润乾
RAQSOFT

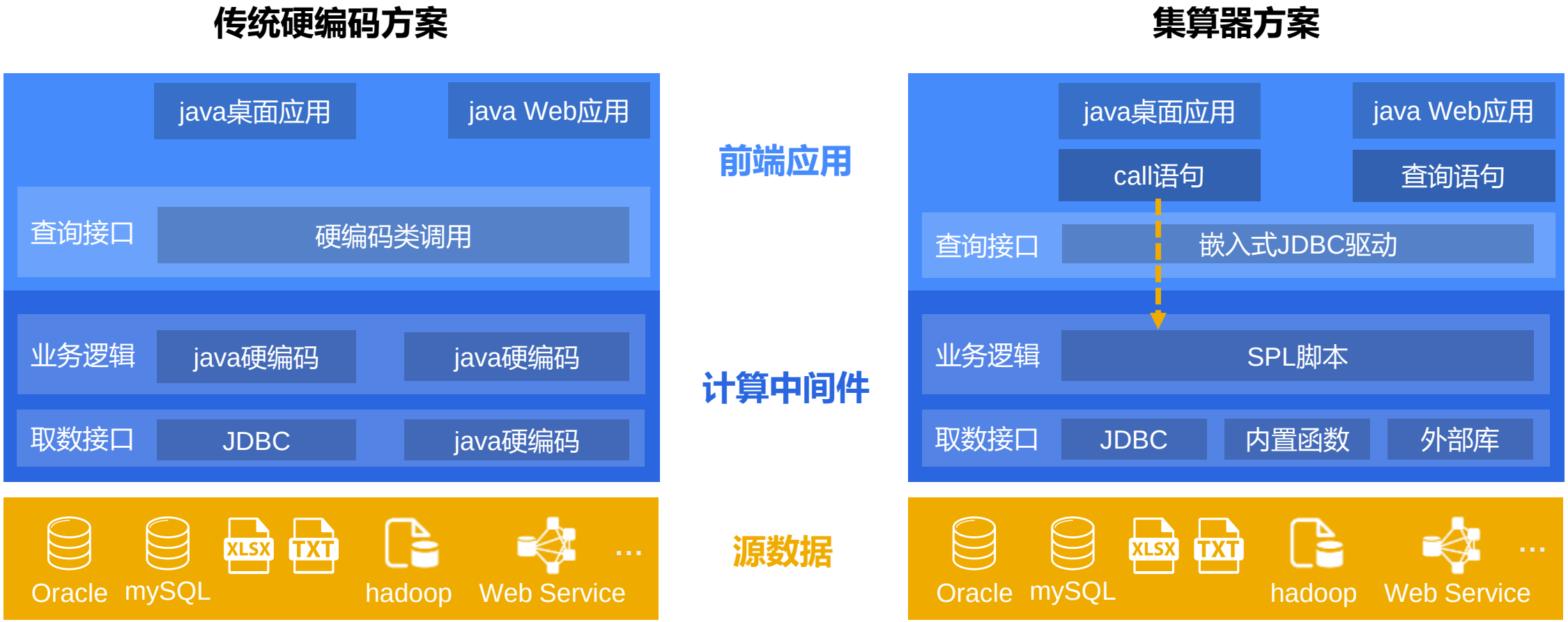
敏捷数据计算中间件

集算器应用场景实施方案

www.raqsoft.com.cn

架构对比

计算中间件：应用与数据之间，独立进行计算的可编程通用软件，常用以解决松耦合、高性能、特殊源计算、多源混算、复杂逻辑等问题。



说明：本文重点讲主流的嵌入式、JAVA应用架构，集算器也支持独立式、非JAVA应用架构。

特点对比

相同点

- 逻辑架构一致
- 实现过程一致

主要区别

- 前端应用查询接口
  集算器方案：JDBC驱动
  硬编码方案：基础类库硬编码
- 业务逻辑实现
  集算器方案：结构化计算函数
  硬编码方案：基础类库硬编码
- 非数据库取数接口
  集算器方案：取数库函数
  硬编码方案：基础类库硬编码

优势

集算器底层能力	中间件特性	优势
独立计算脚本	算法与主应用分离 可实现热切换	降低算法与应用的耦合
完备的计算类库	替代存储过程 不依赖特定数据库	降低库与应用的耦合 降低数据库压力
外部源和数据库可直接混算	减少中间表	降低开发维护难度 降低数据库压力
库内数据可搬出为文件	减少中间表	降低数据库压力
支持离散数据集、有序集合、过程计算、高性能算法	可简化复杂计算	提高开发效率

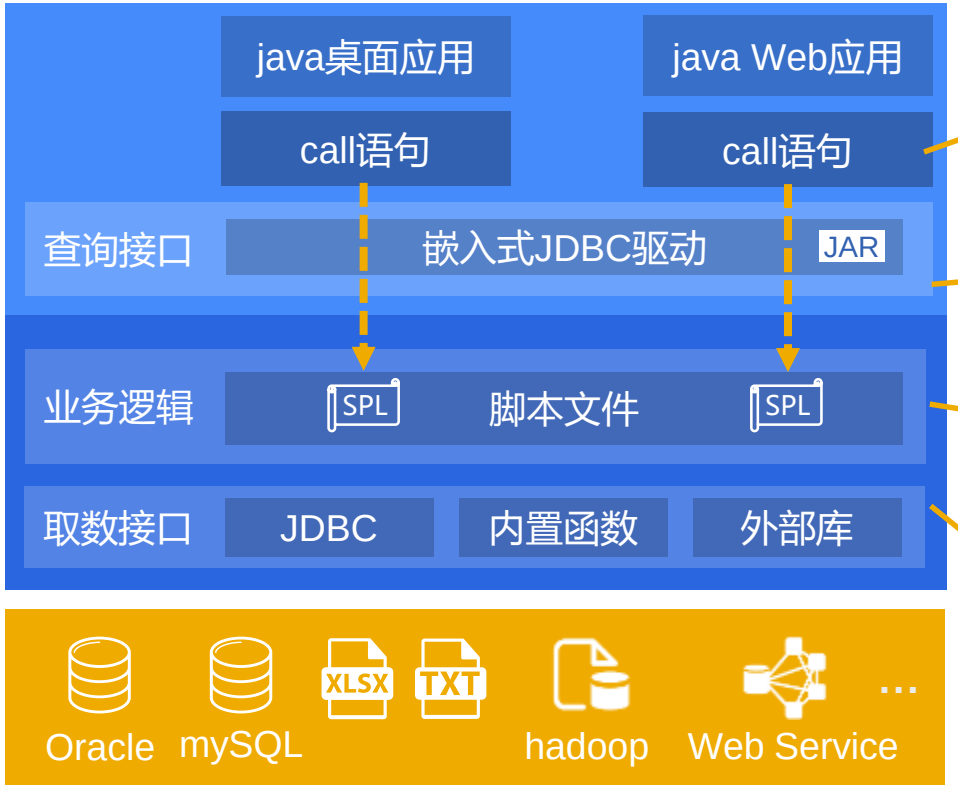
目录 CONTENTS

01

Java应用集成

集成步骤

java应用集成集算器，并调用SPL脚本文件



4.在java代码中执行call语句，通过集算器JDBC调用SPL文件，集算器JDBC解释执行SPL文件，并返回计算结果

1.在java应用中部署集算器驱动jar包

3.在SPL文件中编写具体的业务算法，存储于文件系统。

2.在raqsoftConfig.xml中配置运行环境，如数据源（而不是集算器本身）的JDBC驱动和URL

说明：集算器部署、取数JDBC配置，请参考http://doc.raqsoft.com.cn/esproc/tutorial/jdbcbushu.html

JAVA调用集算器，请参考http://doc.raqsoft.com.cn/esproc/tutorial/bjavady.html

部署集算器JDBC

集算器JDBC提供了公开友好的接口，以便程序员在JAVA应用中调用集算器中间件

部署方法：将驱动程序拷贝到java应用的类路径

esproc_bin_xxxx.jar	集算器计算引擎及JDBC驱动包
icu4j-60.3.jar	处理国际化
jdom-1.1.3.jar	解析配置文件
raqsoftConfig.xml	运行环境配置文件

说明：部署细节请参考http://doc.raqsoft.com.cn/esproc/tutorial/jdbcbushu.html

配置运行环境

集算器运行环境存储于raqsoftConfig.xml，主要是业务数据库或数仓的数据源信息，也包括脚本文件目录、字符集、外部库等。

一个Oracle数据源的配置信息

```
<DB name= “orcl">
        <property name="url" value="jdbc:oracle:thin:@192.168.1.8:1521:runqian"/>
        <property name="driver" value="oracle.jdbc.driver.OracleDriver"/>
        <property name="type" value="1"/>
        <property name="user" value="ydxx"/>
        <property name="password" value="password"/>
        <property name="batchSize" value="0"/>
……
</DB>
```

说明：数据源的JDBC包，同样部署在类路径。

raqsoftConfig.xml的详细配置，请参考http://doc.raqsoft.com.cn/esproc/tutorial/jdbcbushu.html

集算器JDBC vs 数据源JDBC

相同点

- 部署方法类似
- 遵循同一套接口规范

主要区别

- 调用位置

集算器JDBC：java应用中
数据源JDBC：集算器脚本文件中

- 调用方法

集算器JDBC：java代码
数据源JDBC ： SPL函数

- 作用

集算器JDBC：访问集算器脚本文件，获得业务算法的计算结果。
数据源JDBC ： 访问数据库/仓库，获得原始数据。

脚本文件

SPL脚本文件即业务中间件算法，由程序员编写，通常需向从数据源取数，再实现业务算法，最后由集算器JDBC向主程序返回计算结果。

例子：orcl是数据库数据源，java应用需从sales表取数据，部分数据如下

ordered	client	sellerid	amount	orderdate
1	UJRNP	17	392	2012/11/2 15:28
2	SJCH	6	4802	2012/11/9 15:28
3	UJRNP	16	13500	2012/11/5 15:28
4	PWQ	9	26100	2012/11/8 15:28
5	PWQ	11	4410	2012/11/12 15:28

原单线程取数性能较差，下面用集算器做中间件，使用8线程并行查询，并合并查询结果。脚本文件conj.dfx如下：

	A	B	C
1	=8.(range(1,100000000L,~:4))	/	
2	fork A1	=connect("dbsource")	/多线程并行
3		=B2.query@x("select * from sales where orderid>=? and orderid<?",A2(1),A2(2))	/线程内查询
4	=A2.conj()		/合并输出

说明：orderid的起止范围是1-100000000L，A1代码将其等分为8个时间区间，每线程一个，比如2号线程为12500000-25000000

脚本文件

java应用通过集算器jdbc调用脚本文件

```
……
Class.forName("com.esproc.jdbc.InternalDriver");
con=DriverManager.getConnection("jdbc:esproc:local://");
PreparedStatement pstmt = con.prepareStatement( “call conj()");
ResultSet rs=pstmt. executeQuery()
……
```

脚本文件可以带参数，比如按订单金额范围（minAmount,maxAmount）过滤数据，脚本文件如下：

	A	B	C
1	=8.(range(1,100000000L,~:4))	/	
2	fork A1	=connect("dbsource")	/多线程并行
3		=B2.query@x("select * from sales where orderid>=? and orderid<?“ and amount>=? and amount<?, A2(1),A2(2),minAmount,maxAmount)	/线程内查询
4	=A2.conj()		/合并输出

此时JAVA代码应按参数查询编写代码：

```
…
PreparedStatement pstmt = con.prepareStatement( “call conj(?,?)” ); //也可写成call conj(4000,8000)
pstmt.setObject(1, 4000);pstmt.setObject(2, 8000);
ResultSet rs=pstmt. executeQuery()
……
```

//更多JAVA调用集算器内容，请参考http://doc.raqsoft.com.cn/esproc/tutorial/bjavady.html

脚本文件

数据源也可以是文本、Excel等。

源数据为tab分隔的文本文件sales.txt，前几行如下：

ordered	client	sellerid	amount	orderdate
1	UJRNP	17	392.0	2012-11-02 15:28:05
2	SJCH	6	4802.0	2012-11-09 15:28:05
3	UJRNP	16	13500.0	2012-11-05 15:28:05

按参数查询，按client分组，对amount求和，脚本文件run.dfx如下：

	A	B
1	=file("d:/sales.txt").import@t()	/打开文本文件
2	=A1.select(amount>=minAmount && amount<maxAmount)	/参数查询
3	=A1.groups(client;sum(amount):sAmount)	/分组汇总

上述脚本也可合为一句：

	A
1	=file(“d:/sales.txt”).import@t().select(amount>=minAmount &&amount<maxAmount).groups(client;sum(amount):sAmount)

计算结果如下：

client	sAmount
AVU	482640
AYWYN	455320
BTMMU	496226

脚本文件

作为对比，下面用JAVA硬编码（传统计算中间件）实现分组汇总算法

```
Comparator<salesRecord> comparator = new Comparator<salesRecord>() {
                public int compare(salesRecord s1, salesRecord s2) {
                        if (!s1.client.equals(s2.client)) {
                                return s1.client.compareTo(s2.client);
                        } else {
                                return s1.ID.compareTo(s2.ID);
                        }
                }
};
Collections.sort(sales, comparator);
ArrayList<resultRecord> result=new ArrayList<resultRecord>();
salesRecord standard=sales.get(0);
float sAmount=standard.value;
for(int i = 1;i < sales.size(); i ++){
                salesRecord rd=sales.get(i);
                if(rd.client.equals(standard.client)){
                        sAmount=sAmount+rd.value;
                }else{
                        result.add(new resultRecord(standard.client,sAmount));
                        standard=rd;
                        sAmount=standard.value;
                }
}
result.add(new resultRecord(standard.client,sAmount));
return result;
```

脚本文件

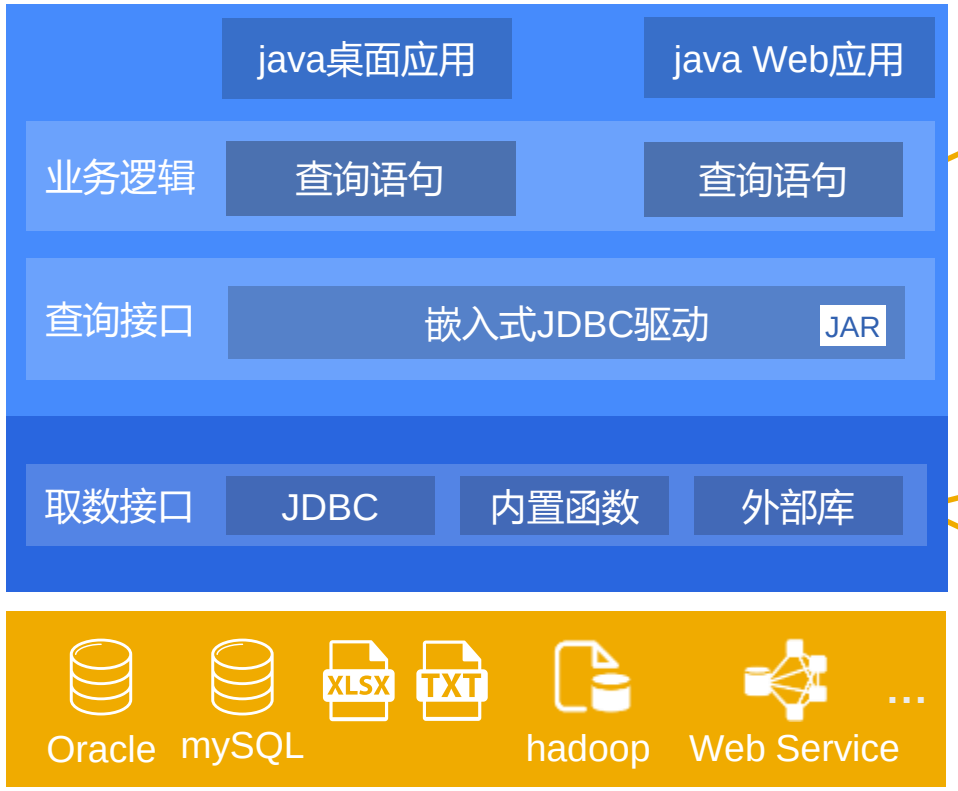
SPL脚本文件存放于操作系统目录

通常情况下建议按功能模块划分目录，也可按应用类型、时间、版本划分目录

```
-主目录
    -客户管理
        -run.dfx()
    -考勤绩效
    -企业资源管理
        -财务管理
            -固定资产统计.dfx
            -现金流量查询.dfx
            -呆账坏账预警分析.dfx
        -库存管理
            ...
```

查询语句

如果脚本文件较简单，可用SPL查询语句替代SPL call语句+脚本文件。前者类似于java调用SQL语句，后者类似调用存储过程。



3.通过查询语句直接实现业务逻辑：

1.在java应用中部署集算器驱动jar包

2.在raqsoftConfig.xml中配置运行环境

查询语句

SPL call语句+脚本文件

```
…
PreparedStatement pstmt =
con.prepareStatement(  “call run(?,?)”  );
pstmt.setObject(1, 4000);
pstmt.setObject(2, 8000);
ResultSet rs=pstmt. executeQuery()
…
```

SPL查询语句

```
…
PreparedStatement pstmt =
con.prepareStatement(“=file(\“d:/sales.txt\”).import@t(
).select(amount>=?
&&amount<?).groups(client;sum(amount):sAmount)");
pstmt.setObject(1, 4000);
pstmt.setObject(2, 8000); );
ResultSet rs=pstmt. executeQuery()
…
```

run.dfx

	A
1	=file(“d:/sales.txt”).import@t().select(amount>=minAmount &&amount<maxAmount).groups(client;sum(amount):sAmount)

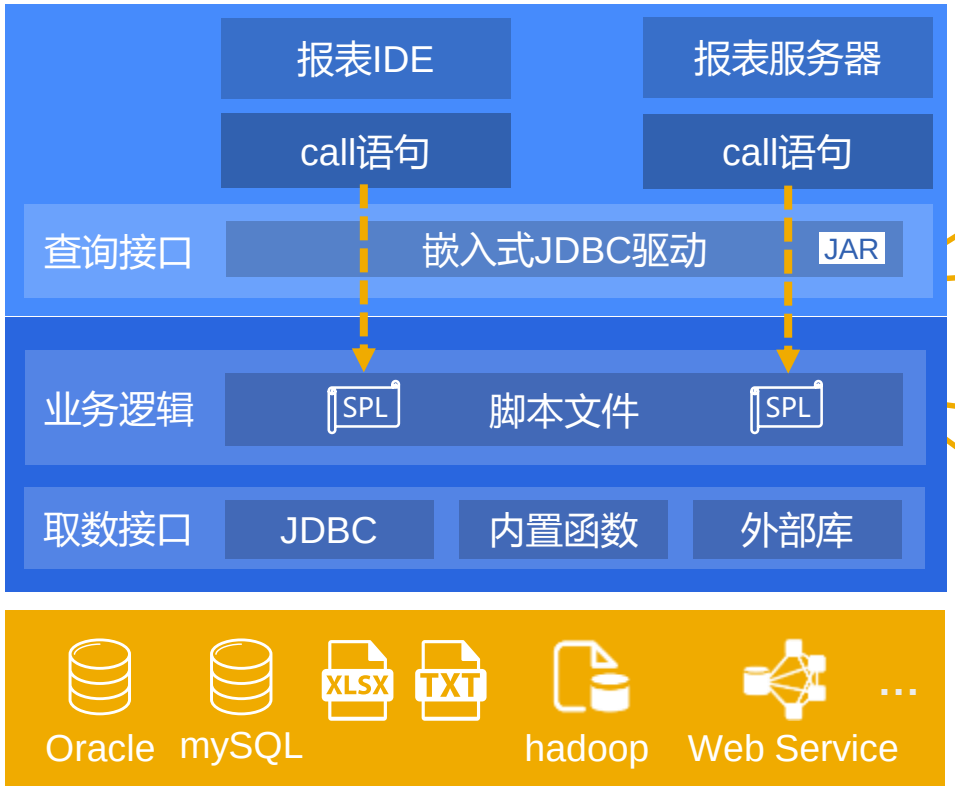
目录 CONTENTS

02

报表集成

集成步骤

java报表工具是JAVA应用的一种，其中报表IDE一般为桌面应用，报表服务一般为Web应用，两者都可以集成集算器，从而实现计算中间件



4.在报表中新建存储过程数据集，调用集算器脚本文件；或新建SQL数据集，直接写SPL查询语句

1.IDE或Web服务中部署集算器驱动jar包

3.在SPL文件中编写具体的业务算法，存储于文件系统。

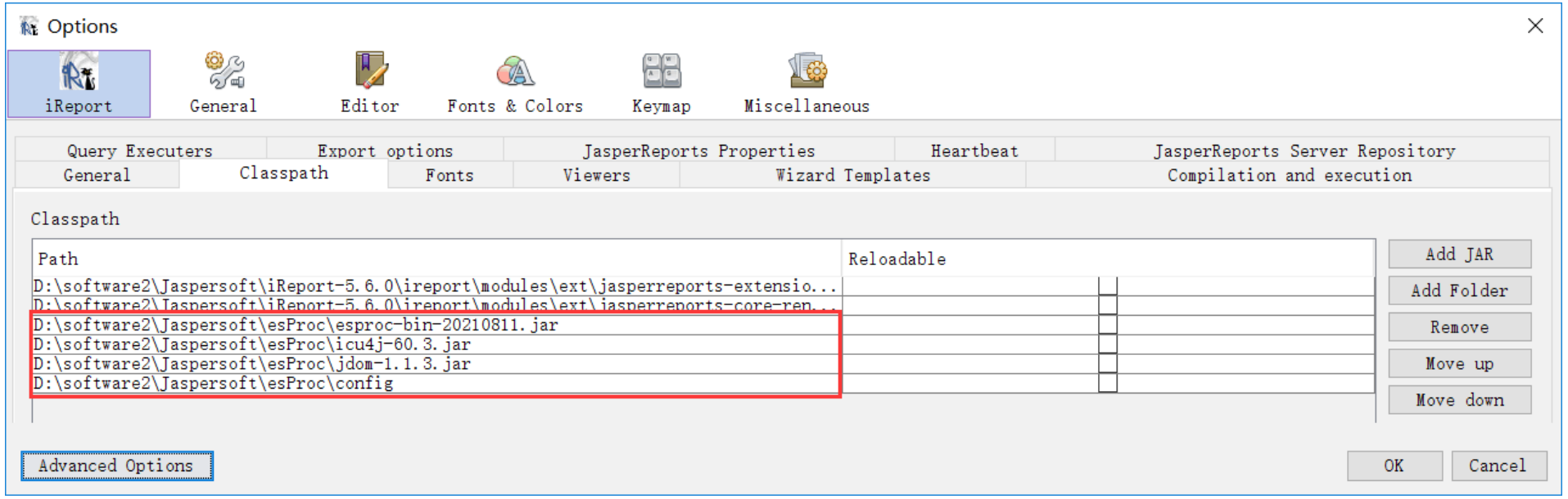
2.在raqsoftConfig.xml中配置运行环境，如数据源（而不是集算器本身）的JDBC驱动和URL

部署集算器JDBC

报表工具部署集算器JDBC的方法，和普通JAVA应用原理相同

比如在开源报表BIRT的IDE中部署，只需把集算器驱动复制到：
[安装目录]\plugins\org.eclipse.birt.report.data.oda.jdbc_4.6.0.v20160607212
配置raqsoftConfig.xml的方式不变，这里不再赘述

有些报表工具提供了可视化界面，可更方便地指定驱动jar包，比如JasperReport



说明：上图中，raqsoftConfig.xml放置于config目录中，也可将其复制到任意jar包里

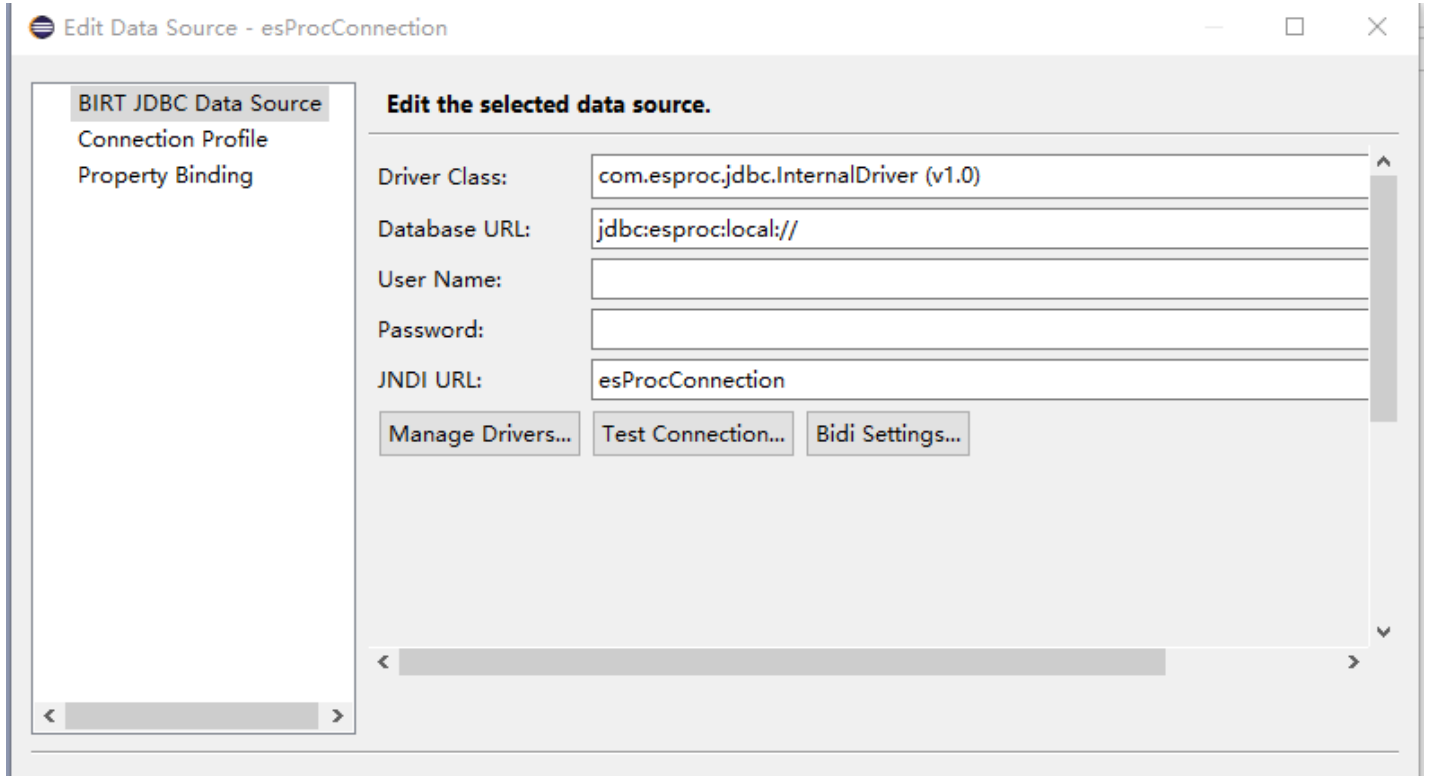
报表调用集算器脚本

写好脚本文件之后，首先应在报表IDE中建立数据源，并指向集算器。下面以BIRT报表为例。

Driver Class选择：com.esproc.jdbc.IntervalDriver (v1.0)

Database URL填写：jdbc:esproc:local://

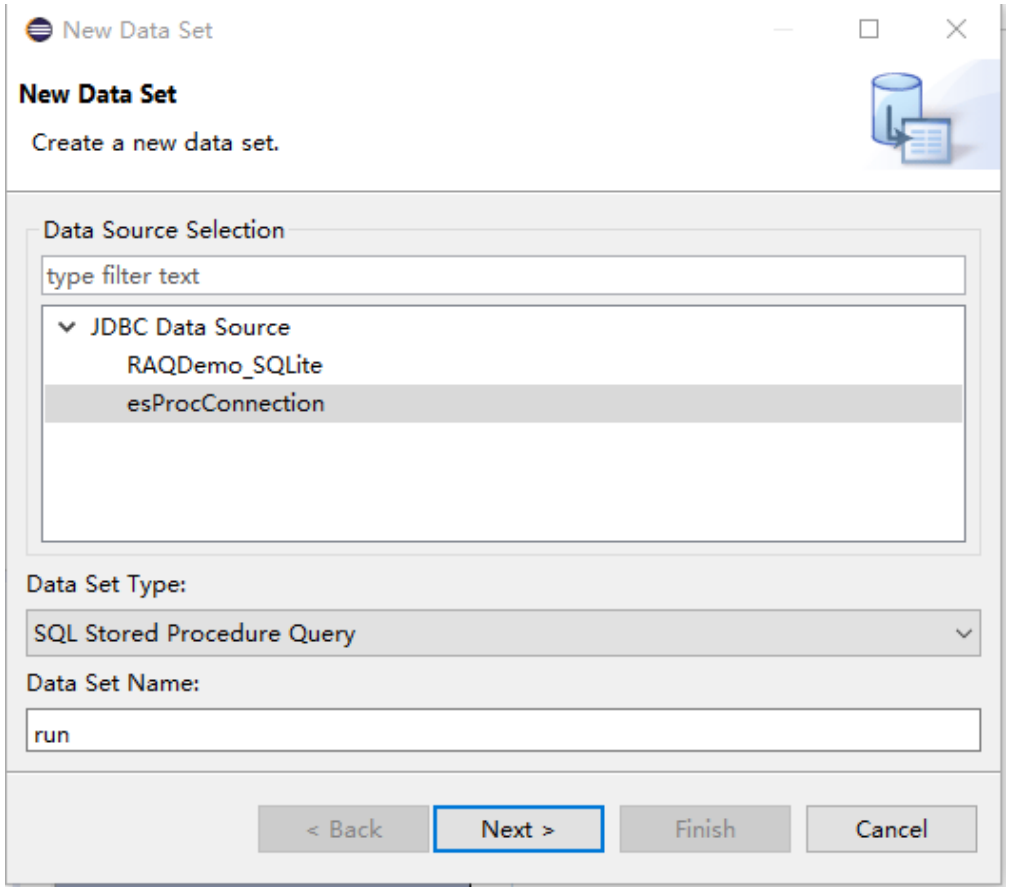
JNDI 数据源名：自由填写

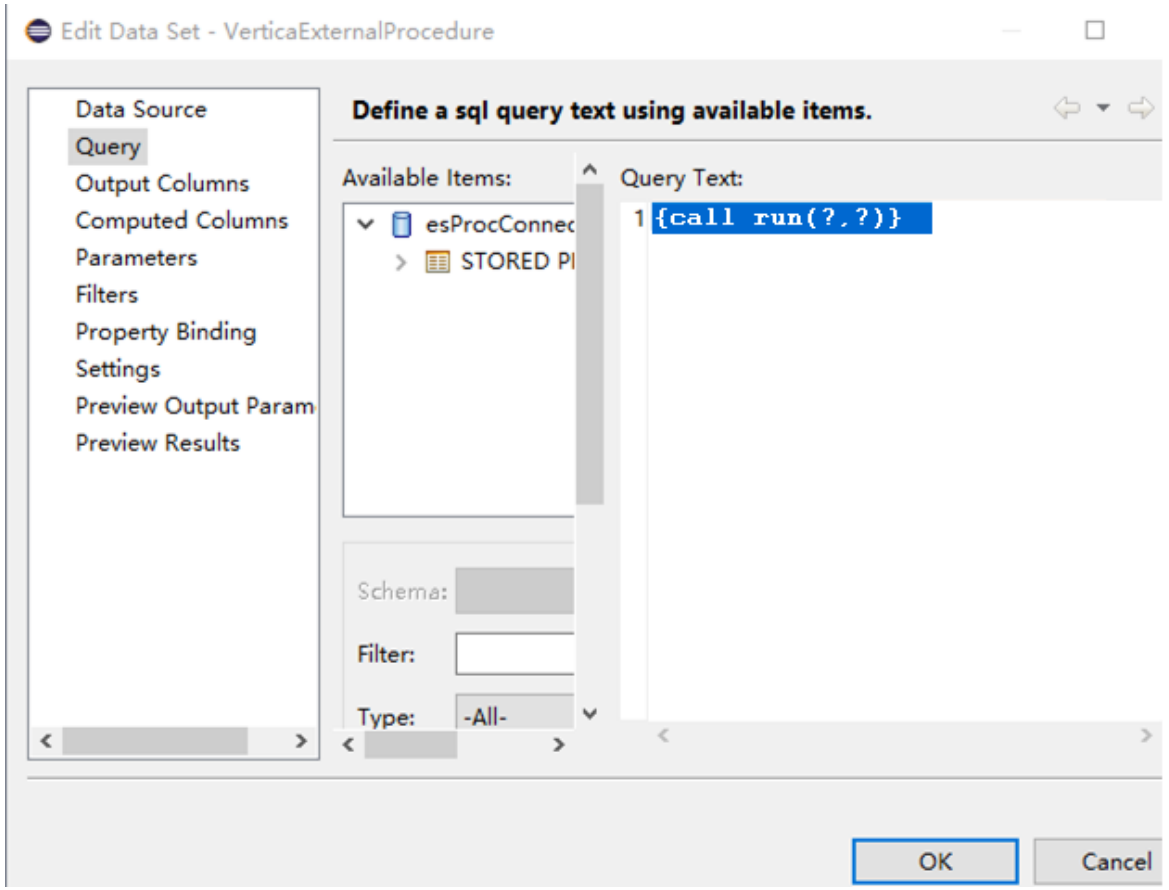


报表调用集算器脚本

新建存储过程数据集，其用法类似数据库存储过程

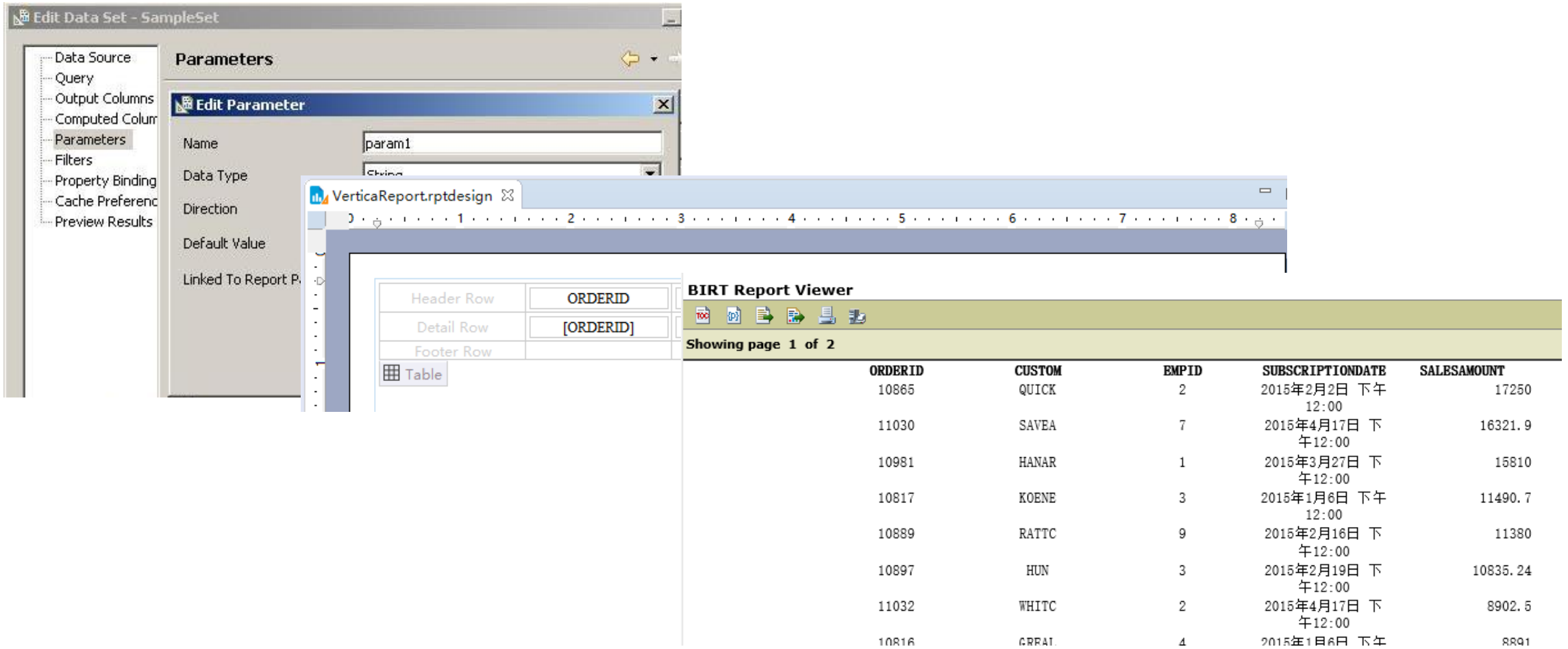
选择集算器数据源，调用集算器脚本文件





报表调用集算器脚本

设定参数、设计报表，预览报表，其用法和普通数据集一样。



ORDERID	CUSTOM	EMPID	SUBSCRIPTIONDATE	SALESAMOUNT
10865	QUICK	2	2015年2月2日 下午12:00	17250
11030	SAVEA	7	2015年4月17日 下午12:00	16321.9
10981	HANAR	1	2015年3月27日 下午12:00	15810
10817	KOENE	3	2015年1月6日 下午12:00	11490.7
10889	RATTC	9	2015年2月16日 下午12:00	11380
10897	HUN	3	2015年2月19日 下午12:00	10835.24
11032	WHITC	2	2015年4月17日 下午12:00	8902.5
10816	GREAL	4	2015年1月6日 下午	8891

算法举例

横向分栏：以栏数pColNum作参数，将员工名单在报表中横向分栏摆放

栏数=2时，报表应呈现

EID	NAME	DEPT	EID2	NAME2	DEPT2
1	Rebecca	R&D	2	Ashley	Finance
3	Rachel	Sales	4	Emily	HR
5	Ashley	R&D	6	Matthew	Sales
7	Alexis	Sales	8	Megan	Marketing
9	Victoria	HR	10	Ryan	R&D
11	Jacob	Sales	12	Jessica	Sales

栏数=3时，报表应呈现

EID	NAME	DEPT	EID2	NAME2	DEPT2	EID3	NAME3	DEPT3
1	Rebecca	R&D	2	Ashley	Finance	3	Rachel	Sales
4	Emily	HR	5	Ashley	R&D	6	Matthew	Sales
7	Alexis	Sales	8	Megan	Marketing	9	Victoria	HR
10	Ryan	R&D	11	Jacob	Sales	12	Jessica	Sales
13	Daniel	Finance	14	Alyssa	Sales	15	Alexis	Sales
16	Christopher	Production	17	Hannah	Marketing	18	Jonathan	Administration

使用脚本文件实现该算法

	A	B	C
1	=demo.query("select EId,Name,Dept from employee ")		
2	=A1.step(pColNum,1)		/取第1栏，作为中间结果
3	for to(2:pColNum)	=A1.step(pColNum,A3)	/取第N栏
4		=A2=A2.join(#,B3:#,EID:${"EID"/A3}, NAME:${"DEPT"/A3}, DEPT:${"DEPT"/A3})	/在中间结果右侧拼上第N栏，作为新的中间结果
5	return A2		/返回最终分栏结果

说明：算法中使用了集算器动态语法中的宏，参考http://doc.raqsoft.com.cn/esproc/tutorial/hong.html
算法中使用了按序号关联的方法，参考http://doc.raqsoft.com.cn/esproc/func/join.html

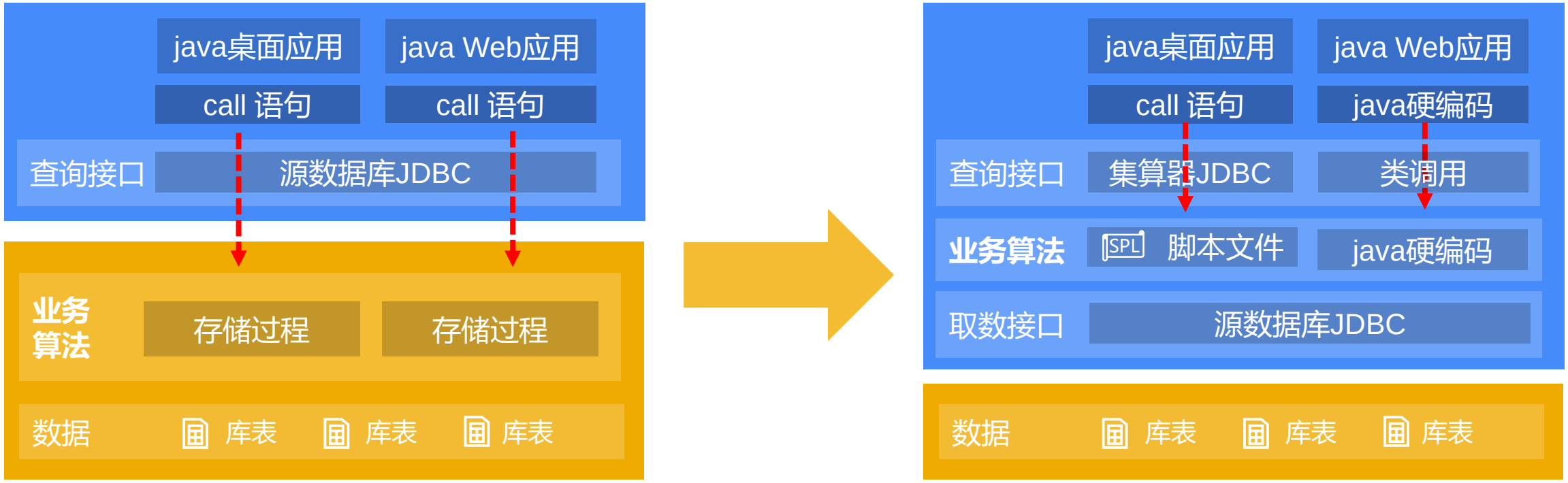
目录 CONTENTS

03

库外存储过程

实现思路

用集算器脚本代替数据库存储过程，从而实现业务算法和应用程序的解耦



算法举例 切换同构数据源

history、current是两个同类型的数据库，两者库结构一样但数据不同。查询计算时，中间件须通过参数选择使用哪个数据库。下面用集算器脚本switch.dfx来实现。

	A	B
1	=${pSource}.query(“select * from sales”)	

pSource是宏参数，代表数据源名称。如果java程序中pSource取值为”history”，则在该库上执行查询，即：

```
PreparedStatement pstmt = con.prepareStatement(“call switch(?)");
pstmt.setObject(1, “history”);
pstmt.execute()
```

计算结果如右：

orderid	client	sellerid	amount	orderdate
1	UJRNP	17	392	2012/11/2 15:28
2	SJCH	6	4802	2012/11/9 15:28
3	UJRNP	16	13500	2012/11/5 15:28
4	PWQ	9	26100	2012/11/8 15:28
5	PWQ	11	4410	2012/11/12 15:28

如果pSource取值为”current”，则计算结果不同

orderid	client	sellerid	amount	orderdate
982	SJCH	12	10900	2019/7/12 15:28
983	GLH	13	8330	2019/7/14 15:28
984	SJCH	19	5684	2019/7/20 15:28
985	YZ	14	27100	2019/7/13 15:28
986	HANAR	10	11100	2019/7/15 15:28

算法举例 切换异构数据源

应用系统可能在多种数据库间迁移，比如从MySQL迁移到Oracle，两者库结构相同。开发时使用与数据库无关的标准SQL，迁移时无需修改SQL语句，只需将数据库类型作为参数传入脚本文件，即可翻译成特定的数据库SQL。

如下是脚本文件run.dfx。

	A	B
1	select left(client,2),year(orderDate) y,sum(amount) sAmount from sales group by left(client,2),year(orderDate)	/SPL标准SQL
2	=A1.sqltranslate(sqlType)	/根据参数将标准SQL翻译成指定数据库的SQL
3	=connect@l("dbsource").query@x(A2)	/执行翻译后的SQL

如果java程序中sqlType取值为”mysql”，则A2中的SQL将被翻译为：

select left(client,2),year(orderDate) y,sum(amount) sAmount from sales group by left(client,2),year(orderDate)

如果sqlType取值为”oracle”，则A2的翻译结果是：

select left(client,2),EXTRACT(YEAR FROM orderDate) y,sum(amount) sAmount from sales group by left(client,2),EXTRACT(YEAR FROM orderDate)

算法举例 动态对齐

很多算法数据库难以实现，或者部分数据库难以实现，比如：外部参数pclient是动态的大客户列表，请按该列表的顺序统计订单金额。如果pclient= [“HL”,“MIP”,“SJCH”]，则计算结果应当如下：

client	samount
HL	305320
MIP	397000
SJCH	500298

使用集算器脚本文件实现：

	A
1	=connect@l("dbsource").query@x("select client,sum(amount) sAmount from sales where client in(?) group by client ",pclient)
2	=A2.align(A1,client)

目录 CONTENTS

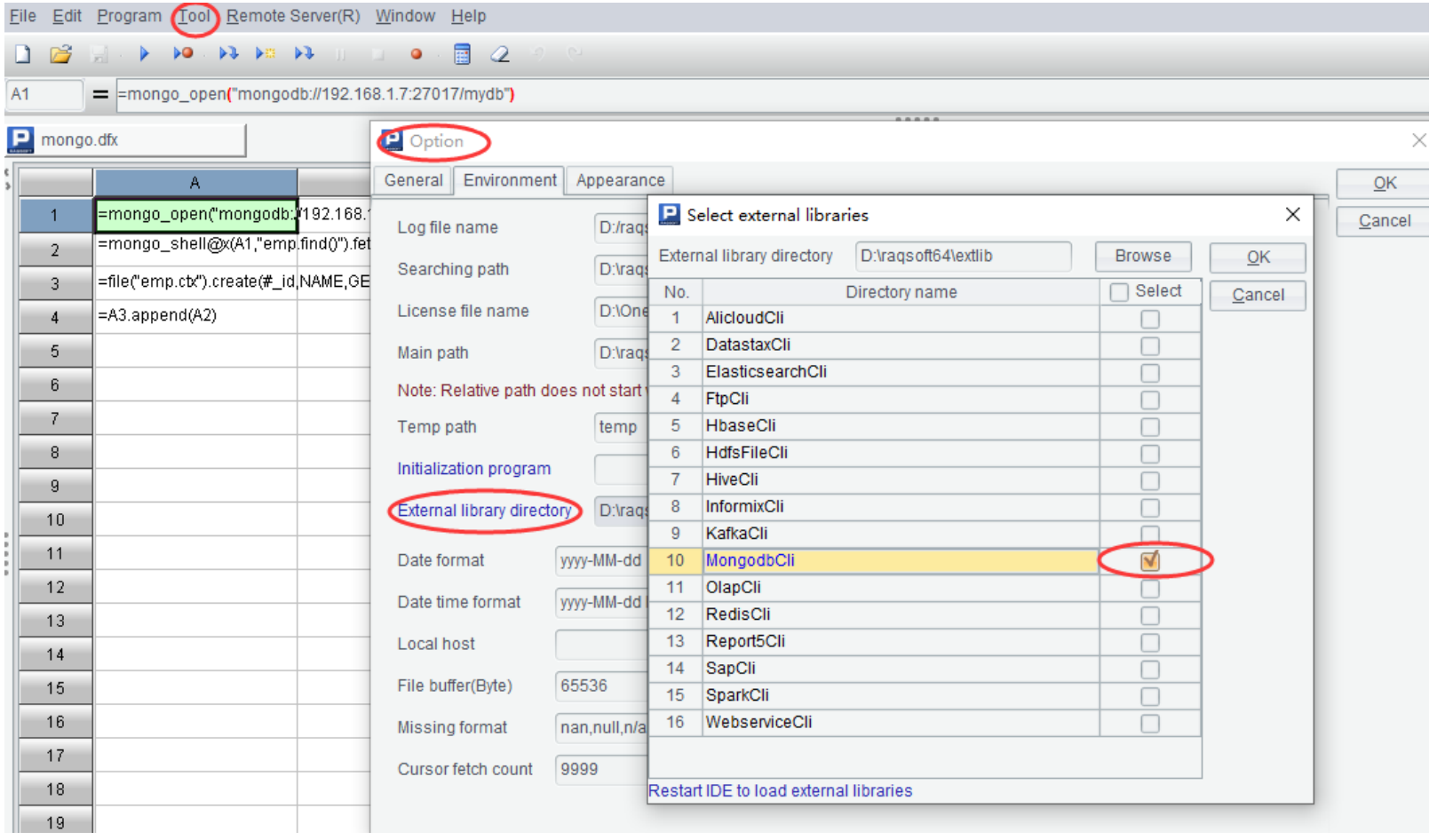
04

多样数据源

特殊数据源

场景：针对webService、MongoDB、Hive等特殊数据源，集算器提供了外部库接口，配合数据源自带的jar包，可实现方便快捷的特殊源计算中间件。

以MongoDB为例，首先配置外部库



特殊数据源

将Mongodb的jar包放在外部库Mongodb目录中，如下：

在脚本文件中，使用shell命令查询emp collection，完成查询和分组计算

	A	B
1	=mongo_open("mongodb://192.168.1.7:27017/mydb")	/连接
2	=mongo_shell@x(A1,"emp.find()")	/查询取数
3	=A2.groups(department;count(empid):total)	/分组汇总

扩展阅读：更多外部库用法，参考http://doc.raqsoft.com/esproc/func/wbk.html

SQL语法

对于文本和Excel，集算器已提供内置函数访问，比如前面的例子：文本分组汇总。

	A	B
1	=file("d:/sales.txt").import@t()	/打开文本文件
2	=A1.groups(client;sum(amount):sAmount)	/分组汇总

除了内置函数，集算器还提供了SQL语法，可以用更方便的方式访问文本，上面的脚本在JAVA可以写成一句SQL：

```
……
Class.forName("com.esproc.jdbc.InternalDriver");
con=DriverManager.getConnection("jdbc:esproc:local://");
ResultSet rs = con.executeQuery(“select client, sum(amount) sAmount from d:/sales.txt")
……
```

SQL举例

过滤

select ID,NAME,GENDER,AGE from students.txt where GENDER='F' and AGE>24

排序

select ID,NAME,GENDER,AGE from students.xlsx order by AGE

集合

select ID,NAME,GENDER,AGE from class1.txt union select ID,NAME,GENDER,AGE from class2.xls

多库汇合

背景：逻辑上的同一张表，被分散在多个物理库中，对这样的数据进行汇合并计算。

mysql存储2015年的订单数据，Oracle存储2013-2014年的数据，请将两个库的数据合并返回

	A	B
1	=connect("org.hsqldb.jdbcDriver","jdbc:hsqldb:hsql://127.0.0.1/demo?user=sa")	=connect("com.mysql.jdbc.Driver","jdbc:mysql://127.0.0.1:3306/demo?user=root&password=password")
2	=A1.query@x("select * from sales")	=B1.query@x("select * from sales")
3	=A2 | B2	

请合并两库数据，并按订单金额排序。

	A	B
1	=connect("org.hsqldb.jdbcDriver","jdbc:hsqldb:hsql://127.0.0.1/demo?user=sa")	=connect("com.mysql.jdbc.Driver","jdbc:mysql://127.0.0.1:3306/demo?user=root&password=password")
2	=A1.query@x("select * from sales order by amount")	=B1.query@x("select * from sales order by amount")
3	=[A2,B2].merge(AMOUNT)	

说明：例子中用到了归并函数merge，比简单排序sort快得多。多库汇总还有很多技巧，参见

http://c.raqsoft.com.cn/article/1536666621882

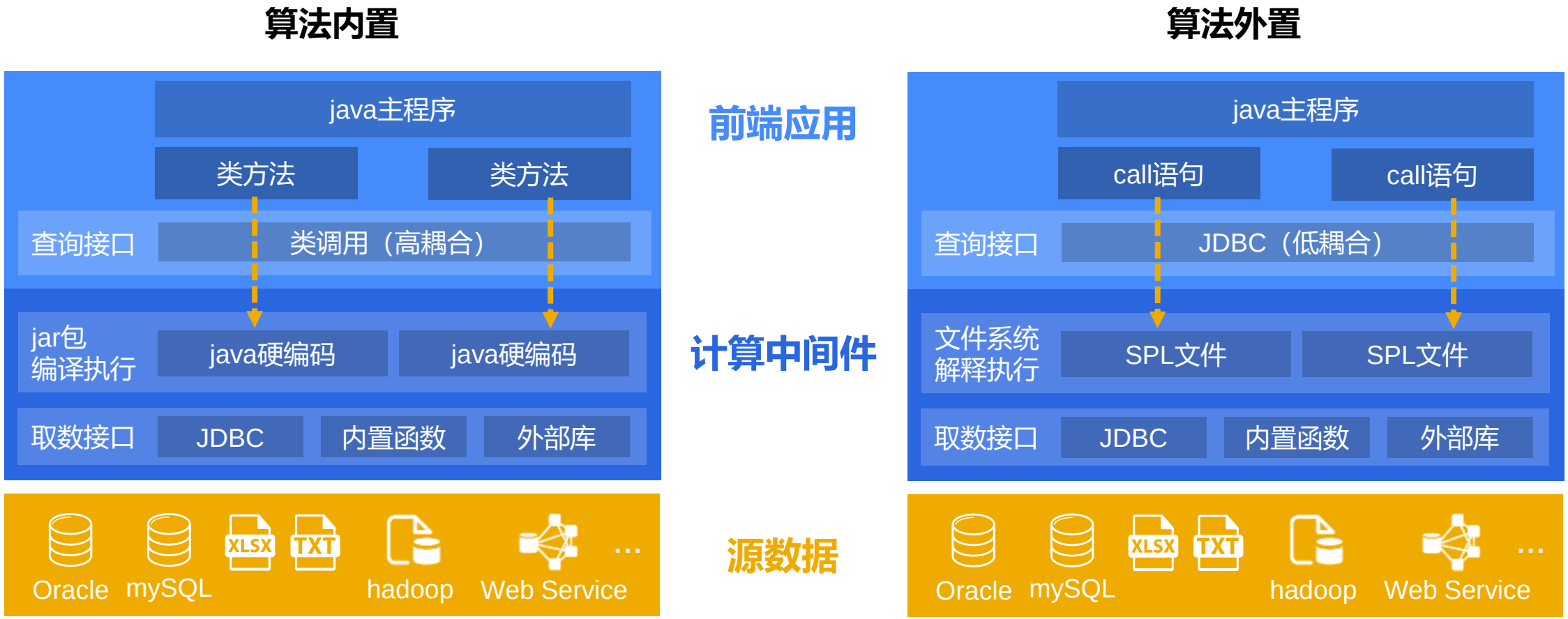
目录 CONTENTS

05

Java算法外置

算法外置

算法外置于文件系统，使用call语句（字符串）降低主程序和算法之间的耦合性



算法外置

热切换：业务算法修改后，直接替换即可，无须编译，无须停机

业务算法queryOrder.dfx原本需要查询文本文件

	A	B
1	=file(“sales.txt”).import@t()	/打开文本文件
2	=A1.select(orderID==pClient)	/返回查询结果，pClient是参数，代表客户编号

将算法改为从Excel查询数据，直接覆盖原算法

	A	B
1	=file(“sales.xlsx”).xlsimport@t()	/打开Excel文件
2	=A1.select(orderID==pClient)	/返回查询结果

json计算

为了彻底降低系统耦合性，有些java应用需要从微服务取数据，这就要求JAVA程序具备计算json的能力

访问微服务，读取多层json，计算每个员工的订单金额，作为员工记录的新字段aMount，结果输出为二维表。源数据json如下：


```
[{
"EID":1,"NAME":"Rebecca","SURNAME":"Moore","GENDER":"F","STATE":"California","BIRTHDAY":"1974-11-20","HIREDATE":"2005-03-
11","DEPT":"R&D","SALARY":7000,
"orders":[{"ORDERID":14,"CLIENT":"JAYB","SELLERID":1,"AMOUNT":7644.0,"ORDERDATE":"2012-11-16 15:28:05"},
        {"ORDERID":77,"CLIENT":"HANAR","SELLERID":1,"AMOUNT":13200.0,"ORDERDATE":"2013-01-17 15:28:05"},
                {"ORDERID":78,"CLIENT":"YZ","SELLERID":1,"AMOUNT":11600.0,"ORDERDATE":"2013-01-20 15:28:05"}
               ]
},{
"EID":2,"NAME":"Ashley","SURNAME":"Wilson","GENDER":"F","STATE":"New York","BIRTHDAY":"1980-07-19","HIREDATE":"2008-03-
16","DEPT":"Finance","SALARY":11000,
"orders":[{"ORDERID":7,"CLIENT":"EGU","SELLERID":2,"AMOUNT":17800.0,"ORDERDATE":"2012-11-06 15:28:05"},
                {"ORDERID":19,"CLIENT":"JOPO","SELLERID":2,"AMOUNT":3430.0,"ORDERDATE":"2012-11-18 15:28:05"},
                {"ORDERID":46,"CLIENT":"UJRNP","SELLERID":2,"AMOUNT":1274.0,"ORDERDATE":"2012-12-20 15:28:05"}
               ]
},{
"EID":3,"NAME":"Rachel","SURNAME":"Johnson","GENDER":"F","STATE":"New Mexico","BIRTHDAY":"1970-12-17","HIREDATE":"2010-12-
01","DEPT":"Sales","SALARY":9000,
"orders":[{"ORDERID":17,"CLIENT":"PJIPE","SELLERID":3,"AMOUNT":7154.0,"ORDERDATE":"2012-11-19 15:28:05"},
 ...
```

json计算

使用集算器外置算法，脚本如下：

	A	B
1	=httpfile("10.0.0.4:8080/recordQuery?comp=cidc").read()	/读取微服务
2	=json(A1)	/将json转为序表
3	=A8.new(EID,NAME,SURNAME,GENDER,DEPT,SALARY,orders.sum(AMOUNT):sAmount)	/计算员工的订单金额

A2的计算结果

	NAME	SURNAME	GENDER	STATE	BIRTHDAY	HIREDATE	DEPT	SALARY	orders
1	Rebecca	ore	F	California	1974-11-20	2005-03-11	R&D	7000	[[14,JAYB,1..
2	Ashley	Wilson	F	New York	1980-07-19	2008-03-16	Finance	11000	[[7,EGU,2, ...
3	Rachel	Johnson	F	New Mexico	1970-12-17	2010-12-01	Sales	9000	[[17,PJIPE,...
4	Emily	Smith	F	Texas	1985-03-07	2006-08-15	HR	7000	[[16,AYWY...
5	Ashley	Smith	F	Texas	1975-05-13	2004-07-30	R&D	16000	[[21,DILRT,..

ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
14	JAYB	1	7644.0	2012-11-16 15:28:05
77	HANAR	1	13200.0	2013-01-17 15:28:05
78	YZ	1	11600.0	2013-01-20 15:28:05

ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
21	DILRT	5	16900.0	2012-11-29 15:28:05
34	HANAR	5	784.0	2012-12-12 15:28:05
43	HANAR	5	196.0	2012-12-15 15:28:05

A3的计算结果

EID	NAME	SURNAME	GENDER	DEPT	SALARY	sAmount
1	Rebecca	Moore	F	R&D	7000	32444.0
2	Ashley	Wilson	F	Finance	11000	22504.0
3	Rachel	Johnson	F	Sales	9000	55154.0
4	Emily	Smith	F	HR	7000	33364.0
5	Ashley	Smith	F	R&D	16000	17880.0

json计算

集算器亦可解析json形式的参数

例子：根据起始日期、终止日期、客户名单查询订单表，参数以json串的形式传递（名为pJson），形如：

{beginDate:2012-01-01, endDate:2012-12-10,
clientList:[{client:UJRNP},{client:UJRNP},{client:PWQ}]
}

用脚本文件实现查询

	A	B	C
1	=json(pJson)		/将json参数转为序表
2	=beginDate=A1.beginDate	=endDate=A1.endDate	/解析起止日期参数，自动转为日期类型
3	=clientList=A1.clientList.(client)		/解析clientList
4	=demo.query("select * from sales where orderdate>=? and orderdate<? and client in(?)",beginDate,endDate,clientList)		/用参数执行SQL

上述算法返回结果为二维表，如果java主程序需要算法返回为json格式，只需在A5单元格编写：

5	=json(A4)		

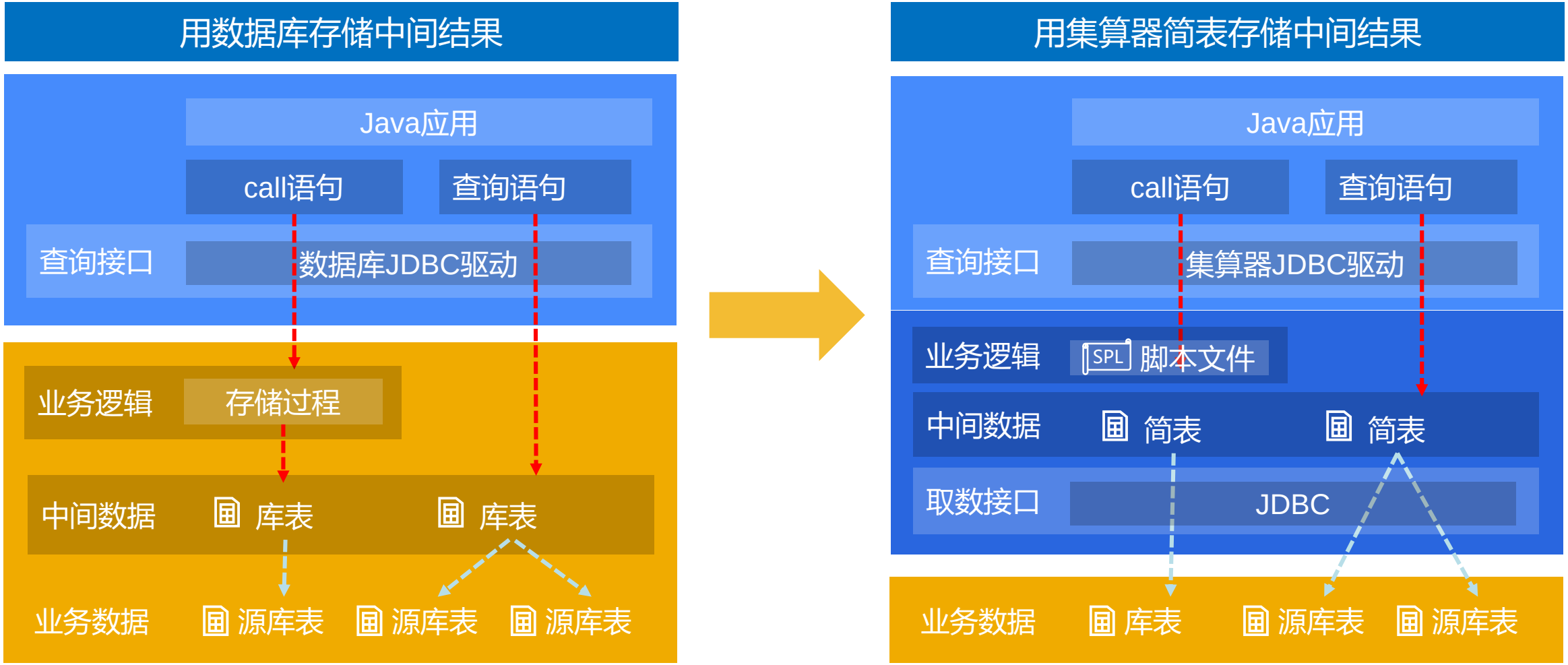
目录 CONTENTS

06

应用数据缓存

应用架构

源数据库压力较大时，通常要事先计算出中间数据，以便应用服务器快速访问。中间数据以前常缓存于源库，使用集算器中间件后，可缓存于应用服务器。



简表和库表

简表：中间数据存储于应用服务器，不耗费数据库资源

库表：中间数据存储于源数据库，仍需耗费数据库资源

	集算器简表	数据库表
计算能力	充足的结构化算法	充足的结构化算法
追加	高性能追加	追加速度较慢
修改	不擅长	擅长
占用空间	小	大
表之间关系	无物理外键和约束关系	可以有物理外键和约束关系
取数的有序性	顺序确定，天然有序	顺序不确定，须order by指定
分段取数	任意分段，方式简单	使用where分段，方式复杂
按组分段	支持	不支持
计算性能	高	低

选取缓存的数据

适合缓存的数据：算法复杂、汇总统计等计算量较大的中间表

不必缓存的数据：计算压力不大的业务表

源数据库	集算器简表
账务平衡表（算法复杂中间表） 预算风险评估（算法复杂中间表） 总账汇总表（汇总中间表） 客户流失统计（汇总中间表） 每月销售情况（汇总中间表） 财务总账（业务表） 预算表（业务表） 订单表（业务表） 客户表（业务表） 销售表（业务表）	-ERP -财务管理 账务平衡表.btx 预算风险评估表.btx 总账汇总表.btx -销售管理 客户流失统计 -每月销售情况.btx -CRM…

说明1：如有必要，也可缓存访问频繁的业务表
说明2：简表存放于操作系统目录，可按功能模块划分目录，或按应用类型、时间、版本划分目录

生成简表

最直接最简单的方式：将源数据库里的中间表导出为简表

源数据里，汇总中间表salesAgg根据业务表sales生成，部分数据如下

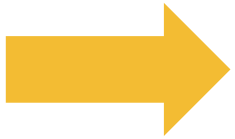
year	month	sellerid	samount	cquantity
2012	11	17	35792	3
2012	11	6	4802	1
2012	11	16	39334	3
2012	11	9	26100	1
2012	11	11	25610	2

将表salesAgg导出为简表

	A	B
1	=connect@l(“db”).cursor(“select * from salesAgg”)	/SQL取数
2	=file(“salesAgg.btx”).export@b(A1)	/创建简表

读取简表后，可以看到其数据与中间表一样

	A	B
1	=file(“salesAgg.btx”).import@b()	/读取简表



year	month	sellerid	samount	cquantity
2012	11	17	35792	3
2012	11	6	4802	1
2012	11	16	39334	3
2012	11	9	26100	1
2012	11	11	25610	2

生成简表

更常见的方式：用SPL脚本改造（替代）原中间表的生成过程，由业务表直接生成简表

用业务表sales直接生成简表

	A	B
1	=connect@l(“db”).cursor(“select year(orderdate) as year,month(orderdate) month as month,sellerid,sum(amount)sAmount,count(1) cQuantity from sales group by year(orderdate),month(orderdate) ,sellerid”)	/SQL取数
2	=file(“salesAgg.btx”).export@b(A1)	/创建简表

增量追加

场景：适用于向历史简表定时追加数据。
注意：源数据须有时间戳，以便取增量。。

每日凌晨从业务表sales取昨日增量数据，并追加到简表salesAgg.btx

	A	B
1	=connect@l(“db”).cursor@x(“select year(orderdate) as year,month(orderdate) month as month,sellerid,sum(amount)sAmount,count(1) cQuantity from sales where orderdate=? group by year(orderdate),month(orderdate), sellerid “, pDate)	按日期取增量数据
2	=file(“salesAgg.btx.btx”).export@ab(A1)	追加到现有简表

说明1：简表无法修改，如数据发生变化，则应重新生成简表。集算器组表用于数据仓库，该文件格式支持修改，详见 http://c.raqsoft.com.cn/article/1570868099462

说明2：取增量的方法较为灵活，可按时间区间取数，也可按日期取数。上面例子里pData是参数，由外部计算出昨日日期并传入，这样可以方便控制要追加哪天的数据。如果要固定抽取昨日数据，也可用SPL表达式date(elapse(now(),-1))代替pDate。

扩展阅读：Oracle有独特的OGG增量追加方式，集算器支持这一用法，详见OGG 增量采集数据入库 http://c.raqsoft.com.cn/article/1567582134531

使用简表

简表支持集算器SQL语法，用法示例如下

查询

```
select * from salesAgg.btx where samount>=10000 && samount<20000
```

聚合

```
select year, month, sum(samuont) as total from salesAgg.btx group by year,month
```

关联

```
select s.year, s.month, e.name from salesAgg.btx s, employee.btx e where s.sellerid=e.eid
```

使用简表

业务逻辑较复杂时，应当使用SPL脚本进行计算

例如：由库存表生成的简表stockAgg.btx，记录着每天每种产品的出库入库数量，部分数据如下

IDATE	INAME	ENTER	ISSUE
2014-04-01	Item1	19	0
2014-04-02	Item1	3	10
2014-04-03	Item1	3	0
2014-04-04	Item1	0	5
2014-04-07	Item1	4	0

按时间段查询简表，计算出如下库存状态：每天每种产品的开库时数量（Open）、入库数量（Enter）、最高库存（Total）、出库数量（Issued）、闭库时数量（Close）

	A	B
1	=fle(“stockAgg.btx”).import@b().select(IDATE>=start &&IDATE<=end)	
2	=A1.group(INAME)	=periods(start,end,1)
3	for A2	=A3.align(B2,IDATE)
4		>c=0
5		=B3.new(A3.INAME,B2(#):IDATE, c:OPENING, ?ENTER,(b=c+ENTER):TOTAL,ISSUE,(c=b-ISSUE):CLOSE)
6		=@|B5
7	return B6	

	A	B	C	D	E	F	G
1	INAME	IDATE	OPENING	ENTER	TOTAL	ISSUE	CLOSE
2	Item1	2014-04-01	0	19	19	0	19
3	Item1	2014-04-02	19	3	22	10	12
4	Item1	2014-04-03	12	3	15	0	15
5	Item1	2014-04-04	15	0	15	5	10
6	Item1	2014-04-05	10		10		10
7	Item1	2014-04-06	10		10		10
8	Item1	2014-04-07	10	4	14	0	14
9	Item1	2014-04-08	14		14		14
10	Item1	2014-04-09	14		14		14

优化存储

生成简表时，还可以根据数据特点进行优化存储，从而达到更好的计算性能。

有序存储：salesAgg.btx经常进行有序计算（比如按年、月分组汇总），则应按分组字段顺序生成简表

	A	B
1	=connect@l(“db”).cursor(“select year(orderdate) as year,month(orderdate) month as month,sellerid,sum(amount)sAmount,count(1) cQuantity from sales group by year(orderdate),month(orderdate) ,sellerid order by year,month”)	/SQL排序
2	=file(“salesAgg.btx”).export@b(A1)	/有序存储

分段存储：salesAgg.btx经常进行分段计算（比如并行查询），则应进行分段存储

	A	B
1	=connect@l(“db”).cursor(“select year(orderdate) as year,month(orderdate) month as month,sellerid,sum(amount)sAmount,count(1) cQuantity from sales group by year(orderdate),month(orderdate) ,sellerid”)	/SQL排序
2	=file(“salesAgg.btx”).export@z(A1)	/分段

有序分段：已知用某字段做分段计算时，可以按该字段排序并分段存储

	A	B
1	=connect@l(“db”).cursor(“select year(orderdate) as year,month(orderdate) month as month,sellerid,sum(amount)sAmount,count(1) cQuantity from sales group by year(orderdate),month(orderdate) ,sellerid order by sellerid”)	/SQL排序
2	=file(“salesAgg.btx”).export@z(A1;sellerid)	/有序分段

SQL+语句

如果已知简表的数据特征，就可以使用SQL+语法进行高性能优化查询。

并行计算	select /*+**parallel** (4) */ * from sales.txt where orderid=100
大表游标查询	select * from **/*+external*/** emp.btx where orderid=100
有序外表关联	select o.Orderid ,o.amount,s.name from sales.btx o **/*+foreign*/** join seller.btx s on s.id = o.sellerid

说明：使用SQL+语句，url要写成jdbc:esproc:local://sqlfirst=plus。

扩展阅读：SQL+ http://doc.raqsoft.com.cn/esproc/func/sqljia.html

生命周期

缓存有三种生命周期：定时缓存、临时缓存、可控缓存

定时缓存：事先生成缓存，供多个业务算法使用

数据库表sales访问频繁，多个算法均会用到，现将其缓存为简表，每周一凌晨可执行如下脚本文件

	A	B
1	=connect@l(“db”).query@x(“select orderid,client,sellerid,amount,orderdate from sales order by eid “)	查询数据
2	=file(“sales.btx”).export@z(A1)	生成简表

业务算法A：按client,sellerid分组汇总

	A	B
1	=file(“sales.btx”).cursor@b(A1)	打开简表
2	=A2.groups(client,sellerid;sum(amount):sAmount,count(1):cOrderid)	分组汇总

业务算法B：按orderid查询记录

	A	B
1	=file(“salse.btx”)	打开简表
2	=A2.iselect@b(10,orderid; orderid,client,amount)	有序查询

生命周期

临时缓存：计算前临时生成缓存，一般供当前算法在本地反复使用，以减少数据库的频繁查询动作

前端应用需取大表数据用于分页展示，这种情况下可临时生成简表，利用简表的I/O和有序取数大幅提高性能。为实现此目标，前端须传入当前算法的唯一标识，以便每次取数时使用同一个简表文件，还需传入记录的起止位置（可转换为页号），以便每次取不同的页。

	A	B	C
1	=file(uuid)		以当前算法唯一标识为简表名
2	if !A1.exists()	=connect@l("dbsource").cursor@x("select * from sales order by orderid")	如果简表不存在，则生成
3		=A1.export@z(B2)	
4	=A1.iselect@b(begin:end,orderid; orderid,client,sellerid,orderdate,amount)		如果简表存在，则分页取数

说明：临时缓存可存储于集算器指定的临时目录，该目录下的文件会定时自动清理

生命周期

可控缓存：由业务算法掌握缓存精确的生命周期，供当前算法或其他算法使用。

对sales表进行分组汇计算时，先检查缓存的时效性是否在一小时内，如果满足时效性，则直接用缓存高速计算；如果时效已过，则临时生成最新缓存。本算法和其他算法都可使用该缓存。

	A	B	C
1	=file（“sales”）		简表句柄
2	if interval@s(A1.date(),now())>=3600 || !A1.exists()	=connect@l("dbsource").cursor@x("select * from sales order by orderid")	如果简表超时或不存在，则生成
3		=A1.export@z(B2)	
4	=A1.groups(client,sellerid;sum(amount):sAmount)		如果简表存在，则分组汇总

其他注意事项

追求更高并发量、更大的计算量、更大的数据量（包括数据压缩）、高性能索引查询、修改数据等特性，可使用集算器组表格式。详情参考《轻量级高性能文件型数据仓库》
http://c.raqsoft.com.cn/article/1570868099462

定时执行生成缓存的SPL脚本（包括可视化工具的ept脚本），可使用操作系统自带调度工具，如计划任务、Crontab，或第三方可视化工具如opencron。

目录 CONTENTS

07

多源混算方法

基本方法

多源混算是计算中间件的主要用法之一，集算器使用高度封装的接口，可简化任意数据源之间的混合计算。

员工表位于Oracle，绩效位于mysql，请跨库关联两表，并算出实际工资

employee(oracle)

key	empID
	empName
	baseSalary
	post
	dept

performance(mysql)

key	empID
	bonus

输出结果

key	empID
	empName
	realSalary
	post
	dept

	A	B
1	=connect(“orcl”).cursor@x(“select * from employee”)	/绩效
2	=connect(“mysql”).query@x(“select * from performance”)	/员工
3	=A1.switch(empID,A2:empID)	/跨库关联
4	=A3.new(empID.empID:empID,empName,baseSalay+empID.bonus:realSalary,post,dept)	/计算实际工资

基本方法

集算器使用统一数据模型访问数据源，可使用相同的方法混算不同的数据源

员工表位于Oracle，绩效位于文本文件，请跨源关联两表，并算出实际工资

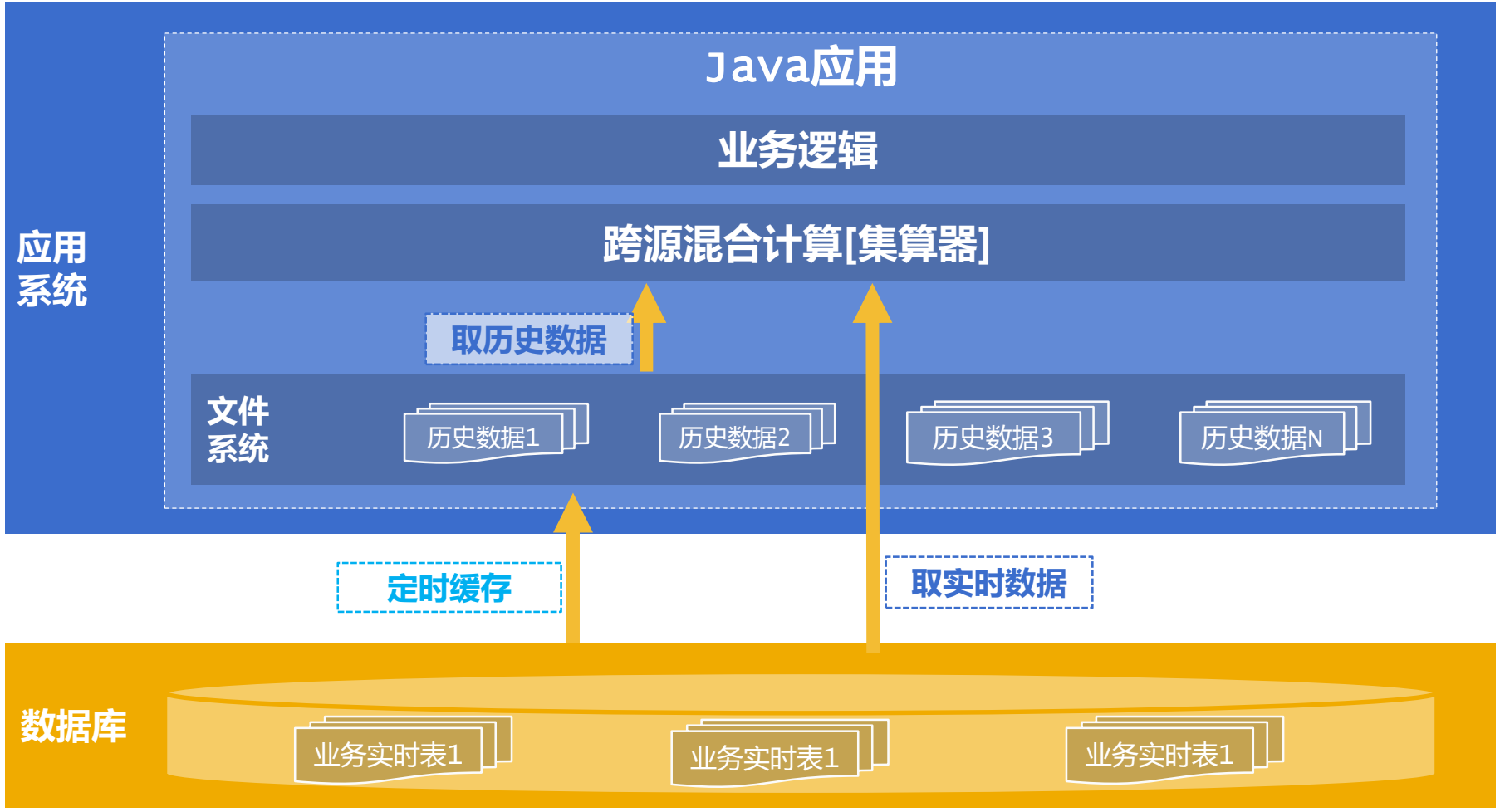
	A	B
1	=connect(“orcl”).cursor@x(“select * from employee”)	/绩效
2	=file(“performance.txt”).import@t()	/员工
3	=A1.switch(empID,A2:empID)	/跨库关联
4	=A3.new(empID.empID:empID,empName,baseSalay+empID.bonus:realSalary,post)	/计算实际工资

员工位于简表，绩效位于文本文件，请跨源关联两表，并算出各部门实际支出的工资成本

	A	B
1	=file(“employee.btx”).cursor@b()	/绩效
2	=file(“performance.txt”).import@t()	/员工
3	=A1.switch(empID,A2:empID)	/跨库关联
4	=A3.groups(dept; sum(baseSalay+empID.bonus):realSalary)	/各部门实际工资

T+0混算

历史数据T用简表（量大用组表）存放，实时数据0用生产数据库存放，将二者实时混合计算，可实现高性能全量数据查询，同时节省宝贵的生产库资源



T+0混算

订单历史存储在简表，实时订单数据存储在生产库。请统计目前为止每个客户的订单数。

	A	B
1	=connect（“db”）.cursor@x（“select customer,count(1) total from sales where orderDate=?”,now() ）	/查询数据库当日订单
2	=connect().cursor@x（“select customer,count(1) total from sales.btx”）	/查询简表历史订单
3	=[A1,A2].conjx().groups(customer,sum(total):total)	/二次分组汇总

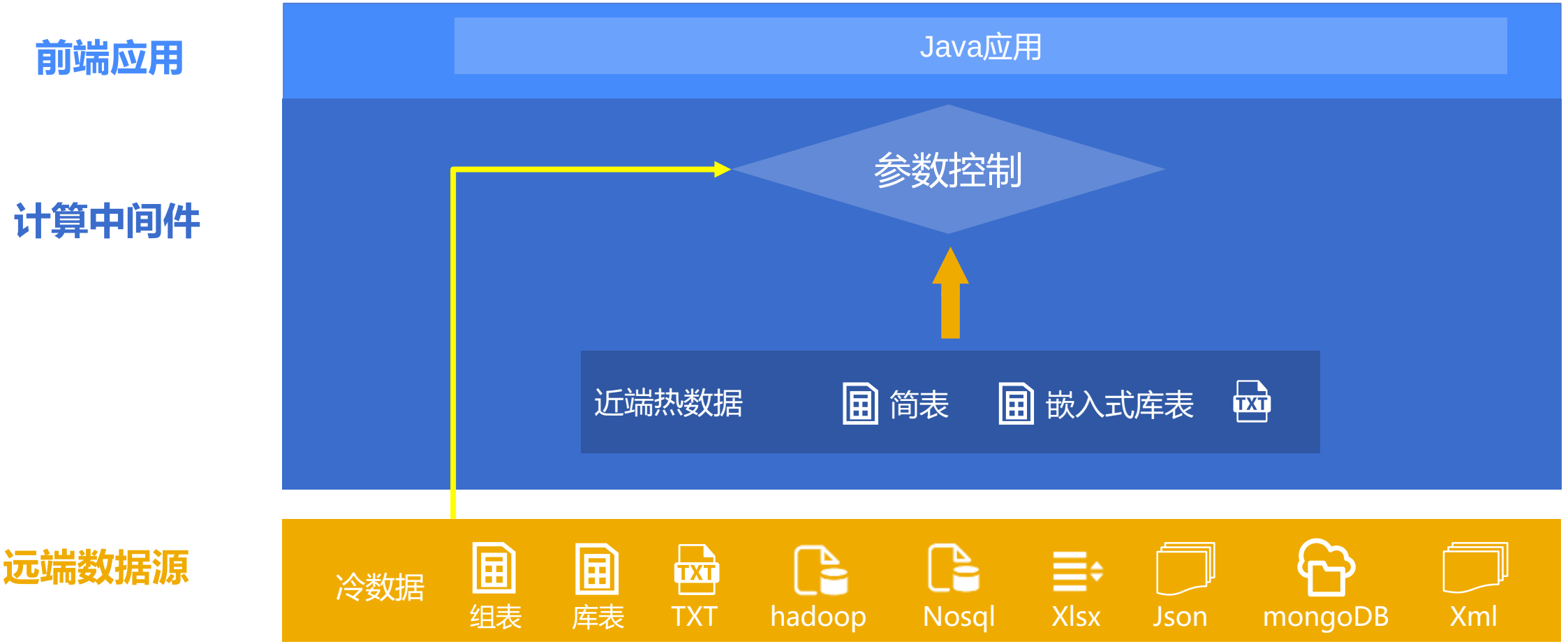
扩展阅读：关于T+0实时混算更全面更细节的情况请参考

《分库后的统计查询》http://c.raqsoft.com.cn/article/1567657792653

《实时报表 T+0 的实现方案》http://c.raqsoft.com.cn/article/1541494770016

冷热路由

频繁访问的热数据可缓存于应用服务器近端，偶尔访问的冷数据可存储于远端，计算时通过判断参数区间，来决定返回冷数据还是热数据。



冷热路由

当年（2016年）热数据以简表的形式存储在应用服务器，往年（2016年以前）的冷数据存储在RDB数据仓库中。现由getDate.dfx实现冷热路由,并实现分组汇总算法。上层JAVA程序调用时需传入参数pBeginDate、pEndDate，即所需数据的起止时间范围。

	A	B
1	=hotcoldLine=date("2016-01-01")	/冷热分界点
2	=file(“saleshot.btx”).iselect@b(max(pBeginDate,hotcoldLine):max(pEndDate,hotcoldLine), orderdate; orderid,client,sellerid,amount,orderdate)	/取热数据
3	=connect@l("demo").cursor@x("select * from sales where orderdate>=? and orderdate<?",min(pBeginDate,hotcoldLine),min(pEndDate,hotcoldLine))	/取冷数据
4	=A2|A3	/冷热合并，其一可能为空
5	=A4.groups(year(orderdate),month(orderdate);sum(amount):sAmount)	/计算

说明：冷数据应按orderDate字段建数据库索引，组表可按orderdate排序分段，以便为空时快速返回。路由的过程可单独分离成公用脚本，供具体脚本调用，以降低耦合性。

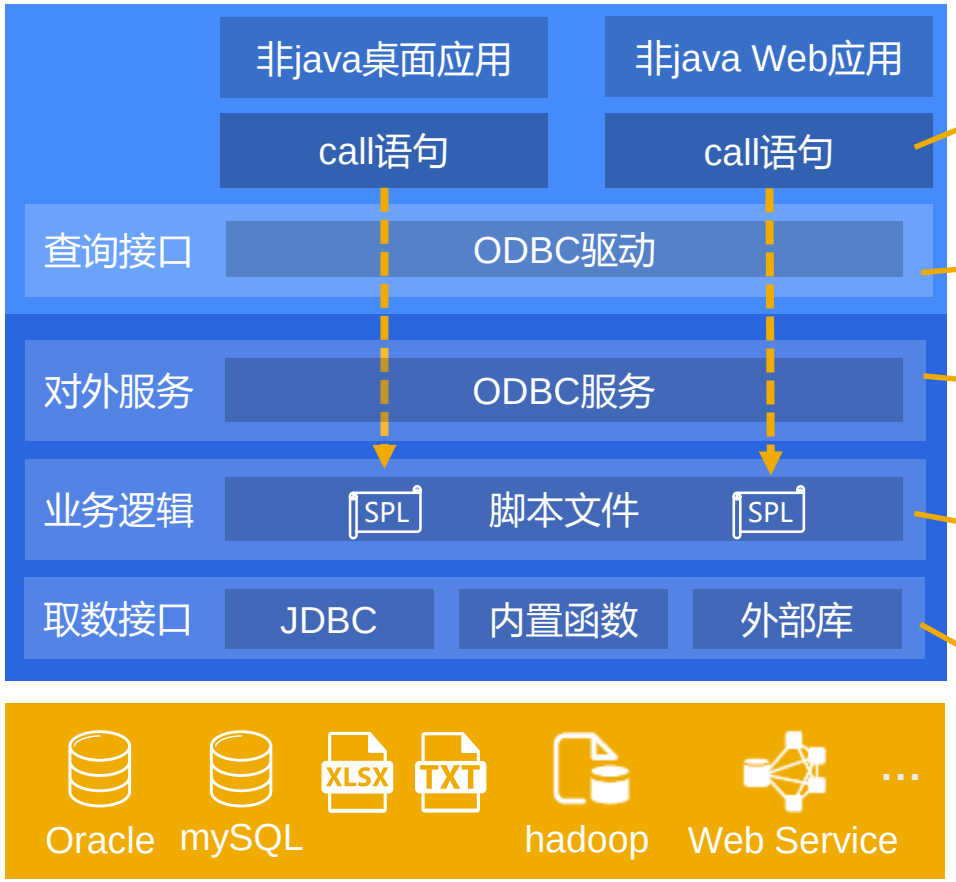
目录 CONTENTS

08

ODBC与HTTP集成

ODBC集成

除了JDBC接口，集算器也提供了ODBC服务，以支持非JAVA的前端应用，比如VB、C#、ASP.net、水晶报表。



5.在非java代码中执行call语句，通过集算器ODBC调用SPL文件，集算器ODBC解释执行SPL文件，并返回计算结果

3.部署集算器ODBC驱动

2.启动集算器ODBC服务

4.在SPL文件中编写具体的业务算法，无变化

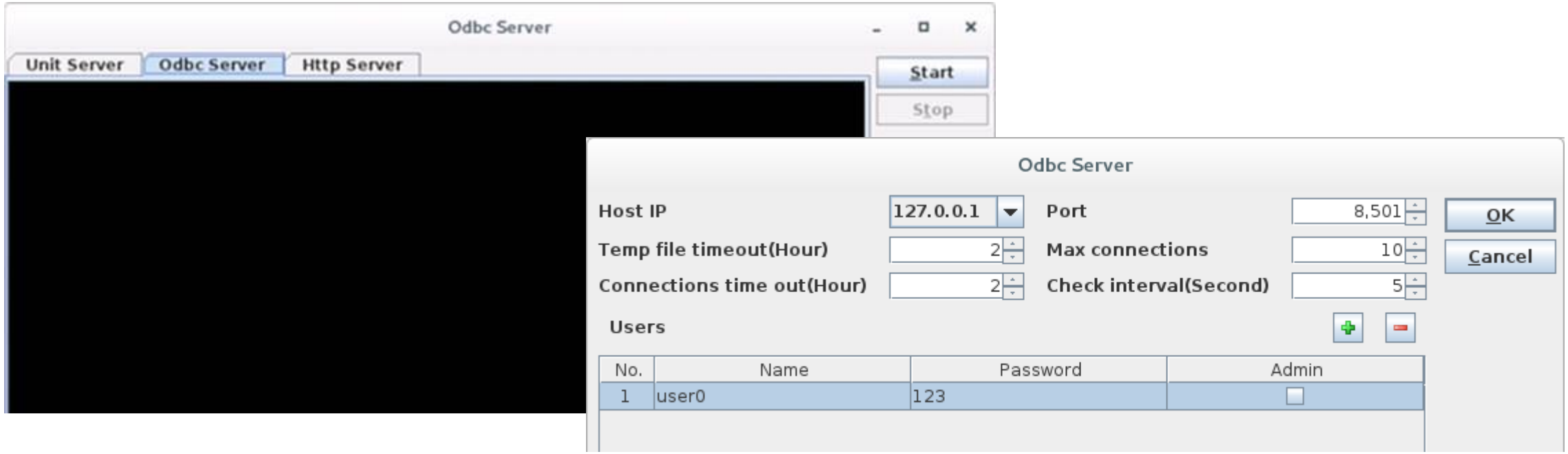
1.在raqsoftConfig.xml中配置运行环境，无变化。

ODBC集成

1.在raqsoftConfig.xml中配置运行环境，这一步无任何变化

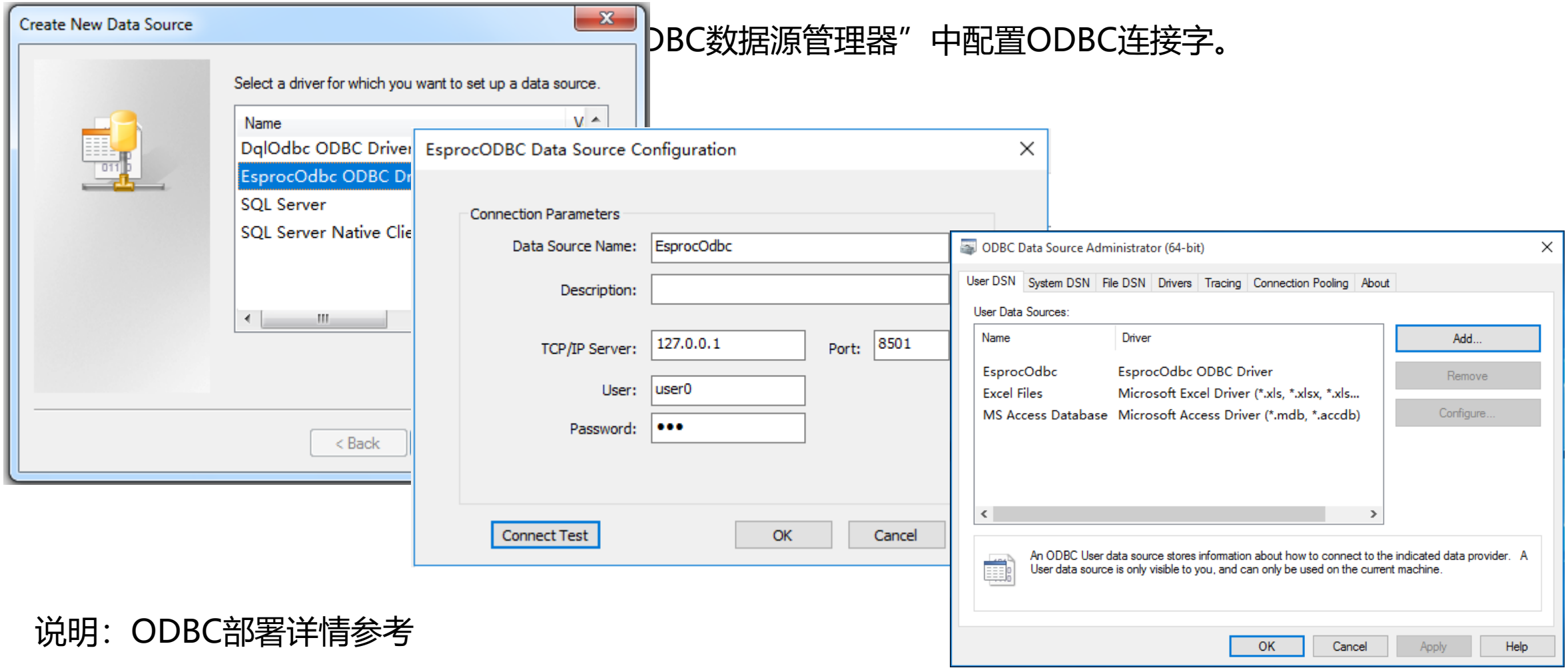
2.启动集算器ODBC服务，这一步是主要的变化

集算器提供了图形化界面，用来启动ODBC服务，详见 http://doc.raqsoft.com.cn/esproc/tutorial/jsqzqdodbcfw.html



ODBC集成

3.部署集算器ODBC驱动。先用管理员权限执行集算器自带的esprocOdbcinst.exe，以自动部署



DBC数据源管理器”中配置ODBC连接字。

说明：ODBC部署详情参考

http://doc.raqsoft.com.cn/esproc/tutorial/odbcbushu.html#ODBC%E9%83%A8%E7%BD%B2

ODBC集成

4.在SPL文件中编写具体的业务算法，这一步没有任何变化。

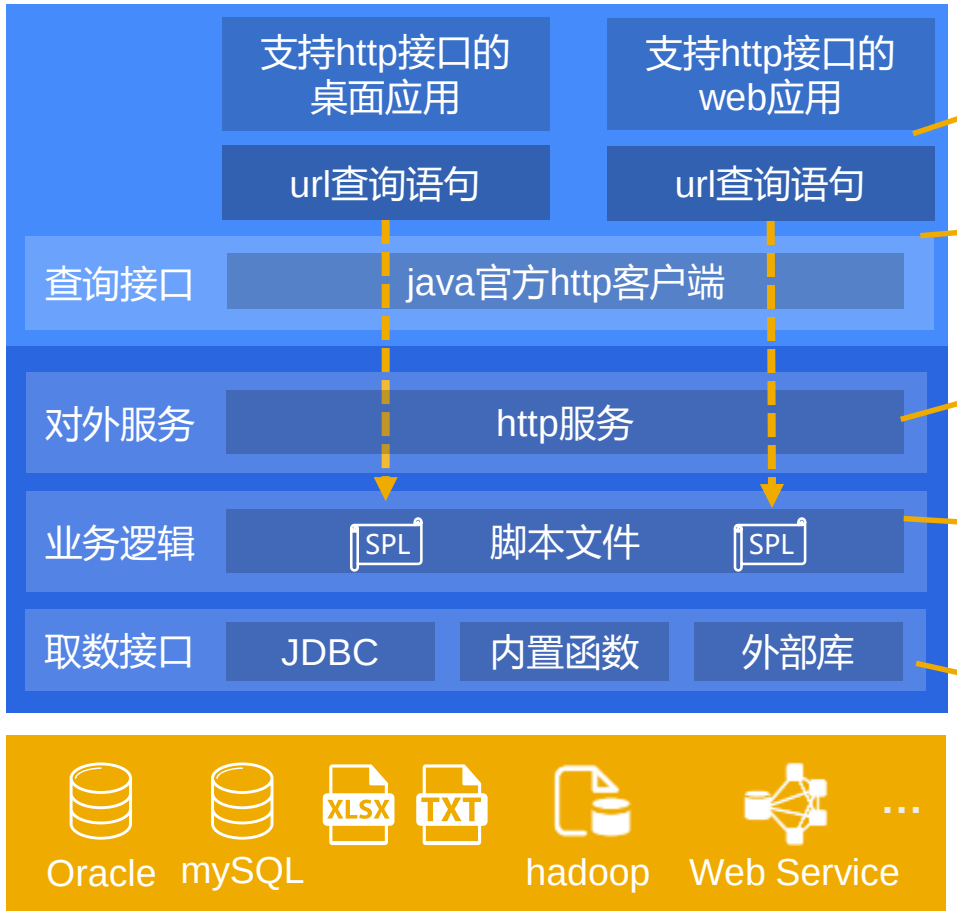
5.在非java代码中执行call语句，这一步只需按标准的ODBC规范编写代码即可。

例如：在ASP.net代码中，通过odbc访问conj.dfx脚本文件

```
……
   OdbcConnection odbcConn = new OdbcConnection(“DSN=testOdbc;”);
   odbcConn.Open();
   OdbcCommand odbcCmd = new OdbcCommand(“call conj(?,?)", odbcConn);
   odbcCmd.Parameters.Add("minamount", OdbcType.Int).Value = 4000;
   odbcCmd.Parameters.Add("maxamount", OdbcType.Int).Value = 8000;
  ……
```

http集成

为了对外提供松耦合的数据服务(比如数据中台），集算器提供了http服务。需要注意的是，这种情况下通常要求独立部署集算器。



5.使用任意语言，通过http协议访问SPL文件，集算器http服务解释执行SPL文件，并返回计算结果

3.部署java官方的http客户端jar包。

2.启动集算器http服务

4.在SPL文件中编写具体的业务算法，存储于文件系统。

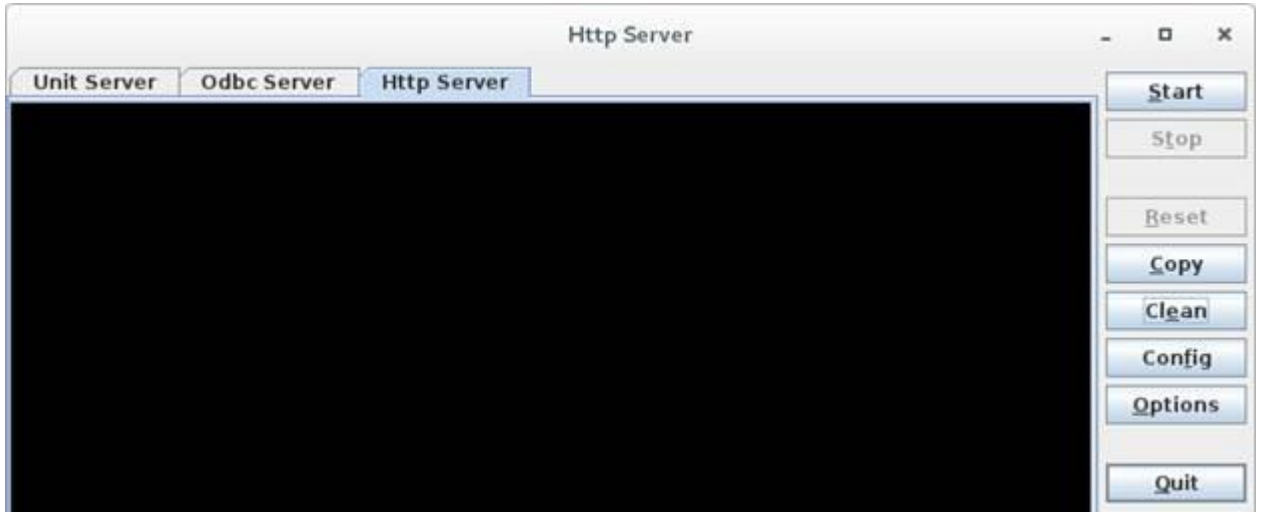
1.在raqsoftConfig.xml中配置运行环境，无变化

http集成

1.在raqsoftConfig.xml中配置运行环境，这一步无任何变化

2.启动集算器http服务，这一步是主要的变化

集算器提供了图形化界面，用来启动http服务，详见http://doc.raqsoft.com.cn/esproc/tutorial/httpfuwu.html



http集成

3.部署java官方的http客户端jar包，这一步与具体http服务无关，请参照java官方标准。

4.在SPL文件中编写具体的业务算法，这一步算法本身无变化，只需处理返回数据的格式。

原算法（非http服务）查询sales表，beginDate,endDate是外部参数，则脚本文件如下：

	A	B	C
1	=connect@l(“demo”).query@x("select * from sales where orderdate>=? and orderdate<?",beginDate,endDate)		/查询数据库

如果希望集算器提供restful json服务，只需用json函数返回结果

	A	B	C
1	=connect@l(“demo”).query@x("select * from sales where orderdate>=? and orderdate<?",beginDate,endDate)		/查询数据库
2	=json(A1)		

用浏览器访问集算器http服务，可看到如下数据结构

说明：想返回xml格式，则应使用xml函数。如果不处理返回结果，则http客户端会接收到二维表格式的字符串，默认情况下以tab分隔。

http集成

对于java中间件而言，返回多层json比较麻烦，而用集算器则相对简单

将销售人员和其订单关联起来，形成多层json并返回

	A	B
1	=connect@l("dbsource")	
2	=A1.query("select * from employee")	=A1.query@x("select * from sales where orderdate>=? and orderdate<?",beginDate,endDate)
3	=A2.run(eid=B2.select(sellerid==eid))	/关联
4	=json(A3)	

在浏览器中查看返回结果


```
[{"eid":[{"orderid":14,"client":"JAYB","sellerid":1,"amount":7644.0,"orderdate":"2012-11-16 15:28:05"},{"orderid":77,"client":"HANAR","sellerid":1,"amount":13200.0,"orderdate":"2013-01-17 15:28:05"},
{"orderid":78,"client":"YZ","sellerid":1,"amount":11600.0,"orderdate":"2013-01-20 15:28:05"},{"orderid":93,"client":"AVU","sellerid":1,"amount":21800.0,"orderdate":"2013-02-05 15:28:05"},
{"orderid":104,"client":"HL","sellerid":1,"amount":26400.0,"orderdate":"2013-02-18 15:28:05"},{"orderid":109,"client":"PWQ","sellerid":1,"amount":17500.0,"orderdate":"2013-02-21 15:28:05"},
{"orderid":120,"client":"FHYBR","sellerid":1,"amount":16000.0,"orderdate":"2013-03-03 15:28:05"},{"orderid":127,"client":"HP","sellerid":1,"amount":13600.0,"orderdate":"2013-03-15 15:28:05"},
{"orderid":189,"client":"DNEDL","sellerid":1,"amount":26100.0,"orderdate":"2013-05-13 15:28:05"},{"orderid":200,"client":"EGU","sellerid":1,"amount":14000.0,"orderdate":"2013-05-20 15:28:05"},
{"orderid":201,"client":"DNEDL","sellerid":1,"amount":7350.0,"orderdate":"2013-05-25 15:28:05"},{"orderid":221,"client":"EGU","sellerid":1,"amount":7742.0,"orderdate":"2013-06-14 15:28:05"},
{"orderid":237,"client":"PWQ","sellerid":1,"amount":23400.0,"orderdate":"2013-06-24 15:28:05"},{"orderid":267,"client":"QUICK","sellerid":1,"amount":17400.0,"orderdate":"2013-07-28 15:28:05"},
{"orderid":278,"client":"FHYBR","sellerid":1,"amount":23900.0,"orderdate":"2013-08-12 15:28:05"},{"orderid":279,"client":"MIP","sellerid":1,"amount":5880.0,"orderdate":"2013-08-12 15:28:05"},
{"orderid":288,"client":"UJRNP","sellerid":1,"amount":4998.0,"orderdate":"2013-08-22 15:28:05"},{"orderid":351,"client":"SAVEA","sellerid":1,"amount":6958.0,"orderdate":"2013-10-19 15:28:05"},
{"orderid":380,"client":"JAYB","sellerid":1,"amount":13000.0,"orderdate":"2013-11-17 15:28:05"},{"orderid":384,"client":"AYWYN","sellerid":1,"amount":22200.0,"orderdate":"2013-11-20 15:28:05"},
{"orderid":401,"client":"DILRT","sellerid":1,"amount":20200.0,"orderdate":"2013-12-08 15:28:05"},{"orderid":461,"client":"EGU","sellerid":1,"amount":5880.0,"orderdate":"2014-02-04 15:28:05"},
{"orderid":465,"client":"AYWYN","sellerid":1,"amount":22400.0,"orderdate":"2014-02-11 15:28:05"},{"orderid":468,"client":"UJRNP","sellerid":1,"amount":15500.0,"orderdate":"2014-02-18 15:28:05"},
```

http集成

5.通过http协议访问SPL文件，这一步与具体http服务无关。需要注意的是，集算器http服务支持两种url风格。

默认风格，http:///*IP:port* /dfx1.dfx(arg1,arg2,...)。java通过http协议调用集算器的部分代码如下：

```
……
   URL url = new URL("http://192.168.1.107:8503/getsales.dfx(2010-03-01,2019-04-01)");
   HttpURLConnection httpUrlConn = (HttpURLConnection) url.openConnection();
 ……
```

SAP风格，http:///*IP:port* /sapPath/dfx/arg1/arg2/...。调用代码如下

```
……
   URL url = new URL(“http://192.168.1.107:8503/getsales/2010-03-01/2019-04-01)");
   HttpURLConnection httpUrlConn = (HttpURLConnection) url.openConnection();
 ……
```

润乾 RAQSOFT

想要了解更多 请联系我们

技术内容请移步 乾学院

http://c.raqsoft.com.cn

优惠价购买请加入 好多乾

http://sys.misdiy.com/hdq.html

www.raqsoft.com.cn