

B-029

グリッド環境における多段的 PNN の実装と評価

Implementation and Evaluation of Multi-step PNN for Grid Computing

肥塚 拓也 *1

Takuya Koezuka

若谷 彰良 *2

Akiyoshi Wakatani

1 はじめに

著者らは 2007 年に PC グリッドを対象としたアルゴリズム “Grid PNN” を提案した [1] [2]. このアルゴリズムは 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) を改良したものである. 本稿ではこの Grid PNN をさらに改良した “Multi-Step Grid PNN” を提案する. これは, 従来マスター PC 上で処理していたものをスレーブ PC で並列処理させるものである.

2 Grid PNN の概略と短所

以下に Grid PNN について概略を述べ, その問題点を挙げる.

トレーニングベクトル数を T , コードブックベクトル数を K , スレーブ PC 数を P とする. 各スレーブ PC には, トレーニングベクトルを分割したものがマスター PC から渡される. そのベクトルのサイズは $\frac{T}{P}$ である. スレーブ PC はそのベクトルに対してサイズが K になるまで PNN を行う (第 1 プロセス). こうして各スレーブ PC にて生成されたベクトルを合計するとサイズ $K \times P$ のものとなる. このベクトルに対して改めてマスター PC で PNN を行い最終的にサイズ K のコードブックを生成する (第 2 プロセス).

このようにしてコードブックを生成するのだが, 最後にマスター PC が行う第 2 プロセスは P が大きくなるにつれて計算量が大きくなり, 結果として処理時間の短縮を阻害していることがわかった. そこで, マスター PC 上での処理を分割してスレーブ PC に再度計算させるアルゴリズムを考案する.

3 Multi-Step Grid PNN

第 2 プロセスを複数のステップで実行することを考える. $P = M^n (M \geq 2)$ とする. 第 1 プロセスが完了した時点では, マスター PC 上にはサイズ $K \times M^n$ のベクトルが存在する. このベクトルをトレーニングベクトルとみなし, プロセッサ数を M^{n-1} とし, 再度スレーブ PC に処理をさせる. これを “第 2 プロセス第 1 ステップ” とする. 第 2 プロセス第 1 ステップが完了した時点では, マスター PC 上にはサイズ $K \times M^{n-1}$ のベクトルが存在することになる. これをプロセッサ数が 1 になるまで繰り返す. ステップ数は計 $\frac{\log P}{\log M}$ 回となる. このアルゴリズムを “Multi-Step Grid PNN” と呼ぶことにする. $M = 2, n = 2$ のケースを図 1 に示す.

4 計算量の比較

Grid PNN と Multi-Step Grid PNN の計算量は次のようになる. Grid PNN の計算量は,

$$\sum_{i=P \times K}^{K+1} \frac{\alpha}{2} (i^2 - i) = \frac{\alpha}{6} ((PK)^3 - PK - K^3 + K) \quad (1)$$

となり, Multi-Step Grid PNN の計算量は,

$$\frac{\alpha}{6} ((MK)^3 - MK - K^3 + K) \frac{\log P}{\log M} + \beta K (P - 1) \frac{M + 1}{M - 1} \quad (2)$$

となる. ただし α は 1 回あたりのベクトル間の距離計算のコストであり, β は 1 ベクトルあたりの通信コストである. 以上の式より, P が十分に大きければ計算量は Multi-Step Grid PNN の方が小さく考えられる. また, β が α に比べて極めて大きい場合は, M として 2 より大きい値をとることが望ましい.

*1, *2 甲南大学

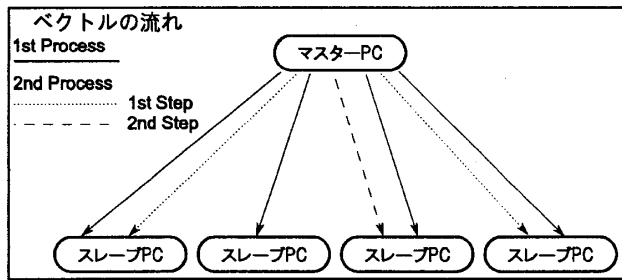


図1 Multi-Step Grid PNN の概念図

5 実験環境

本実験では、マスター PC として P0 (Intel Pentium D 2.80GHz), スレーブ PC として、P1 (Intel Core 2 Duo 1.86GHz (2 Core)), P2 (AMD Opteron DP 242 1.6GHz (2 CPU)), P3, P4 (Intel Xeon E5335 2.00GHz (2 CPU × 4 Core)) の 4 台を用いた。これらの PC の OS は Windows XP Service Pack 3 (P1, P2), Windows Vista (P3, P4) であり、PC 間の接続にはイーサネット (100BASE-T) を用い、AD-POWERS[3] でグリッド環境を構築した。

6 評価

Multi-Step Grid PNN の改良点を確認するため、本実験では第 2 プロセスの処理時間に着目した。プロセッサ数 P (処理分割数) を 4 (P1, P2, P3, P4 それぞれで 1 プロセスを処理), 8 (P1, P2, P3, P4 それぞれで 2 プロセスを処理) とし、生成するコードブックのサイズ K は 100, 300, 500, トレーニングベクトル数 T は 8000 とした。表 1, 2 は第 2 プロセス以降の処理時間の比較とスピードアップを表したものである (単位は秒)。なお、この時間はトレーニングベクトル数には依存しない。

表1 $P=4$ の場合

	Grid PNN	Multi-Step	スピードアップ
$K = 100$	4.8	2.5	1.92
$K = 300$	125	28.3	4.41
$K = 500$	419.6	129.5	3.24

グリッドコンピューティングの性質上、多少のばらつ

表2 $P=8$ の場合

	Grid PNN	Multi-Step	スピードアップ
$K = 100$	38.1	4.2	9.07
$K = 300$	1021.1	38.1	26.80
$K = 500$	4221.02	194.9	21.65

きはあるが、 $P=4$, $P=8$ いずれも処理時間を大幅に短縮できることが示された。とくに、 $P=8$ のときが顕著であり、 P が大きいほど効果があることが分かる。また、式 (1), (2) で示された計算量から、最大次数の項に着目しその比を考えて導出した予想スピードアップは $\frac{P^3 \log M}{M^3 \log P}$ である。たとえば $P=8$, $K=500$ のとき予測値は 21.3 であり、実験結果は 21.65 であることからこの実験結果の妥当性は確認できる。

7 おわりに

Grid PNN を用いた場合、プロセッサ数が大きくなるほどマスター PC で行われる第 2 プロセスのコストが大きくなるという問題があり、容易に PC グリッドの規模を拡大することができなかった。

しかし、今回提案する Multi-Step Grid PNN を用いることにより、プロセッサ数を大きくした時のボトルネックが軽減され、規模の大きなクラスタリングの適応も可能となった。今後は一般的なクラスタリングを目標にし、より実用性を高めたい。

参考文献

- [1] Akiyoshi Wakatani: "PNN Algorithm for PC Grid System", Proc. of the Second International Conference on Software Engineering Advances (ICSEA 2007), Aug., 2007
- [2] 肥塚拓也・若谷彰良: "PC グリッド向け PNN アルゴリズムの評価", FIT2007 第 6 回情報科学技術フォーラム, 2007 年 9 月
- [3] 大日本印刷株式会社: <http://www.ad-powers.jp/>