

1. 成分分解法(LFM A) の概要

成分とは

配列データは必ず幾つかの "より根源的な情報" に分解できる (図1). これを成分という. 成分 B とは位置情報, 値情報, 順序集合などであり, 扱いやすさのために通常は1次元の配列として記述・実装される.

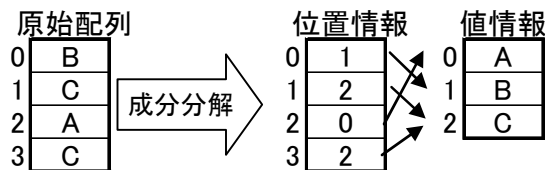


図 1. 原始配列を位置情報, 値情報に分解

表形式データは配列であるので成分に分解できる.

成分分解法とは

検索・ソート・更新・集計などの代表的なデータベース処理は, 元になる表形式データから新たな表形式データを生成する処理である. 従来技術のように表形式データから表形式データを生成するのではなく, 成分群から成分群を生成して新たな表形式データを作成するのが成分分解法である.

ソートでの性能差:

ソートアルゴリズムを例に考える. 従来技術では全要素を比較する必要があり, 処理量はどうしても $O(n \cdot \log(n))$ を下回ることができない.

比べて成分分解法のソートでは, データが成分に分解されているので, 処理に必要な成分のみに選択的にアクセスして $O(n)$ でソートを行うことが出来る. 大規模なソートの場合数桁違いの性能差となる.

多段階処理での効率:

従来技術で多用されるインデックスは処理前の表形式データに対して作成されている. 多段階処理の場合には処理結果の (作成されたばかりの) 表形式データに対してインデック

A LFM: Linear Filtering Method

B 複数の成分を組み合わせないと元のデータを復元することはできないから, 各成分は元のデータの部分でしか無い.

スを設定する必要が生じ不都合である。

一方、成分分解法では、処理の後でも成分分解されているから、多段階の処理でも効率が良い。

並列処理・超並列処理：

加えて、成分に分解されているとテーブルは単純な 1 次元配列（多くは整数配列）の集合になるので並列処理・超並列処理を設計しやすい。

実績：

このようにして成分分解法は既存の処理アルゴリズムの限界をはるかに超える高速性を達成できるので、大量データのバッチ処理を中心に適用実績が積み上がってきている。

2. インメモリとオンディスクの成分分解法

現在のデータベースにはインメモリとオンディスクのものがある。一般にインメモリのそれはより高速であることが期待され、オンディスクのそれはより大規模であることが期待されている。

成分分解法もインメモリ、オンディスクものが考案されている。一般のデータベースと同様、成分分解法においてもインメモリのものは高速でかつ多機能、オンディスクのものは大容量である。下記にその比較表を記す。

	インメモリの成分分解法	オンディスクの成分分解法
ランダムアクセス性能の影響	ランダムアクセスに強いため、アルゴリズム設計の自由度が高い。 そのため多様なアルゴリズムを設計できる。	ランダムアクセスすると極端に性能が低下するため、アルゴリズム設計の自由度が小さい。そのため、実用的なアルゴリズムはたやすく作れない。
容量の影響	容量が強く制約されるため、N gram など大きな容量を使用する方法は実装しにくい。	大容量のため、N gram など大きな容量を使用する方法を実装できる。
揮発性の影響	揮発性があるため、大きなデータの場合、ローディング時間がかかる。100GBのデータをロードするには20分程度かかる。	揮発性がないため、電源投入後直ぐに使用可能である。 使用しない時は電源を落とし節電することもできる。
スケールアウト	難しい。 (ハードウェアによる制約)	易しい。
クラウドでのデータ共有	手間がかかる。 (OSによる制約)	易しい。
コスト	ビット単価が高い。	ビット単価が安い。
省エネ	メモリおよびシステム(CPU等)によるエネルギー消費が大きい。	ディスクだけでデータを保持できるので省エネ性が高い。

上記の比較表にも記されているとおり、インメモリとオンディスクの成分分解法は多くの特性が対照的である。

3. 1/3 構造と 1/3 C (Complementary) 構造の連携

1/3 構造

1/3 構造とはインメモリ環境用の成分分解法で用いられるデータ構造である。成分が3種（順序集合、値番号、値リスト）あり、うち1つ（値リスト）が昇順であることからこのように呼ぶ。

ランダムアクセスが容易なインメモリ環境下では、成分の結合演算が自由にできるため、実際上、すべてのデータベース処理を設計できる。

実際 1/3 構造を用いて作成されたエンジンは非常に多様な処理ができ、生産管理、調達コストの最適化、不整合の検出など幅広い用途で使われている。

1/3 C 構造

1/3 構造に変換するとデータは多くの場合コンパクトになる。しかし元データが TB 級になるとメモリに収容できなくなる。収容できる場合でも、メモリへのローディング時間がかかり不都合である。

このような問題を解決するため、オンディスク環境用に **1/3 C 構造**が開発された。データ保持のための電力もインメモリよりもずっと小さく省エネである。さらにインターネット越しに共有しやすいので、ワールドワイドでの情報発信・情報共有に適する。

1/3 構造と 1/3 C 構造の連携

1/3 構造と 1/3 C 構造はいずれも成分分解法であり、相互の変換は効率的にできる。このことから、以下（図1）のような活用形態が考えられる。

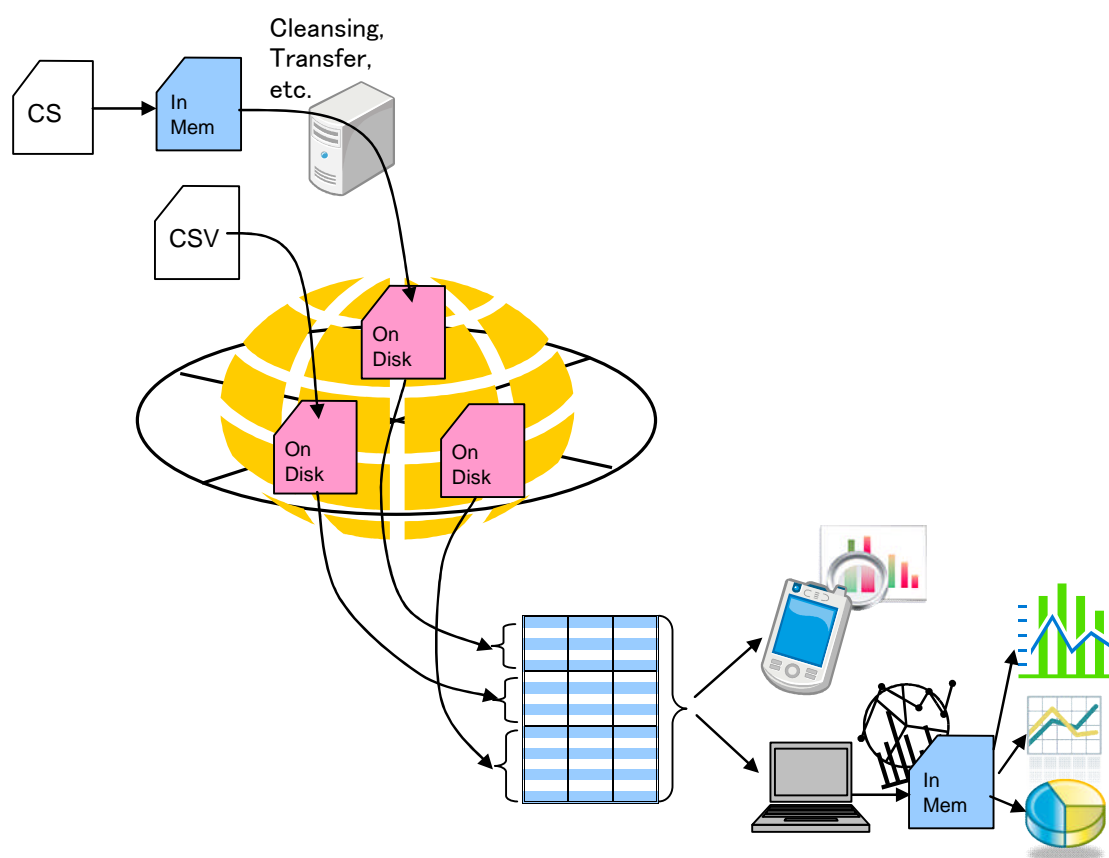


図 2 インメモリ(1/3 構造) および オンディスク(1/3 C 構造) の連携