

# Java专业数据计算包

集算器 · 润乾软件出品



[www.raqsoft.com.cn](http://www.raqsoft.com.cn)

## ➤ Java程序中的数据计算



由于各种各样原因，我们经常需要在Java程序中进行数据计算处理



**系统设计要求，  
业务逻辑与数据存储分离**



**数据分散在  
不同的数据源中**



**计算逻辑复杂，  
SQL实现困难**

## ➤ 在Java中实现数据计算的痛点



代码繁琐冗长



难以实现复杂算法



耦合性高



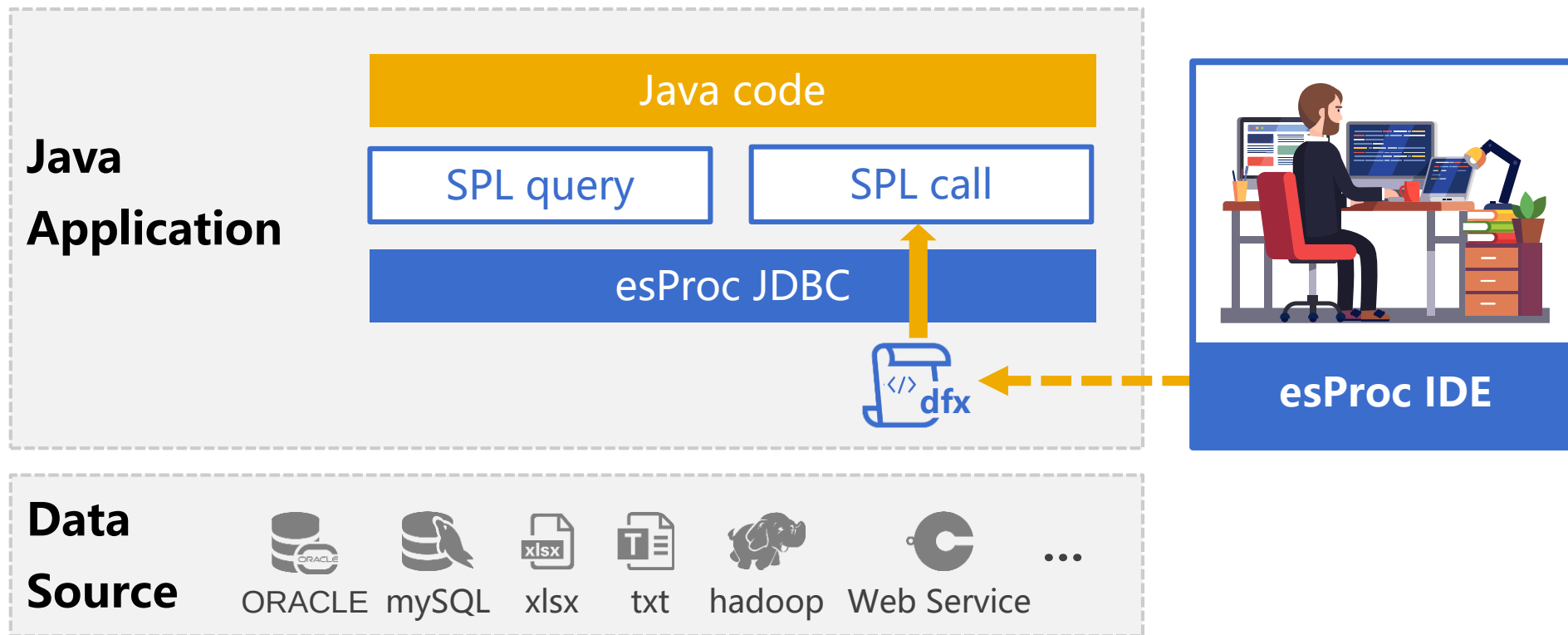
维护困难

## 一段实现简单分组统计的Java代码

```
public static void groupCount(){
    groupBy mainJava = new groupBy();
    List<MainBean> list = mainJava.initList();
    List<MainBean> mainList = new ArrayList<MainBean>();
    Map<String, MainBean> map = new HashMap<String, MainBean>();
    MainBean totalBean = null;
    String employeeId = null;
    for(MainBean mainBean : list) {
        employeeId = mainBean.getEmployeeId();
        if (!map.containsKey(employeeId)) {
            totalBean = new MainBean();
            totalBean.setEmployeeId(employeeId);
            totalBean.setBuyNum(mainBean.getBuyNum());
            totalBean.setGoodsPrice(mainBean.getGoodsPrice());
            totalBean.setTotalBuyNum(mainBean.getTotalBuyNum());
            totalBean.setFreight(mainBean.getFreight());
            totalBean.setTotalGoodsPrice(mainBean.getGoodsPrice() * mainBean.getBuyNum());
            totalBean.setTotalPrice(mainBean.getGoodsPrice() * mainBean.getBuyNum() + mainBean.getFreight());
            mainList.add(totalBean);
            map.put(employeeId, totalBean);
        } else {
            totalBean = map.get(employeeId);
            totalBean.setTotalBuyNum(totalBean.getTotalBuyNum() + mainBean.getTotalBuyNum());
            totalBean.setFreight(totalBean.getFreight() + mainBean.getFreight());
            totalBean.setTotalGoodsPrice(totalBean.getTotalGoodsPrice() + mainBean.getGoodsPrice() * mainBean.getBuyNum());
            totalBean.setTotalPrice(totalBean.getTotalPrice() +
                mainBean.getGoodsPrice() * mainBean.getBuyNum() + mainBean.getFreight());
        }
    }
}
```

这么长的代码仅仅实现了单个字段分组单个字段汇总的功能

## 解决办法：Java专业计算包——集算器



说明：SPL是esProc采用的脚本语言，dfx为脚本文件，相当于外置存储过程，Java通过JDBC接口传入SPL语句实现结构化计算或执行脚本。

## 解决办法：Java专业计算包——集算器



### 专业数据计算语言SPL

离散化与集合性的结合

300多个结构化计算函数

特色网格式代码层次清晰

内存、外存游标、集群运算

循环，分支适应过程式计算

分步计算与完善的调试功能

### 丰富数据接口

RDB: Oracle,DB2,MS SQL,MySQL,PG,....

MongoDB, REDIS, ...

Hadoop: HDFS, HIVE, HBASE

TXT/CSV, JSON/XML, EXCEL

HTTP、ALI-OTS

文件SQL接口

### 专业Jar包

标准JAR包，易于集成

可多线程并行计算

完善的异常处理



# SPL实现分组



## 常规分组

汇总每个人的值班天数

这类基本运算还可以直接用SQL写：  
SELECT name,count(name) FROM  
user/test/duty.xlsx GROUP BY name

|   | A                                               |
|---|-------------------------------------------------|
| 1 | =file( "/Users/test/duty.xlsx" ).importxls@tx() |
| 2 | =A1.groups(name;count(name):count)              |

## 每组Top N

获取每个月、每个人、头三天的加班记录

|   | A                                                 |
|---|---------------------------------------------------|
| 1 | =file( "/Users/test/duty.xlsx" ).importxls@tx()   |
| 2 | =A1.groups(month(workday):mon,name;~.top(3):top3) |

## 对位分组

按顺序分别列出使用Chinese、English、French 作为官方语言的国家数量

|   | A                                                                           |
|---|-----------------------------------------------------------------------------|
| 1 | =connect( "mysql" )                                                         |
| 2 | =A1.query@x( "select * from world.countrylanguage where isofficial= 'T' " ) |
| 3 | [Chinese,English,French]                                                    |
| 4 | =A2.align@a(A3,Language)                                                    |
| 5 | =A4.new(A3(#):name,~.len():cnt)                                             |



# 运算类库



## 专门针对结构化数据表设计

|    | A                                                                           | B                                                       | C |
|----|-----------------------------------------------------------------------------|---------------------------------------------------------|---|
| 1  | =esProc.query("SELECT orderID as contract,                                  | /retrieve sales records                                 |   |
| 2  | =A1.group(salesman)                                                         |                                                         |   |
| 3  | =create(salesman,thisyearAmount, lastyearAmount, custNumber, bigCustNumber) |                                                         |   |
| 4  | for A2                                                                      | =A4(1).salesman                                         |   |
| 5  |                                                                             | =A4.select(year(date)==year).sum(amount)                |   |
| 6  |                                                                             | =A4.select(year(date)==year-1).sum(amount)              |   |
| 7  |                                                                             | =A4.group(customerID).sum(amount)                       |   |
| 8  |                                                                             | =B7.count()                                             |   |
| 9  |                                                                             | =B7.count(>=10000)                                      |   |
|    | A                                                                           | B                                                       | C |
| 1  | =esProc.query("select * from employee")                                     |                                                         |   |
| 2  | =A1.select(sex=="male")                                                     |                                                         |   |
| 3  | =A1.select(birthday>=date("1970-01-01"))                                    |                                                         |   |
| 4  | =A2^A3                                                                      | /intersect, find out male employee born after 1970      |   |
| 5  | =A2&A3                                                                      | /union, find out male employee or employee born after 1 |   |
| 6  | =A2\A3                                                                      | /subtract, find out male employee born before 1970      |   |
| 7  | =A4.sum(salary)                                                             |                                                         |   |
| 8  | =A5.avg(age)                                                                |                                                         |   |
| 9  | =A5.sort(birthday)                                                          |                                                         |   |
| 10 | /set is extensively used as a data type                                     |                                                         |   |
| 11 |                                                                             |                                                         |   |

分组、循环

集合运算

|    | A                                                                              | B                                         | C |
|----|--------------------------------------------------------------------------------|-------------------------------------------|---|
| 1  | =file("traderecord.txt").import@t0                                             |                                           |   |
| 2  | =A1.sort(customerID, tradeDate)                                                |                                           |   |
| 3  | =A2.select(autoType=="Jetta"    autoType=="Passat").dup@t0                     |                                           |   |
| 4  | =A3.derive(interval(tradeDate[-1], tradeDate):space)                           |                                           |   |
| 5  | =A4.select(autoType[-1]=="Jetta" &&autoType=="Passat" &&customerID=customerID) |                                           |   |
| 6  | =A5.avg(space)                                                                 |                                           |   |
| 7  |                                                                                |                                           |   |
| 8  |                                                                                |                                           |   |
| 9  |                                                                                |                                           |   |
|    | A                                                                              | B                                         | C |
| 1  | =esProc.query("select * from employee")                                        |                                           |   |
| 2  | =A1.sort(entryDate)                                                            |                                           |   |
| 3  | =A2.pmin(birthday)                                                             | /select recordNo of employee born at earl |   |
| 4  | =A2(to(A3-1))                                                                  | /directly access employee record via reco |   |
| 5  | =esProc.query("select * from stock where stockCode="000062")                   |                                           |   |
| 6  | =A5.sort(tradeDate)                                                            |                                           |   |
| 7  | =A6.pmax(closePrice)                                                           | /recordNo of highest exchange closing qu  |   |
| 8  | =A6.calc(A7,closePrice/(closePrice-1)*100)                                     |                                           |   |
| 9  |                                                                                |                                           |   |
| 10 |                                                                                |                                           |   |
| 11 |                                                                                |                                           |   |

排序、过滤

有序集合





## 降低JAVA耦合度



### JAVA硬编码

#### 编译执行

硬编码算法必须和主应用一起编译打包

#### 模块化困难

硬编码算法和主应用存在类依赖，难以隔离，耦合度高

#### 难以热切换

硬编码算法修改后会导致整个应用重新编译部署重启，很难做到热切换

### 集算器

#### 解释执行

集算器算法无需事先编译

#### 模块化简单

集算器脚本文件与主应用无依赖关系，可外置于JAVA，可单独维护

#### 容易热切换

无需编译无需重启，可热切换



## 多样性数据源支持



JAVA计算能力难以胜任多样性数据源

集算器内置接口便捷访问

- 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 、 ...

集算器



SQLDB



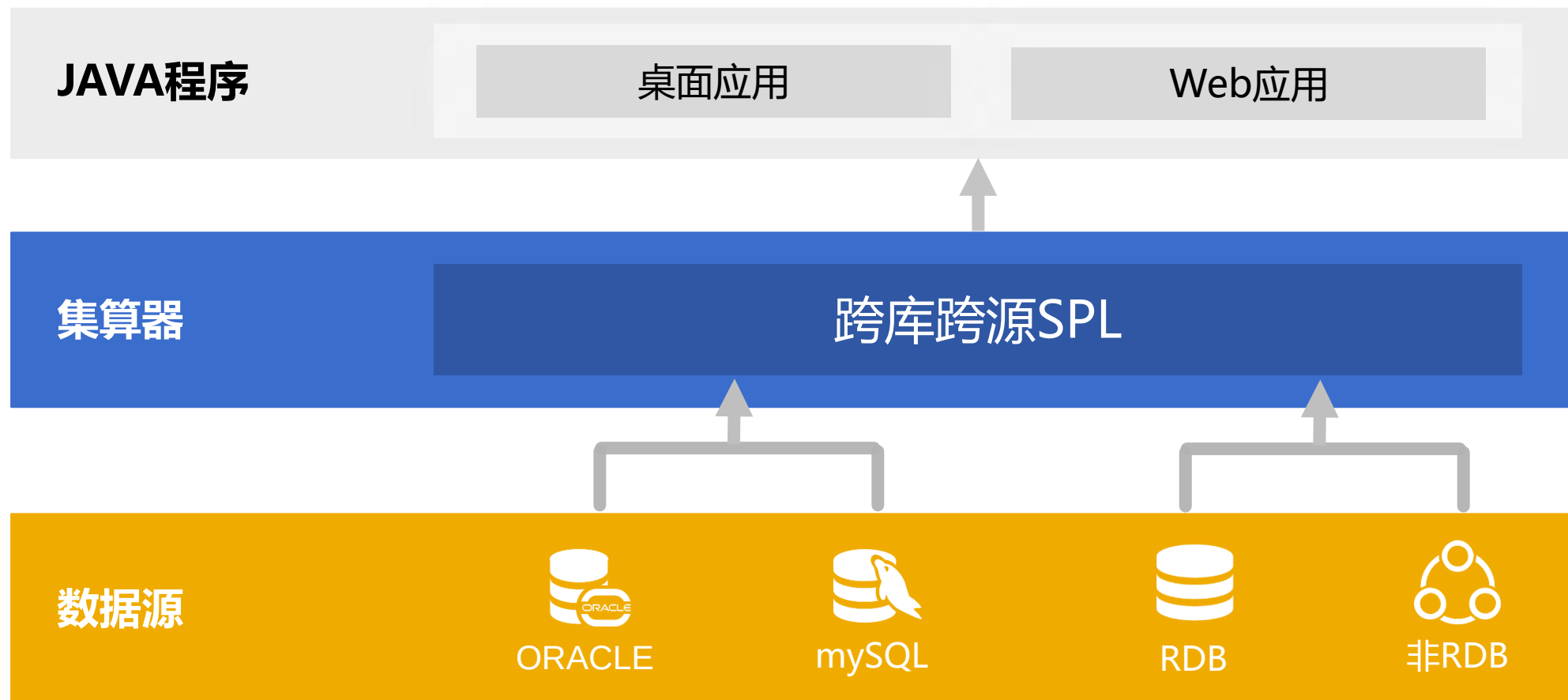
NoSQLDB



File/HDFS



## 跨库计算支持



## ➤ 标准化调用接口

### Java代码

```
...  
Connection con = null;  
Class.forName("com.esproc.jdbc.InternalDriver");  
con= DriverManager.getConnection("jdbc:esproc:local://");  
//调用存储过程, 其中CountName是dfx的文件名  
st =(com. esproc.jdbc.InternalCStatement)con.prepareCall("call CountName()");  
//执行存储过程  
st.execute();  
//获取结果集  
ResultSet rs = st.getResultSet();  
...
```

热部署外置算法  
有效降低耦合性



## 资源链接



- 常用计算的SPL代码 <http://doc.raqsoft.com.cn/esproc/coderefer/>
- 分库后的统计查询 <http://c.raqsoft.com.cn/article/1567657792653>
- 辅助 MongoDB 计算 <http://c.raqsoft.com.cn/article/1570673924802>
- SPL 结构化文本计算 <http://c.raqsoft.com.cn/article/1569400633391>
- JSON 数据计算与入库 <http://c.raqsoft.com.cn/article/1575450305900>
- XML 数据解析与计算 <http://c.raqsoft.com.cn/article/1577363014932>
- JDBC用法参考 <http://c.raqsoft.com.cn/article/1543657951867>
- 安装使用、免费授权 <http://c.raqsoft.com.cn/article/1571895350771>