課題番号 14-MD03

科学技術計算における効率の良い複数拠点利用とそれを実現するユーザ駆動型・拠点協調フレームワークの開発と検証

實本 英之（東京大学）

概要　シミュレーションと可視化を伴うもの、あるいはマルチスケールな構造をもった連成アプリケーションを対象とし、広域に分散した計算機資源を有効に活用するためのフレームワークの開発、検証を実施する。本研究は、 1）実アプリケーションを多拠点で利用した際の影響の検証および実行の効率化、2）多拠点を利用した連成計算アプリケーションを構成・実行するための導入障壁の低いフレームワークの構築、の 2 つから構成され、1）の結果をフィードバックしながら、2）を検討・開発を行った。

1.　共同研究に関する情報

(1) 共同研究を実施した拠点名

東京大学、東京工業大学、九州大学、北海道大学

(2) 共同研究分野

- □　超大規模数値計算系応用分野
- □　超大規模情報システム関連研究分野

(3) 参加研究者の役割分担

拠点毎の分担は以下、さらにメンバーの名前と役割を挙げる。

<u>東京大学（システム設計及びアプリ検証）</u>

實本英之（代表）：研究統括

システム設計・構築

松本正晴：アプリ検証

中島研吾：アプリ提供

片桐孝洋：アプリ情報提供

奥田洋司：アプリ情報提供

<u>九州大学（アプリ検証）</u>

小林泰三（副代表）：アプリ検証と提供

システム設計

<u>東京工業大学（研究環境構築・整備）</u>

三浦信一：RENKEI-VPE 環境調整

奥野喜裕：アプリ情報提供

佐藤仁：ストレージ情報提供

<u>北海道大学（研究環境構築・整備）</u>

棟朝雅晴：検証環境調整

<u>理化学研究所（システム設計補助）</u>

滝澤真一朗：システム設計

2.　研究の目的と意義

本研究は、広域に分散した計算機資源を有効に活用するためのフレームワークの開発、検証を実施するものである。対象としては、シミュレーションと可視化を伴うもの、あるいはマルチスケールな構造をもった連成アプリケーションを対象とし、これを構成する複数のアプリケーションを異なったサイトの計算資源を利用して実行する。また、学際大規模情報基盤共同利用・共同研究拠点を構成する各センターの持つソフトウェアライセンスや、それぞれに分散配置されたデータによる実行制限にも対応する。このため、本研究は大きく分けて、 1）実アプリケーションを多拠点で利用した際の影響の検証および実行の効率化、2）多拠点を利用した連成計算アプリケーションを構成・実行するための導入障壁の低いフレームワークの構築、の 2 つを目的とする。特に、広域分散計算や連成計算の必要条件と要素技術の検討と整備に焦点を合わせる事により、汎用性を備えたフレームワークの基礎を固める。具体的には、A）ロバストかつ柔軟な通信フレームワーク、と B）アプリや通信を管理するフレームワーク、の設計と構築

を行う。構築に当たっては、ユーザ権限と ssh といった一般的に利用可能と考えられるログインサービスのみで、連成計算が行える様なフレームワークをめざす。これは、新たなサービスを各拠点に導入するには、保守や安全性の検証といった面を含めた高コストな手続きが必要で、簡易に拠点案連携を行うにはすでに存在するサービスで構築する必要があるためである。利用される環境としては図 1 のような環境を想定している。また、連成されるアプリケーションは、単方向の通信を持つものを仮定する。

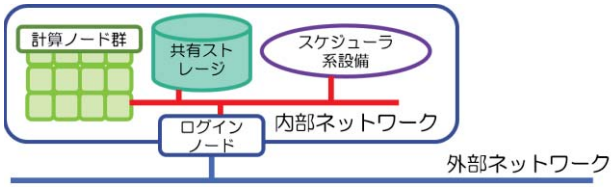

図 1：想定する実行環境

3. 当拠点公募型共同研究として実施した意義

拠点間連携システムの構築、および拠点間アプリケーションの特性を検証するにあたり、実際に多拠点が SINET で結びついた環境を利用できることは大きな利点である。

また、各拠点の計算資源のネットワークへの接続方法、利用法などの知識の提供、さらに構築途上のシステムテストのためにある程度検証に向いた環境設定を各拠点で行える研究者との共同研究は必須であり、運用システムと研究者が密に連携可能である本公募型共同研究によりこれが達成された。

実際に、本共同研究の資源である RENKEI-VPE 環境は、小規模な共同研究で構築することは難しいが、本研究課題の核心となる評価環境として大きな支援を得ている

4. 前年度までに得られた研究成果の概要

なし

5. 今年度の研究成果の詳細

中間発表においては以下の成果を報告した。これは各グループの成果報告前半に記述した。

1) 連成計算アプリケーションの多拠点連携処理検証

2) OpenFOAM による拠点間連携アプリケーションの設計と試験実装

2) 拠点協調フレームワークの設計

また中間発表以降の成果は、以下である。

1) OpenFOAM による拠点間連携アプリケーション実装の洗練

2) 拠点協調フレームワークの実装および追加設計

アプリケーション検証グループ

フレームワークの実装・構築に先立ち、拠点間連携を行う既存手法である「HPCI 先端ソフトウェア運用基盤 分散環境ホスティングサービス」を利用した連成計算アプリケーションの評価を行った。図 2 にその概要を示す。分散環境ホスティングサービスは、複数拠点に存在する仮想マシン（以下、VM）ホスティングリソースに対して、VM の実行管理を利用者主体で行えるシステムであり、HPCI の枠組みの中で提供されている。また、連成計算アプリケーションとして、3 次元熱伝導方程式の有限差分法（FDM）解析コードに可視化処理用のルーチンを導入したもの用いた。このコードは FDM 解析部と可視化処理部がそれぞれ特定の MPI プロセスで分割される。すなわち、各 MPI プロセスと各拠点の VM を対応させることによって、FDM 解析部と可視化部をそれぞれ別拠点で実行することができる。

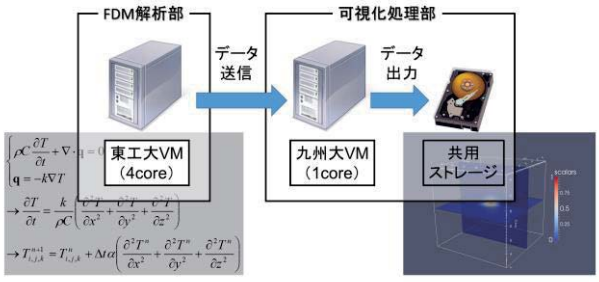

図 2：連成計算アプリケーション評価の概要

本評価では簡単のため、全プロセス数を 2 とし、

FDM 解析部（rank = 0）は東工大で起動した VM（4 core）1 台を用いて OpenMP によるスレッド並列化により計算される一方、可視化処理部（rank = 1）は九州大で起動した VM（1 core）1 台を用いて、可視化のためのデータを必要に応じて東工大 VM より受信し、共用ストレージへと書き出す。FDM 解析部では、熱伝導方程式の空間微分を 2 次精度中心差分で評価し、1 次精度オイラー陽解法で時間積分を行っており、全格子点数は 64×64×64、時間発展のためのメインループの iteration 回数は 1000 とした。可視化処理部は FDM 解析部のメインループ内で呼ばれるが、その際、全格子点数分の温度データの送受信をノンブロッキング通信で行う。

図 3 に、データ出力間隔に対する実行時間を両対数グラフで示す。ここでデータ出力間隔とは、「FDM 解析部メインループの iteration 中で何回に 1 回、可視化処理部を呼び出すか」を示す値であり、例えばデータ出力間隔＝10 であれば、FDM 解析ループ 10 回に 1 回だけ可視化処理部を呼び出し、データを出力する、ということを表す。同図より、データ出力間隔＝100 を境に実行時間が大きく異なることがわかる。データ出力間隔を 100 以下に設定すると、FDM 解析部の実行時間に対して可視化処理部の実行（MPI 通信）に時間がかかることで、全実行時間が大幅に増加する一方で、データ出力間隔を 100 以上に設定すると、可視化処理部の実行が FDM 解析部の実行時間に隠ぺいされる傾向にあることから、実行時間がほぼ一定となる。

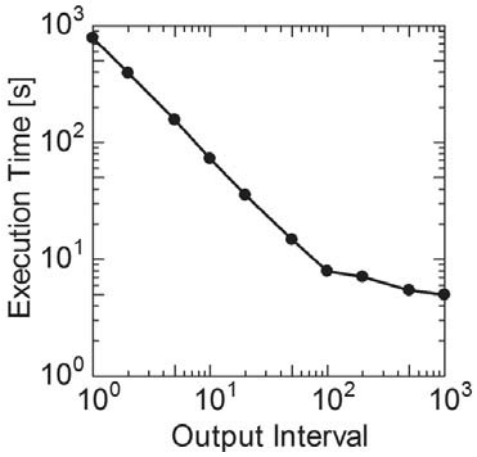

図 3：データ出力間隔に対する実行時間

以上の検証を元に、適切なデータ出力間隔を設定した上で、1)東工大 1 サイト(4core) と 2)東工大＋九州大(5core)で FDM 解析部および可視化処理部を実行した。上記の検証と同じく、並列化数は、FDM 解析部 4 並列、可視化処理部 1 並列のため、1)ではサイトの能力以上の並列処理が走るようになる。この結果、1)では 8 sec.、2)では 9 sec. の実行時間となった。この結果は適切なパラメータ設定が行われた場合、多拠点連携処理に効果がある可能性が示唆される。この結果は、システム設計の検討材料とした。

OpenFOAM を対象アプリケーションにしたポスト処理連携については、MPI 通信を扱うクラスの拡張設計に目処をつけて実装を進め、試験を開始した。基盤センターの資源を用いた大規模な試験は実施できてはいないが、研究室設置の PC クラスター間での疎通テストはほぼ完了した。その成果を国際会議 PANACM2015 にて発表した。

中間発表以後の成果としては、大規模検証に備え、昼間発表時に残っていた、再現性の無い不具合、あるいは実行時間の変動などについて、原因究明をよび解決を行っている。

システム設計グループ

本研究で提案するユーザ駆動型・拠点協調フレームワークは、連成アプリケーションを多拠点実行する際に問題となるものとして、1)アプリケーション間メッセージをどのように最適に送受信するか、2)各拠点の基盤システムの差異を埋めながらアプリケーションを多拠点に同時投入しているように見せかけるにはどうすれば良いかに注目してフレームワークを設計した。

本設計(図 4)では、アプリケーション間メッセージは、ゲートウェイノードを通して、他サイトに転送する。ゲートウェイノードには PoP サーバと呼ばれる、メッセージ通信、ジョブ投入を管理するプロセスを実行する。

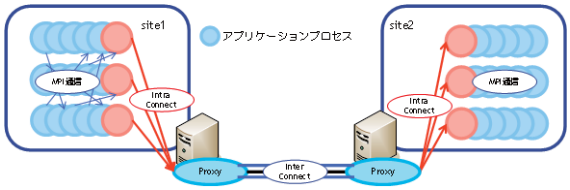

図 4：メッセージの流れ

設計に当たり考慮した要件、およびその理由は以下の通りである。

サイト間ネットワーク(Inter connection)

1.1) 大多数の計算センターで利用せざるをえない Well-known サービスのみで情報のやりとりを行う

目的と意義において述べたとおり、運用ポリシ設計や合意といった導入コストを下げる。

1.2) 数拠点での連携に耐えるスケーラビリティ

性能は実際の大規模計算拠点数および連成プログラムの構造から、数拠点での連携に耐えられるスケーラビリティで充分に担保可能である。

1.3) アプリケーションにおける通信時間と計算時間の釣り合いに関して対応可能

通信時間と計算時間の兼ね合いはアプリケーションに応じて違う。超過してしまった通信時間に対し、計算を中断するか通信を間引くか、あるいは小さすぎる通信量にたいして複数メッセージのバッファリングを行うかといった戦略をアプリケーション毎にまた、実行環境毎に変更する必要がある。

1.4) PoP サーバの想定外の停止に耐える

ログインノードは多くのユーザで共有されているため、プロセスが実行可能な時間がサイトポリシによって決められている。サイト間メッセージは不意のサーバ停止に対して喪失・重複を起こさない必要がある。

サイト内ネットワーク(Intra connection)

2.1) 計算、ログインノード間の通信として直接通信もしくは共有ストレージによるデータ共有を利用可能

運用ポリシや利用法によって情報共有手法として利用できないことがあるため複数の手法が選択可能である必要がある。

2.2) 計算プロセスからの通信に耐える高いスケーラビリティ

結果ファイルを出力する代表プロセスを選出する場合、全てのプロセスから情報を集約してくるため、同期処理が発生し、アプリケーションのボトルネックとなってしまう。これを避けるためにプロセスそれぞれが結果を出力する可能性が高い。

2.3) 同じプロセスから送信された情報の送信順の保持

プログラミングを容易にするためであり、並列プログラミングに広く利用される MPI においても同様のポリシとなっている。

連成ジョブのスケジューリング

3.1) 各拠点においてジョブが同時投入されなくても連成が可能

各拠点にジョブスクリプトを同時に投入したとしても、実行キューの待ち時間は想定が難しく、それらが同時に実行状態になる保証は全くない。また、拠点それぞれが定めるポリシに従い、アプリケーションの実行時間には最大値が決まっており、他拠点で実行されるアプリケーションを無限大に待ち続けることは不可能である。

この要件に基づき、中間発表以降の成果として以下の実装を行っている。

PoP サーバ

図 4 で示した通り、各拠点のログインノード上で動作するプロセスで、メッセージの管理とスケジューリングに関する機能のほとんどをこのプロセスで実現する。

要件 1.1) に対応する為、SSH トンネリングによりサイト間ネットワークを構成する。手順としては、送信元サイトから連携先サイトへ SSH を用いてリモートトンネリング経路の確立と、PoP サーバの立ち上げを行う。連携先で立ち上がった PoP サーバは元サイトへの TCP/IP 接続を行う。これにより、連携先サイトでは送信元サイト数に応じたローカルポートが消費されるが、要件 1.2)によりこれは充分小さいと考えられる。

さらに、要件 2.1) に対応するため、サイト内ネ

ットワークに関してはコネクションを仮想化している。仮想化コネクションの操作は TCP/IP ソケットに類似したインターフェースで行っており、通信プロセスからのメッセージ受信に応じたイベントドリブンな手法をとっている。これには libevent2 ライブラリを利用している。現在は要件 2.2) を容易に満たすために、共有ストレージを用いた手法を優先実装している。共有ストレージは多数のプロセスからの同時アクセスを想定しているため、通信をファイル化することによりこのスケーラビリティを通信に波及させることが可能である。また、要件 3.1) を実現するために、連携先サイトで、ジョブの実行にかかわらず通信内容を保存しておく必要がある。これは現在優先実装している共有ストレージによる通信のファイル化によって容易に達成できる。現実装では、通信用共有ファイル内の既読メッセージ消去はファイル単位で行い、メッセージ単位での消去は行わない。これは環境によって可否があり、また大きなコストがかかるためである。

メッセージには、送信元、受信先を示すヘッダが付与され、送信元では送信プロセス毎、連携先では、受信プロセス毎にファイル化される。これにより、要件 2.3)が達成されている。

ジョブの投入は送信元側で起動した PoP サーバに設定ファイルを渡すことによって行われる。この設定ファイルでは、拠点毎のジョブ投入スクリプトファイル名や1.3)を実現するための設定項目が含まれている。

最後に現在実装計画の段階として、要件 1.4)に対応する為、他サイト PoP サーバとの通信前に、必要に応じて PoP サーバを再起動する。また、メッセージ喪失・重複に対応する為、ACK 送信を用いた図5 のプロトコルにより送受信を行う予定である。

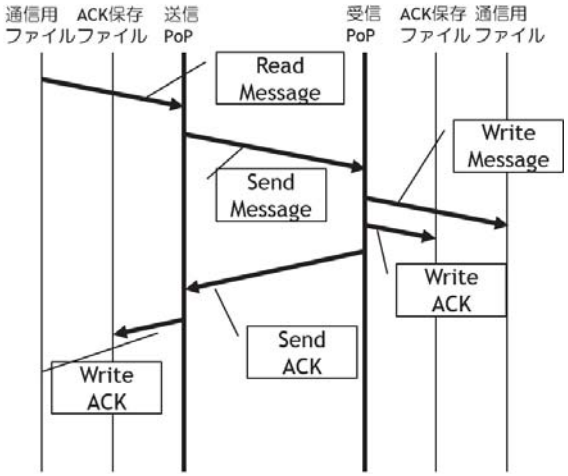

図 5：メッセージ送受信プロトコル

異常終了を検知し、終了されてしまった PoP サーバは再起動を行った後、他方の PoP サーバに現在の ACK 状況を問い合わせる。また、自分が保存しておいた ACK 保存ファイルを読み込み、自分の ACK 送出状況とする。次に送出するメッセージを特定するためには以下のパターンを用いる

a) 送信側 ACK と受信側 ACK が一致する場合

特に対処は必要とせず、次の未 ACK メッセージを送出メッセージとする。

b) 送信側 ACK と受信側 ACK が一致しない場合

受信側から ACK が送信側に戻っておらず、メッセージが破損している可能性があるため、破損メッセージリストに ACK を送出していないメッセージのメタデータを保存する。さらに、通信用共有ファイルを閉じ、新たに新しいファイルを通信用共有ファイルとして作成する。受信側のアプリケーションは破損リストを見ながら、該当メッセージを確認した場合、それを破棄、新しい通信用共有ファイルから再度読み込み始める。

アプリケーション間通信 API ライブラリ

送信プロセスと受信プロセスを識別するタグを設定可能なブロッキング P2P 通信 API(Send/Receive)である。識別タグに関しては、送信プロセス内、受信プロセス内でそれぞれ重複のない整数値をアプリケーションプログラマが任意に利用する。これをヘッダとして付与することで、PoP サーバでの適切なメッセージ振り分けを行う。多くの場合、連成されるアプリケーション

は MPI によって実装されていると考えられるため、それぞれにおけるランクを利用すると重複がない状態が実現できる。要件 2.1)を満たすため、PoP サーバ同様コネクションを仮想化しており、現在は共有ファイルシステムを利用する手法を優先実装している。

本フレームワークの利用手順は以下である。
1)それぞれのアプリケーションを拠点間 P2P 通信 API により改変する。
2)PoP サーバの設定ファイルを作成する。これには連携させる拠点のログインノード名や各拠点で実行するジョブ投入スクリプトファイル名、アプリケーションの依存関係を記述する。
3)拠点毎で実行されるジョブ投入スクリプトを用意する。
4)依存関係に従って最初に実行するアプリケーションを配置する拠点で作成した PoP サーバを起動する。
5)他拠点での PoP サーバ起動に伴い、ログイン ID、パスワードを入力する。

6. 今年度の進捗状況と今後の展望

進捗状況は、FDM アプリケーション側はフレームワーク設計につながる小規模検証を RENKEI=VPE 上で行った。システム開発側の遅れにより、予定していたフレームワーク上での検証が未実施である。

OpenFOAM アプリケーション側は基本的な実装を行い、疎通テストを実施することができた。

システム開発については実装が難航しており、今後行う予定であったログインノードにおけるプロセス強制停止への対処手法の設計を繰り上げて実施した。今年度目標に対する達成度合いは 80%程度であり、双方のアプリケーションにおける提案フレームワークを利用した検証とフレームワーク実装の一部が積み残しとなった。

今後の展望としては、システム側にて実装を大規模検証が可能な最低限まで早急に行い検証を行う。また、検証と並行して追加設計であるログインノードにおけるプロセス強制停止への対処を実装していく。また、大規模検証により洗い出された設計の不足分を実装に生かしていく。最終的にはオープンソースソフトウェアとして公開する予定である。

FDM アプリケーション側では、地震波動－建物振動連成解析アプリに対して、可視化アプリを作成し、計算・可視化間の連成アプリとしてアプリとフレームワーク双方の性能・精度評価、課題点の洗い出し等を行う。これは、これまでの研究において、3 次元熱伝導方程式の有限差分法（FDM）解析コードに可視化処理用のルーチンを導入したアプリによる連成解析を行った知見から発展した内容となる。最終的に可能であれば、地震波動－建物振動連成に使われている単一拠点用のカップラを改造し、3 拠点連成アプリケーションの実施を目指している。

OpenFOAM アプリケーション側では、今後は基盤センターでの大規模検証と、実際のセンターサービスとして一般利用者の利用を想定した負荷テストとユーザ利用のインターフェースの整理が今後の課題である。基盤センターでの大規模検証では、現在発生している再現性に乏しい実行時間の大きな変動の原因究明と対策を施すことと、センターの利用規約との整合性を詰める作業が必要である。この問題に関しては、RIST との打ち合わせを開始しており、「京」での利用を視野に入れて今後の設計・実装に反映させる体制を整えている。また、アプリケーションとポスト処理連携に関しては、PANACM 2015 での発表に対して、 in situ visualization への展開について、関心を寄せられた。可視化に関しては、研究対象と研究内容、さらには研究段階によって可視化の内容が変化するために、一般的なフレームワークとして定義するのが困難であるが、今後の研究展開として予定しているアプリケーションとポスト処理連携から連成計算に進む途中の段階で課題設定をする予定である。さらに、OpenFOAM アプリケーションにおいても,スケーラビリティとロバスト性を改善するために本研究課題で開発したフレームワーク

を利用する予定である。これらも基盤センターや「京」でサービスとして提供可能な品質の実現を目指す。

7. 研究成果リスト

(1) 学術論文 1 件

T. Kobayashi, T. Akamura, Y. Nagao, T. Iwasaki, K. Nakano, K. Takahashi, M. Aoyagi, "Interaction between compressible fluid and sound in a flue instrument", Fluid Dyn. Res. 46 061411, 2014

(2) 国際会議プロシーディングス

なし

(3) 国際会議発表 5 件

T. Kobayashi, Y. Morie, H. Jitsumoto, T. Takami, M. Aoyagi, "A New Bottleneck in Large-Scale Numerical Simulations of Transient Phenomena, and Cooperation Between Simulations and the Post-Processes", 3.17 MS: High Performance Computing and Related Topics I, 1st. Pan-American Congress on Computational Mechanics (PANACM2015), Buenos Aires, Argentina, 27-29 April, 2015

S. Iwagami*, G. Tsutsumi, K. Nakano, T. Kobayashi, T. Takami, K. Takahashi, "Numerical Analysis on the Lighthill Sound Sources of Oscillating Jet", Contributed Session on Advanced Methods in Computational Fluid Dynamics II, 1st. Pan-American Congress on Computational Mechanics (PANACM2015), Buenos Aires, Argentina, 27-29 April, 2015

T. Kobayashi, T. Iwasaki, K. Takahashi, T. Takami, M. Aoyagi, "A numerical simulation for a tone hole of flue musical instrument", XXXIV Dynamics Days Europe, P2-5, 8-12 September 2014, University of Bayreuth, Germany

T. Takami, M. Shimokawa, T. Kobayashi, "Temporal parallel approach to nonlinear problems with multiple time-scales", XXXIV ynamics Days Europe, P2-5, 8-12 September 2014, University of Bayreuth, Germany

K. Takahashi, T. Kobayashi, T. Akamura, Y. Nagao, T. Iwasaki, K. Nakano, M. Aoyagi, "Evaluation of acoustic energy generation and absorption in a flue instrument with Howe's energy corollary", XXXIV Dynamics Days Europe, P2-5, 8-12 September 2014, University of Bayreuth, Germany

(4) 国内会議発表 4 件

實本英之，小林泰三，松本正晴，滝澤真一朗，三浦信一，中島研吾，複数拠点利用を実現するユーザ駆動型・拠点協調フレームワーク，電子情報通信学会技術研究報告 CPSY2014-20 (SWoPP'14), Vol.114, No.155, pp155-159, 2014 July

岩上翔，堤元気，中野健一郎，小林泰三，高橋公也，「エッジトーンにおける流体音源の数値的評価」, 21aPS-20, 日本物理学会 第 70 回年次大会, 2015 年 3 月, 早稲田大学

堤元気，岩上翔，中野健一郎，小林泰三，高橋公也,「エアリード楽器でのジェットの安定性解析」, 21aPS-21, 日本物理学会 第 70 回年次大会, 2015 年 3 月, 早稲田大学

宮川矩昌，小林泰三，高橋公也，吉川茂，「流体音響シミュレーションによる曲管の音響解析」, 21aPS-22, 日本物理学会 第 70 回年次大会, 2015 年 3 月, 早稲田大学

(5) その他（特許，プレス発表，著書等）

なし