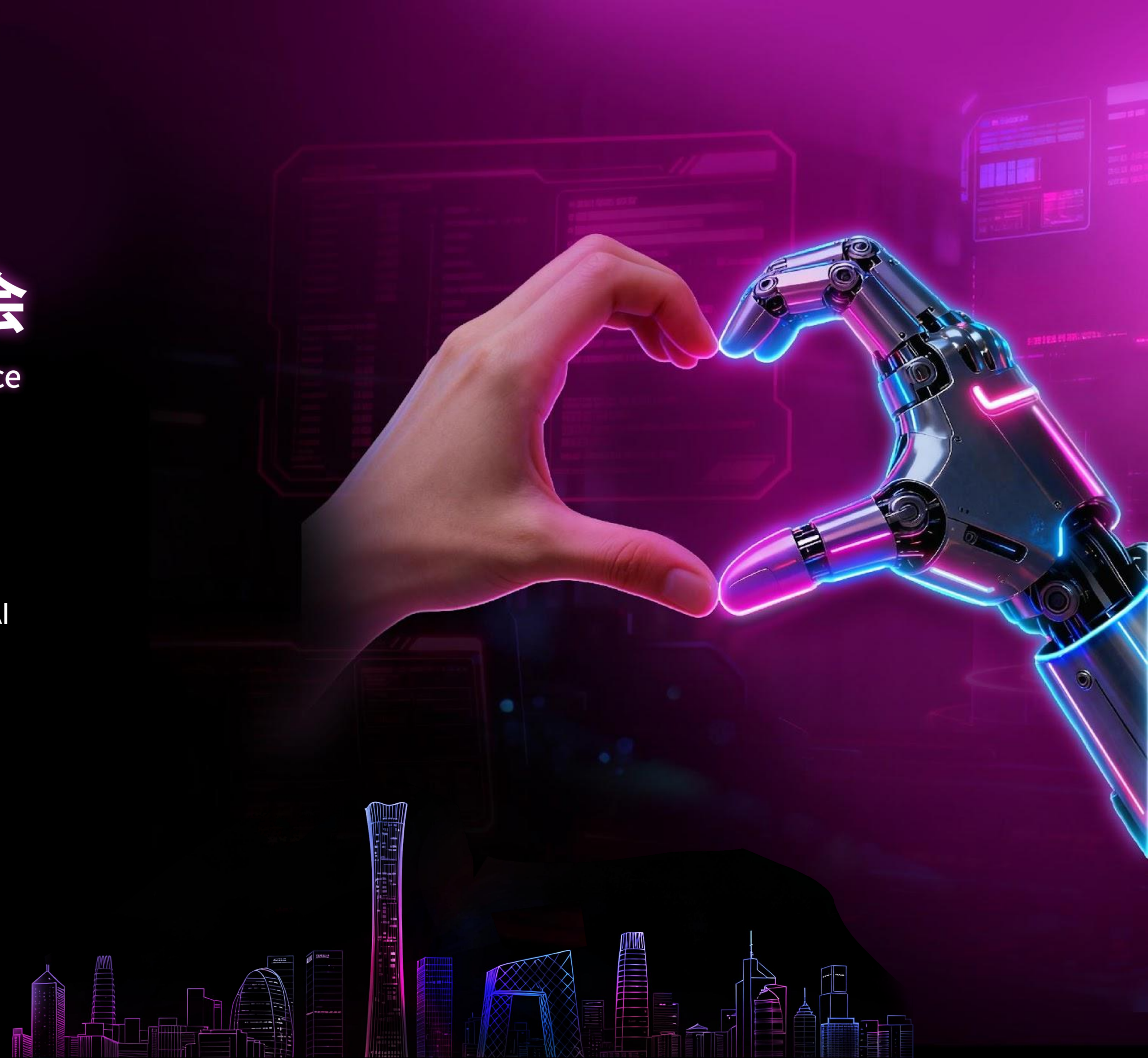
开源社
kaiyuanshe

COSCon'25
第十届中国开源年会

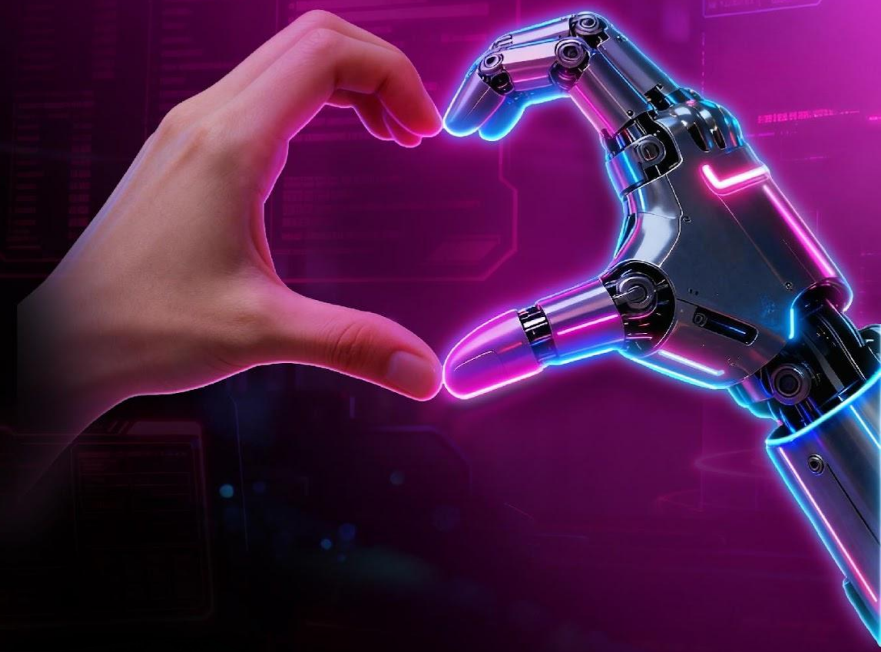
众智开源｜Open Source, Open Intelligence



OSS based Disaggregated Infra for Data & AI

谭涛 Tom Tan

CONTENTS



简介

- 20+ 年业界老兵
- 13年 @ Apple Inc. 数据ML平台总监
- 云从科技副总裁
- Head of SmartNews AI & Data
- 开源 AI & Data Advisory

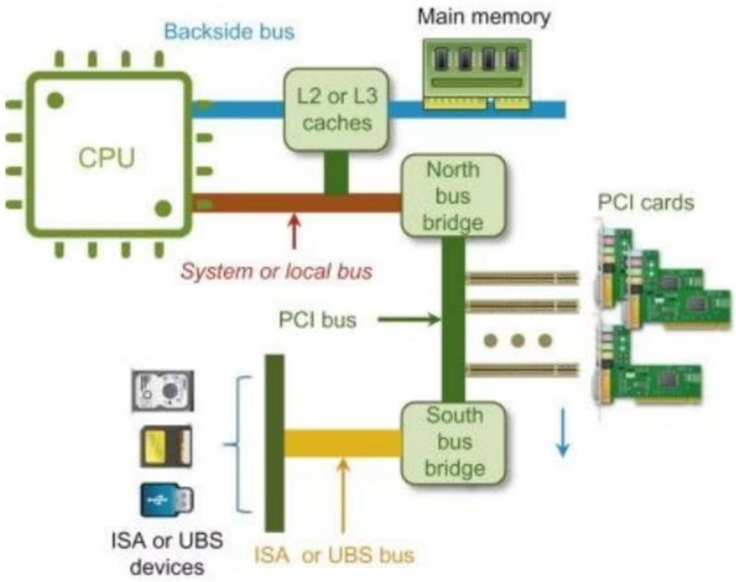
A Once-in-a-Generation Shift in Data & AI Infrastructure

It’s not just a scale problem—it’s a shape change.

Monolithic	Two tier/Multi-tier	Multi Nodes Share-nothing
Hierarchical DB (e.g. IMS) data/code separation	Relational DB	NoSQL Doc, K/V, Wide-column
Vertical Scaling	Vertical Scaling	Horizontal scaling, CAP (Raft, Paxos)
DL/I	SQL, ODBC/JDBC	SQL, MR, API (e.g. dataframe)

?

Computing Architecture: The End of "Free Performance" Gain



Moore's Law Slowdown

Number of transistors used to double every 18 months. Now the same doubling takes 4~5 years.

Cessation of Dennard Scaling

Performance gain can only come with increase in power consumption. CPU clock frequency has stagnated.

Intensified "Memory Wall" Problem

Process core idling - waiting for data from memory. Memory, not compute, becomes the bottleneck in AI, especially at inference.

How Infra responded - System Innovation

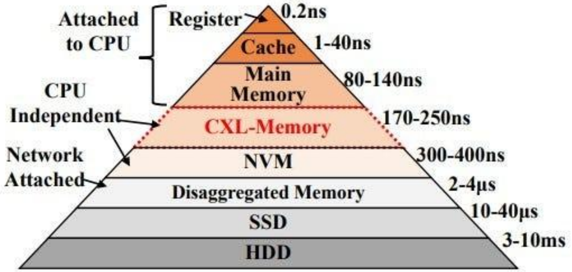
Problem vs Response

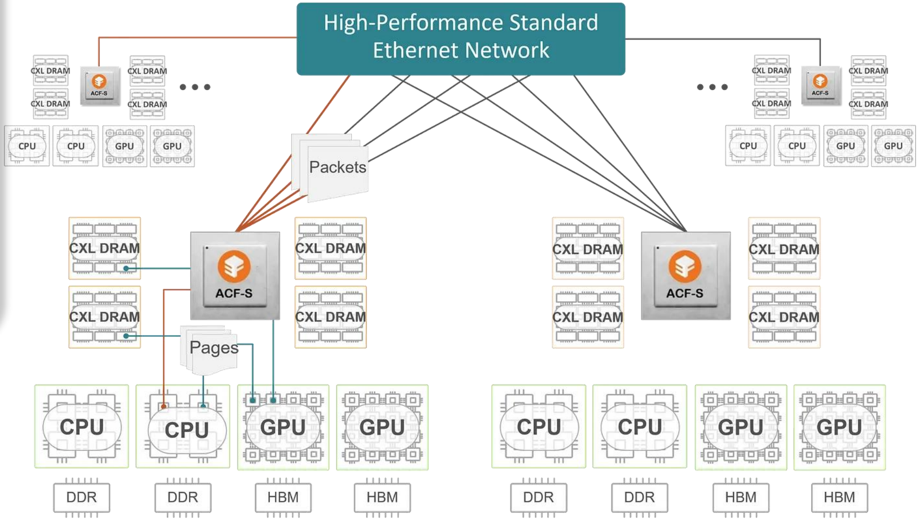
Challenge	Industry Response
Data Volume & Variety	Cloud Object Storage
Memory Wall	Memory pooling: RDMA + CXL (and proprietary tech)
CPU Performance Plateau	GPU/TPU/DPU Specialization
Coupled Resources (in particular the "fixed" CPU/memory ratio)	Disaggregated Physical Infrastructure

While Moore's Law-driven CPU improvements are decelerating (1.6x/decade), disaggregated architectures are projected to deliver 3-5x performance gains by 2030 through Pluggable resource pooling, photonic networking, and workload-specific optimization. This divergence highlights the industry's shift from transistor scaling to system-level innovation for continued progress (source: McKinsey)

How Infra responded - Scale the memory wall

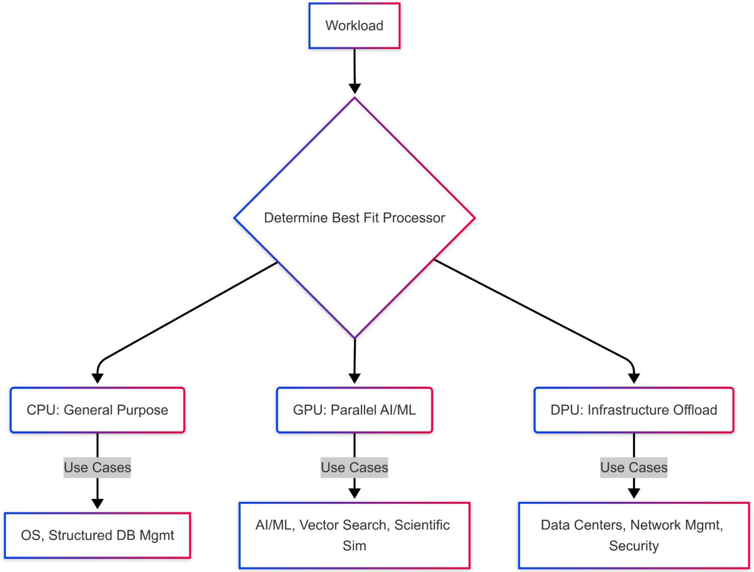
Scale the memory wall	
RDMA	allows one computer to access another's memory directly over a network without involving the CPU, enabling ultra-low-latency data transfer.
CXL	(Compute Express Link) an open standard for high-speed, cache-coherent communication between CPUs and accelerators or memory devices, designed to unify memory and compute resources



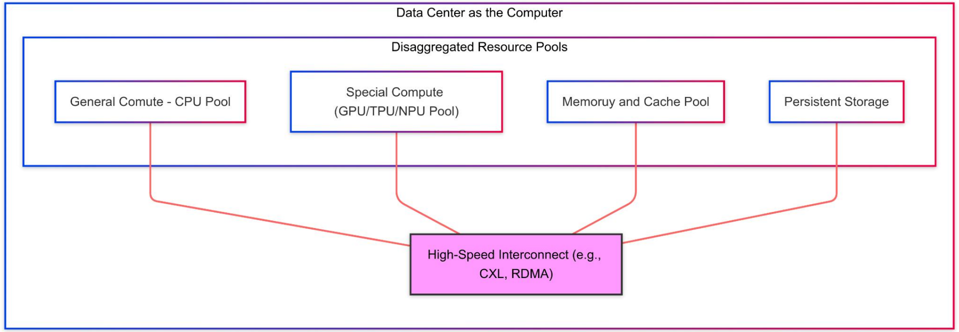


How Infra responded - Specialized processing

The Rise of Accelerators



Disaggregated - Data Center as the computer



Disaggregated Infra Examples

Nvidia Dynamo AI Factory

- Slewed transformer serving workload
  - Prefilling - compute bound
  - Decode - memory bound
- Disaggregated GPU and HBM in cluster to optimize workload mix for mix token throughput

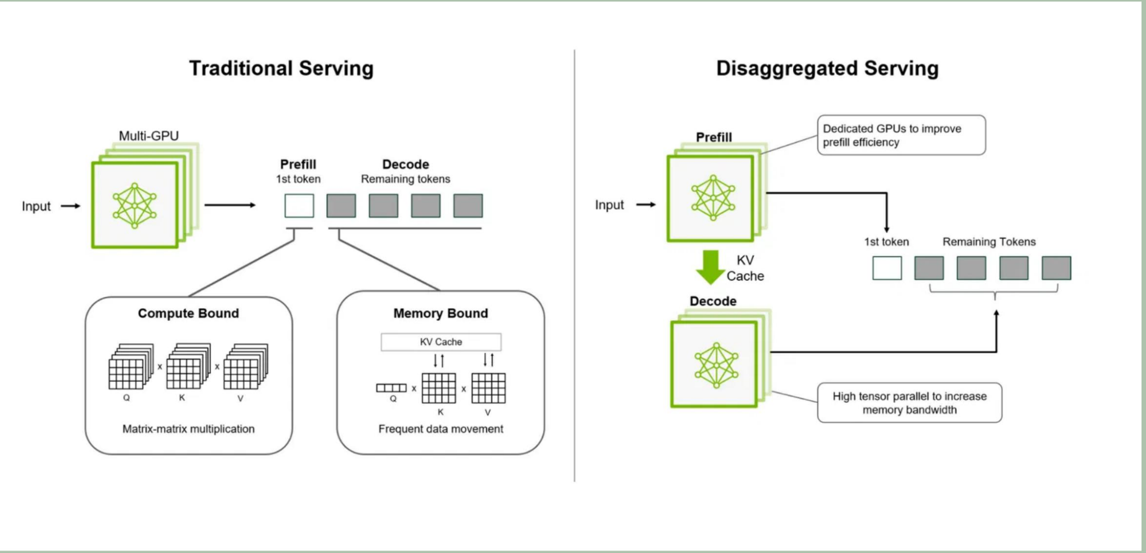


Diagram Source: Nvidia

Disaggregated Infra Examples

DeepSeek Firefly Filesystem

- RDMA connected NVMe drives as shared storage
- Use FoundationDB to serve metadata
- Chain Replication with Apportioned Queries (CRAQ) for consistency
- 6.6TB/s read throughput for a large cluster. 40GB/s as K/V cache, outperforming DRAM cache at inference

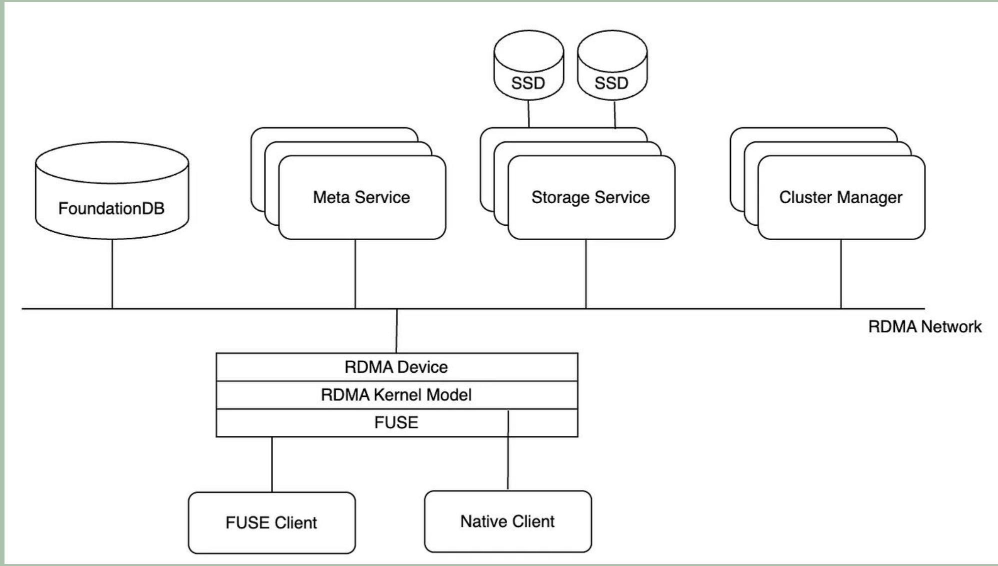
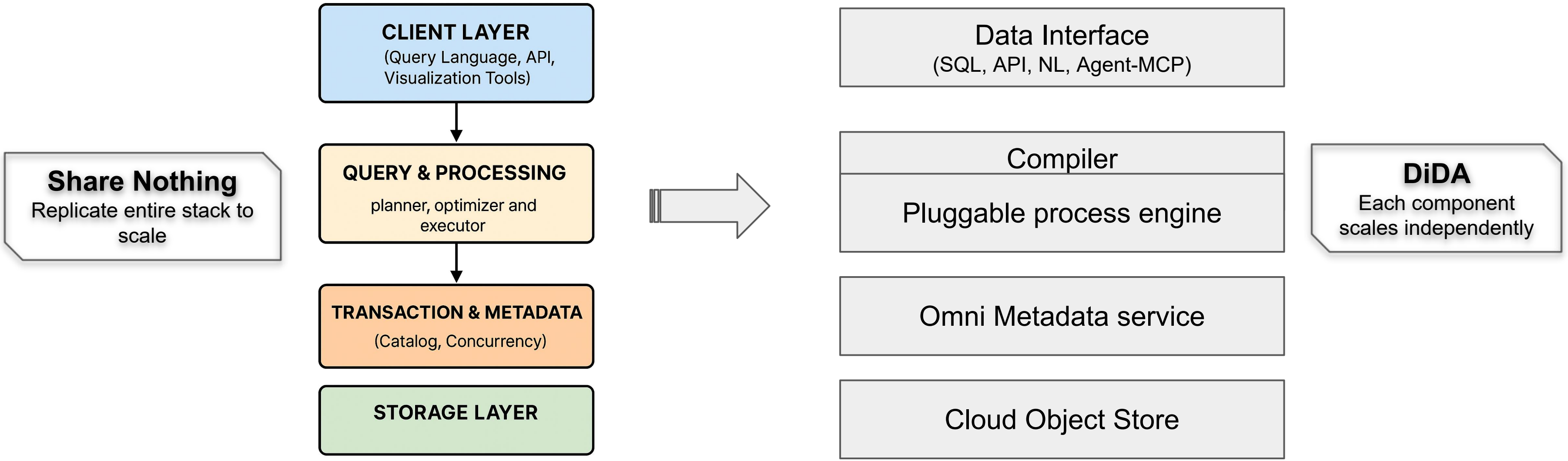


Diagram Source: JuiceFS

Disaggregated Infrastructure for Data and AI

DiDA

Through disaggregate software components, run modern data and AI workloads on a cloud environment to achieve optimized performance, resource utilization and scaling.



Monolithic	Two tier	Multi Nodes Share-nothing	Disaggregated Resource Pooling
Hierarchical DB	Relational DB	NoSQL, NewSQL Doc, K/V, Wide-column	Multimodal Vector, graph
Vertical Scaling	Vertical Scaling	Horizontal scaling (Raft, Paxos)	Heterogeneous,Independent scaling (CPU, GPU, memory, storage)
DL/I	SQL, ODBC/JDBC	SQL, MR, API (e.g. dataframe)	SQL, API NL, Agentic (MCP etc.)

DiDA enables AI workloads

AI Data Volume & Variety

AI driven internet & enterprise growth

IoT, Robotics, Scientific computing

AI Data flywheel

Agentic workflow (agent to agent, agent tool usage, agent memory)

AI native data collection and processing (vector, graph, multimodality)

New Data Interfaces

SQL, API

Natural Language

MCP/A2A

AI workloads - Volume & Variety

80%

Unstructured Data

Of world data by the end of 2025 (IDC)

3X

Growth Rate

Compared to structured data (IDC)

New internet apps

Short videos

Podcasts

AI Agents

600GB

Per Autonomous Vehicle per day

Data generated from just 30 miles of driving = 250M X/Twitter Users

10TB

Per Humonid Per day

Beyond Internet & “classical” enterprises

IoT and wearable (glasses, AI toys)

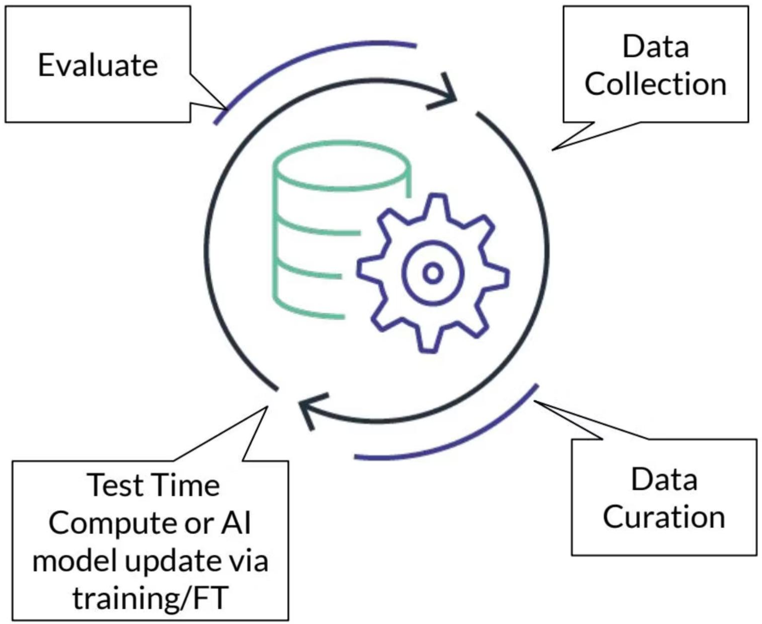
Physical intelligent agents (autonomous driving, Robotics)

AI for Science (e.g. drug discovery)

AI workloads - emerging building blocks

Vector	AI data representation, RAG Supported by all major database providers
Knowledge Graph	Long term memory Domain specific knowledge base
Agentic	Contextual and Temporal state CoT

AI data flywheel - new processing paradigm



Data generated through interaction, used for decision and self improvement

- On-the-fly data type
  - instant schema evolution
- Real-time guardrails
- New collection points

DiDA Principles

1. Scaling

1.1 **Independently scaling each software component and corresponding system resource**
Natively runs on disaggregated physical infrastructure

1.2 **Default on cloud-native storage and open data format**
Unbound capacity, cost efficiency and interoperability

3. Metadata

3.1 **Omni Coverage**
Cover all types of data at all places/engines

3.2 **Physical and Semantic**
Blend physical and semantic meta info of the data.

2. AI Native Processing

2.1 **Unstructured, multimodality data and structured data are all first class citizens**

2.2 **Pluggable engines**
Tailor to different use cases and take advantage of HW accelerators when needed

4. Interfaces

4.1 **Existing proven interfaces**
SQL, API (e.g. python dataframe)

4.2 **New Native AI interfaces**
Natural Language, Agent, MCP

DiDA - cloud object storage

Storage/compute separating
- Independent scaling
- Heterogeneous processing:

Cost effective with multiple tier storage options

Highly durable and available

Unlimited capacity

Open file format supported natively by all major engines
- Parquet, de facto file format for OLAP
- Strong ecosystem around Iceberg, Hudi, Delta Lakehouse

“Cloud-based object storage using open-source formats will be the OLAP DBMS archetype for the next ten years”

Michael Stonebraker
MIT

Andrew Pavlo
CMU

DiDA - Pluggable Process Engines

No one engine fits all. In AI era, diversity of data = diversity of engines
AI workloads vary: Batch, Stream, vector, analytics, fast lookup, transactional

Workload specific optimization

- Local exploration: DuckDB
- Distributed processing: Trino/Spark/Ray
- Vector: Milvus, LanceDB
- GraphDB: Neo4J, Nebula
- Wide-column: Cassandra/ScyllaDB
- NewSQL: TiDB, Spanner, SingleStore
- Time Series: InfluxDB, TDEngine
- TextSearch: Elastic, Solr
- "Pure" query/process engine: DataFusion, Velox, Daft

Shared storage and metadata enable interoperability

Faster innovation, Pluggable upgrade

DiDA - Omni Metadata service

Key differentiators

Classical Metadata:

- Logical Schema. Catalog, Lineage for DB objects
- Physical URI

New responsibilities:

- Logical semantic catalog, physical URI/attributes for all objects in: DB, files, Object store, AI models, prompts ...
- Semantic bridge across engines. Enabler for x-engine consistency
- Discovery, Observability, governance and access control

Metadata service - DiDA’s control plane

01 **Scalability & Performance**
Handling billions of metadata events with low latency and consistency - a must for AI data flywheel

02 **Federation Across Silos**
Integrate metadata across clouds,on-prem, data mesh domain while preserving autonomy

03 **Managing universal, especially AI assets**
Tagging, Versioning and lineage for embeddings, prompts, ML models, raw, intermediate and generated AI data

04 **Semantic Interoperability**
Unified ontologies or knowledge graphs to align terminology across domains

05 **Automation with trust**
AI-driven metadata capture with human oversight

06 **Observability, Governance and Compliance**
Fine-grained access control. Integration with “classic” IT and new AI observability

07 **Transaction ability across different engines**
Data consistency management

Interface - SQL is great, but...

Lingua Franca since 1970:

- Declarative
- Readability
- ACID properties
- Cornerstone of the data domain

Schema Dependency
Evolving schemas; on-the-fly data types

Weak Expressiveness
Complex transformations, conditional logics, UDF is not so elegant

Limited support for non-tabular data
Vector, Graph, Image, audio and multimodal content

Scalability & Performance Issues
Complex joins etc. can degrade performance, engine specific tuning (partition hints, caching strategies)

Non-intuitive for non-technical users; Lack of Contextual Awareness
High learning curve for business analysts, domain experts;
AI engineers, data scientists prefer Python

Interface - Comparing options

Interface	Pros	Cons
SQL	Declarative, standard for structured data, optimizer-backed	Rigid schemas, limited expressiveness, table-only, not AI native
Dataframe/API	Highly expressive, flexible code driven logic, well fit for AI/ML pipelines, non-tabular data	Require dev skills, manual optimization, less declarative
Natural Language	Accessible, intuitive, fast for non-tech users	Ambiguous, restricted to simple use cases, unreliable for complex, high precision tasks
Agentic/MCP	Standardized agent-tool interface, dynamic discovery, unified context across systems; support autonomy and complex workflow	New standard, immature, still evolving

Interface - what the future holds

- **Natural Language Interfaces** will be **primary for self-service analytics**—making data access intuitive and inclusive.
    a. Gartner forecasts: *"By 2026, natural language processing will become the dominant way users interact with enterprise data"*, enabling **10× better data access** across organizations .

    b. Airbyte concurs: adoption of GenAI and semantic layers is driving conversational data modeling driven transformation

Taken with a huge grain of salt

- **Agentic + MCP interfaces** will underpin **automated, tool-driven AI workflows**, especially in agent-based systems and AI-native/AI-assisted stacks.

- **SQL and API-level interfaces** will remain essential for complex or stringent engineering/business workloads and developer-centric tasks.

Compiler

Data Interfaces

SQL API NL MCP

“Retrieve top X active users in the last Y days and explain what they may have in common?”

Interface Compiler

Pluggable process engines

Workload specific optimization

- Local exploration: DuckDB
- Distributed processing: Trino/Spark/Ray
- Vector: Milvus, LanceDB
- GraphDB: Neo4J, Nebula
- Wide-column: Cassandra/ScyllaDB
- NewSQL: TiDB, Spanner, SingleStore
- Time Series: InfluxDB, TDEngine
- TextSearch: Elastic

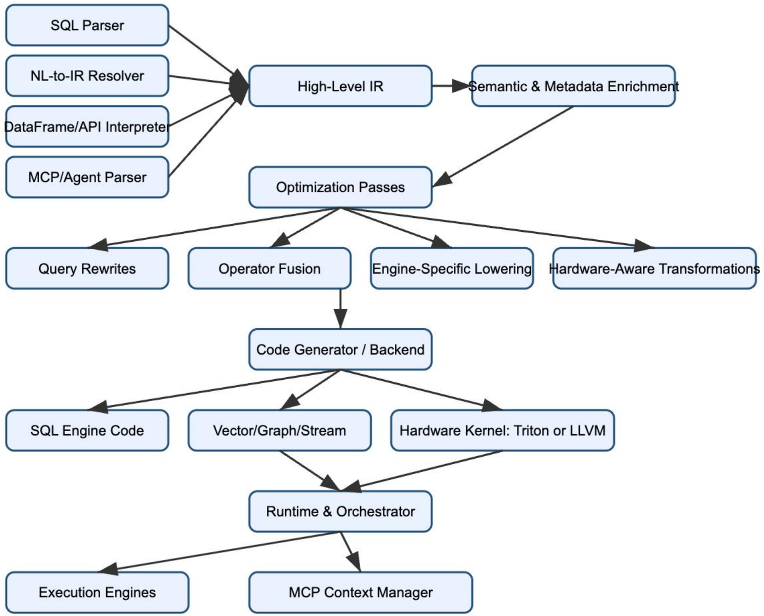
Compiler - How it works

Approach to abstract HW from programming

- General computer language - GCC, LLVM
- AI compilers: Triton, Apache TVM, TileLang
  - Support multiple GPU architectures
  - Near CUDA native performance

We need data interface compiler to abstract different data engines from data interfaces

A data interface compiler may look something like the picture to the right



Compiler - Thought experience

1. **User**: “Retrieve top X active users in the last Y days and find their similarities.”
2. **Frontend**: Parses and maps request to IR graph.
3. **Compiler**:
    - Partitions the graph: SQL engine for structured filtering, vector engine for embedding similarity.
    - Applies optimizations: predicate pushdown, vector quantization, tiling.
    - Plans physical execution: GPU-tiled kernels + distributed query execution.
4. **Execution**:
    - Metadata layer provides schema + location.
    - Engines execute optimized sub-plans.
    - Results merged and returned seamlessly.

DiDA Stack Maturity

Mature	Expanding	Expanding	Emerging	Lacking
Cloud Object Store	Pluggable Engines	Data Interfaces	Omni Metadata	Compiler
Battle proven	Ranging from established players to brand new entries in various spaces DataFusion and Velox are more of “pure” execution engines, though limited to OLAP structured data	SQL and API are well established NL is on the rise MCP-Agentic is hot	Federated, interoperable Metadata service is taking shape	No major players Multimodality: Cortex AISQL Backend independent SQL: SQLGlot, SDF*, Substrait Backend independent API: ibis (python)

Platform builders

Get Ready for DiDA	
Storage	**Take advantage of Cloud (or on-prem) object store as much as possible**
Engine	**Adopt engine for your workload, anticipate new workload types will surface**
Metadata	**Invest in metadata service, pay special attention to interoperability to avoid semantic silos**
Interface	**Judiciously deploy new interface mechanisms for real use cases**

Call for OSS developers

- metadata interoperability
- Develop interface compiler

开源社
kaiyuanshe

COSCon'25
第十届中国开源年会

众智开源 | Open Source, Open Intelligence

Thanks