



润乾

RAQSOFT



集算器

创新大数据计算引擎

临时性数据计算

润乾软件出品



临时性数据计算的广泛应用场景



应对业务部门的临时取数需求

数据挖掘前的数据准备

测试数据生成

大数据计算的优化方案实验

非日常数据清洗与入库



临时性数据计算的特征

需求随意，不可预测



多样性的原始外部数据



必要时可能转变成日常计算



经常只做一次，缺乏复用性



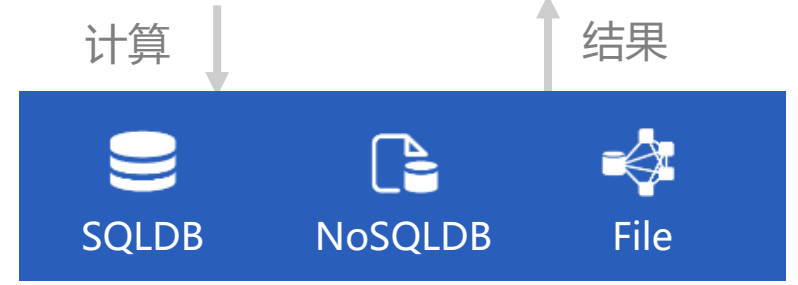
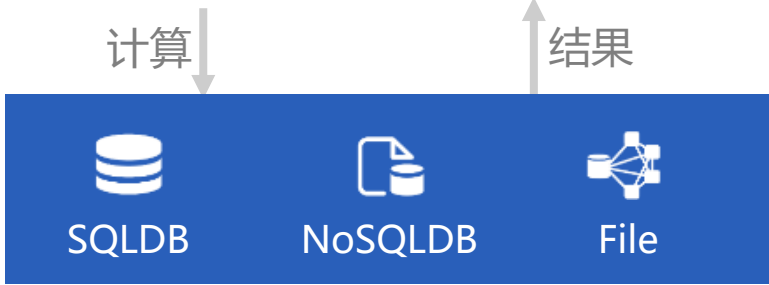
多步骤的过程计算



以（准）结构化数据计算为主



当前常用技术及问题



JAVA

- 计算类库少，代码繁琐
- 需要专业程序员，人员要求高
- 开发环境配置不简单
- 多种数据源计算开发实现困难

SQL

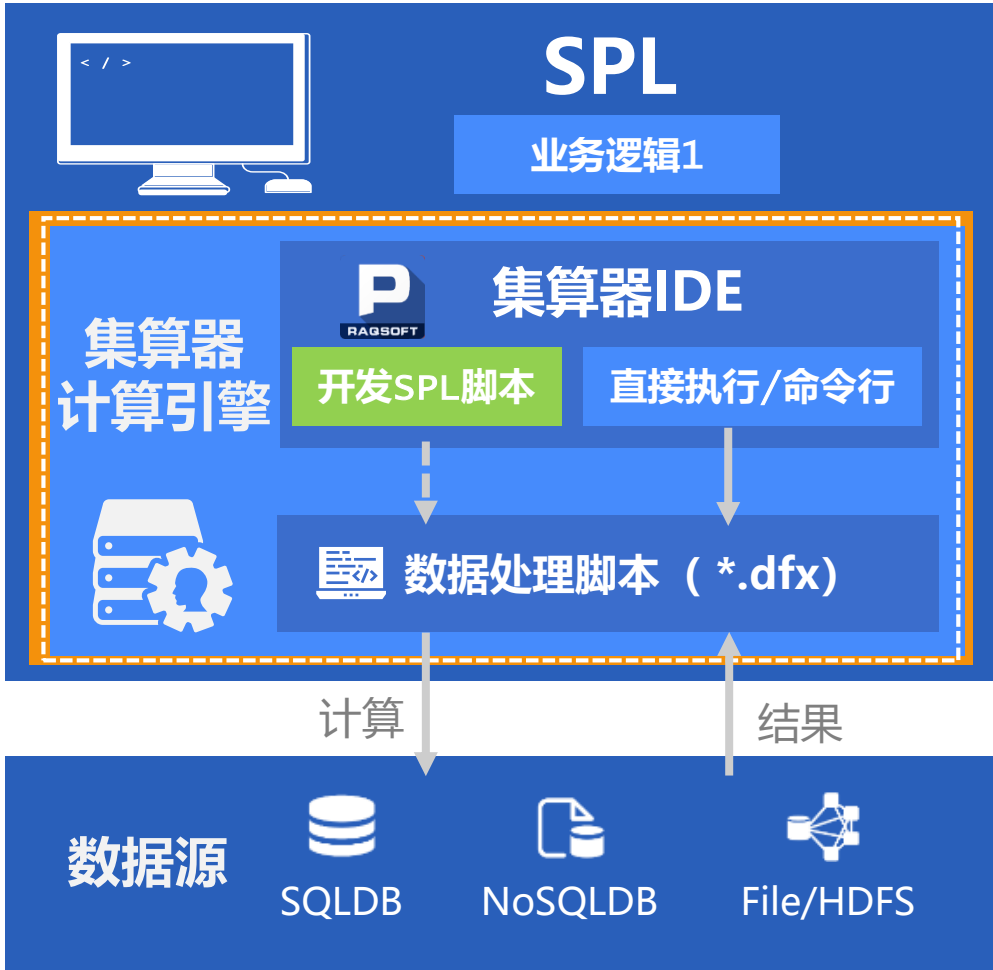
- 复杂 SQL和存储过程调试不便
- 多步运算不是自然思维，实现困难
- 计算封闭，无法计算库外数据，对无计算能力的数据源必须先入库

Python

- 语法不是专为结构化数据计算设计，复杂运算不易开发
- 版本之间存在不兼容问题
- 外围开源包安装配置麻烦
- 难集成，转成日常计算时可能重写



使用集算器实现临时计算



集算器即装即用，环境配置简单

执行、调试执行、单步执行 设置断点

网格结果所见即所得，易于调试；方便引用中间结果

db09.dfx

	A	B	C	D
1	=file("..\demo\zh\bt\Stock.txt").import@t().select(month(Datetime)==6)			
2	=file("..\demo\zh\bt\Sale.txt").import@t().select(month(Datetime)==6)			
3	=file("..\demo\zh\bt\Storage.txt").import@t().select(month(Date)==5)			
4	=file("..\demo\zh\bt\Commodity.txt").import@t()			
5	'08:00:00		'21:30:00	
6	=periods@d(date("2009-6-1"), date("2009-6-30"), 1)			
7	=A1.align@a(A6:~,date(Datetime))			
8	=A2.align@a(A6:~,date(Datetime))			

A2

Datetime	Commodity	Volume
2009-06-01 08:0	20077	28
2009-06-01 08:1	20056	47
2009-06-01 08:1	20094	34
2009-06-01 08:2	20020	19
2009-06-01 08:4	20013	42
2009-06-01 08:4	20077	1
2009-06-01 08:5	20069	19
2009-06-01 09:0	20011	22
2009-06-01 09:0	20007	22

SPL 分步计算的可视化开发环境，适合一般技术人员使用，比JAVA/SQL/Python更直观

系统信息输出，异常随时查看

系统信息输出
[2017-09-28 10:38:11]
DEBUG: Esproc Function Points = 1000 0001 1111 1101



集算器敏捷语法体系



某支股票最长连续涨了多少交易日

Python

```
1 import pandas as pd
2 def iterate(col):
3     prev = 0;
4     res = 0;
5     val = 0;
6     for curr in col:
7         if curr - prev > 0:
8             res += 1;
9         else:
10            res = 0;
11            prev = curr;
12            if val < res:
13                val = res;
14            return val;
15 data = pd.read_excel( 'D:/Stock.xlsx' ,sheet_name=0).
    sort_values( 'Date' ).groupby( 'Company' )[ 'Price' ].apply(iterate);
```

SPL

```
1 =file( "D:/Stock.xlsx" ).importxls@t().sort(Date).group(Company)
2 =A1.((a=0,~.max(a=if(Price>Price[-1],a+1,0))))
```

集算器

SPL语法更敏捷，更符合自然思维，易开发



集算器丰富的计算类库

专门针对（准）结构化数据设计，让开发更简单、高效

	A	B	C
1	=esProc.query("SELECT 订单ID AS 合同,订购日期 AS 日期")		/读取销售记录表
2	=A1.group(销售)		
3	=create(销售,今年销售额,去年销售额,客户数,大客户数)		
4	for A2	=A4(1).销售	
5		=A4.select(year(日期)==年份).sum(金额)	
6		=A4.select(year(日期)-年份-1).sum(金额)	
7		=A4.group(年份).sum(金额)	
8		=B7.count()	
9		=B7.count(==10000)	

SPL 分组、循环

	A	B	C
1	=esProc.query("select * from 员工表")		
2	=A1.select(性别=="男")		
3	=A1.select(出生日期>=date("1970-01-01"))		
4	=A2^A3		/交运算，统计晚于1970年出生的男员工
5	=A2&A3		/并运算，统计男员工或者晚于1970年出生的员工
6	=A2\A3		/运算，统计早于1970年出生的男员工
7	=A4.sum(工资)		
8	=A5.avg(年龄)		
9	=A6.sort(出生日期)		
10			/集合作为基本数据类型
11			

SPL 集合运算

	A	B	C
1	=file("交易记录.txt").import@t0		
2	=A1.sort(客户编码,交易日期)		
3	=A2.select(车辆型号=="捷达" 车辆型号=="迈腾").dup@t0		
4	=A3.derive(interval(交易日期[-1],交易日期):间隔)		
5	=A4.select(车辆型号[-1]=="捷达" && 车辆型号=="迈腾" && 客户编码==客户编码[-1])		
6	=A5.avg(间隔)		

SPL 排序、过滤

	A	B	C
1	=esProc.query("select * from 员工表")		
2	=A1.sort(入职日期)		
3	=A2.pmin(出生日期)		/出生最早的员工的记录序号
4	=A2(to(A3-1))		/直接用序号访问成员
5	=esProc.query("select * from 股价表 where 股票代码='000062'")		
6	=A5.sort(交易日期)		
7	=A6.pmax(收盘价)		/收盘价最高的那条记录的序号
8	=A6.calc(A7.收盘价/收盘价[-1]-1)		
9			/直接用序号访问成员
10			
11			

SPL 有序集合



集算器多样性数据源支持

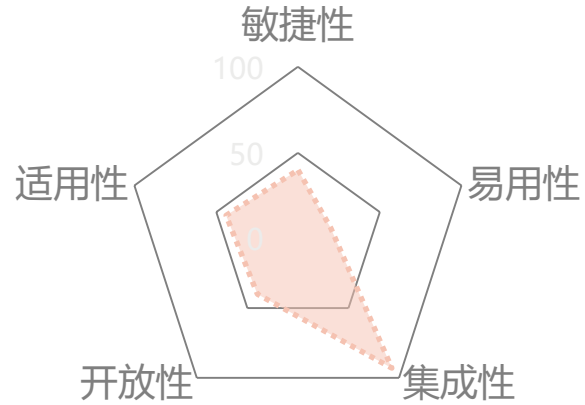
- RDB: Oracle, DB2, MS SQL, MySql, PG...
- TXT/CSV, JSON/XML, Excel, LOG...
- Hadoop: HDFS, Hive, HBase, Spark
- MongoDB, Redis
- HTTP、ALI-OTS
- Ftp、WebService、SAP
- 阿里云、Cassandra
- ES, Kafka
- 内置接口, 即装即用

通过集算器, 可以直接访问多样数据源进行临时性数据计算

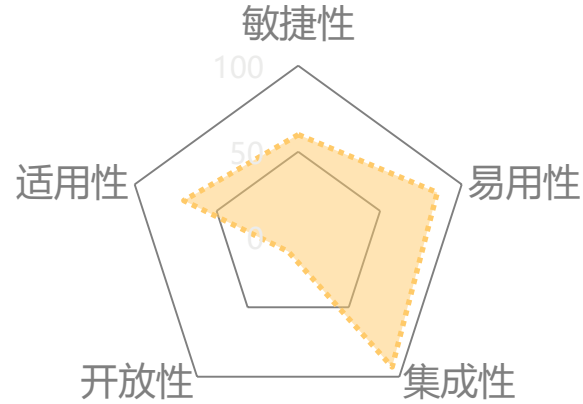


集算器的对比优势

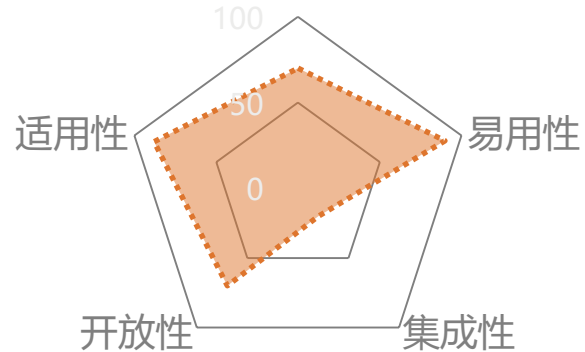
JAVA



SQL

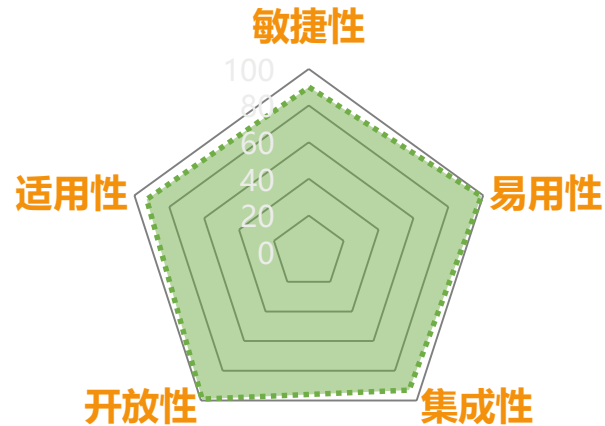


Python



Python

SPL



SPL

SPL技术优势特点

- + **敏捷性**-即装即用，环境配置简单，无须额外插件
- + **适用性**-完善的类库和过程计算，适应复杂数据处理
- + **易用性**-语法简单直观，调试方便，无须专业程序员
- + **开放性**-内置多种数据源访问接口，直接计算
- + **集成性**-无缝嵌入应用系统，易于转为日常计算

扫码关注

润乾软件 公众号



- ✓ 润乾软件最新动态
- ✓ 产品应用场景、案例
- ✓ 了解润乾，联系润乾

《数据蒋堂》 公众号



- ✓ 技术干货分享
- ✓ 每周一期
- ✓ 微信直播交流