【SIGMOD 2013勉強会】

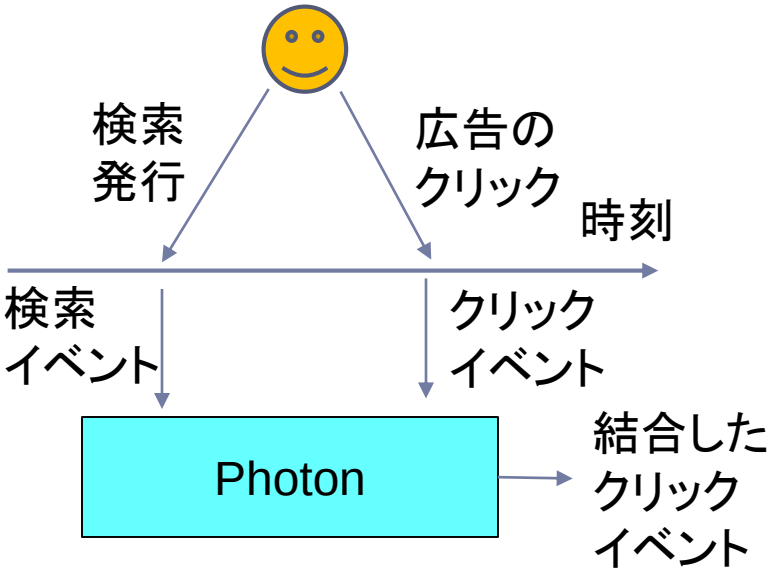
Session 17: Complex Event Processing

担当: 簗井, 姜, 杉浦(名大)

Some figures are copied from SIGMOD 2013 proceedings.

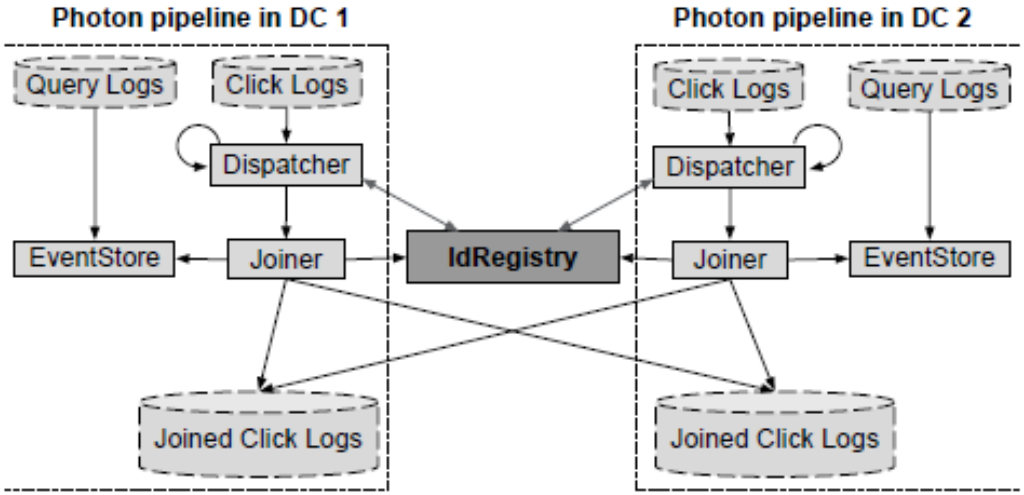
Photon: Fault-tolerant and Scalable Joining of Continuous Data Streams

- Rajagopal Ananthanarayananほか（Google）
- Googleにおける要求：
  - ユーザの検索，広告クリックのログは複数のデータセンターで複製管理：Google File System (GFS) を利用
  - Photonの役割：検索イベントと広告クリックイベントの結合
    - クリックイベントに含まれる検索ID（外部キー）を利用
  - 必要な機能
    - 広告イベントと対応イベントの結合は確実に1回のみ実施
    - フォールトトレランス：大規模な障害にも対応
    - 遅延，非順序処理にも対応



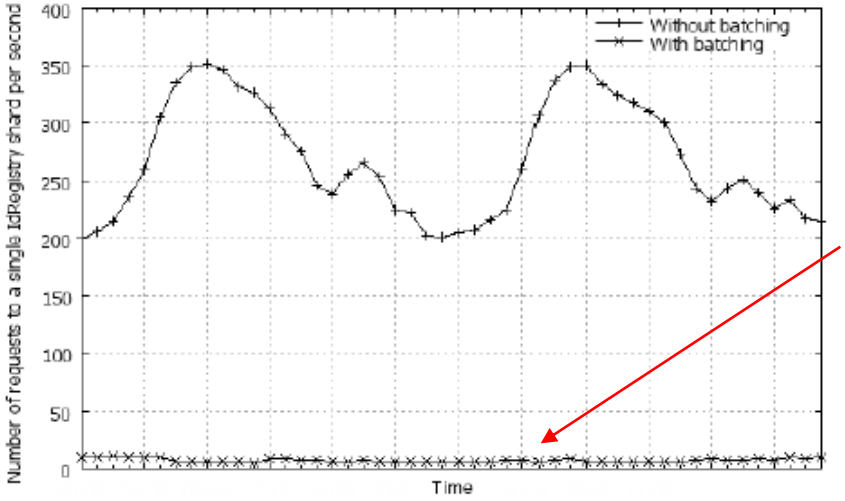
システム技術の特徴

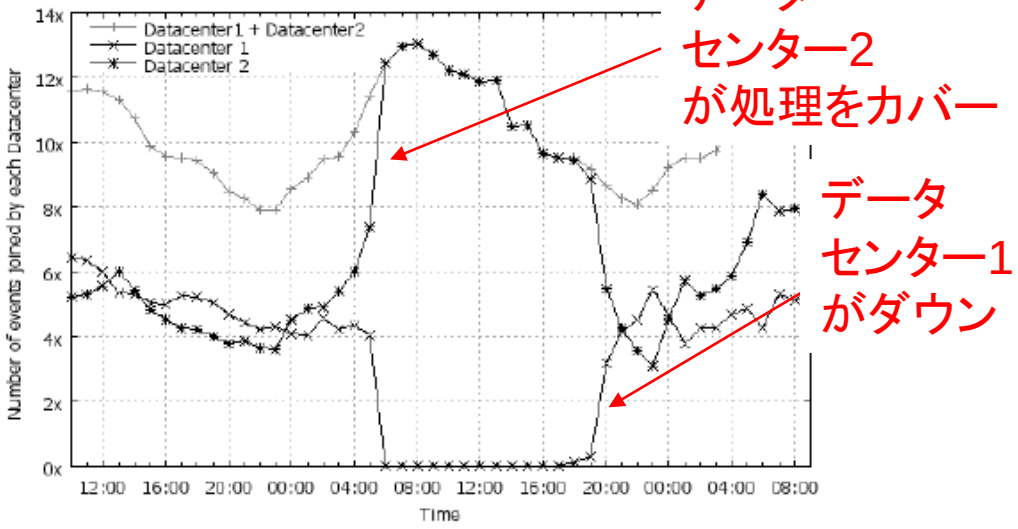
- IDレジストリ
  - 「ただ1回だけ結合」のセマンティクスを実現
    - 結合の際には，ワーカはまずイベントIDをIDレジストリへ登録を試みる：登録されていなければ，次いで結合を実施
    - Paxosアルゴリズム（分散合意形成）を利用
  - サーバサイドのバッチ処理
    - IDレジストリへの多数の登録処理を一つにまとめて効率化
- 処理プロセス
  - IDレジストリで大域的状態を管理
  - EventStoreではインメモリキャッシュ活用



評価結果

- 米国の5データセンター（3つの地域に分散）で運用
- 遅延：90%以上のイベントを7秒以内に結合
- サーバサイドのバッチ処理の効果大（右上図）
  - スループットの大幅な向上
- データセンター障害への対処（右下図）





Utility-Maximizing Event Stream Suppression

- *Di Wang, Yeye He, Elke Rundensteiner, Jeffrey F. Naughton (Worcester Polytechnic Institute, U.Wisconsin-Madison)*
- **研究動機:**
  - 複合イベント処理(CEP)でイベント・ストリームのパターン・マッチを求める時、プライバシー問題は考慮されていない
    - **例: 病院での医療CEPシステム上でのプライバシー問題**

      Q1: SEQ (Exit-Patient-Room, Sanitize) WITHIN 2min

      Q2: SEQ (Wash, Enter-psychiatrist-office) WITHIN 2min

      Q3: SEQ (Exit-patient-room, Enter-psychiatrist-office) WITHIN 5 min

  - 単純な抑制手法で対応すると、ユーティリティは低下
- **研究内容:**

CEP上でプライバシーを保護するとともに、ユーティリティも最大限にするイベントの抑制手法を提案

最大ユーティリティの抑制手法1

- **最適のタイプ・レベルのアルゴリズム** ← *オフライン*
  - ユーティリティの定義

$$\mathcal{U} = \sum_{Q_j \in \mathcal{Q}} w(Q_j) N_T(Q_j) y_j + \sum_{P_k \in \mathcal{P}} w(P_k) N_T(P_k) z_k$$

パブリック(Q)　　プライベート(P)

    - ユーティリティ重み$w(Q_i) > 0$ とペナルティ重み$w(P_j) < 0$ を付ける
    - イベント・ストリーム中パターン・マッチの数$N(P_j)$と$N(Q_i)$ を統計
    - パターンを抑制するかどうかを変数で表示（ $y_j=0$ 抑制、$y_j=1$保存 ） → 抑制策略
  - タイプ・レベルでの抑制
    - ユーティリティを最大にする基準で、イベント・タイプを抑制かどうかを判断

最大ユーティリティの抑制策略は線形最適化問題で求める

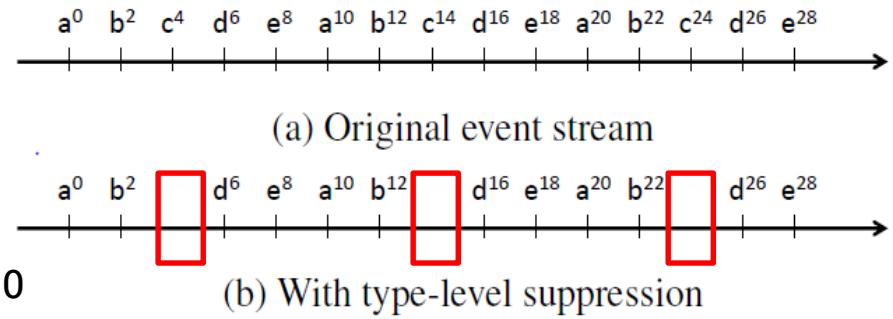
- 例： イベントタイプ集合 Σ = {A,B,C,D,E}

Q1 = SEQ(B,C,D), Window(Q1) = 10, w(Q1) = 5

Q2 = SEQ(A,B), Window(Q2) = 10, w(Q2) = 20

Q3 = SEQ(D,E), Window(Q3) = 10, w(Q3) = 20

P1 = SEQ(A,C,E), Window(P1) = 10, w(P1) = −10



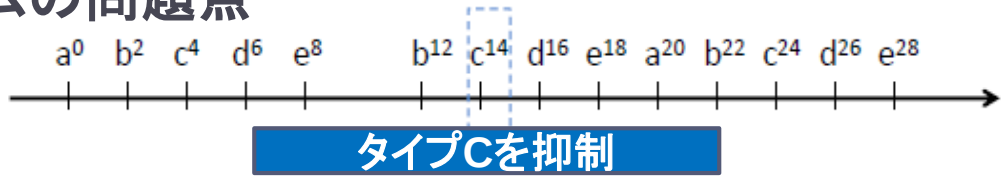
最大ユーティリティの抑制手法2

- ハイブリッドの**実例レベル**のアルゴリズム ← *オンライン*
  - タイプ・レベルのアルゴリズムの問題点

例：P1 = SEQ(A,C,E), w(P1) = −10

Q1 = SEQ(B,C,D), w(Q1) = 5

Q2 = SEQ(A,B), w(Q2) = 20　　Q3 = SEQ(D,E), w(Q3) = 20



  - イベント実例c14がある場合、P1は暴露できない、同時に、Q1はマッチできる
  - 実例レベルでの抑制
    - 今到着したイベントeで部分マッチを得るかどうかを判断　→prefixパターン
    - Est_Match段階で到着予定のイベント実例を推定　→suffixパターン

Suffixパターンの数を推定　***Hybrid***　**タイプ・レベルでの抑制策略を推定として採用**

    - ユーティリティを最大にする基準で、イベント実例を抑制かどうかを判断
  - 例1:a_0 Q2,P1のprefix → Bと(C,E)到着推定→ B,E保存、C抑制→ a_0保存→ Q_2 ○、P_1 ×
  - 例2:c_{14} Q1のprefix (b_{12},c_{14})→ D到着推定→ D保存→ c_{14}保存→ Q_1 ○、P_1 ×

ε-Matching: Event Processing over Noisy Sequences in Real Time

Zheng Li, Tingjian Ge, Cindy X. Chen (U. Massachusetts)

問題設定

入力：ストリームデータ（ノイズあり）
　　イベントの正規表現

目的：ストリームの曖昧さを考慮した正規表現マッチング

- エラーモデルを用いてマッチの曖昧さを計算
- 確率が上位 K 個のマッチを効率的に発見

柔軟な正規表現の実装

- 否定の実装　e.g. $\overline{walk}\ stop^{+}\ run$
- ウィンドウの実装

ストリームデータを高速に処理

曖昧なストリームのマッチ

正規表現：a b

入力：a b a a a …

ノイズを考慮しない場合

a b a a a

- マッチは1つだけ

ノイズを考慮する場合

a b a a a

a b a b a

a b a a b

a a b a a

- マッチは複数存在
- 各マッチが受理される確率を計算

e.g.

$0.8 \times 0.1 = 0.08$

$0.2 \times 0.3 = 0.06$

… etc.

エラーモデル

- 入力ストリームの誤りの傾向をモデル化したもの
- マルコフ連鎖を利用
  - データの正誤は一つ前のデータにのみ依存して決まる

定常分布と遷移行列

左の例の場合

定常分布　$\vec{\lambda} = (0.8, 0.2)$

遷移行列　$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{bmatrix}$

a b a a a　$0.8 \times 0.9 = 0.72$

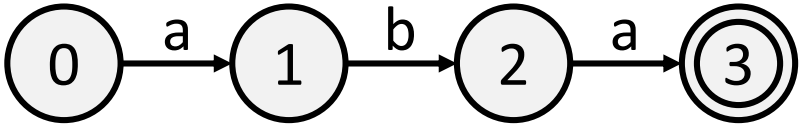
a b a b a　$0.8 \times 0.1 = 0.08$

a a b a a　$0.2 \times 0.3 = 0.06$

ε-Matching アルゴリズム

正規表現：a b a

ATA：



エラーモデル：

$\lambda = (0.8, 0.2)$

$P = \begin{bmatrix} 0.9 & 0.1 \\ 0.7 & 0.3 \end{bmatrix}, P^2 = \begin{bmatrix} 0.85 & 0.15 \\ 0.75 & 0.25 \end{bmatrix}$

マッチング パスの構築

エラーモデルで
パスの確率を計算

状態＼入力	a	b	b
1	(1, 0.8)	(2, 0.2)	(3, 0.2)
2		(2, 0.8 * 0.9)	(3, 0.8 * 0.85) (3, 0.2 * 0.7)
3			(3, 0.8 * 0.9 * 0.1)

…

どの時刻どの入力で
状態遷移したのかを保持