Vol.2014-HPC-143 No.2
2014/3/3

ジョブ間データ転送方式の検討

齋藤 智之[1]　石川 裕[1]　Gerofi Balazs[1]　三好 建正[2]　大塚 成徳[2]　富田 浩文[2]　西澤 誠也[2]
八代 尚[2]

概要：実時間ゲリラ豪雨予測システムを実現するために，100 ケースの 30 秒アンサンブル気象シミュレーション結果と 30 秒毎の最新気象観測データを同化し，その結果から 30 分後の気象予測をする．将来の並列計算機において，5000 プロセスから構成される気象シミュレーションジョブと 5000 プロセスから構成されるデータ同化ジョブの間でデータ転送が行われると見積もっている．ファイル渡しによる非効率なデータ転送ではなく，ファイル I/O API を維持しながら 2 つのジョブのプロセス間で効率の良いデータ転送を提供する File I/O Arbitrator を提案する．

Consideration of Data Transfer between Jobs

Tomoyuki Saito[1]　Yutaka Ishikawa[1]　Gerofi Balazs[1]　Takemasa Miyoshi[2]　Shigenori Otsuka[2]
Hirofumi Tomita[2]　Seiya Nishizawa[2]　Hisashi Yashiro[2]

Abstract: We are designing and developing an innovative real-time severe weather forecasting system that updates 30 minute later severe weather conditions every 30 second. In this system, the results of 100 cases of 30 second ensemble numerical weather simulations and observational data obtained by modern weather equipment's are assimilated every 30 minute, and 30-second weather prediction is performed using the assimilated data. In a next generation supercomputer we assume, it is estimated that data are transferred between 100 case ensemble simulations running on 5000 processes and an assimilation job running on 5000 processes in order to meet required realtimeness in terms of computation. Shortening the execution time of simulations and assimilation, the execution time of transferring data via files becomes bottleneck. In this paper, efficient data transfer middleware called file I/O arbitrator is proposed in order to eliminate exchanging files.

1. はじめに

科学技術振興機構 CREST「「ビッグデータ同化」の技術革新の創出によるゲリラ豪雨予測の実証」プロジェクトでは，2020 年までに，次世代型フェーズドアレイ気象レーダーで取得可能な 30 秒毎の 3 次元観測データや次期気象衛星ひまわり 8 号・9 号で可能となる 2 分 30 秒毎の日本周辺雲画像データを用いて，30 秒毎に更新する 30 分天気予報を実現する実時間天気予測システムの研究開発を進めている．本システムでは，観測データ受信ジョブ，並列実行ジョブ数 100 の 30 秒天気予測シミュレーションジョブ（アンサンブルシミュレーション），観測データとアンサンブルシミュレーション結果を同化するジョブ，同化結果を初期値とする 30 分天気予測シミュレーションジョブから構成される．

シミュレーションプログラムも同化プログラムもデータはファイルから入出力している．2 つのプログラムの負荷分散方法は異なるため，データ同化ジョブではデータをプロセスに分配するために，1 入力ファイルに対して 1 プロセスがファイルを読み込んでからデータを分散している．負荷分散のために計算ノード演算速度は今後 10 倍〜100 倍に向上してもネットワーク性能は数倍にしか向上しない．このためファイル I/O 処理およびデータ分散のための時間がボトルネックになっていく．

我々は，実時間天気予測システムのために必要とされるジョブ間でのデータ転送を効率的に実現するミドルウェア File I/O Arbitrator を設計している．File I/O Arbitrator はジョブ間で必要とされるデータを単純に転送するだけで

1　東京大学情報理工学系研究科
2　理化学研究所計算科学研究機構

なく，プロセスが保持するデータをそのデータを必要とする他のジョブのプロセスに直接転送する機構を提供する．データ生成側プロセスは netCDF API を変更せずに利用できる．データ転送先プロセス群や分散方法は実行時にコンフィグレーションファイル経由でミドルウェアに指示する．データ授受側プロセスはファイル I/O およびデータ分散せずにデータを受信することが可能となる．

本稿では，次節においてゲリラ豪雨予測のための実時間天気予測システムを紹介した後，第 3 節で想定計算機環境と必要とされる計算資源を見積もる．そして，第 4 節で，ファイル渡しによるデータ転送およびデータ配布ではファイル I/O 処理がボトルネックになることを既存システムでの実行結果からの外挿により示す．現在設計中の File I/O Arbitrator の概要を第 5 節で示す．第 6 節では，提案方式の有効性を確認するためにプロセス間で同様のデータ転送パターンを行った時の実行時間を既存システムで計測し，その結果から性能予測を外挿する．

2. データ同化による実時間ゲリラ豪雨予測システム

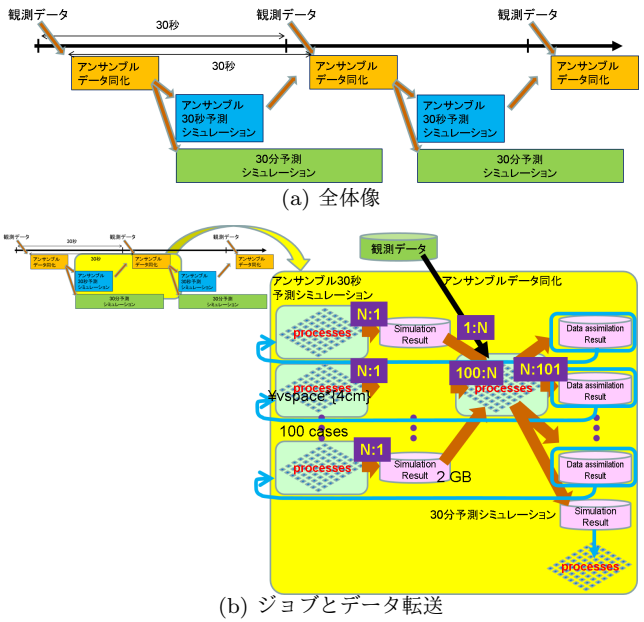
2.1 概要

実時間ゲリラ豪雨の予測には，数値計算による予測結果と高頻度・高解像度観測データを同化することにより高精細シミュレーションを行う必要がある．現在気象庁で実用化されているメソ気象予報では，実時間 3 時間毎，計算時間は 15 分程度であり，解像度は 5 km である．本プロジェクトでは，次世代型フェーズドアレイ気象レーダーで取得可能な 30 秒毎の 3 次元観測データや次期気象衛星ひまわり 8 号・9 号で可能となる 2 分 30 秒毎の日本周辺雲画像データを用いて，30 秒毎に更新する 30 分天気予報を実現する実時間処理システムの実現を目指している．

図 1 にシステムの全体像を示す．30 秒サイクルで，100 ケースアンサンブル 30 秒予測シミュレーション，100 ケース予測結果と観測データとの同化，データ同化された結果に基づく 30 分予測シミュレーションを繰り返す．このため，アンサンブルシミュレーションとデータ同化は 30 秒以内に終了しなければならない．データ同化結果は次のフェーズのアンサンブルシミュレーションの初期値として，また，30 分予測シミュレーションの初期値として使われる．

観測データとしては，次世代型フェーズドアレイ気象レーダーや気象衛星ひまわりのデータはインターネット経由で受信することを想定している．次世代型フェーズドアレイ気象レーダーから，30 秒間隔で反射強度，ドップラー速度，速度幅の 3 パラメータ (37.6 MB/parameter) が取得できる．

数値天気予報モデルとしては，理研で開発されている LES 気象モデル SCALE [1]，気象庁で現業的に運用されている NHM モデルも使う．データ同化には，局所アンサンブル変換カルマンフィルタ（LETKF）[2] を使用する．これらプログラムは独立して開発されており，入力データ，出力データはファイルに格納される．これらプログラムはいずれも MPI 並列化されている．



(a) 全体像

(b) ジョブとデータ転送

図 1　実時間ゲリラ豪雨予測システム

現在想定しているモデルサイズは解像度 100 メートルで，水平方向 50 km × 50 km，垂直方向 100 レベルを扱う．モデルが扱うパラメータ数は 16 であり，生成されるデータ量は，

$$500 \times 500 \times 100 \times 16 \times 4\,\mathrm{Byte} = 1.6\,\mathrm{GB}$$

となり，これらは一つのファイルとしてまとめられる．

2.2 データ転送パターン

データ転送の観点で気象予測シミュレーションコードを概観する．気象予測シミュレーションの並列化は領域分割され，各 MPI プロセスは担当領域を計算する．一つのファイルから各 MPI プロセスが担当する領域データが分配される．シミュレーション結果をファイルに格納するためには各 MPI プロセスが保持しているデータが集約されている．ファイルアクセスパターンを入力ファイル数：プロセス数：出力ファイル数で表現すると，気象予測シミュレーションコードは，1:N:1 アクセスパターンといえる．なお，SCALE ではファイル I/O ライブラリ netCDF [3] を使ってファイルアクセスしている．

LETKF データ同化コードは，アンサンブルシミュレーション結果と観測データを同化するが，各 MPI プロセスへの領域の割当には，計算負荷のアンバランスを考慮する必要がある．観測データは分布が均一であるとは限らなく，また，品質保証のためにノイズを除去する過程もある

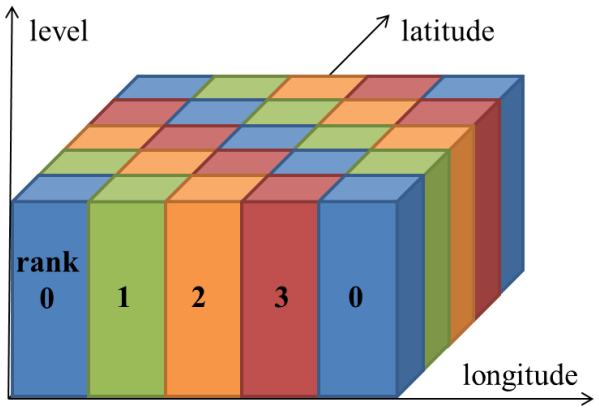


図 2　LETKF におけるデータ配布 (4 プロセス，水平格子 5 × 5)

ため，同化される観測データの分布は動的に変わる．同化処理において，観測データが少ない領域を扱っているプロセスの計算負荷は少なくなるので，シミュレーションと同じ領域の割当による並列化では，負荷がアンバランスになる．LETKF では，水平方向で隣り合っている格子における観測データ数には相関があるという考えに基づき，図 2 に示す通り，シミュレーション結果の水平方向の各格子をラウンドロビンで，別のデータ同化処理 MPI プロセスに割り当てる．

LETKF のファイルアクセスパターンは，気象予測シミュレーションコード同様，一つのファイルに対して 1:N:1 となる．本プロジェクトではアンサンブル数 100 のデータ同化，則ち，100 ケースのシミュレーション結果を同化するので，全体でのファイルアクセスパターンは 100:N:101 となる．出力ファイルが一つ多くなるのは，30 分シミュレーション用データを生成するからである．現在の LETKF はケースファイル毎にファイル I/O 処理するプロセスが割り当てられている．すなわち，100 ケースの場合には 100 のプロセスがデータ入出力に割り当てられる．また，観測データは分割されることなく全体データを各プロセスが保持する．

3. 想定システム環境と必要計算資源

我々が想定する計算機環境の概要を表 1 に示す．理論浮動小数点演算性能，B/F，インターコネクト性能は，それぞれ，京の 20 倍，0.5 倍， 2 倍としている．京コンピュータにおける実行時間から実時間ゲリラ豪雨予測システムに必要とされる計算資源を見積もる．

30 秒以内にアンサンブルシミュレーションおよびデータ同化を完了させるために，それぞれの処理をどのくらいの時間で終了させなければいけないか決めなければならない．ここでは，まず，それぞれ 10 秒以内で処理を終了させ，後の 10 秒をデータ授受に費やせると仮定して計算資源量を見積もる．

表 1　想定ハードウェア諸元と京コンピュータ

	想定	京
理論浮動小数点演算性能	2.56 TF	128 GF
B/F	0.25	0.5
メモリ容量	16 GB	32 GB
インターコネクト性能	10 GB/sec	5 GB/sec
トポロジ	6-D Mesh/Torus	6-D Mesh/Torus

京コンピュータ諸元 : http://www.aics.riken.jp/jp/k/system.html

表 2　必要計算資源量見積り

	必要資源量
ノード	5000
ノードあたりのメモリ	1.6 GB

3.1 必要資源量

アンサンブルシミュレーションおよびデータ同化に必要な計算資源量を表 2 に示す．計算量見積根拠を以降に示す．

3.1.1 アンサンブル 30 秒シミュレーションに必要な計算資源

SCALE を用いた水平解像度 100 m，32 × 32 × 300 領域の 1 秒シミュレーションは，京 1 ノードで 2 秒かかることが分かっている．これを基準モデルとする．京コンピュータにおいて同一サイズの 30 秒シミュレーションはその 30 倍なので，60 秒かかることになる．想定モデルは，500 × 500 × 100 なので，32 × 32 × 300 と比較すると，格子数は 81.4 倍であり，京コンピュータでの想定モデルの実行時間は約 4900 秒となる．想定計算機は京の 20 倍高速だが B/F を 1/2 にしており，気象シミュレーション性能はメモリバンド幅律速となるので 10 倍の性能とみつもる．則ち，想定計算機では， 4900/10 = 490 秒の実行時間となる．このサイズの格子数であれば経験上ストロングスケーリングするので，10 秒以内で実行を終了するためには分割のことも考えると 50 ノードを割り当てるのが妥当となる．この時，1 ノード辺り 100 × 50 × 100 領域を計算することになる．

必要とするメモリ容量について試算する．シミュレーションでは，500 × 500 × 100 領域に対して 16 個のパラメータを持つ．各パラメータが単精度で表現されていたとすると，1.6 GB 必要となる．経験上，この 50 倍のメモリ領域が必要となるので全体で 80 GB となる．50 ノード全体で 80 GB あればよいので，ノード辺り 1.6 GB となる．

100 ケースのアンサンブルシミュレーションを 10 秒以内に終了させなければならないので，全体で 5000 ノード必要となる．

3.1.2 LETKF に必要な計算資源

LETKF が必要とする計算ノードについては，現在，精査中であるが，アンサンブルシミュレーションと同程度の計算量が必要となる．ここでは，5000 ノードと見積もる．

表 3　1 ケース 1.6 GB データフロー 京を想定

	経過時間（秒）
netCDF ファイル書き込み	12.9
LETKF ファイル読み込み	11.7
LETKF データ並び替え 1	12.6
LETKF Scatter	0.4
LETKF Gather	0.4
LETKF データ並び替え 2	12.5
LETKF ファイル書き込み	13.0
netCDF ファイル読み込み	6.2
計	69.7

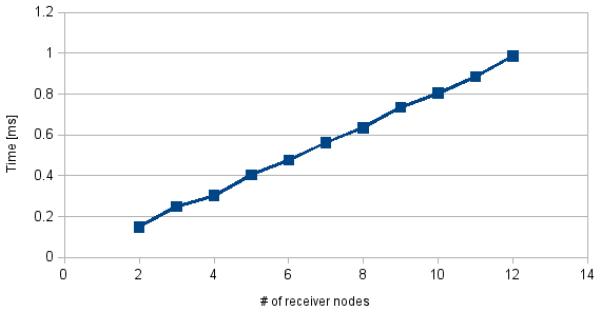


図 3　Scatter performance (320 kB / node)

4. 課題

京での実時間ゲリラ豪雨予測システムにおける，シミュレーション 1 ケース 1.6 GB のデータフローの File I/O 及び通信にかかる時間の見積もりを行う．前節で見積もった通り，気象予測シミューレションは 1 ケース辺り 1:50:1 のファイルアクセスパターン，データ同化は 100:5000:101 のファイルアクセスパターンとなる．現在のデータ同化プログラムの実装では，1 ファイルに出力されるシミュレーション結果 1 ケースに対して，1 プロセスが読み込みを行い，計算負荷のバランスを考慮して，データを並び替えた後，5000 プロセスに Scatter を行う．各プロセスはデータ同化処理を行った後，I/O を担当するプロセスに Gather し，またデータの並び替えをし，I/O プロセスが結果ファイルを書き込む．ここでは，シミュレーションの出力と LETKF の入力におけるデータタイプの変換にかかるコストは検討の対象外とする．

九州大学研究用計算機システム FX10 [4] を用いて，シミュレーション結果の書き込みから，次の時間サイクルのシミュレーションがデータ同化結果を読み込みまでの，各 File I/O 及び通信かかる時間を計測した．実験の結果，データフロー全体においてかかる時間の見積もりは 70 秒である．その内訳を表 3 に示す．見積もりにおいて，京と FX10 のインターコネクト性能，ストレージ I/O 性能は同程度であるとしている．

ファイル書き込み読み込み時にかかったデータの計測は，1.6 GB のファイルを 1 プロセスで書き込む，読み込むのにかかった時間である．気象シミュレーションモデル SCALE の現在の実装では，netCDF を用いてファイル入出力を行っているので，netCDF での書き込み読み込み時間を計測し，シミュレーションプログラムにおいて File I/O にかかる時間とした．

データの並び替えは，1.6 GB の読み込んだデータを並び替えながらメモリ上でコピーする時間を計測したものである．現在の実装と同じく 1 プロセスで，かつ 1 スレッドでコピーを行っている．計算を行ったシステムと京ではメモリバンド幅が違う．バンド幅の使用効率が同程度であると仮定し，最大メモリバンド幅の比をかけている．

1 ケース分を 5000 ノードに分配する実行時間は実験で使える計算ノード数が限られたため，小規模での性能から外挿した．1 ファイル $500 \times 500 \times 100 \times 16 \times 4$ Byte のデータは 5000 に分割配布されるので，各ノードが受け取るデータサイズは 320 kB となる．ノードあたり 320 kB を 2 から 12 ノードに Scatter するのにかかる時間を計測した結果，ノード数に応じて線形に実行時間がかかり，そのバンド幅は 12 ノードで 4 GB/s だった．受信ノード数 5000 の Scatter であっても 4 GB/s で送信すと仮定すると，1.6 GB の送信には 0.4 秒かかることになる．計測した範囲では Gather にかかる時間は Scatter より短かったため，Scatter と同程度の 0.4 秒と見積もっている．

表 3 に示したとおり全体では 70 秒かかる見込みとなる．

5. File I/O Arbitrator の設計

ファイル渡しによるシミュレーションとデータ同化は，本質的にファイルシステムにおいてボトルネックが顕在化することを前節で示した．並列ファイルアクセスをさせることで性能は改善されるが，それでもストレージへのアクセスのオーバヘッドがあり非効率的である．ファイルシステムのボトルネックを解消する方法としてシステムが提供しているファイルシステムを使用するのではなく，ユーザレベルでファイルサーバを実現する方法が考えられる．本システムでは 1 ファイル 1.6GB と容量が小さいで，それぞれのファイルを蓄積するファイルサーバプロセスを立ちあげる方法も考えられる．しかし，各プロセスのデータをそのデータを必要とするプロセスに直接転送する機構を提供して無駄な処理を削減することで，より効率の良いデータ転送を実現することが可能となる．

本システムはリアルタイムシミュレーションなので，100 ケースのアンサンブルシミュレーションジョブもデータ同化ジョブも毎回起動されるのではなく計算ノードに常駐して動作させるようにするのが妥当である．そのような状況であれば，データをファイル渡しにするのではなく，プロセス間でデータ交換をするほうがデータ転送にかかる時間

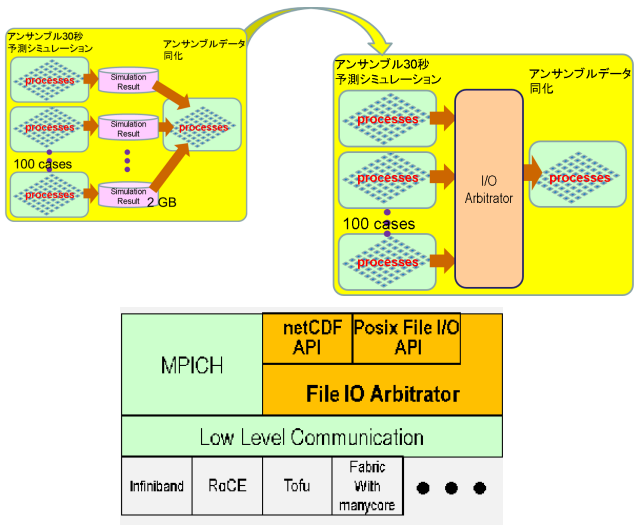


図 4　I/O Arbitrator

を軽減させることができる．

一方，2 つのプログラムは独立して開発されており，それぞれ他のプログラムとの連携も想定されている．このため，2 つのプログラムを大幅に変更するべきではない．SCALE はファイル I/O API に netCDF を使用しているので，netCDF ライブラリにデータ同化プロセスとデータ転送を実現する機能を組み込むことが可能である．LETKF データ同化アプリケーションは，ケース数と同数の I/O プロセスがファイルの読み込み，データ分散，データ集約，ファイル書き込みを行なっている．このため，データ分散・集約部分の変更は必要となる．

提案するミドルウェアの概要を図 4 に示す．I/O Arbitrator は，netCDF API からは透過性を保証する．則ち，各予測シミュレーションプロセスは netCDF でファイルに書き込むインタフェースを変える必要はなく，File I/O Arbitrator によって，LETKF の目的のプロセスへのデータの Scatter が実現される．また，データ配布パターンを別途用意する．

I/O Arbitrator は次世代システムソフトウェアスタックの一つとして設計している LLC(Low Level Communication Library)[5] 上に実装する予定でいる．LLC は RDMA 通信による one sided 通信を基本として，その上に two sided 通信機能等を構築している．LLC を使った MPICH 通信ライブラリの実装を進めている．LLC は，PC クラスタで主流の Infiniband ネットワーク，京や FX10 の Tofu ネットワークでの実装を進めており，I/O Arbitrator は広い計算機プラットフォームで利用できるようになる．

File I/O Arbitrator の下位通信層として LLC を想定しているが，MPI 通信ライブラリを使用することも可能である．この場合，アンサンブルシミュレーションプロセス群とデータ同化プロセス群は MPI_Spawn 機能を使って生成し，Inter Communicator を作成し通信させることにより実現可能である．しかし，将来的に File I/O Arbitrator は可視化システムとの連携など MPI 通信ライブラリが想定していない実行時環境での通信を提供することを考えているため MPI 通信レイヤを下位通信層としては使っていない．

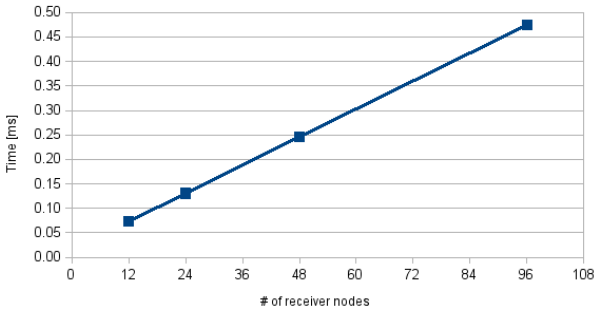


図 5　Scatter performance (6.4 kB / node)

6. 予備実験

シミュレーション側 1 ノードのデータを LETKF に配布することを考える．第 2.2 節で述べたデータ転送パターンに従うと，水平方向 100×50 の 5000 個の格子について垂直方向 100 レベル，16 パラメータのデータを LETKF1000 ノードに配布することになる．格子 1 個のデータサイズは，$100 \times 16 \times 4 = 6.4$ kB である．

実験で使える計算ノード数が限られたため，小規模での性能から外挿することにする．MPI で行う．データサイズは各受信ノードが格子 1 個と同量の 6.4 kB のデータを受信するように，$N \times 6.4$ kB とし，1 ノードから N ノードに対して MPI_Scatter する際にかかった時間を計測した．その結果を図 5 に示す．使用した範囲のノード数 12〜96 の間では，Scatter にかかる時間はノード数，データ総送信量に比例する結果となり，またそのバンド幅は 1.6 GB/s であった．受信ノード数 1000 の Scatter であっても 1.6 GB/s で送信すと仮定すると，1000 格子のデータの Scatter は 4.0 ms，5000 個の格子の送信には 5 倍の 20 ms かかる．

以上より，気象予測シミュレーション 1 ノードの担当領域 (32 MB) のデータを LETKF1000 ノードに配布するのにかかる時間は 20 ms であるという見積りを得た．

ノードあたり 6.4 kB の Scatter では 320 kB の Scatter と比べて，2.65 倍ほどバンド幅が小さい．また実際には複数のノードから Scatter を行うので，全体のバンド幅が高くなるよう，送信サイズについて考える必要がある．上記のデータ配布の仕方ではデータ配布前後のデータの並び替えをしていないが，送信サイズ変える必要があると，並び替えが発生する可能性があり，第 4 節で考慮したデータ並び替えのためのメモリコピーの性能とのバランスも考慮しなければならない．

第 4 節の 1 ケースデータフローからファイル I/O を除

いた場合，想定している計算機環境，メモリバンド幅 10 倍，インターコネクト性能 2 倍であれば，並び替えに合計 2.5 秒，通信に 0.4 秒かかる．本研究では輻輳による通信遅延を評価できなかったが，シミュレーションプログラムと LETKF プログラムのデータ授受を 10 秒以内に終えるためには，ファイル渡しによるデータ授受をなくすことが不可欠であり，また，それをなくすことで目標時間内にデータ授受を終えられる見積もりが得られた．

7. 関連研究

アンサンブルシミュレーションプログラムとデータ同化プログラムといった，元々は独立している複数の並列プログラム間のデータの授受を効率よく実現する手法として Larson, Jacob らによる Model Coupling Toolkit (MCT) [6], [7] がある． MCT では，独立した複数の並列プログラムコンポーネントと，Coupler と呼ばれるコンポーネントを，全て組み合わせた一つの複合プログラムを生成するためのソフトウェアツール群を提供する．複合プログラムでは，各コンポーネントでのデータの分割の仕方，コンポーネント間のデータの配布の仕方が記述され，Coupler がその記述に従いデータ配布を実現する．我々の手法は MCT と同様の目的であると言えるが，我々の手法ではミドルウェアで Coupler の役割を実現し，File I/O API を提供することで，各並列プログラムの修正を抑えることができ，また，コンポーネント間の直接データ転送もサポートするので，より効率の良いデータ転送が可能になる．

8. おわりに

ゲリラ豪雨予測のための実時間天気予測システムに必要とされるジョブ間データ転送ミドルウェア File I/O Arbitrator を設計し予備実験の結果を示した． File I/O Arbitrator は，アンサンブルシミュレーションジョブの各プロセスが保持するデータをそのデータを必要とするデータ同化ジョブのプロセスに直接転送する機構を提供する．これによりファイル I/O 処理ならびにデータ並び替えの処理を削減することが出来る．本発表での実験結果を踏まえて，今後，File I/O Arbitrator を実装し評価していく．

謝辞

本研究の一部は，科学技術振興機構 CREST「科学的発見・社会的課題解決に向けた各分野のビッグデータ利活用推進のための次世代アプリケーション技術の創出・高度化」領域のなかの課題名「「ビッグデータ同化」の技術革新の創出によるゲリラ豪雨予測の実証」による．本研究における計算には，主に九州大学情報基盤研究開発センター研究用計算機システム FUJITSU PRIMEHPC FX10 を利用した．

参考文献

[1] Tomita, H.: SCALE-LES: Strategic development of large eddy simulation suitable to the future HPC, *Solution of Partial Differential Equations on the Sphere* (2012).
[2] Takemasa, Miyoshi, Yamane, S. and Enomoto, T.: Localizing the Error Covariance by Physical Distances within a Local Ensemble Transform Kalman Filter (LETKF), *Scientific Online Letters on the Atmosphere (SOLA)* (2007).
[3] unidata: Network Common Data Form (NetCDF). http://www.unidata.ucar.edu/software/netcdf/.
[4] 九州大学：九州大学情報基盤研究開発センター研究用計算機システム. http://www2.cc.kyushu-u.ac.jp/scp/.
[5] 石川 裕，堀 敦史，Balazs, G.，高木将通，島田明男，清水正明，佐伯裕治，白沢智輝，中村 豪，住元小田和 友仁：次世代高性能並列計算機のためのシステムソフトウェアスタック，情報処理学会研究報告，pp. 1–7 (2013).
[6] Larson, J., Jacob, R., Ong, E., Larson, J. and Jacob, R.: The Model Coupling Toolkit: a new FORTRAN90 toolkit for building multiphysics parallel coupled models, *International Journal of High Performance Computing Applications*, Vol. 19, pp. 277–292 (2005).
[7] Jacob, R., Larson, J. and Ong, E.: M X N Communication and Parallel Interpolation in Community Climate System Model Version 3 Using the Model Coupling Toolkit, *Int. J. High Perform. Comput. Appl.*, Vol. 19, No. 3, pp. 293–307 (online), DOI: 10.1177/1094342005056116 (2005).