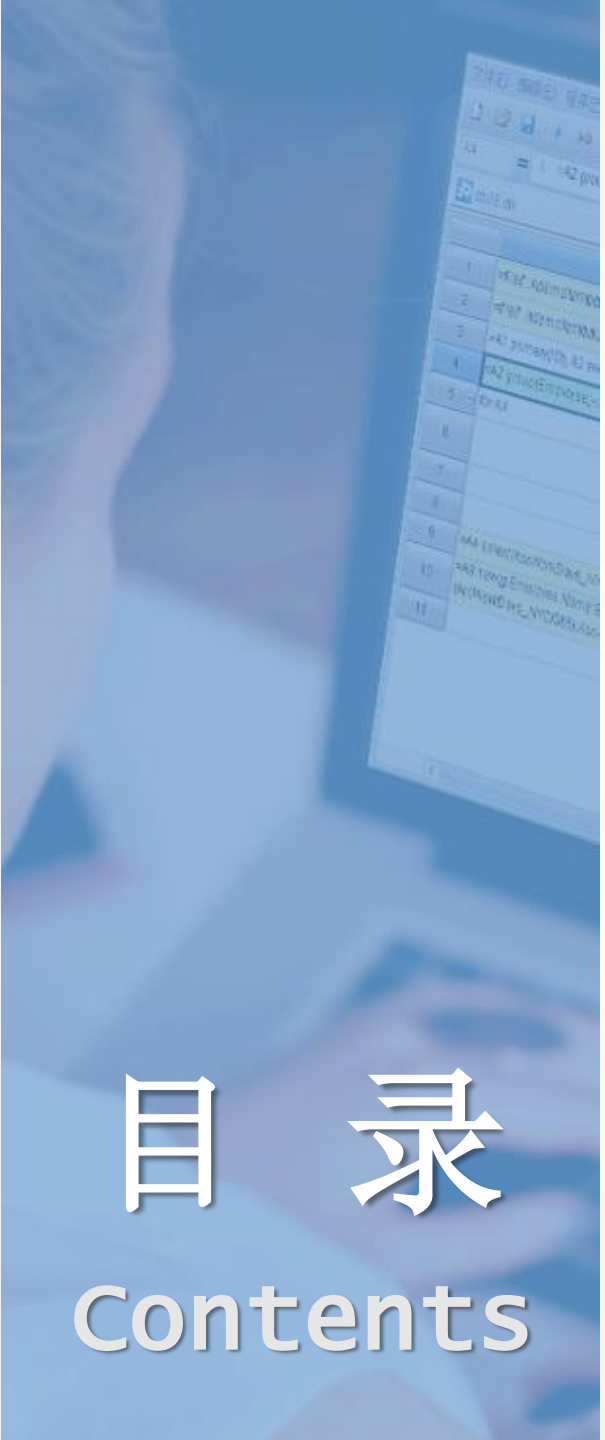
集算器

创新大数据计算引擎

报表后端的计算神器

润乾软件出品



目　录

Contents

1 报表与计算层

2 多样性数据源

3 开发体系结构

4 性能与容量

报表

网格式报表

合同号	产品号	单价	数量	数量单位	合同金额
quiee0010	4	￥1200.00	5.0	吨	￥6100.00
quiee0010	5	￥2400.00	3.0	吨	￥7200.00
quiee0011	1	￥2300.00	5.0	吨	￥11500.00
quiee0011	2	￥3200.00	10.0	吨	￥32000.00
quiee0012	5	￥2300.00	7.0	吨	￥16100.00
quiee0012	6	￥1500.00	5.0	吨	￥7500.00
quiee0013	1	￥23.50	2000.0	吨	￥14700.00
quiee0013	2	￥300.00	20.0	吨	￥6000.00

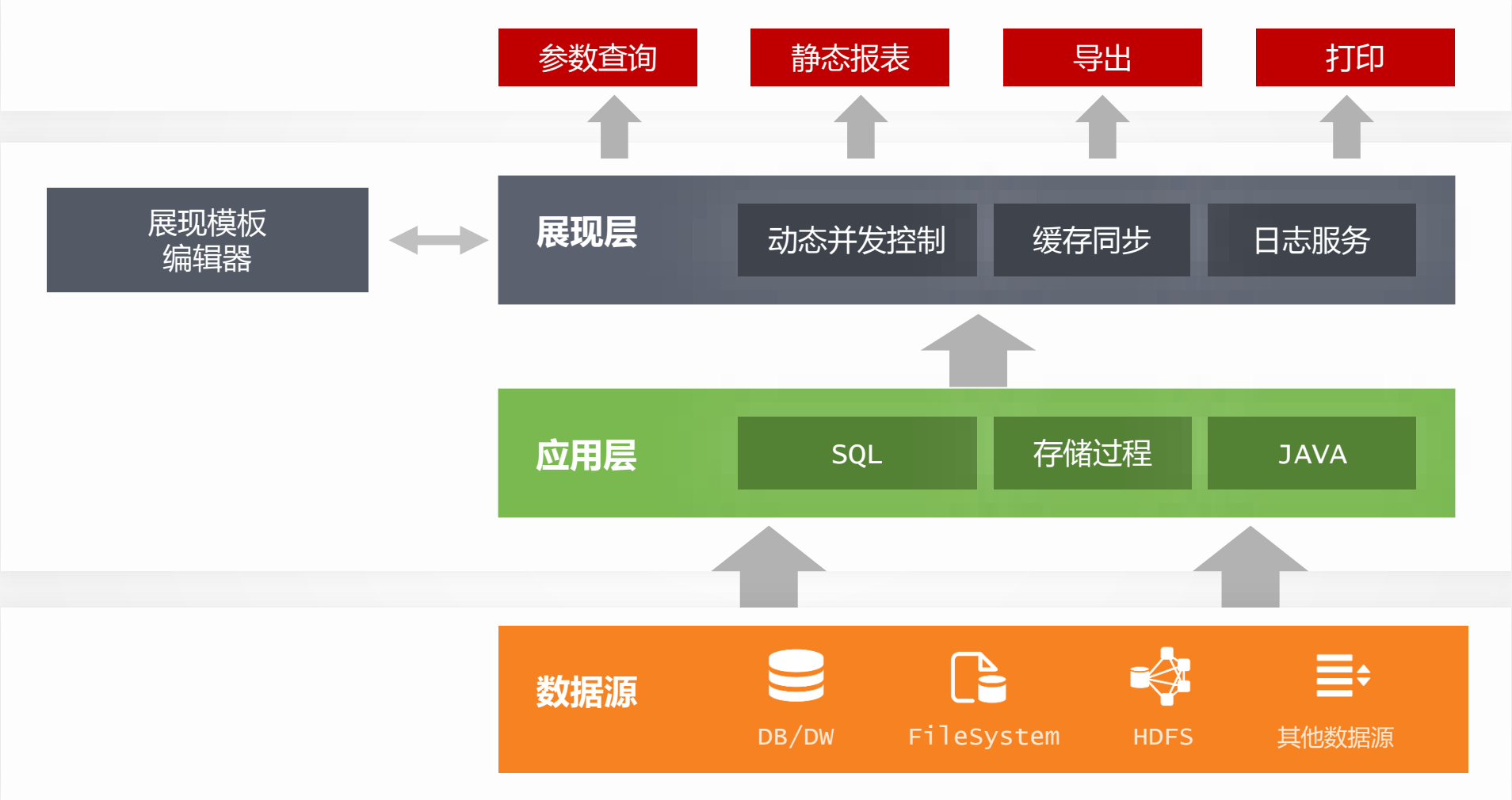
打印出来的就是报表！

报表是具有一定格式、包含一定信息的载体！

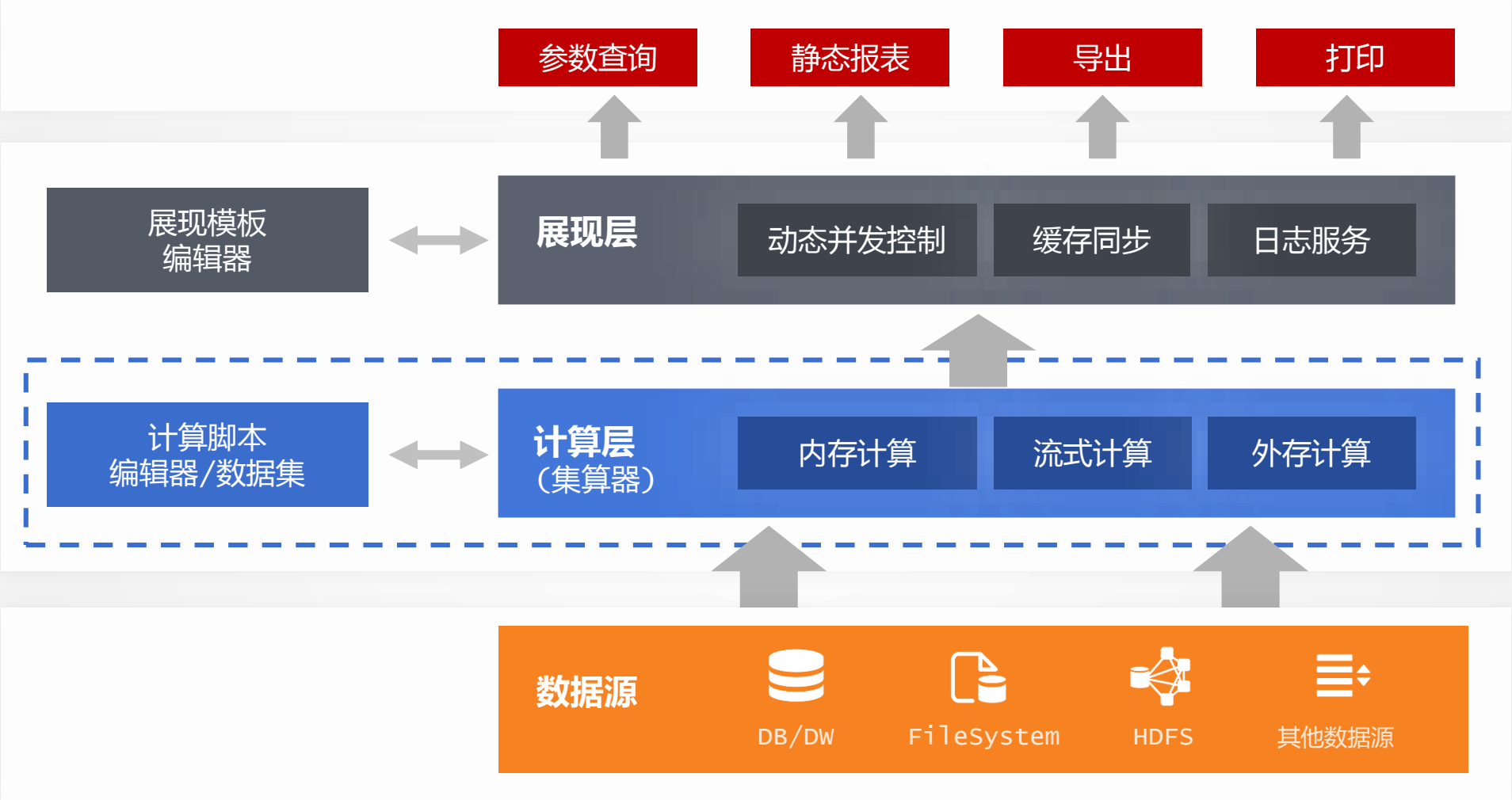
交叉报表

产品 / 销量 / 客户	大豆	色拉油	苯乙酮	苯甲酸钠	苯磺酸	对氯二苯
无锡食品加工厂	￥23.00	￥45.00	￥56.00	￥345.00	￥125.00	￥68.00
天津长城化工厂	￥310.00	￥48.00	￥119.00	￥213.00	￥64.00	￥51.00
江苏天龙食品集团	￥66.00	￥167.00	￥42.00	￥55.00	￥97.00	￥105.00
中国化工原料厂	￥412.00	￥16.00	￥21.00	￥34.00	￥53.00	￥14.00
Crystal	￥27.00	￥32.00	￥61.00	￥18.00	￥79.00	￥31.00
Chemist	￥42.00	￥71.00	￥100.00	￥108.00	￥89.00	￥42.00
Murli	￥68.00	￥56.00	￥62.00	￥87.00	￥112.00	￥261.00
Chemical Mumai	￥287.00	￥179.00	￥91.00	￥342.00	￥63.00	￥41.00

传统模式


参数查询
静态报表
导出
打印
展现模板
编辑器
展现层
动态并发控制
缓存同步
日志服务
应用层
SQL
存储过程
JAVA
数据源
DB/DW
FileSystem
HDFS
其他数据源

加入计算层


参数查询
静态报表
导出
打印
展现模板
编辑器
展现层
动态并发控制
缓存同步
日志服务
计算脚本
编辑器/数据集
计算层
（集算器）
内存计算
流式计算
外存计算
数据源
DB/DW
FileSystem
HDFS
其他数据源

困难计算

跨行计算

举例：按排名顺序呈现某一年各月份的销售额，并计算名次之间的差值，以及当前月份比上个月的增长率。

	A
1	=myDB1.query("select month(OrderDate) month,sum(Amount) amount from salesAll where year(OrderDate)=? group by month order by month",theYear)
2	=A1.derive((amount-amount[-1])/amount[-1]:LRR)
3	=A2.sort(-amount)
4	=A3.derive(amount-amount[1]:DIFF)
5	result A4

month	amount	LRR	DIFF
12	328556.0	0.543313448	47782.0
4	280774.0	0.173804347	7134.0
10	273640.0	0.124623433	9263.0
7	264377.0	0.206159980	20663.0
1	243714.0		397.0
9	243317.0	0.020227008	4117.0
3	239200.0	0.102197483	707.0
8	238493.0	-0.097905642	19304.0
6	219189.0	0.020399707	2168.0
2	217021.0	-0.109525919	2214.0
5	214807.0	-0.234946968	1917.0
11	212890.0	-0.222007016	212890.0

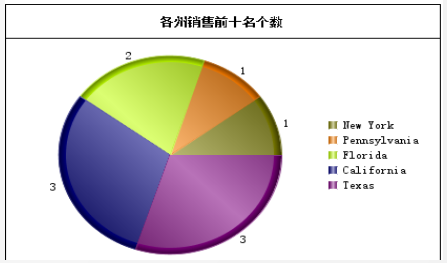
结果集复用

举例：销售额排名、地图分布饼图报表

	A
1	=connect("db2")
2	=A1.query("select * from (select e.eid 雇员编号,max(e.name) 姓名,max(e.dept) 州,sum(amount) 订单总价 from sales s,employee e where e.eid=s.sellerid group by e.eid order by 订单总价 desc) fetch first 10 rows only")
3	=A1.close()
4	=A2.groups(州;count(雇员编号):前十名个数).sort(前十名个数)
5	result A2,A4

销售前十名列表

雇员编号	姓名	州	订单总价
8	Megan	California	$33,627.60
13	Daniel	Florida	$27,259.00
5	Ashley	Texas	$26,942.40
27	Alexis	California	$19,066.60
2	Ashley	New York	$14,508.10
11	Jacob	Texas	$14,452.90
4	Emily	Texas	$14,128.60
26	Timothy	Florida	$12,410.20
29	Olivia	California	$12,129.90
19	Samantha	Pennsylvania	$10,276.10



特殊格式

解决报表工具不支持的格式

员工号	姓名	部门
1	Rebecca	R&D
4	Emily	HR
7	Alexis	Sales
10	Ryan	R&D
13	Daniel	Finance
16	Christopher	Production

员工号	姓名	部门
2	Ashley	Finance
5	Ashley	R&D
8	Megan	Marketing
11	Jacob	Sales
14	Alyssa	Sales
17	Hannah	Marketing

员工号	姓名	部门
3	Rachel	Sales
6	Matthew	Sales
9	Victoria	HR
12	Jessica	Sales
15	Alexis	Sales
18	Jonathan	Administration

举例：横向分栏

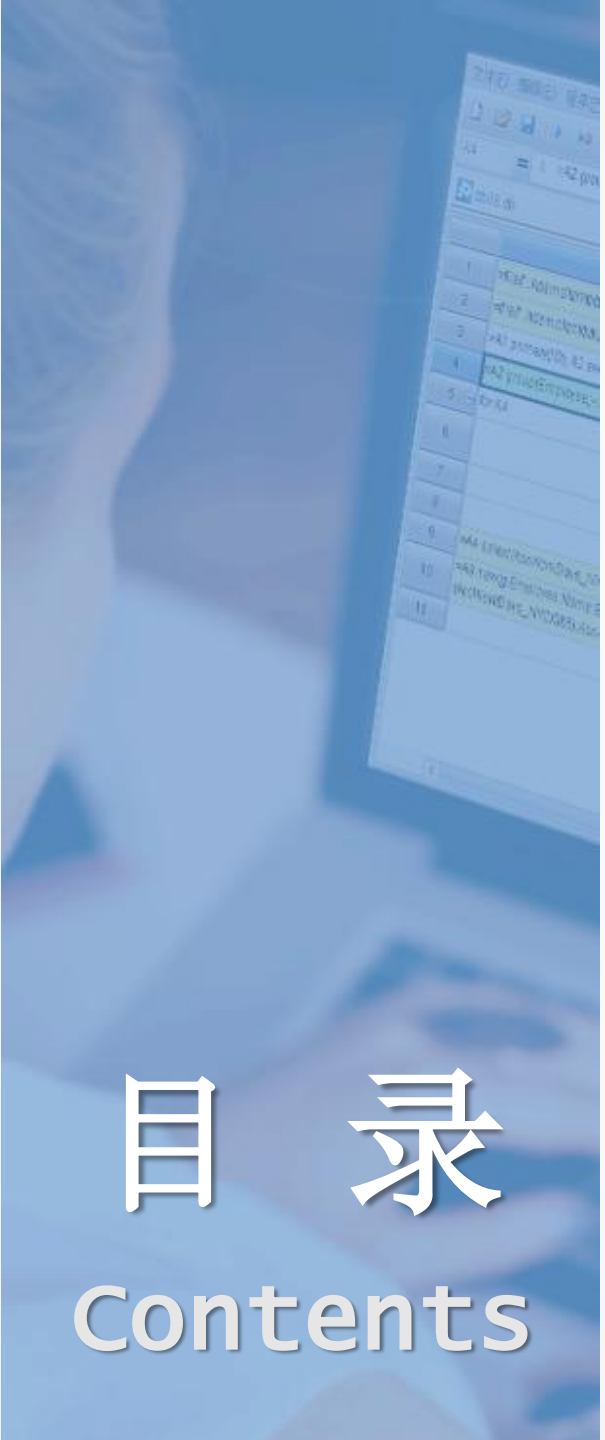
	A	B	C
1	=DB.query("select EId,Name,Dept from emp where EId>=? and EId<=? order by EId ",begin,end)		
2	=A1.step(3,1)	=A1.step(3,2)|[null]	=A1.step(3,3)|[null]
3	=A2.derive(B2(#).EId:EId2,B2(#).Name:Name2,B2(#).Dept:Dept2,C2(#).EId:EId3,C2(#).Name:Name3,C2(#).Dept:Dept3)		

举例：补足空行

	A	B
1	=DB.query("select * from employee where gender ='F'")	
2	=length-A1.len()%length	/计算应补的行数
3	=A1|if(A2!=length,A2*[null])	/补充空行数后的结果集

Index	Member
259	[493,Morgan,Davis, ...]
260	[494,Emily,Johnson, ...]
261	[496,Rachel,Johnson, ...]
262	[499,Nicole,Smith, ...]
263	(null)
264	(null)
265	(null)
266	(null)
267	(null)
268	(null)
269	(null)
270	(null)

每页行数length取15时的结果集



目　录

Contents

1 报表与计算层

2 多样性数据源

3 开发体系结构

4 性能与容量

非数据库上的SQL使用

提供简单SQL，使外部数据能够类似数据库表一样用来查询

举例：分组过滤

	A
1	$(esProcOdbc) select Date, count(ID) Count, sum(Quantity) Sum from Order_Electronics.txt group by Date having sum(Quantity)>110000 order by Sum

Index	Date	Count	Sum
1	2013-12-25	139	110002
2	2013-07-22	136	110149
3	2013-04-19	138	110417
4	2013-07-11	136	110655
5	2013-02-10	139	110717
6	2013-04-10	137	110811
7	2013-08-29	138	110856
8	2013-07-26	139	111163
9	2013-01-05	139	111454
10	2013-04-29	136	111526

在分组后，用having从句，从结果中选出sum(Quantity)大于110,000的数据

举例：数据关联

	A
1	$(esProcOdbc) select C.NAME,C.POPULATION, S.StateId, S.Abbr from states.txt S join cities.txt C on S.StateId=C.STATEID

Index	C.NAME	C.POPULATION	S.StateId	S.Abbr
1	Argentina	435245.0	5	CA
2	Auckland	402777.0	5	CA
3	Anaheim	334425.0	5	CA
4	Bakersfield	308392.0	5	CA
5	Chula Vista	212756.0	5	CA
6	Chicago	2886251.0	13	IL
7	Buffalo	276059.0	32	NY
8	Austin	671873.0	43	TX
9	Arlington	349944.0	43	TX
10	Corpus Christi	285267.0	43	TX

用join语句将两个外部表关联，列出能找到对应州简称的城市信息

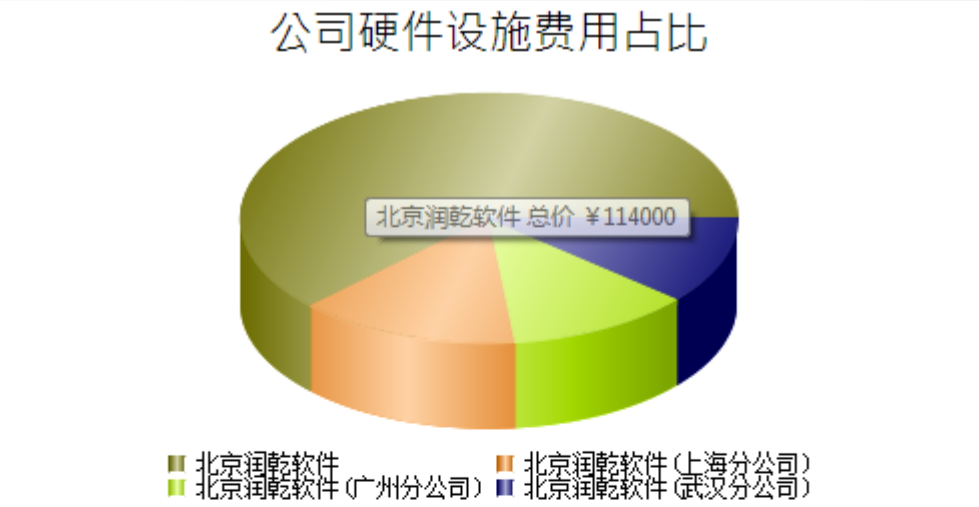
多层数据

简便解析和计算数据为json格式的字符串

举例：将JSON数据作为报表数据源

```
1  {
2      "order": [{
3              "client": "北京润乾软件",
4              "date": "2015-6-23",
5              "item": [{
6                      "product": "惠普笔记本",
7                      "number": 8,
8                      "price": 3200
9                  },
10                 {
11                     "product": "清华同方服务器",
12                     "number": 4,
13                     "price": 22100
14                 }
15             ]
16         },
```

多层JSON



明细清单			
公司	产品	数量	单价
北京润乾软件	惠普笔记本	8	3200
	清华同方服务器	4	22100
北京润乾软件(广州分公司)	苹果笔记本	1	7800
	松下电视	1	9800
	海尔冰箱	1	4400
北京润乾软件(上海分公司)	联想台式机	3	2700
	索尼电视	1	14900
	海尔空调	1	3400
北京润乾软件(武汉分公司)	联想笔记机	2	5100
	夏普电视	1	7800
	格力空调	1	2500

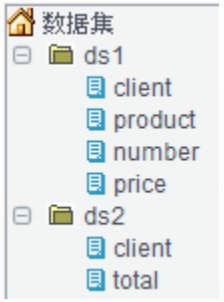
多层数据

简便解析和计算数据为json格式的字符串

举例：将JSON数据作为报表数据源

	A
1	=file("order.json":"utf-8").read().import@j().(order)
2	=A1.news(item;client,product,number,price)
3	=A2.groups(client;sum(number*price):total)
4	return A2,A3

A.news：计算序表的字段值合并生成新序表



一次返回聚合汇总、清单明细结果集

client	date	item
北京润乾软件	2015-6-23	[[惠普笔记本,8,3200],[清华同方...
北京润乾软件(上海分公司)	2017-7-12	[[联想台式机,3,2700],[索尼电视,1...
北京润乾软件(广州分公司)	2016-8-3	[[苹果笔记本,1,7800],[松下电视,1...
北京润乾软件(武汉分公司)	2013-9-20	[[联想笔记机,2,5100],[夏普电视,1...

client	product	number	price
北京润乾软件	惠普笔记本	8	3200
北京润乾软件	清华同方服务器	4	22100
北京润乾软件(上海分公司)	联想台式机	3	2700
北京润乾软件(上海分公司)	索尼电视	1	14900
北京润乾软件(上海分公司)	海尔空调	1	3400
北京润乾软件(广州分公司)	苹果笔记本	1	7800
北京润乾软件(广州分公司)	松下电视	1	9800
北京润乾软件(广州分公司)	海尔冰箱	1	4400
北京润乾软件(武汉分公司)	联想笔记机	2	5100
北京润乾软件(武汉分公司)	夏普电视	1	7800
北京润乾软件(武汉分公司)	格力空调	1	2500

A2为公司采购明细列表

client	total
北京润乾软件	114000
北京润乾软件(上海分公司)	26400
北京润乾软件(广州分公司)	22000
北京润乾软件(武汉分公司)	20500

A3为按公司统计的所有产品总价

跨数据库SQL翻译

将标准SQL翻译成各类数据库的SQL

举例：将标准SQL中”当年中第几星期”对各类数据库进行SQL翻译

标准函数	含义	oracle	sql server	db2	mysql	teradata	hsql	PostgresSQL
WEEKOFYEAR(d)	当年中第几星期	TO_NUMBER(TO_CHAR(d,'WW'))	DATEPART(WW,d)	WEEK(d)	WEEK(d)	TD_WEEK_OF_YEAR(d)	WEEK(d)	EXTRACT(WEEK FROM d)

	A
1	SELECT ID,WEEKOFYEAR(DATES),CUSTOMER,AREA FROM CLUE
2	=A1.sqltranslate("ORACLE")
3	=A1.sqltranslate("SQLSVR")
4	=A1.sqltranslate("MYSQL")

Value
SELECT ID,TO_NUMBER(TO_CHAR(DATES,'WW')),CUSTOMER,AREA FROM CLUE

A2为order对应的SQL语法

Value
SELECT ID,DATEPART(WW,DATES),CUSTOMER,AREA FROM CLUE

A3为sql server对应的SQL语法

Value
SELECT ID,WEEK(DATES),CUSTOMER,AREA FROM CLUE

A4为mysql对应的SQL语法

各类数据库均有自己的**字符串函数,数值函数,时间函数,转换函数,任意函数。**
对这些不同标准进行统一处理，将为开发维护带来极大的便利。
目前支持但不限于：Oracle、SQL Server、DB2、MySQL、HSQL、Teradata、PostgreSQL

动态数据源

报表工具本身不直接支持动态数据源，可借助计算层实现

解决的方法之一是用参数传入 jdbc url/username/password，但会影响安全性（口令在在参数中传递）

举例：在不同的数据源中进行金额大于某阈值的明细查询

	A
1	=${pSource}.query("select * from sOrder where Amount>? ",pAmount)

其中pSource、pAmount是报表参数，pSource代表数据源名，pAmount代表订单金额。

跨库计算

单台数据库数据量太大会影响性能，通常会分库，分担计算压力

简单查询举例：订单表orders分库存储在两个Oracle数据库中，数据源名分别为orclA、orclB。过滤出金额大于等于10000的订单，并按订单金额顺序排序。

	A	B	C
1	=[connect("orclA"),connect("orclB")]		/连接多个数据源
2	SELECT * FROM orders WHERE amount >= 10000 ORDER BY amount		/排序SQL
3	fork A1	=A3.query(A2)	/并行计算
4	=A3.merge(AMOUNT)		/有序结果归并
5	=A3.(~.close())		/关闭连接

分组汇总举例：将订单表按年、月分组，对各组数据的amount字段求和。

	A	B	C
1	=[connect("orclA"),connect("orclB")]		/连接多个数据源
2	SELECT EXTRACT(YEAR FROM orderTime) AS y, EXTRACT(MONTH FROM orderTime) AS m, SUM(amount) AS amount FROM orders GROUP BY EXTRACT(YEAR FROM orderTime), EXTRACT(MONTH FROM orderTime)		/分组汇总SQL
3	fork A1	=A3.query(A2)	/并行计算
4	=A3.conj()		/合并结果
5	=A4.groups(Y,M;sum(AMOUNT):AMOUNT)		/二次分组汇总
6	=A3.(~.close())		/关闭连接

跨库计算

单台数据库数据量太大会影响性能，通常会分库，分担计算压力

连接举例：销售人员表sales是订单表orders的维表，两者以salesID为关联字段，按sales表的部门字段dept分组，求各部门的销售额。假设sales表已全量存储于各数据库。

	A	B	C
1	=[connect("orclA"),connect("orclB")]		/连接多个数据源
2	SELECT sales.dept, SUM(orders.amount) AS amount FROM orders, sales WHERE orders.salesID = sales.salesID GROUP BY sales.dept		/分组汇总SQL
3	fork A1	=A3.query(A2)	/并行计算
4	=A3.conj()		/合并结果
5	=A4.groups(DEPT;sum(AMOUNT):AMOUNT)		/二次分组汇总
6	=A3.(~.close())		/关闭连接

异构库分库举例：orders表分库存储在Oracle和MySQL中，数据源名为ora和my。查询amount字段大于等于10000的记录，并将amount字段截断取整。

	A	B	C
1	=[[connect("ora"),"ORACLE"],[connect("my")] ,"MYSQL"]		/连接数据源，标记数据库类型
2	SELECT ORDERID, ORDERTIME, truncate(AMOUNT, 0) , CLIENTID, SALESID FROM orders WHERE amount >= 10000		/标准SQL
3	fork A1	=A2.sqltranslate(A3(2))	/转为本地SQL
4	=A3.conj()	=A3.query(B3)	/查询
5	=A3.conj()		/合并结果
6	=A3.(~.close())		/关闭连接

NoSQL/Hadoop

报表工具一般仅支持常用ODBC或JDBC方式连接数据库，针对于部分不支持ODBC与JDBC连接的数据库，便可以使用外部库功能。

以MongoDB举例

1、部署外部库：

从网上下载以下两个文件（请根据实际所用版本下载对应的jar文件），放到产品外部库文件夹中；Mongo外部库文件路径为：安装目录\esProc\extlib\MongodbCli；其中润乾核心jar为mongoCli.jar。

bson-3.6.3.jar

mongo-java-driver-3.6.3.jar

2、访问MongoDB数据库：

可使用的外部库函数有mongo_open()、mongo_shell()、mongo_close()。

	A	B
1	=mongo_open("mongodb://localhost:27017/mydb")	/连接mongo server的mydb库
2	=mongo_shell(A1,"emp.find()").fetch()	/查询mydb库中的emp集合中的记录
3	=mongo_close(A1)	/关闭数据库连接

外部库还可以分别实现访问阿里云、elasticsearch、hive、spark、hbase、redis、cassandra、informix数据库、读取报表文件、连接hdfs文件系统、多维数据库、webservice、ftp、sap、kafka系统

多数据源混合

报表工具不支持多源混合或效率很低，可借助集算器实现

举例：销售表中CLIENT列是客户名，AMOUNT列是订单金额。另有potential.json文件中存放潜力客户名列表。要求按照潜力客户文件中的顺序，在销售表中分组汇总订单金额。

	A
1	=sales=demo.query("select * from sales")
2	=potential=file("potential.json").read().import@j().(potential)
3	=sales.align@a(potential,CLIENT)
4	=A3.new(potential(#):CLIENT,~.sum(AMOUNT):AMOUNT)

ORDERID	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
4	PWQ	9	26100.0	2012-11-08 15:28:05

A1中的销售表部分数据

Index	Member
1	GLH
2	EGU
3	YZ
4	MIP
5	VILJX
6	DNEDL
7	PJIPE
8	HANAR
9	BTMMU
10	JOPO

A2为潜力客户列表

Index	Member
1	[[39,GLH,14, ...],[49,GLH,2, ...],[68,GLH,13, ...], ...]
2	[[7,EGU,2, ...],[20,EGU,8, ...],[90,EGU,4, ...], ...]
3	[[30,YZ,19, ...],[65,YZ,8, ...],[78,YZ,1, ...], ...]
4	[[15,MIP,16, ...],[69,MIP,16, ...],[74,MIP,11, ...], ...]
5	[[8,VILJX,7, ...],[35,VILJX,6, ...],[41,VILJX,8, ...], ...]
6	[[18,DNEDL,16, ...],[59,DNEDL,3, ...],[101,DNEDL,5, ...], ...]
7	[[17,PJIPE,3, ...],[26,PJIPE,20, ...],[111,PJIPE,11, ...], ...]
8	[[6,HANAR,18, ...],[34,HANAR,5, ...],[43,HANAR,5, ...], ...]
9	[[23,BTMMU,17, ...],[40,BTMMU,17, ...],[92,BTMMU,2, ...], ...]
10	[[19,JOPO,2, ...],[27,JOPO,20, ...],[80,JOPO,9, ...], ...]

A3中按照潜力客户分组后的结果

ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
39	GLH	14	22200.0	2012-12-15 15:28:05
49	GLH	2	8526.0	2012-12-24 15:28:05
68	GLH	13	24700.0	2013-01-07 15:28:05
75	GLH	6	20140.0	2013-01-13 15:28:05

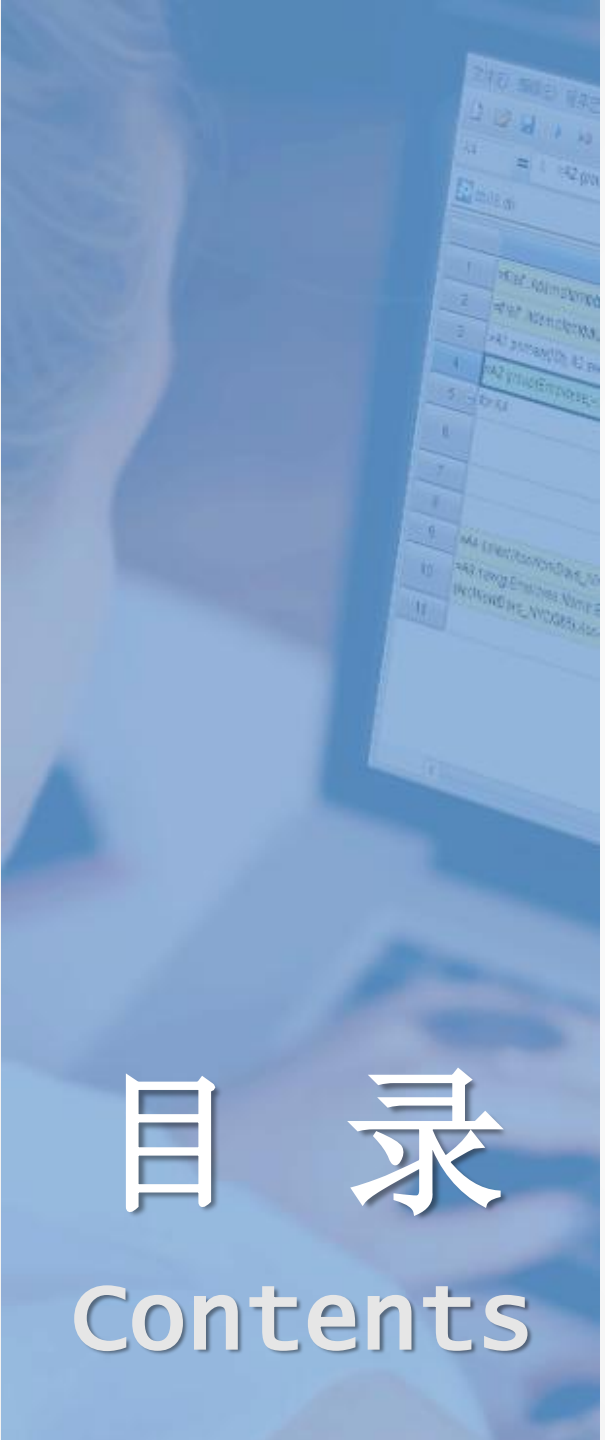
ORDERID	CLIENT	SELLERID	AMOUNT	ORDERDATE
7	EGU	2	17800.0	2012-11-06 15:28:05
20	EGU	8	11700.0	2012-11-21 15:28:05
90	EGU	4	26700.0	2013-01-30 15:28:05
110	EGU	10	11000.0	2013-02-24 15:28:05

… …

A3中前两组中的记录

CLIENT	AMOUNT
GLH	461052.0
EGU	598344.0
YZ	550882.0
MIP	397000.0
VILJX	585320.0
DNEDL	601164.0
PJIPE	443102.0
HANAR	669736.0
BTMMU	496226.0
JOPO	632798.0

A4为潜力客户的汇总金额



目　录

Contents

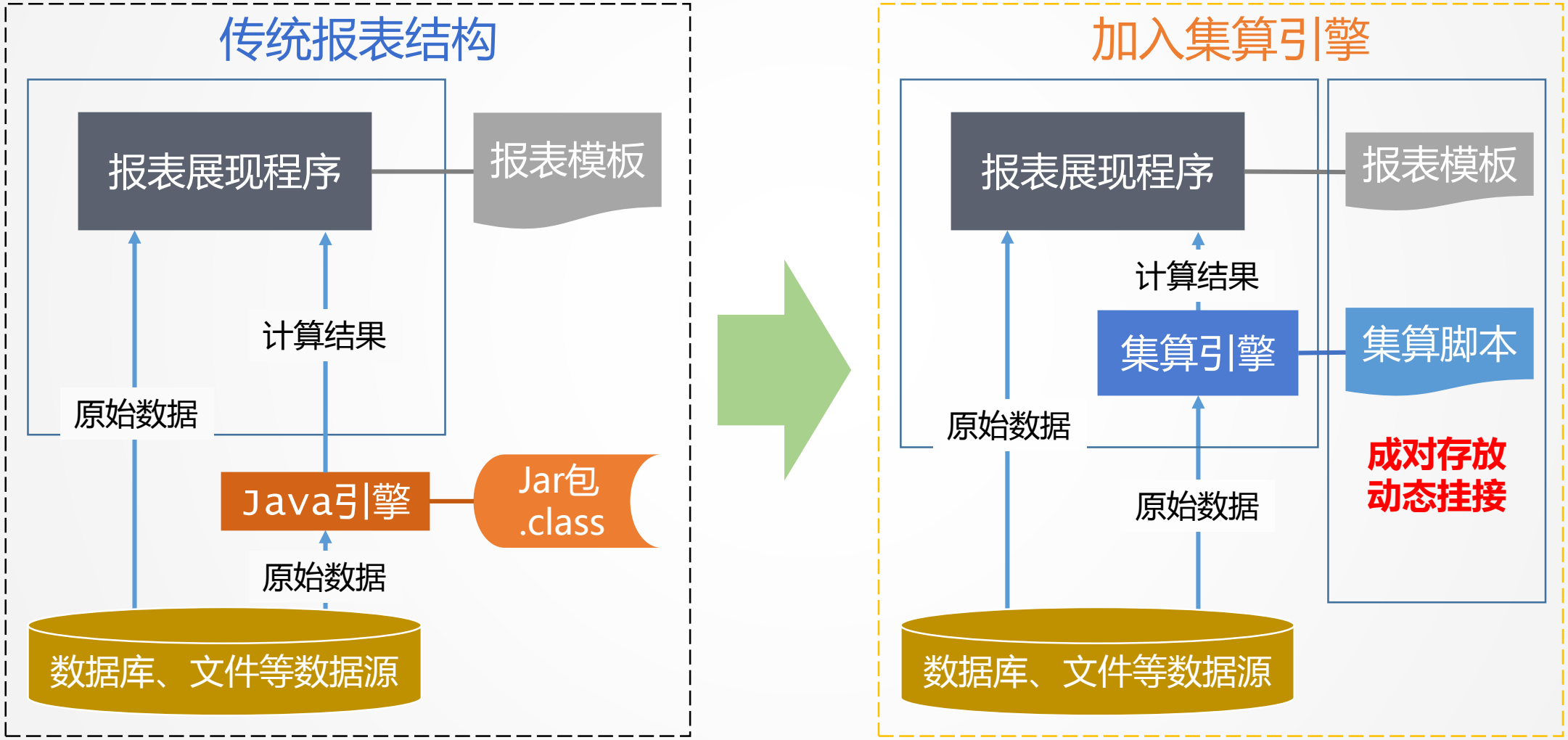
1 报表与计算层

2 数据源

3 开发体系结构

4 性能与容量

应用外挂算法——结构图


传统报表结构
报表展现程序
报表模板
计算结果
原始数据
Java引擎
Jar包
.class
原始数据
数据库、文件等数据源
加入集算引擎
报表展现程序
报表模板
计算结果
集算引擎
集算脚本
原始数据
成对存放
动态挂接
原始数据
数据库、文件等数据源

应用外挂算法

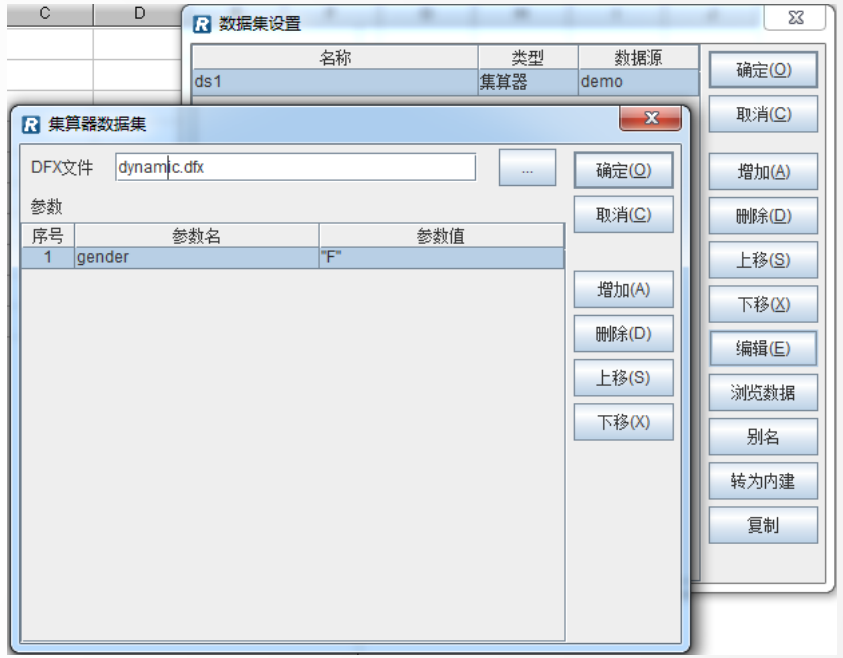
展现模板与算法，成对存放，易于维护，热部署

举例：员工绩效工资报表

	A	B
1	=demo.query("select * from employee where gender = ? ",gender)	
2	=A2.derive(PRETAX)	
…	//此处省略N行复杂代码	
12	result A11	

dynamic.dfx

算法脚本
动态挂接
单独管理



gender参数是dynamic.dfx的输入参数，数据集ds1接收dynamic.dfx返回的结果集

	A	B	C
1	EMPNAME	BASESALARY	PRETAX
2	=ds1.select(EMPNAME)	=ds1.BASESALARY	=ds1.PRETAX

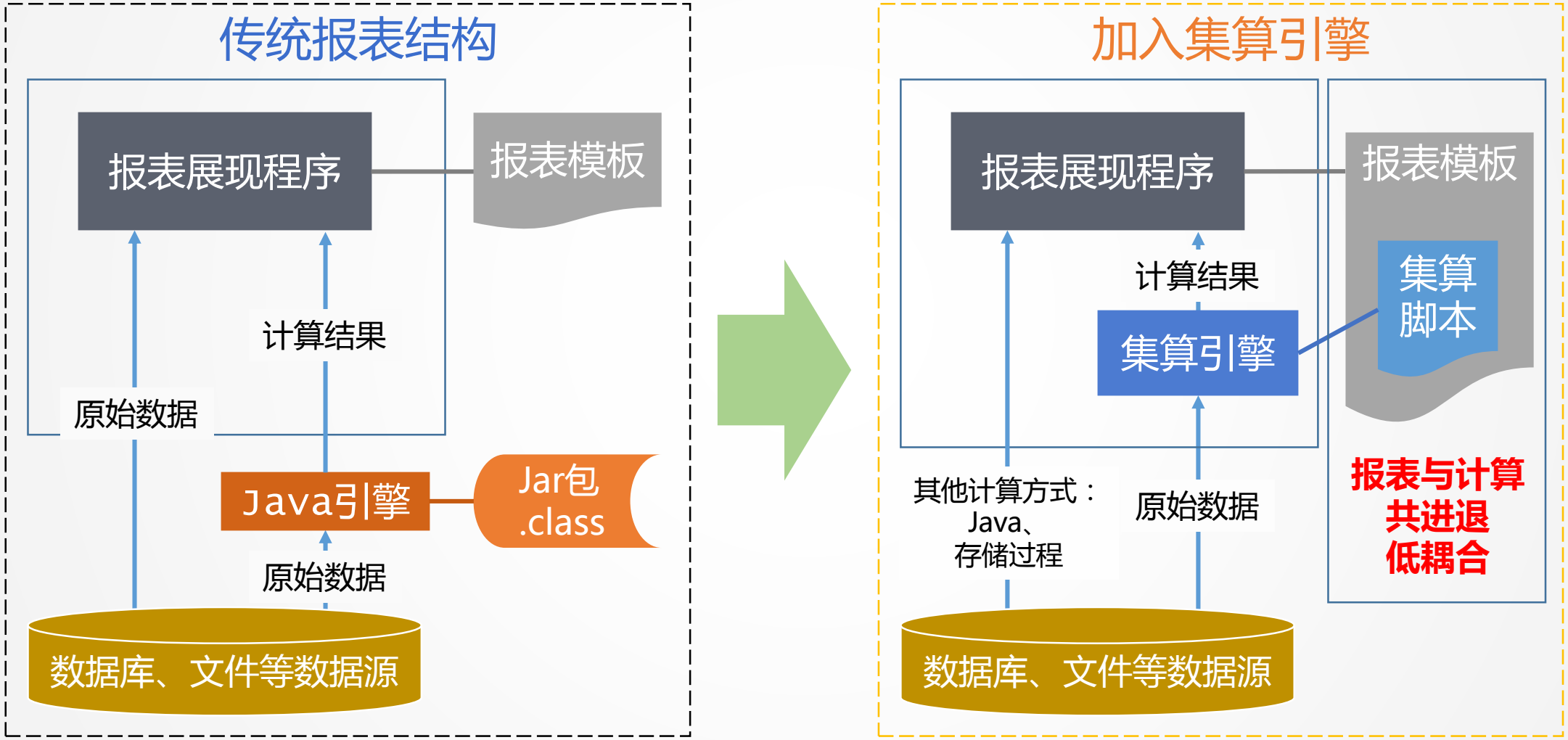
设计报表模板

模板设计简单清爽
避免复杂业务逻辑

EMPNAME	BASESALARY	PRETAX
Rachel	$8000.00	$8052.80
Rebecca	$6000.00	$6000.00
Ashley	$10000.00	$10000.00
Emily	$800.00	$800.00
Ashley	$3000.00	$3000.00

展现效果

报表模板与算法--结构图

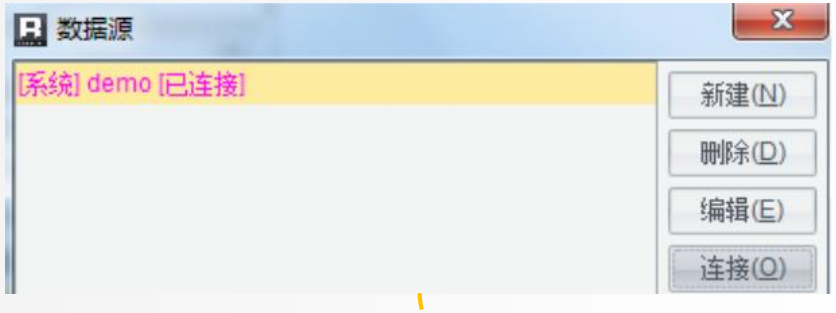

传统报表结构
报表展现程序
报表模板
计算结果
原始数据
Java引擎
Jar包
.class
原始数据
数据库、文件等数据源
加入集算引擎
报表展现程序
报表模板
计算结果
集算
脚本
集算引擎
其他计算方式：
Java、
存储过程
原始数据
报表与计算
共进退
低耦合
数据库、文件等数据源

报表模板与算法

报表模板与算法保存在同一个文件中，共进退，低耦合，易维护

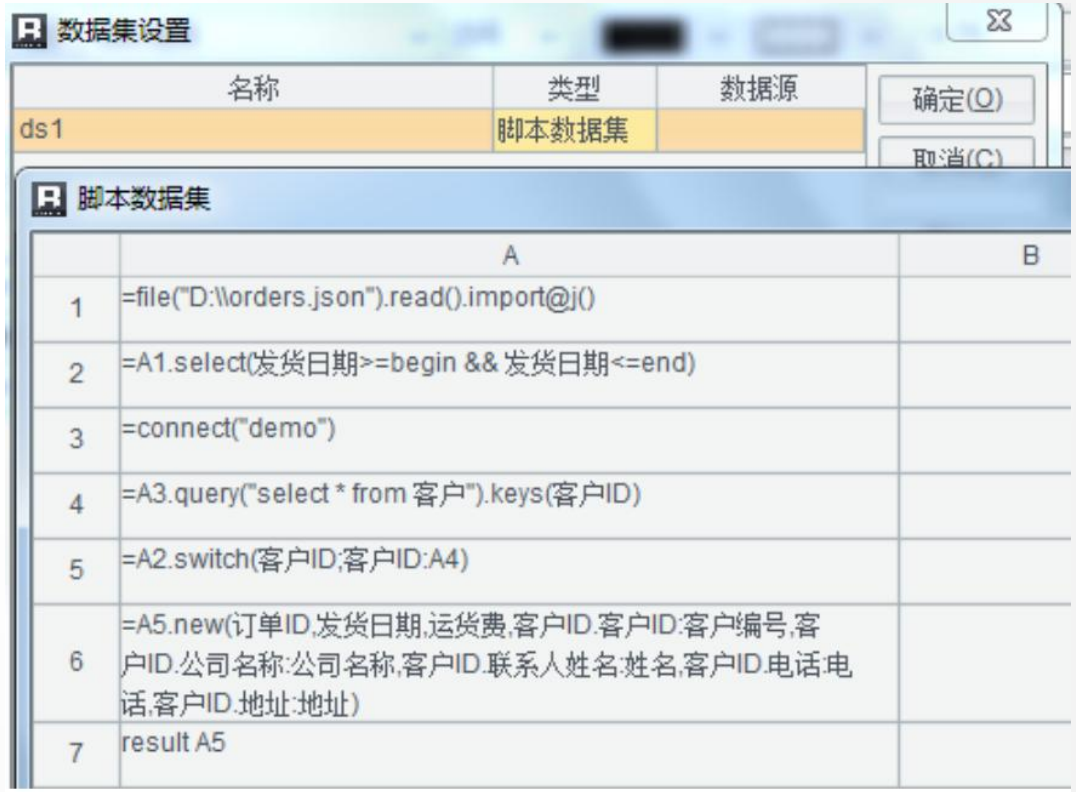
举例：客户订单信息表

1、连接数据源



2、设置报表参数

3、设置数据集



报表模板与算法

报表模板与算法保存在同一个文件中，共进退，低耦合，易维护

举例：客户订单信息表

4、编辑报表表达式

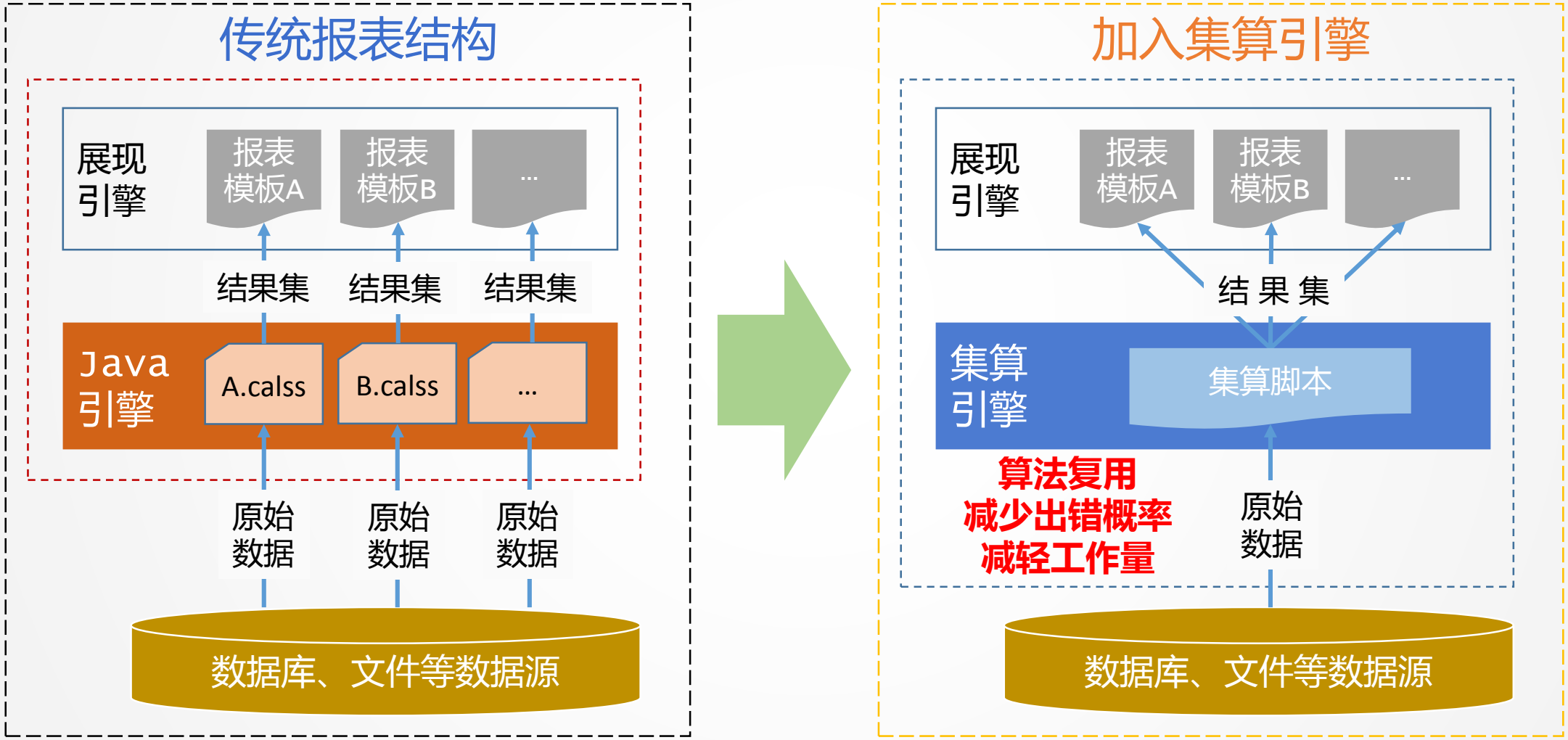
	A	B	C	D	E	F	G	H
1(标)	订单客户信息							
2(头)	订单ID	发货日期	运货费	客户编号	公司名称	姓名	电话	地址
3	=ds1.select	=ds1.发货	=ds1.运货	=ds1.客户	=ds1.公司	=ds1.姓名	=ds1.电话	=ds1.地址

5、报表展现结果

订单客户信息

订单ID	发货日期	运货费	客户编号	公司名称	姓名	电话	地址
10324	1996-10-10	￥214.27	SAVEA	大钰贸易	胡继尧	(071) 8555	冠成园路 321 号
10325	1996-10-14	￥64.86	KOENE	永业房屋	谢丽秋	(025) 5550	东园大路 78 号
10326	1996-10-14	￥77.92	BOLID	迈多贸易	陈先生	(091) 8555	临翠大街 80 号
10327	1996-10-14	￥63.36	FOLKO	五洲信托	苏先生	(087) 6953	沿江北路 942 号
10328	1996-10-17	￥87.03	FURIB	康浦	王先生	(087) 4354	授业路 361 号
10329	1996-10-23	￥191.67	SPLIR	昇昕股份有	谢小姐	(0571) 355	车站南路 721 号
10330	1996-10-28	￥12.75	LILAS	富泰人寿	陈先生	(030) 3311	光伦东路 381 号
10331	1996-10-31	￥10.19	BONAP	祥通	刘先生	(078) 9124	花园东街 90 号

算法复用--结构图


传统报表结构
展现引擎
报表模板A
报表模板B
...
结果集
结果集
结果集
Java引擎
A.calss
B.calss
...
原始数据
原始数据
原始数据
数据库、文件等数据源
加入集算引擎
展现引擎
报表模板A
报表模板B
...
结 果 集
集算引擎
集算脚本
算法复用
减少出错概率
减轻工作量
原始数据
数据库、文件等数据源

算法复用

通用性算法复用，减少出错概率，减轻工作量

举例：“员工绩效工资明细表”和“部门绩效工资汇总表”都要计算员工的绩效工资，算法虽然复杂但基本相同。

	A	B
1	=demo.query("select * from employee")	
2	if (trim(where) !=null)	=A1.select@o(${where})
…	//此处省略N行复杂代码	
13	result A12	

EMPID	SCORE	DEPT	EMPNAME	BASESALARY	STATE	PRETAX
3	1.10	Sales	Rachel	8000	New Mexico	8052.80...
6	1.18	Sales	Matthew	4000	California	4037.76...
1	0.75	R&D	Rebecca	6000	California	6000
2	0.90	Finance	Ashley	10000	New York	10000
4	0.80	HR	Emily	800	Texas	800
5	1.40	R&D	Ashley	3000	Texas	3000

当查询条件where为空时，返回的结果集适合“部门绩效工资汇总表”

EMPID	SCORE	DEPT	EMPNAME	BASESALARY	STATE	PRETAX
6	1.18	Sales	Matthew	4000	California	4037.76...
1	0.75	R&D	Rebecca	6000	California	6000
4	0.80	HR	Emily	800	Texas	800
5	1.40	R&D	Ashley	3000	Texas	3000

当查询条件where为STATE==“California” || STATE==“Texas”时，返回的结果集适合“员工绩效工资明细表”

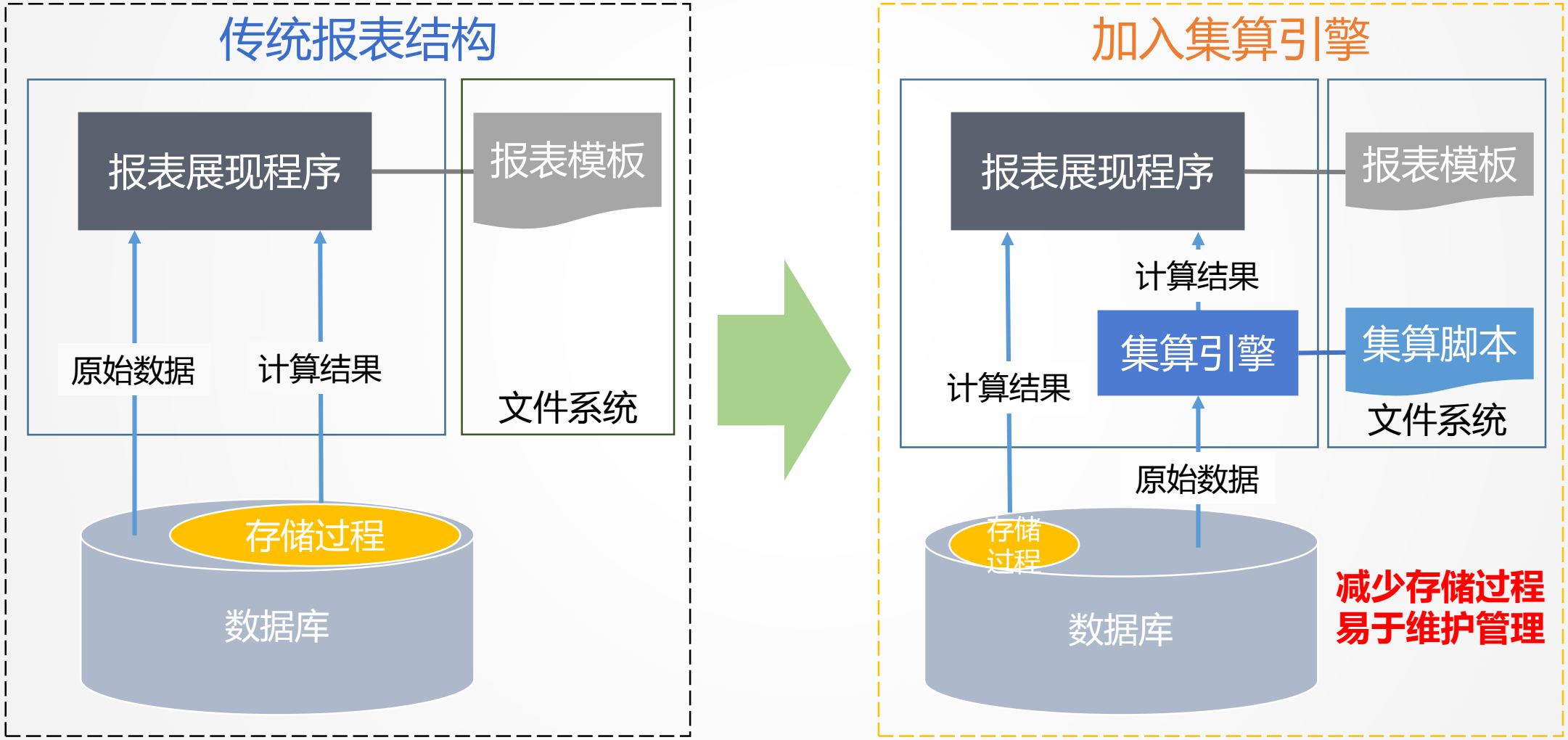
EMPNAME	BASESALARY	PRETAX	STATE
Matthew	$4000.00	$4037.76	California
Rebecca	$6000.00	$6000.00	California
Emily	$800.00	$800.00	Texas
Ashley	$3000.00	$3000.00	Texas

员工绩效工资明细表

department	count	total:basesalary	total:pertax
Finance	1	$10,000.00	$10,000.00
HR	1	$800.00	$800.00
R&D	2	$9,000.00	$9,000.00
Sales	2	$12,000.00	$12,090.56

部门绩效工资汇总表

库外存储过程——结构图


传统报表结构
报表展现程序
报表模板
原始数据
计算结果
文件系统
存储过程
数据库
加入集算引擎
报表展现程序
报表模板
计算结果
集算引擎
集算脚本
计算结果
文件系统
原始数据
存储过程
数据库
减少存储过程
易于维护管理

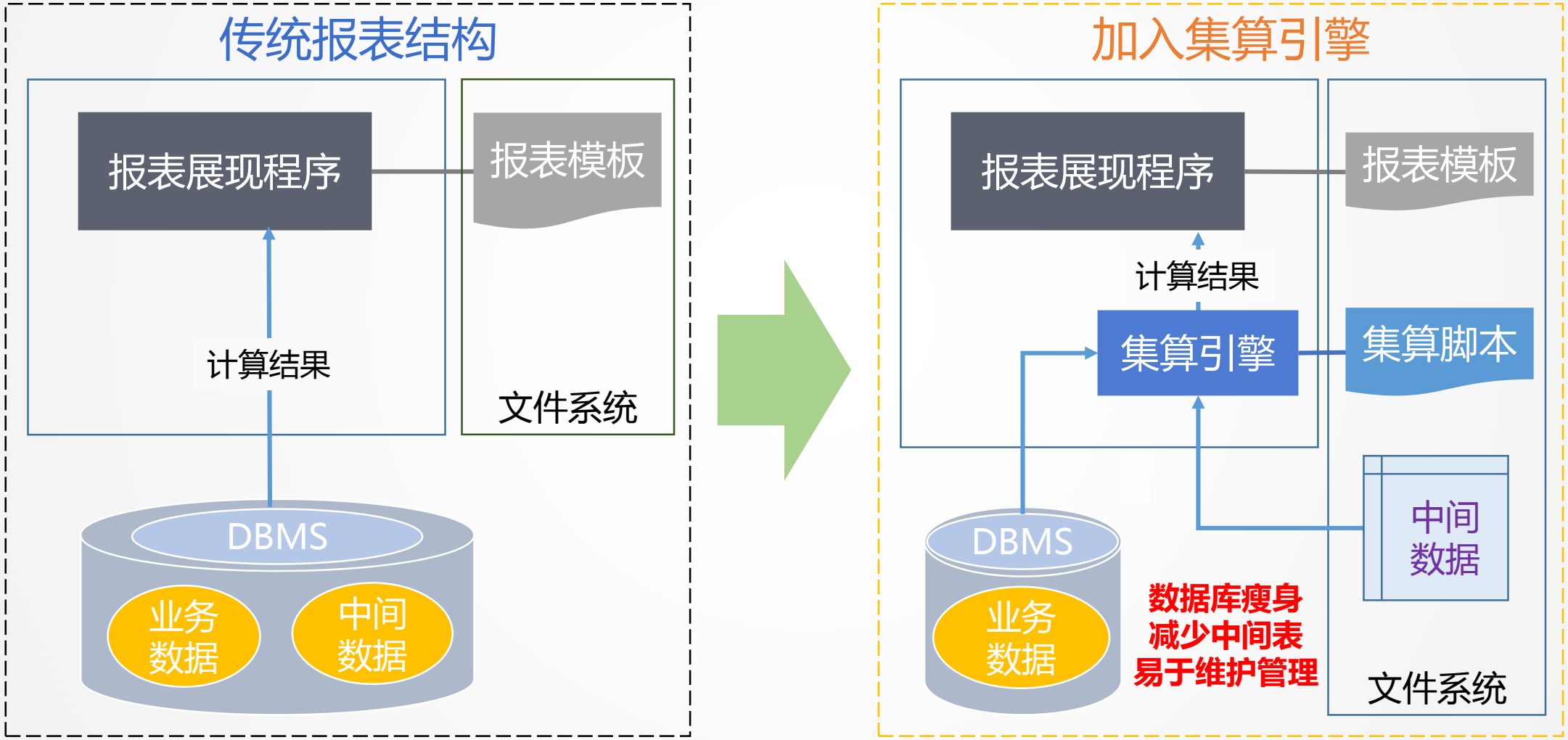
库外存储过程

数据库中不便修改存储过程时，可借助集算器实现库外计算模块

举例：某开发团队的计算目标是：指定一年中的某个时间段，计算该时间段内每个州的销量均在前二十名的优势产品有哪些？统计出这些优势产品每个月的销量，以及每个月比上个月的销量增长率。

	A	B	C
1	=ora.query("select State,Product,Amount,Time from sales where to_char(Time,'YYYMMDD')>=? and to_char(Time,'YYYMMDD')<=? ";start,end)		/根据时间参数从sales table取数据
2	=A1.group(State)	=A2.(~.rank(Amount).pselect@a(~<=20))	/A2：把A1中的数据按state进行分组。B2：在A2的每个组（州）里，求出销量前20的产品的记录序号。
3	=A2.(~(B2(#)).(Product))		/用序号从A2取出每个州对应的产品
4	=A3.isect()		/求A3中各组产品的交集。这里的计算结果就是优势产品
5	=A1.select(A4.pos(Product)>0)		/过滤出优势产品的销量记录
6	=A5.groups(Product,month(Time):Month;sum(Amount):MonthAmount)		/根据A5，汇总出每种产品每个月的销量
7	=A6.derive(Rate)		/为A6增加一个字段Rate，以便将来存储增长率
8	=A7.group(Product)		/将A7按产品重新分组
9	=A8.(~.run(Rate=MonthAmount[-1]/MonthAmount-1))		/计算每种产品每个月比上个月的增长率
10	=A9.union()		/将A9中的分组数据合并。这就是最终计算目标
11	return A10		/返回结果A10

应用内中间表——结构图


传统报表结构
报表展现程序
报表模板
计算结果
文件系统
DBMS
业务数据
中间数据
加入集算引擎
报表展现程序
报表模板
计算结果
集算引擎
集算脚本
中间数据
DBMS
业务数据
数据库瘦身
减少中间表
易于维护管理
文件系统

应用内中间表

数据库中间表的增加会引起数据库性能下降；增加管理、维护成本

举例：某公司客户销售额与去年全年销售额对比报表（年数据量大）。

1、建立中间表：从数据库中读取去年的销售数据，并导出为中间文件。

	A	B
1	=db.query("select client,count(orderid) c,sum(amount) s from sales where year(orderdate)=? group by client",year(now())-1)	/从数据库中读取2009年的销售数据
2	=file(string(year(now())-1)+"sales.b").export@b(A1)	/去年的数据导出成集文件

2、将中间文件（去年数据）与数据库中的（今年数据）共同计算，得出销售额对比结果。

	A	B
1	=db.query("select client,count(orderid) c,sum(amount) s from sales where year(orderdate)=? group by client",year(now()))	/从数据库中计算取出要查询的年份订单数、销售额。
2	=file(string(year(now())-1)+"sales.b").import@b()	/从前一年的数据文件中取出数据
3	=A2.align(A1:CLIENT,CLIENT)	/将A2中的数据按照A1中的CLIENT字段对齐，A1中有但A2中没有的则补空行
4	=A1.new(CLIENT,C:COUNT,S:TOTAl,A3(#).C:lastCOUNT,A3(#).S:lastTOTAL,S/A3(#).S:PROPORTION)	/利用A1来生成新的续表。其中增加了A3的对应行数据，比如A3(#).C:lastCOUNT就是A3的对应行中取出C字段，其中#是A1的当前行号
5	return A4	/返回结果集

某公司客户累计销售额与去年全年销售额对比报表

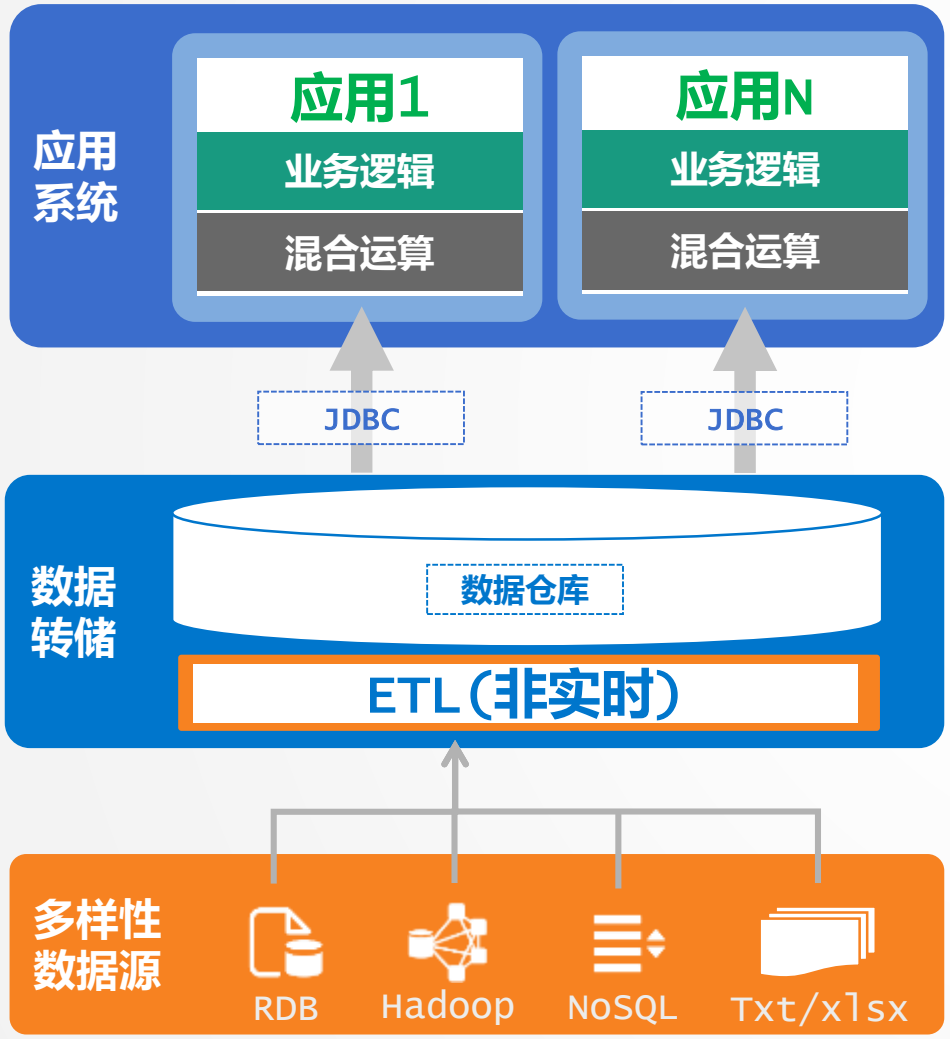
客户	订单数	销售额	去年订单数	去年销售额	销售额/去年销售额
JOPO	12	107010.0	19	255012.0	41.96%
AYWYN	8	136578.0	12	152600.0	89.50%
SAVEA	13	137916.0	13	158874.0	86.81%
PJIPE	8	106436.0	17	225042.0	47.30%
HANAR	18	275464.0	19	242076.0	113.79%
ERNSH	15	158446.0	16	141660.0	111.85%
DNEDL	13	209652.0	17	232994.0	89.98%
BTMMU	15	217186.0	10	168790.0	128.67%
JAYB	13	201706.0	14	188348.0	107.09%
HP	15	217986.0	23	244816.0	89.04%

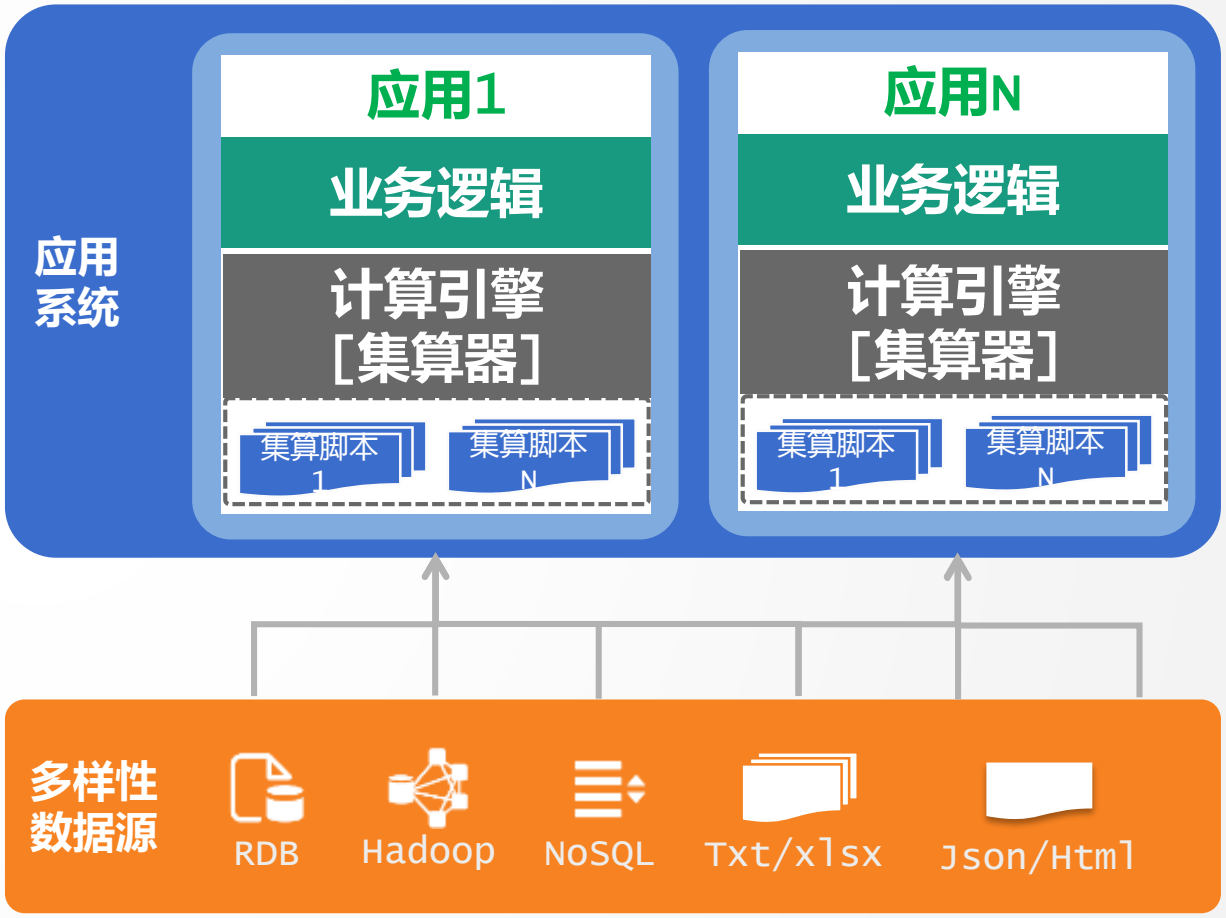
这张报表中的客户、订单数、销售额都是直接从数据库计算出的，即今年的数据。

去年全年的订单数、销售额是从文件系统中的xxxsales.b文件中读取。

"销售额/去年销售额"则是今年和去年的数据共同计算的。

实时多源报表





实时获取数据混合计算
体系结构更精简

没完没了的报表

报表数据准备环节独立出来，报表开发彻底工具化，实现高效率

报表呈现层

数据准备层 专用工具

数据源层 RDB Hadoop NoSQL Txt/xlsx Json/Html

报表开发工具化
报表绘制与数据准备两个环节都可以使用高效开发工具

报表模块独立化
报表呈现及数据准备工作从应用程序和数据库中独立出来，单独维护，与应用解耦

开发人员广泛化
一般技术人员都能做，成本低，响应快

报表模板与计算脚本均置于应用系统之外，可独立团队开发，报表实施过程全面工具化

人员配备和知识传承

应对报表没完没了的问题，是个涉及技术和管理等多方面的综合事务

解决呈现问题
减轻工作量

引入报表工具

提升数据准备的开发效率

引入计算工具

完善沟通机制

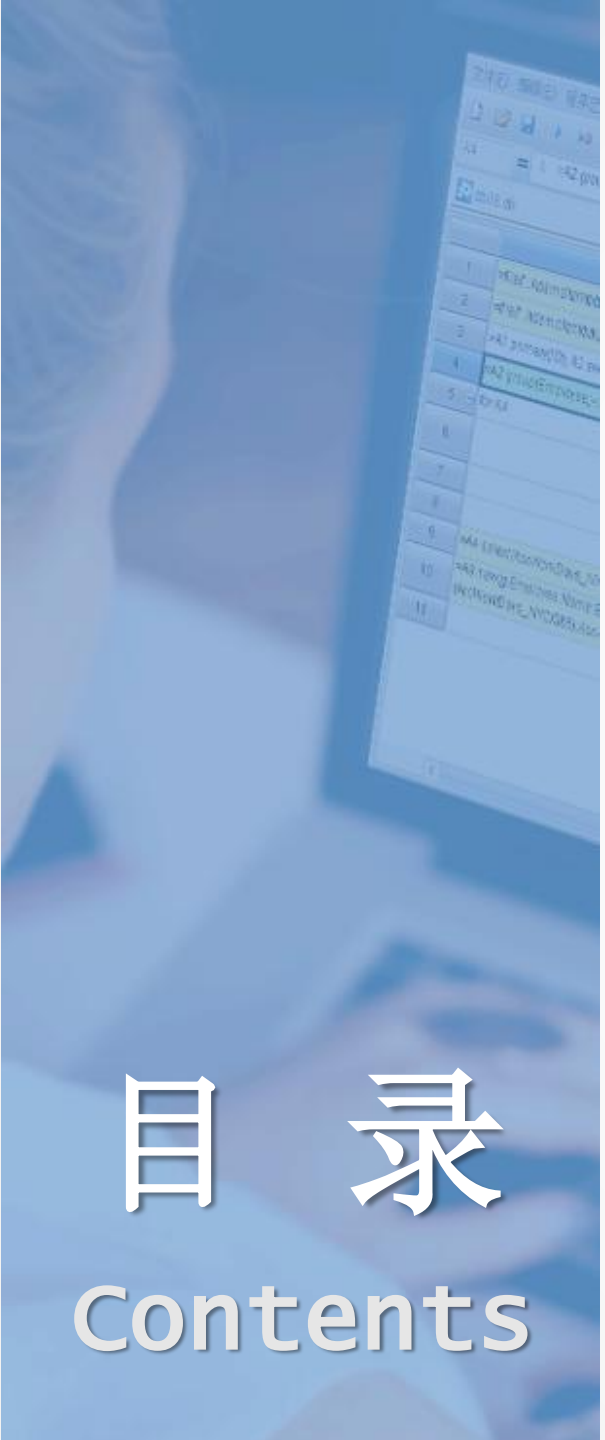
建立有效的沟通机制
减少交流中的误解

独立报表模块

调整应用结构
把报表模块从业务系统中解耦出来

建设报表团队

降低开发人员成本
提升业务的响应速度



目　录

Contents

1 报表与计算层

2 数据源

3 开发体系结构

4 性能与容量

并行取数

数据库JDBC性能较差，可借助集算器中多线程并行解决该问题

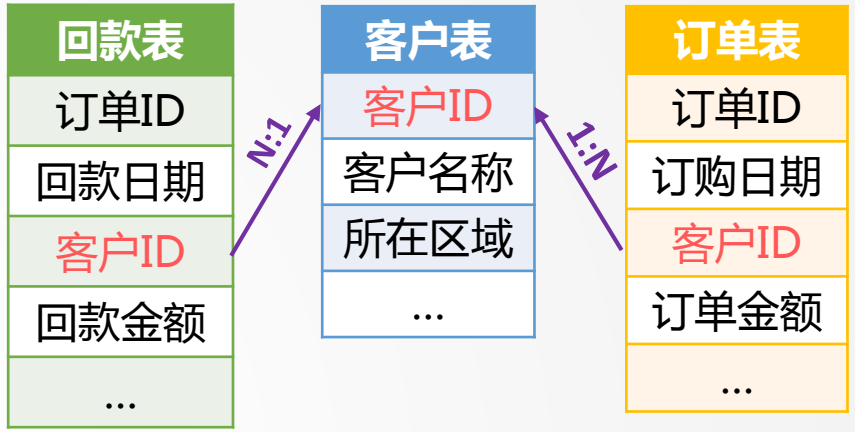
举例：根据表T的数据字段part将数据分为四个部分，每个线程读取一部分数据

	A	B	C
1	fork 4	=connect(db)	/分4线程，要分别建立连接
2		=B1.query@x("select * from T where part=? ",A1)	/分别取每一段
3	=A1.conj()		/合并结果

多表关联

在报表工具中完成多表关联往往导致报表效率不高

举例：报表中有三个数据集，分别来自回款表、客户表和订单表，回款表的客户ID指向客户表主键客户ID，订单表的客户ID也指向客户表主键客户ID。



	A
1	=connect("demo")
2	=A1.query("SELECT 客户ID, SUM(订单金额) AS 订单金额 FROM 订单 WHERE 订购日期 >=? AND 订购日期 <? GROUP BY 客户ID",begin,end)
3	=A1.query(“SELECT 客户ID, SUM(回款金额) AS 回款金额 FROM 回款 WHERE 回款日期 >= ? AND 回款日期 <= ? GROUP BY 客户ID",begin,end)
4	=A1.query@x("select 客户ID, 公司名称 from 客户")
5	=join@1(A4:客户,客户ID;A2:订单,客户ID;A3:回款ID,客户)
6	=A5.new(客户.公司名称:客户名称,订单.订单金额:订单金额,回款.回款金额:回款金额)

客户	订单	回款
[RANCH,大东补...	[RANCH,1694.7]	[RANCH,443.4]
[LILAS,富泰人寿]	[LILAS,4833.399...	[LILAS,3463.6]
[LINOD,保信人寿]	[LINOD,9117.08...	[LINOD,2160.0]
[RATTC,学仁贸易]	[RATTC,19982.5...	[RATTC,14364.2]

客户ID	公司名称
RANCH	大东补习班

客户ID	订单金额
RANCH	1694.7

客户	回款金额
RANCH	443.4

A5按照客户表左关联订单和回款数据

客户名称	订单金额	回款金额
大东补习班	1694.7	443.4
富泰人寿	4833.39...	3463.6
保信人寿	9117.08...	2160.0
学仁贸易	19982.5...	14364.2

A6获得关联结果返回报表数据集

拆解SQL计算

执行路径难以控制，阻碍报表优化，拆解SQL完成报表优化

举例：某明细表，数据量较大，涉及到多张数据库表，数据库表间关联频繁。SQL的执行路径不可控。

```
select syb.org_abbreviation as syb,
        max(xmb.org_abbreviation) as xmb,
        ...
        --省略多个连接、判断、汇总语句
        left join losrrr losr on losr.requisition_id =
            l.requisition_id
        where l.table_type = '1'
            and l.requisition_state =
'0100500005000000006'
            and nvl(l.bsflag, 0) != 1
        group by l.requisition_id,
                l.note,
                --省略多个分组字段
                losr.standby_param3,
                losr.standby_param6
```

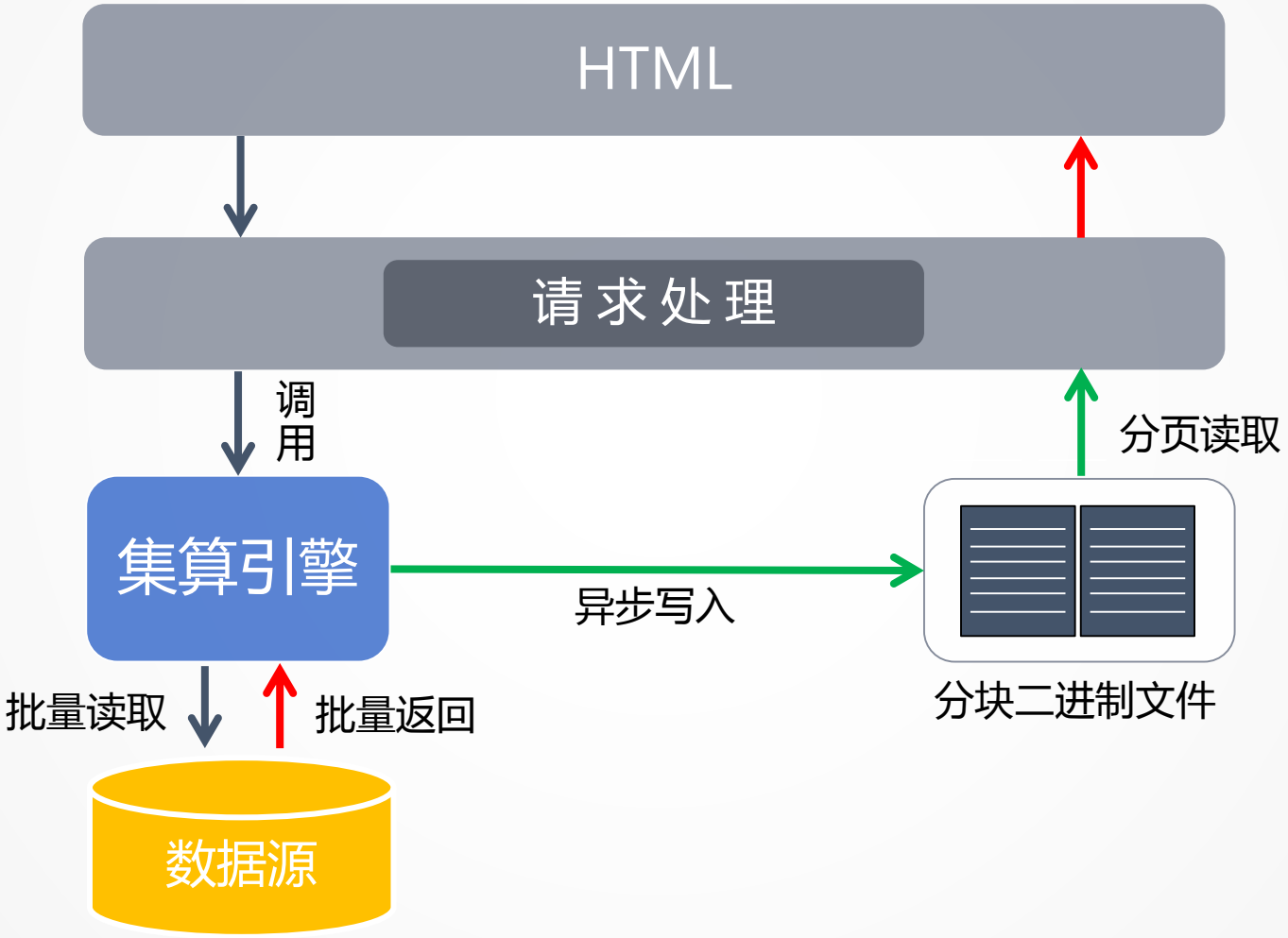
复杂的数据集SQL（近400行）

	A	B
1	=connect("database")	
2	=A1.query("select syb.org_abbreviation as syb ...")	/LEFT JOIN的前半部分SQL
3	=A1.query("select l.requisition_id, l.table_type ...")	/LEFT JOIN的视图部分
4	=A2.switch(REQ_ID,A3:REQUISITION_ID)	
5	=A4.new(#:SN,SYB, XMB, SUB_ID, ORG_ABB, ...)	
6	//此处省略n行N列，求汇总值	
7	>A5.insert(0,"合计","", "", "", ...)	
8	return A5	

拆分原报表数据集SQL。分别把两个子查询sql写到集算脚本中执行，并且在集算脚本中完成关联。

	数据集计算	报表计算展现	总时间
优化前	317秒	85秒	402秒
优化后	52秒	5秒	57秒

大报表---结构图



大清单——SQL源

海量清单报表的秒级查询

举例：根据日期范围查询订单表的交易信息

1、报表参数为查询日期起止：

参数编辑

名称	描述	数据类型	值表达式	参数类型
begin	查询日期...	日期	2012-01-01	普通参数
end	查询日期...	日期	2015-12-31	普通参数

确定(O) 取消(C)

2、数据集根据参数查询订单表 SQL：

数据集设置

名称	类型	数据源
ds1	SQL检索	demo

确定(O) 取消(C)

SQL编辑器

数据表 数据字段 计算列 检索条件 排序 语法 参数

```
SELECT * FROM 订单 where 订购日期 >= ? and 订购日期 <= ?
```

确定(O) 取消(C)

3、报表模板：

	A	B	C	D	E	F	G	H
1(标)	交易记录查询							
2(头)							=string(@begin)+" 到 "+string(@end)	
3(头)	订单ID	客户ID	订购日期	运货费	货主名称	货主城市	货主地区	订单金额
4	=ds1.select(订单	=ds1.客户ID	=ds1.订购日期	=ds1.运货费	=ds1.货主名称	=ds1.货主城市	=ds1.货主地区	=ds1.订单金额

4、与普通报表不同，需要在报表属性（报表 - 报表属性）中设置“大数据集名称”，指向数据集 ds1，直接利用 SQL 完成异步取数。

报表属性

常规 打印 分页 导出 背景图 子报表 其他

报表类型：普通报表

大数据集名称：ds1

报表单位：毫米

图像最大尺寸(像素,0为不限制)：宽度 0 高度 0

显示比例：200% 150% 100% 75% 50% 自定义

5、设置完成后，在报表设计器 IDE 中即可浏览报表：

浏览报表: orders.rpx

正在取数 共 161 行，合 7 页。 到第 1 页 最前页 上一页 下一页 最后页

交易记录查询

2012-01-01 到 2015-12-31

订单ID	客户ID	订购日期	运货费	货主名称	货主城市	货主地区	订单金额
10248	VINET	2012-07-04	￥32.38	余小姐	北京	华北	￥428.00
10249	TOMSP	2012-07-05	￥11.61	谢小姐	济南	华东	￥1842.00
10250	HANAR	2012-07-08	￥65.83	谢小姐	秦皇岛	华北	￥1523.50
10251	VICTE	2012-07-08	￥41.34	陈先生	南京	华东	￥624.95
10252	SUPRD	2012-07-09	￥51.30	刘先生	长春	东北	￥3559.50

大清单——非SQL源

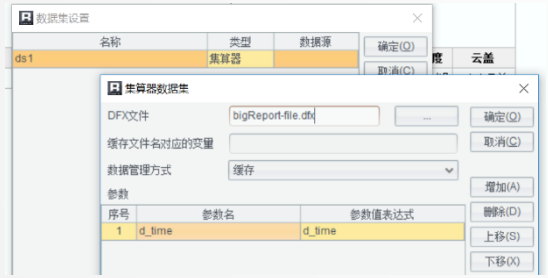
从非RDB数据源取数并缓存数据，再读取缓存并分页展现

举例：按照日期查询某日风速、温度等明细信息。

1、SPL脚本bigReport-file.dfx：

	A	B
1	=file(“source.csv”).cursor@t(;,",")	/建立文件游标
2	=A1.select(时间 == d_time)	/在游标执行时对数据进行过滤（此时游标尚未执行，数据并未取出）
3	return A2	/返回游标过滤结果，为报表提供数据

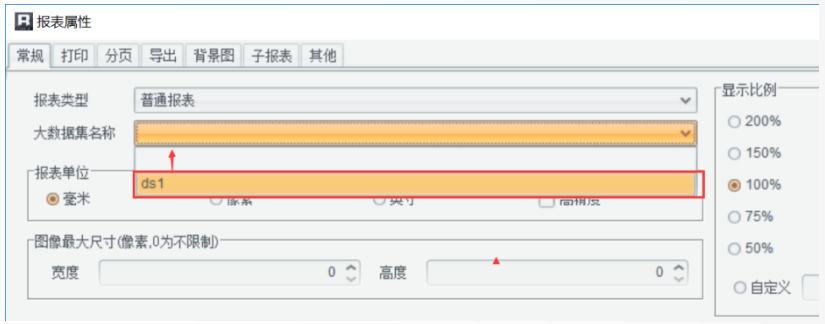
2、在报表中作为数据集引入：



3、报表模板：

	A	B	C	D	E	F	G	H	I	J	K
1(头)	当日气象信息										
2(头)	时间	经度	纬度	气压层(mb)	海拔	风向	风速	温度	具体温度	相对温度	云盖
3	=ds1.select(时间)	=ds1.经度	=ds1.纬度	=ds1.field("	=ds1.海拔	=ds1.风向	=ds1.风速	=ds1.温度	=ds1.具体温	=ds1.相对温	=ds1.云盖

4、设置大数据集：



5、设置完成后，在报表设计器 IDE 中即可浏览报表：

日期： 2013/1/1 查询

当日气象信息

时间	经度	纬度	气压层(mb)	海拔	风向	风速	温度	具体温度	相对温度	云盖
2013/1/1 0:00	-180	90	1	0	422342.065	0.023047427	3.88E-6	222.1052561	35.19579233	4.977473494
2013/1/1 0:00	-180	89.25	1	0	422888.3066	0.023047427	3.88E-6	222.1382242	38.31720027	3.046657209
2013/1/1 0:00	-180	88.5	1	0	423472.4816	0.023047427	3.88E-6	222.2487643	41.19528287	1.208067896
2013/1/1 0:00	-180	87.75	1	0	424087.0034	0.020429316	3.88E-6	222.407787	43.76540683	-0.514493936
2013/1/1 0:00	-180	87.0	1	0	424747.0453	0.020429316	3.88E-6	222.5803848	46.0503839	-2.150778923
2013/1/1 0:00	-180	86.25	1	0	425445.0207	0.020429316	3.88E-6	222.7549218	48.13765912	-3.772188592
2013/1/1 0:00	-180	85.5	1	0	426180.9295	0.020429316	3.88E-6	222.9391554	50.12608343	-5.462024724
2013/1/1 0:00	-180	84.75	1	0	426947.1851	0.020429316	3.88E-6	223.1602357	52.09549794	-7.333339737
2013/1/1 0:00	-180	84.0	1	0	427743.7874	0.017811205	3.88E-6	223.458888	54.13714966	-9.463485284

分组大报表

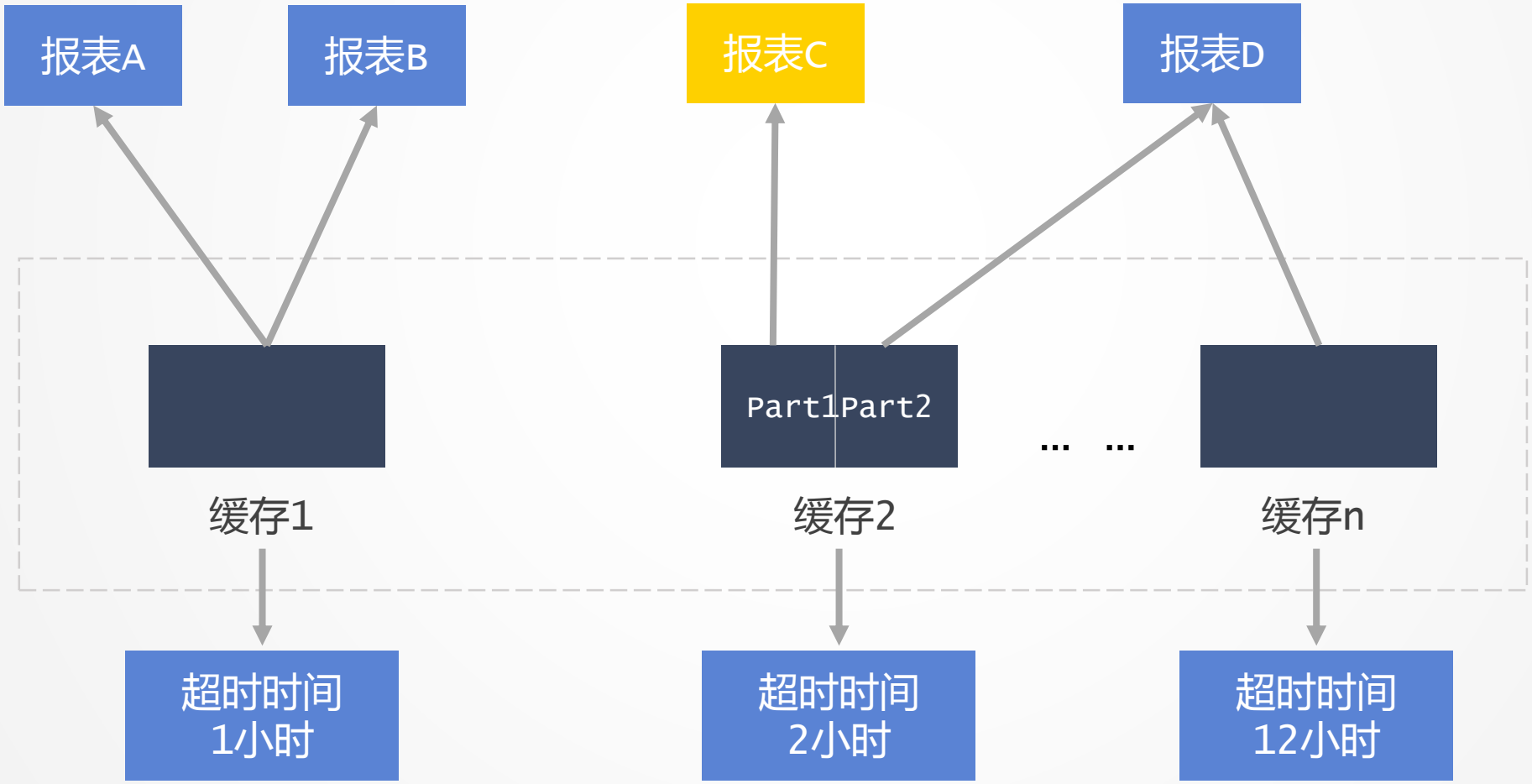
对海量数据进行分组，呈现汇总及明细情况

举例：查询订单数据中按地区分组的明细及订单金额汇总

	A	B	C
1	=connect("db")		
2	=A1.cursor@x("select 货主地区, 订单 ID, 客户 ID, 订购日期, 运货费, 订单金额 from 订单 where 订购日期 >=? and 订购日期 <=? order by 货主地区",begin,end)		/建立数据库游标查询订单数据，并按地区排序
3	for A2; 货主地区	/按地区分组读取数据	
4		=A3.derive(0:flag)	/增加标志位
5		=B4.sum(订单金额)	/分组汇总
6		>B4.insert(0,null,null,null,null,"订单金额小计:",B5,1)	/汇总值添加到明细数据中
7		return B4	/返回明细及分组结果

可控缓存——结构图

可以实现报表的部分缓存、缓存复用、以及不同缓存超时时间



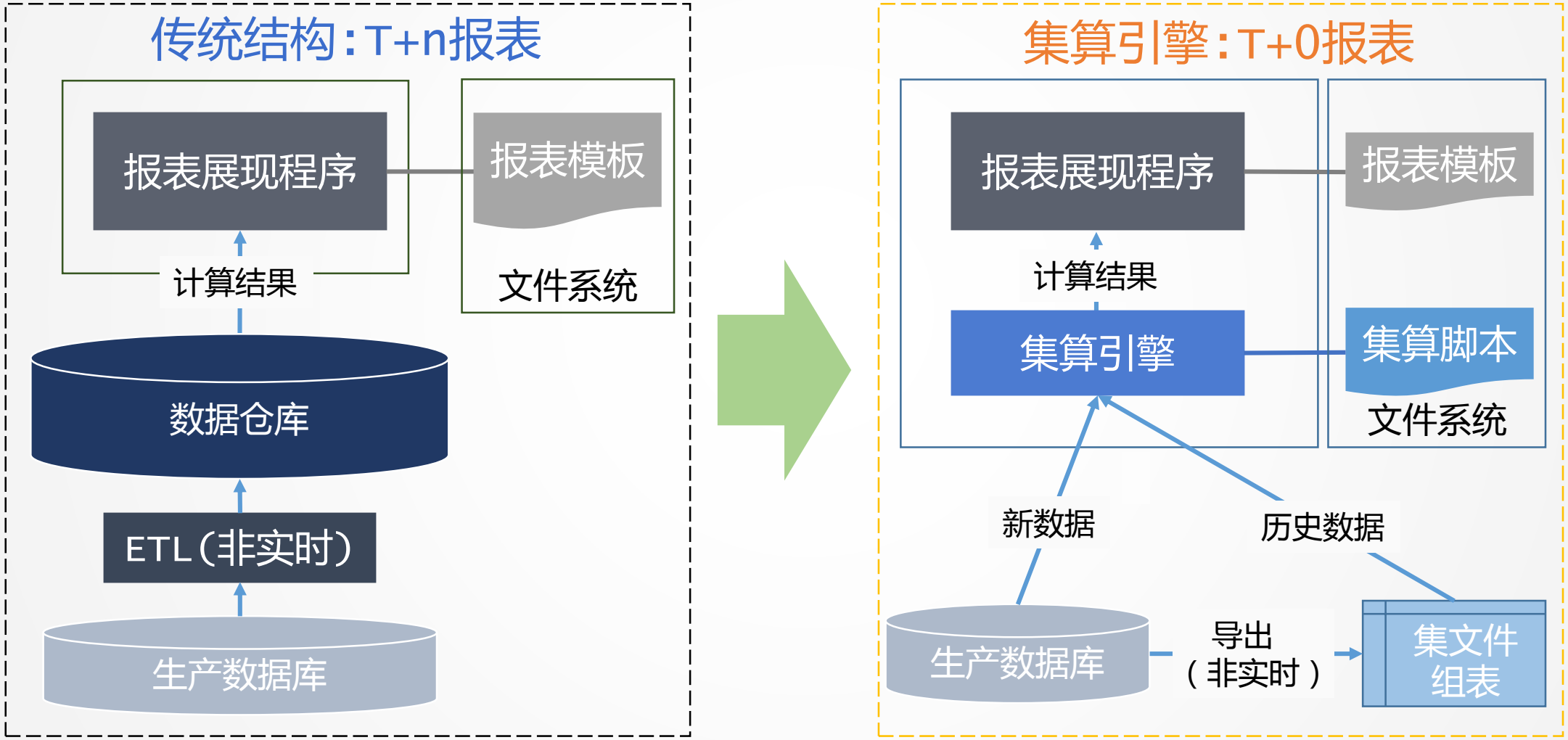
可控缓存

可以实现报表的部分缓存、缓存复用、以及不同缓存超时时间

举例：每次生成缓存文件的一小时内，若查询参数相同，则直接读取缓存文件中的数据，否则从数据库获取数据并生成缓存文件。

	A	B	C
1	E:\\work\\esProc\\	/缓存目录	
2	="sales_"+string(d_date)	/缓存文件名，报表名+参数(dfxName_ParamName)	
3	=file(A1+A2)		
4	if A3.exists() && interval@s(A3.date(),now())<3600	return A3.import@b()	end
5	=connect("demo")		
6	=A5.cursor("SELECT 客户ID,sum(单价*数量) 订单总额 FROM 订单,订单明细 WHERE 订单.订单ID=订单明细.订单ID and year(订购日期)>? group by 客户ID order by 订单总额 desc",d_date)		
7	=A6.fetch(5)	>A5.close()	
8	=A3.export@b(A7)	/生成缓存文件	
9	return A7	/为报表返回结果集	

T+0查询与统计--结构图


传统结构：T+n报表
报表展现程序
报表模板
计算结果
文件系统
数据仓库
ETL（非实时）
生产数据库
集算引擎：T+0报表
报表展现程序
报表模板
计算结果
集算引擎
集算脚本
文件系统
新数据
历史数据
生产数据库
导出
（非实时）
集文件
组表

T+0查询与统计

利用混合数据源能力实现低成本的T+0实时报表

举例：查询“各州销售统计表”，要求查询实时（T+0）数据。

	A
1	=db2.cursor("select s.*,e.name sellername,e.state sellerstate from sales s,employee e where s.sellid=e.eid and days(current date)=days(orderdate)")
2	=file("sales.btx").cursor@b()
3	=[A1,A2].conjx()
4	=A3.groups(SELLERSTATE;count(ORDERID):count,sum(AMOUNT):TOTAL)
5	=A4.sort(TOTAL:-1)
6	return A5

各州销售统计表

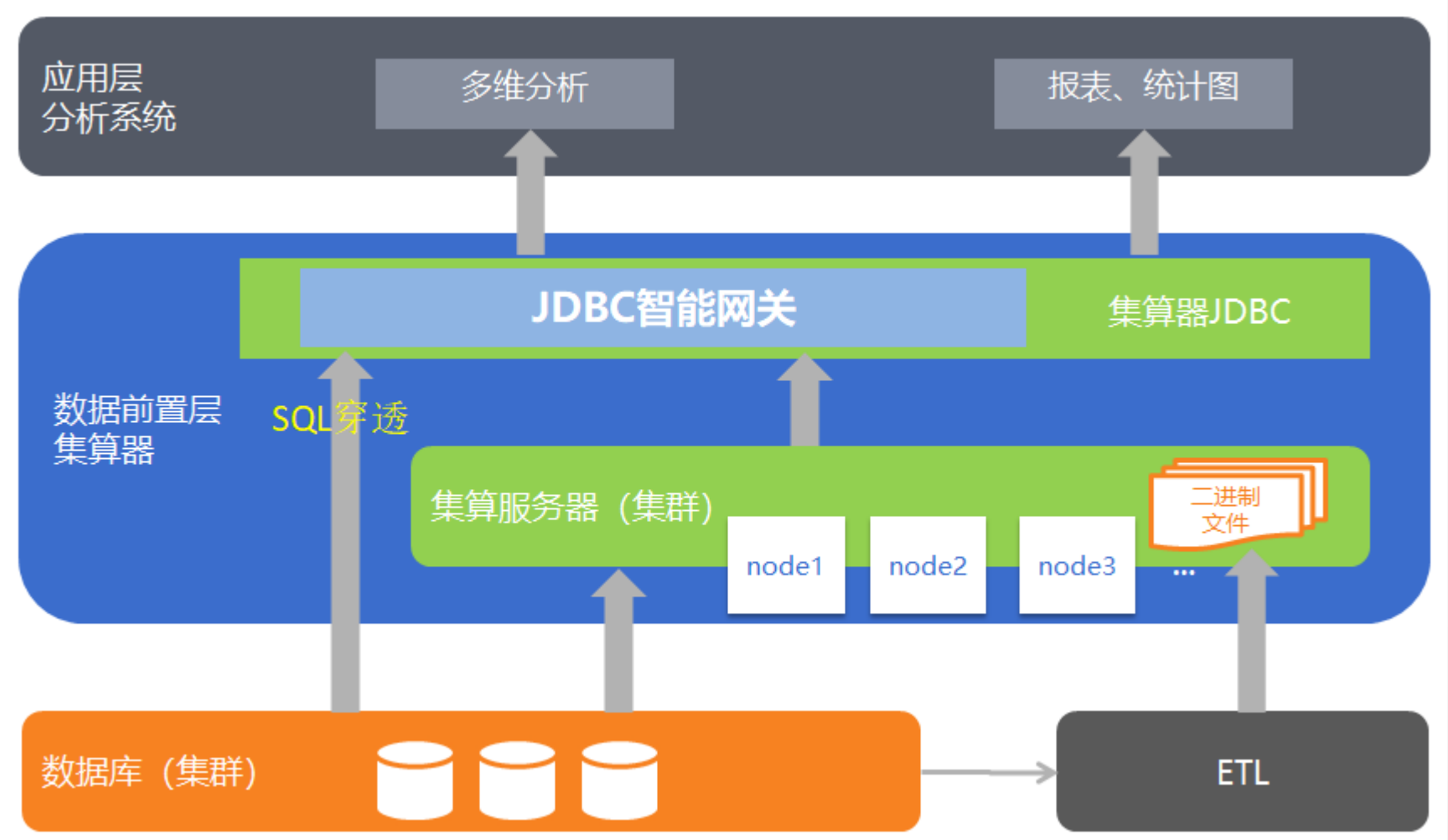
州名称	销售订单数	销售总额
California	34	$80, 492. 50
Texas	35	$72, 957. 90
Florida	35	$72, 089. 00
New York	22	$34, 680. 00
Pennsylvania	16	$16, 346. 10
New Mexico	5	$7, 624. 60

报表展现效果如图所示

A1：建立数据库游标，用简单的sql读取销售数据和销售人员的数据。从where条件中的days(current date)=days(orderdate)可以看到，销售数据仅仅读取当天的新数据。
A2：建立预先导出好的数据文件sales.btx的游标。文件游标可以分批从比较大的数据文件中读取数据，避免出现内存溢出的情况。@b选项是指按照二进制编码来读取文件。
A3：将数据库游标（新数据）和文件游标（历史数据）纵向合并。
A4：利用groups函数，完成对合并后游标的分组汇总。
A5：按照销售总额的降序排列。
A6：返回结果集。

前置数据与路由--结构图

数据前置层将BI系统重构为三层结构



前置数据与路由

集算器JDBC智能网关接收到SQL后，转给gateway.dfx程序处理。gateway.dfx判断是否三年内的查询，如果是，就把表名换成文件名，查本地文件orders.btx返回结果。如果不是，把SQL转换成ORACLE格式，提交数据库处理。

	A	B	C
1	=filename="orders.btx"		
2	=sql.sqlparse@w().split(" ")	=A2.select@1(like(~, "ORDERDATE=date('????-??-?? ') "))	
3	=mid(right(B2,14),3,10)	=year(now())-year(date(A3))	
4	if B3<=3	=connect()	=sql=replace(sql, "from ORDERS","from "+filename)
5		=B4.cursor@x(sql)	return B5
6	else	=connect("orcl")	=sql=sql.sqltranslate("ORACLE")
7		=B6.cursor@x(sql)	return B7

创新技术

推动应用进步！