Session 8 : Privacy and Security

柿澤美穂（お茶大）

Processing Analytical Queries over Encrypted Data (Stephen Tuほか)

- 暗号化データに対する分析クエリの処理
  - “MONOMI”の提案
    - 信頼されていないDBサーバで、機密情報に対して安全な分析を施すシステム
    - DB全体を暗号化して、暗号化されたデータにクエリを実行する
  - MONOMIを実現する3つの戦略
    1. 複雑なクエリを、クライアントとサーバで分けて実行する
    2. 最適化のための4つの技術を適用する
       - 行ごとの仮計算
       - 空間効率を良くする暗号化
       - 似た計算をするカラムのグループ化
       - 仮フィルタリング
    3. 「デザイナー」と「プランナー」を確立する

MONOMIのキーポイント

Processing Analytical Queries over Encrypted Data (Stephen Tuほか)

- 「デザイナー」と「プランナー」
  - デザイナー：物理デザインの選択
    - サーバでデータをどのように暗号化するかを決定する
  - プランナー：実行プランの選択
    - 与えられた物理デザインにおいてベストなクエリ実行方法を決定する

入力

物理デザインの決定

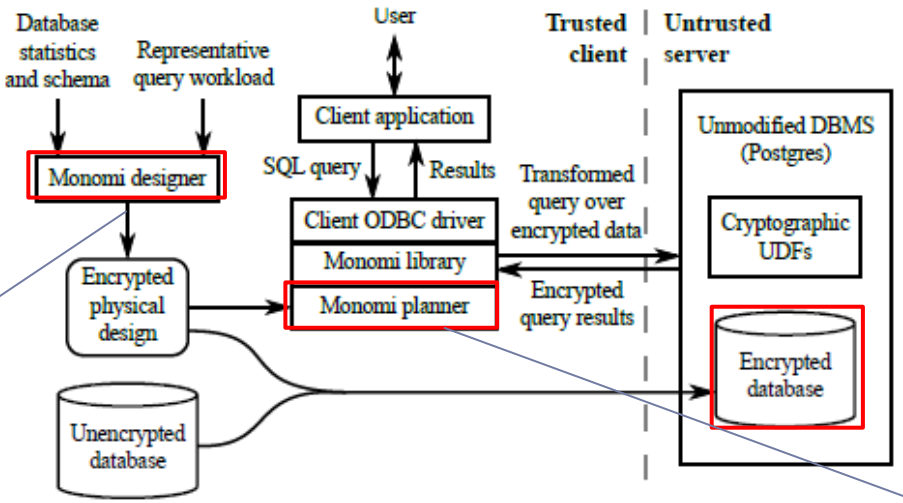

Figure 1: Overall architecture of our MONOMI prototype.

サーバで実行するクエリだけサーバに送られる。残りのクエリは、クライアントで実行する。

ライブラリは、プランナーを使って、クライアントとサーバでの最適クエリ実行プランを決定

※図は論文から参照

Processing Analytical Queries over Encrypted Data (Stephen Tuほか)

- MONOMIの評価
  - 2台の実験用マシン
    - クライアント：MONOMIクライアントライブラリ
    - サーバ：無修正のPostgres8.4
  - 【実験1】MONOMIは、信頼されていないサーバ上で暗号化データに対して効率よくクエリを実行できているか

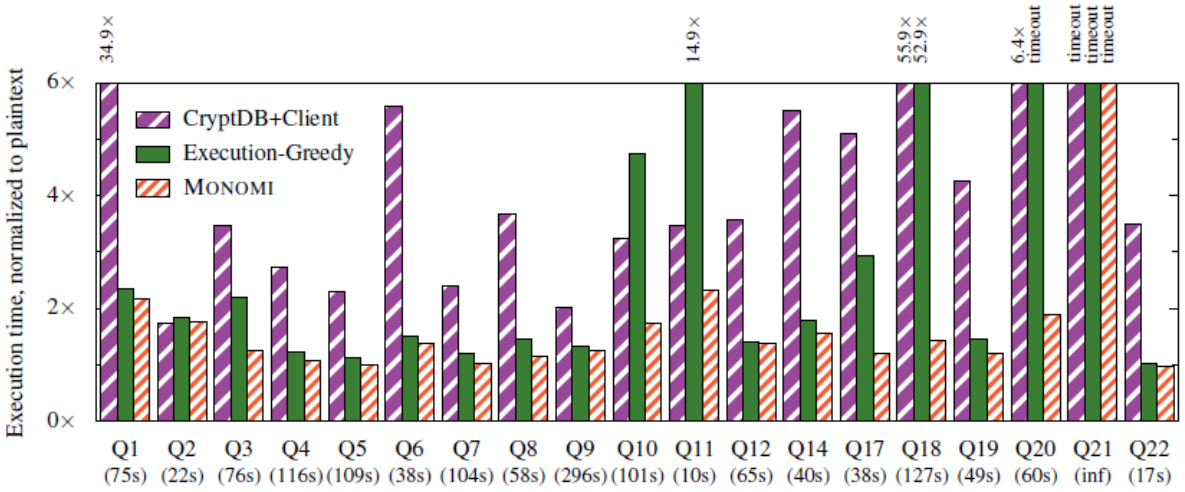

Figure 4: Execution time of TPC-H queries under various systems, normalized to the execution time of plaintext Postgres (shown in parentheses). Query 21 times out on all systems at scale 10 due to correlated subqueries, but incurs a 1.04× overhead with MONOMI relative to plaintext at scale 1.

TPC-Hクエリの実行時間を比較

※図は論文から参照

Processing Analytical Queries over Encrypted Data (Stephen Tuほか)

【実験2】MONOMIのデザイナーの適用効果の検証

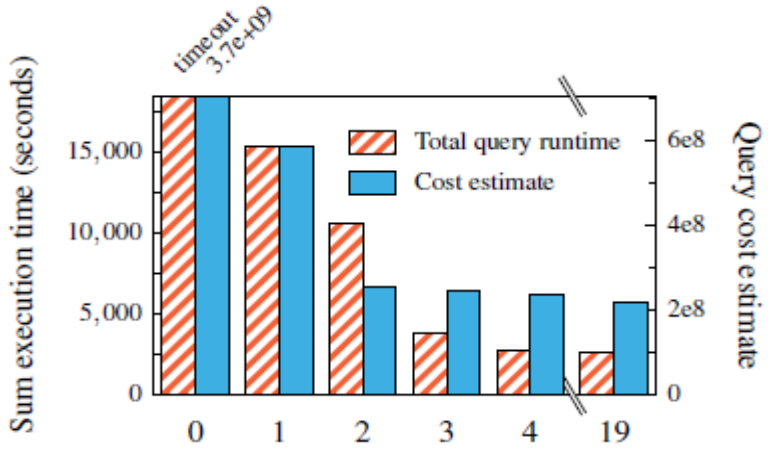

Figure 8: Total TPC-H workload execution time on a physical design chosen using the best $0 \leq k \leq 4$ and $k = 19$ queries as input to MONOMI's designer, along with the cost estimate from MONOMI's designer.

【実験3】MONOMIの消費スペースの測定と、平文処理の場合との比較

System	Size (GB)	Relative to plaintext
Plaintext	17.10	–
CryptDB+Client	71.98	4.21×
Execution-Greedy	32.55	1.90×
MONOMI	29.38	1.72×

Table 2: Server space requirements for the TPC-H workload, under several systems, and the overhead compared to plaintext.

結論

MONOMIは、効率的な物理デザインとクエリ実行プランを選ぶことによって、グリーディーアルゴリズムより効率の良い手法になる。
（ノーマルなPostgresを使った際の1.24倍の実行時間、1.72倍の空間使用で適用可）

※図は論文から参照