

数据来源：数据库产品上市商用时间



第十三届中国数据库技术大会

DATABASE TECHNOLOGY CONFERENCE CHINA 2022

数据智能 价值创新



线上直播 | 2022/12/14-16



让数据库会思考

SQL优化技术的挑战与未来

魏可伟

浪潮开务数据库创新研究院院长

目录

- 01 数据库优化回顾
- 02 优化技术面临的挑战
- 03 优化: Inside-out + Outside-in

01

数据库优化回顾

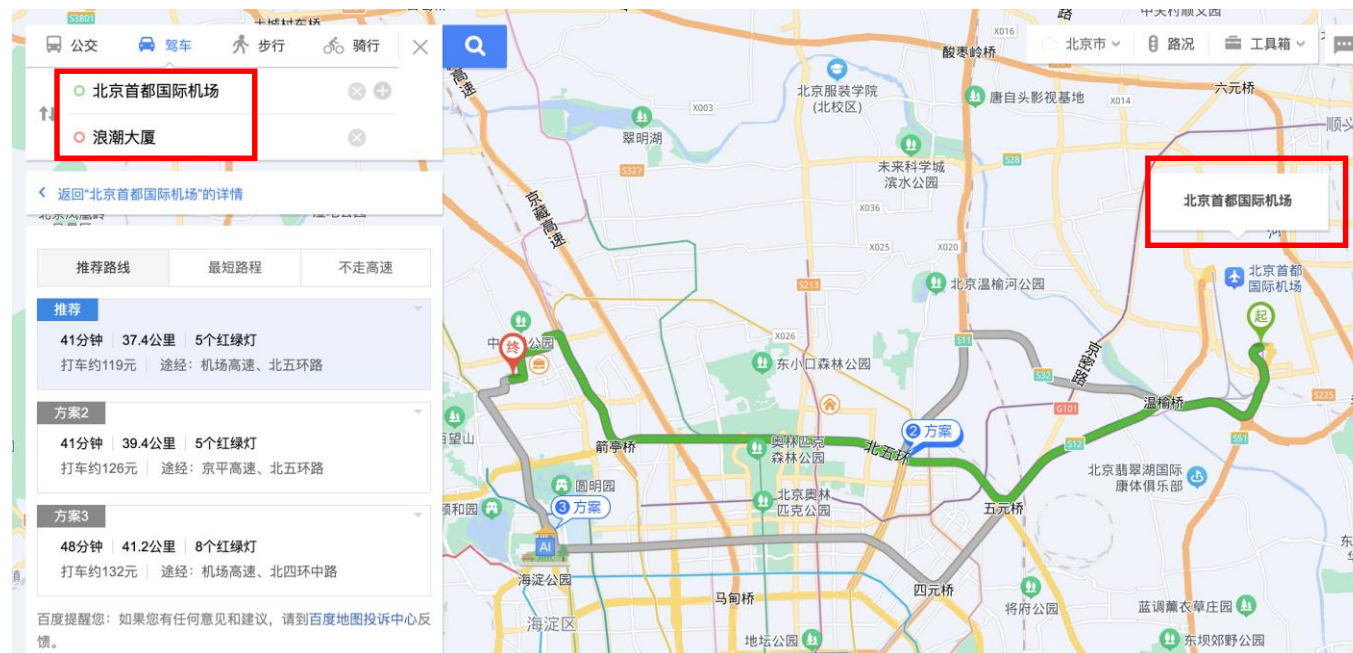
什么是数据库查询优化

- SQL 语言只描述要什么，不描述怎么做
- 查询优化技术为给定的查询(SQL)选择最高效的执行计划

描述式语言



我在首都机场;
我要去浪潮大厦

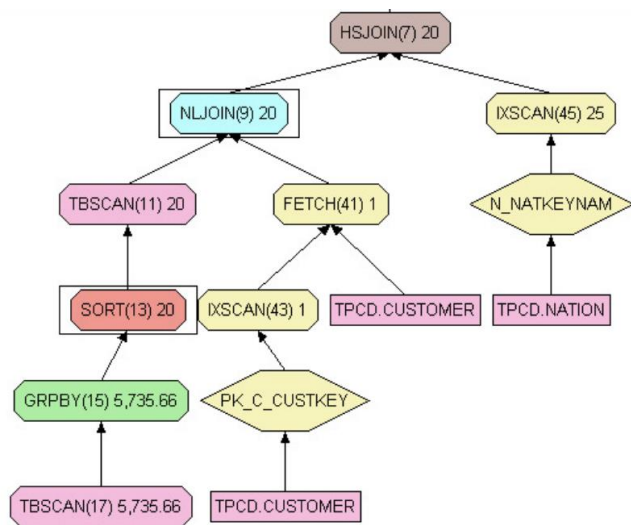


优化器工作原理

基于规则——RBO

基于成本——CBO

执行计划 (Access Plan)



查询



数据库设计



统计信息



数据库及应用参数



优化: Inside-out vs Outside-in

Inside
不断改进的优化器

基于规则——RBO

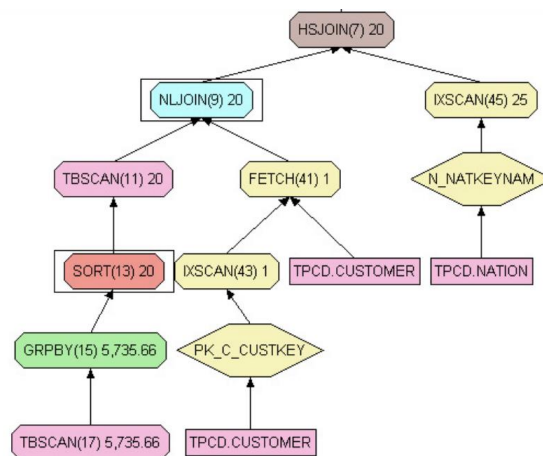
更多规则

基于成本——CBO

更大的搜索空间

更准确的成本估算

更好的
执行计划



Outside: 数据库与应用优化
+ 数据库运维工具

改写查询与应用



优化数据库设计
(索引、分区、分表等)



及时收集
所需统计信息



调整数据库
及应用参数





优化技术面临的挑战

传统查询优化技术所面临的挑战

- 不完整/不准确的统计信息
- 编译时优化的限制
- 单一计划的局限性

- 缺乏全局视角
- 缺失计划生命周期管理
- 缺少异构环境支持

对不完整/不准确的信息缺乏免疫力

- 统计信息是查询优化的基础

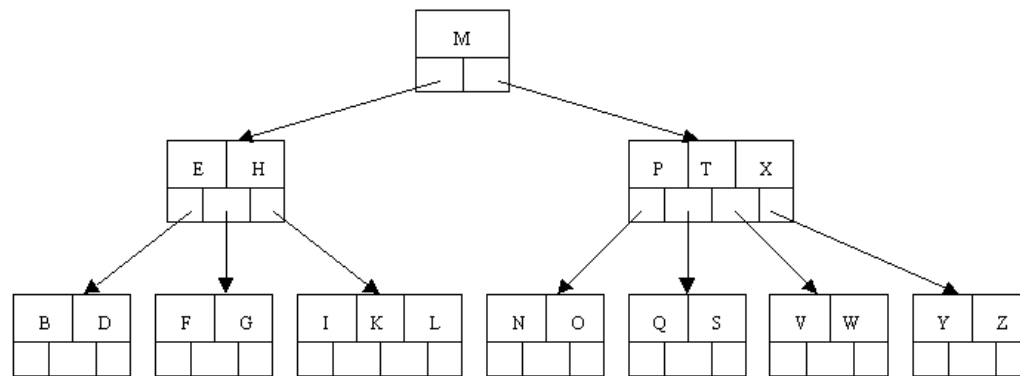
```
SELECT *
FROM ORDERS
WHERE ORDER_DATE = ?;
```

表扫描 (Table Scan)

ID	OrderID	ProductID	UnitPrice	Quantity	Discount
1	10248	11	14	12	0
2	10248	42	9.8	10	0
3	10248	72	34.8	5	0
4	10249	14	18.6	9	0
5	10249	51	42.4	40	0
6	10250	41	7.7	10	0
7	10250	51	42.4	35	0.15
8	10250	65	16.8	15	0.15
9	10251	22	16.8	6	0.05
10	10251	57	15.6	15	0.05
11	10251	65	16.8	20	0
12	10252	20	64.8	40	0.05

VS

索引扫描 (Index Scan)



对不完整/不准确的信息缺乏免疫力

统计信息是查询优化的基础

```
SELECT *  
FROM ORDERS  
WHERE ORDER_DATE = ?;
```

如果统计信息不准确



对不完整/不准确的信息缺乏免疫力

统计信息是查询优化的基础

```
SELECT *  
FROM ORDERS  
WHERE ORDER_DATE = ?;
```

被动导致迟滞

Inside

- 基于规则规避风险
- 利用实时统计信息



Outside

- 及时收集统计信息

编译时优化的限制

编译时的信息缺失带来的挑战

```
SELECT *
FROM CUSTOMERS
WHERE BIRTH_DATE > ?;
```

表扫描

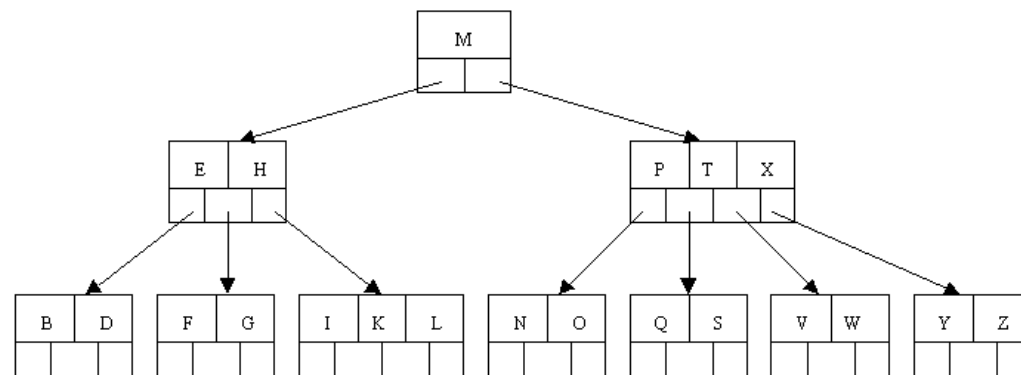
ID	OrderID	ProductID	UnitPrice	Quantity	Discount
1	10248	11	14	12	0
2	10248	42	9.8	10	0
3	10248	72	34.8	5	0
4	10249	14	18.6	9	0
5	10249	51	42.4	40	0
6	10250	41	7.7	10	0
7	10250	51	42.4	35	0.15
8	10250	65	16.8	15	0.15
9	10251	22	16.8	6	0.05
10	10251	57	15.6	15	0.05
11	10251	65	16.8	20	0
12	10252	20	64.8	40	0.05

如果传入参数是2022年?

如果传入参数是1900年?

VS

索引扫描



编译时优化的限制

编译时的信息缺失带来的挑战

```
SELECT *  
FROM CUSTOMERS  
WHERE BIRTH_DATE > ?;
```

如果传入参数是2022年?

如果传入参数是1900年?

缺少整合方案

Inside

- 基于规则规避风险
- 运行时重优化



Outside

- 不使用参数
- 使用优化器HINT

单一执行计划的局限性

单一执行计划不能满足全部需要

```
SELECT *
FROM CUSTOMERS
WHERE SVIP = ?;
```

表扫描

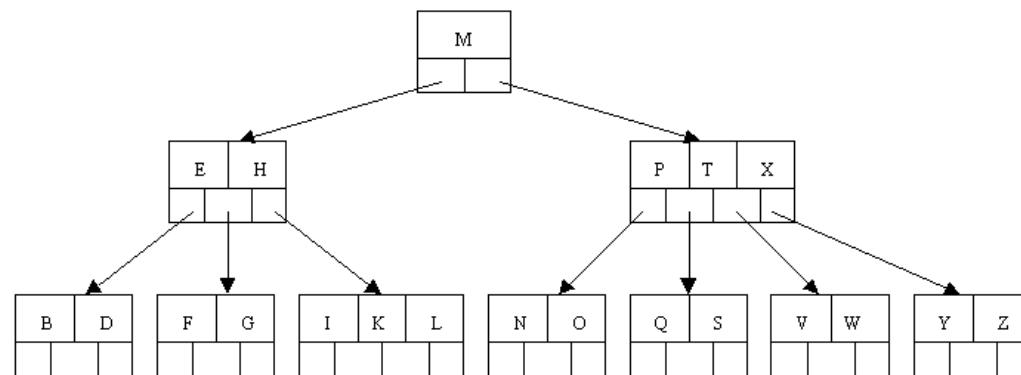
ID	OrderID	ProductID	UnitPrice	Quantity	Discount
1	10248	11	14	12	0
2	10248	42	9.8	10	0
3	10248	72	34.8	5	0
4	10249	14	18.6	9	0
5	10249	51	42.4	40	0
6	10250	41	7.7	10	0
7	10250	51	42.4	35	0.15
8	10250	65	16.8	15	0.15
9	10251	22	16.8	6	0.05
10	10251	57	15.6	15	0.05
11	10251	65	16.8	20	0
12	10252	20	64.8	40	0.05

99900位客户不是SVIP

100位客户是SVIP

VS

索引扫描



单一执行计划的局限性

单一执行计划不能满足全部需要

```
SELECT *  
FROM CUSTOMERS  
WHERE SVIP = ?;
```

99900位客户不是SVIP

100位客户是SVIP

成本/性能不能满足要求

Inside

- 持续重优化



Outside

- 改造应用

缺乏全局视角

- 优化是平衡的艺术

建立索引

提升查询性能
增加更新开销

及时收集统计信息

提升优化效果
增加维护成本

重组数据

提升一些查询性能
牺牲另一些查询性能

随机分区

提升插入性能
牺牲查询性能

- 优化的目标是提升全局性能，提升关键应用负载的性能
- 传统优化器“只见一叶”

缺失计划生命周期管理

- 要不要重优化?

收益

利用新的统计信息
利用新的设计（索引等）
利用新的优化器特性

风险

执行路径变化导致性能变差

人工介入难以避免

Inside

- 计划重用



Outside

- 执行计划管理工具
- 灰度部署

异构环境支持

- 新数据库架构带来的新的挑战
 - 跨数据模型的优化
 - 自动选择合适的计算模型
 - 行存、列存、文档(Json)、图等
 - 子计划可以选择不同的计算模型

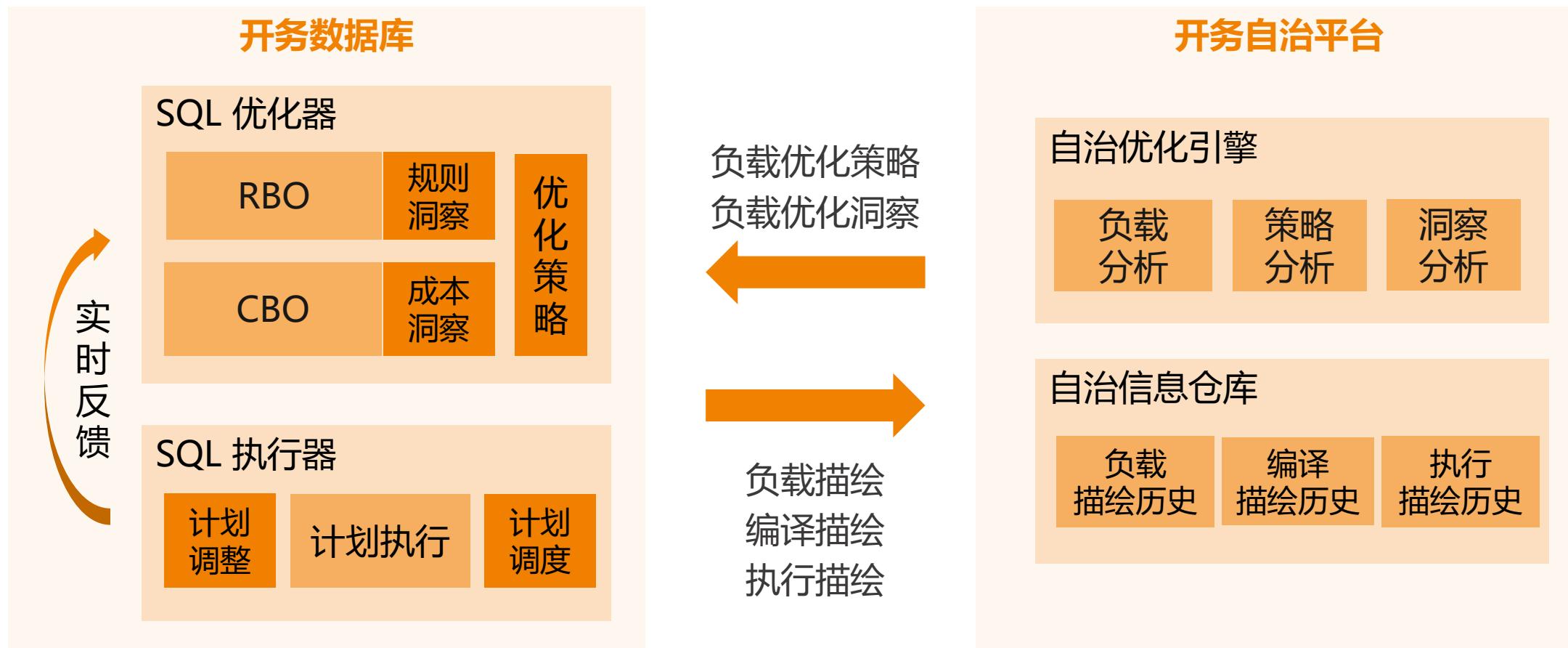
- 跨硬件环境的优化
 - FPGA加速
 - GPU加速



优化：Inside-out + Outside-in

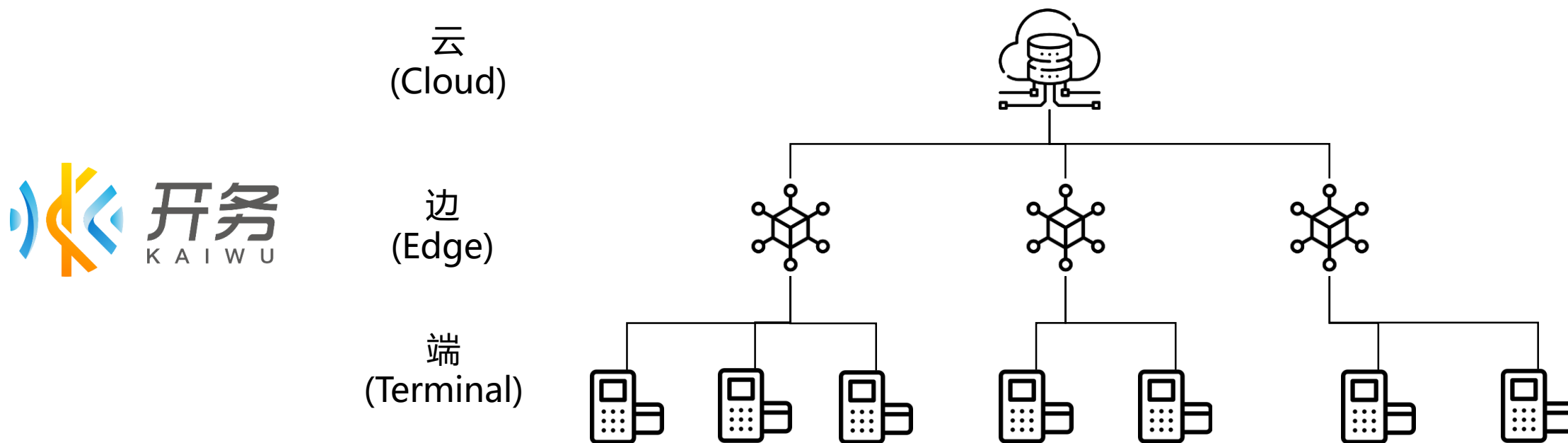
开务数据库：Inside-out + Outside-in

优化不仅仅在数据库内



开务数据库 与 开务数据服务平台

- 开务数据库是面向物联网应用的 **分布式 多模** 数据库。
- 开务数据服务平台(KDP)是以开务数据库为核心的支持 **云边端协同** 的数据平台。



及时收集信息：Inside-out + Outside-in

```
SELECT *  
FROM ORDERS  
WHERE ORDER_DATE = ?;
```

统计信息显示10条记录 ✕

SQL优化器

Inside

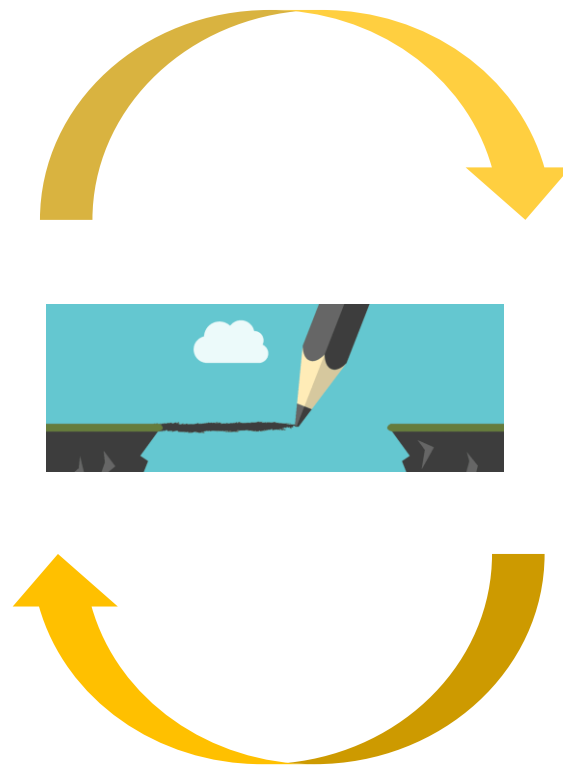
需要但未收集的信息
错误或过时的信息

开务自治平台

Outside

```
SELECT *  
FROM ORDERS  
WHERE PROVINCE = ? AND CITY = ?;
```

缺少多列统计信息 ✕



选择适当时机
只收集必要的统计信息

利用运行时信息：Inside-out + Outside-in

```
SELECT *  
FROM CUSTOMERS  
WHERE BIRTH_DATE > ?;
```

追踪常量实际值

SQL优化器

Inside

开务自治平台

Outside



综合选定最优计划
避免高风险执行计划
根据过滤因子重优化

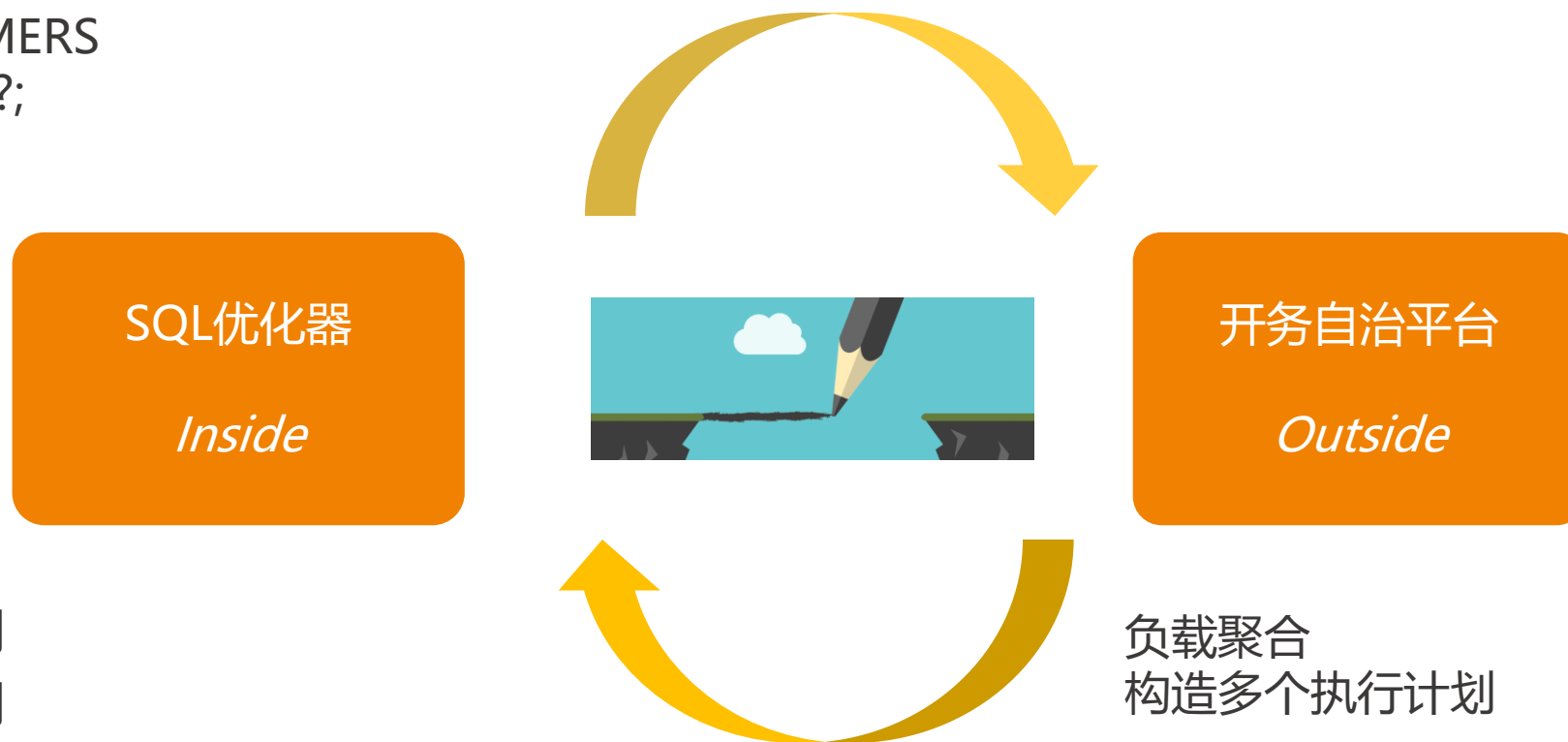
BIRTH_DATE
70前 15%
80后 20%
90后 45%

...

多执行计划动态选取: Inside-out + Outside-in

```
SELECT *  
FROM CUSTOMERS  
WHERE SVIP = ?;
```

追踪常量实际值

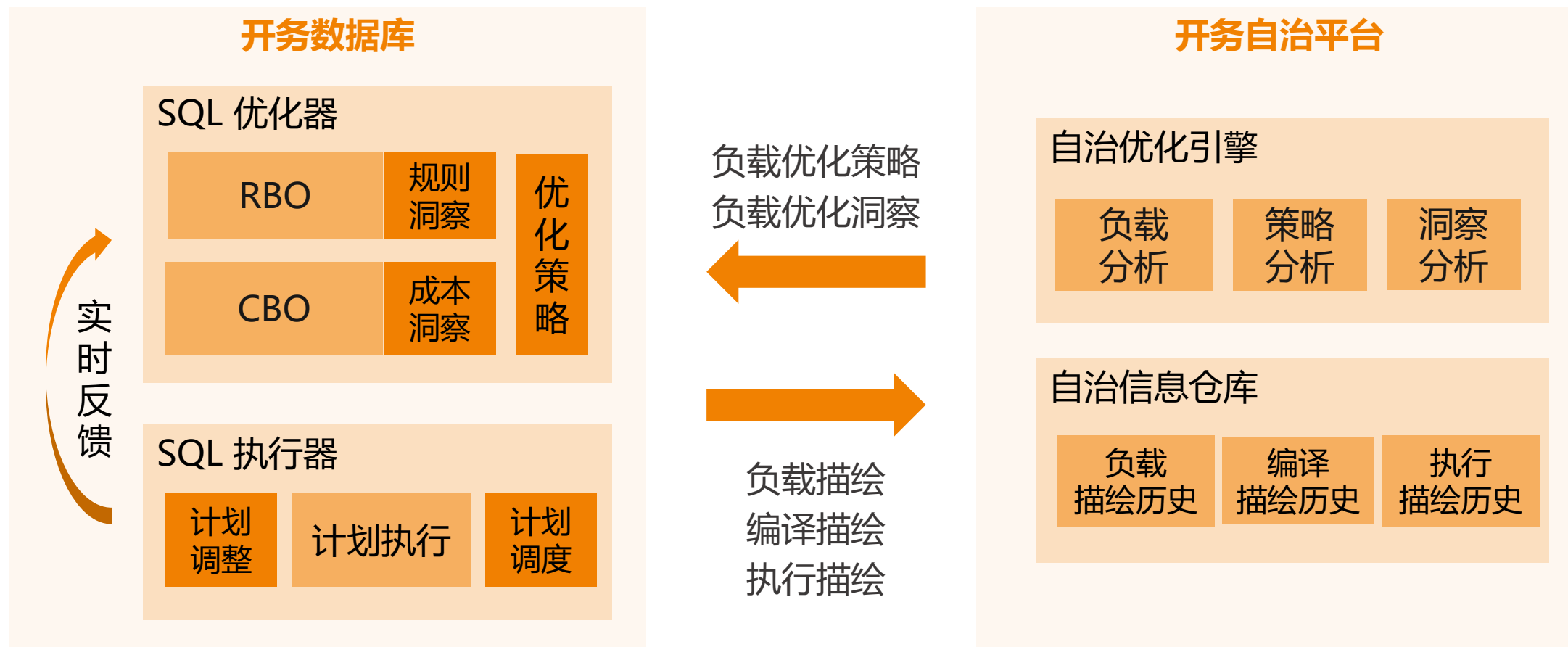


动态选取执行计划
动态修正执行计划

负载聚合
构造多个执行计划

开务数据库：Inside-out + Outside-in

优化不仅仅在数据库内



THANKS



关注开务数据库

SQL Server
vertica
D B 2
G B a s e
O r a c l e
达梦数据库
神舟通用
KingbaseES

2010

2018

openGauss
OceanBase
ArkDB
RASESQL
HotDB
StellarDB
QianBase xTP
云树Shard
GoldenDB
DolphinDB
MatrixDB
DynamoDB
SinoDB
FastData
Galaxybase
KunDB
GDB
GaussDB
PolarDB
TiDB
Spacture
Sequoiadb
CubusDB
ArgoDB
UbiSQL
MongoDB
Tapdata
StarRocks
TiDB
Tapdata
StarRocks