On Optimizing Operator Fusion Plans for Large-Scale Machine Learning in SystemML

Matthias Boehm¹, Berthold Reinwald¹, Dylan Hutchison², Prithviraj Sen³, Alexandre Evfimievski⁴, Niketan Pansare¹

VLDB 2018 Session "DB Techniques for ML" ¹ IBM Research – Almaden; San Jose, CA, USA ² University of Washington; Seattle, WA, USA Contact: Matthias Boehm mboehm7@gmail.com

Motivation/Problem

Fusion Opportunities
- State-of-the-art ML Systems
- DAGs of linear algebra (LA) operations and statistical functions
- Materialized intermediates ⇒ ubiquitous fusion opportunities

Optimizing Fusion Plans
- Problem: Fusion heuristics ⇒ poor plans for complex DAGs (cost/structure), sparsity, and local/distributed ops
- Materialization Points
- Sparsity Exploitation
- Constraints (e.g., memory budget and block sizes)

Candidate Selection

Plan Partitions and Interesting Points
- Determine Plan Partitions
- Materialization points M
- Connected components of fusion plans
- Root and input nodes

Compilation Algorithm MPSkipEnum
- Basic Enumeration
  - Linearize search space: from t to s
  - Evaluate and cost plans
- Cost-Based Pruning
  - Opening: evaluate FA and FNR heuristics first
  - Upper bound: cost Ct of best plan s
- Structural Pruning
  - Observation: Assignments can create independent sub problems
  - Build reachability graph to determine cut sets
  - During enum: probe cut sets, recursive enum, combine, and skip

Candidate Exploration

Memo Table
- Partial Fusion Plans (candidates)
- Memo Table Entry (type, {i1, ..., ik}, closed)

Open-Fuse-Merge-Close
- Template Fusion API
- Open new template
- Fuse/Merge open template
- Close open template
- OFMC Exploration Algorithm
- Bottom-up exploration (single-pass, template-agnostic)
- Linear space and time (O(2|E| * |T|) per node, but ternary ops / 4 templates)

System Architecture

Compiler Overview

Runtime Integration
- Templates
  - Cell, MAgg, Outer, Row
  - Template Skeleton
  - Data access, blocking
  - Multi-threading
  - Final aggregation

Plan Cache
- Enforces inclusion of sub problems
- Enables pruning of search space

Operational Performance

L2SVM End-to-End
- Data Base Gen FA FNR
  - 10^4 x 10, D
  - 10^4 x 10, S

ALS-CG End-to-End
- Data Base Gen FA FNR
  - 10^4 x 10, S (0.01)
  - 10^4 x 10, S (0.01)

Experiments

Experimental Setting
- Cluster setup
  - 1-6 node cluster (head 2x4 Intel Xeon E5530, 64GB RAM; 6 workers 2x6 Intel Xeon E5-2440, 96GB RAM, peak 2x32GB/s 2x115GFLOP/s, 10G Eth)  
  - Spark 2.2.2, 6 executors (24 cores, 65GB), 35GB driver
- Baselines
  - SystemML 1.0.1 + (Feb'18): Base, Fused*, Gen (opt), heuristics: FA, FNR
  - Julia 0.6.2 (Dec'17): Julia (w/o fusion), JuliaGen (fusion via dot syntax)
  - TensorFlow 1.5 (Jan'18): TF (w/o fusion), and TFGen (fusion via XLA)

Data Dimensions Sparsity
- Airline78: 14,462,943 x 29 0.73
- Mnist80: 8,106,000 x 784 0.25
- Netflix: 480,189 x 17,770 0.012
- Amazon: 8,026,324 x 2,339,066 0.000012