From SQL to Chat:

How to Revolutionize Enterprise Data Analysis with NVIDIA

Baidu Ads Data Team

2024.03.20

Overview

SQL-to-Chat Debut

in Complex Enterprise-level Business Analysis Scenarios

- LUI conversational interaction: just chat with data agent like an old friend!
- Low-latency in Response: let conversational interaction not a dream!

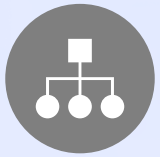
Demo

Motivation

Highly Complex Business Domain, Deep Understanding Required

Complex schema

- 100k+ fields
- Semantic ambiguity
- Complex field parse



Diverse Demand

- Multi-dimension analysis
- For sales, product, R&D users

Frequent updates

- 200 times per month
- 10k+ fields invalid

Chat-like Data Analysis

Little prior knowledge required

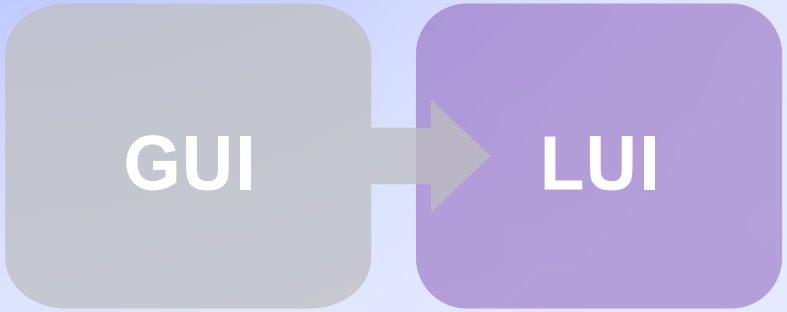
- User demand in natural language
- SQL auto generation

Instant response

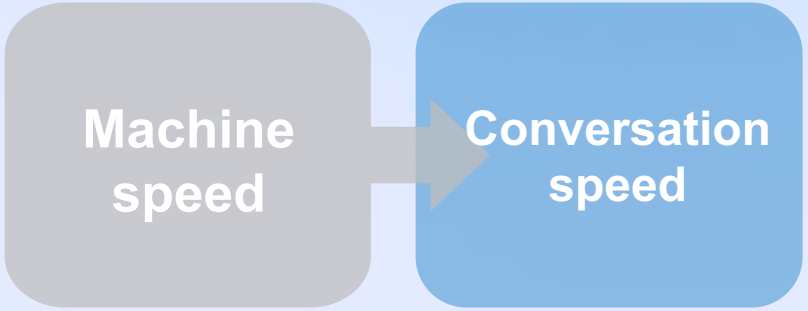
- Chat-like experience
- Speed-up: minutes to seconds

Solution

Little prior knowledge required



Instant response



Text-to-SQL on LLM

- Business Understanding:
  - Schema Alignment, Personalization
- Model Optimization: RAG、SFT、MoE

Hardware-Software Collaborative Acceleration

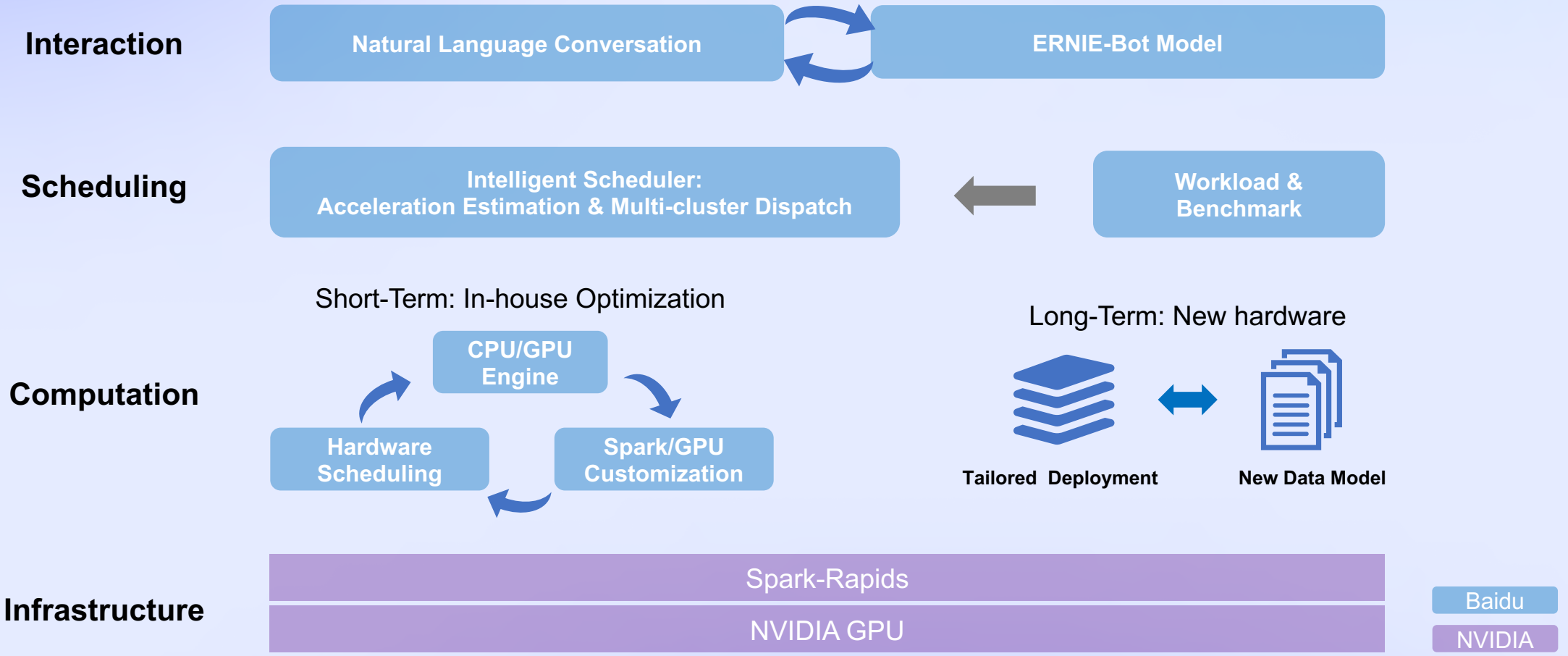
- Software:

  Spark-Rapids application
- Hardware:

  GPU/SSD customization

Solution | Architecture

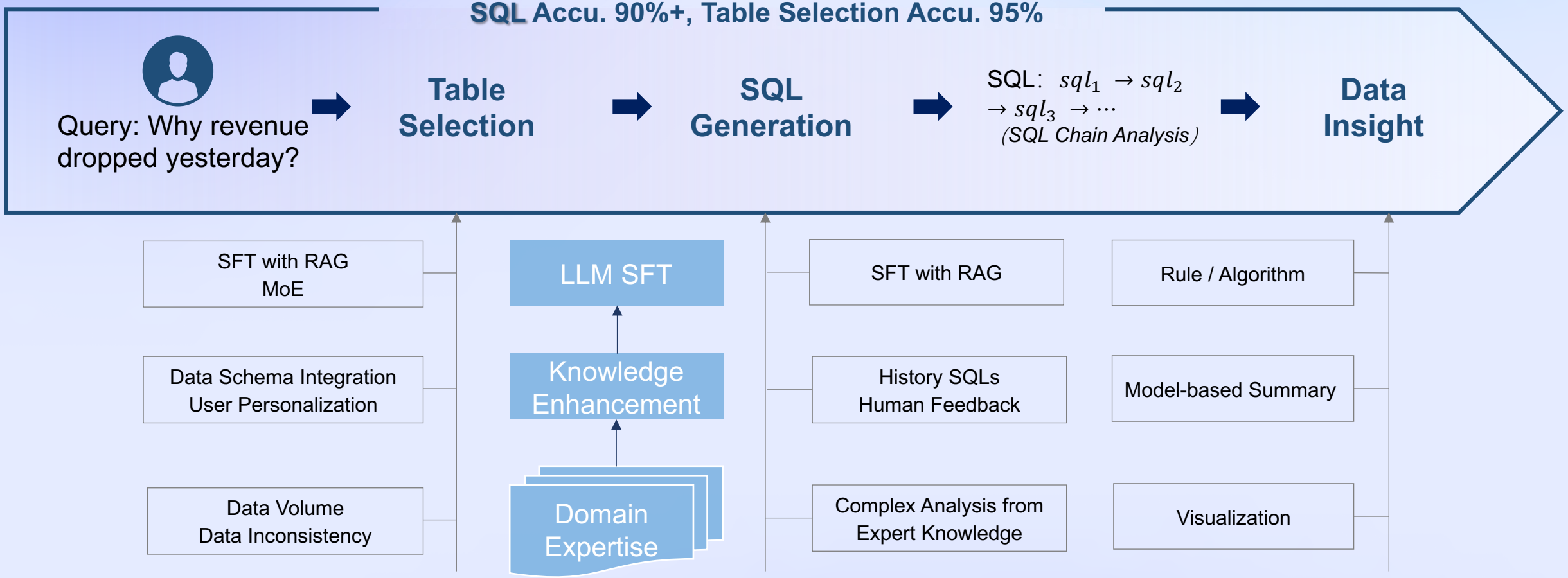
- First Adoption of Conversational Data Analysis
- In-depth Collaboration with NVIDIA on Spark Rapids (up to 5x IMPROV.)



Solution | Revolutionize the Interaction with LLM

Build World-Class Text-to-SQL capability to fulfil complicated business demands based on

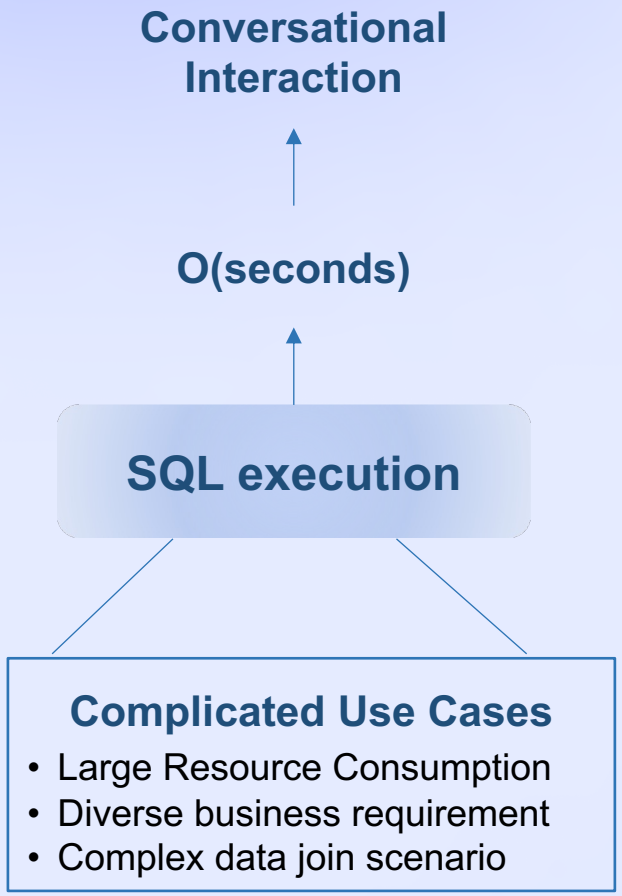
SQL Accu. 90%+, Table Selection Accu. 95%



Computational speed is crucial for conversational interaction!

Solution | Dramatic Acceleration with Spark-rapids

Spark-rapids is the ONLY choice for Conversational Interaction



Hardware-Software Co-Optimization

Software: Spark CPU

- Low ceiling, no 10x increase, cannot break the Von Neumann Architecture

SW/HW Co-op: Spark-rapids + GPU

- ~10x speedup
- Compatible with native Spark
  - Same SQL dialects
  - No Data Integration Costs
  - Suitable for Complex Scenarios

MPP: ClickHouse

- Slow Performance on Joins
- Complicated Schema Integration
- Different SQL dialects

Hardware: AEP + SSD

- Speedup only in I/O, but no 10x e2e
- High integration cost

Solution | Dramatic Acceleration with Spark-rapids

Only Accelerate on Queries that can be accelerated !

Previous Attempts at Baidu

- Spark-rapids: not universally applicable
- Requires thousands of GPUs, low ROI

Failed reasons

- Accelerate everything and all pattern is not feasible
- Not clear about actual distribution of business and underlying data

SQL Identification & Scheduling

Our User Cases

- Diversity in Use Cases
  - 1k+ R&Ds & PMs, 1k tables
  - Execution time: o(Seconds) to o(Hours)
  - Data Tilting: CPU-bound or I/O-Bound
- Fluctuation
  - Seasonality
  - Customer changes

Success experiences

- Only accelerate SQL queries with high ROI and business importance
- The Scheduling layer

Solution | Dramatic Acceleration with Spark-rapids

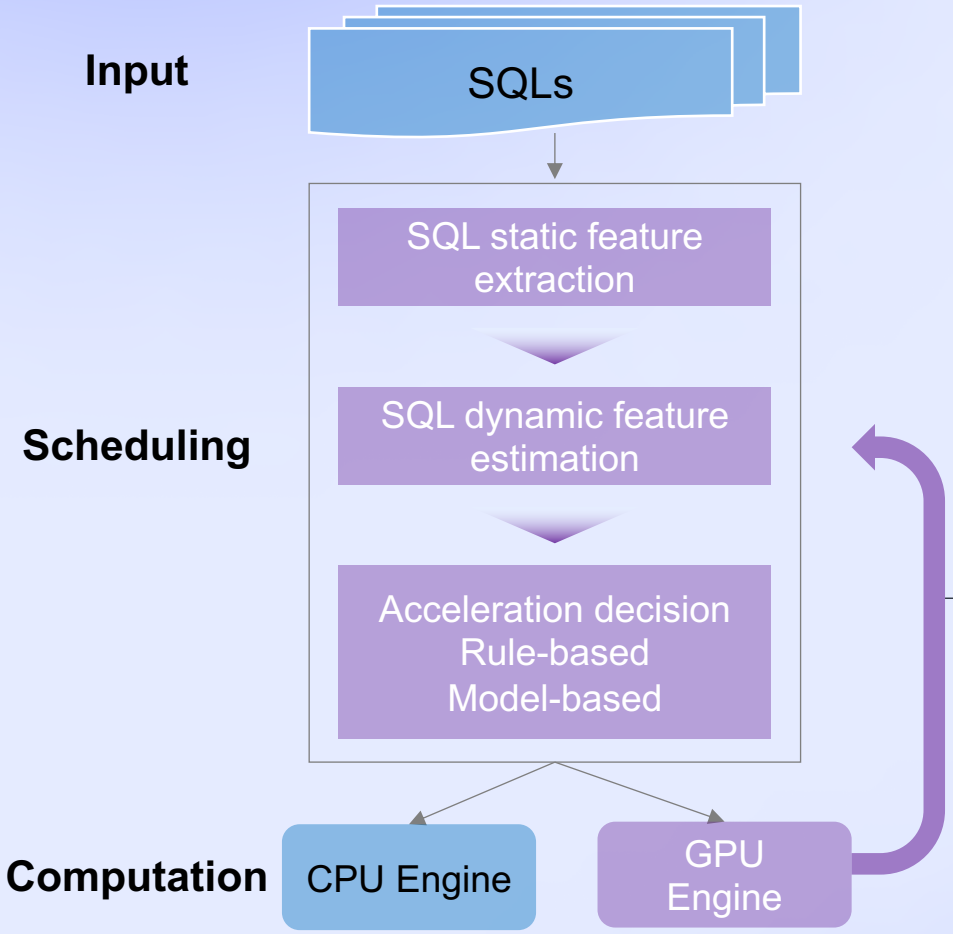
Spark-rapids: Excellent Speed up on diverse scenarios

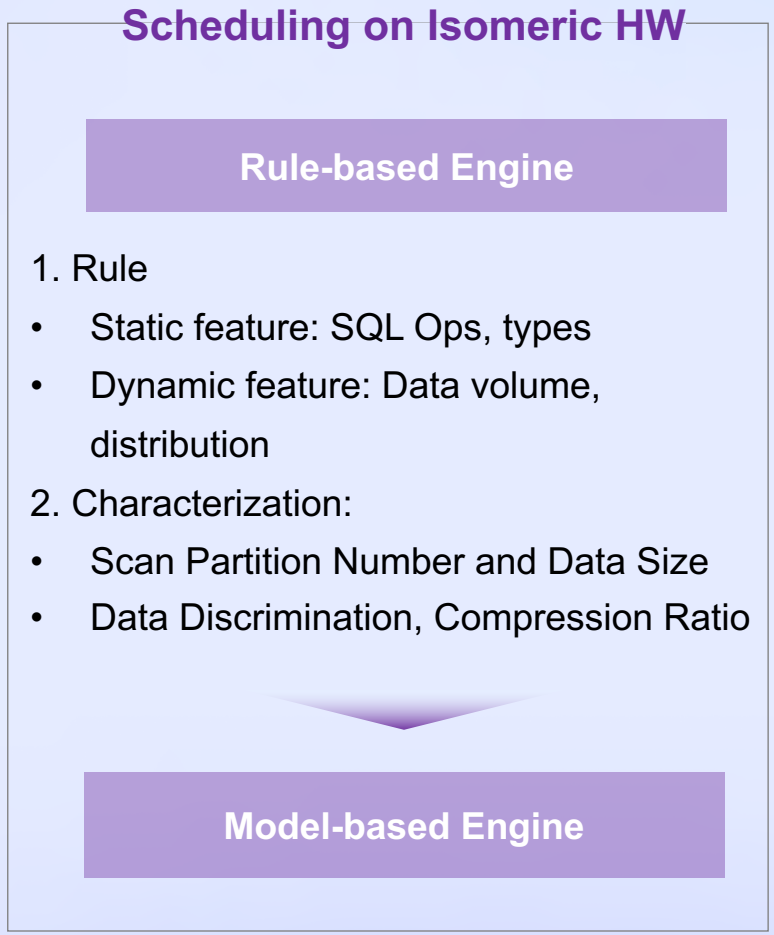
Task Type	Reasons for Acceleration	Speedup Ratio
High Filtering Ratio	Efficient columnar processing with high parallelism on GPU	• Exec-Time: 912s => 239s • Ratio: 3.8
Aggregation on Tilting Data	GPU compute performance far surpasses CPU on tasks with computational bottleneck	• Exec-Time: 705s => 51s • Ratio: 13
Complex Joins	Better algorithm implementation, used GpuShuffledHashJoin to reduce overhead of sorting	• Exec-Time: 237s => 46s • Ratio: 5.1

Solution | Dramatic Acceleration with Spark-rapids

Heterogeneous computing

CPU/GPU Scheduling, new paradigm for Data warehouse





Spark-rapids Sweetspot

- Data Size: 10s to 100s GBs
- Exec Optimization: Join/Agg/Sort/Window Op
- Typical Scenarios:
  - High-filtering Ratio
  - Data Tilting
  - Multiple Tables Join
  - Top N Sorting

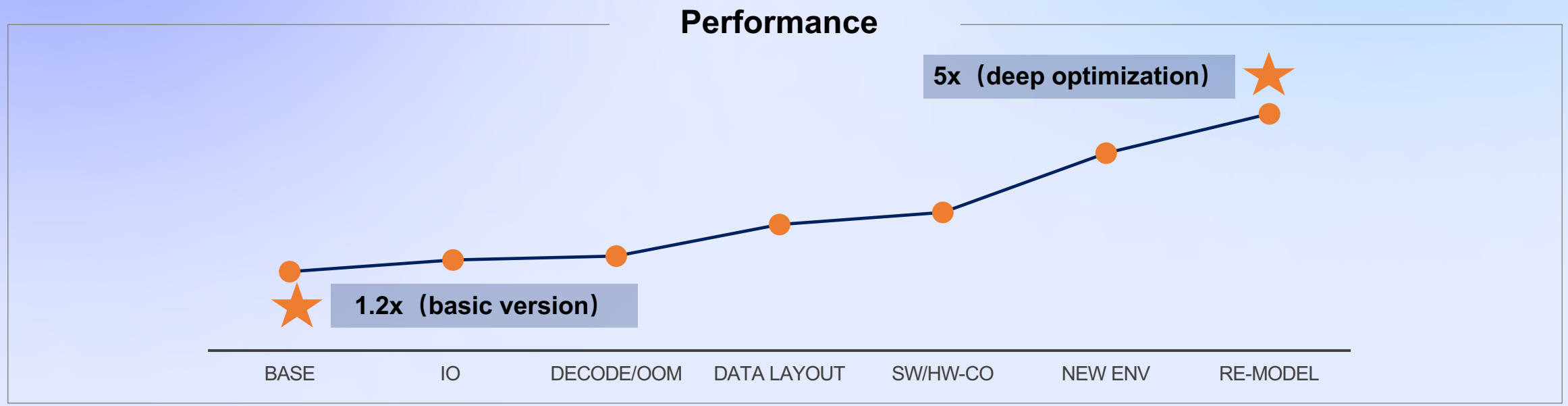
Best Practices in Baidu

Solution | Dramatic Acceleration with Spark-rapids

Deep Collaboration with NVIDIA to enhance Spark Rapids Ecosystem

Scenario / Issue	Solution	Related PR
20% data meets 80% demands	Data hot and cold tiered storage	
Data characters among difference scenarios	Suitable data layout for GPU batches	
OOM during scanning data with high compression ratio	Parquet sub-rowgroup reading	https://github.com/rapidsai/cudf/pull/14360
Performance bottleneck of Parquet: decompression & decode	Optimization on GPU Parquet decode More compression algorithm: lz4_raw Hybrid utilization of CPU and GPU	
Feature & Bug-fix	Support new func: conv, parse_url Enhancing implementation: like, get_json_object	https://github.com/NVIDIA/spark-rapids/pull/8925 https://github.com/NVIDIA/spark-rapids/issues/10254

Conclusion



Business benefits

- First Adoption of Conversational Data Analysis
- Significant efficiency improvement
  - Revenue analysis: day -> seconds
  - Experiment optimization: hours -> minutes

Future Planning

Tailored Deployment on SW/HW

New Data Model for GPU Acceleration

END

Baidu Ads Data Team

2024.03.20