

Oracle 数据库对海量时序化数据的管理策略

公益讲座11: 00准时开始, 请大家先浏览云技术微信公众号技术文章。资料会在各群同步发布, 已入群客户请勿重复入群!



20-23

数据库和云讲座群



甲骨文云技术公众号



B站专家系列课程



* 活动最终解释权归甲骨文公司所有

基于 Oracle 数据库 免费企业数据健康检查

- 及时了解数据库健康状况，发现并解决潜在问题
- 维护数据库系统良好状态，保护数据资产的安全
- 提升数据库性能、稳定性和安全性，降低业务风险

免费咨询热线：
400-699-8888

ORACLE

Oracle 数据库对海量时序化数据的管理策略

Sam Liang

2/2/2024

时序数据的定义

- 时序数据的产生来源
 - 设备、传感器、仪表、服务器、台式机、智能手机
- 时序数据通常都包括哪些内容:
 - **时间戳**: 表示产生数据时候的Timestamp
 - **源地址**: Meter id, IP address, Mobile Device, Vehicle Id 等
 - **数据详情**: 数据值域



Time: 6/16/21:12:05pm
Meter ID: X45-123
Details: {
 Electricity KW-hrs: 0.4
}



Time: 6/17/21:10pm
Host IP address:
192.168.1.125
Details: {
 Free Memory: 12.5GB
 Free Swap: 563.56GB
 Page Faults: 1566
}



Time: 6/11/21:12:12pm
Phone ID: 1955ABC
Details: {
 Location: 37.6N/112.2W
 Battery Level: 60%
}

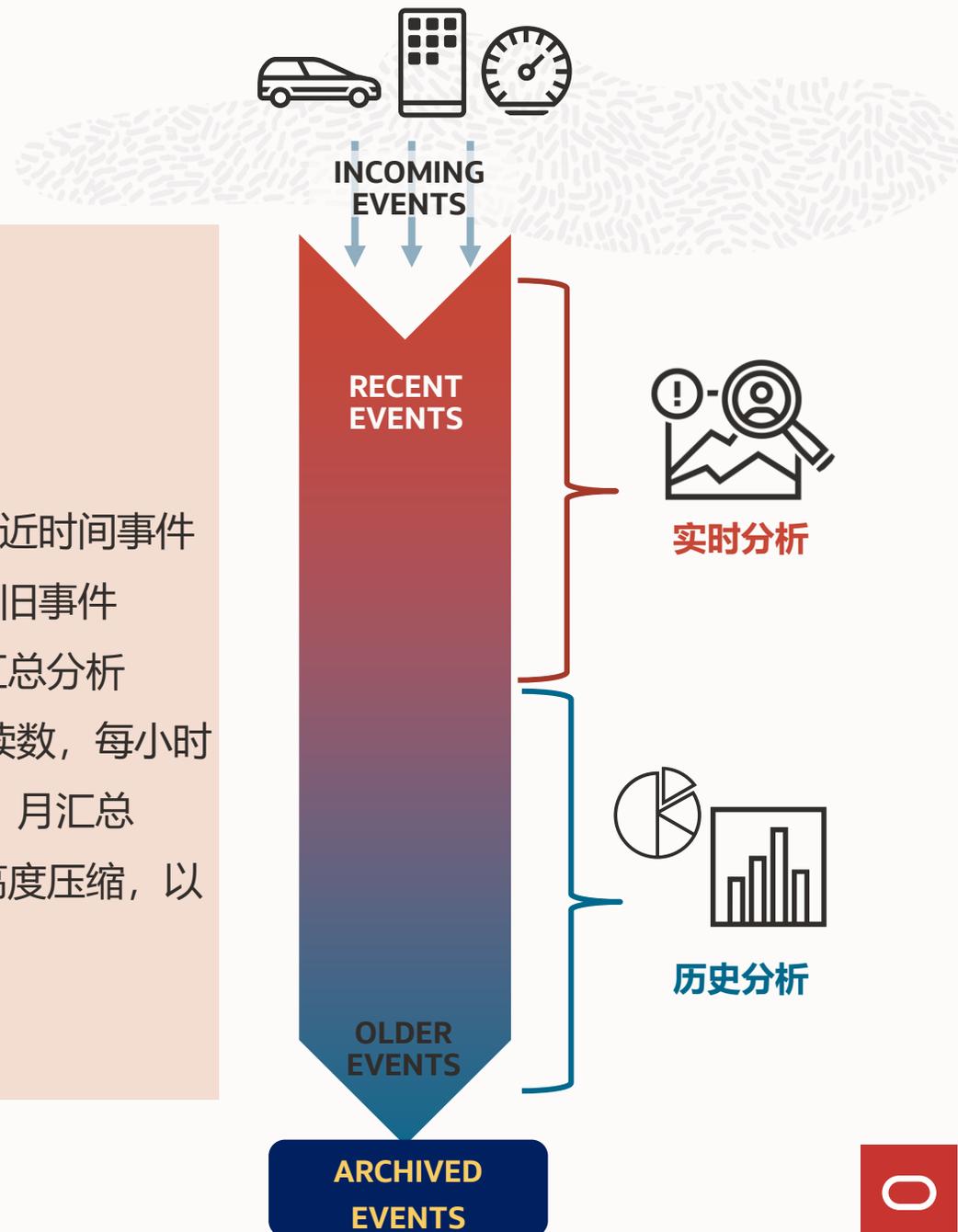
时序数据的属性

高到达率

- 大多数事件处理系统从不同的来源接收大量事件
 - 计费系统每小时收到数百万个使用读数
 - 云监控系统每天从服务器、网络交换机、存储系统接收数十亿个事件

高过时率

- 事件时间过期概念：
 - 实时分析需要最近时间事件
 - 历史报告所需的旧事件
- 基于时间进行压缩和汇总分析
 - 仪表数据的分钟读数，每小时汇总以及天汇总、月汇总
- 存档的事件数据需要高度压缩，以最大限度地节省空间



时序数据流的处理特点

时序数据流处理

持续摄入

持续摄取高频事件数据

及时分析

与批处理不同，时序数据分析需要持续的第一时间完成是对动态数据执行的

数据重组

事件数据随着时间的推移而被越来越多地压缩和总结，然后最终作为存档数据保存下来



时序事件流对系统架构的要求



灵活的数据模型

- 灵活的数据模型，方便储存不同的事件源值
- 有利于事件源值结构的更改



高速摄取

- 能够满足每天数十亿个事件的快速写入
- 应用端负载和数据库端的负载



实时分析

- 即时对事件数据进行分析
- 支持对检测、欺诈、火灾预警、燃气泄漏等关键预警业务



丰富的流式分析能力

- 通过高级分析功能来对跨事件流数据进行窗口操作，如过滤、聚合和汇总等分析



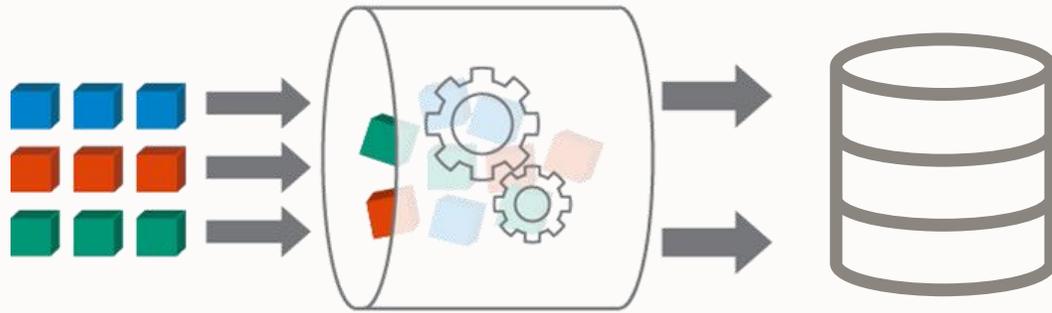
自动数据生命周期管理

- 需要自动对数据进行压缩、汇总、归档，避免数据无限增长



时序事件流处理架构的选择

单一的流处理引擎



Input Data

动态数据:

- 流数据处理引擎

静态数据:

- 数据库

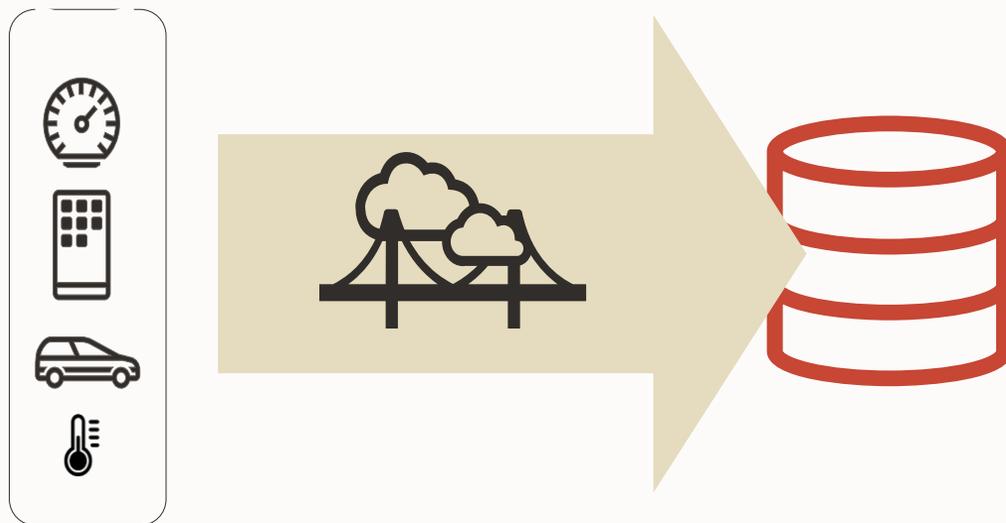
- 一些系统使用应用层流处理器来处理时序事件数据，而仅使用数据库来处理静止的事件数据
- 这种方法有一些缺点：
 - 应用程序层引擎通常无法访问所有数据，通常会导致不支持全量的数据分析
 - 在中间层处理引擎和存档存储之间移动数据需要复杂的数据集成
 - 数据分割管理成本：可管理性、安全性、兼容性等

时序事件流处理架构的选择

多模数据库



- **基于Oracle 融合数据库构建时序数据管理架构**
 - 出于分析目的访问所有数据的好处
 - 高效的自动化数据管理策略，如混合压缩、数据分区、数据加密等
 - 成熟稳定的GlodenGate数据集成方案
 - 融合数据库的优势：访问多种处理算法、统一的安全性、可管理性、HA/DR 技术



灵活的数据模型

Oracle JSON处理时序数据

JSON type column 定义json 数据列

- JSON对于表示可变格式的事件有效负载非常有用
- JSON已与Oracle数据库中的SQL完全集成
 - 自由混合JSON和非JSON数据类型
 - 带有JSON路径的简单SQL语法
 - 使用标准点符号进行导航
 - 使用函数索引对任何JSON字段进行索引
 - 函数支持: JSON_EXISTS、JSON_VALUE、JSON_QUERY等
- 内存中的列式处理还允许对JSON数据进行30-60倍的快速分析

```
CREATE TABLE j_purchaseorder
(id          VARCHAR2 (32) NOT NULL PRIMARY KEY,
date_loaded TIMESTAMP (6) WITH TIME ZONE,
po_document JSON);
```

```
INSERT INTO j_purchaseorder
VALUES (
  SYS_GUID(),
  to_date('30-DEC-2014'),
  '{"PONumber"           : 1600,
   "Reference"          : "ABULL-20140421",
   "Requestor"          : "Alexis Bull",
   "User"               : "ABULL",
   "CostCenter"         : "A50",
   "ShippingInstructions" :
     {"name"           : "Alexis Bull",
      "Address"       : {"street" : "200 Sporting Green",
                        "city"    : "South San Francisco",
                        "state"   : "CA",
                        "zipCode" : 99236,
                        "country" : "United States of Ame",
                        "Phone"   : [{"type" : "Office", "number" : "9",
                                      {"type" : "Mobile", "number" : "4",
                                      "Special Instructions" : null,
```

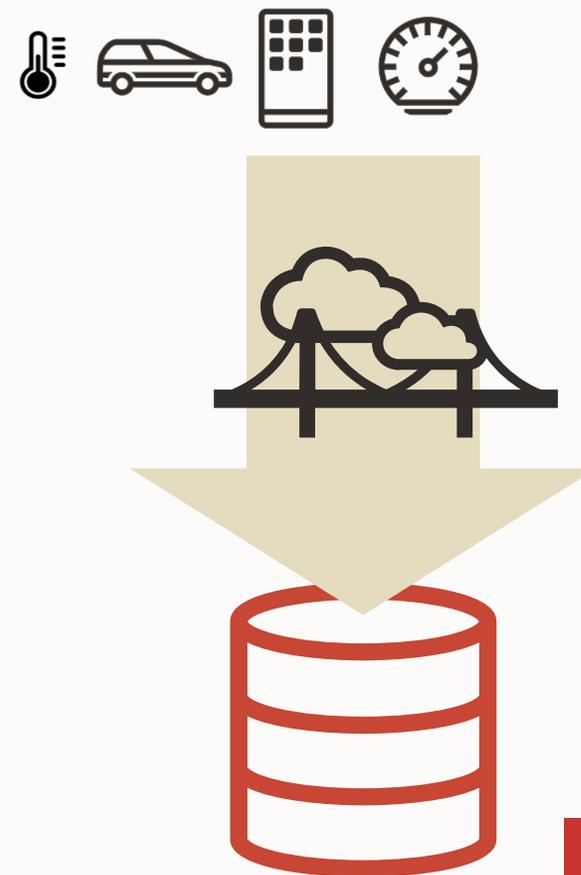
```
SELECT po.po_document.PONumber FROM j_purchaseorder po;
```

高速摄取数据

通过Oracle GoldenGate 快速捕获数据

Leading real-time data capture engine, available as a fully managed service

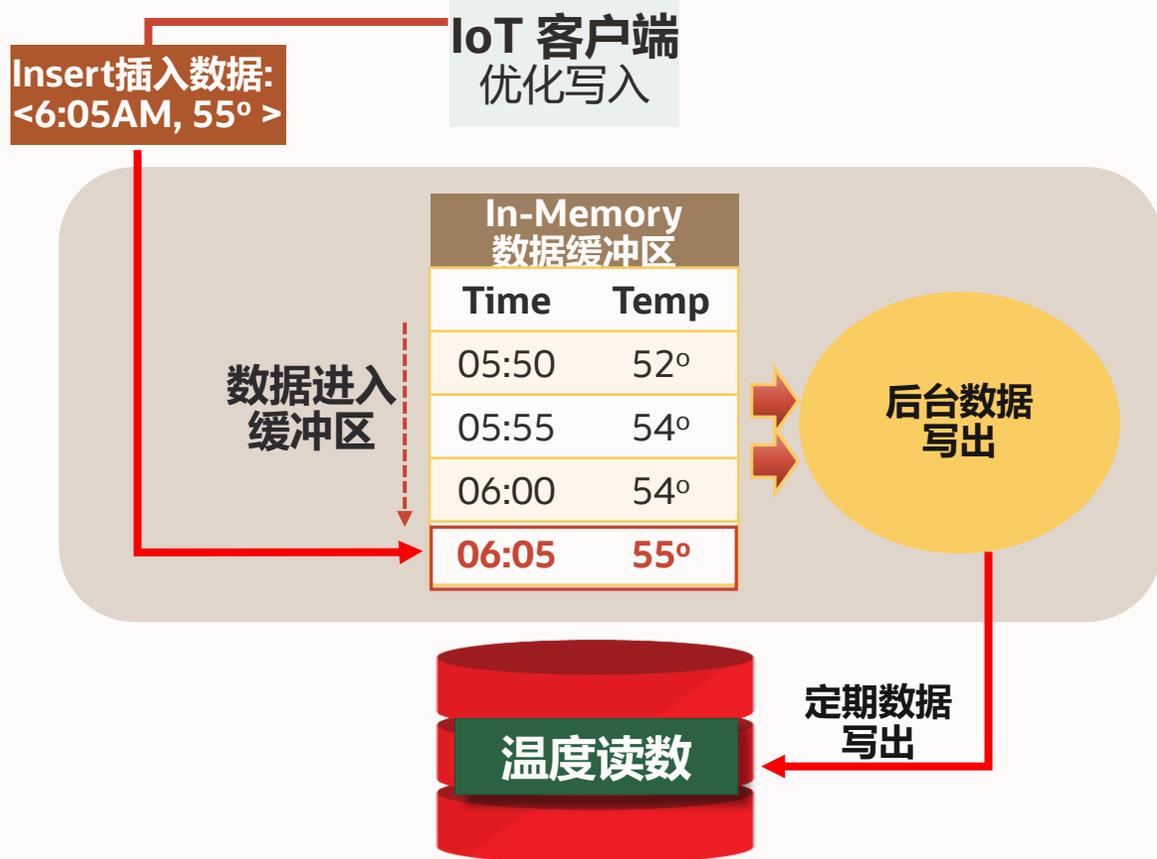
- 实时事件捕获的行业领导者
 - 更改数据捕获 (CDC) 、数据复制、数据集成、流式集成、时间序列等。
 - 除了Oracle数据库之外，还支持尽可能广泛的各种数据源和目标
- 端到端并行最大限度地提高摄取吞吐量
 - 并行捕获与数据库集成并行应用
- 云原生
 - 完全由Oracle管理：升级、修补等。
 - 自动缩放：真正的云弹性，低运营成本
 - 连接到本地，任何Oracle 11.2.0.4+



Memoptmized Rowstore 时序事件数据的内存优化和存储

超快速、内存优化的事件摄取和查询

示例：写入温度读数



新的数据流式写入:

- 定义表声明 MEMOPTIMIZE FOR WRITE
- 客户端执行低延迟写入内存缓冲区
- 后台缓冲区数据流式写出
- 由于服务器发出大批量的延迟写入，因此插入量非常高

性能:

- **吞吐速度是传统的2倍**

读操作: MEMOPTIMIZE FOR READ

在性能上带来的提升:对于基于主键的检索, 吞吐量增加了4倍;



Attribute Clustering

列簇：数据在磁盘上非常接近的以有序的方式进行存储，以有利于最大化事件分析性能

- **Attribute Clustering**

- 在摄取过程中对数据进行预排序，将特定的字段在物理上以非常接近的有序方式存储，大量加速基于字段事件查询
- 最佳的设计，应将存储的事件排序为与查询访问模式匹配
 - 预聚合的索引有利于计数查询
 - 对应用端透明
 - 降低索引维护的开销
 - 获得最佳的压缩性能

```
CREATE TABLE MeterReadings(  
    SourceID VARCHAR2(50),  
    Time      TIMESTAMP,  
    KWhrs     JSON)  
CLUSTER BY LINEAR ORDER  
    (SourceId, Time);
```

	SourceId	Time	KWhrs
↑	1XC22	5:45pm	0.86
	1XC22	5:30pm	0.23
	1XC22	5:15pm	0.4
↓	1XC24	5:45pm	0.5
	1XC24	5:30pm	0.8
	1XC24	5:15pm	1.5
↓	1XC26	5:45pm	0.6
	1XC26	5:30pm	0.9
	1XC26	5:15pm	0.45
↓	1XC28	6:00pm	0.86

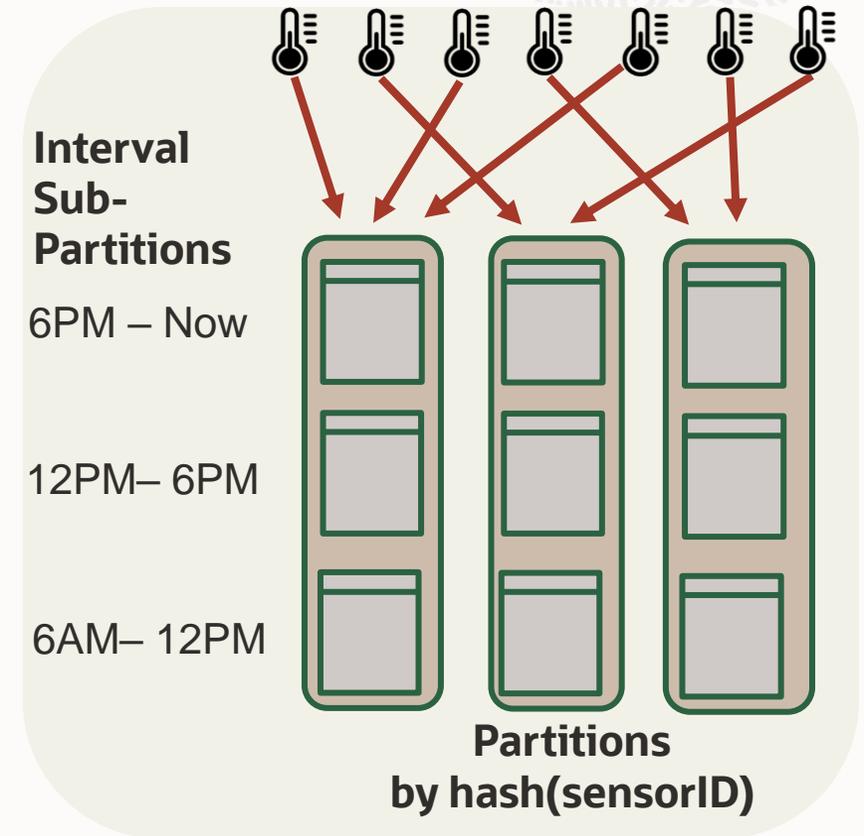
- **线性排序**
 - 根据第一列排序，然后根据第二列排序
 - 数据有序的组织在一起
- **交错排序**



Partitioning

Parallelism and Efficiency for Event Stream Processing

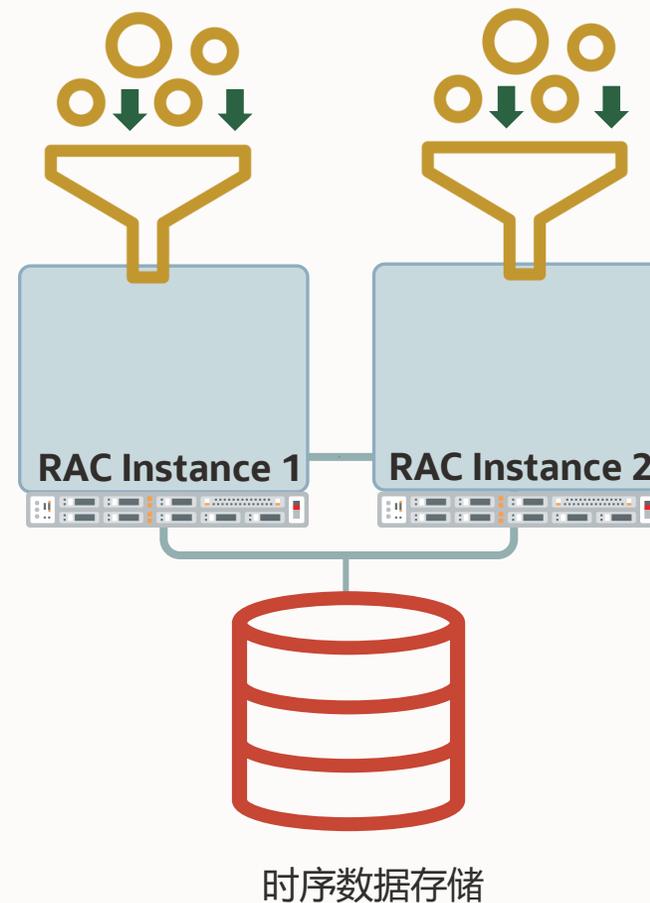
- 将大型表分成多个单元以进行可扩展的摄取
 - 避免单表插入缩放限制
 - 例如，争夺空间分配
- Oracle数据库支持多种分区策略
 - 范围、散列、列表、区间、复合等。
- 事件流数据的典型分区策略是：
 - 按源进行哈希分区以加快摄取速度
 - 按时间间隔进行子分区，以减少分析时的数据访问



Real Application Clusters (RAC)

Industry-Leading Scale-Out For Event Stream Processing

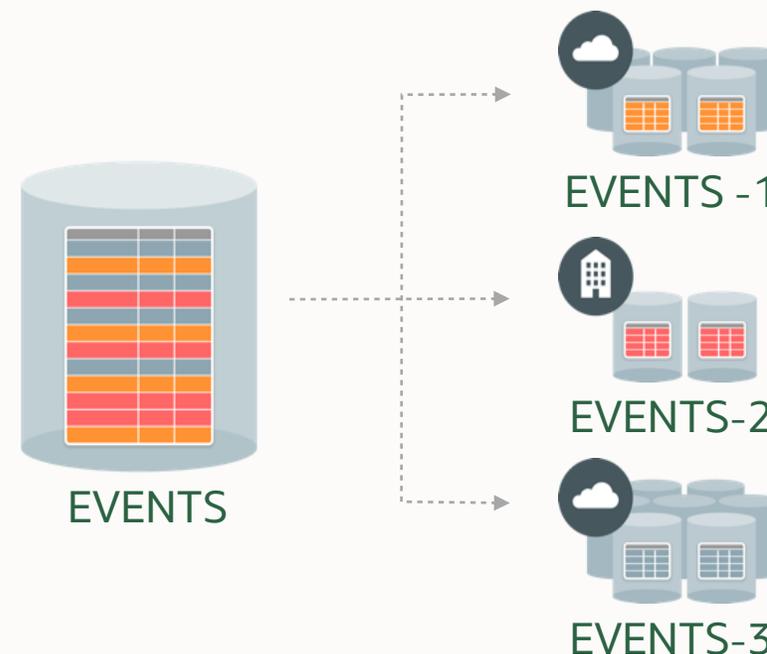
- 透明地跨服务器池扩展，满足高工作负载
- 支持在一个数据库内运行超高速事件摄取以及复杂的事件分析
- 针对 Exadata、Oracle 云和自治数据库进行了优化
 - 扩展性能：更多主机意味着更多的摄取吞吐量
 - 当需要更多吞吐量时，只需添加新主机
 - 扩展容错能力：更多主机意味着更高的可用性
 - 当主机出现故障时，数据库仍然可用



Sharding

Hyperscale Event Stream Processing

- Sharding数据库可天然解决大规模事件流数据扩展问题
 - 避免超大数据库的边缘可扩展性或可用性问题
 - 每个分片都可以通过Data Guard或GoldenGate进行复制
- 基于分片表的SQL查询
 - 基于源作为分片键的事件摄取路由
 - 用于分析的跨分片的查询
 - 在线添加和重组分片
- Sharding可最大化提升数据获取效能的机制
 - 容量和吞吐量的线性可扩展性
 - 由于分片是故障隔离的，因此提高了可用性

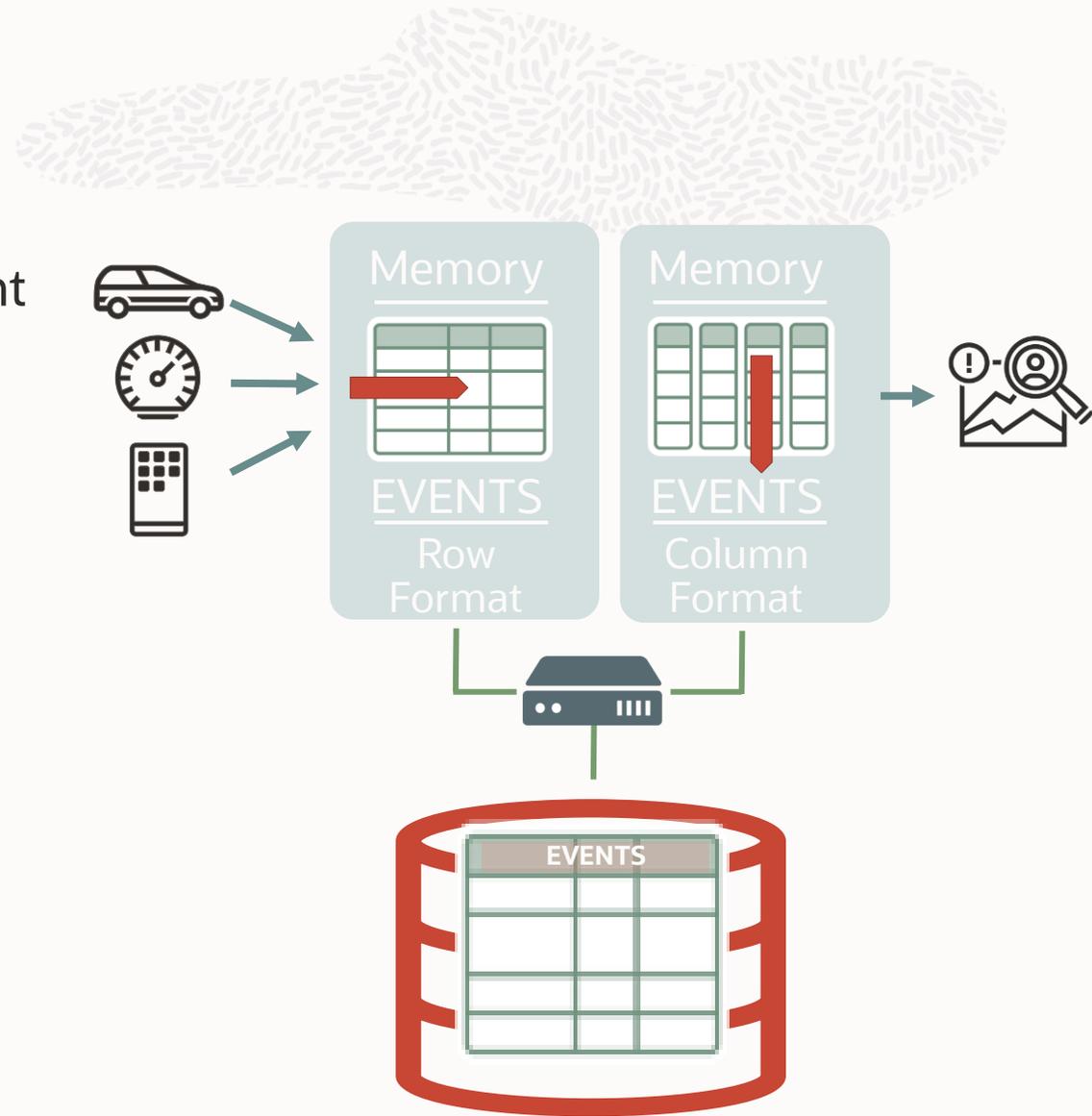


实时分析

Database In-Memory

Real-Time Analytics with Fast Ingest

- 行格式是Event Ingest的理想格式，列格式是Event Analytics的理想格式
- 同一个表可以同时启用行和列格式
- 同时活跃且始终如一
- Event Ingest使用高度优化的行格式
 - 每秒摄入数千万个事件
- 事件实时分析使用内存列格式
 - SIMD矢量指令允许每条指令处理多个值
 - 每秒分析数十亿个事件



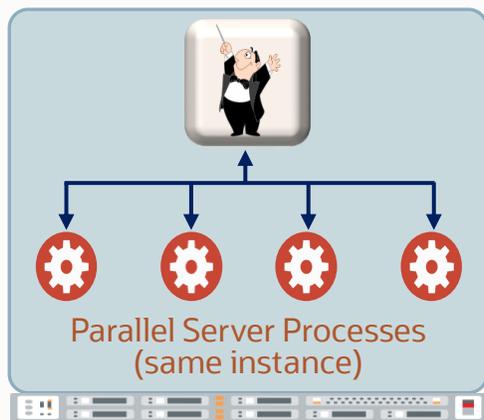
Parallel Execution 并行执行

Scale-Up / Scale-Out Query Processing

- 并行执行通过利用所有可用的CPU核心，最大限度地提高整个事件流查询
 - 协调器进程分配并行执行语句的后台服务器进程
 - 对应用程序完全透明

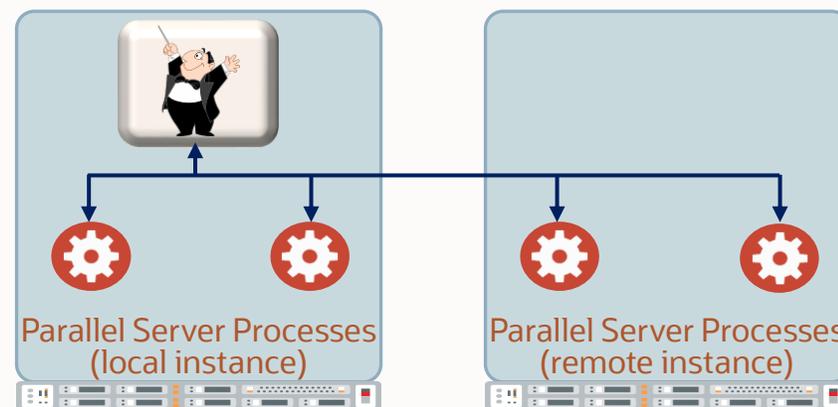
Scale-Up

并行服务器可以在同一实例中运行，以最大限度地提高主机CPU利用率



Scale-Out

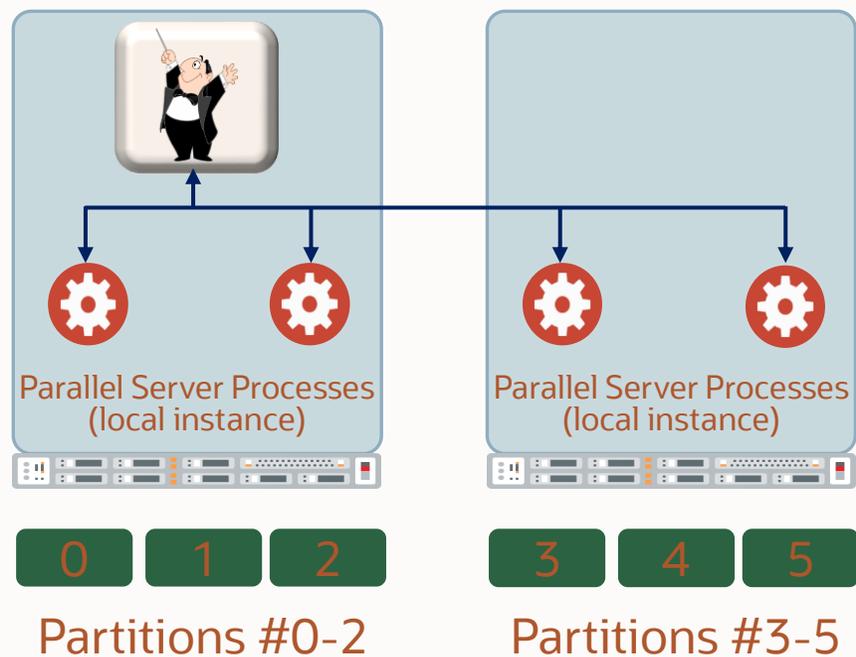
并行服务器还可以跨多个RAC实例运行，以跨多个主机利用CPU



Parallel Execution + Partition

结合无共享和共享存储体系结构的优点

- 并行执行可以与分区相结合，以便不同实例上的服务器在独立的分区集上操作



丰富的流分析功能

事件流分析SQL增强

分析窗口函数

- 将大型表分成多个分区增加可扩展的摄取
- Oracle 数据库拥有业界领先的事件流处理分析功能组合：
 - 行级函数：这些是标准 SQL 函数，为每行输入返回单个值（例如 ROUND、TRUNC、UPPER 等），可用于插值、平滑等。
 - 聚合函数：返回一组行的单个值（例如 MAX、MIN、AVG、SUM 等）
- 窗口函数：每行返回一个值，具体取决于该行所属的行组（由窗口子句指定）
 - 窗口函数对于分析不同时间段的事件特别有用1小时间隔内最大能耗
 - 对一天内的能源消耗区间进行排名
 - 与之前间隔相比消耗量的最大变化

```
SELECT MAX(energy)
      (OVER PARTITION BY time_in_hrs)
FROM MeterReadings;
```

	MeterID	Time	KWhrs	
Window 2 7-8pm	1XC23	8:00pm	0.4	➔ 1.5
	1XC23	7:45pm	0.5	
	1XC23	7:30pm	0.8	
	1XC23	7:15pm	1.5	
Window 1 6-7pm	1XC23	7:00pm	0.7	➔ 0.9
	1XC23	6:45pm	0.6	
	1XC23	6:30pm	0.9	
	1XC23	6:15pm	0.45	
Window 0 5-6pm	1XC23	6:00pm	0.86	➔ 1.3
	1XC23	5:45pm	1.3	
	1XC23	5:30pm	0.55	
	1XC23	5:15pm	1.02	

事件流分析SQL增强

模式匹配

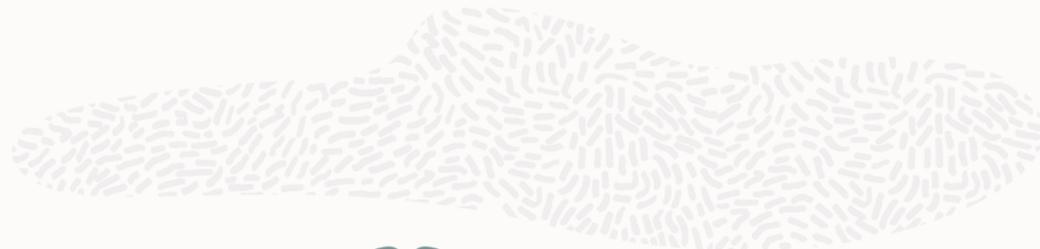
- 可以使用用于SQL模式匹配的MATCH_RECOGNIZE结构来进一步分析事件流数据
- MATCH_RECOGNIZE返回结果集中与结果集指定顺序内的指定模式匹配的行
- 许多用例——检测欺诈、高使用率警报、发现物联网指标异常等：
 - 查找与两个连续增加的读数相对应的时间段 (如图所示)
 - 检测特定股票的二次下跌
 - 检测可疑的信用卡收费模式
- 无需编写具有自联接和嵌套子查询的复杂SQL

```
SELECT * From MeterReadings
MATCH_RECOGNIZE(
  ORDER BY Time
  PATTERN(r r)
  ONE ROW PER MATCH
  DEFINE r as KWHrs > PREV(KWHrs))
```

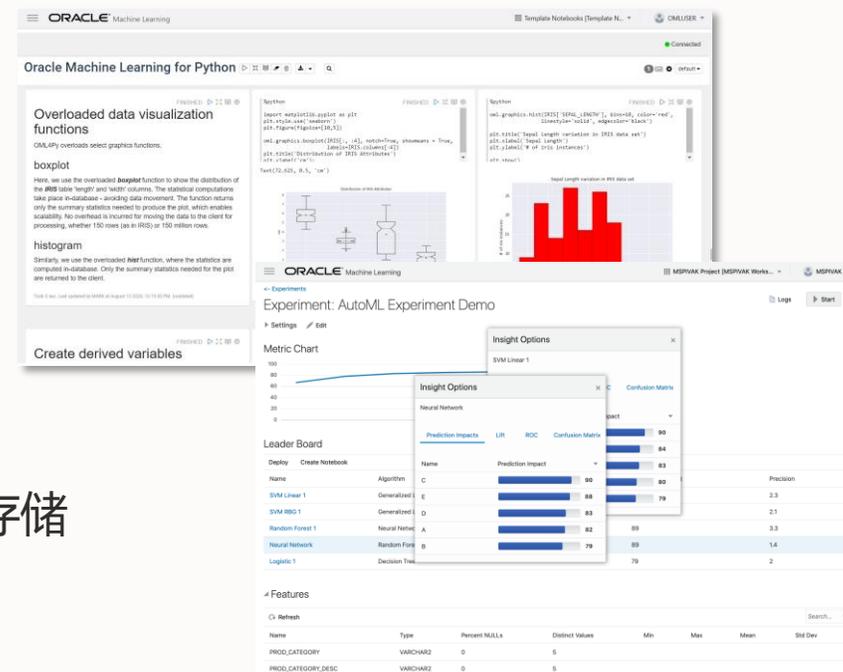
MeterID	Time	KWHrs	
1XC23	8:45pm	0.86	MATCH
1XC23	8:30pm	0.56	
1XC23	8:15pm	0.23	
1XC23	8:00pm	0.4	
1XC23	7:45pm	0.5	
1XC23	7:30pm	0.8	
1XC23	7:15pm	1.5	MATCH
1XC23	7:00pm	0.7	
1XC23	6:45pm	0.6	
1XC23	6:30pm	0.9	



针对事件流的机器学习



- 机器学习在事件流中的许多应用：
 - 故障预测、欺诈检测、情绪分析等
- Oracle数据库拥有非常丰富的用于事件流的机器学习模型组合
- Oracle自动ML
 - 为给定的数据集选择理想的算法
 - 为模型选择正确的数据样本
 - 识别数据中最大限度降低信噪比的关键功能
 - 推理是实时完成的，不需要任何进一步的数据移动到不同的存储

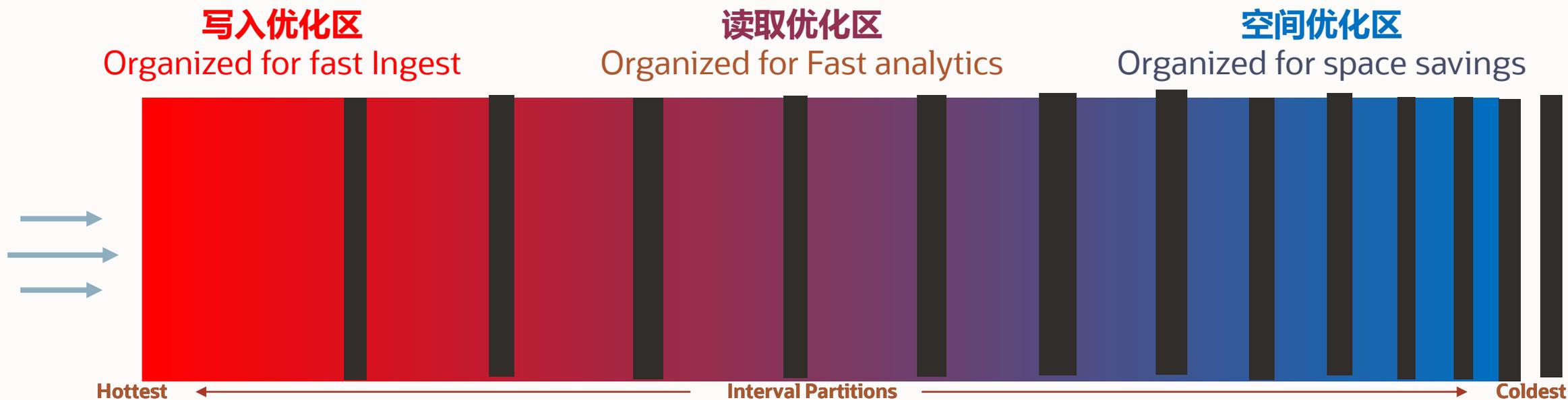


自动化数据管理

基于事件流数据的特点

从热到冷，渐进式压缩

- 事件流具有高数据到达率和相关衰减性
 - 架构设计：需要有按这些特性组织数据的考量
- 高效的事件流数据管理通常涉及三个区的概念设计：



- 横跨多个区进行数据分析的需求：对热数据和长期的冷数据的全量分析



Oracle 自动数据优化和数据库调度



写入优化区

Organized for fast Ingest

读取优化区

Organized for Fast analytics

空间优化区

Organized for space savings



- 使用 MEMOPTIMIZED FOR WRITE 声明分区表
 - 最近数据通常是未压缩的状态，以最快的速度完成数据录入

- 读取优化分区应声明为INMEMORY，以便启用实时分析
 - 请注意Exadata，不适合内存的分区可以在存储上利用内存

- 空间优化分区应使用混合列压缩进行压缩
 - 压缩较新的分区进行查询，而压缩较旧的分区进行存档
 - 旧数据进行采样或汇总，以进一步减少空间

这种写入、读取和空间优化的梯度是通过自动数据优化和DBMS_SCHEDULER实现的



Oracle 自动数据优化 (Automatic Data Optimization ADO) 和数据库调度



写入优化区

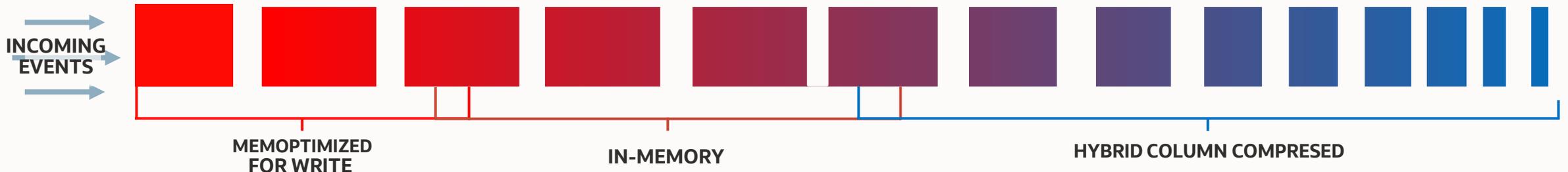
Organized for fast Ingest

读取优化区

Organized for Fast analytics

空间优化区

Organized for space savings



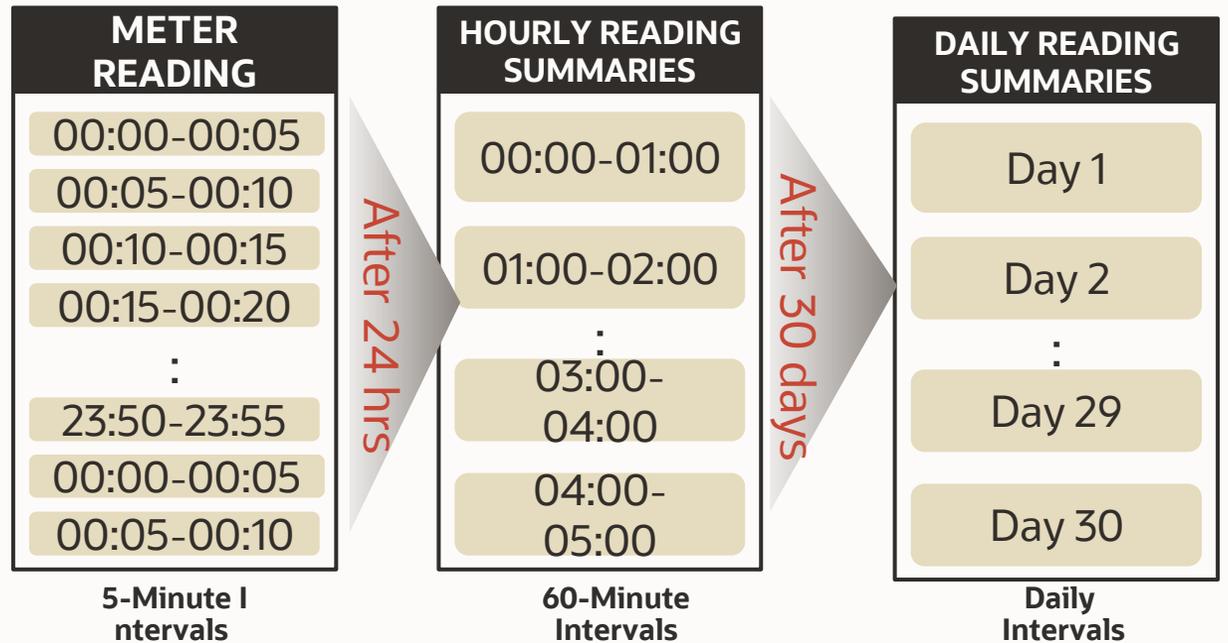
- 自动数据优化 (ADO)：基于用户策略的数据后台重组
 - 将热数据放入内存中的列存储中或从中删除数据，以自动化加速查询
 - 使较冷的数据被压缩，并根据访问的最近程度提高压缩级别
 - 较冷的数据移动到成本较低的存储层
- 数据库调度器：作业调度机制，可用于在指定的时间间隔执行数据的逻辑转换



事件流数据的调度管理

- 可以对较旧的事件进行下采样（在较长的时间段内汇总），尤其是需测量或度量的事件
- 汇总可以通过数据库调度程序作为任务自动运行
- 汇总后，可以删除源数据（通常通过删除分区）

- 智能计量应用程序可能每五分钟接收一次仪表读数
- DBMS_SCHEDULER作业根据超过24小时的时间间隔生成每小时摘要，并将其插入摘要表
- 汇总表可用于提供其他更不精细的汇总（例如，在一个月后生成每日汇总）

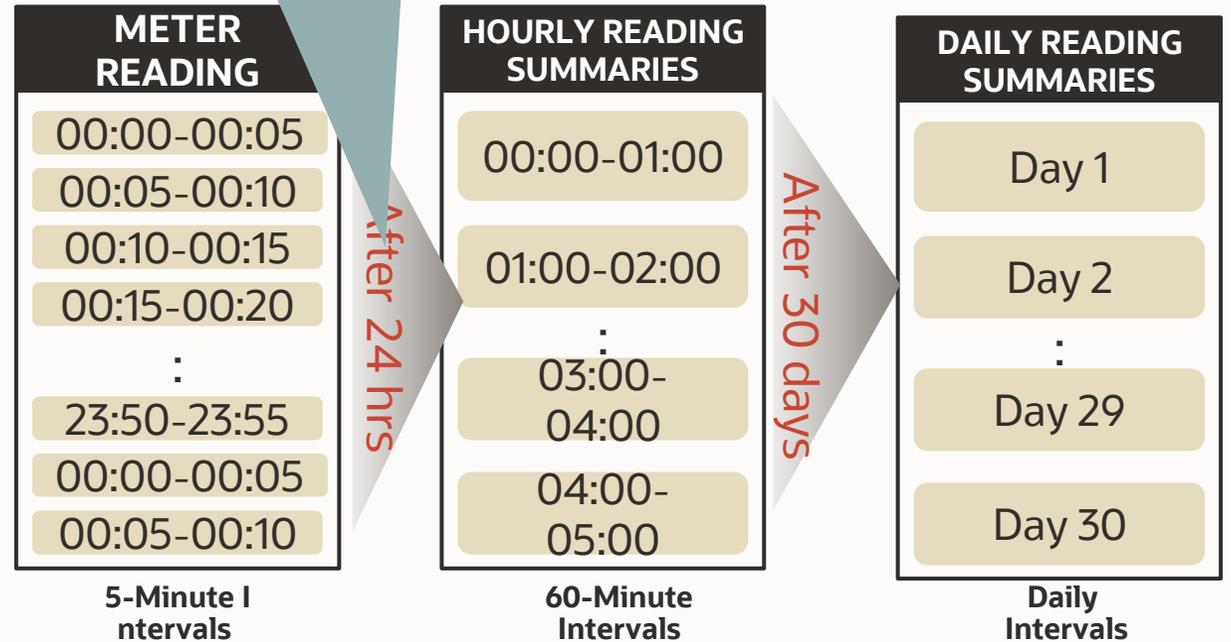


事件流数据的调度管理

- 可以对较旧的事件进行下采样（在较长的时间段内汇总事件）
- 汇总可以通过数据库调度程序作为任务自动运行
- 汇总后，可以删除源数据（通常通过删除分区）

```
INSERT INTO HourlySummaries  
SELECT meter_id,  
       time_in_hrs, sum(KWhrs)  
FROM   MeterReadings  
WHERE  <more than 24hrs old>  
GROUP BY meter_id, time_in_hrs
```

- 智能计量应用程序可能每五分钟接收一次仪表读数
- DBMS_SCHEDULER作业根据超过24小时的时间间隔生成每小时摘要，并将其插入摘要表
- 汇总表可用于提供其他更不精细的汇总（例如，在一个月后生成每日汇总）

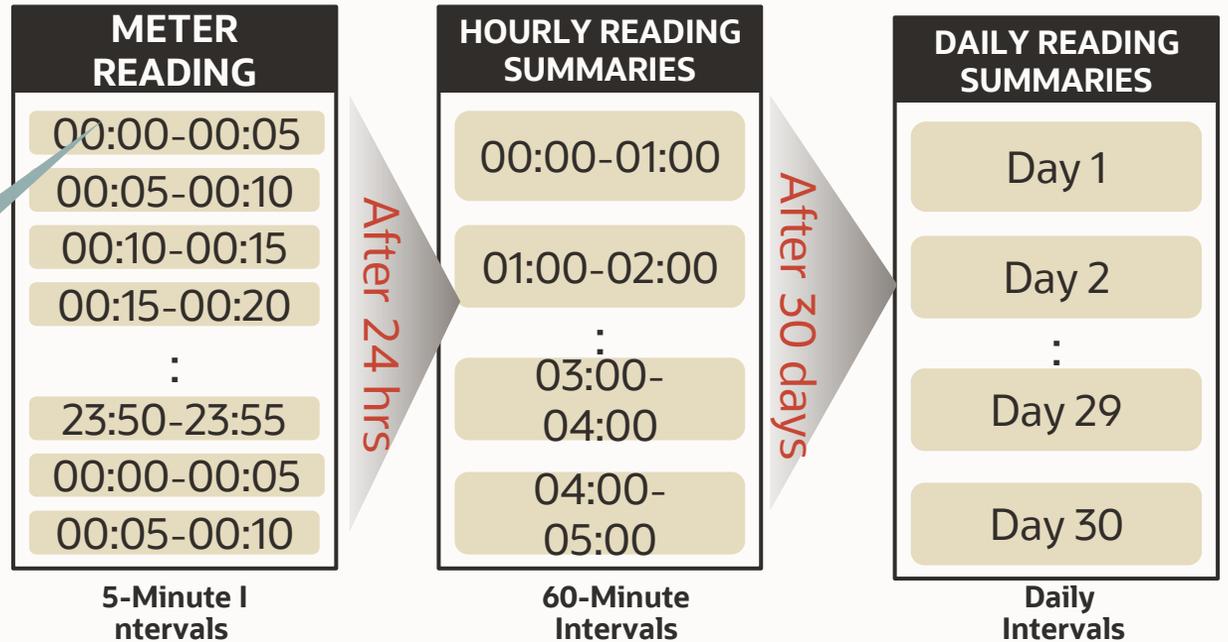


事件流数据的调度管理

- 可以对较旧的事件进行下采样（在较长的时间段内汇总），尤其是需测量或度量的事件
- 汇总可以通过数据库调度程序作为任务自动运行
- 汇总后，可以删除源数据（通常通过删除分区）

- 智能计量应用程序可能每五分钟接收一次仪表读数
- DBMS_SCHEDULER作业根据超过24小时的时间间隔生成每小时摘要，并将其插入摘要表
- 汇总表可用于提供其他更不详细的摘要。例如，在一个月后生成每日汇总

Drop Partition After 24hrs



Oracle 时序数据周期管理策略 | Summary

Component Mechanism

数据写入: MEMOPTIMIZE FOR WRITE

通过缓冲要插入内存的行，优化最近的热点分区以进行高速事件摄取

数据分析: INMEMORY (on DRAM and FLASH)

为热数据创建内存列格式，为中间数据创建闪存列格式支持为查询性能或节省内存而优化的不同级别的内存压缩

数据压缩: HYBRID COLUMN COMPRESSION

将行数据透视到压缩的列单位中
压缩级别用于最大限度地提高查询性能或最大限度地节省空间

数据计算: DOWNSAMPLING

通过汇总基本事件数据，大大降低了存储成本并提高了分析性能

Automation

ADO MEMOPTIMIZE TIERING

数据库自动为遇到高频事件摄取的分区启用内存优化

ADO IN-MEMORY TIERING

数据库根据分析访问频率自动启用内存中的最佳 MEMCOMPRESS 设置

ADO COMPRESSION TIERING

将行数据透视到压缩的列单位中
压缩级别用于最大限度地提高查询性能或最大限度地节省空间

DOWNSAMPLING VIA DBMS_SCHEDULER

通过汇总基本事件数据，大大降低了存储成本并提高了分析性能

相关案例

Oracle 数据库管理的大规模事件流负载场景



电信

移动运营商业务

- 650亿次事件/每天
- 18TB的数据/每天
- 所有数据处理都发生在Exadata上运行的Oracle RAC数据库上

交通

交通信号管理

- 约30亿次事件/天
- 约320亿次查询/天
- 数据库超过1PB
- 部署在Exadata上的Oracle RAC数据库上

制造

制造设备传感器

- 约10亿次活动/天
- 180000条消息/秒
- 约15 TB的数据/天
- 在Exadata上的Oracle RAC数据库中捕获和处理的所有数据



ORACLE
甲骨文

Oracle现代数据分析平台

甲骨文云与数据库公益讲座



陆汉明

- Oracle数据分析专家
- 16年Oracle数据分析产品售前和售后支持经验
- 拥有多个大型数据仓库和分析项目的实施和开发经验

内容简介

OAS/OAC现代数据分析平台新功能介绍

演示：如何定义语义模型？

演示：如何敏捷开发可视化分析看板？

演示：如何使用一键解释等高级分析功能？



Zoom直播

直播时间：2月23日 11:00 - 12:00

扫描二维码进入直播

Zoom ID: 957 9669 6723

密码：20212023



微信扫一扫预约



数据库和云讲座群

20-23



甲骨文云技术公众号



技术专家1V1深入交流

