Shark: SQL and Rich Analytics at Scale

Reynold S Xin, Josh Rosen, Matei Zaharia,
Michael J Franklin, Scott Shenker, Ion Stoica

UC Berkeley

担当: 小山田 昌史 (NEC)

背景・概要

Spark

- UCB AMPLab 開発の大規模データ処理基盤
- (Hadoop と比べての) 特徴
  - ✓ **インメモリ処理** (Hadoop は二次記憶の HDFS 経由)
  - ✓ **記述力の高さ** (Scala の高階関数で**繰り返し処理**も自然に)
  - ✓ **RDD**というデータ構造により**再計算ベースの耐障害性**を実現

Shark

- Spark を利用してつくられた **Hive 互換の DWH**
  - ✓ Hive: Hadoop を利用して作られた DWH
- **SQL と Spark の融合**で高度な分析が簡単にできる

この論文: Shark の学術的な面白さ・工夫を色々と説明

Shark の面白さ・工夫

1. カラム型インメモリDBの実装

- カラム圧縮などのカラムストア技術を利用

2. 動的なクエリ最適化 次ページで説明

- クエリの実行途中に得られた情報をつかい、そのクエリを再び最適化 (revise)

3. Spark の性質がうまく生きる

- Spark のタスクは Hadoop と比べ細粒度
  - ✓ 担当するデータ量は少なく、起動は高速
- 細かくデータをタスクにアサインでき、データの skew にうまく対応できる

実行時の動的なクエリ最適化

モチベーション

- 高度なクエリ最適化: DB内の**データの統計情報**が必要
- Hive や Spark: **まだロード (ETL) されていない生データ** (fresh data) に対するクエリが多い
  - ✓ ロードしながらクエリ処理
  - ✓ データの統計情報がない → **高度なクエリ最適化ができない**

Shark のクエリ最適化

- **クエリ実行前:** ルールベースの単純な最適化のみ
- **クエリ実行中:** 実行時に得られた情報をつかってクエリを動的に書き換えていく
  - ✓ 頻出レコード、データ分布 / skew、オペレータ選択率

実験: 静的な最適化と比べ 3 倍の性能向上

所感

論文の感想

- ほとんどは既存研究
  - ✓だが、実用的なシステムに実装され、効果が出ている
- 読んでいて楽しい
  - ✓細かいところまでよく書かれている

余談 (by AMPLab 留学の鈴木さん@NEC)

- この論文の著者は「Shark は論文にできるのか？」と不安がっていたらしい
- UCB AMPLab の教授陣は「研究は別に論文にしなくともよい」というスタンスらしい
  - ✓実際にはトップ会議が AMPLab 無双に……