

分散スケッチの重複回避測定方式

1. スケッチとは

■ スケッチとは？

- 確率的データ構造
- 多くのデータフローから推定値を効率的に得るための測定方法

■ なぜ、これを使うのか？

- 小コストでデータフローの測定が可能

■ 有名なスケッチの例（Count-Minスケッチ）

- ハッシュ関数の数（d）・ハッシングの範囲(w)で行列を構成
- データを観測したとき、各ハッシュごとにハッシングし、ハッシュ値を得て、該当するインデックスを+1する
- 特定データxの推定値を得るためには、xをハッシングした値の中で最小値を採用

14

86

5

147

101

⋮

⋮

→

Hash1(86) = 4

Hash2(86) = 2

Hash3(86) = 7

Hash4(86) = 1

↓

hash1	0	0	0	2	0	0	0	0
hash2	0	1	0	0	0	1	0	0
hash3	0	0	0	0	1	0	1	0
hash4	1	0	0	0	0	0	0	1

2. 研究背景

データフローの測定

- ネット上には多くのデータフローが毎日生成され、流れる
- この流れを測定するのは現実的に無理（コストの問題）
- 精度が少し低くてもコストの少ない測定法が欲しい

スケッチの導入

- 測定結果の精度が100%ではないが、小コストでデータフローの測定ができる
- 例えば、ペタバイトのデータフローをメガバイトの測定結果として得ることができる

単一のスケッチ処理ノードを使用すると？

- 全てのデータフローを一つのスケッチ処理ノードに送信するため通信コストが高い
- 単一であるため負荷が集中され、障害が起こると使用不可になる

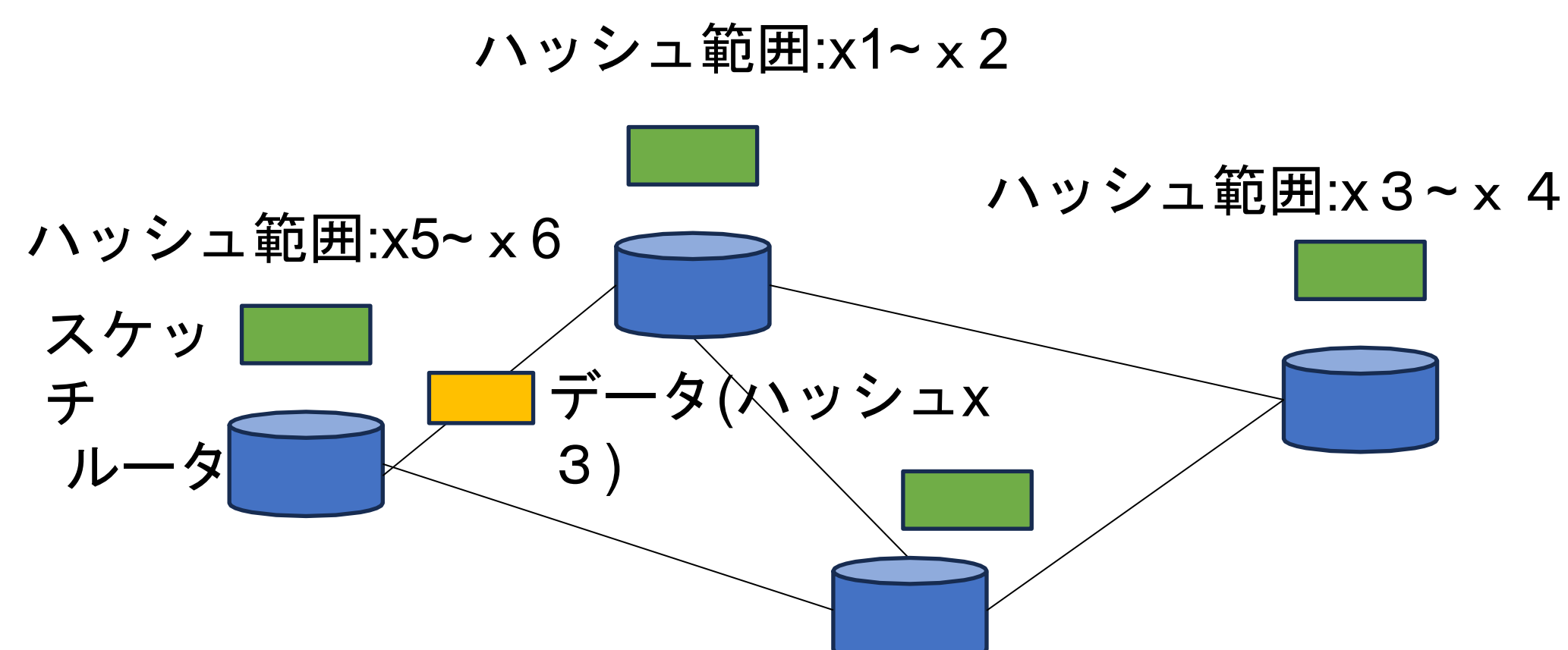
分散してみると？

- スペックの低いスケッチ処理ノードを多数のノードに配置することでコストが下がる
- 負荷分散ができるため、障害が起きても問題ない
- しかし、分散スケッチには分散スケッチならではの問題がある

3. 解きたい問題

■ 分散スケッチの重複問題

- 多数のスケッチ処理ノードを分散して設置した場合、フローを重複して測定してしまう
- 各スケッチ処理ノードにハッシュ空間を用意して、各パケットにはフローIDを付与する。ターゲットのパケットが特定のスケッチ処理ノードのハッシュ範囲に含まれると気のみスケッチを更新することで解決
- 例えば、下の図ではハッシュ範囲x3が右のスケッチが処理するが、データは左のスケッチと中上のスケッチしか測定できない



■ スケッチのカバレッジ問題

- 上記の解決策のため、フロー情報がスケッチ処理ノードを通過しないと測定できない問題が発生
- SDNを用いると解決はできるが、完全分散制御ではない

4. アプローチ

■ ヘッダにフラグを設ける

- ヘッダにフラグを設けて、データフローが測定されているかどうかを確認する
- 複数のスケッチ処理ノードはスケッチ処理されてないデータにのみ処理を行う

■ 最も少ない分散スケッチ処理ノード

- 媒介中心性の高いノードを選定
- 選定したノードにスケッチ処理ノードを配置する
- どれだけデータフローをカバーできるかを観測する