
ArchData
技术峰会上海站
主办方
中生代技术
快CTO
2017年9月9日 上海徐汇区田林路200号C座一楼COCOSPACE

润乾
RAQSOFT

轻量级大数据计算引擎

创新技术推动应用进步

主讲人:蒋步星

RAQSOFT

目录 contents

沉重的大数据计算

1

集群

单机性能不被关注，

依靠集群规模

2

内存

避开外存计算困难，

指望巨大内存

3

框架

框架体系庞大复杂，

试图包罗万象

轻量级计算需求

大数据的技术本质是高性能

提高性能的需求无处不在

不总是有那么大的数据量

低延迟即时响应业务数据量并不大

不总是适合部署大数据平台

即时查询常常有被集成需求

临时性数据处理来不及建设大数据平台

不总是可以扩容硬件（内存）

需求

大数据计算开发难度大

大数据平台对SQL查询关注过多

性能比拼的主要阵地

优化SQL性能几乎无助于降低开发难度

大量过程计算的开发难度还很大

用SQL很难描述，复杂SQL优化效果不好

仍需大量底层的编码，经常编写UDF

提高性能本质上是降低开发难度

复杂运算的自动优化靠不住，需要快速编写高性能算法

难度大

举例：漏斗转换计算

设备ID	时间戳	事件名称	事件属性	应用标识	日期
1111	1490970560539	搜索商品	{类别=A……}	3zprjdg7yyp5	20170227
2222	1490965747548	浏览商品	{价格=50……}	3zprjdg7yyp5	20170227
3333	1490972189107	提交订单	{……}	3zprjdg7yyp5	20170227
……	……	……	……	……	……

集群透明化

大数据平台努力实现集群透明化

单机与集群一致

网络存储系统+自动任务分配

透明化提高代码兼容性，降低开发难度

透明化难以获得最优性能

高性能计算方案因场景而异，可能是矛盾的

透明化只能选择最保险的方法，一般是性能较差的那个

透明化框架对资源消耗严重

透明化与高性能的权衡

数据分布

节点文件系统：可控冗余，内存利用

网络文件系统

任务分配

程序员分配：根据节点能力安排任务，无框架资源消耗

系统自动分配

轻量级计算的技术特征

目标：过程计算

可集成性

数据源开放性

- 直接文件计算

注重单机优化

- 多线程并行

权衡集群透明与高性能

- 节点文件存储，不用网络文件系统
- 多个单机运算，不用统一集群框架

目录 contents

集算器 – 技术特征

面向过程计算

无缝应用集成

多样性数据源接口 直接文件计算

单机优化技术 多线程并行

无中心集群结构 自由数据分布和任务分配

集算器 – 敏捷语法体系

某支股票最长连续涨了多少交易日

```
1  select max(连续日数)
2  from (select count(*) 连续日数
3          from (select sum(涨跌标志) over(order by 交易日) 不涨日数
4                  from (select 交易日,
5                          case when 收盘价>lag(收盘价) over(order by 交易日)
6                          then 0 else 1 end 涨跌标志
7                  from 股价表) )
8          group by 不涨日数)
```

SQL

	A
1	=股价表.sort(交易日)
2	=0
3	=A1.max(A2=if(收盘价>收盘价[-1],A2+1,0))

集算脚本

思考：按照自然思维怎么做？

语法体系更容易描述人的思路

数据模型不限制高效算法实现

集算器 – 面向过程计算

完整的循环分支控制

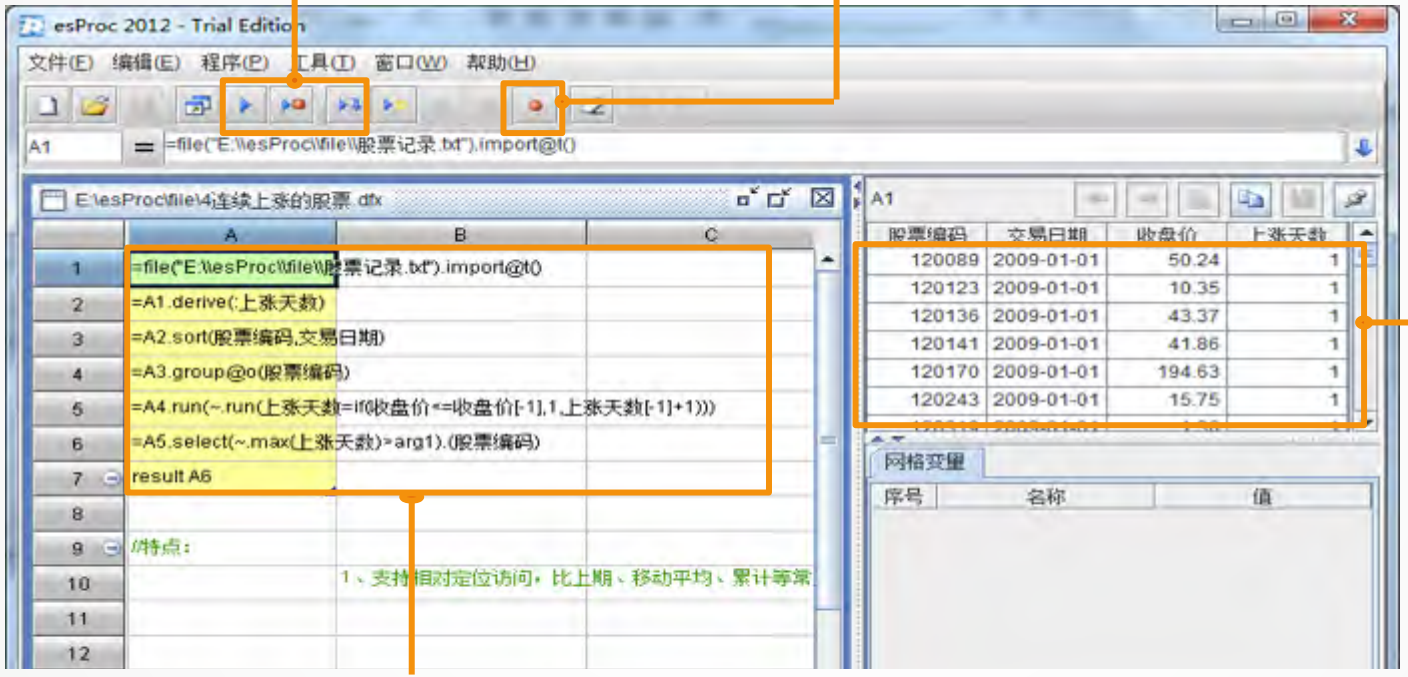
	A	B	C	D	E	F
1	=esProc.query("SELECT 订单ID AS 合同,订购日期 AS 日期,客户 ,订单金额 AS 金额,员工ID AS 销售 FROM 销售记录表 WHERE year(订购日期)=? OR year(订购日					
2	=esProc.query("select * from 员工表")					
3	>A1.run(销售=A2.select@1(编号:A1.销售))		/字段值 是记录			
4	=A1.group(销售)					
5	=create(销售,今年销售额,去年销售额,增长率,客户数,大客户数,大客户占比)					
6	for A4	=A6(1).销售.姓名				
7		=A6.select(year(日期)==年份).sum(金额)		/今年销售额		
8		=A6.select(year(日期)==年份-1).sum(金额)		/去年销售额		
9		=B8/B7-1	/增长率			
10		=A6.group(客户).(~.sum(金额))				
11		=B10.count()	/客户数			
12		=B10.count(~>=B7*0.1)	/大客户数			
13		=B12/B11	/大客户占比			
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**

集算器 – 开发环境

执行、调试执行、单步执行

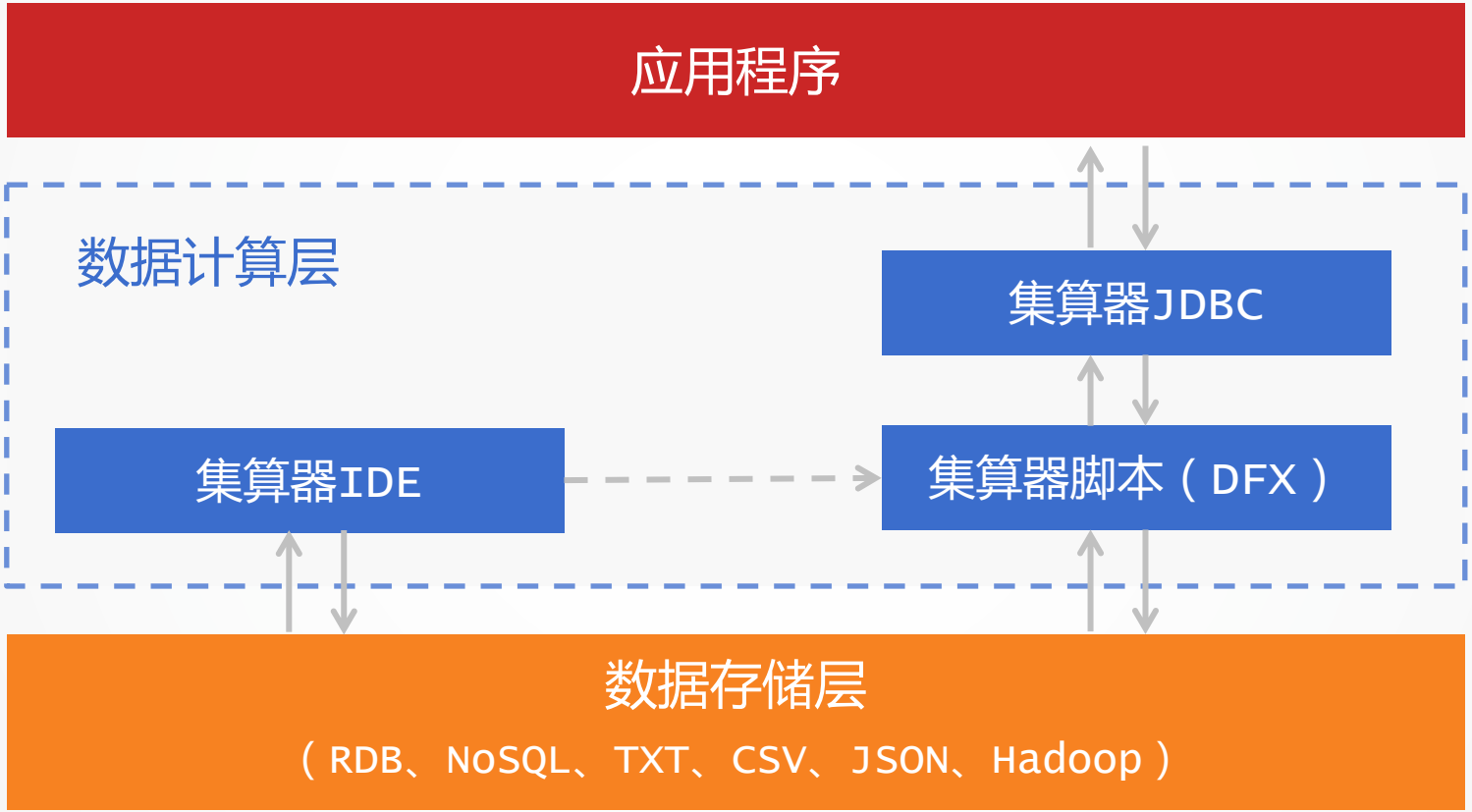
设置断点



网格结果所见即所得，易于调试；方便引用中间结果

语法简单，符合自然思维，比其他高级开发语言更简单

集算器 – 应用结构


应用程序
数据计算层
集算器JDBC
集算器IDE
集算器脚本（DFX）
数据存储层
（RDB、NoSQL、TXT、CSV、JSON、Hadoop）

数据源接口

- 高效二进制压缩文件、列式存储
- RDB：Oracle,DB2,MS SQL,MySQL,PG,....
- TXT/CSV，JSON/XML，EXCEL
- Hadoop：HDFS，HIVE，HBASE
- MongoDB，REDIS，...
- HTTP、ALI-OTS

... ...

- 内置接口，即装即用

目录 contents

单机计算技术

1 遍历技术

2 连接解决

3 存储格式

4 使用索引

5 分段并行

1 遍历技术

遍历是大数据计算的基础

1 遍历技术 延迟游标

游标概念

流式读入数据，每次仅计算一小部分

延迟计算

在游标上定义运算，返回结果仍然是游标，可再定义运算

不立即计算，最终一次性遍历和计算

	A	B
1	=file(“data.txt”).cursor@t()	/创建游标
2	=A1.select(product==“1”)	/过滤
3	=A2.derive(quantity*price:amount)	/计算列
4	=A3.sum(amount)	/实际计算

1 遍历技术 遍历复用

外存计算优化方向是减少访问量

可复用的遍历减少外存访问量

	A	
1	=file(“data.txt”).cursor()	
2	=channel().groups(;count(1))	配置同步计算
3	>A1.push(A2)	绑定
4	=A1.sortx(key)	排序，遍历过程中处理绑定计算
5	=A2.result().#1	取出绑定计算的结果，即总记录数
6	=A4.skip((A5-1)\2).fetch@x(2-A5%2).avg(key)	取出中位数记录并计算中位数

一次遍历可返回多个分组结果

1 遍历技术 聚合理解

从一个集合计算出一个单值或另一个集合都可理解为聚合

高复杂度的排序问题转换为低复杂度的遍历问题

	A	
1	=file(“data.txt”).cursor@t()	
2	=A1.groups(;top(10,amount))	金额在前10名的订单
3	=A1.groups(area;top(10,amount))	每个地区金额在前10名的订单

1 遍历技术 有序游标

针对已有序的数据可一次遍历实现大结果集分组运算，减少外存交换

	A	
1	=file(“data.txt”).cursor@t()	
2	=A1.groupx@o(uid;count(1),max(login))	

复杂处理需要读出到程序内存中再处理

有序游标有效减少查找和遍历数量

	A	B	C
1	=file(“user.dat”).cursor@b()		/按用户id排序的源文件
2	for A1;id	…	/从游标中循环读入数据，每次读出一组id相同
3		…	/处理计算该组数据

2 连接解决

连接计算是结构化数据的最大难点

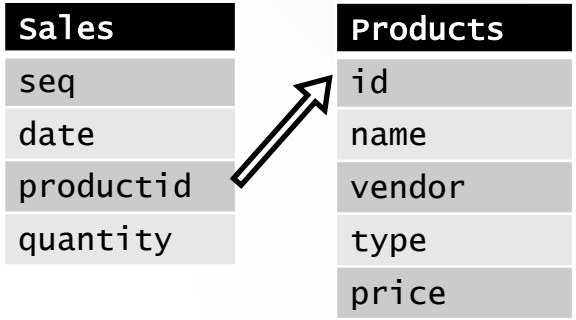
2 连接解决 区分JOIN！

外键维表1:N	同维表1:1	主子表1:N
指针化 序号化	有序归并	有序归并

2 连接解决 外键指针化

外键需要随机小量频繁访问

内存指针查找大幅提高性能



	Java指针连接	Oracle
单表无连接	0.57s	0.623s
五表外键连接	2.3s	5.1s

	A	
1	=file(“Products.txt”).import()	读入商品列表
2	=file(“Sales.txt”).import()	读入销售记录
3	>A2.switch(productid,A1:id)	建立指针式连接，把商品编号转换成指针
4	=A2.sum(quantity*productid.price)	计算销售金额，用指针方式引用商品单价

2 连接解决 外键序号化

序号化相当于外存指针化

	A	
1	=file(“Products.txt”).import()	读入商品列表
2	=file(“Sales.txt”).cursor()	根据已序号化的销售记录建立游标
3	=A2.switch(productid,A1:#)	用序号定位建立连接指针，准备遍历
4	=A3.groups(;sum(quantity*productid.price))	计算结果

不需要再计算Hash值和比较

2 连接解决 有序归并

同维表和主子表连接可以先排序后变成有序归并

追加数据的再排序也仍然是低成本的归并计算

	A	
1	=file(“Order.txt”).cursor@t()	订单游标，按订单id排序
2	=file(“Detail.txt”).cursor@t()	订单明细游标，也按订单id排序
3	=joinx(A1:O,id;A2:D,id)	有序归并连接，仍返回游标
4	=A3.groups(O.area;sum(D.amount))	按地区分组汇总金额，地区字段在主表中，金额字段在明细子表中

3 存储格式

有效的存储格式是高性能的保证

3 存储格式 压缩二进制

数据类型已存入，无须解析

轻量级压缩

减少硬盘空间
很少占用CPU时间

泛型存储，允许集合数据

可追加

3 存储格式 自由列存

自由分配列组

行列存统一

重复值压缩

对上透明访问

组1	组2		组k	
列1	列2	列3	列4	…
1	…	…	…	
2	…	…	…	
…				
1023				
1024				

过滤优化

只读取与条件相关的列组

3 存储格式 序号主键

多层序号式主键

外存指针式外键，高速引用

外存游离记录表示，离散性

天然有序，易于查找

分组针对外键

结果自然对齐有序

北京		1
	海淀	1,1
	朝阳	1,2
	…	
上海		2
	浦东	2,1
	…	
…		

3 存储格式 主子合一

多层复式表

层次式有序集合

每层均可以有数据结构

同维表与主子表统一

消除对齐式连接

北京		面积	人口	GDP	…
	海淀				
	朝阳				
	…				
上海					
	浦东				
	…				
…					

订单		客户	地区	日期	…
	订单明细	产品	数量	单价	…

4 使用索引

有序对分查找

有序数据提供对分查找，快速定位

普通定位索引

按键值找到数据

两段式索引提高维护性能

片状索引

连续记录的索引

4 使用索引 应用：切片的双向逆序索引

CUBE切片索引的困难

列存索引太大

数据必须实际排序

双向逆序索引

按D1,…,Dn和Dn,…,D1双倍排序

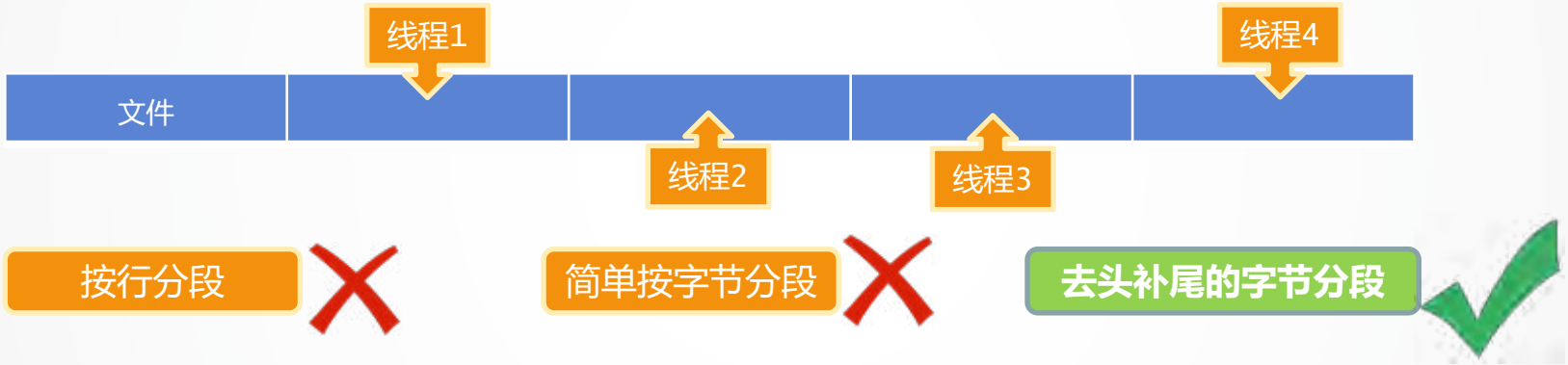
只针对前半部分维度使用片式索引

5 分段并行

数据分段是并行计算的基础

5 分段并行 文本分段

并行处理需要将数据文件分段，每个线程处理一段



文本并行解析

	A	
1	=file(“data.txt”).cursor@tm(amount)	/定义并行取数的游标（并行）
2	=A1.groups(;sum(amount):amount)	/遍历游标汇总amount（串行）

5 分段并行 倍增分段

分段数足够大，以适应变化的并行数

每段记录数相对平均

数据追加时不必重写所有数据，保持连续性

	A	B	
1	=file(“data.bin”)		
2	fork 4	=A1.cursor@b(amount;A2:4)	
3		=B2.groups(;sum(amount):a)	
4	=A2.conj().sum(a)		

追加前	追加后
1	1
2	
3	2
4	
5	3
6	
7	4
8	
…	
1023	512
1024	
	513
	…

5 分段并行 列存分段

倍增分段方案解决列存并行困难

各列同步分段

同列连续存储

段	列1	列2	…
1	1,…,k	1,…,k	
2	k+1,…,2k	k+1,…,2k	
3	2k+1,…,3k	2k+1,…,3k	
4	3k+1,…,4k	3k+1,…,4k	
…			
1023			
1024			

	A	B	
1	=[1,3].(file(“col”/~/”.bin”))		
2	fork 4	=A1.cursor(amount;A2:4)	
3		=B2.groups(;sum(amount):a)	
4	=A2.conj().sum(a)		

5 分段并行 有序与对位分段

有序数据的分段点要落在组边界上才能分段并行

	A	
1	=file(“userlog.txt”).cursor@t()	原始数据游标
2	=A1.sortx(id)	按id排序
3	>file(“userlog.dat”).export@z(A2;id)	写成按id分段的文件

同维表、主子表需同步分段

	A	
1	=file(“Order.txt”).cursor@t()	
2	=A1.sortx(id)	按id排序
3	>file(“Order.dat”).export@z(A2;id)	写成按id分段的文件
4	=file(“Detail.txt”).cursor@t()	
5	=A4.sortx(id)	按id排序
6	>file(“Detail.dat”).export@z(A2;id,file(“Order.dat”),id)	和Order.dat同步按id分段

5 分段并行 多路游标

建立多路游标，可继续进行过程计算，会自动并行执行，简化书写难度。

	A	
1	=file(“Order.txt”).cursor@tm()	建立多路游标，自动并行执行
2	=A1.select(month(Date)==10)	条件过滤
3	=A2.groups(ID;sum(COST*WEIGHT):VALUE)	分组汇总

目录 contents

集群方案

集群设计原则

确保容错

减少网络传输量

考虑负载均衡

集群计算技术

集群方案 – 无中心设计

服务器集群无中心

集算器没有框架，没有永久的中心主控节点，程序员用代码控制参与计算的节点

无中心不会发生单点失效

所有节点地位都等同，不会发生单点失效，某个节点有故障时整个集群仍然可以运行

计算任务有主控节点

在计算过程中由主控节点临时寻找其它节点参与计算

集群方案 – 负载均衡与容错

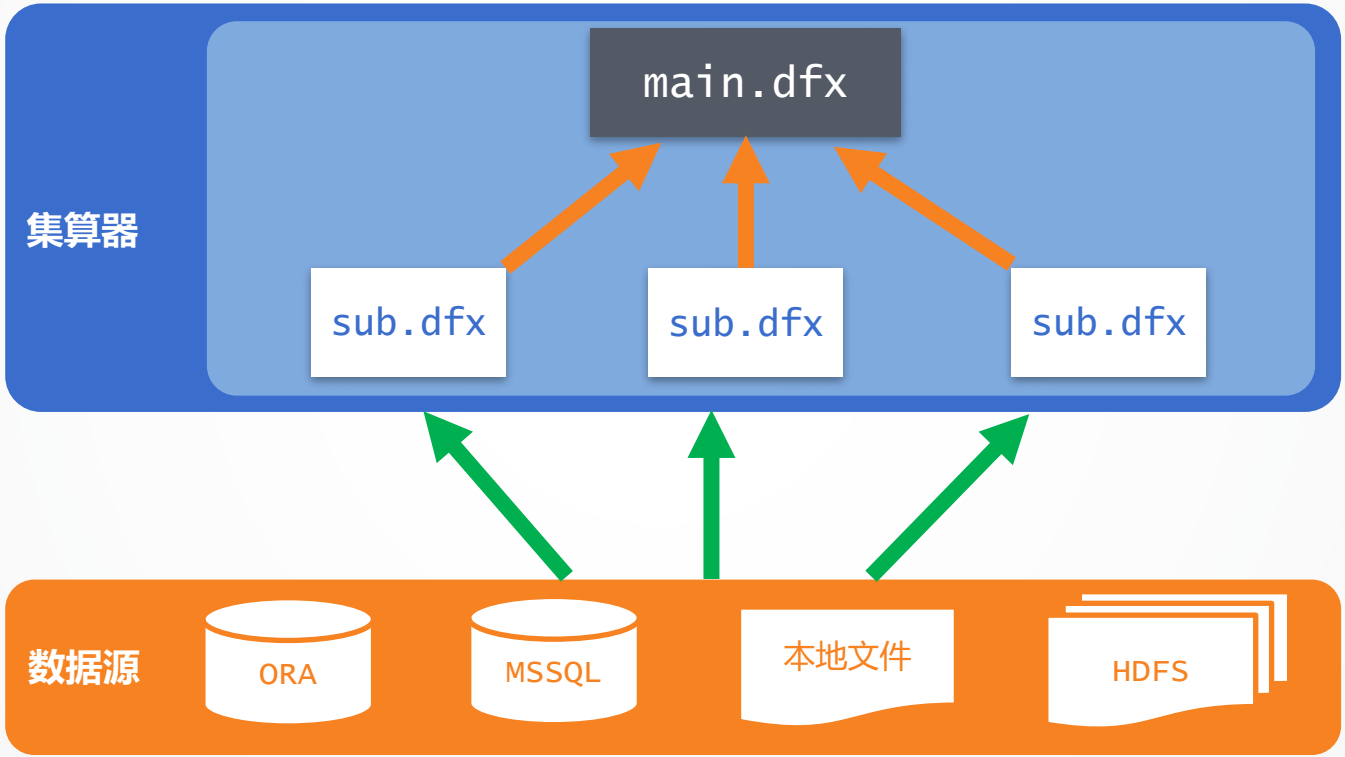
具备负载均衡能力

根据每个节点空闲程度（当前正在运行的线程数量）决定是否分配任务，实现负担和资源的有效平衡

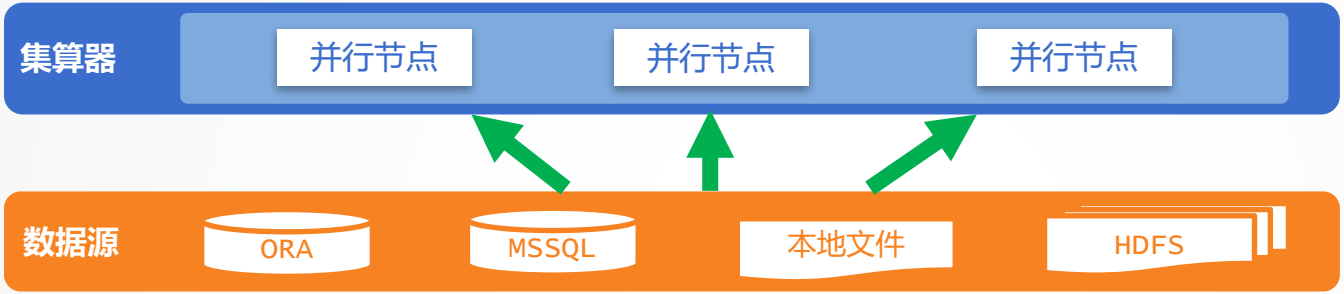
具备容错能力

某个节点失效导致子任务失败，主控程序还会再次将这个子任务分配给其它有效的节点

集群方案 – 并行逻辑结构



共享式数据计算分布



共享数据源方式：计算分布实现，数据共享读取

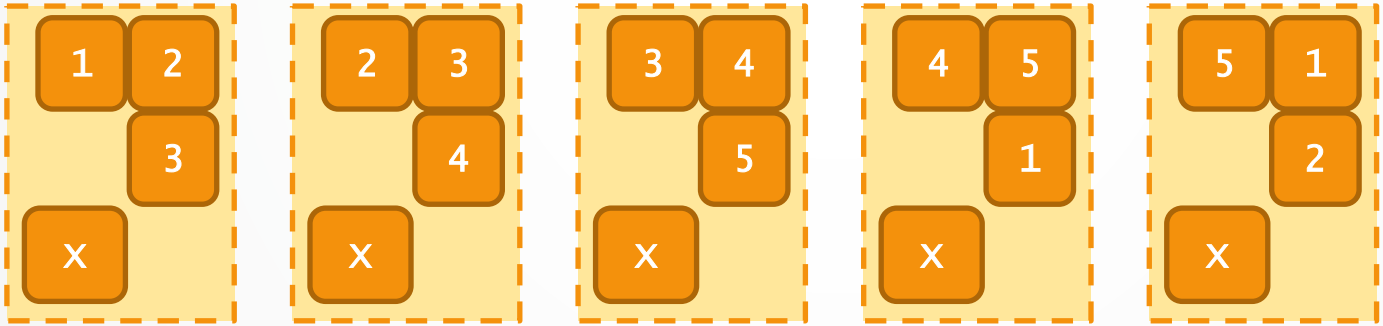
	A	B	
1	=4.(“192.168.0.”/(10+~)/”:1234”)		节点机列表，4个
2	fork to(8);A1		到节点机上执行，分成8个任务
3		=hdfsfile(“hdfs:\\192.168.0.1\persons.txt”)	HDFS上的文件
4		=B3.cursor@t(;A2:8)	分段游标
5		=B4.select(gender==’M’).groups(;count(1):C)	过滤并计数
6	=A2.conj().sum(C)		汇总结果

冗余式外存分布

所有任务都需要用到的公共维表复制存储

事务数据分成N个区，根据需要的容错指数循环分区

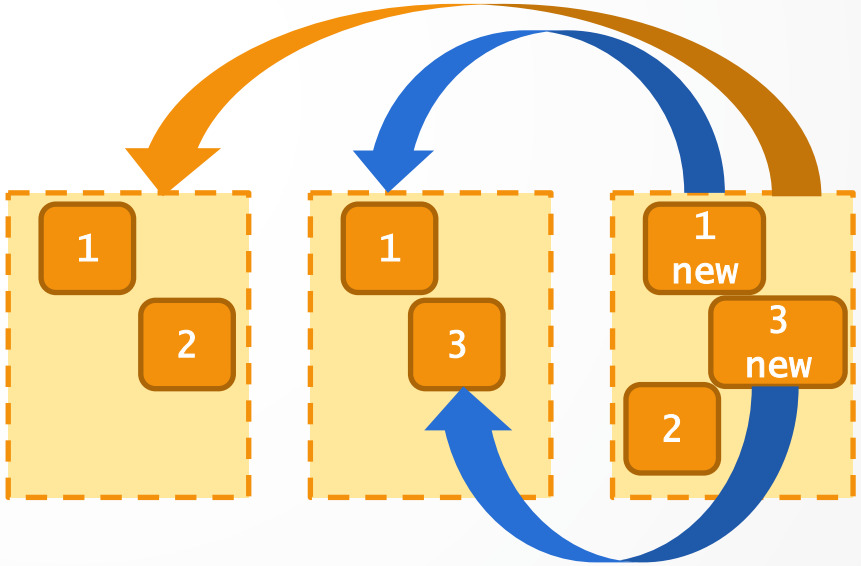
存储利用率约为1/k（允许k-1个节点失效）



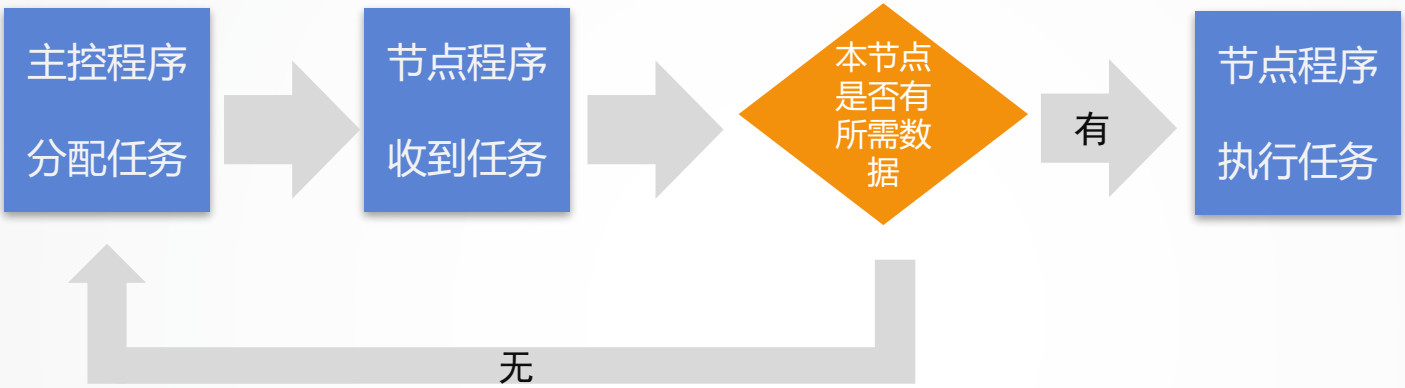
外存分布 – 数据同步

集算器提供节点之间的同区数据同步功能

每个节点都是独立的计算机，可以应用内存和外存的计算方法。



外存分布 – 动态任务分配



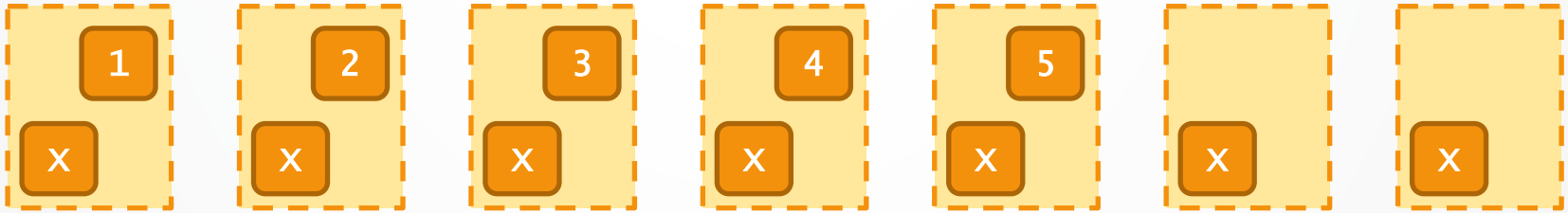
	A	B	C	
1	=4.("192.168.0."/(10+~)/":1234")			节点机列表，4个
2	fork to(8);A1			到节点机上执行，分成8个任务
3		=file("person.txt",A2)		A2为数据区号
4		if !B3.exists()	end "data not find"	找不到返回错误再分配
5		=B3.cursor()		
6		=B5. select(gender=='M').groups(;count(1):C)		计算
7	=A2.conj().sum(C)			汇总

备胎式内存分布

数据分区分别加载进节点内存

n个节点（每节点一个分区）+k个备份节点

内存利用率n/(n+k)



内存分布 – 静态任务分配

预留备份节点（备胎）不加载任何数据分区

发现有分区在所有可用节点都找不到时，启动找备份节点执行加载该分区

任务直接分配到相应节点，不再动态询问

	A	主程序
1	=8.("192.168.0."/(10+~)/":1234")	节点列表，共8个
2	=hosts(A1,to(4),"init.dfx")	在其中寻找4个节点加载内存数据
3	=callx@a("sub.dfx",to(4);A2)	调用这些节点上程序计算
4	=A3.sum()	汇总结果

	A	节点初始化程序init.dfx
1	=file("Product.txt",z).import()	读入第z分区的数据
2	>env(T,A1)	将数据记入环境
3	>zone(z)	登记本节点的分区

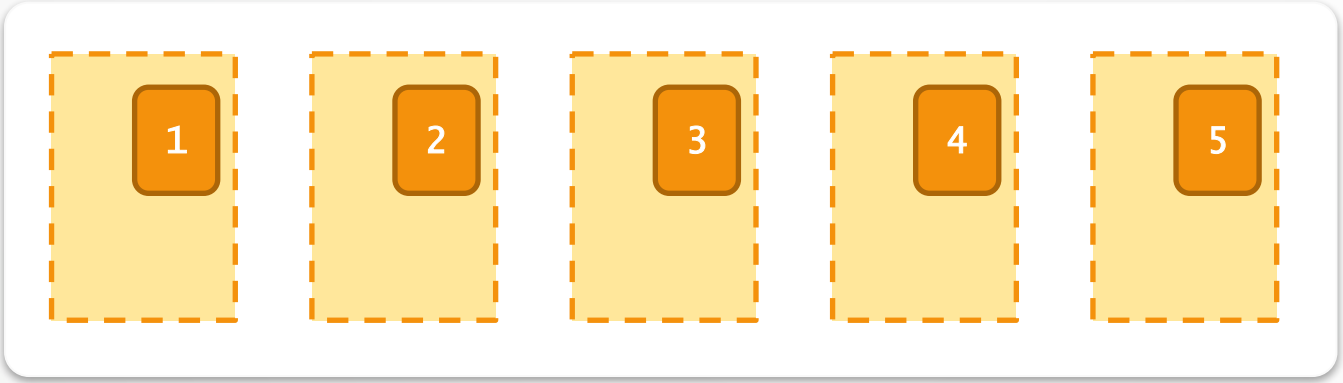
	A	节点程序sub.dfx
1	=T	从环境中取出数据
2	=A1.count(gender=='M')	过滤并计数
3	return A2	返回结果

集群维表

大维表分段存入节点机内存

利用内存特性提供随机访问

批量访问+节点过滤降低网络负担



JOIN总结

外键式：小维表、大维表

对齐式：有序归并

避免低效率Hash分段算法

集群维表 – 示例

	A	主程序
1	=8.(“192.168.0.”/(10+~)/”:1234”)	节点机列表，共8个
2	=hosts(A1,to(4),”init.dfx”)	选择4个节点加载内存维表
3	=callx(“sub.dfx”,8*A2,to(8);A1)	调用节点程序计算，传入内存维表节点
4	=A3.sum()	汇总结果

	A	节点初始化程序init.dfx
1	=file(“Product.txt”,z).import()	读入第z分区的维表
2	>env(T,A1)	将数据记入环境
3	>zone(z)	登记本节点的分区

	A	B	节点程序sub.dfx
1	=file(“Sales.txt”,z)		z区事实数据
2	if !A1.exists()	end “data not find”	找不到返回错误
3	=createx(T,h)		基于节点组h上变量T建立集群维表
4	=A1.cursor()		
5	=A4.join(productid,A3,price)		与集群维表A3做连接计算
6	=A5.sum(quantity*price)		合计
7	return A6		返回结果

MapReduce->ForkReduce

MapReduce的问题

任务太碎，管理成本过高

难以实现关联运算

Hash Shuffle随意性太强

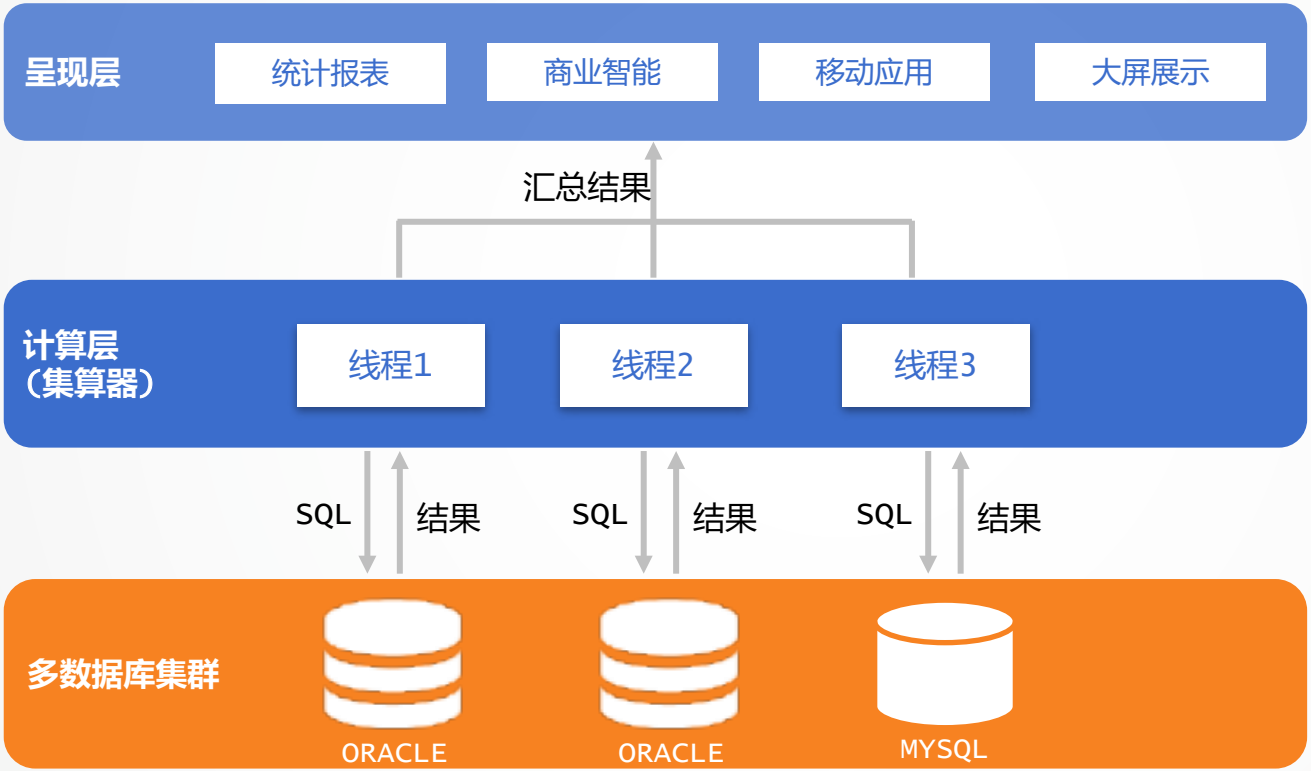
ForkReduce

批量任务，调度成本低

结合对位分段技术实现关联运算

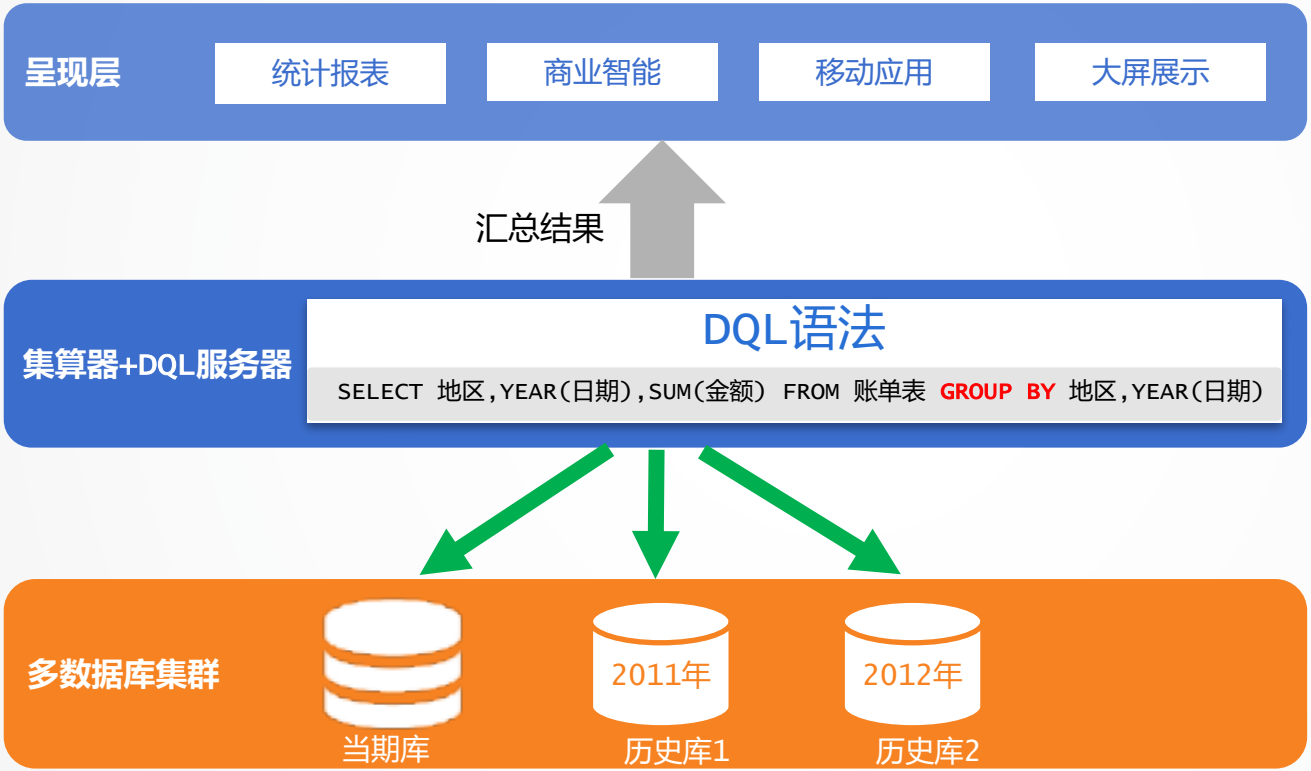
Shuffle结果有确定分布方案

集群应用 – 异构数据库集群



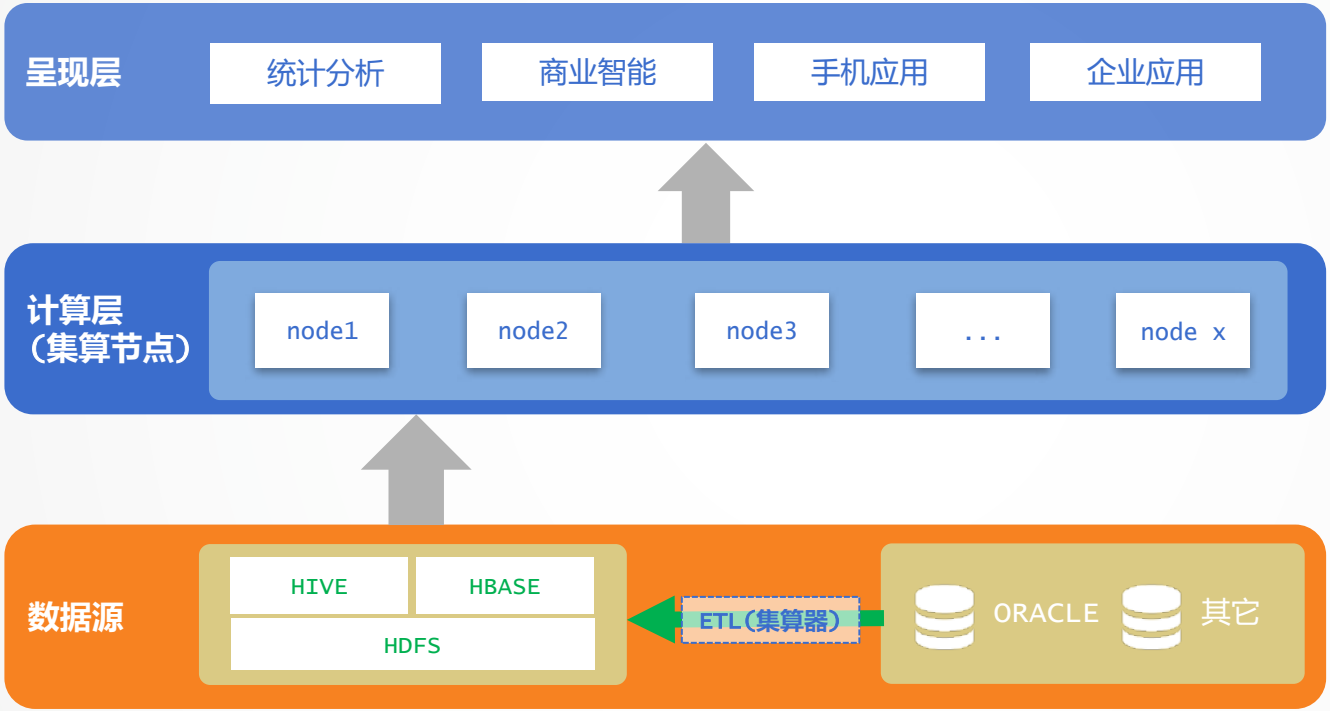
集算器可以基于同构或异构数据库集群进行结果汇总，为报表输出汇总后的结果集

集群应用 – 异构数据库集群(联合查询)



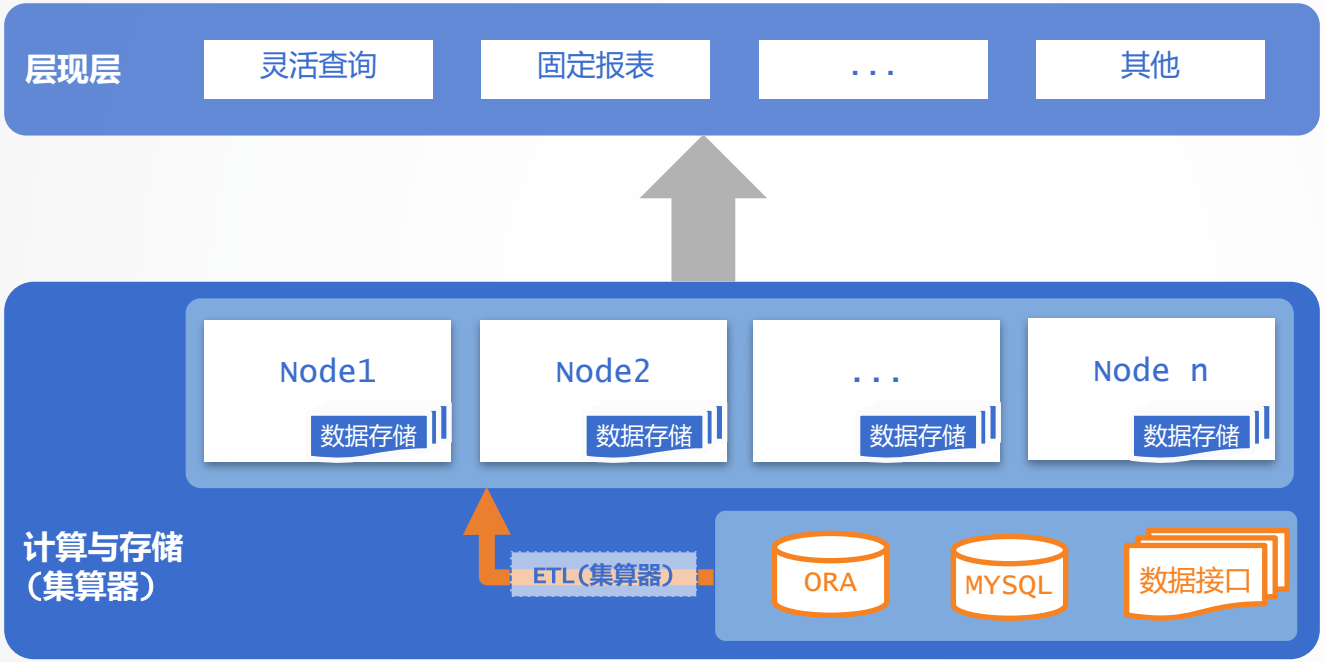
集算器与报表V5合作提供DQL语言，可基于异构数据库集群进行联合查询，结果汇总后为上层应用提供数据输出服务

集群应用 – 协助HADOOP工作



集算器是轻量级、高性能、分布式的计算引擎，解决HADOOP上SQL能力弱、开发难、运算慢的问题，可以极大地缩减大数据平台的搭建周期和实施成本

集群应用 – 独立工作



集算器作为独立的大数据计算引擎，可自己管理数据，不需要网络文件系统，通过标准的接口为上层应用提供数据输出服务

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

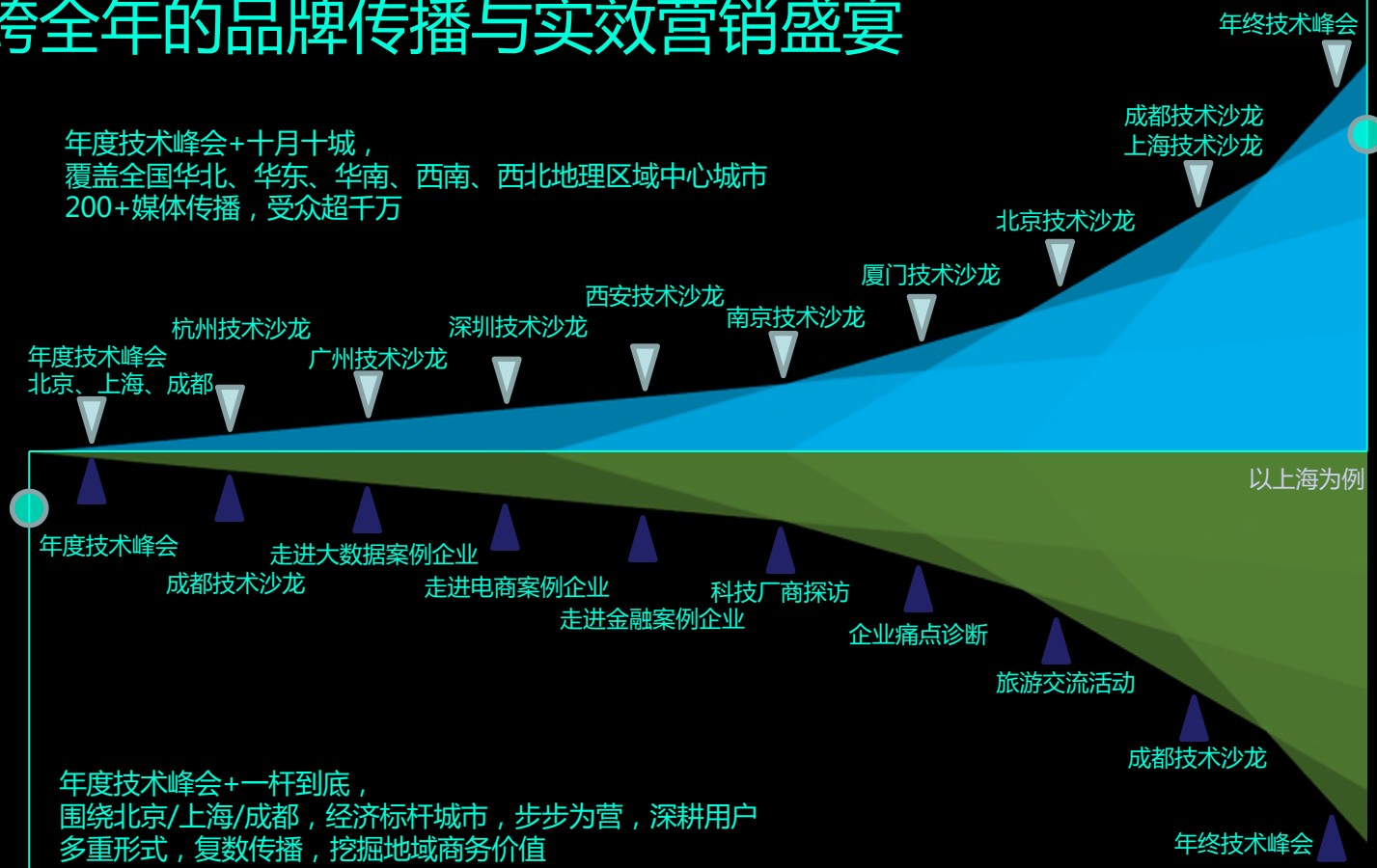
- ✓ 技术干货分享
- ✓ 每周一期
- ✓ 微信直播交流

创新技术 推动应用进步

www.raqsoft.com.cn

谢谢大家！

横跨全年的品牌传播与实效营销盛宴



年度技术峰会+十月十城，
覆盖全国华北、华东、华南、西南、西北地理区域中心城市
200+媒体传播，受众超千万

年度技术峰会+一杆到底，
围绕北京/上海/成都，经济标杆城市，步步为营，深耕用户
多重形式，复数传播，挖掘地域商务价值

合纵连横，在中国开发者群体中缔造品牌营销奇迹

中生代技术
FRESHMAN TECHNOLOGY

中生代技术，打造本地化线下技术交流的最后一公里！

总覆盖会员30000+人；资深架构、总监等职位以上3000+人。

定期在线分享近80次

线下技术沙龙超过70次、覆盖北上广深、杭州、南京、厦门、成都、重庆、西安等城市！

即将推出技术服务、培训板块！！！

联系人微信：zsdwyq

预告：ArchData峰会（北京）精彩内容抢先看！9月24日

关键词：AI、区块链、大数据、智能运维、深度学习!!!

09:00–09:50	史凯	AI驱动的企业创新架构
09:50–10:40	李艳鹏	区块链原理解析
10:40–10:50	短歇	
10:50–11:40	王东	微服务下的APM全链路监控
13:30–14:20	严静	漏斗转换运算的优化过程
13:30–14:20	吴金龙	深度学习与智能对话机器人
14:20–15:10	裴丹	智能运维中的AI问题
15:10–15:20	短歇	
15:20–16:10	涂威威	Towards AI for Everyone
16:10–16:50	余军	分布式数据库在金融行业的创新实践
16:50–18:00	何文斌	主题待定

报名链接：http://t.cn/RN22yfP

ArchData技术峰会上海站

ArchData上海站