ベンチマーク結果報告

Sum Preservation vs Permutation

ベンチマーク概要

- ツール: Criterion.rs を使用した統計駆動のマイクロベンチマークで実施しました :contentReference[oaicite:0]{index=0}。
- 対象関数:
  - `shuffle_sum_preservation`（コミットメント合計の復号＋加算）
  - `shuffle_permutation`（単純なインデックスコピー）
    :contentReference[oaicite:1]{index=1}。
- 入力サイズ: 100, 1 000, 10 000 要素で比較を行いました :contentReference[oaicite:2]{index=2}。

結果①: **Sum Preservation**

サイズ	実行時間 **(99% CI)**	スループット
100	[850.65 µs 851.04 µs 851.52 µs]	[117.44 K elem/s 117.50 K 117.56 K] :contentReference[oaicite:3]{index=3}
1000	[8.5057 ms 8.5097 ms 8.5144 ms]	[117.45 K elem/s 117.51 K 117.57 K] :contentReference[oaicite:4]{index=4}
10000	[85.126 ms 85.228 ms 85.364 ms]	[117.15 K elem/s 117.33 K 117.47 K] :contentReference[oaicite:5]{index=5}

- Sum Preservation は各要素を復号し加算するため処理コストが高い設計です :contentReference[oaicite:6]{index=6}。

結果②: **Permutation**

サイズ	実行時間 **(99% CI)**	スループット
100	[189.21 ns 189.32 ns 189.43 ns]	[527.89 M elem/s 528.22 M 528.53 M]
1 000	[1.7406 µs 1.7452 µs 1.7508 µs]	[571.17 M elem/s 572.99 M 574.51 M]
10 000	[34.985 µs 34.996 µs 35.009 µs]	[285.64 M elem/s 285.75 M 285.84 M]

- Permutation は要素の単純コピーのみで、非常に高速かつ軽量な実装です :contentReference[oaicite:7]{index=7}。

規模依存性

- 両手法ともアルゴリズムとしては **O(N)** の逐次処理ですが、
  - `sum_preservation`: 復号＋累積加算が入るため要素数増加に伴い線形に実行時間が増加します :contentReference[oaicite:8]{index=8}。
  - `permutation`: 配列コピーのみで定数乗のコストしか伴わず、N 増加でも相対的に高速を維持します :contentReference[oaicite:9]{index=9}。

外れ値検出

- Criterion.rs は Tukey 箱ひげ図に基づく手法で、1.5×IQR 以上を「mild outlier」、3×IQR 以上を「severe outlier」として検出します :contentReference[oaicite:10]{index=10}。
- ベンチマーク中、環境ノイズ（スケジューリング遅延等）による外れ値が数%〜16% 見られたため、静かなマシンでの再実行を推奨します :contentReference[oaicite:11]{index=11}。

まとめ & 推奨

- 合計不変性検証（**Sum Preservation**）
  - 暗号復号＋加算込みで約0.8 ms/100要素 → 85 ms/10 000要素
  - スループット: 約117 K elem/s
- 順序置換検証（**Permutation**）
  - 単純コピーのみで約0.19 µs/100要素 → 35 µs/10 000要素
  - スループット: 数百 M elem/s
- 選択基準:
  - 性能重視かつ改竄リスクを許容可能なら **Permutation** を推奨
  - 完全性担保が必須なら **Sum Preservation + Permutation** 証明 の組み合わせを検討

簡単まとめ

- **Permutation**
  - 処理内容：要素の並べ替え（コピー）
  - 複雑度：O(N)／要素あたり数十ナノ秒
  - ベンチスループット：数百 Melem/s
- **Sum Preservation**
  - 処理内容：復号＋群演算（加算）
  - 複雑度：O(N)・暗号コスト付き
  - ベンチスループット：数十 Kelem/s

よって、「Permutation** のほうが早い」**と言えます。

以上の結果を踏まえ、用途に応じた手法選択をご検討ください。
::contentReference[oaicite:12]{index=12}