

直播将于 **19:30** 准时开始，请耐心等待~



需求&问题收集



关注 SelectDB



Apache Doris Variant 半结构化数据分析利器

肖 康 Apache Doris PMC Member

李航宇 Apache Doris Committer



目录

01 半结构化数据的特点与挑战

02 传统方案和局限性

03 Variant数据类型

04 Variant应用场景

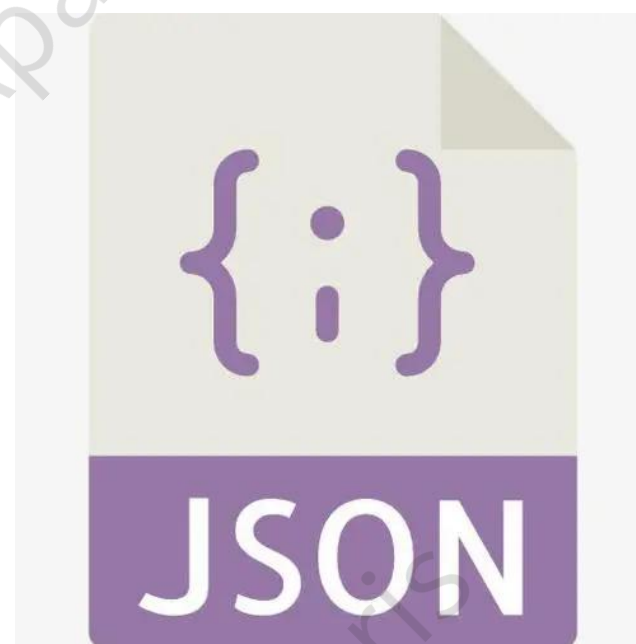
1 半结构化数据的特点与挑战

结构化 半结构化 非结构化数据



结构化数据

结构严格固定，典型代表 关系数据库表



半结构化数据

有结构但不固定，典型代表 JSON XML



非结构化数据

没有明确的结构，典型代表 文本、音视频

半结构化数据的特点

From Wikipedia, the free encyclopedia

Semi-structured data^[1] is a form of **structured data** that **does not obey the tabular structure of data models** associated with **relational databases** or other forms of **data tables**, but nonetheless **contains tags or other markers to separate semantic elements** and enforce hierarchies of records and fields within the data. Therefore, it is also known as **self-describing structure**.

In semi-structured data, **the entities belonging to the same class may have different attributes** even though they are grouped together, and the attributes' order is not important.

不遵循结构化表模型

自描述结构但不固定

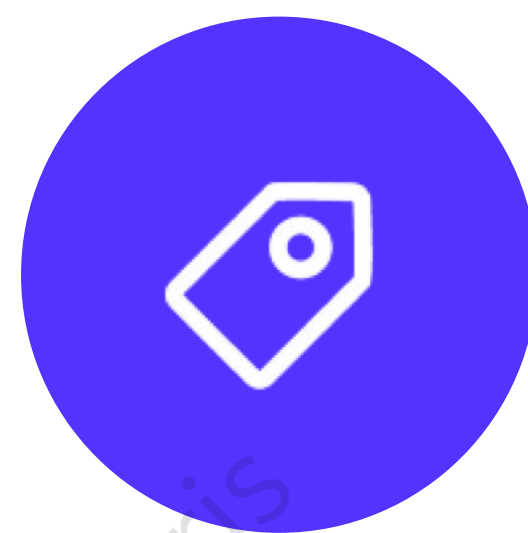
常有嵌套结构

半结构化数据分析的挑战



灵活schema

- 字段增减、类型变化
- 嵌套结构



高效存储

- 存储空间
- 读写效率



高性能分析

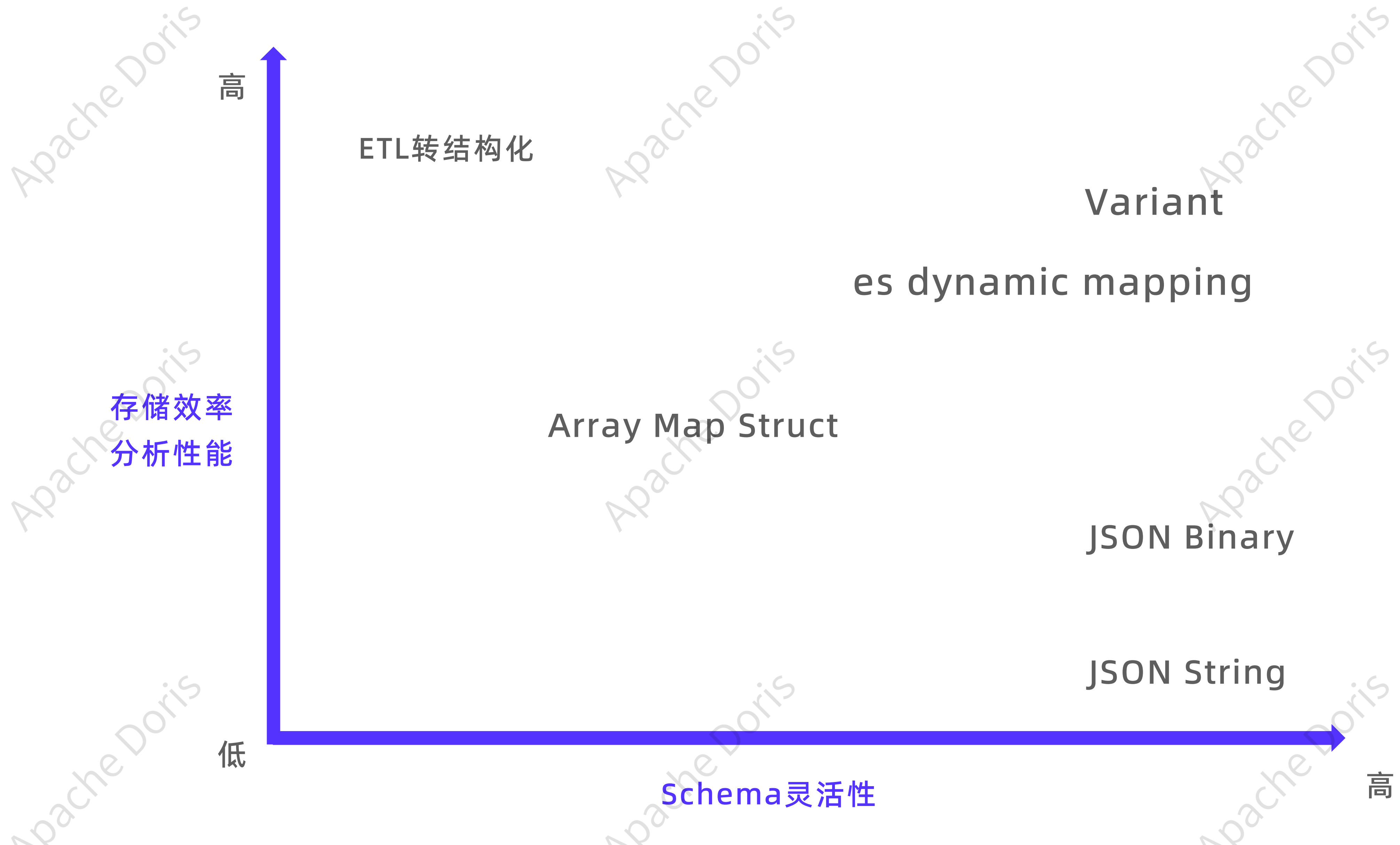
- 分组、聚合
- 过滤

2 传统方案和局限性

传统方案和局限性

方案	优势	问题
ETL 转成结构化	列式存储压缩率高，分析性能好	字段变化需要改表结构，非常不灵活
Array Map Struct	列式存储压缩率高	分析性能一般 只能存储预定义好的嵌套结构
JSON String	schema灵活，可存储任意文本	行存储压缩率低 分析性能低，因为要读取整个文本并解析
JSON Binary	schema灵活，可存储任意JSON	行存储压缩率低 分析性能比String好，但仍然比较低

传统方案的局限性



高

ETL转结构化

Variant

es dynamic mapping

存储效率
分析性能

Array Map Struct

JSON Binary

JSON String

低

Schema灵活性

高

3 Doris Variant 半结构化数据类型

Variant - 专为半结构化数据设计

高效存储任意灵活 schema 的 JSON 半结构化数据，并提供高性能分析能力



- 自动识别 JSON 字段和类型
- 扩展 DESC 展示 schema



- JSON 字段拆成子列存储，压缩率高
- 稀疏列合并避免过多子列



- 列式存储聚合分析、过滤性能高
- 支持索引进一步提升过滤性能

Variant 使用介绍

```
CREATE TABLE IF NOT EXISTS github_events (  
  id BIGINT NOT NULL,  
  type VARCHAR(30) NULL,  
  actor VARIANT NULL,  
  repo VARIANT NULL,  
  payload VARIANT NULL,  
  public BOOLEAN NULL,  
  created_at DATETIME NULL,  
  INDEX idx_payload (`payload`) USING INVERTED PROPERTIES("parser" = "english")  
)  
DUPLICATE KEY(`id`)  
DISTRIBUTED BY HASH(id) BUCKETS 10
```

Variant 使用介绍

```
-- 未开启扩展列展示
```

```
mysql> desc github_events;
```

Field	Type	Null	Key	Default	Extra
id	BIGINT	No	true	NULL	
type	VARCHAR(30)	Yes	false	NULL	NONE
actor	VARIANT	Yes	false	NULL	NONE
repo	VARIANT	Yes	false	NULL	NONE
payload	VARIANT	Yes	false	NULL	NONE
public	BOOLEAN	Yes	false	NULL	NONE
created_at	DATETIME	Yes	false	NULL	NONE

```
-- 开启Variant扩展列展示
```

```
mysql> set describe_extend_variant_column = true;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> desc github_events;
```

Field	Type	Null	Key
id	BIGINT	No	true
type	VARCHAR(*)	Yes	false
actor	VARIANT	Yes	false
actor.avatar_url	TEXT	Yes	false
actor.display_login	TEXT	Yes	false
actor.id	INT	Yes	false
actor.login	TEXT	Yes	false
actor.url	TEXT	Yes	false
created_at	DATETIME	Yes	false
payload	VARIANT	Yes	false
payload.action	TEXT	Yes	false
payload.before	TEXT	Yes	false
payload.comment.author_association	TEXT	Yes	false
payload.comment.body	TEXT	Yes	false
...			

Variant 使用介绍

```
mysql> SELECT
->     count() FROM github_events
->     WHERE cast(payload['comment']['body'] as text) MATCH 'doris';

+-----+
| count() |
+-----+
|      3 |
+-----+

1 row in set (0.04 sec)
```

Variant 性能优势

方案	存储空间	第一次查询时间	第二次查询时间	第三次查询时间
预定义静态列	12.618 GB	233.79 s	86.02 s	83.03 s
Variant 类型	12.718 GB	248.66 s	94.82 s	92.29 s
JSON 类型	35.711 GB	大部分超时	789.24 s	743.69 s

4 Variant 应用场景

应用场景（一） - Log

半结构化数据：log 扩展字段

```
time: 2024-04-09 16:06:03.684886
source: "fluentd_tcp"
host: "host1"
status: info
filepath: "path1"
container_id: "xxx"
container_name: "xxx"
message: "....."
ext: {
  "nameSpace": "kube-public",
  "level": "info",
  "service": "console-media",
  "category": "debug",
  "tag": "[]",
  "serviceID": "console-media-6b57fc7758-nplck",
  "date_ns": 886517,
  "index": "default",
  "cluster": "UcloudKubernetes"
}
```

建表和查询

```
CREATE TABLE log (
  `time` datetime(6) NULL,
  `source` text NULL,
  `host` text NULL,
  `status` text NULL,
  `filepath` text NULL,
  `container_id` text NULL,
  `container_name` text NULL,
  `message` text NULL,
  `ext` variant NULL,
  INDEX idx1 (`message`) USING INVERTED
  PROPERTIES("parser" = "chinese")
)

SELECT
  hour_floor(`time`) as hour,
  cast(ext['service'] as text) as service,
  count() as cnt
FROM log WHERE time >= t1 AND time < t2
AND (`status` = "error") AND (`source` = "abc")
AND ext['nameSpace'] = "bcd"
GROUP BY hour, service
ORDER BY hour LIMIT 100;
```

优势

- 允许字段类型变化，不会因为数据变化而写入异常
- 自动处理扩展字段增减，过期删除的数据字段不会残留
- 相比 String 或 JSON 压缩率高

应用场景（二） - Trace

半结构化数据：trace 属性字段

```
time: 2024-04-09 16:13:51.381141
message: "...
__namespace: "tracing"
source: "opentelemetry"
service: "iov-fleet-mqtt"
operation: "fleet/ListFleets"
status: "ok"
parent_id: "0"
resource: "fleet/ListFleets"
span_id: "xxxx"
trace_id: "xxxx"
duration: 3810
start: 1712650431181241
attrs:
  "rpc_system": "grpc",
  "rpc_method": "ListFleets",
  "service_sub": "iov-fleet-mqtt",
  "k8s_cluster": "sh-test-uk8s",
  "date_ns": 141210,
  "rpc_service": "fleet.FleetService",
  "service_name": "iov-fleet-mqtt",
  "rpc_grpc_status_code": 0
}
```

建表和查询

```
CREATE TABLE trace (
  `time` datetime(6) NULL,
  `message` text NULL,
  `source` text NULL,
  `service` text NULL,
  `endpoint` text NULL,
  `operation` text NULL,
  `status` text NULL,
  `parent_id` text NULL,
  `resource` text NULL,
  `span_id` text NULL,
  `trace_id` text NULL,
  `duration` bigint(20) NULL,
  `start` bigint(20) NULL,
  `attrs` variant NULL
)

SELECT
  `time`, `message`, `source`, `service`,
  `endpoint`, `operation`, `status`, `parent_id`,
  `resource`, `span_id`, `trace_id`, `duration`,
  `start`, cast(`attrs` as text) as `attrs`
FROM trace
WHERE time >= t1 AND time < t2
AND `service` = "pms-java-gate"
AND attrs['error_stack'] IS NOT NULL
ORDER BY `time` DESC LIMIT 50
```

优势

- 按扩展字段筛选比 JSON 和 MAP 效率高
- 相比 String 或 JSON 压缩率高
- trace 属性字段自动根据数据扩充

应用场景3 - IoT 车联网

半结构化数据：设备标签字段

```
date: 20240114
plate_name: "xxx"
base_dir: "xxx"
pack_dir: "xxx"
naive_gt_dir: "xxx"
gt_dir: "xxx"
sync_dir: "xxx"
sync_index: [1755]
lidar_dir: "xxx"
calib_dir: "xxx"
weather: NULL
scene: NULL
tags: {
  "a": 2,
  "b": [0,1,0,0,1],
  "c": [0,0,0,0,0],
  "d": 0
}
```

建表和查询

```
CREATE TABLE perception (
  `date` VARCHAR(10),
  `plate_name` TEXT,
  `base_dir` TEXT,
  `pack_dir` TEXT,
  `naive_gt_dir` TEXT,
  `gt_dir` TEXT,
  `sync_dir` TEXT,
  `lidar_dir` TEXT,
  `calib_dir` TEXT,
  `weather` TEXT,
  `scene` TEXT,
  `tags` VARIANT
)
```

优势

- OLAP 分析功能强，列式存储分析性能好
- 支持上千稀疏列
- 相比 String 或 JSON 压缩率高

欢迎加入 Apache Doris

加入社区用户微信群

扫码添加 Doris 小助手，备注“加群”

提问赢好礼

提问被选中回复的小伙伴，请添加小助手微信领取 Doris 精美周边，数量有限，先到先得~

Doris 问答论坛

地址：ask.selectdb.com



Thanks !

