MaxCompute半结构化数据思考与创新

周宇睿-阿里云-高级技术专家

DataFunSummit # 2023

阿里云 DataFun.

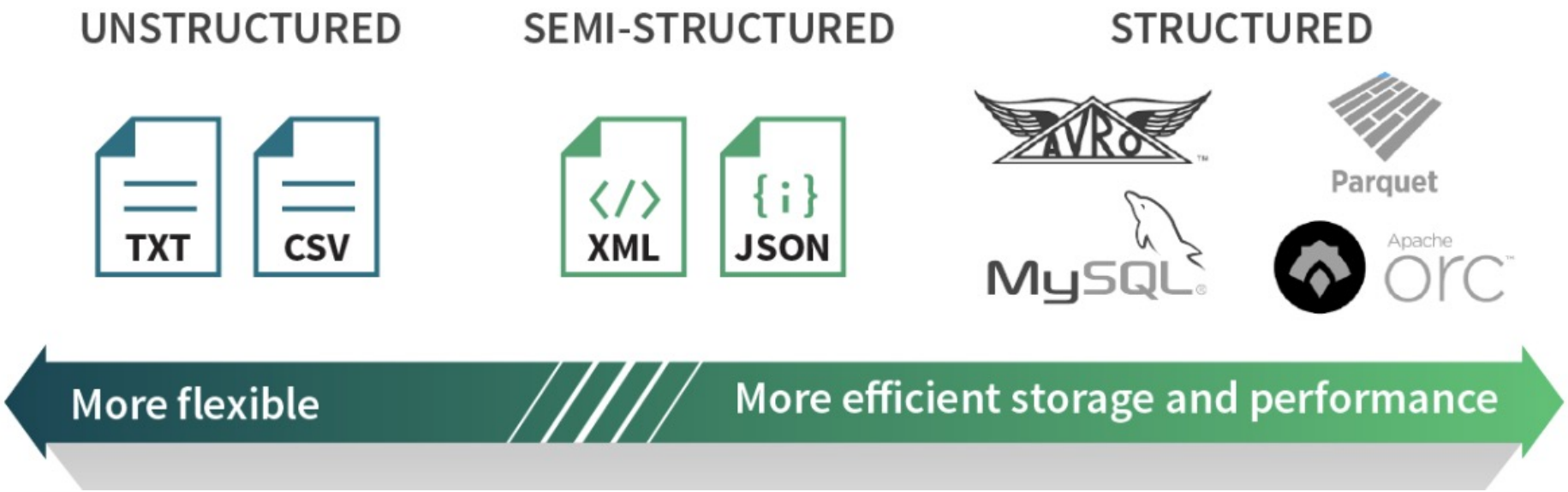
目录
CONTENT

01
半结构化数据场景简析

DataFunSummit # 2023

阿里云 DataFun.

什么是半结构化数据



- 数据包含一定的 schema，但是 schema 灵活，没有强约束
- 数据的 schema 是自描述的
- 一般都有多层级、嵌套结构
- 典型的半结构化数据
  - JSON：几乎就是半结构化数据的代名词，最流行的一种数据交换格式
  - XML：用途很接近于 JSON

为什么需要半结构化数据

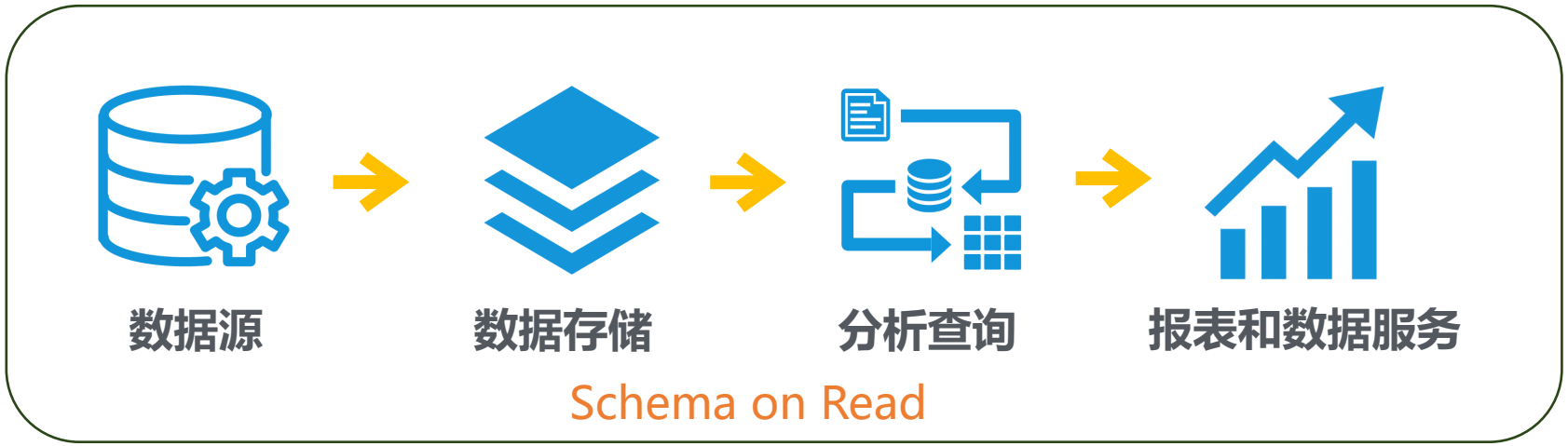
- 结构灵活通用，适用场景广泛
- 终端原始数据，信息丰富
- 生态完备，多平台协同

02
传统半结构化数据解决方案

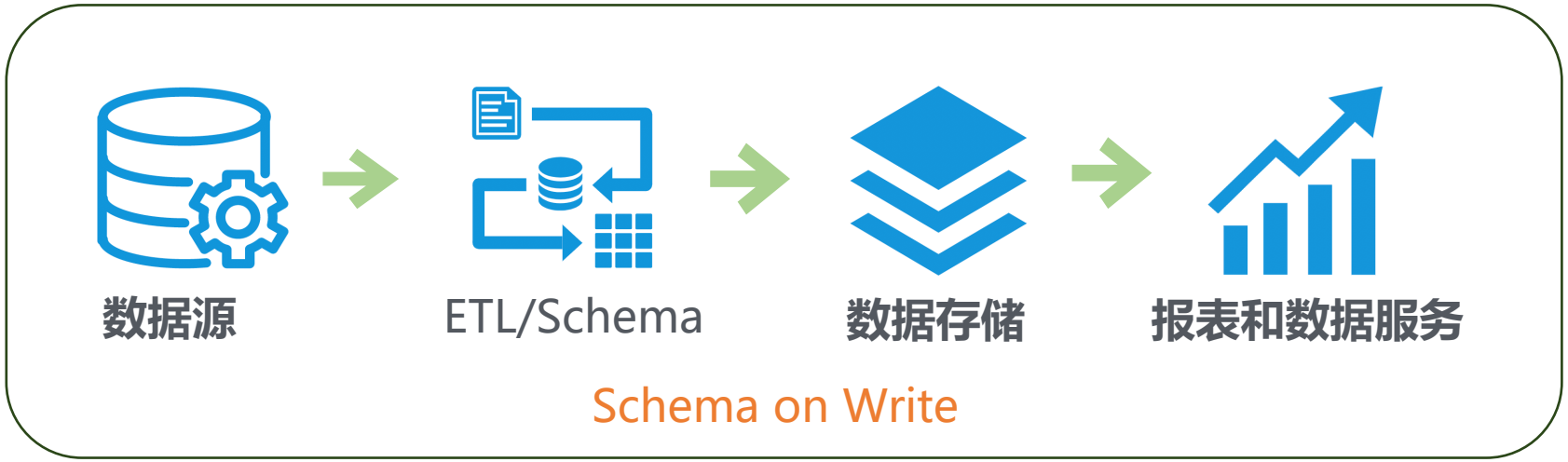
DataFunSummit # 2023

阿里云 DataFun.

Schema-on-Read vs Schema-on-Write



Schema on Read
- 写入不校验，读取时动态解析
- 场景灵活



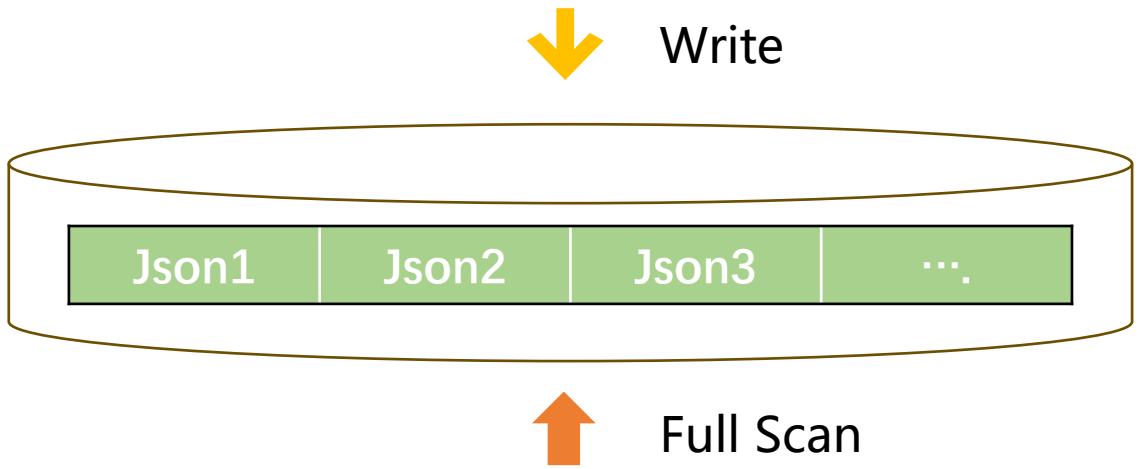
Schema on Write
- 写入时确定数据结构
- 性能更好

Schema on Read

	Data Source
Json1	{"id":1001, "name": "Adam", "gender": "Male", "age": 16}
Json2	{'id': 1002, "name": "Bob", "gender": "Male", "age": 41}
Json3	{"id":1003, "name": "Claire", "gender": "Female", "age": 21}
…	…

执行慢！

- 数据存储开销大
- 查询都需要FullScan，数据解析效率低，访问性能差



解压

Parser/Extractor

Age
16
41
21
..

```
SELECT count(*) from json_table where cast(json_data[“Age”] as int) > 18;
```

Schema on Write

	Data Source
Json1	{"id":1001, "name": "Adam", "gender": "Male", "age": 16}
Json2	{'id': 1002, "name": "Bob", "gender": "Male", "age": 41}
Json3	{"id":1003, "name": "Claire", "gender": "Female", "age": 21}
…	…

ETL/Parser

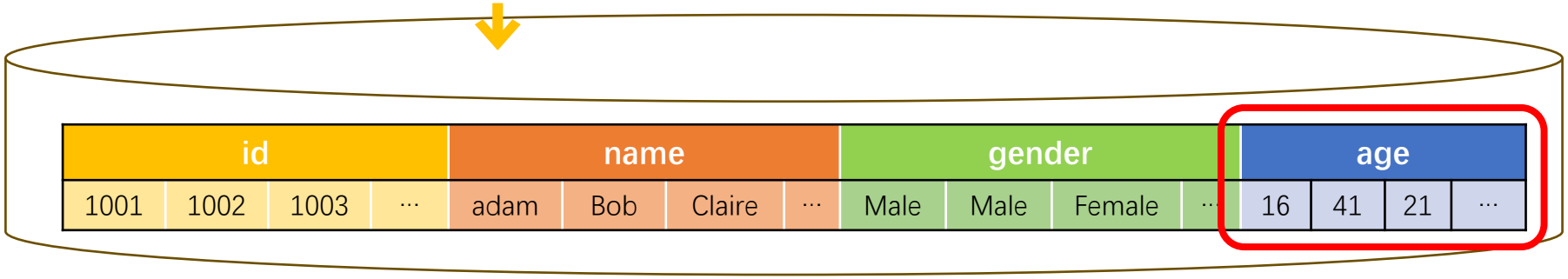
Schema<Id:int, name:string, gender:string, age:int>

优点：

- 数据分析查询效率提升
- 存储效率高

SELECT count(*) from json_table where age > 18;

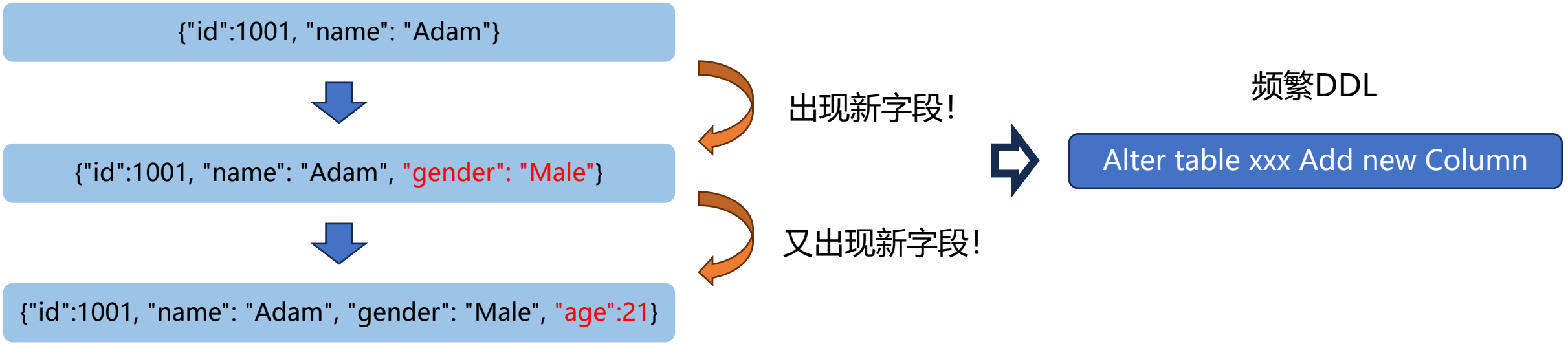
Column Pruning



解压

Age
16
41
21
..

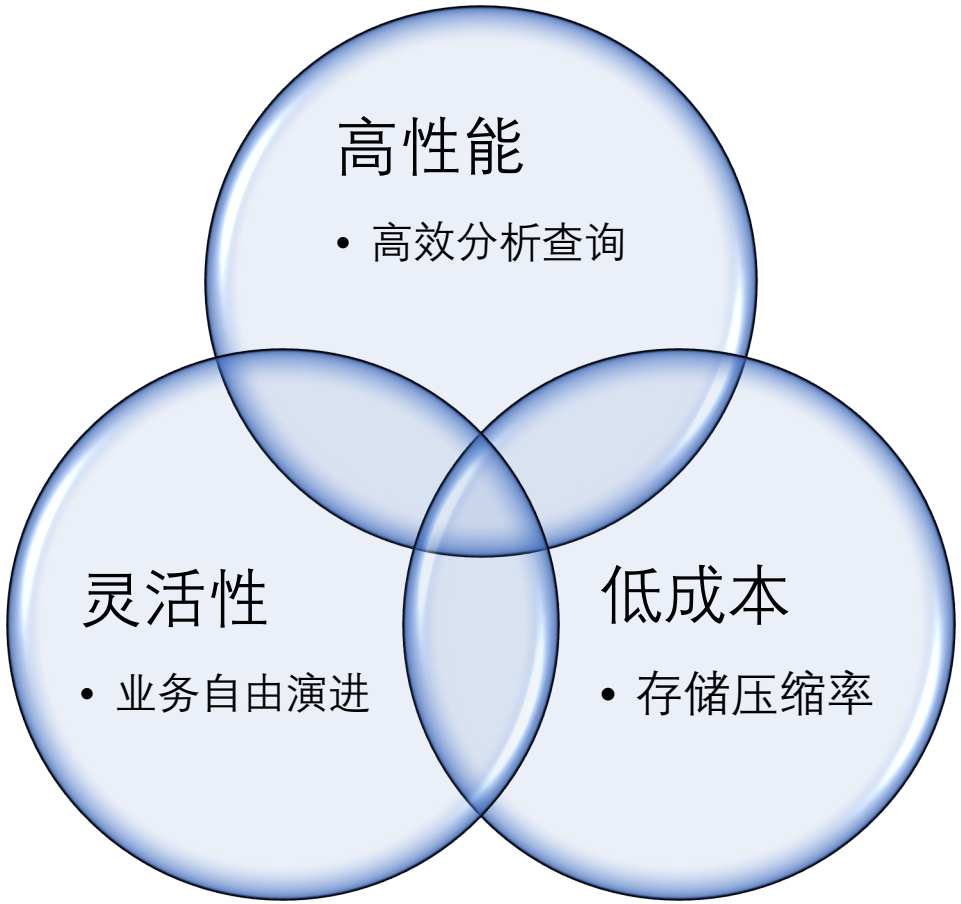
Schema on Write 带来的挑战



- 运维效率低
  - 无法事先确定字段
- 协同成本高
  - 上游业务快速迭代，下游维护疲于奔命

既要? 又要? 还要?

数仓半结构化场景核心需求



Schema On Write

➢ 数据结构动态提取与转换

+

Schema On Read

➢ 自适应数据访问

03
MaxCompute半结构化数据解决方案

DataFunSummit # 2023

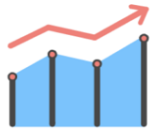
阿里云 DataFun.

MaxCompute概述

交互式分析

挖掘预测


海量日志分析处理


分析报表


用户画像


快速云上建仓

MaxCompute

多租户隔离的项目空间 和按需独占购买/按需共享使用付费的资源组

批处理
(SQL/MR/Spark)

交互式分析
(MC-Hologres)

科学计算/机器学习
(Mars/PAI)

多租户安全管理/
资源监控

New SQL

Java

Python

MaxCompute 并行计算引擎

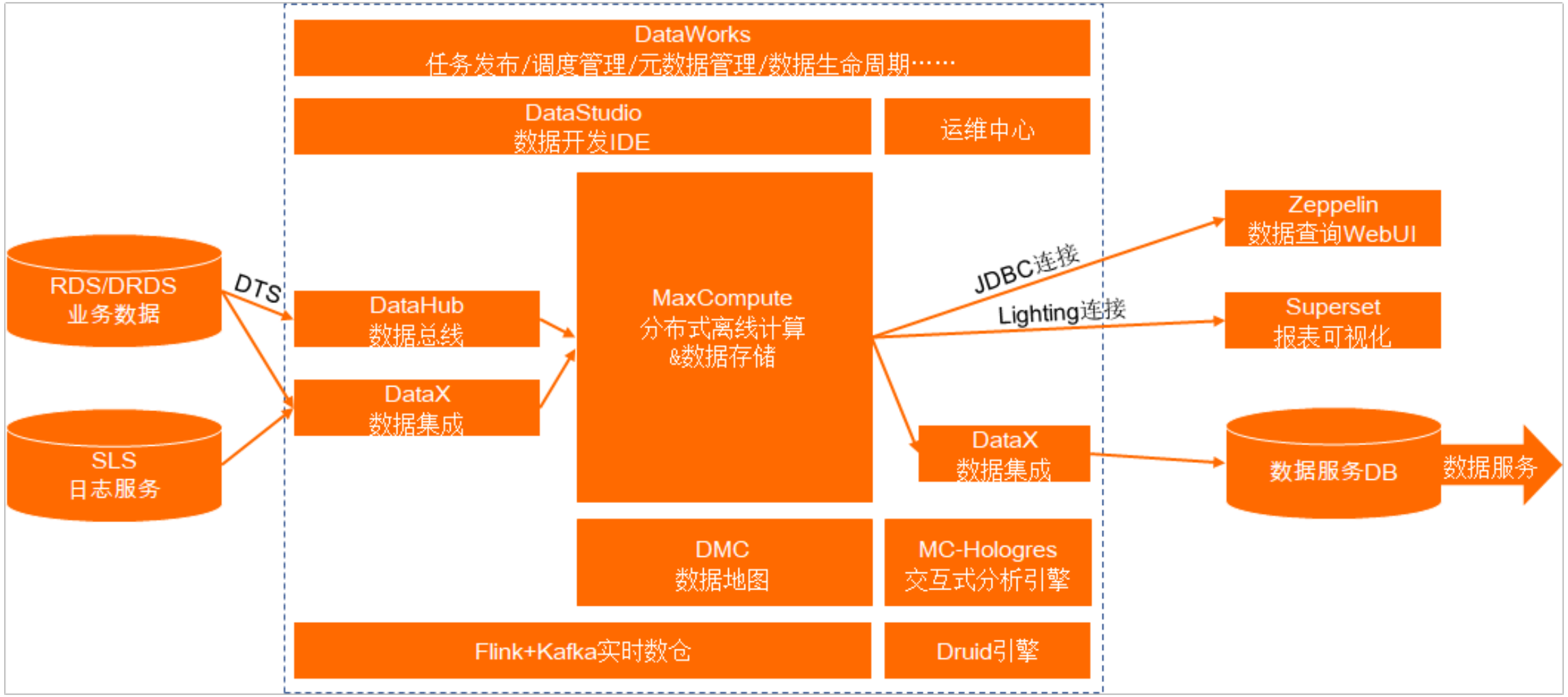
弹性资源管理及任务调度 fuxi

弹性数据存储 pangu

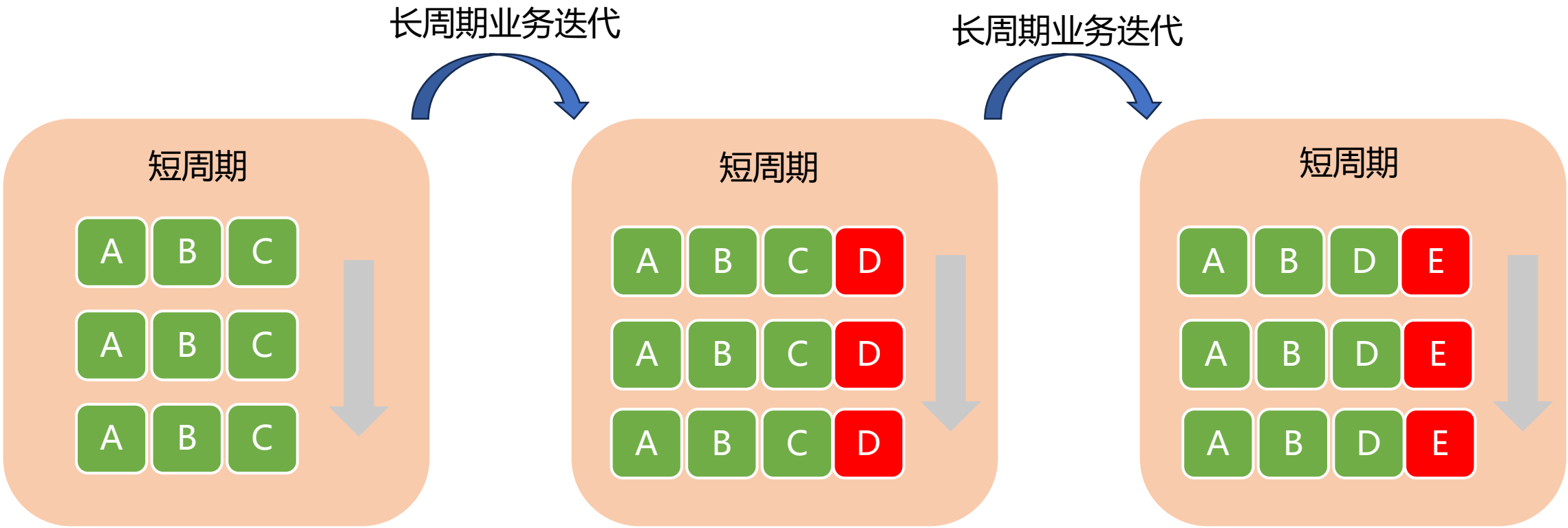
全局元数据

云上 / 用户专有环境 大规模弹性资源池

MaxCompute半结构化数据场景



半结构化场景观察与思考



- 短周期，
  - 数据结构保持稳定，字段类型和数量保持不变
- 长周期
  - 数据结构缓慢演进，字段类似和数量发生变化

- 短周期
  - 公共Schema自动提取
  - 半结构化数据列存化
- 长周期
  - 动态类型自适应演进

Json数据列存化

	Json Data
Json1	{"id":1001, "name": "Adam", "gender": "Male", "age": 16}
Json2	{'id': 1002, "name": "Bob", "gender": "Male", "age": 41}
Json3	{"id":1003, "name": "Claire", "gender": "Female", "age": 21}
…	…

Full Scan

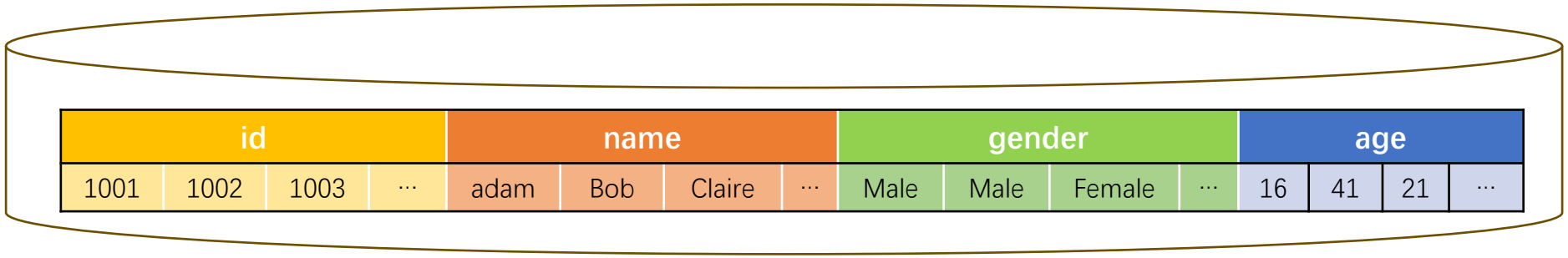
Schema Extractor

Schema提取

Schema<Id:int, name:string, gender:string, age:int>

Json Converter

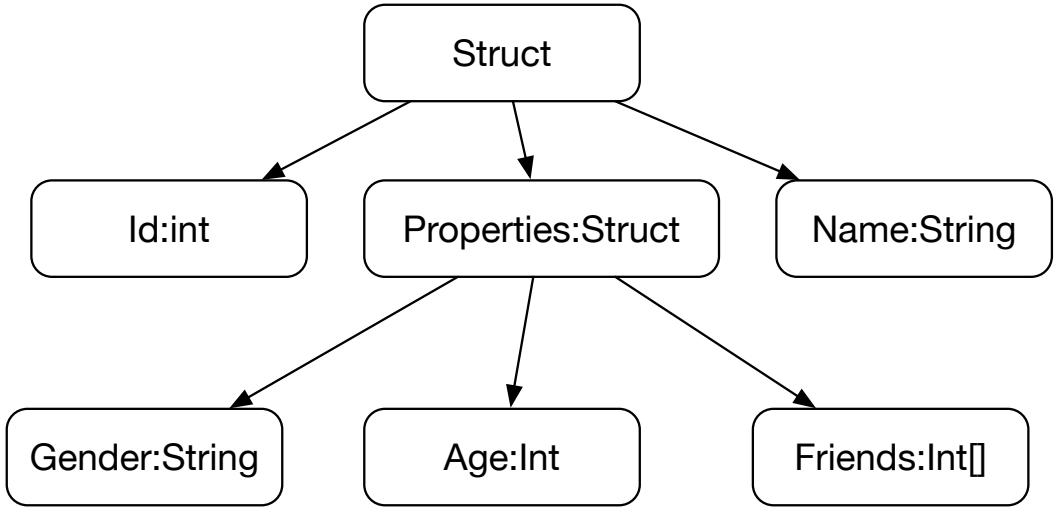
数据列存化



嵌套结构处理


```
{
  “Id”:1001,
  “Name”:”Adam”,
  “Properties ”: {
     “Gender”: “male”,
     “Age”: 13,
     “Friends”: [1002, 1003, 1004]
  }
}
```

Schema 提取



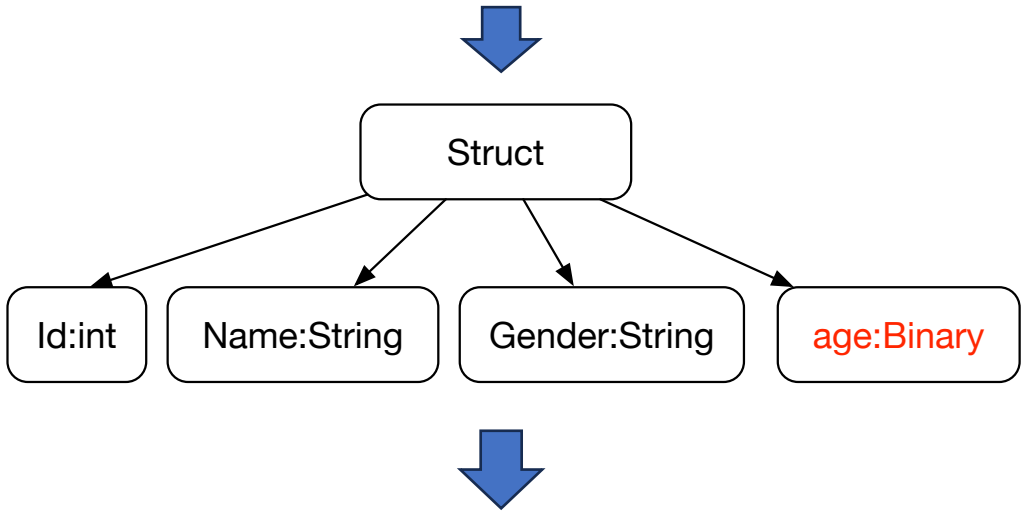
AliORC列存
- 基于Apache ORC
- 天然支持嵌套结构
- 完整保存节点嵌套关系
- JsonPath与类型节点映射

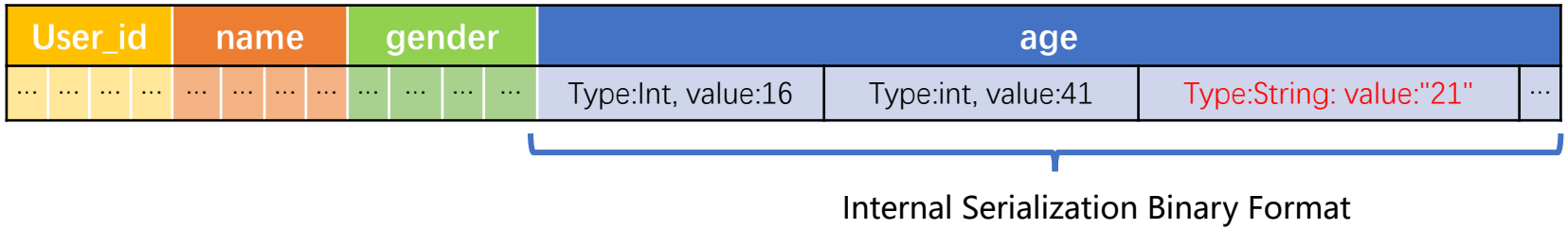
Root: Struct				
Id: int	Properties: struct			Name:String
	Gender:String	Age:Int	Frends:int[]	
1001	male	13	[1002, 1003,1004]	Adam
…	…	…	…	…
…	…	…	…	…

脏数据处理

- 灵活性与易用性，脏数据难以避免
- 平台无法区分脏数据，据需要完整保存信息
- 利用二进制格式，额外保存类型信息
- 脏数据类型与普通类型隔离
  - 最大限度保障存储查询效率
  - 最大限度利用列存效率

	Data Source
Json1	{"id":1001, "name": "Adam", "gender": "Male", "age": 16}
Json2	{'id': 1002, "name": "Bob", "gender": "Male", "age": 41}
Json3	{"id":1003, "name": "Claire", "gender": "Female", "age": "21"}
…	…

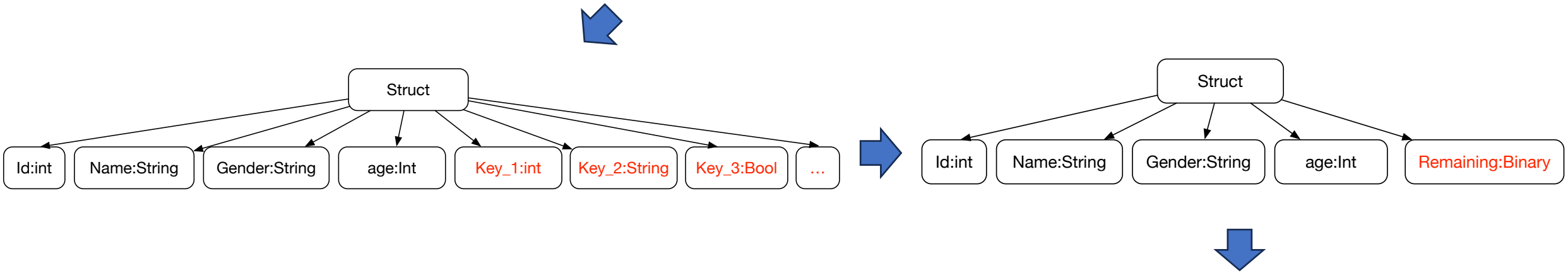




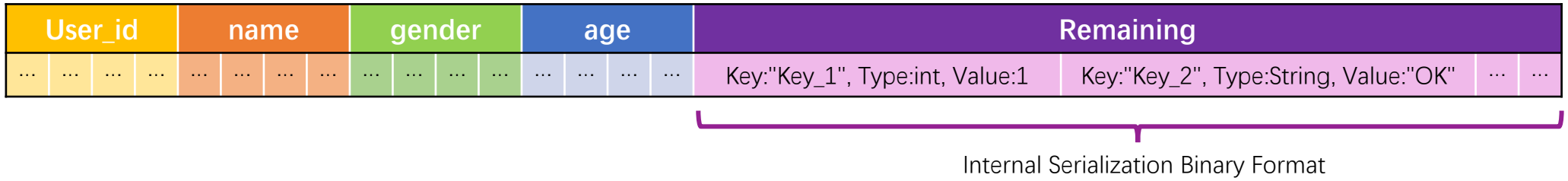
稀疏类型处理

- 稀疏类型导致列数膨胀
- Schema 提取过程频率统计
  - 低频类型统一归入Remaining 序列化存储

	Data Source
Json1	{“id”:1001, “name”: “Adam”, “gender”: “Male”, “age”: 16, "Key_1": 1}
Json2	{'id": 1002, "name": "Bob", "gender": "Male", "age": 41, "Key_2":"OK"}
Json3	{"id":1003, "name": "Claire", "gender": "Female", "age": 21, "Key_3": false}
…	…

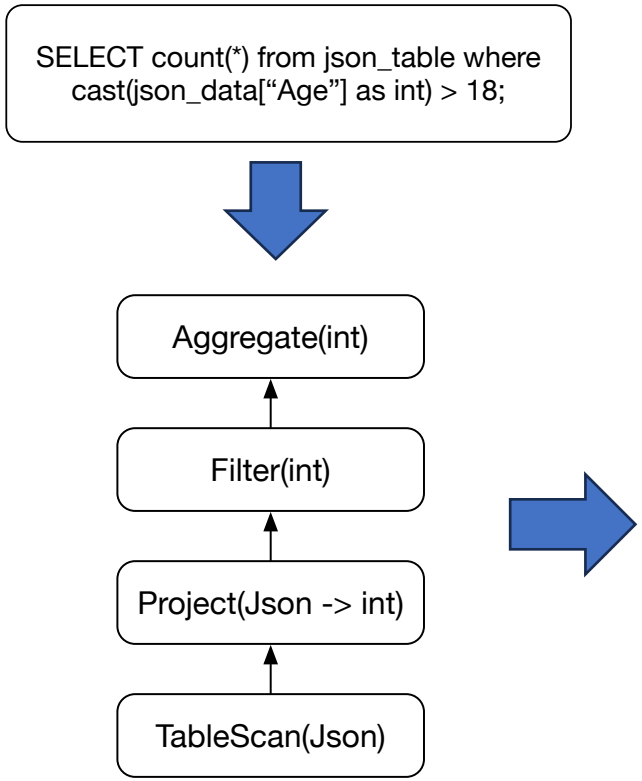


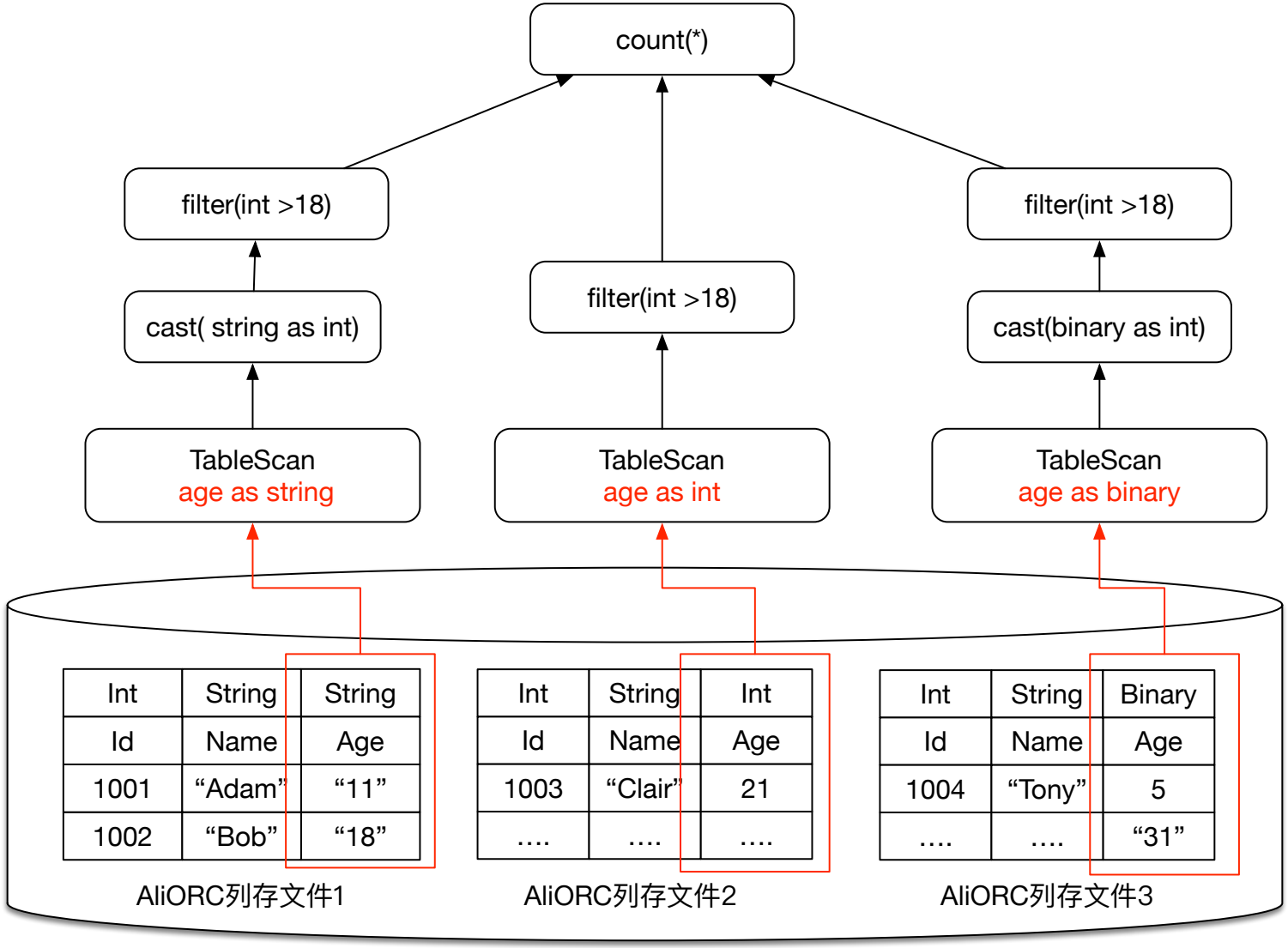
数据文件Byte Order



数据自适应访问-Column Pruning

- 数据类型跨文件变化
- 自适应类型转换，动态类型匹配



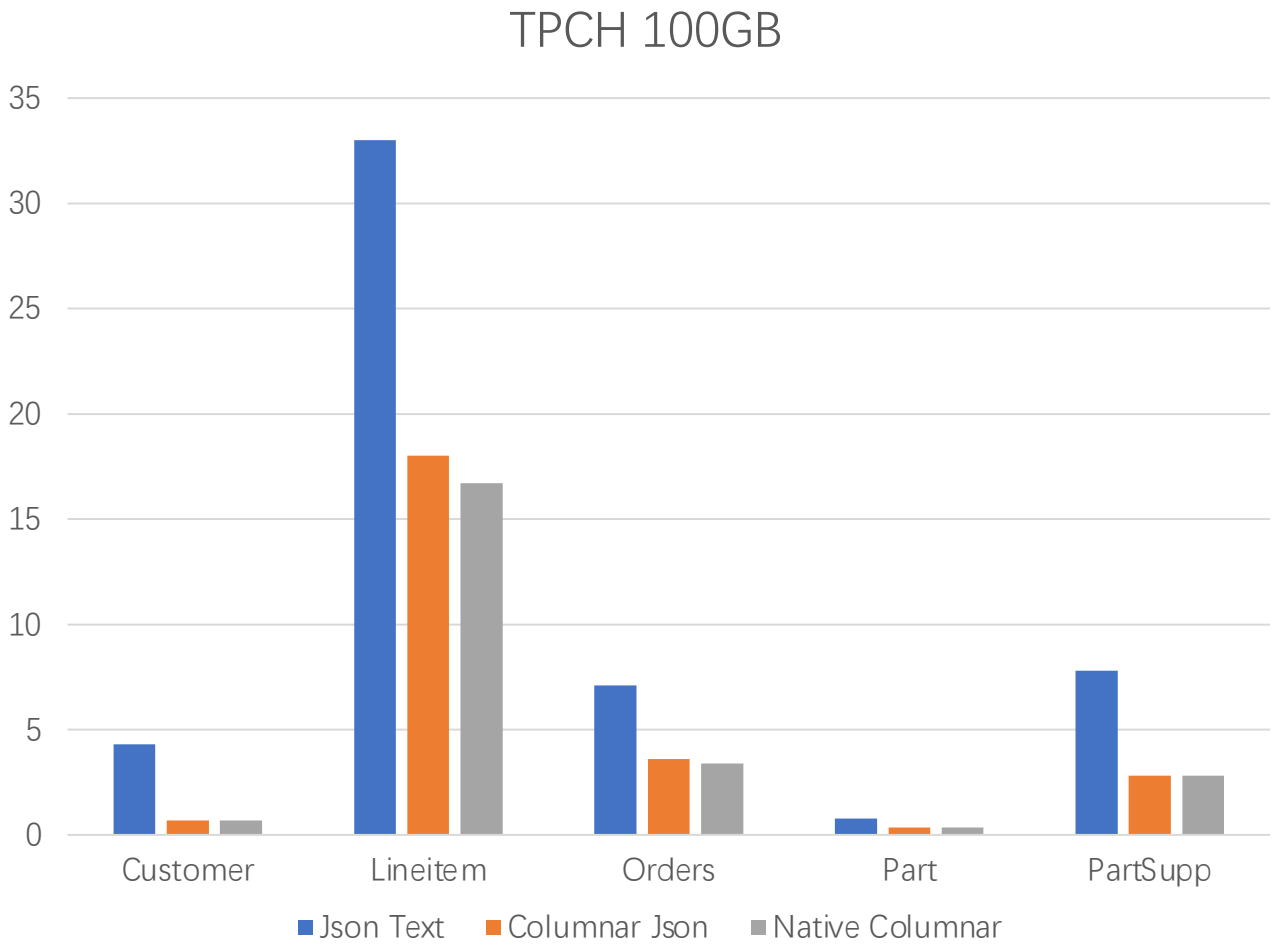


04
收益分析

收益分析—存储收益

- 测试数据集：TPCH
- 测试方案：
  - TPCH Table合并成Json Object
  - 每一列相当于一个Json Attribute

```
{
        "l_orderkey": 1,
        "l_partkey": 1551894,
        "l_suppkey": 76910,
        "l_linenumber": 1,
        "l_quantity": 17,
        "l_extendedprice": 33078.94,
        "l_discount": 0.04,
        "l_tax": 0.02,
        "l_returnflag": "N",
        "l_linestatus": "O",
        "l_shipdate": "1996-03-13",
        "l_commitdate": "1996-02-12",
        "l_receiptdate": "1996-03-22",
        "l_shipinstruct": "DELIVER IN PERSON",
        "l_shipmode": "TRUCK",
        "l_comment": "egular courts above the"
}
```



- 相比JsonText，Json列存化存储空间节省50%
- Json列存化与原生列存存储效率接近

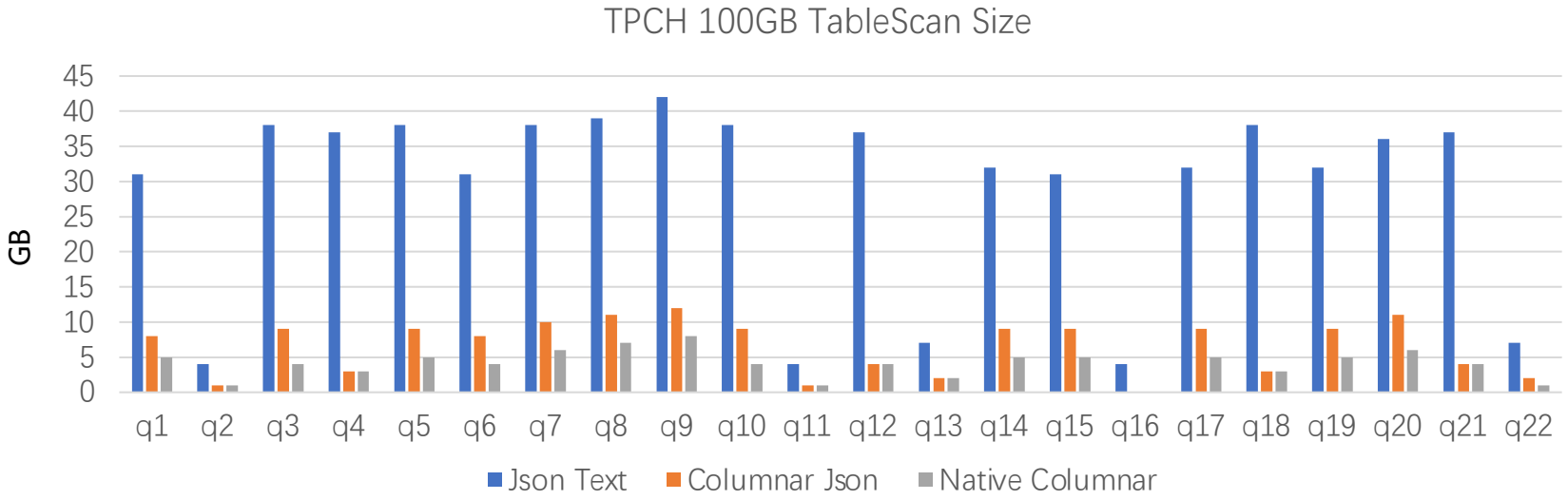
收益分析—查询性能收益

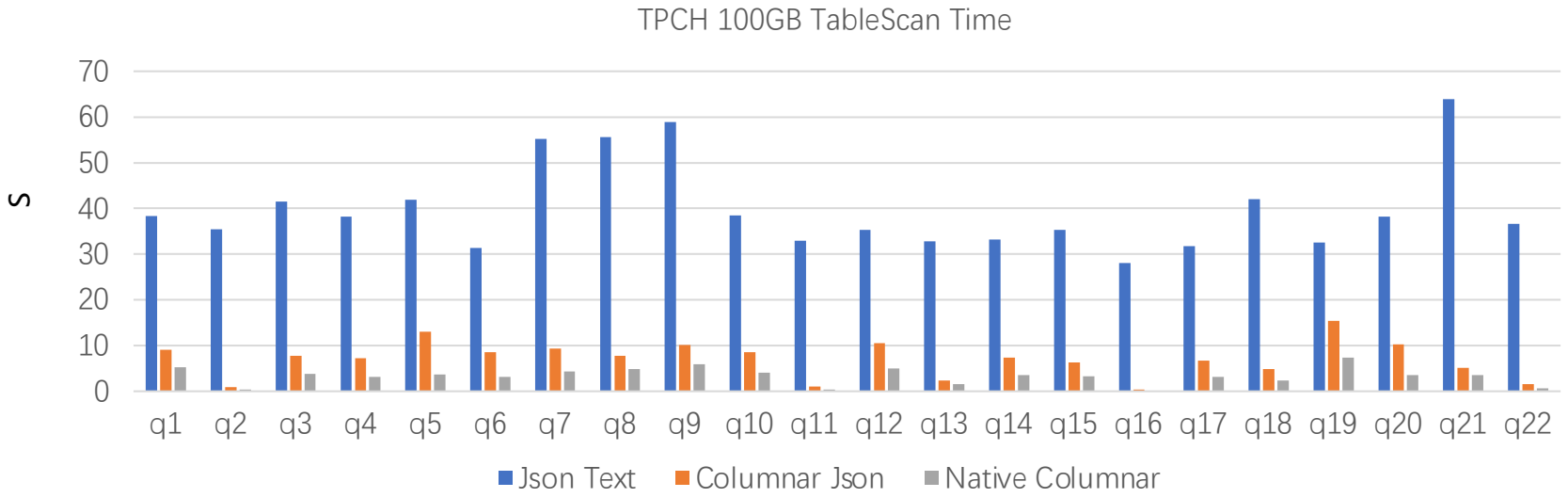
Json列存化相比Json Text

- 读表数据减少80% ~ 90%
- Table时间减少60% ~ 90%

Json列存化相比原生列存，性能仍有提升空间

更精确的数据类型识别与优化





半结构化数据方案总结

全场景适用

高性能分析查询

低成本存储

MaxCompute
5000CU*H免费试用

实时数仓Hologres
5000CU*H免费试用

DataWorks免费试用


感谢观看
阿里云
DataFun.