

京での大量データの並列相関計算を支援するソフトウェアの提案

吉永 一美^{1,a)} 徳久 淳師¹ 大野 善之¹ 亀山 豊久¹ 堀 敦史¹ 城地 保昌³ 初井 宇記²
石川 裕^{4,1}

概要: データ間の相関係数を求めるための相関計算処理は、データクラスタリングにおいて必要となる処理である。本論文では、大量データに対する相関計算処理を、京を用いて高速に並列化する手法を提案する。提案手法では、各プロセス間のロードバランスを均等に保つとともに、ストレージからのデータ読み込み量を最低限に抑えることで読み込み時間の短縮を図り、高速な並列相関計算処理を実現する。提案手法を適用して実装したアプリケーションを用いた評価の結果、読み込み量を抑えることで実行時間を短縮できることを確認した。

A New Parallelization Method for Correlation Calculation of Large Amount of Data on the K computer

KAZUMI YOSHINAGA^{1,a)} ATSUSHI TOKUHISA¹ YOSHIYUKI OHNO¹ TOYOHISA KAMEYAMA¹
ATSUSHI HORI¹ YASUMASA JOTI³ TAKAKI HATSUI² YUTAKA ISHIKAWA^{4,1}

Abstract: The correlation calculation is used in data clustering to compute correlation coefficient between two data sets. In this paper, we introduce a new parallelization method for the correlation calculation on the K computer. In order to realize effective parallelization, our proposed method keeps load balance between processes and reduces file input time by minimizing total size of the input from storage. Evaluation results show our method can reduce execution time by eliminating unnecessary file input from storage.

1. はじめに

X線自由電子レーザー (X-ray Free Electron Laser : XFEL) 施設である SACLA (SPring-8 Angstrom Compact Free Electron Laser, [1]) を用いることで新たな研究成果の導出が期待されており、多くの研究が進められている。高分解能単粒子構造解析 [2] はその研究の一つであり、これまで不可能だった結晶化されていない生体試料の三次元立

体構造解析を、SACLA を用いることで実現しようとするものである。

より高分解能な立体構造解析を実現するためには、SACLA による大量の回折像データの収集と、その解析が必要である。しかしこの解析処理の中で、得られた画像の分類を行うために、画像データ群の一部を用いた総当たりでの相関計算と、その結果から選出された画像データ群と残りの画像データ群の直積集合要素に対する相関計算という、二つのパターンの大量画像データを用いた大規模な相関計算処理を実施しなければならない。また、SACLA での XFEL 照射実験は数日間連続して実施される。解析処理を SACLA での XFEL 照射実験と並行して迅速に実施し、その結果を実験中の SACLA に対してフィードバックをすることにより、照射実験の精度を高めることが可能となるため、解析処理の高速化への期待は非常に高い。

¹ 理化学研究所 計算科学研究機構
RIKEN AICS

² 理化学研究所 放射光科学総合研究センター
RIKEN SPring-8 Center

³ 高輝度光科学研究センター
JASRI

⁴ 東京大学
The University of Tokyo

a) kazumi.yoshinaga@riken.jp

そこで我々の研究グループでは、SACLA による観測データを京へと転送し、京を用いて大規模な解析処理の並列計算を行うことで、実験期間中のフィードバックも可能な高速な高分解能単粒子構造解析の実現を目指している [3].

本稿では、その実現のために必要不可欠である、大量画像データに対する相関計算処理の高速な並列化手法を提案する。提案手法ではまず、プロセスごとに担当する計算領域を決定し、データ読み込みを行う。そしてその後データをバケツリレーの要領で隣接ランクのプロセスへと送信して順に計算していくことで、全体の計算を完了させる。一連の処理において全てのデータはただ一度読み込まれるのみであり、ストレージからの読み込み量を最低限に抑えることで高速化を図る。

以下、第 2 章において、高分解能単粒子構造解析について簡単に紹介したのちに、処理中に存在する大規模計算である、一つのデータに対する総当たりでの相関計算と、二つのデータ群の直積集合要素に対する相関計算について紹介する。続いて第 3 章において、その大規模計算を京を用いて高速に実行するために考案した並列化手法について述べ、第 4 章においてその手法により実装した高分解能単粒子構造解析の分類処理を用いた評価を行う。そして第 5 章において関連研究を紹介し、第 6 章でまとめと今後の課題について述べる。

2. 高分解能単粒子構造解析における相関計算処理

SACLA を利用した単粒子構造解析は、構造を解析したい粒子を結晶化すること無くそのままの状態ですべての試料として用い、その試料に対して XFEL を照射し、得られた回折像から三次元構造の再構築を行う手法である。しかし、生体単粒子試料の散乱能は非常に弱く、強い量子ノイズを伴った S/N 比の悪い回折像が得られる。また、分子のどの方向に対し XFEL が照射されたかは不明である。加えて、回折像から実像を復元する際に必要な位相情報は失われている。

そこで、(1) 分類・平均処理による S/N 比の向上、(2) 相対位置の決定、(3) 位相の回復、という三つのステップにより、得られた回折像から三次元構造の構築を行う。(1) のステップに含まれる回折像分類処理は、回折像間の相関係数に基づいて類似画像のグループへと分類するものである。しかし、単純に全ての回折像間での相関係数を求める計算を行うとすれば、回折像数 N に対して $O(N^2)$ の計算となるため、大量の画像に対しては適さない。そこで徳久らにより図 1 のような 2 段階での分類処理スキームによる計算量の削減手法が考案されている [2].

まずタスク 1 では、代表回折像と呼ばれる各類似グループの代表となる回折像の選出を行う。このタスクでは、 N 枚の全回折像の中から N_{cand} 枚の画像を代表候補回折像として選択し、その回折像群に対して総当たりで相関係数の

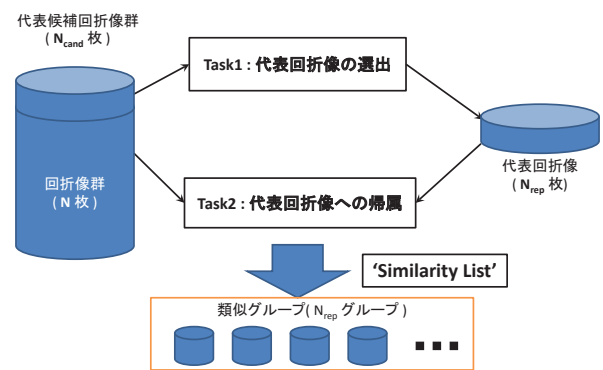
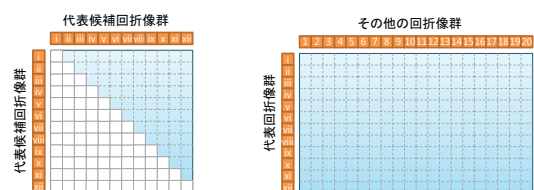


図 1 高分解能単粒子構造解析における分類スキーム
Fig. 1 Representative-All Pair Classification Scheme

計算を行う。そしてその相関係数を用い、 N_{rep} 枚の代表回折像を選出する。続いてタスク 2 では、タスク 1 で選出された代表回折像群 N_{rep} 枚と、代表候補回折像以外の回折像群 $N - N_{cand}$ 枚、二つの回折像群の直積集合要素に対して相関係数の計算を行う。そしてその相関係数を元に N_{rep} グループへの分類を行う。

図 2 は、二種類の相関計算を例示したものである。オレンジ色の四角形は画像データを表し、大きな四角形の中にある各マスは二画像間での相関計算を表している。

図 2(a) は、総当たり要素に対する相関計算を行うタスク 1 の例として、12 枚の代表候補回折像を用いた相関計算を表したものである。相関計算内での画像の順序は無関係であるため、総当たり計算を行うためには対角線に関して線対称としてどちらかのマスの計算を行えばよい。この図では単純に色の付いた部分である、上半分の三角形領域を計算する場合を示している。図 2(b) は、タスク 2 で行う代表回折像群と他の回折像群間の相関計算の例である。この例では、12 枚の代表回折像 (行) と、20 枚のその他回折像 (列) の直積集合要素に対して相関計算を行っている。



(a) タスク 1: 総当たり要素への計算
(Task1: Calculation for All Pairs Pattern)
(b) タスク 2: 直積集合要素への計算
(Task2: Calculation for Cartesian Product Pattern)

図 2 各タスクでの相関計算イメージ
Fig. 2 Examples of Correlation Calculation in Each Task

以上に述べたように、高分解能単粒子構造解析の回折像分類処理の実現には、

a) 一つのデータ群を読み込み、その総当たり要素に対す

る相関計算

- b) 二つのデータ群を読み込み、そのデータ群の直積集合要素に対する相関計算

の二種類の相関計算を行う必要がある。SACLA で観測可能な解像度に基づき設定されたパラメータでは、150 万枚程度の回折像を 2 万程度のグループへと分類することを目標としている。また、タスク 1 において 2 万枚程度の代表回折像を選出するために必要な代表候補回折像の枚数 N_{cand} は 15 万枚程度と見積もられている。そのためタスク 1 においては 15 万枚の画像を読み込み、その総当たり要素数であるおよそ 1.125×10^{10} 回の相関計算を、タスク 2 においては代表回折像 2 万枚とその他回折像 150 万枚を読み込み、その直積集合要素数である 3×10^{10} 回の相関計算を実施する必要がある。

このような大量のデータ読み込みを伴う膨大な回数の相関計算処理を対象とし、京の計算資源を利用し並列に処理することで、高速化を目指す。

3. 京を用いた並列相関計算手法

3.1 設計指針

京を用いての実行であるため、実装は MPI+OpenMP によるハイブリッド並列を前提とする。今回対象としている相関計算はそれぞれ独立であり、一つ一つの相関計算の計算量が同じであるため、各 MPI プロセスの処理量を均一にするために相関計算回数がほぼ均等になるような処理領域の分割を行う。

その上で、大量データのストレージからの読み込み時間を短縮することにより、全体の処理時間の削減を狙う。そのために、以下の二つを考慮した設計を行う。

3.1.1 総読み込みデータ量の削減

まずはプログラム全体でのストレージからの読み込みデータ量を最小限に抑えることを考える。総当たりあるいは直積集合要素に対する相関計算を並列化する場合、計算領域をどのように分割しても、その領域の計算に必要な入力データに重なりが生じてしまう。このため、単純に各プロセスが必要なデータを全てストレージから読み込むように実装した場合、重複データの読み込みにより総読み込みデータ量が増大する。そこで、プログラム全体において一つのデータはただ一度だけしか読み込まれないような設計を考え、一度読み込みを完了したデータはそのデータを保持するノードとの通信により取得することで、ストレージからの読み込みを削減し高速化を図る。

予備実験として、実際に高分解能単粒子構造解析の実験用模擬画像データである netcdf 形式・14MiB のデータを用いて、実験用画像 1 枚あたりの読み込み時間を京の 1 ノードに 1 プロセスを起動して測定したところ、50ms 程度であった。一方、同じ画像データの通信時間を 2 ノード間の

Send-Recv による単純な Ping-Pong 通信にて測定したところ、片道 6.2ms 程度であり、ストレージからの読み込みと比較しておよそ 1/8 となった。プログラム全体を通して一度でも読み込まれたデータは、再度ストレージから取得するのではなく通信により取得することで、高速化が期待できる。

3.1.2 ランクディレクトリの利用

京のファイルシステムは Lustre ファイルシステムを拡張した FEFS[4] を採用しており、グローバルファイルシステム (GFS) とローカルファイルシステム (LFS) の 2 種類のファイルシステムが存在する。計算ノードがアクセスできるのは LFS のみであり、アプリケーション実行前に必要なファイルを GFS から LFS へコピーするファイルステージングという処理を行わなければならない。

この LFS には、2 種類のディレクトリが存在する。一方は共有ディレクトリである。これは一般的な Lustre 上のディレクトリと同様のもので、共有ディレクトリに配置されたファイルは、複数のディスクにストライピングして格納される。他方はランクディレクトリと呼ばれる、MPI のランクを意識したものである。ランクディレクトリは各 MPI ランクに対応し、そのランクが配置されるノードに直近の一つのストレージ上に作成される。そのため、ランクディレクトリ内のファイルは対応する MPI ランクのプロセスから高速にアクセスすることができる。プログラム実行前にランク毎に読み込むファイルが決定しているのであれば、そのファイルを対応したランクディレクトリに配置することで読み込み時間の短縮が期待できるため、各プロセスの読み込むデータが実行前に静的に決定するように設計する。

3.2 二つの画像群の直積集合要素に対する計算手法

以上の方針のもと、タスク 1 での処理である総当たり要素への計算と、タスク 2 での処理である直積集合要素に対する計算の、二種類の相関計算について並列化手法を考案した。総当たり要素への相関計算は、直積集合要素に対する相関計算の特殊な場合と考えることができるため、まず基本となる直積集合要素に対する相関計算の並列化について述べる。

図 3 に提案する手法による処理の流れを示す。この図では、図 2(b) に示した代表回折像 12 枚とその他回折像 20 枚を用いたタスク 2 の処理を、プロセス数 $M = 4$ で並列処理する場合を例として示している。

まず、各プロセスの計算量がほぼ均一にするために、全体の計算領域をほぼ均等な列数になるように M 個に区切り、それぞれをプロセスに割り当てる (図 3(a))。この例では、プロセス 0 から順にその他回折像 1-5, 6-10, 11-15, 16-20 に対する相関計算を行う割り当てとなっている。計算領域を列で区切って分割したため、計算に必要なその

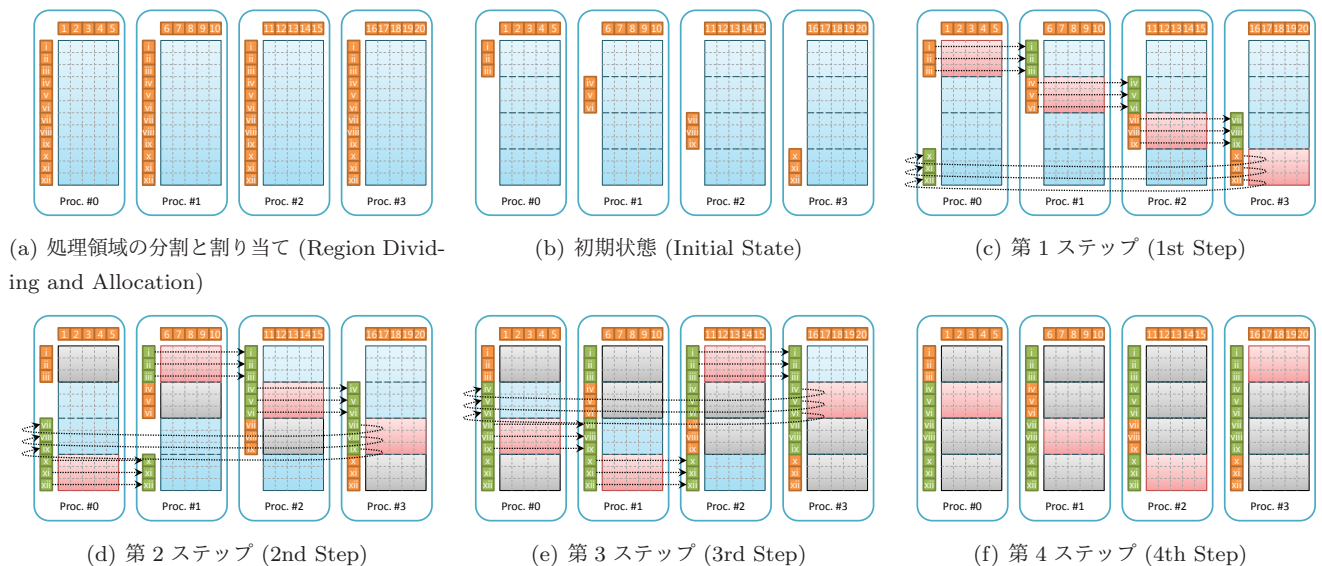


図 3 直積集合要素に対する相関計算処理並列化手法 (12×20 要素, 4 プロセス)

Fig. 3 A Workflow for Cartesian Product Pattern (12×20 Elements, 4 Processes)

の他回折像のデータについては、プロセスごとに固定でかつ重複もない。各プロセスが実行開始時に必要な部分を一度だけストレージから読み込めば、あとは読み込みも転送も必要なく、計算終了まで保持し続けるだけでよいことになる。

一方、代表回折像にあたるデータである i -xii については、各プロセスが全てのデータを利用する必要がある。ここで、3.1.1 で述べたようにストレージからの重複した読み込みを無くすために、データ全体を各プロセスで分担して最初に読み込み、その後のデータ取得は通信によって行うようにする。本手法では、読み込み量のバランスを考慮し、読み込むデータ量が各プロセスでほぼ均等になるように M 個に分割して各プロセスに割り当てる。図 3(b) は、今回の例において各プロセスが自分の担当として割り当てられたデータを読み込み終えた状態であり、重複なく、全てのデータが読み込まれている。この状態を初期状態とし、相関計算を開始する。

各プロセスは第 1 ステップの計算として、初期状態で読み込んだデータを用いて計算できる領域 (図 3(c) の赤色の領域) について相関計算を開始する。ここで、各プロセス内では OpenMP を用い、一つ一つの相関計算をスレッド並列で実行する。初期状態で読み込んだデータを用いて計算できるのはここまでであり、その他の領域を計算するためには、プロセス間でのデータ転送が必要である。そこで、各プロセスは現在計算に用いている行方向のデータ (初期状態で読み込んだ行方向のデータ) を、次のランクのプロセスへと送信する。図中では、プロセス 0 がプロセス 1 へ、 i -iii のデータを転送している。なお、オレンジ色のデータはストレージから読み込んだものを、緑色のデータは通信により取得したものを示している。

続いて第 2 ステップでは、前のプロセスから受け取った新たなデータにより計算可能となった領域 (図 3(d) の赤色の領域) の相関計算を行う。グレーの領域は既に計算が終わった領域である。そして第 1 ステップと同様に、現在計算に利用しているデータを次のランクのプロセスへと送信する。

この処理を M ステップほど繰り返すことで、各データが全てのプロセスへと一巡して全体の相関計算が完了する。図 3 の例は $M = 4$ であるので、第 3 ステップ (図 3(e)) を経て、第 4 ステップ (図 3(f)) を終わると処理が終了する。最終ステップではデータの転送は不要となるので行わない。

最終的に、プログラムがストレージから読み込んだデータは初期状態で読み込まれているデータのみであり、各データが全体を通じてただ一度だけしか読み込まれない状況が実現できている。つまり全体での総読み込みデータ量は、ストレージ上での総データ容量と同一という、最小限の量に抑えられている。さらに、この分割方法では各プロセスが読み込む領域を実行前に一意的に決定することが可能である。そのため、3.1.2 で説明した、京のランクディレクトリを利用したファイルアクセスの高速化も同時に図ることができる。

この一連の処理において、行方向のデータ格納バッファを二つ用意し、計算利用バッファと通信受信用バッファを交互に切り替えて利用することで、計算中のバックグラウンドでの通信が実現できる。高分解能単粒子構造解析での相関計算において、二つの画像間の相関係数を求める計算には FFT などが含まれ、京で実行すると相関係数の計算一回あたり約 60ms の処理時間となる。そのため、3.1.1 の予備実験により得られた 6.2ms という画像転送時間は十分

に隠蔽可能である。また、通信するプロセスは常に隣接するランクのプロセスである。京では、利用するノード指定を三次元で指定することで、各ランクはトラス状に配置され、隣接ランク間がワンホップで通信できるランク配置になることが保証されている [5]。隣接ランク間での通信のみの本手法には、通信の衝突もなく安定して高速にデータが転送されることが期待できる。

3.3 一つのデータ群の総当たり要素に対する計算手法

図 2(a) のような一つのデータ群の総当たり要素に対する相関計算処理は、同内容の二つのデータ群による直積集合要素への相関計算処理と考えることで、3.2 節の方式を用いれば実装可能である。しかし、全要素の約半数は重複した無駄な計算となるために、そのまま適用すると非効率的である。無駄な計算を除去し、各プロセス間のロードインバランスが生じないような手法が必要となる。

総当たり要素に対する相関計算の処理並列化手法の例を図 4 に示す。この例は 15 の入力データをプロセス数 $M = 5$ で並列化した場合である。

基本的な処理の流れは直積集合要素の場合とほぼ同じである。まずは計算領域を各プロセスで均等になるように分割し、担当する列と読み込むデータを決定する。直積集合要素に対する手法とは異なり、二つのデータ群は同一のデータ群であるため、この時点で読み込みデータの分担は終了である。図 4(a) はデータの読み込みを終えた初期状態である。各プロセスはオレンジ色で塗りつぶされた部分のデータを読み込めば、黒枠の部分のデータも準備できたことになる。

その後、第 1 ステップとして相関計算処理を開始し、バックグラウンドで自分の持つデータを隣のプロセスへと送信する (図 4(b))。ただし、ここで計算する領域内には重複計算要素が存在するため (図中の白抜き部分)、その部分を除外する。全てのプロセスにおいて均等に除外部分が発生するため、処理量の差異は生じない。

続いて第 2 ステップで、受け取ったデータにより計算可能となった領域の計算をしつつ、データの送信を行う (図 4(c))。図中の×印部分に示される領域は、このステップで計算する領域と重複するために計算が不要になった領域である。

このステップを繰り返すことで、奇数プロセスであれば $(M + 1)/2$ ステップで全体の相関計算処理が完了する。図 4 の例では第 3 ステップ (図 4(d)) で完了となる。なお、偶数プロセスで分割した場合も同様の手順を追うことで、 $M/2 + 1$ ステップで相関計算が完了する。ただしこの場合は最後に計算する領域に重複が発生するため、最終ステップでの担当計算領域を定義する必要がある。

4. 回折像分類処理への提案手法の適用と評価

3 章にて提案した並列化手法を用いて、実際に高分解能単粒子構造解析の回折像分類処理に必要な相関計算アプリケーションを実装し、評価実験を行った。実験では模擬回折像データを用いており、このデータは回折像 1 枚を 1 ファイルとした netcdf 形式で、ファイルサイズは 14MiB である。

4.1 ファイル読み込み削減による高速化の評価

まず、ファイル読み込みを削減したことによる高速化の評価を行うために、(1) 提案手法の通りデータ転送を行うことでデータ読み込み量を削減したアプリケーションと、(2) 提案手法と同様の手順を踏むが、転送によるデータの取得は行わずに必要なデータは全てストレージから読み込むように変更したものの、二種類のアプリケーションを用いた比較実験を行った。

この実験では京の 192 ノードを用い、ランクディレクトリを利用している。利用する回折像枚数は行方向に 1920 枚、列方向に 768 枚とした。この枚数は京のステージイン可能なファイル数制約を考慮して設定したものである。表 1 に 5 回実行した結果を示す。

表 1 読み込みデータ量の削減による実行時間の比較

Table 1 Elapsed Time of Application with/without Data Exchange

通信によるデータ 読み込み量削減の有無	実行時間 [s]		総読み込み時間 [s]	
	平均	最大	平均	最大
(1) 有り	178.82	184.28	12.30	18.41
(2) 無し	321.20	406.37	154.84	196.73

この結果から見て取れるように、通信によるデータ取得によりストレージからのデータ読み込み量を削減したことで、全体の実行時間の短縮が実現できている。また、全てのデータをストレージから読み込むために総読み込みデータ量が多くなる (2) の実装においては、読み込み時間の変動幅が大きくなり、その影響により全体の実行時間の変動も大きくなっている。読み込み時間が変動する原因として、他ユーザのアプリケーション実行前/実行後に発生するファイルステージング処理や、実行中の他アプリケーションによるストレージアクセスなどの外的要因が考えられる。これらの要因は共用計算機である京において避けることのできないものであり、外的要因による性能低下の影響を軽減するためにも、データ読み込み量を削減した提案手法は有用である。

4.2 大規模実行でのスケーラビリティ評価

続いて、実際の高分解能単粒子構造解析で想定している

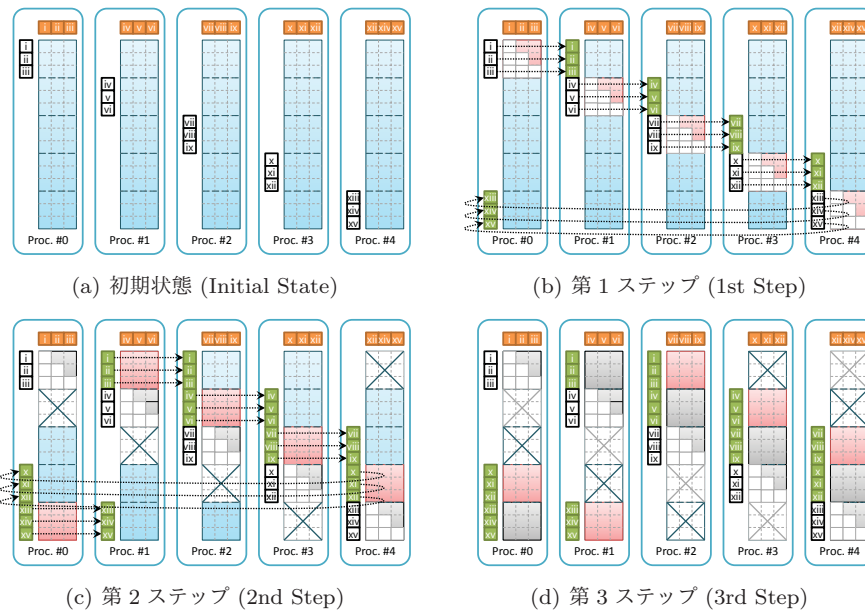


図 4 総当たり要素に対する相関計算処理並列化手法

Fig. 4 A Workflow for All Pairs Pattern

パラメータに基づいた大規模な相関計算の実行を行った。この実験の目的は、実際に SACLA にフィードバックすることが可能な時間で回折像分類処理が完了するかどうかを確認するとともに、大規模実行によるスケーラビリティを確認するものである。

この実験では、実験用模擬回折像 15 万枚を用いて実際に計算を行い選出された 20,951 枚の代表回折像(行方向のデータ)と、1,586,000 枚の模擬回折像(列方向のデータ)を用いた、分類スキームのタスク 2 である直積集合要素に対する相関計算を、京のフルノードである 82,944 ノードにより実行した。なお、提案した手法は行・列方向のファイルをそれぞれプロセス数 M で均等に分割した読み込みを行い、 M ステップで処理を行うという実装であるため、プロセス数が行・列どちらかのファイル数を上回る場合はそのままでは適応できない。本実験においては、図 5 のように計算領域・ノードをそれぞれ 27 ブロックに分割し、各ブロック毎に割り当てて実行している。

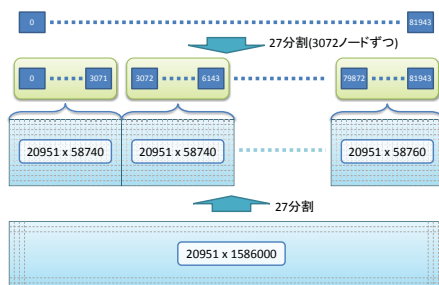


図 5 大規模実行時の分割処理

Fig. 5 Job and Node Allocation of Huge Execution

大規模実行の結果、全体の計算は 6,904 秒で完了した。

利用したノード時間はおよそ 159,000 ノード時間であり、京を用いることで SACLA での照射実験中に回折像分類処理を終了させ、フィードバックが可能であることが確認できた。

一方で、全体の計算時間である 6,900 秒のうち、各プロセスが画像間の相関計算自体に要した時間は概ね 3,600 秒程度となっており、残りのおよそ 3,300 秒が初期化や終了処理の他、画像の読み込み時間や通信時間に費やされていた。全プロセスにおいてログの計測を行っていないために詳細な解析はできていないが、計測したプロセスにおける画像読み込み時間は平均して 1 画像あたり 50 秒程度となっており、小規模実行時と比較して非常に悪化していることが確認できた。

そこで、実行規模を変化させた場合の画像読み込み時間の変化を確認するために、1 ノードあたりの読み込みファイル数を大規模実行時と同一に固定し、ノード数を変化させた際の 1 ファイルの読み込みに要する時間の変化を測定した。図 6 にその結果を示す。ノード数の増加とともに読み込み性能が悪化していることが確認できる。この原因として、ファイル読み込みノードの増加による I/O ノードへの処理の集中や、ファイル数の増大によるメタデータサーバへの処理の集中が考えられる。後者のメタデータサーバへの処理集中については、現在入力として 1 画像を 1 ファイルとしているが、複数画像を一つのファイルにまとめることで改善可能だと考えている。

5. 関連研究

大量のデータを用いた並列計算を支援するフレームワークとして、MapReduce 処理系 [6] が有名である。京にお

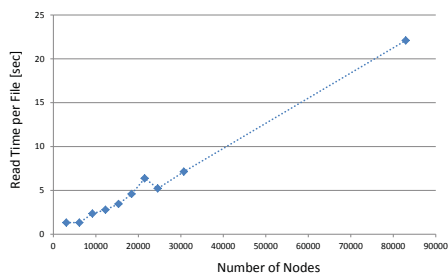


図 6 ファイル読み込み時間の変化
Fig. 6 Average File Read Time

いてもこの MapReduce 処理系を実装したフレームワークとして、松田らにより K MapReduce(KMR)[7] が開発され、提供されている。KMR は京のファイルシステムや通信特性を考慮し、京で高いパフォーマンスを示すような実装がなされている。しかし、今回我々が対象とした総当たり要素や直積集合要素を用いる問題に対しては、一般に MapReduce 処理系には向かないと言われている [8]。MapReduce 処理系において最も単純にこの処理を実装する場合、全ての組み合わせを Mapper への入力データとして前もって準備しておく必要がある。小規模な計算であれば実現可能だが、2 章で述べたような、組み合わせが 3×10^{10} 個存在するような大規模な計算を実現する場合は、列挙することは現実的に不可能である。今回のように予め全体のデータ数と計算領域が確定している場合は、その領域を複数の部分領域に分割し、各 Mapper がその部分領域を計算するように実装することで、直積集合要素に対する相関計算処理が実現できる。しかし各計算領域の計算に必要なデータには必ず重複があるため、同一データの重複読み込みが発生してしまう。

データクラスタリング手法として、K-Means 法 [9] が広く用いられている。K-means 法はユーザが指定した数である K 個のクラスタへとデータを分類するもので、

- (1) ランダムに各データをクラスタへ所属させる。
- (2) 各クラスタの中心点を計算する。
- (3) 各クラスタの中心点との相関係数を全データに対して計算し、その値に基づき最も類似するクラスタへと所属を変更する。
- (4) 所属の変化がないか、中心点の変化量が指定した閾値以下であれば終了する。そうでなければ 2 へ戻る。

という手順により実行される。高分解能単粒子構造解析における分類処理は、一つの画像を複数のグループへと所属させることを許容している。K-Means 法では所属するクラスタは単一であったが、高分解能単粒子構造解析のように複数クラスタへの帰属を許容するために、K-Means 法を拡張した手法として Fuzzy C-Means(FCM) 法 [10] が存在

する。K-Means 法と FCM 法のどちらにおいても 3 において相関計算は発生するため、我々の提案した手法を利用することで高速な並列化が可能だと考えている。

6. まとめ

本稿では、高分解能単粒子構造解析に必要となる、大量データを読み込み、その総当たりまたは直積集合要素に対しての相関計算処理を、京を用いて高速に処理する手法について提案した。実際に提案した手法を高分解能単粒子構造解析の回折像分類処理に適用し、その動作を確認するとともに、データ読み込みを削減し通信することによる実行時間の短縮を確認した。また、大規模実行の結果、SACLA での実験中のフィードバックが可能なノード時間で処理が完了することを確認した。

今後の課題として、大規模実行で判明したファイル読み込み時間のスケラビリティ低下に対する改善が挙げられる。ファイル数の削減の他、読み込みタイミングをずらすなど、改善するための実装を進めていく。また、総当たり要素や直積集合要素への計算を必要とする別のアプリケーションへの適用も進めていく。容易に適用を可能とするために、提案手法を実装したフレームワークを作成し提供していく予定である。

謝辞 本研究を進めるにあたり、KMR の開発者である理研の松田元彦氏と滝澤真一朗氏には、多くの助言を頂きました。どうもありがとうございました。本論文の結果は、理化学研究所のスーパーコンピュータ「京」を利用して得られたものです (課題番号:hp120214)。

参考文献

- [1] RIKEN: SACLA(XFEL), RIKEN (online), available from <http://xfel.riken.jp/index.html> (accessed 2013-11-14).
- [2] Tokuhisa, A., Taka, J., Kono, H. and Go, N.: Classifying and assembling two-dimensional X-ray laser diffraction patterns of a single particle to reconstruct the three-dimensional diffraction intensity function: resolution limit due to the quantum noise, *Acta Crystallographica Section A*, Vol. 68, No. 3, pp. 366–381 (online), DOI: 10.1107/S010876731200493X (2012).
- [3] Tokuhisa, A., Arai, J., Joti, Y., Ohno, Y., Kameyama, T., Yamamoto, K., Hatanaka, M., Gerofi, B., Shimada, A., Kurokawa, M., Shoji, F., Okada, K., Sugimoto, T., Yamaga, M., Tanaka, R., Yokokawa, M., Hori, A., Ishikawa, Y., Hatsui, T. and Go, N.: High-speed classification of coherent X-ray diffraction patterns on the K computer for high-resolution single biomolecule imaging, *Journal of Synchrotron Radiation*, Vol. 20, No. 6, pp. 899–904 (online), DOI: 10.1107/S0909049513022152 (2013).
- [4] 酒井憲一郎, 住元真司, 黒川原佳: スーパーコンピュータ「京」の高性能・高信頼ファイルシステム, 雑誌 Fujitsu, Vol. 63, No. 3, pp. 280–286 (2012).
- [5] Yuuichirou Ajima, Tomohiro Inoue, S. H. and Shimizu, T.: Tofu: Interconnect for the K computer, *Fujitsu sci-*

- entific & technical journal*, Vol. 48, No. 3, pp. 280–285 (2012).
- [6] Dean, J. and Ghemawat, S.: MapReduce: simplified data processing on large clusters, *Commun. ACM*, Vol. 51, No. 1, pp. 107–113 (2008).
 - [7] Motohiko Matsuda, N. M. and Takizawa, S.: K MapReduce: A Scalable Tool for Data-Processing and Search/Ensemble Applications on Large-Scale Supercomputers, *Proceedings of IEEE Cluster 2013* (2013).
 - [8] Miner, D. and Shook, A.: *MapReduce Design Patterns*, O'Reilly Media (2012).
 - [9] MacQueen, J. B.: Some Methods for Classification and Analysis of MultiVariate Observations, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability* (Cam, L. M. L. and Neyman, J., eds.), Vol. 1, University of California Press, pp. 281–297 (1967).
 - [10] Bezdek, J. C.: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell, MA, USA (1981).