

大規模計算のマイルストーン

宮下 夏苗

情報社会基盤研究センター

概要

ナノマテリアルから地震, 天体のシミュレーション, マシンラーニングやビッグデータ処理など, 並列・大規模計算は様々な分野で必須の技術である. JAIST の並列計算機環境は学生, 研究者が初めて大規模計算を知り, 手法を学び, 計算機を使いこなしてより大規模な計算実験も実現できるプロフェッショナルの研究者への道を歩むためのマイルストーンとなることを目指している.

技術職員の業務の一つは, 学生, 研究者らが実際の大規模計算機に触れ, 学ぶために必要なリソースを準備し, 提供することである. 本稿では 2015 年度に行った大規模計算機関係のコンテンツ提供, ユーザサポートと今後の課題について列記する.

1 講習会

並列計算機初心者, 中級者を対象とした講習会を企画, 開催した.

2015 年内には以下の講習会を実施した.

6 月

MATLAB 入門セミナー

GPGPU/OpenACC プログラミング初級者講習会

Cray XC30/MPI Parallel Programing 初級者講習会

SGI AltixUV1000/Parallel Programming 初級者講習会

並列計算機利用者オリエンテーション

7 月

MATLAB+PC Cluster ミニセミナー

10 月

Altix + Intel Itac 講習会

Cray XC30 中級者講習会

Materials Studio 相談会

11 月

HPCC + Java 講習会

2 利用問い合わせ対応(11 月以降分抜粋)

並列計算機に関して, 以下のような問い合わせに対応, コンパイル方法の調査, インストール作業等を行った.

2.1 2015 年 11 月

GPU ノードで IB 経由の OpenMPI

Lammps コンパイル方法

2.2 2015年12月

UV1000, PC クラスタ上での Gaussian09, GaussView 利用方法
UV1000 に VNL-ATK 導入とモジュール作成

2.3 1月

PC クラスタ上での Python 利用方法
XC30 での Wien2K コンパイル方法

2.4 2月

XC30 での TurboRVB コンパイル方法

2.5 3月

XC30 での MKL を含むコンパイル方法

3 新規導入機器の動作確認, サンプル作成

2016年3月, UV3000 の導入にあたって, 日本 SGI と協力してユーザが並列計算機 SGI UV3000 を利用するための動作確認, ユーザ向け利用サンプルの作成を行った.

3.1 サンプル作成

ジョブ実行のため, 以下サンプルのバッチスクリプト, 利用メモを用意した.

- Gaussian09 用
- MPI+OpenMP 用
- Non-MPI(Single)ジョブ用
- Python 用

4 ユーザ向けライブラリのサポート

研究室で利用されている以下ライブラリをコンパイルし, 並列計算機上での利用方法を調査した.

4.1 OpenCV

4.2 Graphillion

5 JAIST festival

2015年10月10日に開催された JAIST フェスティバルにおいて, 情報社会基盤研究センターの展示として, PC クラスタを利用し計算の並列数で性能の違いを体験するデモンストレーションを準備した.

具体的には OpenCV を用いた顔認識を行い, 現地に設置したカメラからリアルタイムに取得した動画を用いて, 画像内の人間の顔と認識される部分に丸を描いて出力する計算をマルチコアで動作させた.

2 コア 2 並列と 10 コア 10 並列での動作を二つのディスプレイに並べて表示し, 来場した客が動作速度の違いを視認で



図 1: リアルタイム顔認識展示会場

きるようにした。

6 課題

次項のような点を現在の課題と考えている。

6.1 マニュアルの不足

大規模計算機に触れようとするユーザにとって、コンテンツ(マニュアル)の不足は大きな障壁になる。

しかし各種計算機に合わせた利用マニュアルを用意するにはコストがかかり、更新もままならない。納入ベンダー各社はとりあえずプログラムを動かすために必要な知識を提供してくれるが、A 機器で利用していたプログラムを B 機器でも動作させるためにはどうするか、などのような環境に依存する問題には対応しにくく、またユーザが利用するプログラムも千差万別なため、一から十までベンダーや情報センターがサポートできるわけではない。

こうした問題を解決するため、担当職員である自分自身が第一のユーザで有り続け、ユーザが引かかるポイント、判りにくい点を自ら見つけて、ポイントを押さえたマニュアルを用意し、更新していく必要がある。

6.2 システムの統一化

同じ計算を実行するのに、あのシステムではこのオプション、このシステムではこのオプションと、本質的でないオプションの相違が発生している状況を可能な限り回避したい。計算機を代わるたびに、マニュアルを開いてオプションの意味を確かめてバッチファイルを修正する、というのはできるだけ行いたくない作業である。

しかしデフォルト設定で納入された機器で、しかもそれがまったくアーキテクチャの違う機器であれば、相違が発生するのも無理はない。

これをできるだけ避けるためにはあらかじめシステムについて知り、調整し、どのように設定すれば各機器の利用環境が近くなるか、無意味な相違点を作らずシンプルに利用できるかを、一システムのみならず、すべての計算機において透過的に考慮して、判りやすい利用環境を構成できるような設計を考えておく必要がある。

6.3 インタフェースの統一

前項のように、システムによって実行時のオプションが異なる状況を回避するもうひとつの策として、統一インタフェースの作成を考えている。Web インタフェースから計算を投入し、オプションはバックエンドで(ユーザに意識させずに)指定させるというイメージである。

便利だが実装にコストがかかり、まだ実現に至っていない。

6.4 ユーザコミュニティ

以前は大規模計算を利用するユーザ同士のコミュニティがあり、上級者が初級者のサポートも行っていた。しかし現在、コミュニティを意識して他人のサポートまで行えるユーザは居ない。

わからなければ黙って利用を止めるか、情報センターに問い合わせるしか方法がなく、情報センター内にも計算機に対応する職員が数少ないためどうしても対応が遅れがちになるのが現在の大規模計算機環境の問題点である。ユーザはいるのだから、ユーザ同士で相互解決できるようなコミュニティの基礎を固められないかと考えている。

7 まとめ

学生、研究者が大規模計算を知り、学び、使いこなしてより大規模な計算実験も実現できるプロフェッショナルへの道を歩む道程をサポートすることを意図した計算機環境と、それを維持する学習、研究機関は、国内にほとんど存在しない。

これに対して JAIST では「パソコン」に近い PC クラスタ、これに続き、パソコンと同じプログラムも利用でき、プログラムを並列化することで更なる高速化も見込める「超大型パソコン」としての SMP 型の計算機、さらに、大規模計算の実行そのものに並列化が要求されるが、大規模計算の実行速度は世界ランクにも入る計算機の 3 種の計算機を、大規模計算の学びへの Hop-Step-Jump として維持し、ユーザに提供している。計算機分野の担当スタッフとして、一人でも多くのユーザがまずは大規模計算に興味をもち、Hop-Step-Jump の道程を踏み出してくれるよう、リソースを提供しサポートしてゆきたい。