサーバからの読み出し、書き込みが極めて性能が悪いのである。ファイルサーバ（NFS？）の設定に問題があるのではないか？ たとえば、別のユーザーが巨大ファイルを読み書きしているあるいは別のユーザーがノード間でデータの大量通信をしていると、大して CPU を食っていないジョブスクリプトやソースのエディットをしているユーザーのホームディレクトリのファイルへのアクセスが帯域を奪われて、レスポンスを失ってしまうのではないか？ システムの構成は当方の関与するところではないが、たとえばユーザーのホームで巨大なファイル I/O をしない作業場所のファイルシステムと、超大規模ファイル等を読み書きするテンポラリのファイルシステムはサーバー上で同居しているのだろうか？
- 継続利用させていただく予定です。９月上旬の講習会もわかりやすくて、良かったです。
- 使用目的がパソコンで使用している fortran で記述したプログラムを高速にまたは､大きな系で計算する程度でした。普段使用しているインテルのコンパイラで動いたものが使えなかったので修正の手間をかけるよりは使えるようになるまで待って検討したいと思います。
  結局 Gaussian を中心に使用させていただきましたが大容量のメモリを生かした系を計算しようとすると 24h では短く計算を再開させるのが面倒でしたので。
  その他使用方法や演算性能等については特に問題ないと思います。
- ①フロントエンドで ls コマンドを実行した際のレスポンスが極めて遅い。
  ②他のスパコンに比べてファイルの書き出し、読み込みに時間がかかる。
- ログインノードからホームディレクトリへのファイルアクセスが非常に遅いので、コンパイル、デバッグの際に多大なストレスを感じた。システム全体としてファイルサーバの遅さはバランスを崩していると感じた。
- 研究予算の関係で継続利用しなかった。情報基盤センターの HP は、他の共同利用スパコンの HP に比べて若干見劣りするような気がします(必要な情報は書いてありますが)。
- フロントエンドの性能が低いのかコンパイルにすごく時間がかかるのをなんとかしてほしい。
- ログインノードが混んでいる時が多かったので、ログインノードを増やして欲しい。
  また、負荷が小さいノードを探すのが面倒なので、負荷が小さい、またはログインしているユーザー数が少なノードに自動的に接続されるようして欲しい。
- ターミナルのレスポンスが遅いので、かなり使い辛いです。
  ls コマンド等、表示されるまでに数十秒かかりますので、スムーズな作業のためにも改善をお願いしたいと思います。
- ライセンスの要るソフトウェアの実行ではまっている。
- 中島先生の講義（並列計算プログラミング、先端計算機演習）で使用しただけなので、該当する質問のみ回答させていただきました。
- 今回の利用は、アプリケーションの開発、実行ではなく、接続の確認とコマンド実行の確認だけだったため、アンケートのほとんどの項目にお答えできずにすみません。今後はアプリケーション実

行なども行う予定です。

ご質問に付きましては、今後、本センターの Web ページ上の FAQ 等にて、回答させて頂きます。また、ファイルシステムが遅いというご指摘を多くの方から頂きましたが、改善のために現在検討を重ねております。

以上のアンケート結果をもとに、さらに運営形態の改善を行っていく予定です。
ご協力、ありがとうございました。

以上