

## PC グリッド向け PNN アルゴリズムの評価

## Evaluation of PNN Algorithm for PC Grid

肥塚 拓也<sup>†</sup>

Takuya Koezuka

若谷 彰良<sup>‡</sup>

Akiyoshi Wakatani

## 1 はじめに

今日、一般の企業および大学においても PC を大量導入することが多くなった。しかし、それらの PC は、常に計算に時間を費やしているわけではなく、「遊休 PC」となっているものも多い。近年、これらの PC を用いてグリッド環境を構築し、企業・大学の所有する計算機資源を有効活用することが考えられてきている。

一般に PC グリッドは、マスター PC とボランティア PC からなり、マスター PC によって与えられたジョブが管理され、ボランティア PC によって分割されたジョブを処理し、マスター PC に送信するというシステムとなっている。

本稿では、VQ 画像圧縮でのコードブック生成アルゴリズムの 1 つ、PNN (Pairwise Nearest Neighbor) を取り上げ、それをグリッド環境向けに改良したアルゴリズム (Grid PNN) を提案し、異なる性能の PC が混在する研究室レベルでの PC グリッド環境上で実行し、性能を評価する。PNN は階層型クラスターリングと類似しているため、本実験において肯定的な結果を出すことができれば、クラスターリングやパターン認識などに応用することが期待できる。

## 2 Grid PNN

PNN は、VQ (Vector Quantization) 画像圧縮で用いられるコードブック生成アルゴリズムの一つであり、高品質なコードブックを生成することができる。しかし、PNN は比較的計算量の大きなアルゴリズムであり、PNN の改良アルゴリズムもいくつか提案されてきた [1]。本稿では、オリジナルの PNN を並列化することで

高速化することを考える。

PNN のアルゴリズムの概略は、1)  $T$  個のトレーニングベクトルを用意し、2) すべての組み合わせのトレーニングベクトル間の距離を計算し、3) 距離が最小となるトレーニングベクトルのペアを融合し一つのベクトルにし、4) トレーニングベクトルの数が目標とするコードブックのサイズ  $K$  になっていれば終了し、そうでなければ、2) のステップに戻る、という手続きからなる。

一方の GridPNN [2] では、トレーニングベクトルを並列化するタスクの数  $P$  で分割し、分割したタスク毎にプロセッサに割り当てる。なお、分割するタスク数は使用するプロセッサ数よりも多くてもよい。アルゴリズムは次のようになる。1)  $T/P$  個のトレーニングベクトルをタスクとし、それをボランティア PC に渡し、2)  $K$  個のベクトルになるように PNN を行い、3) 生成された  $K$  個のベクトルをマスター PC に渡し、4) マスター PC において  $K \times P$  個のベクトルから  $K$  個のベクトルになるように PNN を行う。すなわち、ボランティア PC で分割した PNN を行った後、マスター PC で再度の PNN を行う。なお、オリジナルの PNN で生成されるコードブックと GridPNN で生成されるコードブックは異なるものとなるが、その違いによる画像圧縮の品質の差は顕著ではない [2]。また、計算量の詳細は文献 [2] で議論している。

## 3 AD-POWERs の概略

AD-POWERs は、大日本印刷株式会社の開発した、Windows 上で動作するグリッド環境構築ソフトウェアである [3]。グリッド環境の PC は、ジョブの管理を行うマスター PC と、分割されたジョブを実行するボランティア PC で構成される。マスター PC は、各ボランティア PC でジョブが実行できる状態にあるかを監視し、実行可能なボランティア PC に随時ジョブを送る。

<sup>†</sup> 甲南大学大学院

<sup>‡</sup> 甲南大学

この実行できる状態とは、ボランティア PC 上でスクリーンセーバが起動している状態のことを指す。そして、すべての分割ジョブが完了すると、マスター PC 上で終了処理を行いジョブが完了する。

## 4 実験環境

本実験では、マスター PC として P0 (Intel Pentium D 2.80GHz), ボランティア PC として, P1 (Intel Core 2 Duo 1.86GHz (2 Core)), P2 (AMD Opteron DP 242 1.6GHz (2 CPU)), P3 (Intel Xeon 2.80GHz (2 CPU)) の3台を用いた。これらの PC はいずれも Windows-XP (SP2) 上で動作し, PC 間の接続にはイーサネット (100BASE-T) を用い, AD-POWERS でグリッド環境を構築した。

また生成するコードブックのサイズは 64, トレーニングベクトル数は 4096 とする。

## 5 評価

本実験では, a) 各ボランティア PC 単体, b) 各ボランティア PC 単体 (ただし2ジョブを並列実行), c) 各ボランティア PC 上で並列実行, (計3ジョブを並列実行) d) 各ボランティア PC 上で並列実行 (ただし各 PC 上で2ジョブを並列実行, 計6ジョブを並列実行) の組み合わせで Grid PNN を実行した。このときのジョブの分割数は6で固定である。したがって, すべてのパターンにおいて得られるコードブックは等しい。

その実行にかかった時間は図1のグラフのようになった。グリッド環境のため各 PC の性能にばらつきがあり, 単純に比較はできないが, 6ジョブを並列かつ同時に実行したときの処理時間 (d) は, 各ボランティア PC1 台のみで実行したときの処理時間 (a) の平均と比べると約2.9倍の速度向上が見られた。また, マルチコア (P1), マルチプロセッサ (P2, P3) 上の処理においても, それぞれ約1.6倍, 約1.7倍, 約1.5倍の速度向上が見られた。グリッドを構成する PC の性能のばらつきを考慮すると, マスター PC での後半の PNN 処理のオーバーヘッドを考慮してもこの結果は期待される性能向上度を満たしていると考えられる。

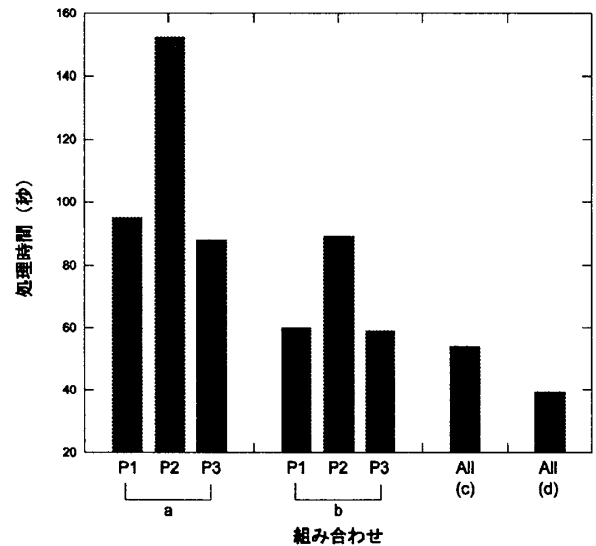


図1 各 PC の組み合わせ別 Grid PNN の処理時間

## 6 おわりに

以上の実験より, Grid PNN および研究室の一般的な PC でのグリッドの有用性が確認できた。しかし, 本実験においてはユーザによるジョブの中断, すなわちスクリーンセーバの終了を考慮していない。今後は, ジョブの中断も含めたより実用的なグリッド環境の構築を行い, さらに他のアプリケーションへの適用を検討する予定である。

## 参考文献

- [1] Franti P. and Kaukoranta T.: "Fast implementation of the optimal PNN method", Proc. of Int'l. Conf. on Image Processing, Chicago 1998
- [2] Akiyoshi Wakatani: "PNN Algorithm for PC Grid System", Proc. of the Second International Conference on Software Engineering Advances (ICSEA 2007), Aug., 2007 (to appear)
- [3] 大日本印刷株式会社: <http://www.ad-powers.jp/>