【ICDE2015勉強会】

Research Session 3-1: Distributed Storage and processing

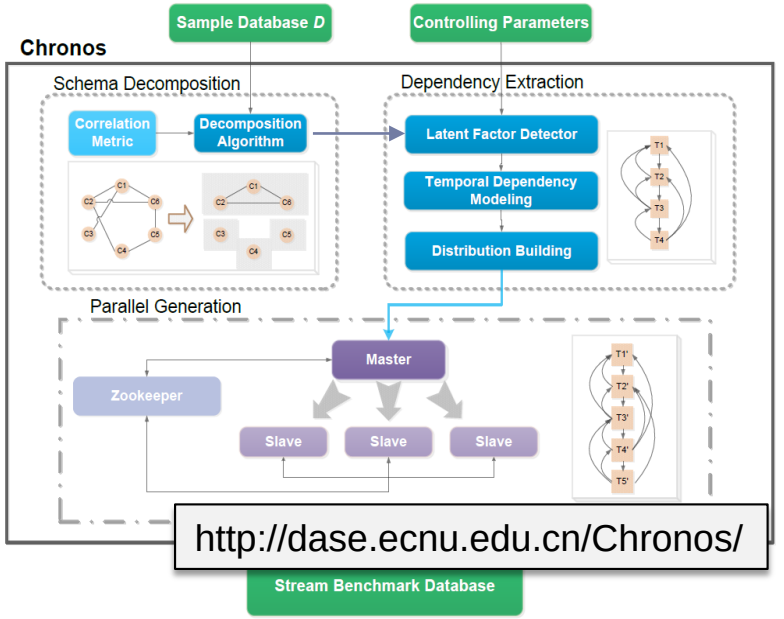
担当：山口晃広（名古屋大学，東芝）

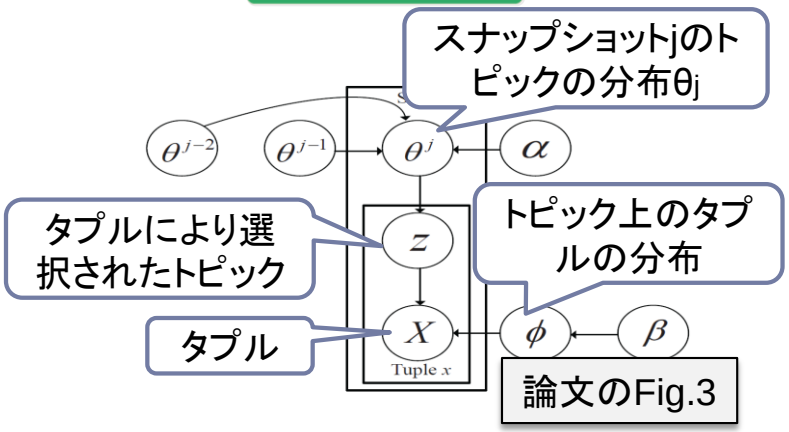
Chronos: An Elastic Parallel Framework for Stream Benchmark Generation and Simulation

- 背景・問題点
  - ストリーム処理の入力タプルを生成するシミュレータが必要
    - ストリーム処理システム（Spark, Twitter Storm, etc）では，入力データの急激な増加に対して，テストを行うことが重要
  - 従来のシミュレータでは，以下に対応できない
    - 実データのように，タプル間の時間的な依存を模擬できない
    - ストレージ/通信のオーバーヘッドで生成レートを正しく模擬できない
- この論文の提案
  - ストリーム処理における実データに近い入力タプルを高速に生成するシミュレータ
    - 現実のストリームに対して，タプル間及びタプル内の列間における関連性，同一のスナップショットにおけるタプルの値の傾向を保つ
    - 入力データ量が増えてもシステムを並列化し，ロードバランスすることで効率的にタプルを生成する

提案されているフレームワーク

- 前提
  - タプルの全属性を離散化→有限個のcellで管理
- (1) Schema Decomposition
  - 関連性が低い列間から分解していき列コンボを構成
    - その列間の独立性をカイ2乗検定し，そのp値を用いる
  - 全ての列コンボのcellサイズが閾値を下回るまで繰返
- (2) Dependency Extraction
  - 各列コンボに対して，個別に以下を実施
  - タプルに潜在的なトピックがあると考え2段方式で推定
    - タプルの観測からトピックに関する潜在変数をLDAで推定
    - 得られたθjから線形回帰式を用いて，トピック間の関連（つまり式中のw値）を推定
- (3) Parallel Generation
  - 各スナップショットに対して，マスターノードは，各スレーブノードが生成するタプル量を決めて，（2）で推定された確率モデルからタプルのcellの確率を算出し，それらをzookeeperに書き込む
  - 各スレーブノードは，それに基づきタプルを生成し，進捗報告をzookeeperに書き込む
  - マスターノードは，その進捗報告から各スレーブノードの処理量をロードバランスするように決める





$$\theta^j = w_0 \mathrm{Dir}(\alpha) + w_1 \theta^{j-1} + \ldots + w_\delta \theta^{j-\delta}$$

論文の線形回帰式

評価：実データに対する近似

評価環境

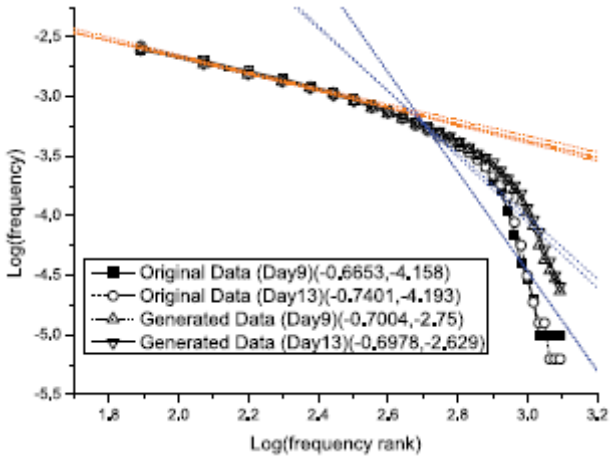
株売買，電話ログ，Web広告

IBM X5 3950 rack servers ×２

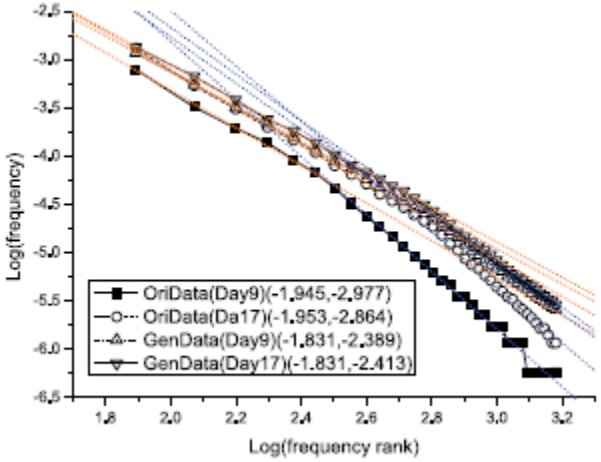
	Stock	Telecom	Advertising
# of Columns	20	18	10
Num. Columns	2	0	0
Cat. Columns	18	18	10
# of records	2,675,788	2,856,181	40,205,029
Data Size	368 MB	276 MB	370 MB
Time Duration	16 Days	12 Days	16 Days

タプル間の関連性に対して多くの評価を実施

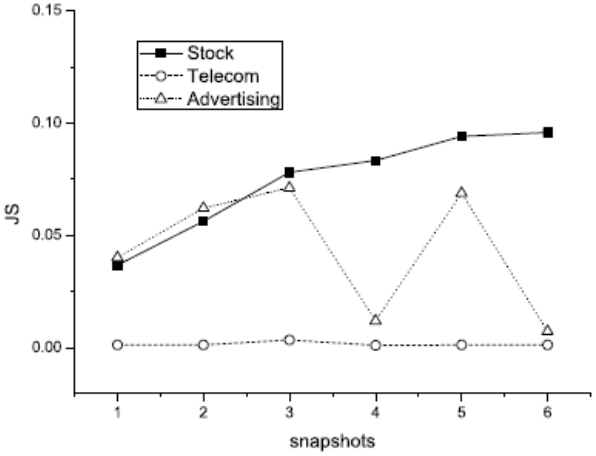
- 各Cellの出現頻度による評価（下図のa,b）
  - 各Cellを出現頻度でソートしlog-logをとってプロット→実データと類似した曲線
  - この曲線の上半分と下半分の傾きを実データと比較→実データに十分近い
- Jensen-Shannon divergence（JS）による比較（下図のc）
  - JSの値域[0,1]に対して，いずれも0.1以下であるため，実データに十分近い



(a) 株売買のCell出現頻度

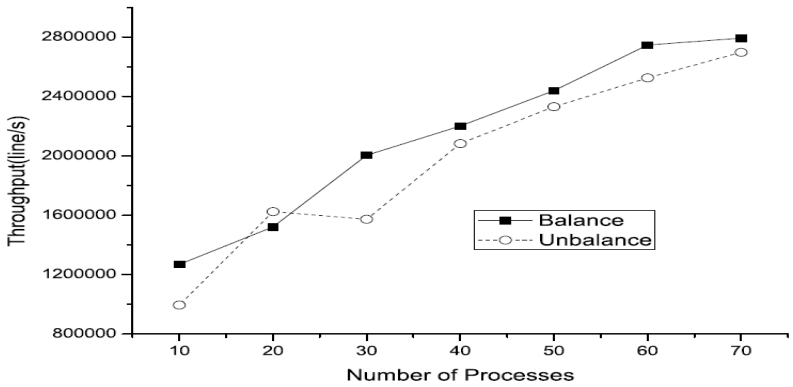


(b) Web広告のCell出現頻度

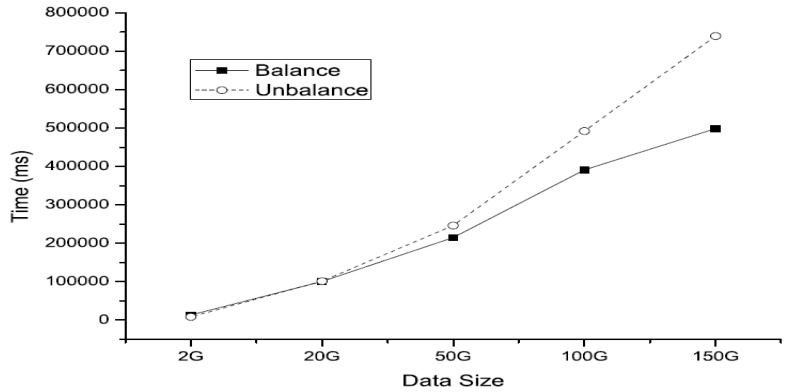


(c) 全アプリのJS

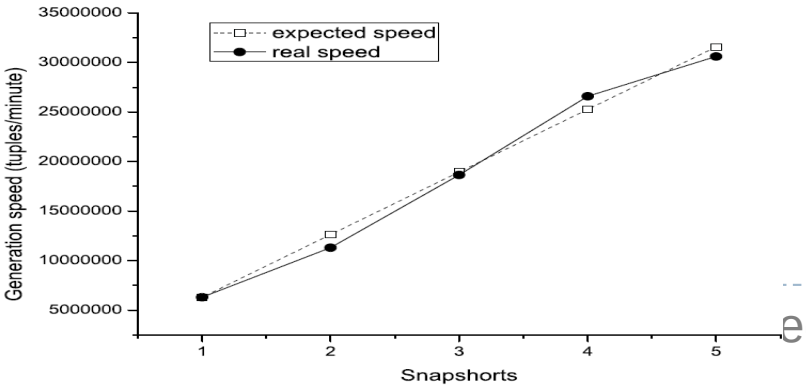
評価：タプル生成の並列処理



- 並列化するノード数を増やしたときのスループットの変化
→ ロードバランスすることで，10%スループットを向上



- 生成されるデータ量を増やしたとき処理が完了するまでの時間
→ 高負荷時になるほどロードバランスの有効性が高い



- 1分毎のスナップショットで，タプルを生成するスピードを600万→3000万タプルまで順に大きくした場合の処理時間
→ 期待したように処理時間が推移
→ スレーブノード数を適切に決めることで，処理時間の調整は難しくない