

原

基于hadoop生态圈的数据仓库实践 —— OLAP与数据可视化（二）

2016年08月19日 12:35:08

wzy0623

阅读数：25707

更多

版权声明：本文为博主原创文章，未经博主允许不得转载。 <https://blog.csdn.net/wzy0623/article/details/52249187>

二、Hive、Spark SQL、Impala比较

Hive、Spark SQL和Impala三种分布式SQL查询引擎都是SQL-on-Hadoop解决方案，但又各有特点。前面已经讨论了Hive和Impala，本节先介绍一下Spark SQL，然后使用场景几个角度比较这三款产品的异同，最后附上分别由cloudera公司和SAS公司出示的关于这三款产品的性能对比报告。

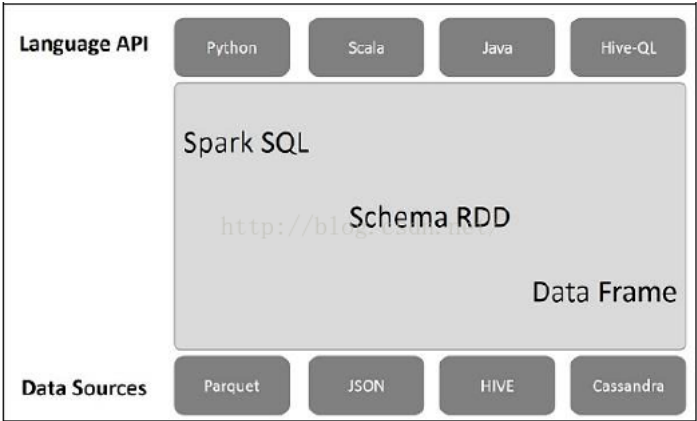
1. Spark SQL简介

Spark SQL是Spark的一个处理结构化数据的程序模块。与其它基本的Spark RDD API不同，Spark SQL提供的接口包含更多关于数据和计算的结构信息，从而可以执行优化。可以通过SQL和数据集API与Spark SQL交互，但无论使用何种语言或API向Spark SQL发出请求，其内部都使用相同的执行引擎，这种统一性使得Spark SQL会更容易被开发者在行切换。

Spark SQL具有如下特性：

- 集成——将SQL查询与Spark程序无缝集成。Spark SQL可以将结构化数据作为Spark的RDD（Resilient Distributed Datasets，弹性分布式数据集）进行查询，并整合了Scala、Java、Python、R等多种集成可以使开发者只需运行SQL查询就能完成复杂的分析算法。
- 统一数据访问——通过Schema-RDDs为高效处理结构化数据而提供的单一接口，Spark SQL可以从Hive表、parquet或JSON文件等多种数据源查询数据，也可以向这些数据源装载数据。
- 与Hive兼容——已有数据仓库上的Hive查询无需修改即可运行。Spark SQL复用Hive前端和元数据存储，与已存的Hive数据、查询和UDFs完全兼容。
- 标准的连接层——使用JDBC或ODBC连接。Spark SQL提供标准的JDBC、ODBC连接方式。
- 可扩展性——交互式查询与批处理查询使用相同的执行引擎。Spark SQL利用RDD模型提供容错和扩展性。

Spark SQL架构如下图所示。



此架构包括Language API、Schema RDD、Data Sources三层。

- Language API——Spark SQL与多种语言兼容，并提供这些语言的API。
- Schema RDD——Schema RDD是存放列Row对象的RDD，每个Row对象代表一行记录。Schema RDD还包含记录的结构信息（即数据字段），它可以利用结构信息高效地存储数据。Schema RDD还支持各种操作。
- Data Sources——一般Spark的数据源是文本文件或Avro文件，而Spark SQL的数据源却有所不同。其数据源可能是Parquet文件、JSON文档、Hive表或Cassandra数据库。

2. Hive、Spark SQL、Impala比较

(1) 功能

Hive：

- 是简化数据抽取、转换、装载的工具
- 提供一种机制，给不同格式的数据加上结构
- 可以直接访问HDFS上存储的文件，也可以访问HBase的数据
- 通过MapReduce执行查询
- Hive定义了一种叫做HiveQL的简单的类SQL查询语言，用户只要熟悉SQL，就可以使用它查询数据。同时，HiveQL语言也允许熟悉MapReduce计算框架的程序员添加定制的mapper和reducer插件功能不支持的复杂分析。
- 用户可以定义自己的标量函数（UDF）、聚合函数（UDAF）和表函数（UDTF）
- 支持索引压缩和位图索引
- 支持文本、RCFile、HBase、ORC等多种文件格式或存储类型
- 使用RDBMS存储元数据，大大减少了查询执行时语义检查所需的时间
- 支持DEFLATE、BWT或snappy等算法操作Hadoop生态系统内存储的数据
- 大量内建的日期、数字、字符串、聚合、分析函数，并且支持UDF扩展内建函数。

Spark SQL：		
• 支持Parquet、Avro、Text、JSON、ORC等多种文件格式		
• 支持存储在HDFS、HBase、Amazon S3上的数据操作		0
• 支持snappy、lzo、gzip等典型的Hadoop压缩编码方式		
• 通过使用"shared secret"提供安全认证		8
• 支持Akka和HTTP协议的SSL加密		
• 保存事件日志		
• 支持UDF		
• 支持并发查询和作业的内存分配管理（可以指定RDD只存内存中、或只存磁盘上、或内存和磁盘都存）		
• 支持把数据缓存在内存中		
• 支持嵌套结构		
Impala：		
• 支持Parquet、Avro、Text、RCFile、SequenceFile等多种文件格式		
• 支持存储在HDFS、HBase、Amazon S3上的数据操作		
• 支持多种压缩编码方式：Snappy（有效平衡压缩率和解压缩速度）、Gzip（最高压缩率的归档数据压缩）、Deflate（不支持文本文件）、Bzip2、LZO（只支持文本文件）		
• 支持UDF和UDAF		
• 自动以最有效的顺序进行表连接		
• 允许定义查询的优先级排队策略		
• 支持多用户并发查询		
• 支持数据缓存		
• 提供计算统计信息（COMPUTE STATS）		
• 提供窗口函数（聚合 OVER PARTITION, RANK, LEAD, LAG, NTILE等等）以支持高级分析功能		
• 支持使用磁盘进行连接和聚合，当操作使用的内存溢出时转为磁盘操作		
• 允许在where子句中使用子查询		
• 允许增量统计——只在新数据或改变的数据上执行统计计算		
• 支持maps、structs、arrays上的复杂嵌套查询		
• 可以使用impala插入或更新HBase		
(2) 架构		
Hive：		
构建在Hadoop之上，查询管理分布式存储上的大数据集的数据仓库组件。底层使用MapReduce计算框架，Hive查询被转化为MapReduce代码并执行。生产环境建议做元数据。支持JDBC、ODBC、CLI等连接方式。		
Spark SQL：		
底层使用Spark计算框架，提供有向无环图，比MapReduce更灵活。Spark SQL以Schema RDD为核心，模糊了RDD与关系表之间的界线。Schema RDD是一个由Row D，附带包含每列数据类型的结构信息。Spark SQL复用Hive的元数据存储。支持JDBC、ODBC、CLI等连接方式，并提供多种语言的API。		
Impala：		
底层采用MPP技术，支持快速交互式SQL查询。与Hive共享元数据存储。Impalad是核心进程，负责接收查询请求并向多个数据节点分发任务。statestored进程负责监进程，并向集群中的节点报告各个Impalad进程的状态。catalogd进程负责广播通知元数据的最新信息。		
(3) 场景		
Hive：		
适用场景：		
• 周期性转换大量数据，例如：每天晚上导入OLTP数据并转换为星型模式；每小时批量转换数据等。		
• 整合遗留的数据格式，例如：将CSV数据转换为Avro；将一个用户自定义的内部格式转换为Parquet等。		
不适用场景：		
• 商业智能，例如：与Tableau结合进行数据探查；与Micro Strategy一个出报表等。		
• 交互式查询，例如：OLAP查询。		
Spark SQL：		
适用场景：		
• 从Hive数据仓库中抽取部分数据，使用Spark进行分析。		
不适用场景：		
• 商业智能和交互式查询。		

Impala：	
适用场景：	
<div><div></div><div>秒级的响应时间</div></div>	0
<div><div></div><div>OLAP</div></div>	8
<div><div></div><div>交互式查询</div></div>	
不适用场景：	
<div><div></div><div>ETL</div></div>	
<div><div></div><div>UDAF</div></div>	

3. Hive、SparkSQL、Impala性能对比

(1) cloudera公司2014年做的性能基准对比测试，原文链接：<http://blog.cloudera.com/blog/2014/09/new-benchmarks-for-sql-on-hadoop-impala-1-4-wide>，先看一下测试结果：

- 对于单用户查询，Impala比其它方案最多快13倍，平均快6.7倍。
- 对于多用户查询，差距进一步拉大：Impala比其它方案最多快27.4倍，平均快18倍。

下面看看这个测试是怎么做的。

配置：

所有测试都运行在一个完全相同的21节点集群上，每个节点只配有64G内存。之所以内存不配大，就是为了消除人们对于Impala只有在非常大的内存上才有好性能的错觉。

- 双物理CPU，每个12核，Intel Xeon CPU E5-2630L 0 at 2.00GHz
- 12个磁盘驱动器，每个磁盘932G，1个用作OS，其它用作HDFS
- 每节点64G内存

对比产品：

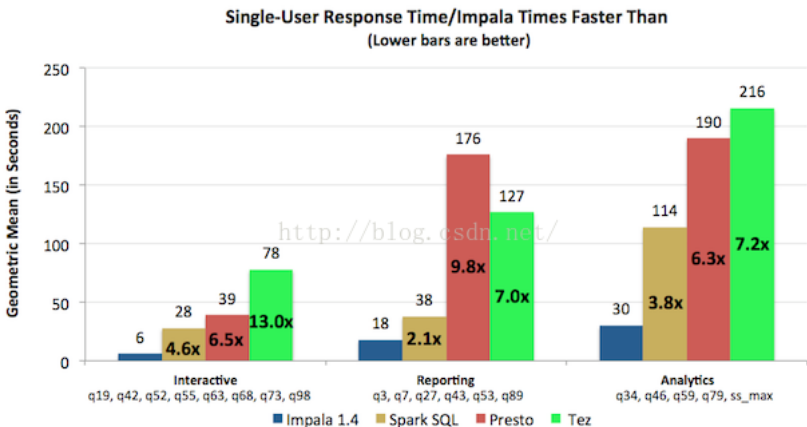
- Impala 1.4.0
- Hive-on-Tez 0.13
- Spark SQL 1.1
- Presto 0.74

查询：

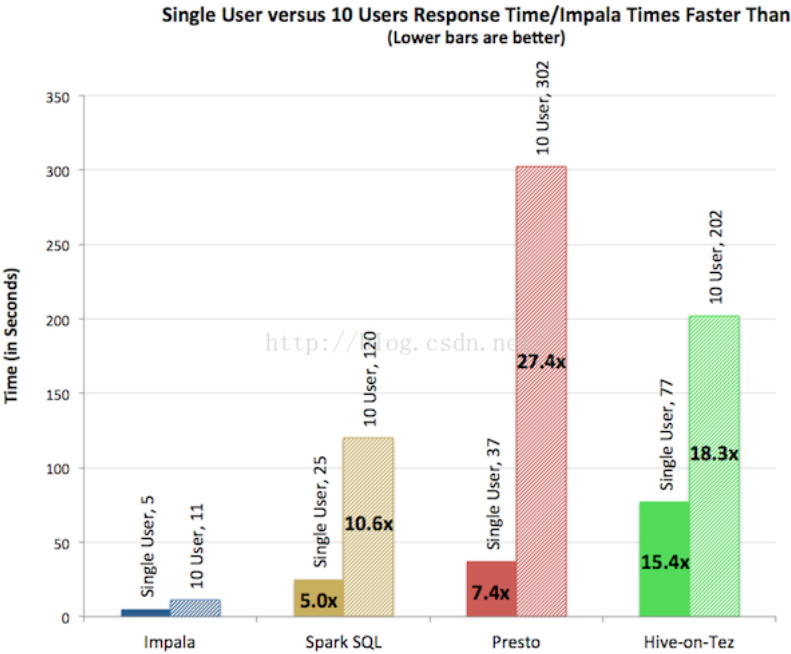
- 21个节点上的数据量为15T
- 测试场景取自TPC-DS，一个开放的决策支持基准（包括交互式、报表、分析式查询）
- 由于除Impala外，其它引擎都没有基于成本的优化器，本测试使用的查询都使用SQL-92标准的连接
- 采用统一的Snappy压缩编码方式，各个引擎使用各自最优的文件格式，Impala和Spark SQL使用Parquet，Hive-on-Tez使用ORC，Presto使用RCFile。
- 对每种引擎多次运行和调优

结果：

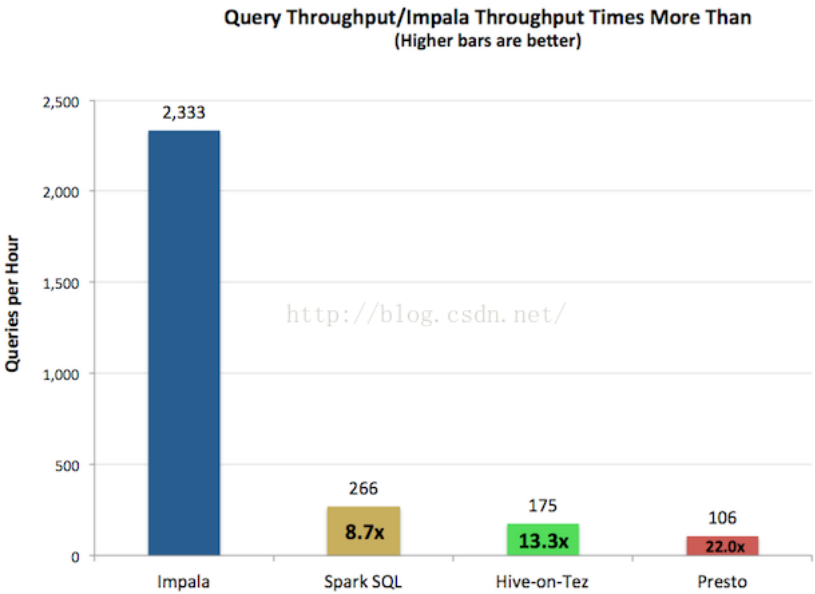
单用户如下图所示。



多用户如下图所示。



查询吞吐率如下图所示。



Impala本身就是cloudera公司的主打产品，因此只听其一面之词未免有失偏颇，下面就再看一个SAS公司的测试。

(2) SAS2013年做的Impala和Hive的对比测试

硬件：

- Dell M1000e server rack
- 10 Dell M610 blades
- Juniper EX4500 10 GbE switch

刀片服务器配置

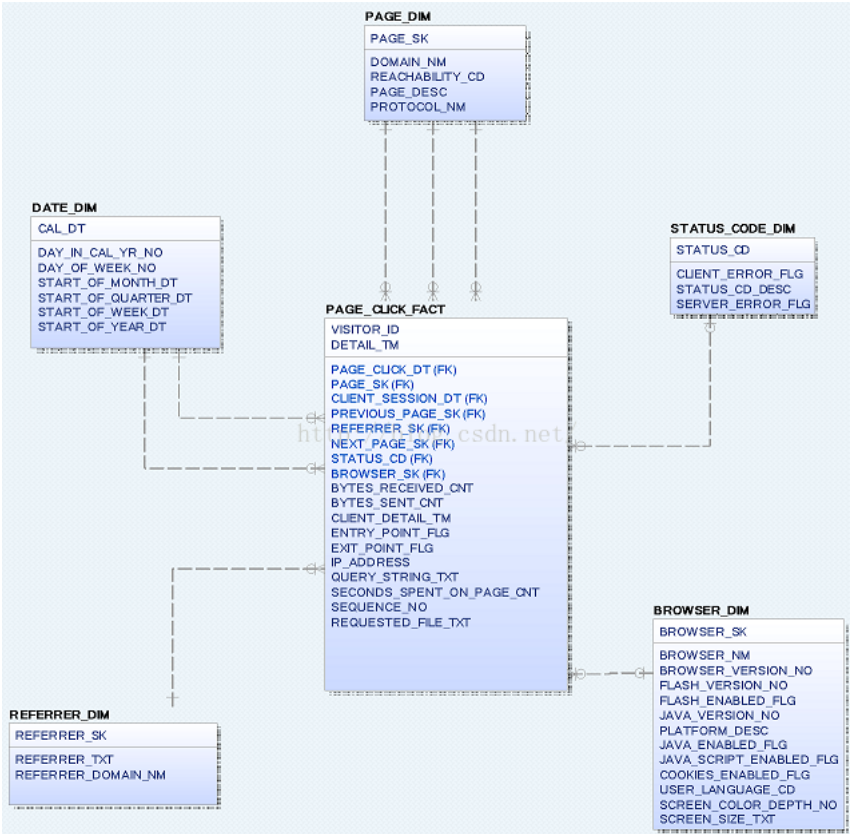
- Intel Xeon X5667 3.07GHz processor
- Dell PERC H700 Integrated RAID controller
- Disk size: 543 GB
- FreeBSD iSCSI Initiator driver
- HP P2000 G3 iSCSI dual controller
- Memory: 94.4 GB

软件：

- Linux 2.6.32
- Apache Hadoop 2.0.0
- Apache Hive 0.10.0

- Impala 1.0
- Apache MapReduce 0.20.2

数据：
数据模型如下图所示。



各表的数据量如下图所示。

Table Name	Rows
PAGE_CLICK_FACT	1.45 billion
PAGE_DIM	2.23 million
REFERRER_DIM	10.52 million
BROWSER_DIM	164.2 thousand
STATUS_CODE	70

PAGE_CLICK_FLAT表使用Compressed Sequence文件格式，大小124.59 GB。

查询：
使用了以下5条查询语句

```
1 -- What are the most visited top-level directories on the customer support website for a given week and year?
2 select top_directory, count(*) as unique_visits
3   from (select distinct visitor_id, split(requested_file, '[\\\/]')[1] as top_directory
4         from page_click_flat
5        where domain_nm = 'support.sas.com'
6              and flash_enabled='1'
7              and weekofyear(detail_tm) = 48
8              and year(detail_tm) = 2012
9       ) directory_summary
10  group by top_directory
11  order by unique_visits;
12
13 -- What are the most visited pages that are referred from a Google search for a given month?
14 select domain_nm, requested_file, count(*) as unique_visitors, month
15   from (select distinct domain_nm, requested_file, visitor_id, month(detail_tm) as month
16         from page_click_flat
17        where domain_nm = 'support.sas.com'
18              and referrer_domain_nm = 'www.google.com'
19       ) visits_pp_ph_summary
20  group by domain_nm, requested_file, month
21  order by domain_nm, requested_file, unique_visitors desc, month asc;
```

22 |

23 | -- What are the most common search terms used on the customer support website for a given year?

24 | select query_string_txt, count(*) as count

25 | from page_click_flat

26 | where query_string_txt <> ''

27 | and domain_nm='support.sas.com'

28 | and year(detail_tm) = '2012'

29 | group by query_string_txt

30 | order by count desc;

31 |

32 | -- What is the total number of visitors per page using the Safari browser?

33 | select domain_nm, requested_file, count(*) as unique_visitors

34 | from (select distinct domain_nm, requested_file, visitor_id

35 | from page_click_flat

36 | where domain_nm='support.sas.com'

37 | and browser_nm like '%Safari%'

38 | and weekofyear(detail_tm) = 48

39 | and year(detail_tm) = 2012

40 |) uv_summary

41 | group by domain_nm, requested_file

42 | order by unique_visitors desc;

43 |

44 | -- How many visitors spend more than 10 seconds viewing each page for a given week and year?

45 | select domain_nm, requested_file, count(*) as unique_visits

46 | from (select distinct domain_nm, requested_file, visitor_id

47 | from page_click_flat

48 | where domain_nm='support.sas.com'

49 | and weekofyear(detail_tm) = 48

50 | and year(detail_tm) = 2012

51 | and seconds_spent_on_page_cnt > 10;

52 |) visits_summary

53 | group by domain_nm, requested_file

54 | order by unique_visits desc;

0

8

结果：
Hive与Impala查询时间对比如下图所示。

Impala Hadoop Flat File	Query	Average Hive (MM:SS)	Impala Time (MM:SS)	Improvement (Hive to Impala)
	1	01:22	00:10	01:12
	2	01:29	00:04	01:25
	3	05:59	15:48	(09:49)
	4	01:34	00:04	01:30
	5	04:43	06:46	(02:03)

可以看到，查询1、2、4Impala比Hive快的多，而查询3、5Impala却比Hive慢很多。这个测试可能更客观一些，而且也从侧面说明了一个问题，不要轻信厂商宣传的数自己的实际测试情况得出结论。

想对作者说点什么？

我来说两句

- 张普：我认为SparkSQL是可以做OLAP的，楼主能给分析一下吗 (08-11 12:53 #6楼) 查看回复(1)
- HongdengLi：不错，有收获，谢谢 (09-04 21:08 #5楼)
- dongxuxucool：SparkSQL基于内存的怎么会慢，已经满足了BI基本需求 (08-30 21:40 #4楼)

查看 8 条热评

基于Hadoop生态圈的数据仓库实践 —— ETL（一）

1.7万

一、使用Sqoop抽取数据 1. Sqoop简介 Sqoop是一个在Hadoop与结构化数据存储（如关系数据库）之间高...

基于hadoop生态圈的数据仓库实践 —— OLAP与数据可视化（三）

5049

三、Impala OLAP实例 本节使用前面销售订单的例子说明如何使用Impala做OLAP类型的查询，以及实际遇...

基于hadoop生态圈的数据仓库实践 —— OLAP与数据可视化（一）

1.3万

一、OLAP与Impala简介1. OLAP简介 OLAP是Online Analytical Processing的缩写，意为联机分析处理。此...

基于Hadoop生态圈的数据仓库实践 —— ETL（二）

二、使用Hive转换、装载数据1. Hive简介（1）Hive是什么

Hive是一个数据仓库软件，使用SQL读、写、管...

0
8

数据仓库和Hadoop大数据平台有什么差别？

广义上来说，Hadoop大数据平台也可以看做是新一代的数据仓库系统， 它也具有很多现代数据仓库的特征，也...

[完]基于Hadoop的数据仓库Hive 基础知识

Hive是基于Hadoop的数据仓库工具，可对存储在HDFS上的文件中的数据集进行数据整理、特殊查询和分析处...

实战hadoop海量数据处理系列03 ：数据仓库的设计

#实战hadoop海量数据处理系列03 ：数据仓库的设计 鉴于我们之前两章提前预热的开发环境，我们现在来讨论...

基于Hadoop生态圈的数据仓库实践 —— 环境搭建（一）

一、Hadoop版本选型

主流的Hadoop生态圈有Apache、Cloudera、HortonWorks、MapR几个不同版本，...

深度解读Hadoop与数据仓库概念

那些想要弄清楚“大数据”概念的组织需要做出一个选择，是要采用传统的数据仓库概念和现有的数据仓库架构，...

基于Hadoop的数据仓库Hive 学习指南

【版权声明：本指南为厦门大学林子雨开设的《大数据技术原理与应用》课程新增配套学习资料，版权所有，转...

相关热词

——基于

基于流和基于包

基于流基于包

mqtt基于基于

基于寄存器基于栈

个人资料



wzy0623

博客专家

关注

原创

粉丝

喜欢

评论

308

1337

151

285

等级： 博客 1

访问： 166万+

积分： 1万+

排名： 1704

勋章：  

最新文章

初学乍练：redis复制与哨兵

初学乍练：redis持久化

初学乍练：redis事务与脚本

初学乍练：redis主从复制及哨兵维护脚本

初学乍练：一键部署redis集群

博主专栏



基于hadoop生态圈的数据仓库实践

阅读量：167504

25 篇



Pentaho Work with Big Data

阅读量：36501

8 篇



更好的Hadoop数据仓库解决方案——HAWQ技术解析

阅读量：95927

19 篇



更好的Hadoop数据仓库解决方案——HAWQ实战演练

阅读量：8347919 篇

展开

个人分类

BI	241篇
DBA	8篇
Linux	59篇
MySQL	71篇
Oracle	62篇

展开

归档

2018年9月	4篇
2018年8月	3篇
2018年7月	8篇
2018年5月	6篇
2018年4月	2篇

展开

热门文章

- 让Hive支持行级insert、update、delete
阅读量：29996
- 基于hadoop生态圈的数据仓库实践 —— OLAP与数据可视化（二）
阅读量：25694
- kylin 安装配置实验
阅读量：21519
- Oracle中的三种 UPDATE FROM 的解决方案
阅读量：18131
- HAWQ技术解析（一） —— HAWQ简介
阅读量：17906

最新评论

- HAWQ + MADlib 玩转数...
Lswx2006：跟 低秩矩阵分解 推荐算法 的效果和适用场景 区别在哪里？
- HAWQ取代传统数仓实践（一） —— ...
LSB19930706：[reply]zhongqi2513[reply] hive擅长写（数据仓库），读（实时查询）...
- 第二本书出版啦！
wzy0623：[reply]kenlee07[reply] QQ:1073498799
- 第二本书出版啦！
kenlee07：买了您的大作：，，里边有些地方看弄明白，我想问下，怎么和您互动呢？phone? email? QQ...
- HAWQ技术解析（二） —— 安装...
qq_26803795：OushuDB？

联系我们



请扫描二维码联系客服

✉ webmaster@csdn.net

☎ 400-660-0108

💬 QQ客服 💬 客服论坛

0
8

[关于](#)

[招聘](#)

[广告服务](#)

[网站地图](#)

©2018 CSDN版权所有 京ICP证09002463号

 百度提供搜索支持



经营性网站备案信息

网络110报警服务

中国互联网举报中心

北京互联网违法和不良信息举报中心

CSDN APP

0
8