



集算器

离线自由计算

润乾软件出品





离线计算的广泛应用场景



数据清洗与入库 (ETL)

报表与BI的数据准备

应对业务部门的临时查询需求

人工智能/数据挖掘算法前的数据整理

对手边数据进行自助分析

测试数据生成

大数据计算的优化方案实验

.....



离线自由计算的特征

需求随意，难以预测



多样性的原始外部数据



必要时可转变成日常应用中计算



经常只做一次，缺乏复用性



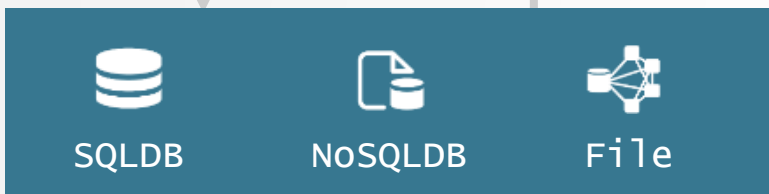
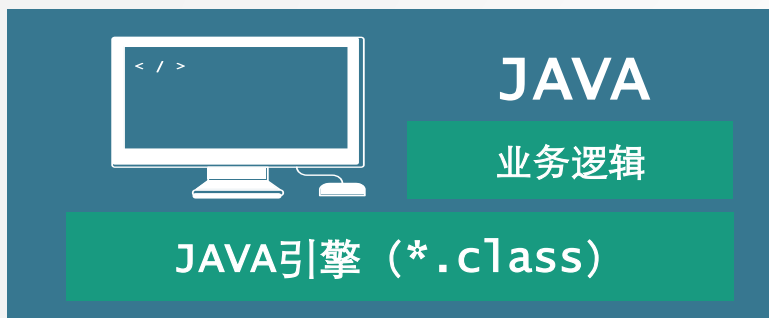
多步骤的过程计算



以（准）结构化数据计算为主



当前常用技术有什么问题？



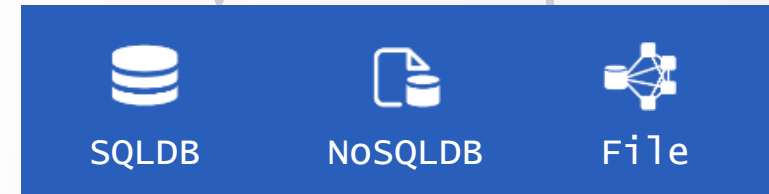
JAVA

- 计算类库少，代码繁琐
- 需要专业程序员，人员要求高
- 开发环境配置复杂
- 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
- 多步过程运算书写繁琐
- 计算封闭,无法计算库外数据,对无计算能力的数据库必须先入库



Python

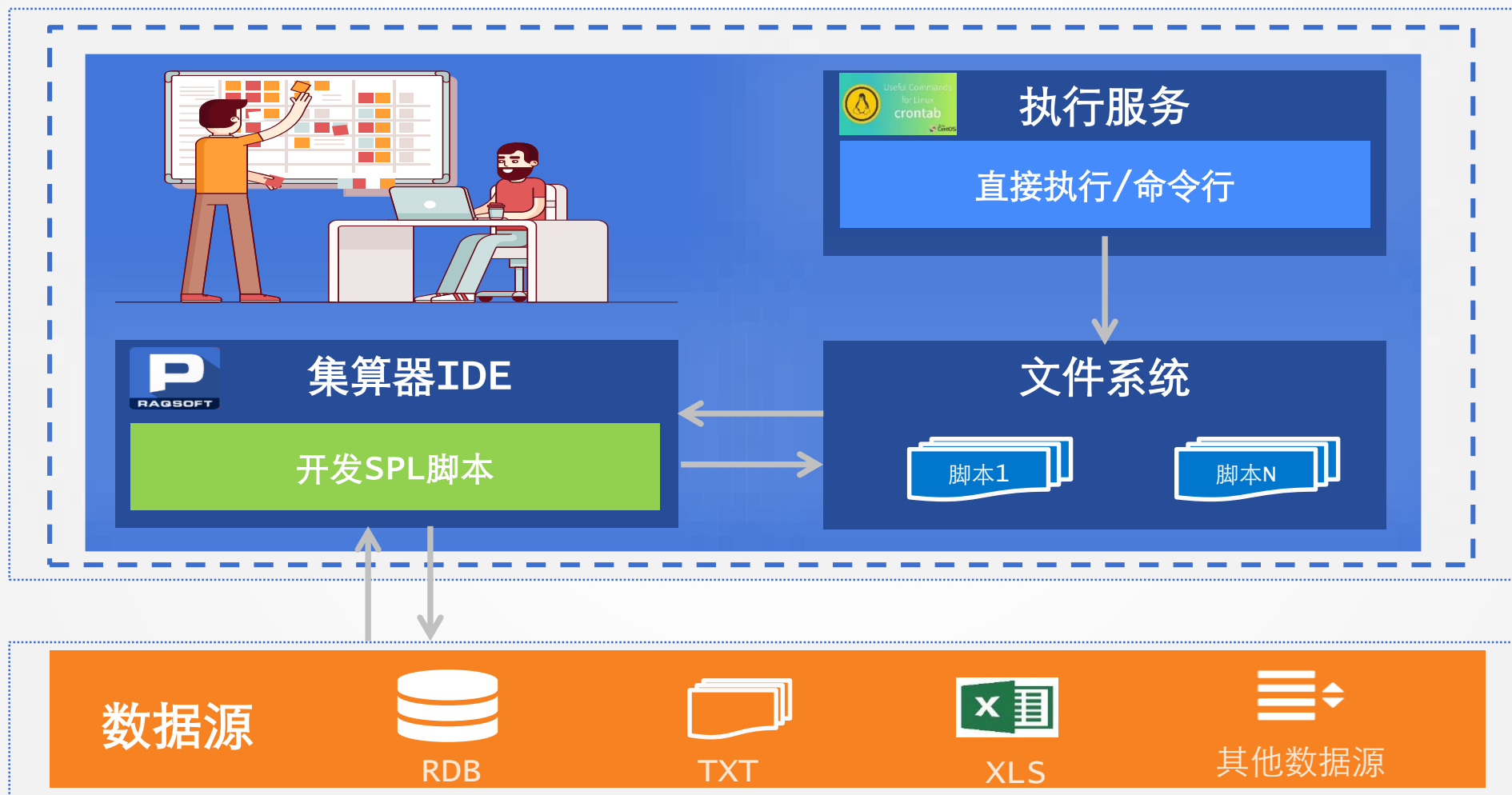
- 语法不是专为结构化数据计算设计
- 复杂运算不易开发
- 版本之间存在不兼容问题
- 外围开源包安装配置麻烦
- 难集成,转入应用计算可能重写



使用集算器实现自由计算

不依赖于数据库的计算能力，直接计算多样性数据进行自由分析！

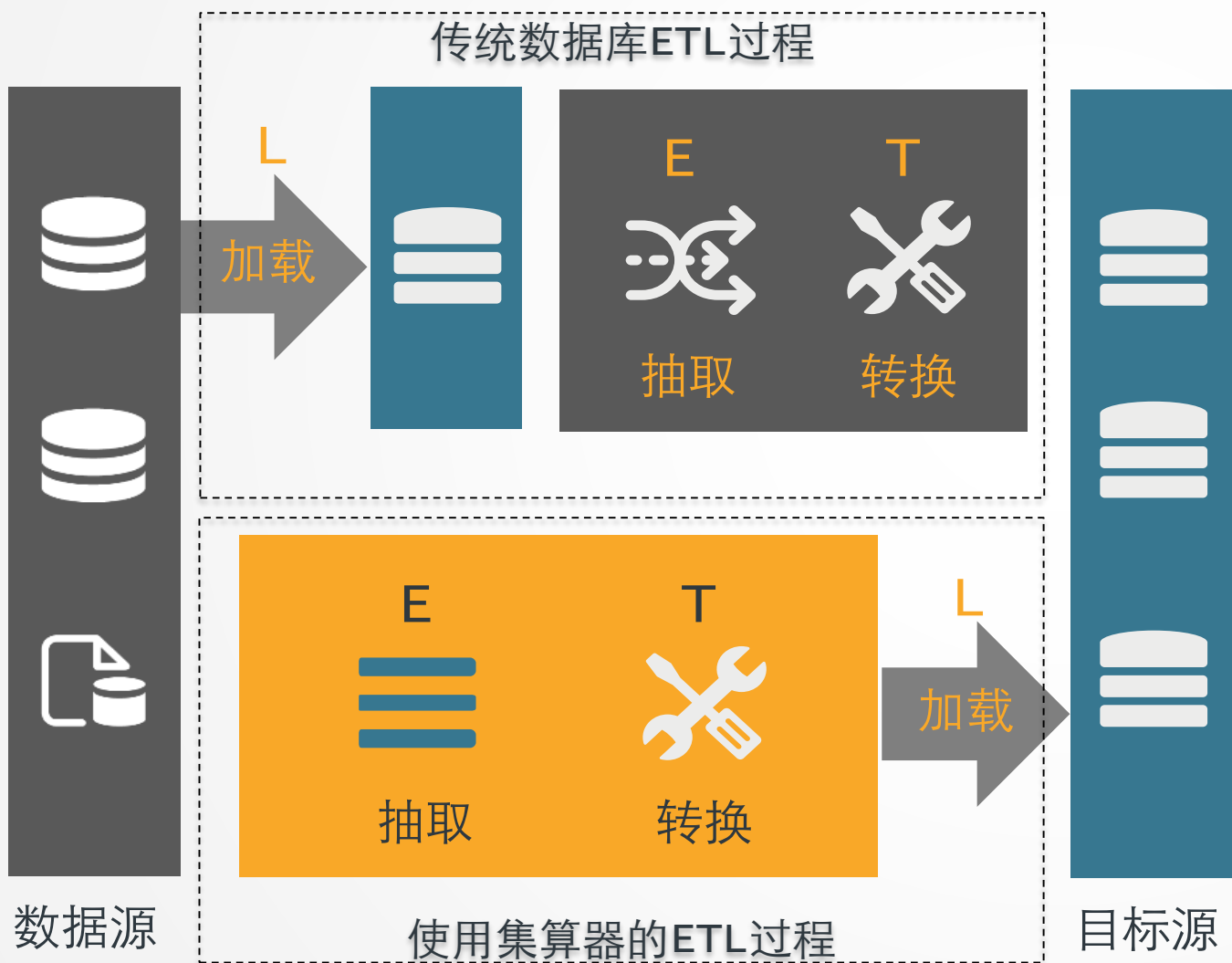
桌面级工具，即装即用，环境配置简单！





使用集算器实现ETL

独立计算引擎，合理地实现 E、T、L 的过程！

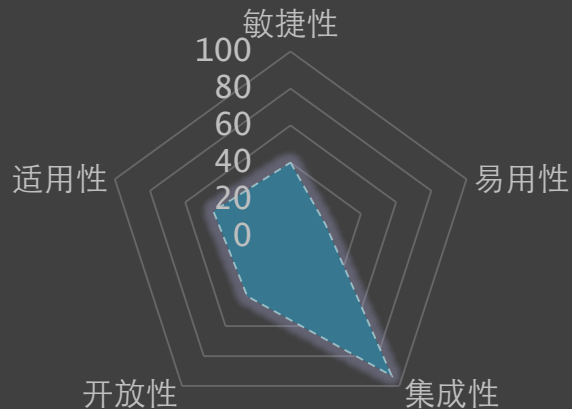


- 库外处理，为数据库减负
- 减少IO，缩短时间窗口
- 实现简单灵活
- 允许多源混合处理

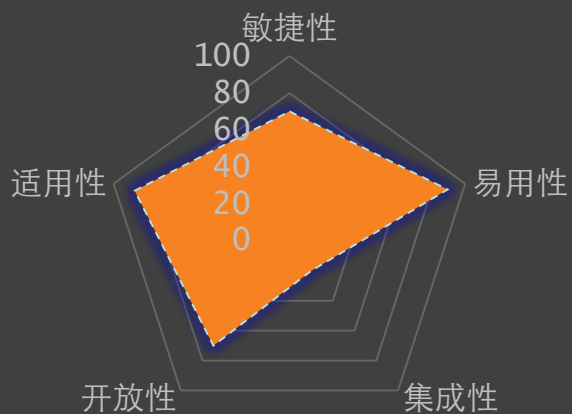
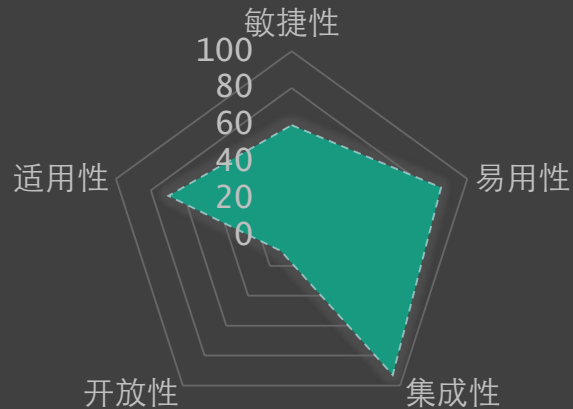


集算器与常用语言的对比优势

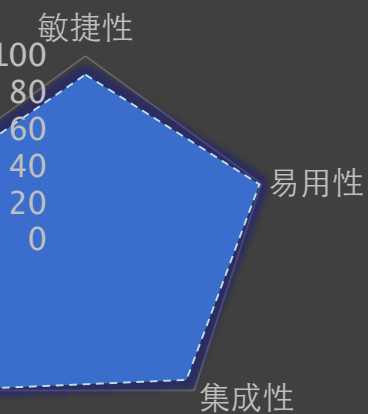
JAVA



SQL



Python



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，易于转为应用内计算



便利开发环境

即装即用，调试功能完善

执行、调试执行、单步执行

设置断点

The screenshot displays a development environment with a menu bar (File, Edit, Program, Tool, Window, Help) and a toolbar. The main workspace shows a code editor with the following code:

```
1 =file("../demo\zh\tx\Sale.txt").import@t().select(month(Datetime)==6)
2 =file("../demo\zh\tx\Sale.txt").import@t().select(month(Datetime)==6)
3 =file("../demo\zh\tx\Storage.txt").import@t().select(month(Date)==5)
4 =file("../demo\zh\tx\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))
9 =A4.new(ID:Commodity,0:Stock,;OosTime,0:TotalOosTime)
10 >A9.keys(Commodity)
11 =A3.run(A9.find(Commodity) Stock=Stock)
```

Below the code editor is a console window showing log levels: WARNING and INFO. To the right, a data grid (A2) displays the following data:

Index	Datetime	Commodity	Volume
1	2009-06-01 08:05:00	20077	28
2	2009-06-01 08:11:40	20056	47
3	2009-06-01 08:18:20	20094	34
4	2009-06-01 08:21:40	20020	19
5	2009-06-01 08:41:40	20013	42
6	2009-06-01 08:45:00	20077	1
7	2009-06-01 08:51:40	20069	19
8	2009-06-01 09:05:00	20011	22
9	2009-06-01 09:08:20	20007	22
10	2009-06-01 09:11:40	20005	39
11	2009-06-01 09:18:20	20085	31
12	2009-06-01 09:21:40	20054	8

系统信息输出，异常随时查看

网格结果所见即所得，易于调试；方便引用中间结果

语法简单，符合自然思维，比其他高级开发语言更简单



敏捷语法体系



某支股票最长连续涨了多少交易日

```
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);
```

Python

	A
1	=file("D:/Stock.xlsx").importxls@t().sort(Date).group(Company)
2	=0
3	=A1.max(A2=if(Price>Price[-1],A2+1,0))

SPL

集合运算领域专业语法，同样
过程代码更简洁！



面向过程计算

完整的循环分支控制

	A	B	C	D	E	F
1	=esProc.query("SELECT orderID as contract, orderDate as date, customer, amount, empID as salesman FROM sales where year(orderDate)=? OR year(orderDate)					
2	=esProc.query(select * from employeeInfo")					
3	>A1.run(salesman=A2.select@1(ID:A1.salesman))		/field value is record			
4	>A1.group(salesman)					
5	=create(salesman, thisyearAmount, lastyearAmount, growthRate, custNumber, bigCustNumber, bigCustProportion)					
6	for A4	=A6(1).salesman.name				
7		=A6.select(year(date)==year).sum(amount)				
8		=A6.select(year(date)==year-1).sum(amount)				
9		=B8/B7-1	/growth rate			
10		=A6.group(customer).(~.sum(amount))				
11		=B10.count()	/number of customer			
12		=B10.count(~>=10000)	/number of big customer			
13		=B10/B11				
14		=A5.insert(0,B6,B7,B8,B9,B11,B12,B13)				
15	result A5					

天然分步、层次清晰、直接引用单元格名无需定义变量



多样性数据源接口

- ✓ 商用 RDBMS: Oracle、MS SQL Server、DB2、Informix
- ✓ 开源 RDBMS: MySQL、PostgreSQL
- ✓ 开源 NOSQL: MongoDB、Redis、Cassandra、ElasticSearch
- ✓ Hadoop家族: HDFS、HIVE、HBase
- ✓ 应用软件: SAP ECC、BW
- ✓ 文件: Excel、Json、XML、TXT
- ✓ 其他: Http Restful、Web Services、OLAP4j 、 ...

内置接口，即装即用

集算器实现自由计算 - 总结



桌面级

开发环境配置简单，开箱即用



易管理

算法外置于脚本，文件形式易管理



开放性

内置多种数据源接口，直接计算



易集成

无框架，轻量级体系易于嵌入集成



低成本

无须代码层引用，降低人员要求



高效率

丰富高效类库和敏捷语法体系

创新技术 推动应用进步！

