集算器

创新大数据计算引擎

生产制造业库龄计算案例

润乾软件出品

项目背景

企业为了提高供应链的整体效率，通常都会把库龄计算作为整体经营业绩的考核指标之一。通过全局范围的库龄分析，及时了解热销商品和滞销商品及其分布情况，合理地进行库存调度和市场促销，可以大大提高库存周转率，促进销售的同时降低库存资金积压风险！

库龄由库存量去反推，找到按日期从大到小排列入库记录，直到入库记录相加大于等于库存量！

现有量表1

时间	组织	子库	物料	库存量
20160930	A	A	A	1000
…	…	…	…	…

关联字段

事务处理表

时间	组织	子库	物料	入库量
20160927	A	A	A	200
20160917	A	A	A	200
20160907	A	A	A	200
…	…	…	…	…

现有量表2

时间	组织	子库	物料	库存量
20160930	A	A	A	300
…	…	…	…	…

库龄计算情况一

所有入库量(600)还小于库存量(1000)，需要补齐记录:库存量(1000)-所有入库量和(600)，库龄为"60天以上"

库龄计算情况二

当入库量累计(到20160917已经为400)大于库存量(300)时，最后一条记录的库存量为100

库龄数据1

时间	组织	子库	物料	入库量	库龄(天)
20160927	A	A	A	200	3
20160917	A	A	A	200	13
20160907	A	A	A	200	23
20160701	A	A	A	400	60天以上
…	…	…	…	…	…

库龄数据2

时间	组织	子库	物料	入库量	库龄(天)
20160927	A	A	A	200	3
20160917	A	A	A	100	13
…	…	…	…	…	…

原计算方案的不足

原测试方案是构建Hive数据仓库，从源数据Oracle定时同步到Hive，所有查询都基于Hive数据仓库，从而获取想要的库龄分析指标！

源数据量大

源数据存储在Oracle中，库存量与事务处理等相关表的数据总量在3亿

SQL算法长

单个库龄计算场景，写了4段Hive sql才实现，共计140行

执行效率慢

单个库龄计算场景，4部分sql执行一次计算的总时间为9分钟

原SQL和执行效率

1 获取现有量SQL

```
1  INSERT OVERWRITE TABLE hive_bak.MC_TEMP_DM_FIN_AR_GERP_MOQ
2  SELECT f.manage_org_wid,
3         f.direct_wid,
4         NULL  AS sales_region_wid,
5         NULL AS sales_cen_wid,
6         NULL AS customer_wid,
7         f.organization_id,
8         f.subinventory_code,
9         f.item_code,
10        f.item_wid,
11        sub.icp,
12        sub.erp_segment3,
13        NVL(locator_id,'0') AS locator_id,
14        SUM(quantity) AS qty
15    FROM dw.dw_inv_onhand_quantity_f f
16    LEFT JOIN DIM.dim_subinventory_d SUB
17      ON f.subinventory_CODE = SUB.subinventory_CODE
18     AND f.organization_id = CAST(SUB.organization_id AS STRING)
19   WHERE f.SOURCE_CODE = 'GERP'
20     AND f.part_dt = '2016-09-30'
21     AND f.organization_id='201'
22   GROUP BY f.manage_org_wid,
23            f. direct_wid,
24            f.organization_id,
25            f.subinventory_code,
26            f.item_code,
27            sub.icp,
28            sub.erp_segment3,
29            f.item_wid,NVL(locator_id,'0')
30  HAVING SUM(quantity) <> 0;
```

1分0.5秒

2 事务处理表SQL

```
1  INSERT OVERWRITE TABLE hive_bak.MC_TEMP_DM_FIN_AR_GERP_MMT
2  SELECT organization_id,
3         subinventory_code,
4         item_code,
5         transaction_date,
6         locator_id,
7         qty,
8         SUM(qty) over(PARTITION BY organization_id, subinventory_code, locator_id,item_code
9         ORDER BY transaction_date DESC) AS sum_qty,
10        ROW_NUMBER() OVER(PARTITION BY organization_id, subinventory_code,locator_id, item_code
11        ORDER BY transaction_date DESC) RN
12    FROM (SELECT MMT.organization_id,
13                 MMT.subinventory_code,
14                 MMT.item_code,
15                 SUBSTRING(MMT.transaction_date, 1, 10) AS transaction_date,
16                 NVL(LOCATION_ID,'0') AS locator_id,
17                 SUM(MMT.quantity) AS QTY
18            FROM DW.dw_mtl_transactions_f MMT
19           INNER JOIN DIM.dim_subinventory_d SUB
20              ON MMT.subinventory_CODE = SUB.subinventory_CODE
21             AND MMT.organization_id = CAST(SUB.organization_id AS STRING)
22           WHERE part_sc = 'GERP'
23             AND SUBSTRING(mmt.transaction_date, 1, 10) <= '2016-09-30'
24             AND mmt.organization_id='201'
25          --    and sub.erp_segment2 in ('1407010101', '1406010101')
26          --    and sub.served_domain = '工厂'
27             AND quantity > 0
28           GROUP BY MMT.organization_id,
29                    MMT.subinventory_code,
30                    MMT.item_code,
31                    SUBSTRING(MMT. transaction_date, 1, 10),NVL(LOCATION_ID,'0')
32          HAVING SUM(MMT.quantity) > 0
33          UNION ALL
34          SELECT moq.organization_id,
35                 moq.subinventory_code,
36                 moq.item_code,
37                 DATE_ADD('2016-09-30',-60)    AS transaction_date,
38                 NVL(moq.locator_id,'0') AS locator_id,
39                 9999999999999999 AS QTY
40        FROM
41           hive_bak.MC_TEMP_DM_FIN_AR_GERP_MOQ moq ) A
```

6分14秒

原SQL和执行效率

3 取对应的事务处理,找到事务处理记录(第一部分)

1分6.7秒

```
INSERT OVERWRITE TABLE hive_bak.MC_TEMP_DM_FIN_AR_GERP_MMT_SORT
SELECT
  MMT.organization_id,
       MMT.subinventory_code,
       MMT.item_code,
       CASE WHEN a.qty <0 THEN '2016-09-30' ELSE MMT.transaction_date END transaction_date,
       MMT.locator_id,
       MMT.qty,
       MMT.sum_qty,
       MMT.RN,
       A.RN AS MAX_RN,
       CASE WHEN ROUND(A.QTY - MMT.SUM_QTY,5) = 0 THEN 'Y' ELSE 'N' END   AS FULL_FLAG
  FROM hive_bak.MC_TEMP_DM_FIN_AR_GERP_MMT MMT,
       (SELECT MMT.organization_id,
               MMT.subinventory_code,
               MMT.locator_id,
               MMT.item_code,
               MIN(MMT.RN) AS RN,
               moq.qty AS qty
          FROM hive_bak.MC_TEMP_DM_FIN_AR_GERP_MOQ moq
         INNER JOIN hive_bak.MC_TEMP_DM_FIN_AR_GERP_MMT mmt
            ON mmt.organization_id = moq.organization_id
          AND mmt.subinventory_code = moq.subinventory_code
          AND MMT.locator_id =MOQ.locator_id
          AND mmt.item_code = moq.item_code
         WHERE ROUND(MOQ.QTY - MMT.SUM_QTY,5) <= 0
         GROUP BY MMT.organization_id, MMT.subinventory_code,MMT.locator_id, MMT.item_code,M
        ) A
 WHERE mmt.rn <= A.rn
   AND mmt.organization_id = A.organization_id
   AND mmt.subinventory_code = A.subinventory_code
   AND mmt.item_code = A.item_code
   AND MMT.locator_id =A.locator_id
```

4 取对应的事务处理,找到事务处理记录(第二部分)

36.2秒

```
INSERT OVERWRITE TABLE hive_bak.MC_TEMP_DM_FIN_AR_SALES_FACTORY
 SELECT
 moq. manage_org_wid,
 icp_org.manage_org_wid AS manage_org_icp_wid,
 CASE
   WHEN icp_org.bd_code = org.bd_code THEN
    1
   WHEN icp_org.bd_code IS NOT NULL AND icp_org.bd_code <> org.bd_code AND
        moq.icp <> '0' THEN
    2
   ELSE
    3
 END AS TRANSACTION_CLASS_WID,
 moq.direct_wid,
 moq.sales_region_wid,
 moq.sales_cen_wid,
 moq.customer_wid,
 moq.organization_id,
 moq.subinventory_code,
 moq.item_code,
 moq.item_wid,
 moq.erp_segment3,
 CASE WHEN FULL_FLAG='N' AND  MMT.RN=MMT.MAX_RN THEN MMT.QTY+MOQ.QTY-MMT.SUM_QTY ELSE MMT.QTY END  AS qty,
 mmt.transaction_date,
 MMT.locator_id
  FROM hive_bak.MC_TEMP_DM_FIN_AR_GERP_MOQ moq
  LEFT JOIN  hive_bak.MC_TEMP_DM_FIN_AR_GERP_MMT_SORT   MMT
    ON mmt.organization_id = moq.organization_id
   AND mmt.subinventory_code = moq.subinventory_code
   AND MMT.locator_id=MOQ.locator_id
   AND mmt.item_code = moq.item_code
  LEFT JOIN dim.dim_manage_org_d org
    ON moq.manage_org_wid = org.manage_org_wid
  LEFT JOIN dim.dim_manage_org_d icp_org
    ON moq.icp = icp_org.org_code
```

单个完整的库龄计算场景，共计139行SQL代码，查询了8分57.4秒！

引入数据计算层

不变动基础层架构，在应用层集成数据计算引擎中间件，它拥有不依赖于数据库的计算能力，使用一致的结构化计算模型，为前端应用提供统一计算服务！

轻量级数据仓库

独立计算引擎使文件系统拥有了计算能力，可直接作为数据仓库使用

开发更简单

完备的开发调试功能，伪代码风格，容易理解和排错，让数据计算更有效率

指标响应更快

高性能的私有文件存储格式（集文件、组表），可获得比传统数仓更高IO性能

降低维护成本

解释执行脚本，当需求更变时，修改计算逻辑不会影响其他代码，易于迭代升级

集算器的实现思路

2、打开IDE
3、编写脚本

	A	B	C	D	E	F
1	=now()	=date="2016-09-30"				
2	=file("D:\\TEST\\现有量.b").cursor@bt(organization_id,subinventory_code,item_code,qty).select(qty>0)					
3	>A2.run(item_code=string(item_code,"#"))					
4	=A2.groups(organization_id,subinventory_code,item_code,sum(qty):quantity5)					
5	=file("D:\\TEST\\事物处理.b").cursor@bt(organization_id,subinventory_code,item_code,transaction_date,qty).select(qty>0).fetch()					
6	=A5.group@u(organization_id,subinventory_code,item_code)					
7	=A4.join@(organization_id:subinventory_code:item_code,A6:organization_id:subinventory_code:item_code,~:sub)					
8	>A7.run(sub=sub.groups@u(organization_id,subinventory_code,item_code,transaction_date,sum(qty):quantity,sum(null):cum))					
9	>A7.run(sub=sub.sort@z(transaction_date))					
10	>A7.run(sub.run(cum=cum[-1]+quantity))	/组内累计事务处理表的数量				
11						
12	=create(组织,子库,物料,时间,库存,库龄)	/建立结果表				
13	for A7	if A13.sub !=	=A13.sub.pselect(cum>=A13.quantity5)			
14			if C13 >0	>A13.sub=A13.sub.to(C13)	>A13.sub(C13).quantity=A13.quantity5-A13.sub.m(C13-1).cum	>A12.insert(0:A13.sub,organization_id,subinventory_code,item_code,A13.sub.transaction_date,floor(quantity,1)
15			else	>A12.insert(0:A13.sub,organization_id,subinventory_code,item_code,A13.sub.transaction_date,quantity,interval(A13.sub.transaction_date,date))	>A12.insert(0,A13.organization_id,A13.subinventory_code,A13.item_code,after(A13.sub.transaction_date,-90),A13.quantity5-A13.sub.m(-1).quantity,"90天以上")	
16		else	>A12.insert(0,A13.organization_id,A13.subinventory_code,A13.item_code,after(date,-90),A13.quantity5,"90天以			
17	return A12					

17行

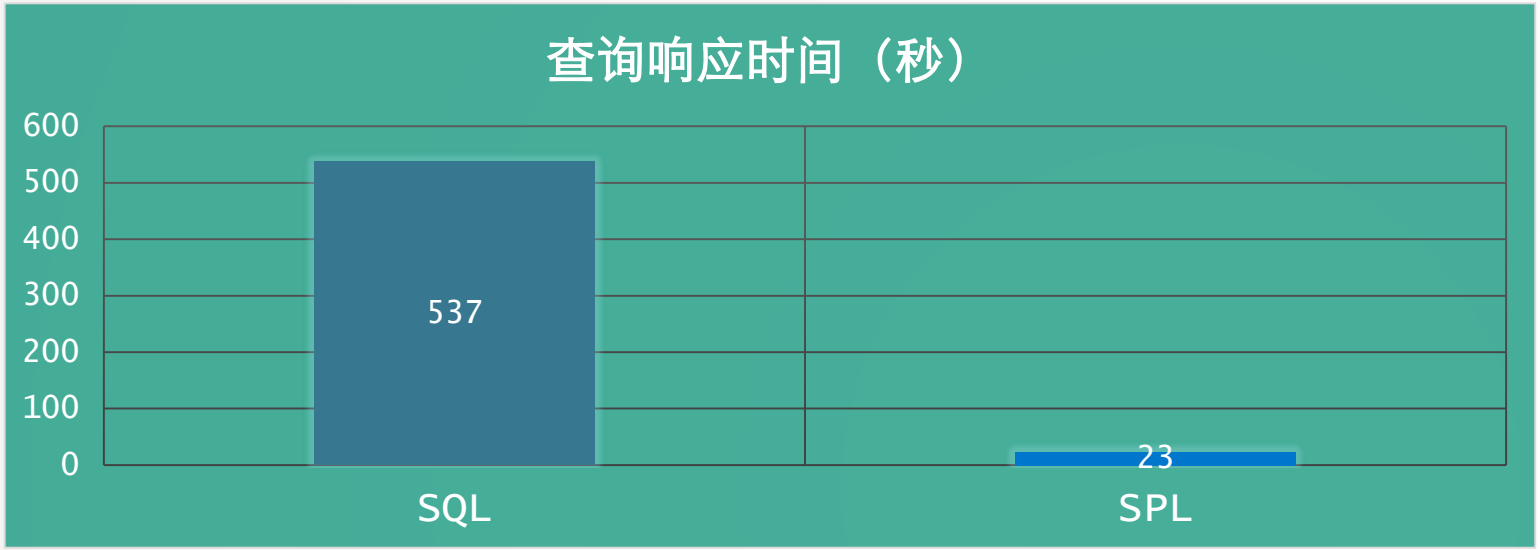
1、按照自然思维该怎么写?

```
1  造目标表
2  for 事务处理.sort(时间:-1).group(组织,子库,物料)
3        找出现有量(可先建索引表)
4              for 分组成员
5                    往目标表里插入记录；现有量减法，减到0 break
6              现有量还不是0，再插入一条60天以上的…
```

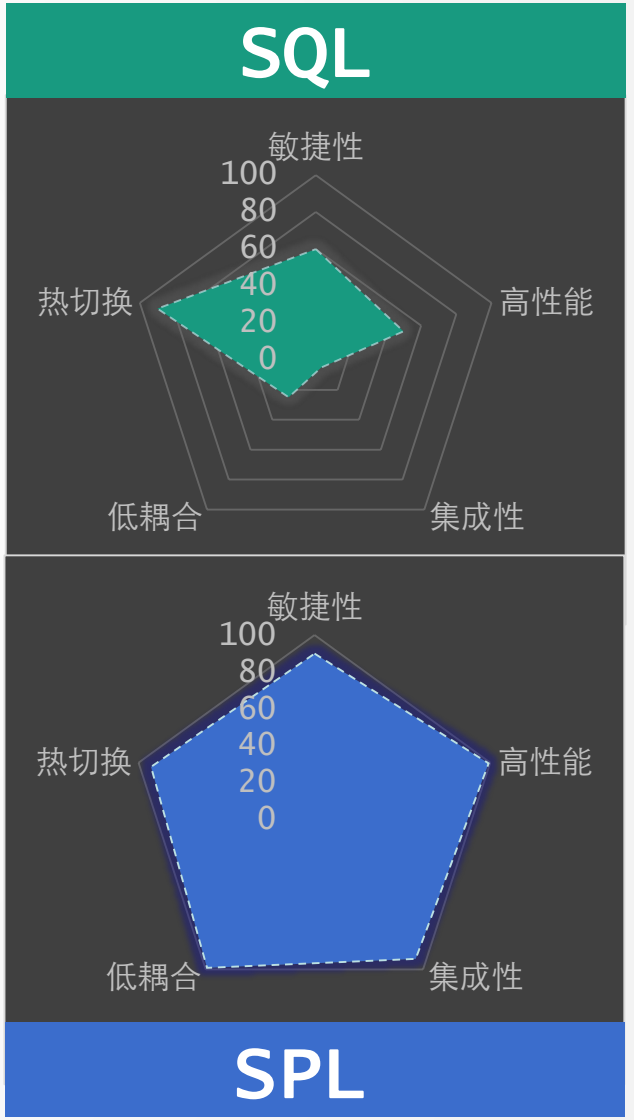
4、生成库龄数据

时间	组织	子库	物料	入库量	库龄(天)
20160927	A	A	A	200	3
20160917	A	A	A	200	13
20160907	A	A	A	200	23
20160701	A	A	A	400	60天以上
…	…	…	…	…	…

实测：集算器开发效率在各方面的工作量评估



对比指标	SQL	SPL	变化
代码行	139	17	代码量减至10%
开发周期	15人天	2人天	工作量减至13%
执行性能	8分57秒	23秒	性能提升20倍以上
应用耦合	高	低	模块化、工具化
可维护性	较差	好	易管理、热切换



为什么集算器能有如此效率

高性能计算

私有的数据存储格式，让文件拥有比传统数据仓库更高的IO性能

敏捷语法体系

语法简洁易学，更接近人的自然思维，特别适合做复杂过程计算

维护成本低

IDE自带调试和排错功能，解释执行，算法更变无须重启应用

优化体系结构

数据计算从应用程序与数据库中分离出来，彻底工具化、独立化

创新技术
推动应用进步！


各设备故障分析
近7天设备的总故障率
近7天
近1天
近6小时
1.53%
0.95%
2.40%
近7天故障情况
全年故障综合分析
近7天故障次数前5的设备
服务器CPU使用率
今日
昨日
今天硬盘情况