DATA ANALYTICS USING DEEP LEARNING

GT 8803

FALL 2018

POOJA BHANDARY

TOWARDS A UNIFIED ARCHITECTURE FOR IN-RDBMS ANALYTICS

CREATING THE NEXT®

TODAY'S PAPER

- SIGMOD 2012
  - In-RDBMS Analytics
- Hazy project at the Department of Computer Science, University of Wisconsin, Madison.

TODAY'S AGENDA

- Motivation
- Problem Overview
- Key Idea
- Technical Details
- Experiments
- Discussion

MOTIVATION

Advanced Analytics

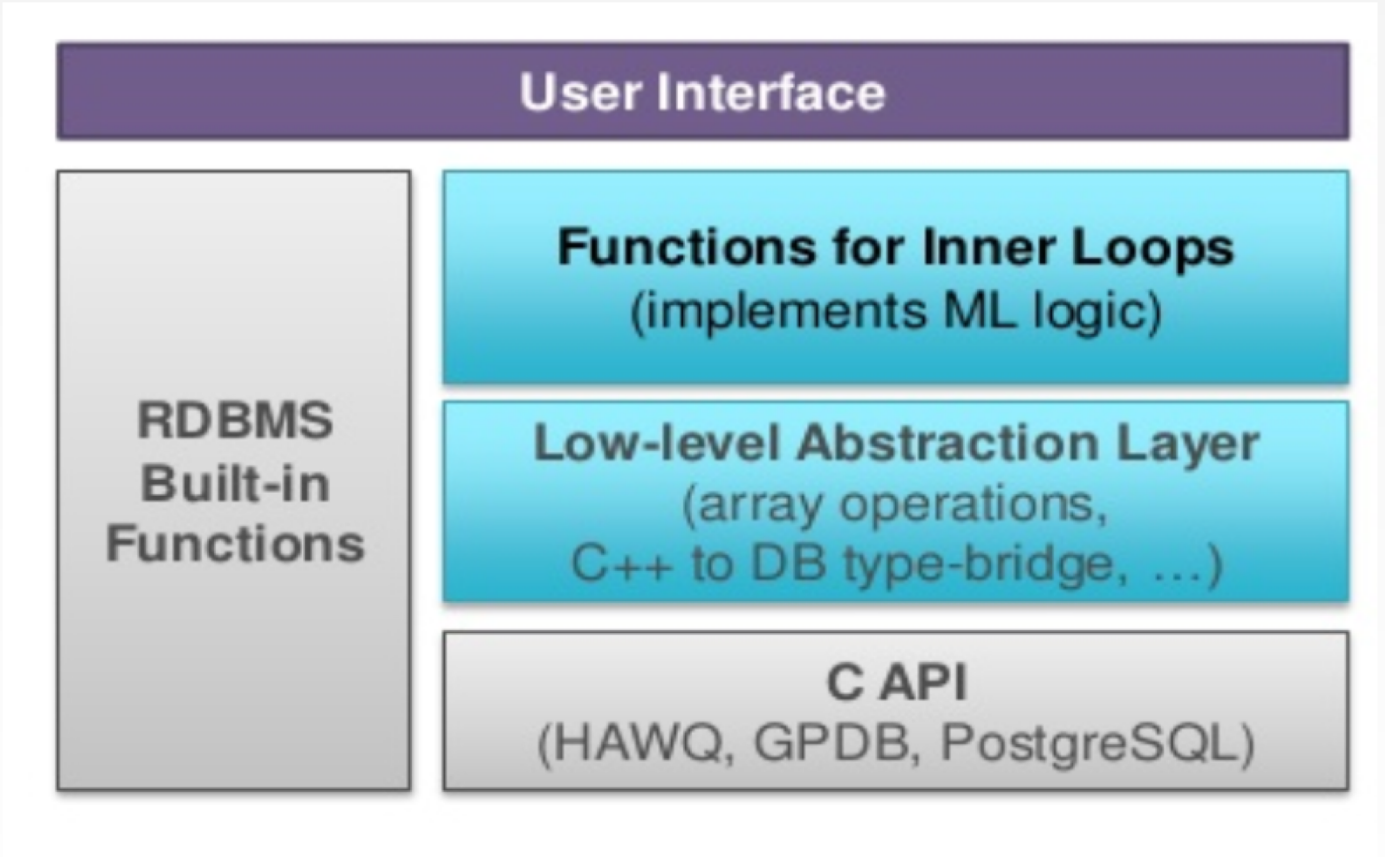
Classification

Recommendation

PROBLEM OVERVIEW

- Ad hoc development cycle for incorporating new analytical tasks.
- Performance optimization on a per module basis.
- Limited code reusability.

IN-RDBMS ANALYTICS ARCHITECTURE



HIGH LEVEL IDEA

- Devise a unified architecture that is capable of processing multiple data analytics techniques.

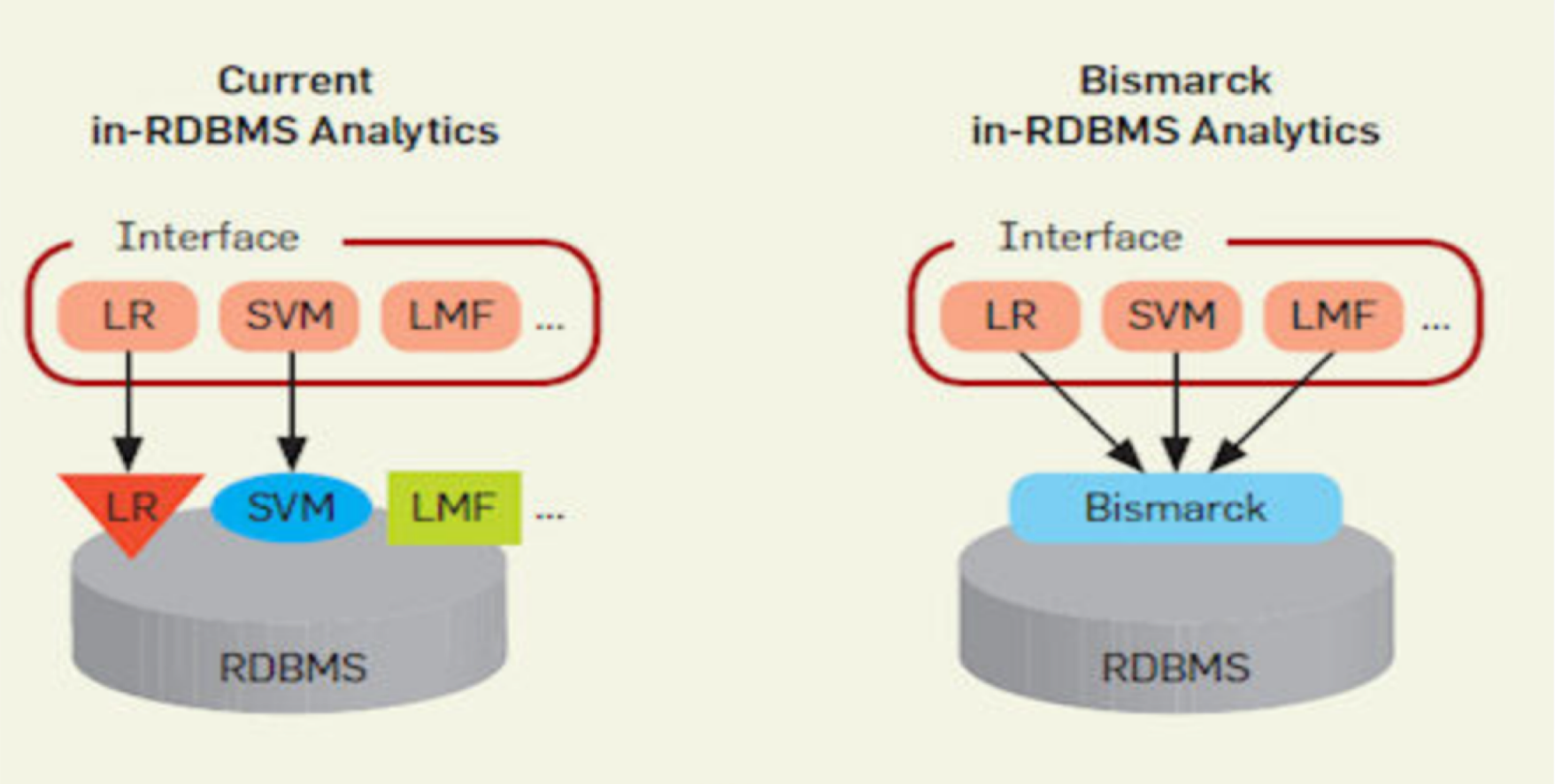
- Frame analytical tasks using Convex Programming.

MAIN CONTRIBUTIONS

- Bismarck

- Identification of factors that impact performance and suggesting relevant optimizations.

BISMARCK


Current
in-RDBMS Analytics
Interface
LR
SVM
LMF
...
LR
SVM
LMF
...
RDBMS
Bismarck
in-RDBMS Analytics
Interface
LR
SVM
LMF
...
Bismarck
RDBMS

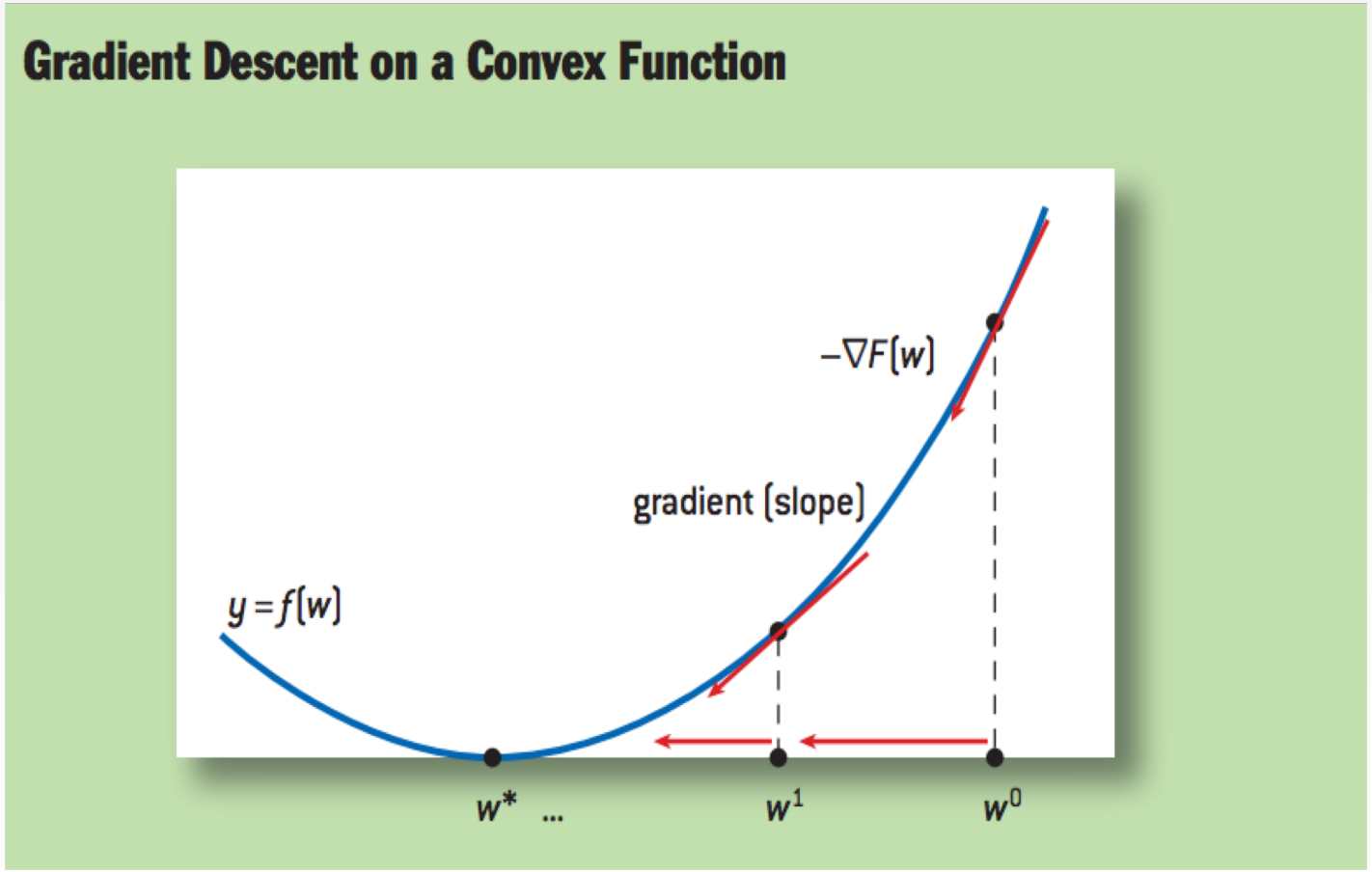
Convex Optimization

$$\min_{w \in \mathbb{R}^d} \quad \sum_{i=1}^{N} f(w, z_i) + P(w)$$

What is convex ?
Linear Regression, Linear SVM
Kernel SVMs, Logistic Regression,

What is not convex ?
Graph mining, Deep Learning

GRADIENT DESCENT



INCREMENTAL GRADIENT DESCENT

- $w^{(k+1)} = w^k \quad - \quad \alpha_k \nabla \mathcal{F}(w^k, z_i)$

INCREMENTAL GRADIENT DESCENT

- Data-access properties are amenable to an efficient in-RDBMS implementation.

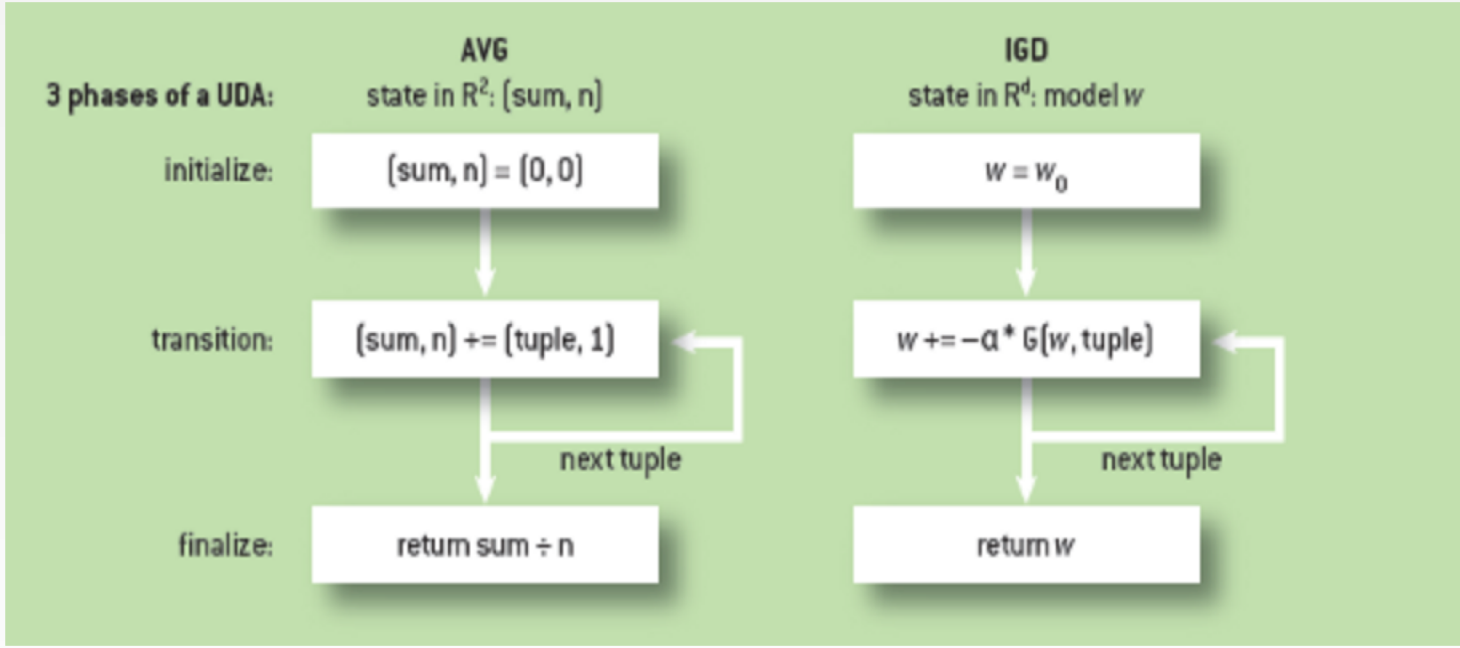
- IGD approximates the full gradient ∇F using only one term at a time.

TECHNICAL APPROACH

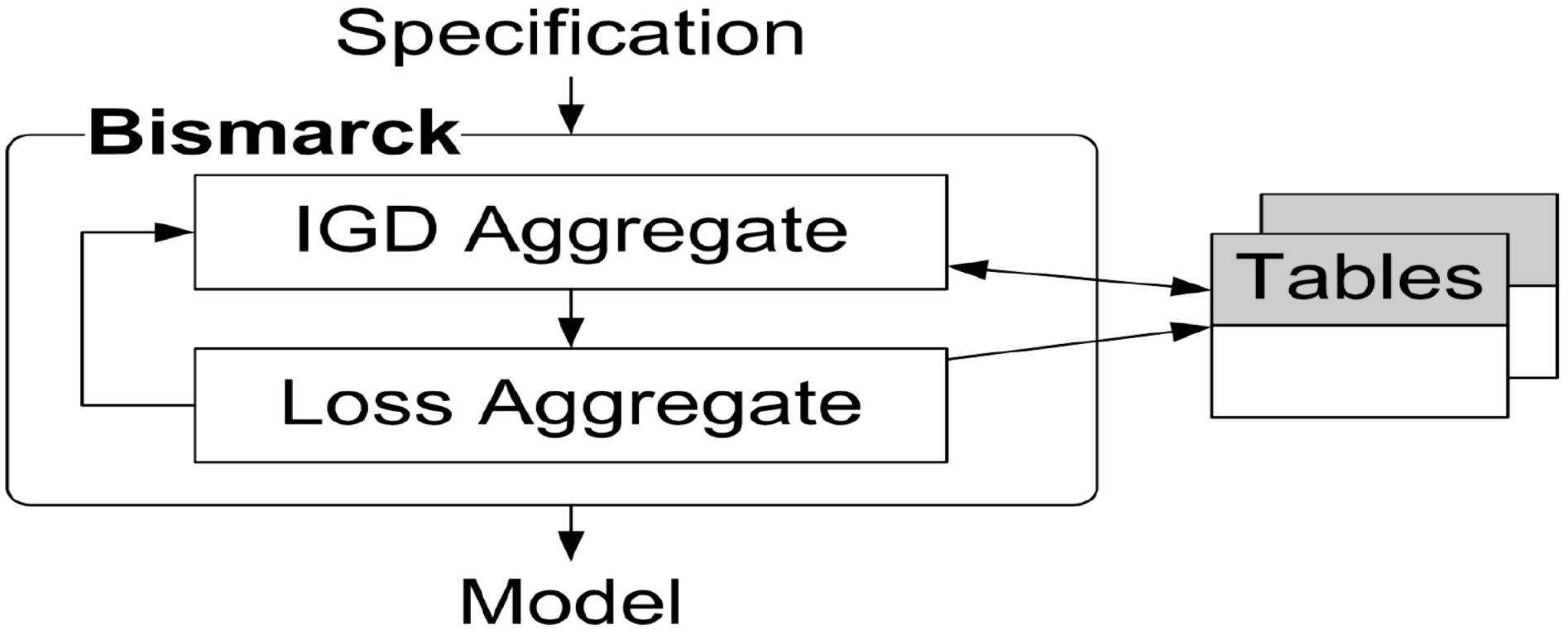
- IGD can be implemented using a classic RDBMS abstraction called a UDA (*user-defined aggregate*).

USER DEFINED AGGREGATE(UDA)

- Initialize
- Transition
- Finalize



Bismarck Architecture



PERFORMANCE OPTIMIZATIONS

- Data Ordering
- Parallelizing Gradient Computations
- Avoiding Shuffling Overhead

DATA ORDERING

- Data is often clustered in RDBMSs which could lead to slower convergence time.
- Shuffling at every epoch can be computationally expensive.

Solution: Shuffle once

PARALLELIZING GRADIENT COMPUTATIONS

- **Pure UDA - Shared Nothing**

Requires a merge function. Can lead to sub-optimal run time results.

- **Shared-Memory UDA**

Implemented in the user space. The model to be learned is maintained in shared memory and is concurrently updated by parallel threads.

AVOIDING SHUFFLING OVERHEAD

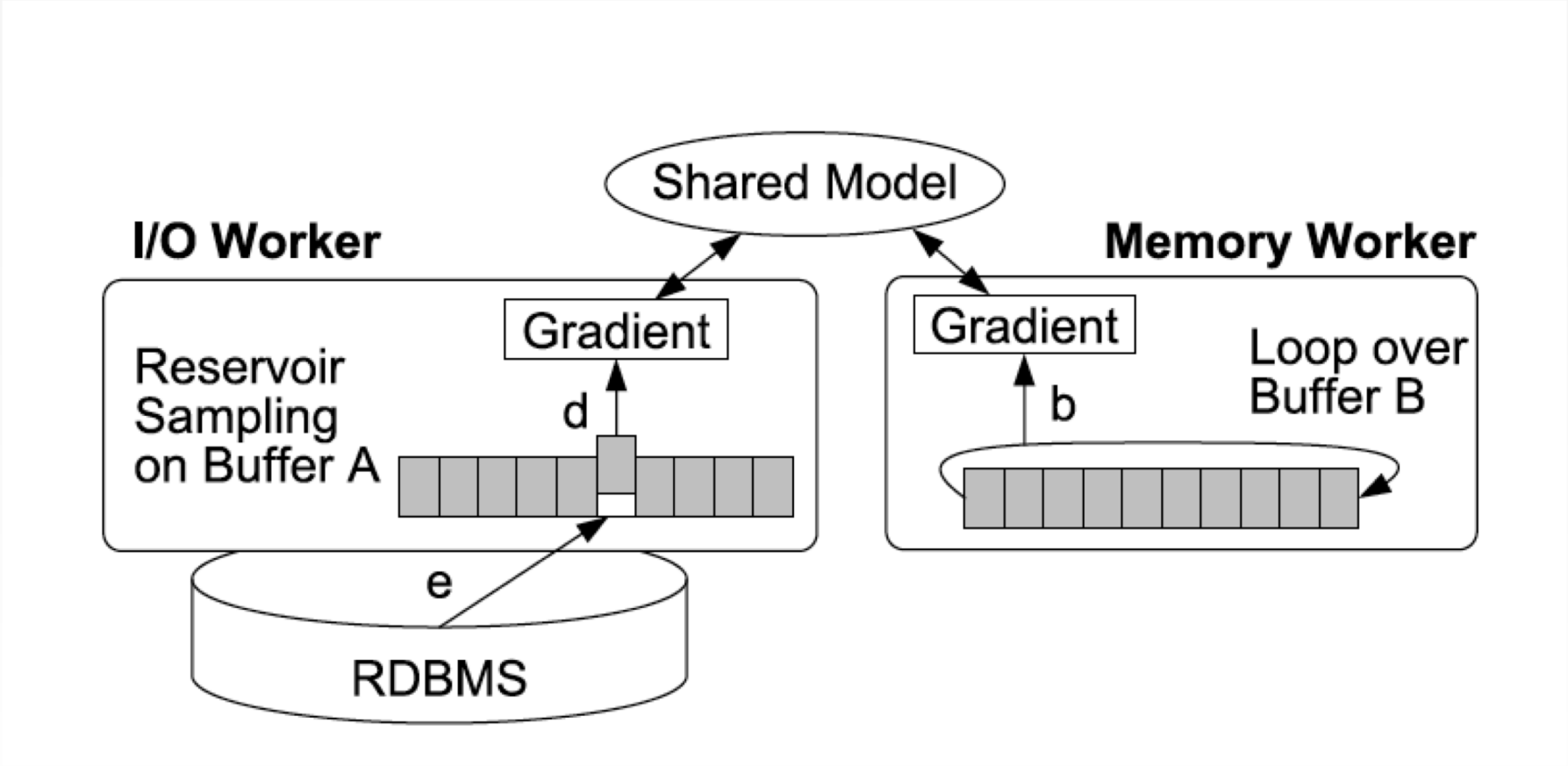
- Shuffling once might not be feasible for very large datasets.
- Straightforward reservoir sampling could lead to slower convergence rate by discarding data items that could lead to a faster convergence.

MULTIPLEXED RESERVOIR SAMPLING

- Combines the reservoir sampling idea with the concurrent update model.

- Combine or multiplex, gradient steps over both the reservoir sample and the data that is not put in the reservoir buffer

MULTIPLEXED RESERVOIR SAMPLING



EVALUATION

1) Implement Bismarck over PostrgreSQL and two other commercial databases.

2) Compare its performance with native analytical tools provided by the RDBMSs.

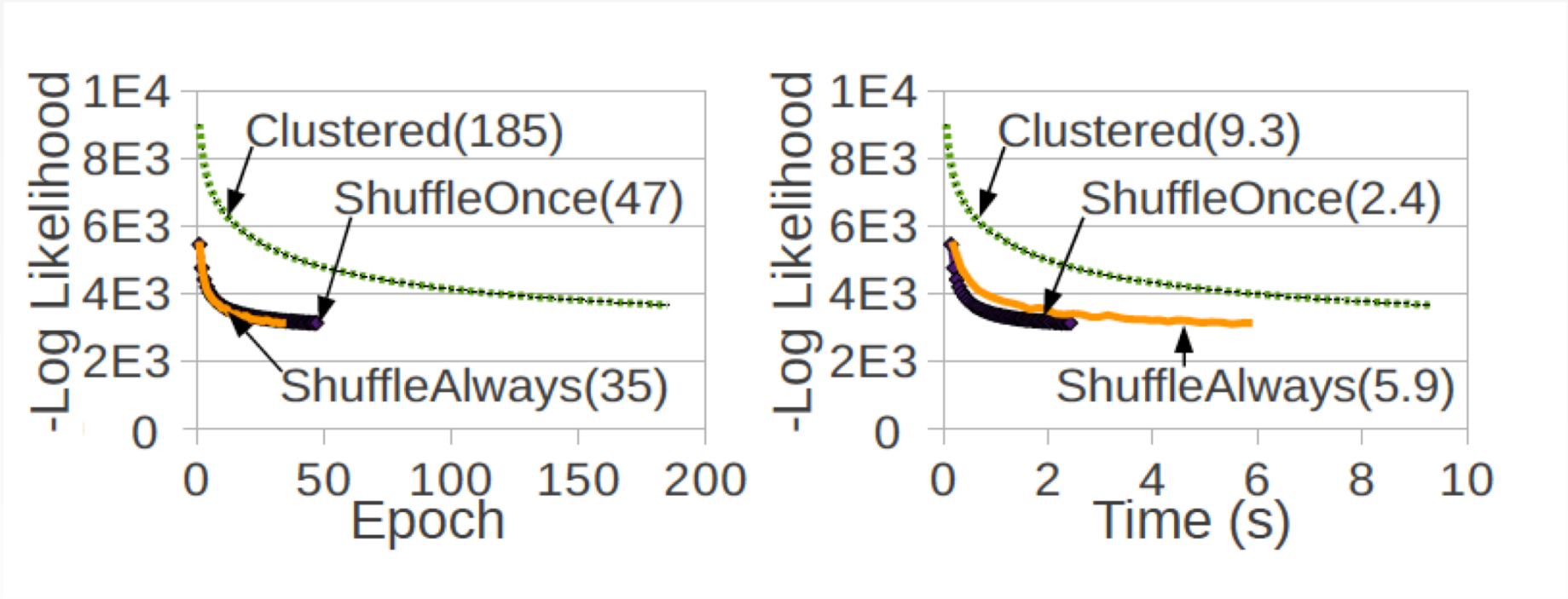
TASKS AND DATASETS

1. **Logistic Regression (LR) -** Forest, DBLife
2. **Support Vector Machine (SVM)**- Forest, DBLife
3. **Low Rank Matrix Factorization (LRM)**– MovieLens
4. **Conditional Random Fields Labeling(CRF)** - CoNLL

BENCHMARKING RESULTS

Dataset	Task	PostgreSQL		DBMS A		DBMS B(8 segments)	
		BISMARCK	MADlib	BISMARCK	Native	BISMARCK	Native
Forest (Dense)	LR	8.0	43.5	40.2	489.0	3.7	17.0
	SVM	7.5	140.2	32.7	66.7	3.3	19.2
DBLife (Sparse)	LR	0.8	N/A	9.8	20.6	2.3	N/A
	SVM	1.2	N/A	11.6	4.8	4.1	N/A
MovieLens	LMF	36.0	29325.7	394.7	N/A	11.9	17431.3

IMPACT OF DATA ORDERING



SCALABILITY TEST

Task	Dataset			Bismarck PostgreSQL	DBMS A (Native)	In-memory Tools
	Name	#Examples	Size			
LR	Classify300M	300M	135GB	✓	✓	X
SVM				✓	✓	X
LMF	Matrix5B	5B	190GB	✓	N/A	X
CRF	DBLP	2.3M	7.2GB	✓	N/A	X

STRENGTHS

1. Incorporating a new task requires only a few lines of code change.
2. Shorter development cycles.
3. Performance optimization is generic.

WEAKNESSES

- Theoretical inference made about the effect of clustering on the convergence rate.
- Only applies to analytical tasks that can be expressed as a convex optimization problem.

REFLECTIONS

REFERENCES