

分散システム上の Operational Transformation における Confluence Property について

情報科学研究科 2年 佐藤龍之介

発表の流れ

分散システムとは？

OT (Operational Transformation) の仕組み

4ノード以上での Confluence Property

発表の流れ

分散システムとは？

OT (Operational Transformation) の仕組み

4ノード以上での Confluence Property

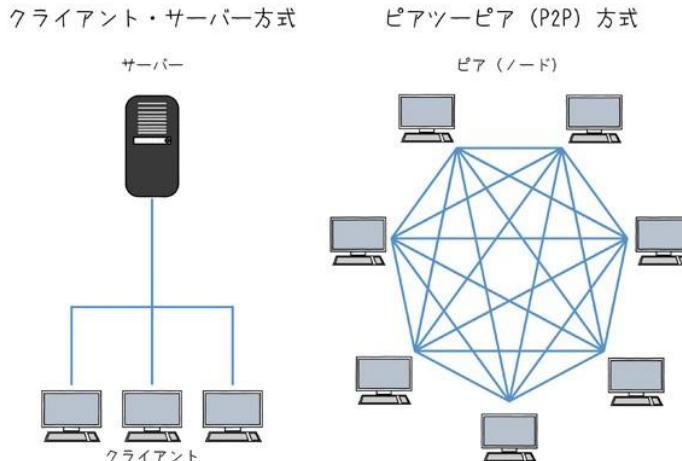
分散システムとは？

複数のコンピュータがネットワーク上で連携して一貫性を保ちながら動作するシステム

目的

- ・拡張性 *Scalability*
- ・透明性 *Transparency*
- ・フォールトトレランス *Fault Tolerance*

種類



引用:<https://www.shoisha.co.jp/book/article/detail/302>

分散システムの例

- Webサービス → クライアントサーバー方式
- クラウドコンピューティング → 両方
- 分散データベース → P2P方式

分散システムの問題点

- Consistency - 一貫性の維持
- Concurrency - 同時に行われる複数の操作のコンフリクト
- Network issues - 通信の切断によるローカルでの操作とその反映

→ この問題を解決する仕組みとして、Operational Transformation がある

発表の流れ

分散システムとは？

OT (Operational Transformation) の仕組み

4ノード以上での Confluence Property

OT (Operational Transformation) とは？

目的：共同編集において文書の一貫性を保つ

背景：複数人でgoogle docsを編集しても共通の編集箇所において、文字の挿入や削除という操作の衝突（コンフリクト）が起きない

OTにおける重要な用語とその定義

OTでは、以下の事項においてそれぞれの表記法を用いる

操作と状態 操作の集合を $O = \{p, q\}$ 、状態を S と表す

操作の適用 操作 p を状態 S に適用すると、次の状態 $S' = S \cdot p$ と表せる

同時性の定義
Concurrency 操作 p と q に因果関係がなく並行であることを $p \# q$ と表す

操作の変換
Transformation Function T 二つの操作 p, q が $p \# q$ であるとき、操作 p を適用した後に q を適用しても、操作 q の後に p を適用しても結果は同じになる $T(p, q) = (p', q')$

収束性の定義
Convergence 全ての複製データ(replicas)は操作の順序に関係なく最終的に同じ状態に到達する

共同編集におけるOTの例（2人のユーザーの場合）

AさんとBさんが共同で一つのGoogle Docsのファイルを開いている。一行目には今日の日付が書いてある。

20250515

AさんとBさんが同時に以下の操作を行う：

p : Aさんが5文字目に「/」を加える → 2025/0515

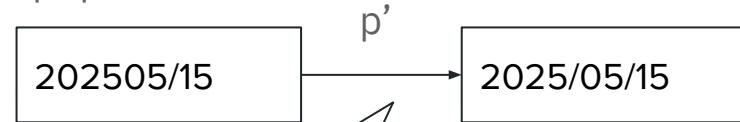
q : Bさんが7文字目に「/」を加える → 202505/15

1. p→qのとき



indexのズレの
調整が自動的に
行われる

2. q→pのとき



操作qによる
indexのズレは
右側のみ

複製において、OTは以下の4つの性質を満たしている

Local Confluence Property … 2つのノード間でどの順番で操作を適用しても、同じ最終結果になる

Minimal Property … 必要最小限の修正が行われる

Symmetric Property … 操作の順序に関係なく、それぞれの変換が対照的である

Compositional Property … 一つの小さな操作をまとめても結果が変わらない

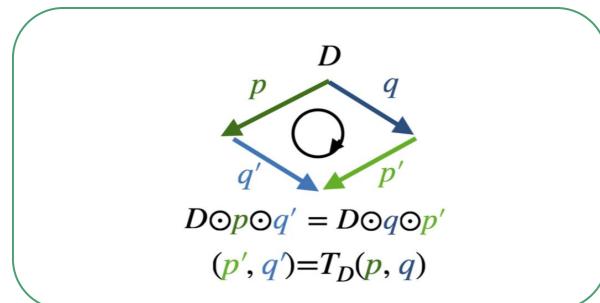


Fig1. Local Confluence Property

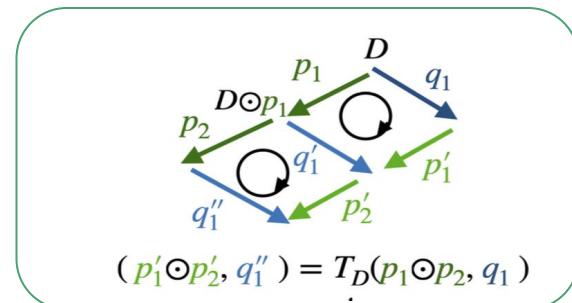


Fig2. Compositional Property

Algebraic Structures of Confluence Property

Identity element $\cdots p\#! = !\#p = p$

Idempotency $\cdots p\#p = p$ (by Minimal Property)

Commutativity $\cdots p\#q = q\#p$ (by TP1-Property)

Associativity $\cdots (p\#q)\#r = p\#(q\#r)$ (by TP2-Property)

合流性の代数的構造

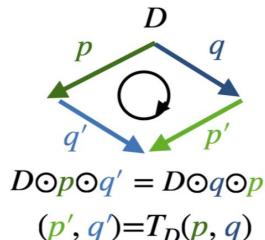


Fig3. TP1 - Property

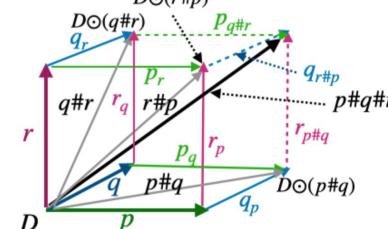


Fig4. TP2 - Property

発表の流れ

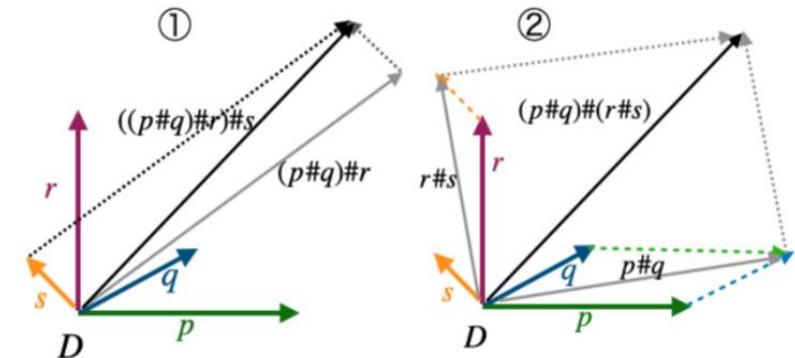
分散システムとは？

OT (Operational Transformation) の仕組み

4ノード以上での Confluence Property

4ノード以上ではConfluence Propertyが成立する?

- Commutativity
- Associativity



CRDTs (Conflict-free Replicated Data Types) とは？

目的：分散システムにおいてデータの更新によるコンフリクトを避けるためのデータ構造

背景：複数人でgoogle docsを編集しても共通の編集箇所において、文字の挿入や削除という操作の衝突（コンフリクト）が起きない

参考文献：

1. Coordination-free Collaborative Replication based on Operational Transformation - Masato Takeichi

研究の方向性

Rocqで分散システムの問題点が
保証されていることを示す