文本复制检测报告单(全文对照)

№:ADBD2019R_20190305131041201904121151347051967035 90　　检测时间:2019-04-12 11:51:34

检测文献:　谢冲
作者:　谢冲
检测范围:　中国学术期刊网络出版总库
中国博士学位论文全文数据库/中国优秀硕士学位论文全文数据库
中国重要会议论文全文数据库
中国重要报纸全文数据库
中国专利全文数据库
图书资源
优先出版文献库
学术论文联合比对库
互联网资源(包含贴吧等论坛资源)
英文数据库(涵盖期刊、博硕、会议的英文数据以及德国Springer、英国Taylor&Francis 期刊数据库等)
港澳台学术文献库
互联网文档资源
CNKI大成编客-原创作品库
个人比对库
时间范围:　1900-01-01至2019-04-12

检测结果

去除本人已发表文献复制比：22.8%　　跨语言检测结果：0%
去除引用文献复制比：20.8%　　总文字复制比：22.8%
单篇最大文字复制比：5.1%（姚晓闯_B1311686_矢量大数据管理关键技术研究）

重复字数：	[9230]	总段落数：	[7]
总字数：	[40483]	疑似段落数：	[6]
单篇最大重复字数：	[2059]	前部重合字数：	[2763]
疑似段落最大重合字数：	[3851]	后部重合字数：	[6467]
疑似段落最小重合字数：	[111]		

指　标：☑疑似剽窃观点　☑疑似剽窃文字表述　☐疑似自我剽窃　☐疑似整体剽窃　☐过度引用

表　格：0　　公　式：0　　疑似文字的图片：0　　脚注与尾注：0

3.6%（111）　中英文摘要等（总3076字）
52.8%（3851）　第1章绪论（总7294字）
42.7%（3133）　第2章技术理论基础（总7343字）
13.3%（1165）　第3章时空矢量对象的索引设计及存储（总8789字）
5.1%（452）　第4章时空矢量对象的查询算法设计（总8801字）
0%（0）　第5章实验性能评价及结果分析（总2787字）
21.6%（518）　第6章总结与展望（总2393字）

（注释：无问题部分　文字复制部分　引用部分）

疑似剽窃观点　(1)

第6章总结与展望

1. 目前，我认为还可以在以下几个方面继续开展相关研究，以推动大数据时代矢量数据管理基础理论和关键技术的快速发展。

1. 中英文摘要等　　总字数：3076

相似文献列表

去除本人已发表文献复制比：3.6%(111)　　文字复制比：3.6%(111)　　疑似剽窃观点：(0)

1	姚晓闯_B1311686_矢量大数据管理关键技术研究	3.6%（110）

	姚晓闯 - 《学术论文联合比对库》- 2017-06-01	是否引证：否
2	矢量大数据管理关键技术研究	1.6%（50）
	姚晓闯(导师：郧文聚;朱德海) - 《中国农业大学博士论文》- 2017-05-01	是否引证：否

原文内容	相似内容来源
1 此处有 50 字相似 ckin数据等数据爆发式增长。这些数据具有流式动态、时空多维、规模巨大、蕴含价值、稀疏等特征。在这样的时空大数据背景下， 基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求， 基于关系型数据库的存储管理模式也面临着严峻挑战：1）对于异源异构的非结构化时空数据，其难以提供自适应的存储组织方案；2）	矢量大数据管理关键技术研究_姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.往往涉及到国家经济、国防等基础设施建设特殊因素，导致针对矢量大数据共享困难、应用研巧相对较少；另一方面，基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求。因此，本文研巧是在"空间大数据时代"的背景下，开展针对海量矢量大数据存储、管理、可视化、应用等关键技术的难题攻关，不仅 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.于大规模矢量数据往往涉及到国家经济、国防等基础设施建设特殊因素，导致针对矢量大数据共享困难、应用研究相对较少；另一方面，基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求。因此，本文研究是在“空间大数据时代”的背景下，开展针对海量矢量大数据存储、管理、可视化、应用等关键技术的难题攻关，不仅有利于促
2 此处有 61 字相似 n the background of such spatio-temporal big data, based on the existing GIS commercial spatial data management platform, the ability to manage vector big data is also near the limi	姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.fits, has become a big problem in the practical application.The existing GIS commercial spatial data management platforms have been close to their limits. The sudden emergence of the

指　标

疑似剽窃文字表述

1. 基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求，
2. the existing GIS commercial spatial data management platform,

2. 第1章绪论

总字数：7294

相似文献列表

去除本人已发表文献复制比：52.8%(3851)　文字复制比：52.8%(3851)　疑似剽窃观点：(0)

1	姚晓闯_B1311686_矢量大数据管理关键技术研究	21.7%（1582）
	姚晓闯 - 《学术论文联合比对库》- 2017-06-01	是否引证：否
2	矢量大数据管理关键技术研究	20.6%（1501）
	姚晓闯(导师：郧文聚;朱德海) - 《中国农业大学博士论文》- 2017-05-01	是否引证：否
3	基于Hadoop平台的时空数据索引和查询技术研究	17.5%（1280）
	何立志(导师：李龙海) - 《西安电子科技大学硕士论文》- 2018-06-01	是否引证：否
4	基于空间填充曲线高维空间查询算法研究	7.3%（533）
	徐红波(导师：郝忠孝) - 《哈尔滨理工大学博士论文》- 2010-03-01	是否引证：否
5	Z曲线网格划分的最近邻查询	2.9%（208）
	刘润涛;陈琳琳;田广悦; - 《计算机工程与应用》- 2012-12-03 1	是否引证：否
6	王广钰	1.9%（138）
	王广钰 - 《学术论文联合比对库》- 2017-04-10	是否引证：否
7	基于Hadoop的时空大数据的分布式检索方法	1.9%（138）
	王广钰(导师：李英玉) - 《中国科学院大学(中国科学院国家空间科学中心)硕士论文》- 2017-05-01	是否引证：否

8	数据网格QoS保障与资源优化关键技术研究	1.2%（89）
	曲明成(导师：杨孝宗) -《哈尔滨工业大学博士论文》- 2011-04-01	是否引证：否
9	大数据时代的交管综合应用云平台	1.2%（84）
	徐文斌; -《第八届中国智能交通年会论文集》- 2013-09-26	是否引证：否
10	基于iOS的汽车租赁客户端的设计与实现	0.9%（64）
	李文慧(导师：朴勇) -《大连理工大学硕士论文》- 2018-03-01	是否引证：否
11	基于知识图谱的Web of Scholars系统设计与实现	0.8%（61）
	康文杰(导师：夏锋) -《大连理工大学硕士论文》- 2018-04-01	是否引证：否
12	HBase中半结构化时空数据存储与查询处理	0.8%（59）
	封孝生;张翀;陈晓莹;唐九阳;葛斌; -《国防科技大学学报》- 2016-06-28	是否引证：否
13	改进K-means算法在文本聚类中的应用	0.7%（53）
	周本金(导师：陶以政) -《中国工程物理研究院硕士论文》- 2018-05-01	是否引证：否
14	基于Hadoop的IPTV故障预判算法的研究及系统实现	0.7%（50）
	王堂辉(导师：周亮) -《南京邮电大学硕士论文》- 2018-11-14	是否引证：否
15	基于HBase的空间数据分布式存储和并行查询算法研究	0.6%（47）
	丁琛(导师：鲍培明) -《南京师范大学硕士论文》- 2014-04-30	是否引证：否
16	基于云计算的BIM关键技术应用研究	0.6%（46）
	毕振波(导师：王慧琴) -《西安建筑科技大学博士论文》- 2015-06-01	是否引证：否
17	探索计算机技术发展新方向中的云计算	0.6%（41）
	李大伟; -《科技资讯》- 2018-11-03	是否引证：否

原文内容	相似内容来源
1	此处有 53 字相似 据等数据爆发式增长[1-2]。这些数据具有流式动态、时空多维、规模巨大、蕴含价值、稀疏等特征。在这样的时空大数据背景下， 基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求[3]， 基于关系型数据库的存储管理模式也面临着严峻挑战：1）对于异源异构的非结构化时空数据，其难以提供自适应的存储组织方案；2）	矢量大数据管理关键技术研究_姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.往往涉及到国家经济、国防等基础设施建设特殊因素，导致针对矢量大数据共享困难、应用研巧相对较少；另一方面，基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求。因此，本文研巧是在"空间大数据时代"的背景下，开展针对海量矢量大数据存储、管理、可视化、应用等关键技术的难题攻关，不仅 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.于大规模矢量数据往往涉及到国家经济、国防等基础设施建设特殊因素，导致针对矢量大数据共享困难、应用研究相对较少；另一方面，基于现有的GIS商业化空间数据管理平台对矢量大数据的管理能力也几近极限，无法满足全国行业应用的需求。因此，本文研究是在“空间大数据时代”的背景下，开展针对海量矢量大数据存储、管理、可视化、应用等关键技术的难题攻关，不仅有利于促
2	此处有 53 字相似 对流式生成的时空数据，其难以提供强大的高并发读写能力支持；3）对于持续快速增长的数据量，其难以提供高可扩展性支持。随着云 计算、NoSQL[4-5]数据库等新一代IT技术的发展与成熟，相关理论与方法已经开始逐步渗透到GIS领域， 其分布式存储模式、高并发读写能力、以及高可扩展性/高可用性特征促使数据库管理模式逐渐向分布式数据库转变[6]。目前很多研	矢量大数据管理关键技术研究_姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.ormationSystem, GIS)的"血液"。空间大数据的兴起，也带来了传统GIS行业的变革，尤其是基于云计算、NoSQL数据库等新一代高性能计算技术的发展与成熟，相关理论和方法已经开始逐步渗透到G投领域。 2014年国务院在发布的《关于促进地理信息产业发展的意见》中更是明确指出结合下一代互联网、物联网、云计算 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1. Information System，GIS）的“血液”。空间大数据的兴起，也带来了传统GIS行业的变革，尤其是基于云计算、NoSQL数据库等新一代高性能计算技术的发展与成熟，相关理论和方法已经开始逐步渗透到GIS领域。

		2014年国务院在发布的《关于促进地理信息产业发展的意见》中更是明确指出结合下一代互联网、物联网、云计算等新技术的发展趋
3	此处有 60 字相似]。目前很多研究学者基于NoSQL数据库开展了广泛的时空数据存储管理研究。而以HBase[7]为代表的NoSQL数据库仅 支持Key-Value模式的简单查询，对多维复杂查询能力支持不足。目前许多工作都围绕如何提高多维或空间查询性能展开研究。 其中部分研究仅针对矢量数据，但不支持其时间维度的查询，无法进行时空查询；部分研究仅支持点数据对象的存储及查询，不支持更为	HBase中半结构化时空数据存储与查询处理 封孝生;张翀;陈晓莹;唐九阳;葛斌; -《国防科技大学学报》- 2016-06-28（是否引证：否） 1.、可伸缩、实时读写的分布式数据库,可支持集群存储海量数据,极大弥补了传统数据库和Mongo DB的不足。然而HBase仅支持键值对(key-value)模式的查询,对多维复杂查询能力支持不足,因此现有的工作基本围绕如何提高HBase多维或空间查询性能展开,主要集中研究建立空间数据的索引结构。一方面,这些工作没有考虑空间对象的时间维属性。对于时空对象,时间和空间属性是密不可分的,简单将时
4	此处有 64 字相似 引、查询检索的方案，使该系统能够在大数据场景下存储海量的时空矢量数据对象，且同时支持低延迟的时空查询、k-NN查询。 1.2 国内外研究现状 1.1.1 时空数据库的研究现状 空间数据库是管理时空数据的有效手段，也是数据查询、分析应用的基础。 目前，海量时空大数据的存储和管理主要采用三种方式：一是基于分布式扩展的关系型数据库进行存储；二是基于分布式文件系统进行存	基于空间填充曲线高维空间查询算法研究 徐红波 -《哈尔滨理工大学博士论文》- 2010-03-01（是否引证：否） 1.大量数据点的集合分成若干类，使得每个类中的数据之间最大程度地相似，而不同类中的数据最大程度地不同。 1.2 研究现状 1.2.1 索引结构 空间数据库是存储和管理空间数据的数据库系统。为了高效、快速地访问海量空间数据，提出了许多空间索引方法[7, 8]。从空间数据类型的观点 矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.加W解决，未来将会在很大程度上阻碍GIS大数据的共享和应用的发展。1.2.2 矢《大数据存储空间数据库是管理矢量数据的有效手段，也甚矢量数据查询、分析和应用的基拙。近半个世纪W来，空间数据库管理技术主要经历了四个阶段的演变["]，即文件系统（20世纪70年代)、文件关 姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.和规范化[31]等方面如果不加以解决，未来将会在很大程度上阻碍GIS大数据的共享和应用的发展。矢量大数据存储空间数据库是管理矢量数据的有效手段，也是矢量数据查询、分析和应用的基础。近半个世纪以来，空间数据库管理技术主要经历了四个阶段的演变[15]，即文件系统（20世纪70年代）、文件关系混合系统（2
5	此处有 83 字相似 国内外研究现状 1.1.1 时空数据库的研究现状 空间数据库是管理时空数据的有效手段，也是数据查询、分析应用的基础。 目前，海量时空大数据的存储和管理主要采用三种方式：一是基于分布式扩展的关系型数据库进行存储；二是基于分布式文件系统进行存储；三是基于非关系型NoSQL数据库进行存储。 其中第一种方式，在海量数据的情况下，可以通过提高水平扩展性存储更多的数据，但由于关系型数据库其本身的ACID局限性，不适	矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.结构也发生着革命巧的变化。如今，在大数据时代，随着计算机技术的发展，在空间数据库方面出现了新的研巧成果。目前，矢量大数据的存储和管理方式主要采用W下H种，一是基于支持分布式扩展的关系型数据库进行存储；二是基于非关系型NoSQL数据库进行存储；H是基于分布式文件系5|||阳农化人巧梢七巧化讫义 第--巧绪论统进行存储。(1)支持分 姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.GIS软件的体系结构也发生着革命性的变化。如今，在大数据时代，随着计算机技术的发展，在空间数据库方面出现了新的研究成果。目前，矢量大数据的存储和管理方式主要采用以下三种，一是基于支持分布式扩展的关系型数据库进行存储；二是基于非关系型NoSQL数据库进行存储；三是基于分布式文件系统进行

		存储。g) 支持分布式扩展的关系型数据库基于关系型空
6	此处有 365 字相似 数据库其本身的ACID局限性，不适用于高并发读写的场景，且无法适应非结构化的存储。在这里主要介绍第二、三类方式。 1. 2.1.1分布式文件系统 分布式文件系统（Distributed File System）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连；该系统对普通计算机具有较好的支持，在集群方面也具有良好的可扩展性和容错性；同时对大规模用户并发情况下的数据密集型服务也能提供良好的支持。其中，Apache软件基金会的Hadoop Distributed File System（HDFS）[8]文件系统是具有这种特点的分布式文件系统的典型代表。在基于HDFS分布式文件系统的矢量数据存储方面，国内外做了大量的研究。Tan.H基于Hadoop[9]对空间数据分析进行了研究[10]，提出了一个适合HDFS存储的GPS数据模型，实现了R-tree索引的并行构建，并基于此研发了多种时空数据查询功能。 Ablimit Aji研制了Hadoop-GIS[11]，基于HDFS和MapReduce建立了一套高性能空间数据仓库系	姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.QL数据库存储空间大数据不仅方便部署，同时也能够很好地对多源空间数据进行高效集成，有利于数据密集型的并行计算。i) 分布式文件系统分布式文件系统（Distributed File System）是通过计算机网络与节点相连形成系统管理的存储资源，系统对普通计算机具有较好的支持，在集群方面也具有良好的可扩展性和容错性； 2.利于数据密集型的并行计算。i) 分布式文件系统分布式文件系统（Distributed File System）是通过计算机网络与节点相连形成系统管理的存储资源，系统对普通计算机具有较好的支持，在集群方面也具有良好的可扩展性和容错性；同时对大规模用户并发情况下的数据密集型服务也能提供良好的支持。其中，Apache 软件基金会的Hadoop Distributed File System（HDFS）文件系统是具有这种特点的分布式文件系统的典型代表。在基于HDFS分布式文件系统的矢量数据存储方面，国内外也做了大量的研究。香港科技大学[67]基于Hadoop对是空间数据分析进行了研究，提出了一个新适合HDFS存储的GPS数据模型（ID/Location/time），实现了R-tree索引的并行构建，并基于此研发了多种时空数据查询功能。尹芳等[68]提出了基于Key/Value的矢量数据文本存储格式，通过HDFS来进行空间数据的存储和管理，并初步探讨了矢 矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.数据密集型的并行计算。巧分布式文件系巧分布式文件系统（Distributed File System)是通过计算机网络与节点相连形成系统管理的存储资源，系统对普通计算机具有较好的支持，在集群方面也具有良好的可扩展性和容错性；同时对大规模用户并发情况下的数据密集型服务也能提供良好的支持。其中，Apache软件基金会的HadoopDistributed则eSystem (HDFS)文件系统是具有这种特点的分布式文件系统的典型代表。在基于HDFS分布式文件系统的矢量数据存储方面，国内外也做了大量的研究。香港科技大学[671基于Hadoop对空间数据分析进行了研巧，提出了一个新适合HDFS存储的GPS数据模型(ID/Location/time)，实现了 R-tree索引的并行构建，并基于此研发了多种时空数据查询功能。尹芳等呢出了基于Key/Vahie的矢量数据文本存储格式，通过HDFS来进行空间数据的存储和管理，并初步 数据网格QoS保障与资源优化关键技术研究 曲明成 -《哈尔滨工业大学博士论文》- 2011-04-01（是否引证：否） 1.的性能，而在数据块不大（几兆~十几兆）时就没有评测，而这时其性能与保守调度法的差别甚微[82-84]。2.1.3 分布式文件系统 分布式文件系统（Distributed File System）是指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连。分布式文件系统的设计基于客户机/服务器模式。为迅速增长的数据处理需求，Google 设计并实现了 Goo 大数据时代的交管综合应用云平台 徐文斌; -《第八届中国智能交通年会论文集》- 2013-09-26（是否引证：否） 1.System,以及在国外已经非常成熟的Intel和IBM,各种“云计算”的应用服务范围正日渐扩大,影响力也无可估量。8.2分布式文件系统分布式文件系统(Distributed File

		System)是指文件系统管理的物理存储资源不一定直接连接在本地节点上,而是通过计算机网络与节点相连。8.3虚拟化虚拟化是一个广义的术语,是指计算元件在虚拟的基础上而不是真实的基础上运行,是一个为了简化管理,优化资源的解决 基于云计算的BIM关键技术应用研究 毕振波 -《西安建筑科技大学博士论文》- 2015-06-01（是否引证：否） 1.所管理的存储空间的抽象。分布式文件系统（Distributed File System）指文件系统管理的物理存储资源不一定直接连接在本地节点上，而是通过计算机网络与节点相连。其设计基于客户机/服务器模式，一个典型的网络可包括多个供多用户访问的服务器。通过分布式文件系统可以有效解决数 基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1. 为整个集群提供一致性服务；Ambari 是基于 Web 的客户端工具，方便用户管理与监控集群中的各个模块。2.1.1 分布式文件系统 HDFSHDFS[42]（Hadoop Distributed File System）可被视为 Google 公司 GFS 的开源版本，作为 Hadoop 的核心技术之一，为整个分布式平台提供最基本的 探索计算机技术发展新方向中的云计算 李大伟; -《科技资讯》- 2018-11-03（是否引证：否） 1.业首先需要对用户的大量数据文件进行储存,所以需要大量的储存空间,所以云计算采用分布式文件系统进行大量数据的存储。这种方式文件的存储不是在本地,而是通过计算机网络与节点相连。除此之外,因为云计算系统中有很多用户的信息和存储的文件,因此需要有效的数据管理技术,还应该有保证数据和信息安全的技术确保
7	此处有 91 字相似 DFS存储的GPS数据模型，实现了R-tree索引的并行构建，并基于此研发了多种时空数据查询功能。Ablimit Aji 研制了Hadoop-GIS[11]，基于HDFS和MapReduce建立了一套高性能空间数据仓库系统，通过空间分割实现多种空间数据查询，其实验表明该系统明显优于并行空间数据关系系统。 Ahmed Eldawy提出了一套基于开源Hadoop框架——SpatialHadoop[12-15]，通过HDFS存储	矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.本存储格式，通过HDFS来进行空间数据的存储和管理，并初步探讨了矢量数据的MapReduce计算过程。埃默里大学研制了 Hadoop-GIS，基于HDFS和MapReduce建立了一套窩性能空间数据合库系统，通过空间分割实现多种空间数据查询，其实验表明该系统明显优于并行空间数据关系系统（ParallelSDBMS)。明尼巧趣大学提出了一套基于开源Hadoop框架SpatialHadoop[5i 姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.本存储格式，通过HDFS来进行空间数据的存储和管理，并初步探讨了矢量数据的MapReduce计算过程。埃默里大学[69]研制了Hadoop-GIS，基于HDFS和MapReduce建立了一套高性能空间数据仓库系统，通过空间分割实现多种空间数据查询，其实验表明该系统明显优于并行空间数据关系系统（Parallel SDBMS）。明尼苏达大学提出了一套基于开源Hadoop框架——SpatialHadoop[51,
8	此处有 344 字相似 间数据仓库系统，通过空间分割实现多种空间数据查询，其实验表明该系统明显优于并行空间数据关系系统。 Ahmed Eldawy 提出了一套基于开源Hadoop框架——SpatialHadoop[12-15]，通过HDFS存储空间数据	基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1.空数据存储与查询技术[11]等。为叙述方便，本文简称 VL Hong提出的技术为 STCode-HBase。ST-Hadoop 是基于 Spatial Hadoop 的支持时空数据的

	集，除此之外该系统涵盖了基于Pig的空间高级编程语言、空间数据索引、空间查询以及可视化等一整套空间大数据解决方案，解决了矢量大数据空间分析的基本难题，在业界非常具有代表性和借鉴价值。ST-Hadoop[16]是基于Spatial Hadoop的支持时空数据的Map Reduce框架，其内部建有二级时空索引，第一级为时间索引而第二级为空间索引，空间索引直接继承自Spatial Hadoop的Quad-Tree[17]和 kd-Tree[18]等。其中，ST-Hadoop 将数据按照不同尺寸的时间分片（Time-Slice）进行多次划分，并冗余存储到HDFS，运用了一种以存储资源 换查询效率的思想策略。Hadoop云计算平台提供的分布式文件系统HDFS不仅支持基于磁盘存取的MapReduce并行处理	Map Reduce 框架，其内部建有二级时空索引，第一级为时间索引而第二级为空间索引，空间索引直接继承自Spatial Hadoop 的 Quad-Tree[12]和 Kd-Tree[13]等。其中，ST-Hadoop 将数据按照不同尺寸的时间分片（Time-Slice）进行多次划分，并冗余存储到HDFS，运用了一种以存储资源（Storage）换查询效率（Efficient Performance）的思想策略。MD-HBase 是基于 H 矢量大数据管理关键技术研究_姚晓闯-《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.割实现多种空间数据查询，其实验表明该系统明显优于并行空间数据关系系统（ParallelSDBMS)。明尼巧趣大学提出了一套基于开源Hadoop框架SpatialHadoop[5ii?-73]，通过HDFS存储空间数据集：除此之外，该系统涵盖了基于Pig的空间高级编程语言、空间数据索引、空间查询W及可视化等一整套空间大数据解决方案，解决了矢量大数据空间分析的基本难题，在业界很具有代表性和借鉴价值。浙江大学合作[74)研发的ScalaGiST系统基于Hadoop完成了矢量数据的存储与管理，包括B+-tree索 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯-《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.空间分割实现多种空间数据查询，其实验表明该系统明显优于并行空间数据关系系统（Parallel SDBMS）。明尼苏达大学提出了一套基于开源Hadoop框架——SpatialHadoop[51, 70-73]，通过HDFS存储空间数据集，除此之外，该系统涵盖了基于Pig的空间高级编程语言、空间数据索引、空间查询以及可视化等一整套空间大数据解决方案，解决了矢量大数据空间分析的基本难题，在业界很具有代表性和借鉴价值。浙江大学合作[74]研发的ScalaGiST系统基于Hadoop完成了矢量数据的存储与管理，包括B+-tree索引和R-
9	此处有 213 字相似 doop 将数据按照不同尺寸的时间分片（Time-Slice）进行多次划分，并冗余存储到HDFS，运用了一种以存储资源换 查询效率的思想策略。Hadoop云计算平台提供的分布式文件系统HDFS不仅支持基于磁盘存取的MapReduce并行处理计算，同时对基于内存的Spark并行计算[19]也有无缝的支持。因此，基于HDFS和Spark组合的矢量数据存储于管理平台也成为了可能。例如，Jia Yu研发的GeoSpark[20]，采用三层架构，即Apache Spark层、空间弹性分布数据层和空间查询操作层，其中数据存储采用HDFS作为存储系统。 1.2.1.2 非关系型NoSQL数据库 NoSQL，即Not Only SQL，泛指非关系型数据库。目前，主流的No	矢量大数据管理关键技术研究_姚晓闯-《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.基于Hadoop完成了矢量数据的存储与管理，包括B+-tree索引和R-tree索引，实现了空间范围查询和最邻近查询算法。Hadoop云计算平台提供的分布式文件系统HDFS不仅支持基于磁盘存取的MapReduce并行处理计算，同时对基于内存的Spark并行计算也有无缝的支持。因此，基于HD巧和Spark组合的矢量数据存储于管理平台也y为了可能。例如，亚利桑那州立大学研发的GeoSpark[H'W，采用H层架构，即Apa油eSparki、空间弹性分布数据层和空间查询操作层，其中数据存储采用HDFS作为存储系统。上海交通大学合作研发的Simba[77]，采用基于HDFS、RDBMS等混合架构进行矢里数据存储，同时扩展了 S 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯-《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.ST系统基于Hadoop完成了矢量数据的存储与管理，包括B+-tree索引和R-tree索引，实现了空间范围查询和最邻近查询算法。Hadoop云计算平台提供分布式文件系统HDFS不仅支持基于磁盘存取的MapReduce并行处理计算，同时对基于内存的Spark并行计算也有无缝的支持。因此，基于HDFS和Spark组合的矢量数据存储于管理平台也成为了可能。例如，亚利桑那州立大学研发的GeoSpark[75, 76]，采用三层架构，即Apache Spark层、空间弹性分布数据层和空间查询操作层，其中数据存储采用HDFS作为存储系统。上海交通大学合

		作研发的Simba[77]，采用基于HDFS、RDBMS等混合架构进行矢量数据存储，同时扩展了Spark
10	此处有 395 字相似 pache Spark层、空间弹性分布数据层和空间查询操作层，其中数据存储采用HDFS作为存储系统。 1.2.1.2 非关系型NoSQL数据库 NoSQL，即Not Only SQL，泛指非关系型数据库。目前，主流的NoSQL数据库包括基于文档的存储方式的MongoDB[21]；基于列存储的BigTable[22]、HBase；基于键值对进行存储的Redis[23]等。由于非关系型NoSQL数据库具有分布式、可扩展性、不需要预先定义表结构等诸多优点，近年来深受研究学者和商业公司的青睐，在矢量大数据存储和管理领域也有很多研究工作。吴飞[24]基于MongoDB的LBS数据管理系统，充分考虑系统集群和数据分片与系统负载均衡之间的关系，有效解决了矢量数据存储查询的技术瓶颈；雷德龙[25]等提出了基于MongDB存储和处理矢量空间数据的方法，采用Geojson格式以文件形式来存储矢量数据，设计并实现了基于数据存储层、核心业务层和表现层组成的矢量数据云存储与处理系统，有效提高了数据的并行访问和处理性能。 张景云[26]提出了一种以内存数据库Redis的轻量级矢量地理组织方法，能在高并发情况下有效提高矢量地理数据服务性能。范	矢量大数据管理关键技术研究_姚晓闯-《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.地与原有系统进行高效的集成。而缺点则也非常明显，由于关系型数据库目前不支持复杂的空间数据分析功能。口）非关系型NoSQL數巧库NoSQL，即Not Only SQL，泛指非关系型数据库。目前，主流的NoSQL数据库包括基于文档存储方式的MongoDB;基于列存储的BigTable、Hbase;基于键/值（Key/Value)对进行存储的Redis等。由于非关系型NoSQL数据库具有分布式、可扩展性、不需要预先定义表结构等诸多优点，近年来，深受研巧学者和商业公司的青睐，而且，在矢量大数据存储于管理领域也有颇多研巧。如吴飞P9谭基于MongoDB的LBS数据管理系统，充分考虑系统集群和数据分片与系统负载均衡之间的关系，有效解决丫矢量数据化储巧询的技术瓶颈；甫德龙巧提出了基于MongoDB存储和处理矢量空间数据的方法，采用GeoJSON格式W文件形式来存储矢量数据，设计并实现了基于数据存储层、核屯、业务层和表现层组成的矢量数据云存储与处理系统，有效提高了数据的并行访问和处理性能。范永健等基于Hbase设计了矢量数据的表结构，其中Key为空间对象ID，S个列族分别用于存储空间对象的属性信息、 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯-《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.面能够很好地与原有系统进行高效的集成。而缺点则也非常明显，由于关系型数据库目前不支持复杂的空间数据分析功能。h) 非关系型NoSQL数据库NoSQL，即Not Only SQL，泛指非关系型数据库。目前，主流的NoSQL数据库包括基于文档存储方式的MongoDB；基于列存储的BigTable、Hbase；基于键/值（Key/Value）对进行存储的Redis等。由于非关系型NoSQL数据库具有分布式、可扩展性、不需要预先定义表结构等诸多优点，近年来，深受研究学者和商业公司的青睐，而且，在矢量大数据存储于管理领域也有颇多研究。如吴飞[59]等基于MongoDB的LBS数据管理系统，充分考虑系统集群和数据分片与系统负载均衡之间的关系，有效解决了矢量数据存储查询的技术瓶颈；雷德龙[60]等提出了基于MongoDB存储和处理矢量空间数据的方法，采用GeoJSON格式以文件形式来存储矢量数据，设计并实现了基于数据存储层、核心业务层和表现层组成的矢量数据云存储与处理系统，有效提高了数据的并行访问和处理性能。范永健[61]等基于Hbase设计了矢量数据的表结构，其中Key为空间对象ID，三个列族分别用于存储空间对象的属性信息、
11	此处有 70 字相似 。张景云[26]提出了一种以内存数据库Redis的轻量级矢量地理组织方法，能在高并发情况下有效提高矢量地理数据服务性能。 范永健[27]等基于HBase设计了矢量数据的表结构，其中Key为空间对象ID，S个列族分别用于存储空间对象的属性信息、坐标信息和拓扑信息。 Nishimura面向LBS服务，设计了md-hbase[28]来存储管理空间数据，其通过空间划分构建了基于四叉树和kd	矢量大数据管理关键技术研究_姚晓闯-《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.了基于数据存储层、核屯、业务层和表现层组成的矢量数据云存储与处理系统，有效提高了数据的并行访问和处理性能。范永健等基于Hbase设计了矢量数据的表结构，其中Key为空间对象ID，S个列族分别用于存储空间对象的属性信息、坐标信息和拓扑信息。加州大学合作研制的基于冊ase构建了 MD-册a化系统[62]，用于管理LBS位置数据，并通过空间范围划分构建了 姚晓闯_B1311686_矢量大数据管理关键技术研究_姚晓闯-《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.数据，设计并实现了基于数据存储层、核心业务层和

		表现层组成的矢量数据云存储与处理系统，有效提高了数据的并行访问和处理性能。范永健[61]等基于Hbase设计了矢量数据的表结构，其中Key为空间对象ID，三个列族分别用于存储空间对象的属性信息、坐标信息和拓扑信息。加州大学合作研制的基于Hbase构建了MD-Hbase系统[62]，用于管理LBS位置数据，并通过空间范围划分构建了基于
12	此处有 64 字相似 滤从而提高查询效率。Ma Y[35]将时间进行粗粒度划分并建立B+[36]树，且在各个时间段下单独构建R树进行空间划分。 基于非关系型NoSQL数据库存储空间大数据不仅方便部署，同时也能够很好地对多源空间数据进行高效集成，有利于数据密集型的并行计算。 李雪梅[37]设计了一种基于分布式数据库HBase的GIS数据管理系统。针对矢量空间数据的存储、索引与检索，提出了一种新	姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.ongoDB数据库，基于NoSQL数据库和内存数据库，提出了空间大数据分布式存储策略与综合处理方法，并取得了良好的效果。基于非关系型NoSQL数据库存储空间大数据不仅方便部署，同时也能够很好地对多源空间数据进行高效集成，有利于数据密集型的并行计算。i) 分布式文件系统分布式文件系统（Distributed File System）是通过计算机网络与节点相连形成系 矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.。基于非关系型NoSQL数据库存储空间大数据不仅方便部署，6中国农业大学博±学位论文 第一章绪论同时也能够很好地对多源空间数据进行高效集成，有利于数据密集型的并行计算。巧分布式文件系巧分布式文件系统（Distributed File System)是通过计算机网络与节点相连形成系
13	此处有 50 字相似 量地理信息时，能通过HBase的Rowkey迅速定位需要返回的数据。丁琛[38]提出了面向HBase的k-NN查询算法， 基于查询热点并行化构建了索引表；使用索引表快速查找到k近邻对象,降低了查询时间,提高了查询效率。 1.1.2 时空索引技术 时空数据是一种描述空间状态随时间变化的数据类型，有助于帮助人们了解过去、把握现在、预测未来，	基于HBase的空间数据分布式存储和并行查询算法研究 丁琛 -《南京师范大学硕士论文》- 2014-04-30（是否引证：否） 1.查询算法：面向点的K近邻查询算法PointKNN和基于索引表的K近邻查询算法IndexKNN。在IndexKNN方法中,基于查询热点并行化构建了索引表；使用索引表快速查找到K近邻对象,降低了查询时间,提高了查询效率。 ★实验结果表明,提出的算法对空间数据查询是有效的。HBase表模式;;MapReduce;;区域查询;;K近邻查询;;索
14	此处有 141 字相似 询算法，基于查询热点并行化构建了索引表；使用索引表快速查找到k近邻对象,降低了查询时间,提高了查询效率。 1.1.2 时空索引技术 时空数据是一种描述空间状态随时间变化的数据类型，有助于帮助人们了解过去、把握现在、预测未来，当前采集到的科学数据基本是带有时间属性的，时间维度的增加极大地丰富了空间对象的数据信息。时空数据查询是时空数据分析处理的重要手段，而时空数据索引技术是时空数据查询的基础。 1.2.2.1 基于R树索引 目前，大多数主流时空索引是通过基于空间R树索引的变形扩展获得的。 R树是Guttman在	王广钰 王广钰 -《学术论文联合比对库》- 2017-04-10（是否引证：否） 1.架构两部分来提高数据的检索效率，为有效地分析这些空间科学数据提供支持[70]。1.2 国内外研究现状 1.2.1 时空数据索引方法时空数据是一种描述空间状态随时间变化的数据类型，有助于帮助人们了解过去、把握现在、预测未来，当前采集到的科学数据基本是带有时间属性的，时间维度的增加极大地丰富了空间对象的数据信息[2]。时空数据查询是实现对时空数据分析处理的前提，而时空数据索引技术是时空数据查询的基础[71]。时空数据索引方法分为：过去时空数据索引、当前时空数据索引、将来时空数据索引和全时时空数据索引{张林, 2010 #1}[基于Hadoop的时空大数据的分布式检索方法 王广钰 -《中国科学院大学(中国科学院国家空间科学中心)硕士论文》- 2017-05-01（是否引证：否） 1.析这些空间科学数据提供支持[70]。 第 1 章 绪论21.2 国内外研究现状1.2.1 时空数据索引方法时空数据是一种描述空间状态随时间变化的数据类型，有助于帮助人们了解过去、把握现在、预测未来，当前采集到的科学

		数据基本是带有时间属性的，时间维度的增加极大地丰富了空间对象的数据信息[2]。时空数据查询是实现对时空数据分析处理的前提，而时空数据索引技术是时空数据查询的基础[71]。时空数据索引方法分为：过去时空数据索引、当前时空数据索引、将来时空数据索引和全时时空数据索引[3]，本文
15	此处有 75 字相似 数据查询的基础。 1.2.2.1 基于R树索引 目前，大多数主流时空索引是通过基于空间R树索引的变形扩展获得的。 R 树是Guttman在1984年提出的空间数据的索引结构[39]，现在广泛用于各种商业应用和原型系统。R树是一种多级平衡树，它是B树在多维空间上的扩展， 其中的每个节点代表一个区域，是包含所有子节点区域的的最小包围矩形（MBR），当中的各个叶子节点则指向各个空间对象。其查询	基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1. R 树索引为基础经过变形扩展所得，因此在介绍 4 种离散型时空索引之前，本小节将对 R 树做简单介绍。R 树是 Guttman 在 1984 年提出的一种针对空间数据的索引结构，现被广泛应用于各种商业应用和原型系统中。R 树是 B 树在多维空间的扩展，一棵 M 阶的 R 树包含两种节点：叶子节点和非叶子节点（也可称为目录节点），具体节点结构描述如下：叶子节点
16	此处有 89 字相似 而R*树是R树的变体[40]，其通过减少节点MBR之间的重叠关系从而降低多余的节点访问次数。 图1-1 R树示意图 3DR-Tree是由Y. Theodoridis和M.Vazirgiannis等人在1996年提出的时空索引结构[41]。核心思想是将时间视为对象的另一个空间维度，并使用R-Tr ee执行三维空间索引。也就是说，时间段[T1，T2]中空间位置（X，Y）处的运动物体可以由从 （X，Y，T1）到（X，Y，	基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1.e 的优点是结构几乎和传统的 R-Tree 一致，结构相对简单，便于使用。2.2.3 3DR-Tree3DR-Tree 是由 Y.Theodoridis 和 M.Vazirgiannis 等人在 1996 年提出的时空索引结构，其核心思想是：将时间完全视作移动对象的另一个空间维度，然后用 R-Tree 进行空间索引。即在时间段[t1,t2]内在空间位置（x，y）处的移动对象可用从（x，y，t1）到（x，y，t
17	此处有 117 字相似 是说，时间段[T1，T2]中空间位置（X，Y）处的运动物体可以由从（X，Y，T1）到（X，Y，T2）的三维空间线段表示。 传统空间R树的二维最小外包矩形增加了维度T（时间）并成为三维最小外包立方体。 3DR-Tree可以被认为是R-Tree的多维扩展，两者都是树结构一致的。3DR-Tree的基本操作，包括节点插入和删除，节点分裂和节点查询，类似于R-Tr ee的基本操作。 3DR-Tree使用n + 1维空间。表示n维时间3DR-Tree的数据也有其局限性，适用于对象空间位	基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1.，并且该线段能用 3DR-Tree 索引。3DR-Tree 的时空对象表示和与之对应的树状结构如图 2.8 所示，传统空间 R-Tree的二维最小外包矩形（MBR）多加了一个维度 T（时间）后，变成了三维的最小外包立方体（MBB）。3DR-Tree 可以认为是 R-Tree 的多维扩展，两者的树状结构是一致的。3DR-Tree 的基本操作，包括节点插入与删除、节点分裂和节点查询等，也与 R-Tree第二章 相关技术概述17的基本操作类似。 TXYR1R2AB
18	此处有 545 字相似 包立方体的MBR过大，导致MBR在整个索引中大量时间和空间重叠的概率很大，这极大地影响了3DR-Tree的检索效率。 HR-Tree 的全称是History R-Tree，也属于R-Tree家族中的一员，是R-Tree的一种时空变体，该索引结构由M Nascimento和J Silva在1998年提出[42]。HR-Tree的核心思想是：采用二级索引机制，首先将时间维度数据单独分割出来并为之构建第一级索引，然后再针对各个时间片下的空间数据用R-Tree作为第二级索引。HR-Tree的实例结构图如图1-2所示，其首先将所有移动对象的时间戳信息提取出来，并按照时间顺序对其进行排	基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1.相对较低，然而轨迹完整性的维护使得其能用最小代价完成特定移动对象的轨迹查询。2.2.5 HR-TreeHR-Tree 的全称是 History R-Tree，也属于 R-Tree 家族中的一员，是 R-Tree 的一种时空变体，该索引结构由 M Nascimento 和 J Silva 在 1998 年提出。HR-Tree 的核心思想是：采用二级索引机制，首先将时间维度数据单独分割出来并为之构建第一级索引，然后再针对各个时间片下的空间数据用 R-Tree 作为第二级索引。HR-Tree 的实例结构图如图 2.9 所示，首先将所有移动对象

	序，生成有序的时间戳列表，如图中的List(T0，T1，T2)然后对各个时间戳下的移动对象根据其空间位置信息构建对应的R-Tree，如图中的R1、R2和R3。为了节省存储空间，HR-Tree采用了“子树重叠”策略，策略的中心思想是：对于时间戳相邻的两棵R-Tree，若在该时间间隔内存在地理空间位置没有发生变化的对象，即两个树之间存在相同的子树，则直接对相同的子树部分进行沿用，只保留其一个版本。以图中R1和R2 为例说明，在时间间隔[T0，T1]内，只有对象c1进行了位移，故R2 直接沿用了R1没有发生变化的子树B1和C1，仅新生成了A2 分支。并且在 A2分支中仍直接沿用了与R1相同的a1 和 b1，只新建了c2。 图1-2 HR树示意图 Hilbert R树提高了结点存储利用率，优化了R树结构[43]。Hilbert R树的思想是	的时间戳信息提取出来，并按照时间流逝的顺序对其进行排序，生成有序的时间戳列表，如图中的List(T0，T1，T2)。然后对各个时间戳下的移动对象根据其空间位置信息构建对应的 R-Tree，如图中的 R1、R2 和 R3。为了节省存储空间，HR-Tree 采用了“子树重叠”策略，策略的中心思想是：对于时间戳相邻的两棵R-Tree，若在该时间间隔内存在地理空间位置没有发生变化的对象，即两个树之间存在相同的子树，则直接对相同的子树部分进行沿用，只保留其一个版本。以图中R1 和 R2 为例说明，在时间间隔[T0，T1]内，只有对象c 进行了位移，故 R2 直接沿用了 R1 没有发生变化的子树 B1 和 C1，仅新生成了 A2 分支。并且在 A2 分支中仍直接沿用了与 R1 相同的 a1 和 b1，只新建了 c2。 R1 R3A1 B1 C1 A2C3R2a1 b1b1 c1 d1 e1 f1 g1 h1 i1 c2
19	此处有 341 字相似 生成了A2 分支。并且在 A2分支中仍直接沿用了与R1相同的a1 和 b1，只新建了c2。 图1-2 HR树示意图 Hilbert R树提高了结点存储利用率，优化了R树结构[43]。Hilbert R树的思想是利用Hilbert曲线对d维空间中的数据进行线性排序，进而对结点进行排序，从而得到面积、周长最小化的结点。QR*[44]树是一种基于四叉树和R*树的空间索引结构，它结合了四叉树和R*树的优点，可以将查询范围限定在索引空间的某一部分，然后再用类似R*树的查询算法进行查询，在数据量非常巨大的应用领域中提高了查询效率。 R树及其变种树在建树过程中存在最小外包矩形之间重叠的现象，在高维空间中该重叠现象尤为严重，空间查询操作将访问大量多余的结点。基于R树的空间查询算法的执行时间指数依赖于空间维数在高维空间中查询操作几乎是线性扫描整个数据空间。这种现象被称为“维数灾难”。 1.2.2.2 基于空间填充曲线的索引 空间填充曲线最初是由意大利科学家皮亚诺于1890 年构造的，由希尔伯特于1891年正式提出，之	基于空间填充曲线高维空间查询算法研究 徐红波 -《哈尔滨理工大学博士论文》- 2010-03-01（是否引证：否） 1.利用率，减少了结点分裂次数，使得索引结点的最小外包矩形更趋于正方形，从而显著提高了查询性能。Hilbert R 树提高了结点存储利用率，优化了 R 树结构。Hilbert R 树的思想是利用 Hilbert 曲线对 d 维空间中的数据进行线性排序，进而对结点进行排序，从而得到面积、周长最小化的结点。 QR*树是一种基于四叉树和R*树的空间索引结构，它结合了四叉树和R*树的优点，可以将查询范围限定在索引空间的某一部分，然后再用类似R*树的查询算法进行查询，在数据量非常巨大的应用领域中提高了查询效率。 由于其特殊的分类算法，Compact R[35]树的存储利用率几乎可以达到100%。在更新操作中 Compact 2.图 R 树[45]吸收了位图索引的思想；其他 R 树的变种树在文献[46]中有详细介绍。 R 树及其变种树在建树过程中存在最小外包矩形之间重叠的现象，在高维空间中该重叠现象尤为严重，空间查询操作将访问大量多余的结点。基于R 树的空间查询算法的执行时间指数依赖于空间维数 d[47]，在高维空间中查询操作几乎是线性扫描整个数据空间。这种现象被称为“维数灾难”[48]。 1.2.2 查询算法 由于空间对象的数据结构复杂，导致空间数据库中对空间对象的操作既 - 9 -
20	此处有 222 字相似 高维空间中查询操作几乎是线性扫描整个数据空间。这种现象被称为“维数灾难”。 1.2.2.2 基于空间填充曲线的索引 空间填充曲线最初是由意大利科学家皮亚诺于1890 年构造的，由希尔伯特于1891年正式提出，之后空间填充曲线得到了深入的研究和广泛的应用。空间填充曲线是一类可以“填充”N维数据空间的曲线[46-50]。按照特定的填充规则，空间填充曲线可以通过有限次数的近似运算将多维数据空间划分为许多非常小的网格。空间填充曲线像线一样穿过每个网格并按照线性顺序对这些网格编号，它对每个网格只访问 1 次，且互不交叉。空间填	基于空间填充曲线高维空间查询算法研究 徐红波 -《哈尔滨理工大学博士论文》- 2010-03-01（是否引证：否） 1.充曲线仅仅穿过每个网格一次。空间填充曲线按照线性顺序对这些高维空间中的网格进行统一编号。 空间填充曲线是由意大利科学家皮亚诺于 1890 年首次构造出来的，并由希尔伯特于 1891 年正式提出的，之后空间填充曲线就得到了深入的研究和广泛的应用[76-80]。空间填充曲线是一类可以将 d 维数据空间“填满”的曲线。按照特定的填充规则，空间填充曲线通过有限次数的逼近操作可以把多维数据空间划分为众多体积非常小的网格，且无论逼近操作的程度如何，总是能够发现一条连续的空间填充曲线通过所有网格而不相互重叠。 从数学的角度

	充曲线将空间数据点一一映射到一维空间。 常用的空间填充曲线包括Z-order和Hilbert两类。m 阶空间填充曲线将二维空间划分成大小相等、互不重叠的22m个	Z曲线网格划分的最近邻查询 刘润涛;陈琳琳;田广悦; -《计算机工程与应用》- 2012-12-03 1（是否引证：否） 1.询算法,考虑了数据点的分布和网格形状对查询的影响,能够得出精确的查询结果。2相关知识将数据空间划分成大小相等、互不重叠的网格,空间填充曲线像线一样穿过每个网格并按照线性顺序对这些网格编号,它对每个网格只访问1次,且互不交叉。空间填充曲线将空间数据点一一映射到一维空间。通常选取维数的2倍阶曲线对数据空间进行填充[8],也可根据点的实际分布情况选择适当阶数。m阶Z曲线将二维空间划分成大小相
21	此处有 56 字相似 空间划分成 2dm个网格，m 阶 Z 曲线相当于对坐标轴分割了 m 次，点所在网格的 x 坐标或 y 坐标的分量为 m个。 当维数d一定时，曲线的阶数越高，空间 分割越细，每个网格越小。以二维空间为例，3 阶 Z曲线将数据空间划分成 22 * 3=64 个网格。图1-3是3阶 Z 曲线以及Hilbert曲线的示意图，编码值反映的是网格的位序，由网格的编	Z曲线网格划分的最近邻查询 刘润涛;陈琳琳;田广悦; -《计算机工程与应用》- 2012-12-03 1（是否引证：否） 1.的实际分布情况选择适当阶数。m阶Z曲线将二维空间划分成大小相等、互不重叠的22m个网格,将d维空间划分成2dm个网格。当维数d一定时,曲线的阶数越高,空间分割越细,每个网格越小。以二维空间为例,3阶Z曲线将数据空间划分成22'3=26=64个网格。所分成的网格单元个数称为粒度。一维空间的每个网格为线段,二维空间每个网格为矩形,三维空间每
22	此处有 80 字相似 阶 Z曲线将数据空间划分成 22 * 3=64 个网格。图1-3是3阶 Z 曲线以及Hilbert曲线的示意图，编码值 反映的是网格的位序，由网格的编码值便可得知其在空间中的位置，而通过计算数据点的编码值即可知道点具体在哪个网格，编码值的计算通过对网格坐标进行位交叉操作得到。 图1-3 3阶空间填充曲线 1.3 主要研究内容 以上内容总结了在大数据的环境下，国内外研究机构或学者针对矢量大数据	Z曲线网格划分的最近邻查询 刘润涛;陈琳琳;田广悦; -《计算机工程与应用》- 2012-12-03 1（是否引证：否） 1.2表示数的二进制形式,也可化为等值十进制形式()10。本文规定凡是给出一个点,即为已知它的二进制网格坐标和空间坐标。Z值反映的是网格的位序。由网格的Z值便可得知其在空间中的位置,而通过计算数据点的Z值即可知道点具体在哪个网格。Z值的计算通过对网格坐标进行位交叉操作得到。3最近邻查询思想Hue Ling Chen等人提出的最近邻查询算法基本思想为[6-7]:只访问查询点所在网格及其周边网格
23	此处有 57 字相似 值即可知道点具体在哪个网格，编码值的计算通过对网格坐标进行位交叉操作得到。 图1-3 3阶空间填充曲线 1.3 主要 研究内容 以上内容总结了在大数据的环境下，国内外研究机构或学者针对矢量大数据相关方面开展的研究工作，其中部分研究 仅针对矢量数据，但不支持其时间维度的查询，无法进行时空查询；部分研究仅支持点数据对象的存储及查询，但不支持更为复杂的线、	姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.ark R-tree; Range Query;Spatial Join;KNN; 上海交通大学Utah（2016）研究现状评述以上内容总结了在大数据的环境下，国内外研究机构或学者针对矢量大数据相关方面开展的研究工作，从研究现状可以看出，目前，国内外针对矢量大数据方面已经开展了诸多相关的研究，大数据相关的技术、理论和方法已经逐步渗透到了GIS 矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.6) KNN;8中国农业大学博±学位论文 第一章绪论1.2.4研究现状评述W上内容总结了在大数据的环境下，国内外研巧机构或学者针对矢量大数据相关方面开展的研巧工作，从研巧现状可W看出，目前，国内外针对矢量大数据方面已经开展了诸多相关的研巧，大数据相关的技术、理论和方法已经逐步
24	此处有 114 字相似	基于Hadoop平台的时空数据索引和查询技术研究 何立志 -《西安电子科技大学硕士论文》- 2018-06-01（是否引证

法设计及优化策略

针对点线面矢量对象，设计时空k-NN查询算法。并通过考虑空间分布，自

适应设定搜索网格的边长，从而
提高k-NN查询的效率。

1.4 论文组织结构

本论文一共分为6个章节，各章节的内容安排如下所述：

第一章为绪论，首先介绍本课题的研究背景以及说明本研究的意义所在；然后简要阐述本课题研究的相关技术在国内外的研究和发展现状；
接着重点列举本课题所完成的主要工作内容；最后介绍本文的撰写逻辑与章节安排。

第二章则重点介绍Hadoop平台的概述，以

：否）

1.测试以及行键设计、并发访问数与集群节点数对查询性能影响的相关测试。并且实验后还对测试结果进行理论分析。1.4 论文章节安排本论文一共分为 6 个章节，各章节的内容安排如下所述：第一章为绪论，从 4 个方面对本文的基本轮廓进行概述。首先介绍本课题的研究背景以及说明本研究的意义所在；然后简要阐述本课题研究的

2.本论文一共分为 6 个章节，各章节的内容安排如下所述：第一章为绪论，从 4 个方面对本文的基本轮廓进行概述。首先介绍本课题的研究背景以及说明本研究的意义所在；然后简要阐述本课题研究的相关技术在国内外的研究和发展现状，该部分主要从 Hadoop 平台和时空数据库两个方面进行阐述；接着重点列举本课题所完成的主要工作内容；最后介绍本文

基于iOS的汽车租赁客户端的设计与实现 李文慧 -《大连理工大学硕士论文》- 2018-03-01（是否引证：否）

1.且经济型的小汽车处于市场的主导地位。目前规模较大的一些汽车租赁企业主要集中在北京、上海、深圳等一线城市，布局发展不太均衡。1.3 论文组织结构本论文一共分为六个章节，总体组织结构如下：第一章 绪论。介绍本文汽车租赁客户端系统的研究背景和国内领域现状，探讨本论文课题的意义。 基于 i OS 的汽车租赁客户端的设计与

基于知识图谱的Web of Scholars系统设计与实现 康文杰 -《大连理工大学硕士论文》- 2018-04-01（是否引证：否）

1.学者的关系网络。（11）共同被引关系模块研究：该模块主要描述了中心学者与共同被引学者的关系网络。1.4 论文组织结构论文一共分为七个章节，具体如下所示：第一章：绪论。绪论首先介绍了系统的背景与意义，提出存在的问题以及解决方法，针对国内外关于学者关系及知识图谱的内容展开调研，根据调研内容确的分析定本文的研究

改进K-means算法在文本聚类中的应用 周本金 -《中国工程物理研究院硕士论文》- 2018-05-01（是否引证：否）

1.上就是聚类中心向孤立点方向靠近，会导致簇的中心偏离实际聚类中心，而且随着孤立点比例增加，偏离程度越大。1.3?论文组织结构论文一共分为六个章节，结构如下图1.3所示：绪论部分分别介绍聚类和k-xneans研究现状;第二章介绍聚类的基本原理和方法;第三章是改进k-means算法的正文部分，针

基于Hadoop的IPTV故障预判算法的研究及系统实现 王堂辉 -《南京邮电大学硕士论文》- 2018-11-14（是否引证：否）

1.理功能主要是针对后台 hadoop 集群的管理，如集群机器的添加、删除、重命名、集群软件的升级等。 本论文的组织结构本论文一共分为六个章节，其中每个章节的具体安排如下：第一章，绪论。首先简要介绍了研究 IPTV 故障预判系统的背景和意义，以及建立 IPTV故障预判模型的关键难点。然后介绍了目前众多专家、学者在非均衡

姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否）

1.系统架构和功能进行了设计和实现，验证了本文研究内容的可行性、现实意义和使用价值；本文最后一章为本文的研究结论与展望。第一章，绪论部分，首先介绍

		了论文的研究背景和意义，围绕大数据时代的GIS、矢量大数据存储与管理以及矢量大数据并行化算法等方面阐述了国内外研究现状及其发展趋势，并进行了详细
25	此处有 45 字相似 一章为绪论，首先介绍本课题的研究背景以及说明本研究的意义所在；然后简要阐述本课题研究的相关技术在国内外的研究和发展现状； 接着重点列举本课题所完成的主要工作内容；最后介绍本文的撰写逻辑与章节安排。 第二章则重点 介绍Hadoop平台的概述，以及HBase数据库的系统架构，逻辑存储及物理存储。 第三章对时空矢量对象的索引设计及存储	基于Hadoop平台的时空数据索引和查询技术研究 何立志 - 《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1.要阐述本课题研究的相关技术在国内外的研究和发展现状，该部分主要从 Hadoop 平台和时空数据库两个方面进行阐述；接着重点列举本课题所完成的主要工作内容；最后介绍本文的撰写逻辑与章节安排。第二章则重点研究分析与本课题相关的知识技术，包括 Hadoop 与时空索引两部分。其中，Hadoop 相关技术的研究以 HDF
26	此处有 64 字相似 查询中最大递归次数与平均查询时间的性能对比。在k-NN查询算法中，则验证不同参数k下算法的性能以及考虑数据分布后k-NN 算法的优化。 第六章对本论文所做的技术工作与研究成果进行总结，并以此为基础提出进一步优化的方案，制定下一步的研究工作计划。	基于Hadoop平台的时空数据索引和查询技术研究 何立志 - 《西安电子科技大学硕士论文》- 2018-06-01（是否引证：否） 1.查询的算法实现与优化策略。西安电子科技大学硕士学位论文6然后对各种查询算法的性能进行相关测试并对测试结果进行理论分析。第六章对本论文所做的技术工作与研究成果进行总结，并以此为基础提出进一步优化的方案，制定下一步的研究工作计划。 第二章 相关技术概述7第二章 相关技术概述2.1 分布式云平台 Hadoop 相关概述Had

指　标

疑似剽窃文字表述

1. 1.2 国内外研究现状
   1.1.1 时空数据库的研究现状
   空间数据库是管理时空数据的有效手段，也是数据查询、分析应用的基础。目前，海量时空大数据的存储和管理主要采用三种方式：一是基于分布式扩展的关系型数据库进行存储；二是基于分布式文件系统进行存储；三是基于非关系型NoSQL数据库进行存储。
2. 非关系型NoSQL数据库
   NoSQL，即Not Only SQL，泛指非关系型数据库。
3. 由于非关系型NoSQL数据库具有分布式、可扩展性、不需要预先定义表结构等诸多优点，近年来深受研究学者和商业公司的青睐，在矢量大数据存储和管理领域也有很多研究工作。
4. 基于非关系型NoSQL数据库存储空间大数据不仅方便部署，同时也能够很好地对多源空间数据进行高效集成，有利于数据密集型的并行计算。
5. 时空索引技术
   时空数据是一种描述空间状态随时间变化的数据类型，有助于帮助人们了解过去、把握现在、预测未来，当前采集到的科学数据基本是带有时间属性的，时间维度的增加极大地丰富了空间对象的数据信息。时空数据查询是时空数据分析处理的重要手段，而时空数据索引技术是时空数据查询的基础。
6. 3DR-Tree可以被认为是R-Tree的多维扩展，两者都是树结构一致的。3DR-Tree的基本操作，包括节点插入和删除，节点分裂和节点查询，类似于R-Tr
7. HR-Tree的核心思想是：采用二级索引机制，首先将时间维度数据单独分割出来并为之构建第一级索引，然后再针对各个时间片下的空间数据用R-Tree作为第二级索引。
8. 为了节省存储空间，HR-Tree采用了“子树重叠”策略，策略的中心思想是：对于时间戳相邻的两棵R-Tree，若在该时间间隔内存在地理空间位置没有发生变化的对象，即两个树之间存在相同的子树，则直接对相同的子树部分进行沿用，只保留其一个版本。
9. Hilbert R树的思想是利用Hilbert曲线对d维空间中的数据进行线性排序，进而对结点进行排序，从而得到面积、周长最小化的结点。

10. R树及其变种树在建树过程中存在最小外包矩形之间重叠的现象，在高维空间中该重叠现象尤为严重，空间查询操作将访问大量多余的结点。基于R树的空间查询算法的执行时间指数依赖于空间维数在高维空间中查询操作几乎是线性扫描整个数据空间。这种现象被称为“维数灾难”。
    1.2.2.2
11. 空间填充曲线最初是由意大利科学家皮亚诺于1890 年构造的，由希尔伯特于1891年正式提出，之后空间填充曲线得到了深入的研究和广泛的应用。
12. 按照特定的填充规则，空间填充曲线可以通过有限次数的近似运算将多维数据空间划分为许多非常小的网格。空间填充曲线像线一样穿过每个网格并按照线性顺序对这些网格编号，它对每个网格只访问 1 次，且互不交叉。空间填充曲线将空间数据点一一映射到一维空间。
13. 当维数d一定时，曲线的阶数越高，空间
    分割越细，每个网格越小。以二维空间为例，3 阶 Z曲线将数据空间划分成
14. 反映的是网格的位序，由网格的编码值便可得知其在空间中的位置，而通过计算数据点的编码值即可知道点具体在哪个网格，编码值的计算通过对网格坐标进行位交叉操作得到。
15. 研究内容
    以上内容总结了在大数据的环境下，国内外研究机构或学者针对矢量大数据相关方面开展的研究工作，其中部分研究
16. 提高k-NN查询的效率。
    1.4 论文组织结构
    本论文一共分为6个章节，各章节的内容安排如下所述：
    第一章为绪论，首先介绍本课题的研究背景以及说明本研究的意义所在；然后简要阐述本课题研究的相关技术在国内外的研究和发展现状；接着重点列举本课题所完成的主要工作内容；最后介绍本文的撰写逻辑与章节安排。
    第二章则重点
17. 算法的优化。
    第六章对本论文所做的技术工作与研究成果进行总结，并以此为基础提出进一步优化的方案，制定下一步的研究工作计划。

3. 第2章技术理论基础

总字数：7343

相似文献列表

去除本人已发表文献复制比：42.7%(3133)　　文字复制比：42.7%(3133)　　疑似剽窃观点：(0)

1	Java之美[从菜鸟到高手演练]之Hadoop原理及架构 - 智慧演绎，无处不在 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》- 2017	20.0%（1472） 是否引证：否
2	分布式计算框架Hadoop - John's blog - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》- 2013	19.8%（1457） 是否引证：否
3	分布式计算框架Hadoop - yarsen的专栏 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》- 2017	19.8%（1457） 是否引证：否
4	被动雷达系统控制平台的设计和实现初稿2.0(1) -《学术论文联合比对库》- 2017-04-24	14.6%（1072） 是否引证：否
5	3911173__电子政务云平台的设计与实现 -《学术论文联合比对库》- 2017-03-28	14.5%（1062） 是否引证：否
6	基于Hadoop的电子政务平台设计与实现 龙琦(导师：刘晓玲;王全贵) -《湖南大学硕士论文》- 2017-04-10	14.4%（1059） 是否引证：否
7	面向教学应用的虚拟现实模型构建及场景管理关键技术研究 -《学术论文联合比对库》- 2013-03-22	12.6%（928） 是否引证：否
8	bs_随_201021111111 随 -《学术论文联合比对库》- 2013-03-28	12.6%（928） 是否引证：否
9	201021110001-l1l_7 l -《学术论文联合比对库》- 2013-04-12	12.6%（923） 是否引证：否
10	基于Hadoop的空间矢量数据的分布式存储与查询研究 陈俊欣(导师：张凤荔) -《电子科技大学硕士论文》- 2016-03-18	9.5%（698） 是否引证：否
11	201321060644陈俊欣 -《学术论文联合比对库》- 2016-03-21	9.3%（680） 是否引证：否
12	一种ABE的设计与实现 -《学术论文联合比对库》- 2015-04-08	8.5%（626） 是否引证：否
13	李军 李军 -《学术论文联合比对库》- 2015-09-13	7.3%（537） 是否引证：否
14	媒体稿件管理平台的设计与实现 汉超(导师：孔令波) -《北京交通大学硕士论文》- 2017-06-01	6.5%（480） 是否引证：否
15	921_J201163342_冯学龙	6.3%（465）

	冯学龙 - 《学术论文联合比对库》- 2015-05-15	是否引证：否
16	企业电子发票信息系统的设计与实现	5.8%（427）
	朱口天 - 《学术论文联合比对库》- 2017-09-26	是否引证：否
17	论文9.30——董再旺	5.7%（420）
	董再旺 - 《学术论文联合比对库》- 2014-09-30	是否引证：否
18	921_J201163342_冯学龙	5.5%（404）
	冯学龙 - 《学术论文联合比对库》- 2015-05-26	是否引证：否
19	Hadoop分布式文件系统架构和源码分析报告-百度文库	5.5%（402）
	-《互联网文档资源（http://wenku.baidu.c）》- 2012	是否引证：否
20	HBase多条件复杂查询的实现方法研究	5.4%（398）
	廖一陈(导师：沈波) -《北京交通大学硕士论文》- 2017-03-01	是否引证：否
21	98_陈俊欣_基于Hadoop的空间矢量数据的分布式存储与查询研究	5.4%（396）
	陈俊欣 - 《学术论文联合比对库》- 2016-03-10	是否引证：否
22	基于Hadoop的数据统计系统的设计与实现	4.2%（307）
	叶溟 - 《学术论文联合比对库》- 2014-06-11	是否引证：否
23	27-叶溟	4.2%（307）
	叶溟 - 《学术论文联合比对库》- 2014-06-06	是否引证：否
24	Hadoop平台下基于移动充电器模式的电动汽车充电研究	3.5%（259）
	张祥民 - 《学术论文联合比对库》- 2017-03-09	是否引证：否
25	姚晓闯_B1311686_矢量大数据管理关键技术研究	2.4%（178）
	姚晓闯 - 《学术论文联合比对库》- 2017-06-01	是否引证：否
26	赵龙_S20153080844_基于Hadoop的全国耕地连片度计算方法研究	1.8%（130）
	赵龙 - 《学术论文联合比对库》- 2017-11-27	是否引证：否
27	基于大数据技术的人口数据分析平台设计与实现	1.7%（126）
	杨麟(导师：章韵) -《南京邮电大学硕士论文》- 2018-11-14	是否引证：否
28	数据通信网分布式测量系统的设计与实现	1.3%（99）
	尚立(导师：高会生;许俊现) -《华北电力大学硕士论文》- 2018-06-01	是否引证：否
29	基于MapReduce的结构化查询机制的设计与实现	1.2%（91）
	范波(导师：段翰聪) -《电子科技大学硕士论文》- 2011-03-25	是否引证：否
30	使用HBase Coprocessor协处理器 - - CSDN博客	1.1%（82）
	-《网络（http://blog.csdn.net）》- 2017	是否引证：否
31	交通安防大数据的实时快速检索关键技术研究	0.5%（40）
	雷力(导师：刘鹏) -《浙江大学硕士论文》- 2017-01-14	是否引证：否

原文内容		相似内容来源
	此处有 2253 字相似	Java之美[从菜鸟到高手演练]之Hadoop原理及架构 - 智慧演绎，无处不在 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》-（是否引证：否）
1	第2章技术理论基础 2.1 Hadoop概述 Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。 Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，而MapReduce则为海量的数据提供了计算。 1.1.1 分布式文件系统HDFS HDFS [51]架构原理采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名	1.式计算存储提供了底层支持。采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数量积达到大型主机处理性能。 HDFS 架构原理 HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。 DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点(当然也不排除每个物理节点可以有多个DataNode，不过生产环境里不建议这么做)。DataNode负责管理节点上它们拥 2.DataNode都对应于一个物理节点(当然也不排除每个物理节点可以有多个DataNode，不过生产环境里不建议这么做)。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同

空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。图2-1为HDFS系统架构图，主要有三个角色，Client、NameNode、DataNode。

图2-1HDFS架构图

文件写入时：Client向NameNode发起文件写入的请求。NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。Client将文件划分为多个block块，并根据DataNode的地址信息，按顺序写入到每一个DataNode块中。当文件读取：Client向NameNode发起文件读取的请求。NameNode返回文件存储的block块信息、及其block块所在DataNode的信息。Client读取文件信息。HDFS 数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。

备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。副本的选择：为了降低整体的带宽消耗和读取延时，HDFS会尽量读取最近的副本。如果在同一个rack上有一个副本，那么就读该副本。如果一个HDFS集群跨越多个数据中心，那么将首先尝试读本地数据中心的副本。安全模式：系统启动后先进入安全模式，此时系统中的内容不允许修改和删除，直到安全模式结束。安全模式主要是为了启动检查各个DataNode上数据块的安全性。

1.1.2 并行计算框架MapReduce

MapReduce[52]编程模式是由Google于2004年研发，主要用于超大集群环境下TB级海量数据的并行处理和计算。MapReduce模型由Map“映射”阶段和Reduce“规约”两个阶段组成。它是以一组键值对Key-Value的集合作为输入，而另一对键值对为输出，因此这种编程模式特别适合于非结构化和结构化的海量数据的搜索、挖掘，分析和机器学习等。Map Reduce 框架这种“分而治之”的思想运用有许多方面，当数据量足够大的时候，许多操作就会分派节点来进行，整个集群中则是主节点 Job Tracker 管理下的各个子节点 Task Tracker 一起来完成

时周期性的将其所有的block块信息发送给NameNode。 下图为HDFS系统架构图，主要有三个角色，Client、NameNode、DataNode。 文件写入时Client向NameNode发起文件写入的请求。NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。 Client将文件划分为多个block块，并根据DataNode的地址信息，按顺序写入到每一个DataNode块中。 当文件读取Client向NameNode发起文件读取的请求。 NameNode返回文件存储的block块信息、及其block块所在DataNode的信息。 Client读取文件信息。 HDFS 数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。 备份数据的存放 备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。 副本的选择 为了降低整体的带宽消耗和读取延时，HDFS会尽量读取最近的副本。如果在同一个rack上有一个副本，那么就读该副本。如果一个HDFS集群跨越多个数据中心，那么将首先尝试读本地数据中心的副本。 安全模式 系统启动后先进入安全模式，此时系统中的内容不允许修改和删除，直到安全模式结束。安全模式主要是为了启动检查各个DataNode上数据块的安全性。 MapReduce来源 MapReduce是由Google在一篇论文中提出并广为流传的。它最早是Google提出的一个软件架构，用于大规模数据集群分布式运算。任务的分解（Ma

分布式计算框架Hadoop - John's blog - 博客频道 - CSDN.NET - 《网络（http://blog.csdn.net）》- （是否引证：否）

1.式计算存储提供了底层支持。采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数量积达到大型主机处理性能。 HDFS 架构原理 HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。 NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。 DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。 下图

，然后所得到每个子节点的中间结果，合并得到最终的结果。在整个并行计算的任务处理过程中，会抽象出两个函数：Map和Reduce函数。Map 会把任务分解成多个小任务，Reduce 则会整合分解后的多小任务来得到最终处理的结果.。Map Reduce 框架一直都是围绕一个键值对来进行的，无论是数据分割和数据合并，都是从一个输入到一个输出的键值对，它最关键的两个函数就是Map和 Reduce，Map 完成的任务就是得到键值对的中间值，这是需要函数输出的一系列相同的值，而 Reduce 函数则是接收 Map 函数的输出，主要做一个键值对的合并关键操作，这样就能得到大量数据的一个分布式多线程操作，最后的键值对会保存下来作为记录输出，当然它也是一个键值对。

从不同的需求来看，Map 和 Reduce 函数都是需要根据不同的需要而重新编写的，这样就可以利用这个并行的计算框架来解

为HDFS系统架构图，主要有三个角色，Client、NameNode、DataNode。 文件写入时： Client向NameNode发起文件写入的请求。 NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。 Client将文件划分为多个block块，并根据DataNode的地址信息，按顺序写入到每一个DataNode块中。 当文件读取： Client向NameNode发起文件读取的请求。 NameNode返回文件存储的block块信息、及其block块所在DataNode的信息。 Client读取文件信息。 HDFS 数据备份 HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。 备份数据的存放： 备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。 副本的选择： 为了降低整体的带宽消耗和读取延时，HDFS会尽量读取最近的副本。如果在同一个rack上有一个副本，那么就读该副本。如果一个HDFS集群跨越多个数据中心，那么将首先尝试读本地数据中心的副本。 安全模式： 系统启动后先进入安全模式，此时系统中的内容不允许修改和删除，直到安全模式结束。安全模式主要是为了启动检查各个DataNode上数据块的安全性。 MapReduce MapReduce 来源 MapReduce是由Google在一篇论文中提出并广为流传的。它最早是Google提出的一个软件架构，用于大规模

分布式计算框架Hadoop - yarsen的专栏 - 博客频道 - CSDN.NET - 《网络（http://blog.csdn.net）》- （是否引证：否）

1.式计算存储提供了底层支持。采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数量积达到大型主机处理性能。 HDFS 架构原理 HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。 NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。 DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。 下图为HDFS系统架构图，主要有三个角色，Client、

NameNode、DataNode。 文件写入时： Client向NameNode发起文件写入的请求。 NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。 Client将文件划分为多个block块，并根据DataNode的地址信息，按顺序写入到每一个DataNode块中。 当文件读取： Client向NameNode发起文件读取的请求。 NameNode返回文件存储的block块信息、及其block块所在DataNode的信息。 Client读取文件信息。 HDFS 数据备份 HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。 备份数据的存放： 备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。 副本的选择： 为了降低整体的带宽消耗和读取延时，HDFS会尽量读取最近的副本。如果在同一个rack上有一个副本，那么就读该副本。如果一个HDFS集群跨越多个数据中心，那么将首先尝试读本地数据中心的副本。 安全模式： 系统启动后先进入安全模式，此时系统中的内容不允许修改和删除，直到安全模式结束。安全模式主要是为了启动检查各个DataNode上数据块的安全性。 MapReduce MapReduce 来源 MapReduce是由Google在一篇论文中提出并广为流传的。它最早是Google提出的一个软件架构，用于大规模

3911173_电子政务云平台的设计与实现 -《学术论文联合比对库》- 2017-03-28（是否引证：否）

1.分布式技术平台构建云平台，需要运用相关分布式技术平台，其中Hadoop是当前热门平台之一，也是本课题选择的关键技术。Hadoop是Apache的开源分布式计算平台，其可在不了解分布式底层细节的情况下，快速实现在分布式存在的各软硬件资源上构建起云平台，为用户提供有效的云平台构造解决方案，实现海量的数据处理和并发式运算处理

2.中数据存储管理的基础。它是一个高度容错的系统，能检测和应对硬件故障，用于在低成本的通用硬件上运行。（1）架构原理HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名

空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。下图为HDFS系统架构图，主要有三个角色，Client、NameNode、DataNode。图2. 4HDFS系统架构图（2）数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成bloc

3.主要有三个角色，Client、NameNode、DataNode。图2. 4HDFS系统架构图（2）数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。2.4.2MapReduce分布式计算模型MapReduce 是一种计算模型，用以进行大数据量的计算。Hadoop

被动雷达系统控制平台的设计和实现初稿2.0(1) -《学术论文联合比对库》- 2017-04-24（是否引证：否）

1.发性的运算需求，本课题拟采用Hadoop大数据处理技术，有效利用分布式基础设施资源，为本管控平台提供强大的运算能力。Hadoop概述Hadoop是Apache的开源分布式计算平台，其可在不了解分布式底层细节的情况下，快速实现在分布式存在的各软硬件资源上构建起云平台，为用户提供有效的云平台构造解决方案，实现海量的数据处理和并发式运算处理

2.中数据存储管理的基础。它是一个高度容错的系统，能检测和应对硬件故障，用于在低成本的通用硬件上运行。（1）架构原理HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。下图为

HDFS系统架构图，主要有三个角色，Client、NameNode、DataNode。图2. 5HDFS系统架构图（2）数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成bloc

3.主要有三个角色，Client、NameNode、DataNode。图2. 5HDFS系统架构图（2）数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。MapReduce分布式计算模型MapReduce 是一种计算模型，用以进行大数据量的计算。Hadoop 的MapR

基于Hadoop的电子政务平台设计与实现 龙琦 -《湖南大学硕士论文》- 2017-04-10（是否引证：否）

1.术平台构建云平台，需要运用相关分布式技术平台，其中Hadoop是当前热门平台之一，也是本课题选择的关键技术。Hadoop是Apache的开源分布式计算平台，其可在不了解分布式底层细节的情况下，快速实现在分布式存在的各软硬件资源上构建起云平台，为用户提供有效的云平台构造解决方案，实现海量的数据处理和并

2.基础。它是一个高度容错的系统，能检测和应对硬件故障，用于在低成本的通用硬件上运行。(1)架构原理HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。1! 基于Hadoop的电子政务平台设计与实现对于整个集群来说，HDFS通过NameNode对用户提供了一个

3.息、以及每一个block块归属的DataNode的信息。1! 基于Hadoop的电子政务平台设计与实现对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给 NameNode。下图为HDFS系统架构图，主要有三个角色，Client、

NameNode、DataNode。HDFS Architecture/ \ Metadata (Name, replicas,...):Metadata

4. yRack 1 \Writez Rack 2戀图2.4 HDFS系统架构图 (2)数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。HDFS中的文件默认规则是write one (—次写、多次读）的，并且严格要求在任何时候只有一个writer。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一 rack的DataNode上，一个在指定DataNode同一 rack的DataNode上。这种策略综合12工程硕士学位论文考虑了同一rack失效、以及不同rack之间数据复制性能问题。2.4.2 MapReduce分布式计算模型MapReduce是一种计算模型，用以进行大数据量的计算。Hadoop

面向教学应用的虚拟现实模型构建及场景管理关键技术研究 -《学术论文联合比对库》- 2013-03-22（是否引证：否）

1.量数据存储系统的体系结构，系统分布式存储与分布式计算的实现。设计了一个存储检索系统，并对系统的性能特点进行了说明。Hadoop是由Apache基金会开发的分布式系统基础架构，是一个易操作的平台，其最核心的模块是分布式文件系统HDFS和分布式计算模型Map/Reduce。Hadoop是由多台，乃

2.educe。Hadoop是由多台，乃至上千台PC服务器组成的系统集群，能够以一种可靠、容错的方式分布式处理客户端的请求。用户可以在不了解分布式底层细节的情况下，开发分布式程序，充分利用集群的威力高速运算和存储。图1 hadoop框架结构在Hadoop系统中，有一台master机和多台slave机，master机主要负责NameNode

3.eNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问，维护着文件系统内所有的文件和目录，它会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块所属的DataNode的信息。这些信息以两个文件形式保存在本地磁盘上：命名空间镜像文件和编辑日志文件。除了数据块的位置信息是在系统启动时由数据节点重建

4.空间镜像文件和编辑日志文件。除了数据块的位置信息是在系统启动时由数据节点重建的之外，其它的信息是永久保存在本地磁盘上的。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多

个，通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。HDFS将所有的文件存储成block块组成的序列，除了最后一块，所有的blocks块大小都一样，文件的所有block块都会因为容

5.Tracker的工作，TaskTracker则是主要负责在本地数据上执行Map/Reduce任务。分布式文件系统HDFSHDFS是一个可以跨机器存储海量数据的框架，采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数据来达到大型主机处理的性能，它为分布式计算存储提供了底

6.为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。HDFS将所有的文件存储成block块组成的序列，除了最后一块，所有的blocks块大小都一样，文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，也可以在后期修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer而且写操作总是将数据添在文件末尾，它不支持具有多个写入者的操作，也不支持在文件的任意位置修改。NameNode负责管

7.何时候只有一个writer而且写操作总是将数据添在文件末尾，它不支持具有多个写入者的操作，也不支持在文件的任意位置修改。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。6.4.3分布式计算模型 Map/ReduceMap/Reduce最早是由Google提出的一个软件架构，用于大规

bs_随_201021111111 随 -《学术论文联合比对库》- 2013-03-28（是否引证：否）

1.量数据存储系统的体系结构，系统分布式存储与分布式计算的实现。设计了一个存储检索系统，并对系统的性能特点进行了说明。Hadoop是由Apache基金会开发的分布式系统基础架构，是一个易操作的平台，其最核心的模块是分布式文件系统HDFS和分布式计算模型Map/Reduce。Hadoop是由多台，乃

2.educe。Hadoop是由多台，乃至上千台PC服务器组成的系统集群，能够以一种可靠、容错的方式分布式处理客户端的请求。用户可以在不了解分布式底层细节的情况下，开发分布式程序，充分利用集群的威力高速运算和存储。图1 hadoop框架结构在Hadoop系统中，有一台master机和多台slave机，master机主要负责NameNode

3.eNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问，维护着文件系统内所有的文件和目录，它会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块所属的DataNode的信息。这些信息以两个文件形式保存在本地磁盘上：命名空间镜像文件和编辑日志文件。除了数据块的位置信息是在系统启动时由数据节点重建

4.空间镜像文件和编辑日志文件。除了数据块的位置信息是在系统启动时由数据节点重建的之外，其它的信息是永久保存在本地磁盘上的。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个，通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。HDFS将所有的文件存储成block块组成的序列，除了最后一块，所有的blocks块大小都一样，文件的所有block块都会因为容

5.Tracker的工作，TaskTracker则是主要负责在本地数据上执行Map/Reduce任务。分布式文件系统HDFSHDFS是一个可以跨机器存储海量数据的框架，采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数据来达到大型主机处理的性能，它为分布式计算存储提供了底

6.为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。HDFS将所有的文件存储成block块组成的序列，除了最后一块，所有的blocks块大小都一样，文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，也可以在后期修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer而且写操作总是将数据添在文件末尾，它不支持具有多个写入者的操作，也不支持在文件的任意位置修改。NameNode负责管

7.何时候只有一个writer而且写操作总是将数据添在文件末尾，它不支持具有多个写入者的操作，也不支持在文件的任意位置修改。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。6.4.3分布式计算模型 Map/ReduceMap/Reduce最早是由Google提出的一个软件架构，用于大规

04-12（是否引证：否）

1.实用性的开发平台，淘宝就是国内率先使用Hadoop的公司之一。设计一个存储检索系统，并对系统的性能特点进行了说明。Hadoop是由Apache基金会开发的分布式系统基础架构，是一个易操作的平台，其最核心的模块是分布式文件系统HDFS和分布式计算模型Map/Reduce。Hadoop是由多台，乃

2.educe。Hadoop是由多台，乃至上千台PC服务器组成的系统集群，能够以一种可靠、容错的方式分布式处理客户端的请求。用户可以在不了解分布式底层细节的情况下，开发分布式程序，充分利用集群的威力高速运算和存储。(c)三维模型的内容检索三维建模过程中，需要用到三维模型检索技术。三维模型检索首先需要对三维模型进行特征提取得到特

3.eNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问，维护着文件系统内所有的文件和目录，它会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块所属的DataNode的信息。这些信息以两个文件形式保存在本地磁盘上：命名空间镜像文件和编辑日志文件。除了数据块的位置信息是在系统启动时由数据节点重建

4.空间镜像文件和编辑日志文件。除了数据块的位置信息是在系统启动时由数据节点重建的之外，其它的信息是永久保存在本地磁盘上的。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个，通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。HDFS将所有的文件存储成block块组成的序列，除了最后一块，所有的blocks块大小都一样，文件的所有block块都会因为容

5.Tracker的工作，TaskTracker则是主要负责在本地数据上执行Map/Reduce任务。分布式文件系统HDFSHDFS是一个可以跨机器存储海量数据的框架，采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数据来达到大型主机处理的性能，它为分布式计算存储提供了底

6.为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。HDFS将所有的文件存储成block块组成的序列，除了最后一块，所有的blocks块大小都一样，文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，也可以在后期修改。HDFS中的文件默认规则是write one（一次写、多次读）的，并且严格要求在任何时候只有一个writer而且写操作总是将数据添在文件末尾，它不支持具有多个写入者的操作，也不支持在文件的任意位置修改。NameNode负责管

7.何时候只有一个writer而且写操作总是将数据添在文件末尾，它不支持具有多个写入者的操作，也不支持在文件的任意位置修改。NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据

包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。6.4.3 分布式计算模型 Map/ReduceMap/Reduce最早是由Google提出的一个软件架构，用于大

一种ABE的设计与实现 -《学术论文联合比对库》- 2015-04-08（是否引证：否）

1.（MapReduce）的基础组件，同时也是BigTable（如HBase、 HyperTable）的底层分布式文件系统。HDFS采用master/slave架构。一个HDFS集群是有由一个Namenode和一定数目的 Datanode组成。Namenode是一个中心服务器，负责管理文件系统的namespace和客户端对文件

2.群中一般是一个节点一个，负责管理节点上它们附带的存储。在内部，一个文件其实分成一个或多个block，这些block存储在Datanode集合里。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。图3.1是HDFS系统的示意图，它包括三个部分：Client、NameNod和DataNode。图3.1 HDFS系统示意图当文件被写入到HDFS时，Client会向NameNode发起文件写入的请求。由Name

3.意图，它包括三个部分：Client、NameNod和DataNode。图3.1 HDFS系统示意图当文件被写入到HDFS时，Client会向NameNode发起文件写入的请求。由NameNode根据文件大小和文件块配置情况，返回给Client它所管理的DataNode的信息。Client把文件划分为若干个block，并根据DataNode的地址，按照顺序写入到每一个DataNode中。当文件从HDFS中被读取时，Client向NameNode发起文件读取的请求。NameNode返回文件存储的block信息、及其block所在DataNode的信息。Client负责读取文件信息。可见NameNode就像主人一样指挥调度DataNode。Hadoop的MapReduce起源于Google公司里的

李军 李军 -《学术论文联合比对库》- 2015-09-13（是否引证：否）

1.的代码实现，并最终实现整个系统。根据实际功能，进行了测试用例设计，并进行测试。第6章，结论。对全文进行总结。第2章相关技术研究2.1 Hadoop架构Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下

，开发分布式程序。充分利用集群的威力进行高速运算和存储。通过对Hadoop分布式计算平台最核心的分布式文件系统HDFS、MapReduce处理过程，以及数据仓库工具Hive和分布式数据库Hb

2.个典型例子就是在网络数据上运行的搜索算法。Hadoop，最初只与网页索引有关，迅速发展成为分析大数据的领先平台。Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。图4.1 系统总体框架图Fig. 4.1 the overall framework of the system diagram

3.ce，而两者只是理论基础，不是具体可使用的高级应用，Hadoop旗下有很多经典子项目，比如HBase、Hive等。其中，HDFS采用主从（Master/Slave）结构模型，一个HDFS集群是由一个NameNode和若干个DataNode组成的（在最新的Hadoop2.2版本已经实现多个NameNode的配置-这也是一些大公司通过修改hadoop源代码

4.ent。NameNode是管理者，DataNode是文件存储者、Client是需要获取分布式文件系统的应用程序。文件写入：1 Client向NameNode发起文件写入的请求。2 NameNode根据文件大小和文件块配置情况，返回给Client它管理的DataNode的信息。3 Client将文件划分为多个block，根据DataNode的地址，按顺序将block写入DataNode块中。文件读取：1 Client向NameNode发起读取文件的请求。2 NameNode返回文件存储的DataNode信息。3 Client读取文件信息。HDFS作为分布式文件系统在数据管理方面可借鉴点：文件

5.Node信息。3 Client读取文件信息。HDFS作为分布式文件系统在数据管理方面可借鉴点：文件块的放置：一个Block会有三份备份，一份在NameNode指定的DateNode上，一份放在与指定的DataNode不在同一台机器的DataNode上，一根在于指定的DataNode在同一Rack上的DataNode上。备份的目的是为了数据安全，采用这种方式是为了考虑到同一Rack失败的情况，以及不同数据拷贝带来的性能的问题。2.1.

媒体稿件管理平台的设计与实现_汉超-《北京交通大学硕士论文》-2017-06-01（是否引证：否）

1.专业学位论文扩展，可根据自己的需求扩展控件。其正在以快速迭代的方式完善和提高其功能特点。2.6 Hadoop 介绍Hadoop是一个由Apache基金会所开发的分布式系统基础架构[13]。Hadoop的框架最核心的设计就是：HDFS (Hadoop Distributed File System)和

2.adoop的框架最核心的设计就是：HDFS (Hadoop Distributed File System)和 MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算HDFS架构原理：HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多

个。通常每一个DataNode都对应于一个物理节点。主要负责数据的备份。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。其架构图如图2-3所示。HDFS Architecture~—— ^ Metadata (Name, rep

921_J201163342_冯学龙_冯学龙 -《学术论文联合比对库》- 2015-05-15（是否引证：否）

1.；ASP.NET同时还集成.NET Framwork，使得许多功能都适用于Web程序[12]。2.2 分布式计算相关技术 2.2.1 Hadoop相关技术Hadoop是Apache软件基金会所开发的并行计算框架与分布式文件系统。最核心的模块包括Hadoop Common、HDFS与MapReduce[13]。HDFS

2.层支持[14]。采用Java语言开发，可以部署在多种普通的廉价机器上，以集群处理数量积达到大型主机处理性能[15]。HDFS 架构原理：HDFS采用master/slave架构。一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问[16]。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode[17]。HDFS系统架构图如图2-1所示，主要有三个角色，Client、NameNode、DataNode。图2-1 HDFS架构图MapReduce是由Google在一篇论文中提出并广为流传的。它最早是Google提出的一个软件架构，用于大规模

201321060644陈俊欣 -《学术论文联合比对库》- 2016-03-21（是否引证：否）

1.个开源系统。MapReduce与Hadoop Common、HDFS —起，构成了 Hadoop的最核心的部分[19]。MapReduce框架这种“分而治之”的思想运用有许多方面，当数据量足够大的时候，许多操作就会分派节点来进行，整个集群中则是主节点JobTracker管理下的各个子节点TaskTracker一起来完成，然后所得到每个子节点的中间结果，合并得到最终的结果。在整个并行计算的任务处理过程中，我们会抽象出两个函数：Map和Reduce函数。Map会把任务分解成多个小任务，Reduce则会整合分解后的多小任务来得到最终处理的结果[19]。MapReduce框架一直都是围绕一个键值<key,value>[20]对来进行的,无论是数据分割和数据合并，都是从一个输入到一个输出的键值对，它最关键的两个函数就是Map和Reduce，Map完成的任务就是得到键值对的中间值，这是需要函数输出的一系列相同的值，而Reduce函数则是接收Map函数的输出，主要做一个键值对的合并关键操作，这样就能得到大量数据的一个分布式多线程操作，最后的键值对会保存下来作为记录输出，当然它也是一个键值对<key,value>。图2-1介绍了MapReduce的整个过程。从不同的需求来看，Map和Reduce函数都是需要根据不同的需

企业电子发票信息系统的设计与实现_朱口天-《学术论文联合比对库》- 2017-09-26（是否引证：否）

1.，采用搭建Hadoop+HBase的大数据存储系统，为发票信息数据和发票版式文件提供高性能、易扩展的非结构化数据库。Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高

2.S放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。如下图所示，Hadoop分布式文件存储系统（HDFS）将基于一个存储集群，形成分布式存储资源池，其具体特点如下：图4-5 分布式文件存储系统（1）高效：在进行文件存

3.，NameNode、DataNode、Zookeeper、HMaster、HRegionServer的具体功能如下：NameNode：负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体块的信息、以及每个块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode：负责数据存储，将存储划分为多个块，同时周期性的将其所有的块存储信息发送给NameNode。Zookeeper：

4.及每个块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode：负责数据存储，将存储划分为多个块，同时周期性的将其所有的块存储信息发送给NameNode。Zookeeper：Zookeeper存储ROOT-表地址、HMaster地址；HRegionServer把自己以Eph

基于Hadoop的空间矢量数据的分布式存储与查询研究_陈俊欣-《电子科技大学硕士论文》- 2016-03-18（是否引证：否）

1. Reduce 与 Hadoop Common、HDFS 一起，构成了 Hadoop 的最核心的部分[19]。Map Reduce 框架这种“分而治之”的思想运用有许多方面，当数据量足够大的时候，许多操作就会分派节点来进行，整个集群中则是主节点 Job Tracker 管理下的各个子节点 Task Tracker 一起来完成，然后所得到每个子节点的中间结果，合并得到最终的结果。在整个并行计算的任务处理过程中，我们会抽象出两个函数：Map 和Reduce 函数。Map 会把任务分解成多个小任务，Reduce 则会整合分解后的多小任务来得到最终处理的结果[19]。Map Reduce 框架一直都是围绕一个键值<key,value>[20]对来进行的,无论是数据分割和数据合并，都是从一个输入到一个输出的键值对，它最关键的两个函数就是Map 和 Reduce，Map 完成的任务就是得到键值对的中间值，这是需要函数输出的一系列相同的值，而 Reduce 函数则是接收 Map 函数的输出，主要做一个键值对的合并关键操作，这样就能得到大量数据的一个分布式多线程操作，最后的键值对会第二章 分布式存储与并行处理技术概述7保存下来作为记录输出，当然它也是一个键值对<key,value>。图 2-

论文9.30——董再旺_董再旺 -《学术论文联合比对库》- 2014-09-30（是否引证：否）

1.章的延伸，将分析第三章、第四章国家域名日志可视化分析监控系统设计和实现过程中需要使用到的相关技术。2.1 数据分析技术2.1.1 Hadoop介绍Hadoop是一个分布式系统基础架构，由Apache基金会所开发。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储[11]。Hadoop实现了一个分布式文件系统（Hadoop Distributed File System），简称HDFS。HDFS有高

2.S放宽了（relax）POSIX的要求，可以以流的形式访问（streaming access）文件系统中的数据。Hadoop的框架最核心的设计就是：HDFS和MapReduce.HDFS为海量的数据提供了存储，则MapReduce为海量的数据提供了计算。Hadoop是Doug Cutting（Apache Lucene）开发的使用广泛的文本搜索库。Hadoop起源于Apa

3.地将所有存在的Block信息发送给NameNode。Client就是需要获取分布式文件系统文件的应用程序。2）文件写入，Client向NameNode发起文件写入的请求。NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。Client将文件划分为多个Block，根据DataNode的地址信息，按顺序写入到每一个DataNode块中。3）文件读取，Client向NameNode发起文件读取的请求。NameNode返回文件存储的DataNode的信息。Client读取文件信息。图2-1 HDFS总体结构示意图3.HDFS的优缺点

4.，MapReduce框架负责处理了并行编程中分布式存储、工作调度、负载均衡、容错均衡、容错处理以及网络通信等复杂问题，把处理过程高度抽象为两个函数：map和reduce，map负责把任务分解成多个任务，reduce负责把分解后多任务处理的结果汇总起来。需要注意的是，用MapReduce来处理的数据集（或任务）必须具备这样的特点：待处理的数据集可

921_J201163342_冯学龙_冯学龙 -《学术论文联合比对库》- 2015-05-26（是否引证：否）

1.；ASP.NET同时还集成.NET Framwork，使得许多功能都适用于Web程序[12]。2.2 分布式计算相关技术2.2.1 Hadoop相关技术Hadoop是由Apache软件基金会开发一套开源系统，它提供了并行计算框架和分布式文件系统的实现。最核心的模块包括Hadoop Common、分布式文件系统与分布式计算框架[

2.[15]。HDFS 架构采用主-从结构。对于一个HDFS集群来说，它由一个单独的名称结点和多个数据结点组成。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问[16]。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode[17]。HDFS系统架构图如图2-1所示，主要有三个角色

，Client、NameNode、DataNode。图2-1 HDFS架构图MapReduce基于Google对于分布式计算的概念而提出的。最初是由于Google需要对自身的海量数据进行分布

Hadoop分布式文件系统架构和源码分析报告-百度文库 -《互联网文档资源（http://wenku.baidu.c）》- （是否引证：否）

1.要角色： NameNode、 DataNode 和 Client。 以下通过三个操作来说明它们之间的交互关系： 1、文件写入： Client 向 NameNode 发起文件写入的请求。 NameNode 根据文件大小和文件块配置情况，返回给 Client 它所管理 部分 DataNode 的信息。 Client 将文件划分为多个 Block，根据 DataNode 的地址信息，按顺序 写入到每一个 DataNode 块中。 2、 文件读取： Client 向 NameNode 发起文件读取的请求。 NameNode 返回文件存储的 DataNode 的信息。 Client 读取文件信息。 3、 文件 Block 复制： NameNode 发现 部

2. 保存的文件的副本数目。文件副本的数目称为文件的副本系数，这个信息也是由 Namenode 保存的。 1.4 数据复制 HDFS 被设计成能够在一个大集群中跨机器可靠地存储超大文件。它将每个 文件存储成一系列的数据块，除了最后一个数据块外，其余所有的数据块都是同样大小的。为了容错，文件的所有数据块都会有副本。每个文件

3.序可以指定某个文件的副本数目。副本系数可以 在文件创建的时候指定，也可以在之后改变。HDFS 中的文件都是一次性写入的， 并且严格要求在任何时候只能有一个写入者。 Namenode 全权管理数据块的复 制，它周期性地从集群中的每个 Datanode 接收心跳信号和块状态报告 (Blockreport)。接收到心跳信号意味着该 Datanode 节点工作正常。块状态报告 包含了

4.通讯需要经过交换机。在大多数情况下，同一个机架内的两台机器 间的带宽会比不同机架的两台机器间的带宽大。通过一个机架感知的过程， Namenode 可以确定每个 Datanode 所属的机架 id。 一个简单但没有优化的策略就 是将副本存放在不同的机架上。这样可以有效防止当整个机架失效时数据的丢 失，并且允许读数据的时候充

5.程序进行读取的过程中，为了降低整体的带宽消耗和读取延时，HDFS 会尽量让读取程序读取离它最近的副本。 如果在读取程序的同一个机架上有一个 副本，那么就读取该副本。如果一个 HDFS 集群跨越多个数据中心，那么客户端 也将首先读本地数据中心的副本。 还有一点值得注意的是 Namenode 启动后会进 入一个称为安全模式的特殊状态。处于安全模式的 Namenode 是

基于Hadoop的数据统计系统的设计与实现 叶溟 -《学术论文联合比对库》- 2014-06-11（是否引证：否）

1.处理大规模数据的软件平台,是Appach的一个用Java语言实现开源软件框架，实现在大量计算机组成的集群中对海量数据进行分布式计算。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力高速运算和存储。Hadoop框架中最核心设计就是：HDFS和MapReduce。HDFS提供了海量数据的存储，MapReduce提供了对数据的计算。以下将对

HDFS和Map Reduce做详细的介绍。2.1.1 HDFSHadoop Distributed

2.ode以获取文件的元数据，而真正的文件I/O操作是直接和DataNode进行交互的。文件的写入流程如下：1.Client向NameNode发起文件写入的请求。2.NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。3.Client将文件划分为多个Block，根据DataNode的地址信息，按顺序写入到每一个DataNode块中。文件的读取流程如下：1.Client向NameNode发起文件读取的请求。2.NameNode返回文件存储的DataNode的信息。3.Client读取文件信息。图2–2 HDFS的体系结构2.1.2 Map

27-叶溟 叶溟 -《学术论文联合比对库》- 2014-06-06（是否引证：否）

1.处理大规模数据的软件平台,是Appach的一个用Java语言实现开源软件框架，实现在大量计算机组成的集群中对海量数据进行分布式计算。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力高速运算和存储。Hadoop框架中最核心设计就是：HDFS和MapReduce。HDFS提供了海量数据的存储，MapReduce提供了对数据的计算。以下将对HDFS和Map Reduce做详细的介绍。2.1.1 HDFSHadoop Distributed

2.ode以获取文件的元数据，而真正的文件I/O操作是直接和DataNode进行交互的。文件的写入流程如下：1.Client向NameNode发起文件写入的请求。2.NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。3.Client将文件划分为多个Block，根据DataNode的地址信息，按顺序写入到每一个DataNode块中。文件的读取流程如下：1.Client向NameNode发起文件读取的请求。2.NameNode返回文件存储的DataNode的信息。3.Client读取文件信息。图 2–2 HDFS的体系结构2.1.2 Ma

Hadoop平台下基于移动充电器模式的电动汽车充电研究 张祥民 -《学术论文联合比对库》- 2017-03-09（是否引证：否）

1.ssingHadoop由HDFS、MapReduce、HBase、Hive和ZooKeeper等子项目组成。其中，最核心的设计为HDFS和MapReduce，HDFS为海量数据提供了存储，MapReduce为海量数据提供了计算。图2.2为Hadoop生态系统图。图2. 2 Hadoop生态系统图Fig2. 2 The ecosystem

2.程序的数据，大大提高了整个系统的数据吞吐量，非常适合应用于具有超大规模数据集的应用程序中。(1)HDFS体系架构HDFS的体系架构采用主从结构(master/slave)，如图2.4所示。一个典型的HDFS集群包含一个NameNode节点和多个DataNode节点。图2. 4 HDFS部署结构图Fig2. 4 The deployment structure diagram of

3. 5 HDFS file write operationHDFS文件写操作如图2.5所示。具体写操作过程如下：①Client向NameNode发起文件写入的请求；②NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息；③Client将文件划分为多个文件块

，根据DataNode的地址信息，按顺序写入到每一个DataNode块中。2)文件读操作图2. 6 HDFS文件读操作Fig2. 6 HDFS file read operationHDFS

98_陈俊欣_基于Hadoop的空间矢量数据的分布式存储与查询研究 陈俊欣 -《学术论文联合比对库》- 2016-03-10（是否引证：否）

1. Hadoop的最核心部分[19]。MapReduce框架所采用“分而治之”的思想，把海量数据比较复杂的操作分发给一个主节点JobTracker管理下的各个子节点TaskTracker一起来完成，然后所得到每个子节点的中间结果，合并得到最终的结果。在整个并行计算的任务处理过程中，我们会抽象出两个函数：Map和Reduce函数。Map会把任务分解成多个小任务，Reduce则会整合分解后的多小任务来得到最终处理的结果[19]。MapReduce 编程模型的原理是：通过一个已知输入的键值对<key,value>来计算产生另一个输出的键值对<key,value>[20

姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否）

1.盘空间下降到一定程度，按照均衡策略，系统会自动把数据从这个节点移动到其他数据节点。MapReduce并行计算模型MapReduce[120]编程模式是由Google于2004年研发，主要用于超大集群环境下TB级海量数据的并行处理和计算[121]。MapReduce模型由Map“映射”阶段和Reduce映射“规约”两个阶段组成。它是以一组键值对（Key/Value）的集合作为输入，而以另一对键值对为输出，因此这种编程模式特别适合于非结构化和结构化的海量数据的搜索、挖掘、分析和机器学习等。Hadoop云计算环境下进行程序研发过程当中，开发人员可以投入更多的精力来集中编写Map和Reduce两个函数的内容，其

基于大数据技术的人口数据分析平台设计与实现 杨麟 -《南京邮电大学硕士论文》- 2018-11-14（是否引证：否）

1.体分析、社交网络分析和移动分析这 6 个关键技术领域。（1）Hadoop 技术 Hadoop 是一中分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop 实现了一个分布式文件系统（Hadoop Distributed File System），简称 HDFS。HDFS

2.POSIX 的要求，可以以流的形式访问（streaming access）文件系统中的数据。Hadoop 的框架最核心的设计就是：HDFS 和 Map Reduce。HDFS 为海量的数据提供了存储，则 Map Reduce 为海量的数据提供了计算。图 2.1 Hadoop 生态系统组成图如图 2.1 所示，Hadoop 由许多元素构成。其最底部是 Hadoop

赵龙_S20153080844_基于Hadoop的全国耕地连片度计算方法研究 赵龙 -《学术论文联合比对库》- 2017-11-27（是否引证：否）

1.平台；第二种是提供分布式存储和分布式计算能力的云计算平台，如基于HDFS和MapReduce的Hadoop平台[34]。Hadoop是一个由Apache基金会所开发的分布式系统基础架构，它主要由分布式文件系统HDFS和并行数据处理系统MapReduce两部分组成，在海量数据存储与计算方面有良好性能，目前国

2.高度容错性的系统，适合部署在廉价的机器上，且能提供高吞吐量的数据访问，非常适合大规模数据集上的

		应用。如图2-3所示，HDFS采用主从（master/slave）架构。一个HDFS集群是由一个主节点（Namenode）和多个子节点（Datanodes）组成。主节点作为中心服务器，主要负责存储与文件系统相关的元数据和客户端对文件的访问，元数据 3.pReduce论文，发表于2004年12月，Hadoop MapReduce是google MapReduce 克隆版，主要用于超大集群环境下TB级海量数据的并行处理和计算[56]。MapReduce是一种计算模型，用以进行大数据量的计算。它分为“Map（映射）”阶段和“Reduce（规约）”阶段，其中Map对数 数据通信网分布式测量系统的设计与实现 尚立 -《华北电力大学硕士论文》- 2018-06-01（是否引证：否） 1.有一些大型的商用软件和系统，也有一些免费或者开源的应用软件。下面主要介绍两种可用于网络测量的平台和系统。2.2.1 HadoopHadoop 是一个由 Apache 基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。Hadoop 实现了一个分布式文件系统（Hadoop DistributedFile System），简称 HDFS。HDF
2	此处有 233 字相似 对的合并关键操作，这样就能得到大量数据的一个分布式多线程操作，最后的键值对会保存下来作为记录输出，当然它也是一个键值对。 从不同的需求来看，Map 和 Reduce 函数都是需要根据不同的需要而重新编写的，这样就可以利用这个并行的计算框架来解决许多分布式的关键问题，比如分布式存储与数据处理，工作调度、负载平衡，数据传输等等。因此，利用Map Reduce 处理海量的空间矢量数据是一个比较好的选择，利用Map 函数归一化然后将结果交给Reduce函数处理。采用Map Reduce 的计算模式将传统的空间查询操作并行化，理论上可以有效提高空间数据查询的效率。 2.2 数据库HBase 介绍 创建于2007年2月的HBase[53-56]项目，是Google BigTable的开源实现，在Hadoop生	基于Hadoop的空间矢量数据的分布式存储与查询研究 陈俊欣 -《电子科技大学硕士论文》- 2016-03-18（是否引证：否） 1.来作为记录输出，当然它也是一个键值对<key,value>。图 2-1 介绍了Map Reduce 的整个过程。从不同的需求来看，Map 和 Reduce 函数都是需要根据不同的需要而重新编写的[21]，这样就可以利用这个并行的计算框架来解决许多分布式的关键问题，比如分布式存储与数据处理，工作调度、负载平衡，数据传输等等。图 2-1 Map Reduce 处理大数据集的过程因此，利用 Map Reduce 处理海量的空间矢量数据是一个比较好的选择，利用Map 函数归一化然 2.等。图 2-1 Map Reduce 处理大数据集的过程因此，利用 Map Reduce 处理海量的空间矢量数据是一个比较好的选择，利用Map 函数归一化然后将结果交给 Reduce 函数处理[22]。采用 Map Reduce 的计算模式将传统的空间查询操作并行化，理论上可以有效提高空间数据查询的效率。2.3 分布式数据库 HBase 概述 HBase 数据库也可以被称作是 Hadoop 数据库。目前 HBase 已经有可用的稳定版本，HB 201321060644陈俊欣 -《学术论文联合比对库》- 2016-03-21（是否引证：否） 1.的键值对会保存下来作为记录输出，当然它也是一个键值对<key,value>。图2-1介绍了MapReduce的整个过程。从不同的需求来看，Map和Reduce函数都是需要根据不同的需要而重新编写的[21]，这样就可以利用这个并行的计算框架来解决许多分布295275127317500式的关键问题，比如分布式存储与数据处理，工作调度、负载平衡，数据传输等等。图2-1 2.问题，比如分布式存储与数据处理，工作调度、负载平衡，数据传输等等。图2-1 MapReduce处理大数据集的过程因此，利用MapReduce处理海量的空间矢量数据是一个比较好的选择，利用Map函数归一化然后将结果交给Reduce函数处理[22]。采用MapReduce的计算模式将传统的空间查询操作并行化，理论上可以有效提

		高空间数据查询的效率。2.3 分布式数据库HBase概述 HBase数据库也可以被称作是Hadoop数据库。目前 HBase已经有可用的稳定版本，HBase数据库能够成为 H 98_陈俊欣_基于Hadoop的空间矢量数据的分布式存储与查询研究_陈俊欣 -《学术论文联合比对库》- 2016-03-10（是否引证：否） 1.负载平衡等不需要直接编写程序去解决，其上的框架已经在帮我们实现了。图2-1 MapReduce处理大数据集的过程因此，利用MapReduce处理海量的空间矢量数据是一个比较好的选择，利用Map函数归一化然后将结果交给Reduce函数处理[22]。采用MapReduce的计算模式将传统的空间查询操作并行化，理论上可以有效提高空间数据查询的效率。2.3分布式数据库HBase概述 HBase数据库也可以被称做是Hadoop数据库。目前 HBase已经有可用的稳定版本，HBase数据库能够成为 H
3	此处有 43 字相似 er的负载较低，也不会成为集群查询的瓶颈。如此的架构设计，使得HBase有着相当出色的负载均衡和容错的性能。 1.1. 2 HBase表的逻辑存储与物理存储 HBase数据与存储模型在数据模型上，HBase采取与Google BigTable相同的数据模型，以表的形式存储数据，表具体的逻辑视图如表2-1所示。HBase底层	98_陈俊欣_基于Hadoop的空间矢量数据的分布式存储与查询研究_陈俊欣 -《学术论文联合比对库》- 2016-03-10（是否引证：否） 1.次则是逻辑模型所表现的层次，列族就表现在物理模型中，表中的物理视图的空NULL是不会被HBase表所存储的。2.3.2 HBase表的逻辑存储以及物理存储 2.3.2.1 HBase表的逻辑存储HBase表的逻辑存储模型可以看做是一个有序映射的映射集合，从逻辑视图中，HBase可以视为键值数据存储的另一种视角，HBase
4	此处有 54 字相似 存储数据，表具体的逻辑视图如表2-1所示。HBase底层存储采用Key-Value形式存储，表拥有一个大的映射关系，其中 行键（RowKey）、列族（Column Family）、列限定符（Column Qualifier）和时间 戳（TimeStamp）共同组成了Key，而Value由这4维坐标唯一确定。行键是每行数据记录的唯一标识，类似于关系型数	基于Hadoop的空间矢量数据的分布式存储与查询研究_陈俊欣 -《电子科技大学硕士论文》- 2016-03-18（是否引证：否） 1.数据模型，存储的过程中不会在意数据类型。2.3.1 HBase 表视图HBase 表的逻辑视图是基于行键（rowkey）、列族（column family）、列限定符（column qualifier）和时间版本（version）。这一组数据就是数据存储的一个单位，整个单位就是一个键值对，对应的值（value）就是真实 201321060644陈俊欣 -《学术论文联合比对库》- 2016-03-21（是否引证：否） 1.许灵活的、动态的数据模型，存储的过程中不会在意数据类型。2.3.1 HBase表视图HBase表的逻辑视图是基于行键（rowkey）、列族（column family）、列限定符（column qualifier）和时间版本（version）。这一组数据就是数据存储的一个单位，整个单位就是一个键值对，对应的值（value）就是真实的数据信 98_陈俊欣_基于Hadoop的空间矢量数据的分布式存储与查询研究_陈俊欣 -《学术论文联合比对库》- 2016-03-10（是否引证：否） 1.许灵活的、动态的数据模型，存储的过程中不会在意数据类型。2.3.1 HBase表视图HBase表的逻辑视图是基于行键（rowkey）、列族（column family）、列限定符（column qualifier）和时间版本（version）。把这一组坐标看做是键，对应的单元数据就是一个值（value），每个元素值保存了同一份数据的多个版

5	此处有 45 字相似 难，HBase将数据的写入过程做了一个中转，先把新数据缓存到MemStore，利用内存的高速读写能力做好排序工作，然后再 持久化到HDFS。 StoreFile是只读文件，一旦创建无法修改，所以HBase的更新 操作没有删除改写，而是一个将新数据以一个个StoreFile文件不断append至数据库中的过程。当StoreFile文	HBase多条件复杂查询的实现方法研究 廖一陈 -《北京交通大学硕士论文》- 2017-03-01（是否引证：否） 1.，等待刷新成一个StoreFile。同时，在ZooKeeper中记录一个重做起点，用来表示此时刻前的数据变更已经持久化。StoreFile是只读文件，一旦创建，无法修改，所W HBase的更新是一个不断追加的过程。当一个8北京交通大学硕±学位论文 相关技术研巧Store中StoreFil
6	此处有 118 字相似 所以HBase的更新操作没有删除改写，而是一个将新数据以一个个StoreFile文件不断append至数据库中的过程。当 StoreFile文件数量超过一定阈值后，会自发地触发一次合并（compact）操作，对相同RowKey的数据项作合并，生成一个更大的StoreFile。当然StoreFile也不会无限扩大，如果StoreFile的文件大小达到阈值时， 又会平均地分裂（split）该StoreFile文件为两个新的StoreFile。因为排序工作已经在内存中完成了，Sto	HBase多条件复杂查询的实现方法研究 廖一陈 -《北京交通大学硕士论文》- 2017-03-01（是否引证：否） 1.8北京交通大学硕±学位论文 相关技术研巧Store中StoreFile的数量达到某个阀值时，又会进行一次合并操作，将对相同RowKey的数据修改合并，生成一个更大的StoreFile。当StoreFile的文件大小达到阈值时，会对StoreFile进行分裂操作，平均分裂为两个新的StoreFile。所更新是不断追加的过程，读请求时，都要 媒体稿件管理平台的设计与实现 汉超 -《北京交通大学硕士论文》- 2017-06-01（是否引证：否） 1.并成一个StoreFile，同时进行版本合并和数据删除-> 当81〇?^丨165Compact后，逐步形成越来越大的StoreFile ->单个StoreFile大小超过一定阈值后，触发Split操作，把当前Region Split成2个Region，Region会下线，新Split出的2个孩 基于MapReduce的结构化查询机制的设计与实现 范波 -《电子科技大学硕士论文》- 2011-03-25（是否引证：否） 1.gion 的合并和拆分示意图?每一次对 Memcache 的操作都会创建新的 StoreFile，当磁盘上的 StoreFile 文件数量增长到一定数量，将会触发合并 Compact 的操作，将多个 StoreFile 合并成一个 StoreFile，合并的过程会进行版本的合并和数据删除。合并
7	此处有 42 字相似 值时，又会平均地分裂（split）该StoreFile文件为两个新的StoreFile。因为排序工作已经在内存中完成了，StoreFile中的数据都是有序的，而且StoreFile包含了内存索引，所以合并 效率特别高。HBase即以这样的（compact-split）机制，控制着StoreFile数量与大小之间的平衡。 表	HBase多条件复杂查询的实现方法研究 廖一陈 -《北京交通大学硕士论文》- 2017-03-01（是否引证：否） 1.中全部的StoreFile数据和MemStore数据，并按照RowKey进行合并。由于StoreFile和MemStore都是有序的，而且StoreFile包含内存索引，所合并进行得很迅速。2.1.3 HBase查询机制在HBase中，很多的操作都是通过HRegionServer
8	此处有 41 字相似 列族 Store Cell单元值 StoreFile/HFile --- MemStore --- HLog 除了StoreFile和MemStore，数据还以HLog的格式存储在HDFS上， 以防在系统出错或宕机时，如果某个节点掉线了，其MemStore的内存数据就不会丢失。HLog的原理与	HBase多条件复杂查询的实现方法研究 廖一陈 -《北京交通大学硕士论文》- 2017-03-01（是否引证：否） 1.le包含HFile; memStore存储在内存中，S化reFileW HFile格式保存在HDFS上。除了 Hfile和memStore，数据还可能Hlog File的格式存储在HDFS上，因为在分布式环境中，系统出错和岩机是常态，如果某个HRegionServer掉线，其MemStore中的

	MemStore相似	
9	此处有 38 字相似 onServer都有一个日志对象，当数据写入时，同时在MemStore和HLog中刷写一份相同的数据，HLog定期作数据 持久化，然后覆盖已存在的数据。当RegionServer节点掉线了，Mast er主机会向Zookeeper发消息，Master会先处理HLog，将不同Region的HLog数据进行划分，放到对应的	HBase多条件复杂查询的实现方法研究 廖一陈 -《北京交通大学硕士论文》- 2017-03-01（是否引证：否） 1.Log对象，每次写入Memstore时，也写到HLog中。HLog定期持久化数据到StoreFile中，并删除已持久化数据。当HRegionServer 掉线后，HMaster 通过ZooKe巧er 感知，HMaster 会先处理 HLog,将不同Region的HLog数据进行划分，放到
10	此处有 142 字相似 作数据持久化，然后覆盖已存在的数据。当RegionServer节点掉线了，Master主机会向Zookeeper发消息， Master会先处理HLog，将不同Region的HLog数据进行划分，放到对应的Region目录下，然后重新分配失效的Region。分配到Region的节点在加载Region的过程中，如果有HLog，就会读取其数据写入MemStore中，然后刷新到StoreFile，完成数据恢复。 HBase的数据模型与实际物理存储模型之间的对应关系大致如表2-2所示。 1.1.3 HBase协处理器 HBase	HBase多条件复杂查询的实现方法研究 廖一陈 -《北京交通大学硕士论文》- 2017-03-01（是否引证：否） 1.e中，并删除已持久化数据。当HRegionServer 掉线后，HMaster 通过 ZooKe巧er 感知，HMaster 会先处理 HLog,将不同Region的HLog数据进行划分，放到对应的民egion目录下，然后重新分配失效的Region,分配到Region的HRegionServer在加载Region的过程中，如果发现有HLog,就会读取HLog中的数据写入MemStore中，然后刷新到StoreFiles，完成数据恢复。综上，数据在更新时首先写入MemS化re和HLog,在MemStore中进行排序，当MemStore中数据量到达 基于MapReduce的结构化查询机制的设计与实现 范波 -《电子科技大学硕士论文》- 2011-03-25（是否引证：否） 1.gionServer 异常下线，它处理下线的 RegionServer 遗留在底层分布式文件系统上的日志文件，将不同 Region 的日志数据进行分离，保存到相应的 Region 目录中，然后将失效的 Region 重新分配。负责这些失效 Region 的 RegionServer 会在加载这些 Region 的时候，会发现有遗留的
11	此处有 83 字相似 1.3 HBase协处理器 HBase与传统的关系型数据库相比，其写入性能高了一个数量级，但其查询性能要低一个数量级。 通常访问HBase数据的方式是，使用Scan方法做全表扫描或者Get方法直接获取，根据需要可以使用Filter过滤掉冗余的部分数据，最后在获取到的数据上进行业务运算。 但如果查询结果非常庞大，对如此大体量的数据做传输势必造成极大的网络延迟，而且在客户端还要进行复杂的运算，对服务器的内存和	使用HBase Coprocessor协处理器 - - CSDN博客 -《网络（http://blog.csdn.net）》-（是否引证：否） 1.破坏。此外，因为没有资源隔离，一个即使不是恶意设计的但表现不佳的Coprocessor也会严重影响集群的性能和稳定性。通常我们访问HBase的方式是使用scan或get获取数据，使用Filter过滤掉不需要的部分，最后在获取到的数据上进行业务运算。但是在数据量非常大的时候，比如一个有上亿行及十万个列的数据集，再按常用的方式移动获取数据就会在网络层面遇到瓶颈。客户端也
12	此处有 41 字相似 Server进程内运行的框架，可在运行期间动态加载，然后执行相应的功能。HBase Coprocessor的运行原理非常 类似于MapReduce的分析处理组件，但极大简化了MapReduce的处理模型。 MapReduce的加载在初始化阶段耗时长，且每次运行又要重新加载，难以满足线上请求的即时性；而Coprocessor偏	交通安防大数据的实时快速检索关键技术研究 雷力 -《浙江大学硕士论文》- 2017-01-14（是否引证：否） 1.本引入的新特性，其研发思路来源于GoogleBigTable中的Coprocessor。HBase协处理器是一个类似于MapReduce的分析处理组件，但它极大的简化了 MapReduce处理模型，让子任务独立地在各个存储节点上并行运行。它采用相同的设计思路即移动计算的代价远比移动数据低，把31

指　标
疑似剽窃文字表述

1. 第2章技术理论基础
   2.1 Hadoop概述
   Hadoop是一个由Apache基金会所开发的分布式系统基础架构。用户可以在不了解分布式底层细节的情况下，开发分布式程序。充分利用集群的威力进行高速运算和存储。 Hadoop的框架最核心的设计就是：HDFS和MapReduce。HDFS为海量的数据提供了存储，而MapReduce则为海量的数据提供了计算。
2. 一个HDFS集群包含一个单独的NameNode和多个DataNode。NameNode作为master服务，它负责管理文件系统的命名空间和客户端对文件的访问。NameNode会保存文件系统的具体信息，包括文件信息、文件被分割成具体block块的信息、以及每一个block块归属的DataNode的信息。对于整个集群来说，HDFS通过NameNode对用户提供了一个单一的命名空间。DataNode作为slave服务，在集群中可以存在多个。通常每一个DataNode都对应于一个物理节点。DataNode负责管理节点上它们拥有的存储，它将存储划分为多个block块，管理block块信息，同时周期性的将其所有的block块信息发送给NameNode。图2-1为HDFS系统架构图，主要有三个角色，Client、NameNode、DataNode。
   图2-1HDFS架构图
   文件写入时：Client向NameNode发起文件写入的请求。NameNode根据文件大小和文件块配置情况，返回给Client它所管理部分DataNode的信息。Client将文件划分为多个block块，并根据DataNode的地址信息，按顺序写入到每一个DataNode块中。当文件读取：Client向NameNode发起文件读取的请求。NameNode返回文件存储的block块信息、及其block块所在DataNode的信息。Client读取文件信息。HDFS 数据备份HDFS被设计成一个可以在大集群中、跨机器、可靠的存储海量数据的框架。它将所有文件存储成block块组成的序列，除了最后一个block块，所有的block块大小都是一样的。文件的所有block块都会因为容错而被复制。每个文件的block块大小和容错复制份数都是可配置的。容错复制份数可以在文件创建时配置，后期也可以修改。
3. NameNode负责管理block块的复制，它周期性地接收集群中所有DataNode的心跳数据包和Blockreport。心跳包表示DataNode正常工作，Blockreport描述了该DataNode上所有的block组成的列表。
   备份数据的存放是HDFS可靠性和性能的关键。HDFS采用一种称为rack-aware的策略来决定备份数据的存放。通过一个称为Rack Awareness的过程，NameNode决定每个DataNode所属rack id。缺省情况下，一个block块会有三个备份，一个在NameNode指定的DataNode上，一个在指定DataNode非同一rack的DataNode上，一个在指定DataNode同一rack的DataNode上。这种策略综合考虑了同一rack失效、以及不同rack之间数据复制性能问题。副本的选择：为了降低整体的带宽消耗和读取延时，HDFS会尽量读取最近的副本。如果在同一个rack上有一个副本，那么就读该副本。如果一个HDFS集群跨越多个数据中心，那么将首先尝试读本地数据中心的副本。安全模式：系统启动后先进入安全模式，此时系统中的内容不允许修改和删除，直到安全模式结束。安全模式主要是为了启动检查各个DataNode上数据块的安全性。
4. MapReduce模型由Map“映射”阶段和Reduce“规约”两个阶段组成。它是以一组键值对Key-Value的集合作为输入，而另一对键值对为输出，因此这种编程模式特别适合于非结构化和结构化的海量数据的搜索、挖掘，分析和机器学习等。Map Reduce 框架这种“分而治之”的思想运用有许多方面，当数据量足够大的时候，许多操作就会分派节点来进行，整个集群中则是主节点 Job Tracker 管理下的各个子节点 Task Tracker 一起来完成，然后所得到每个子节点的中间结果，合并得到最终的结果。在整个并行计算的任务处理过程中，会抽象出两个函数：Map和Reduce函数。Map 会把任务分解成多个小任务，Reduce 则会整合分解后的多小任务来得到最终处理的结果.。Map Reduce 框架一直都是围绕一个键值对来进行的，无论是数据分割和数据合并，都是从一个输入到一个输出的键值对，它最关键的两个函数就是Map和Reduce，Map 完成的任务就是得到键值对的中间值，这是需要函数输出的一系列相同的值，而Reduce 函数则是接收 Map 函数的输出，主要做一个键值对的合并关键操作，这样就能得到大量数据的一个分布式多线程操作，最后的键值对会保存下来作为记录输出，当然它也是一个键值对。从不同的需求来看，Map 和 Reduce 函数都是需要根据不同的需要而重新编写的，这样就可以利用这个并行的计算框架来解决许多分布式的关键问题，比如分布式存储与数据处理，工作调度、负载平衡，数据传输等等。因此，利用Map Reduce 处理海量的空间矢量数据是一个比较好的选择，利用Map 函数归一化然后将结果交给Reduce函数处理。采用Map Reduce 的计算模式将传统的空间查询操作并行化，理论上可以有效提高空间数据查询的效率。
5. 2 HBase表的逻辑存储与物理存储
   HBase数据与存储模型在数据模型上，HBas
6. 持久化到HDFS。
   StoreFile是只读文件，一旦创建无法修改，所以HBase的更新
7. StoreFile中的数据都是有序的，而且StoreFile包含了内存索引，所以合并
8. 除了StoreFile和MemStore，数据还以HLog的格式存储在HDFS上，
9. Master会先处理HLog，将不同Region的HLog数据进行划分，放到对应的Region目录下，然后重新分配失效的Region。分配到Region的节点在加载Region的过程中，如果有HLog，就会读取其数据写入MemStore中，然后刷新到StoreFile，完成数据恢复。
10. 通常访问HBase数据的方式是，使用Scan方法做全表扫描或者Get方法直接获取，根据需要可以使用Filter过滤掉冗余的部分数据，最后在获取到的数据上进行业务运算。
11. 类似于MapReduce的分析处理组件，但极大简化了MapReduce的处理模型。

4. 第3章时空矢量对象的索引设计及存储

总字数：8789

相似文献列表

去除本人已发表文献复制比：13.3%(1165)　　文字复制比：13.3%(1165)　　疑似剽窃观点：(0)

1	201091303526472 -《学术论文联合比对库》- 2013-05-28	9.5%（833） 是否引证：否
2	基于Redis的矢量数据组织研究 张景云(导师：江南) -《南京师范大学硕士论文》- 2013-04-18	9.4%（824） 是否引证：是
3	Java实现数据序列化工具Avro的例子 - 专注于大数据技术研究和应用 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》- 2017	2.3%（201） 是否引证：否
4	AVRO - u013061459的博客 - CSDN博客 -《网络（http://blog.csdn.net）》- 2017	1.9%（170） 是否引证：否
5	Avro序列化操作（1）：环境搭建和Schema处理 - hua245942641的专栏 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》- 2017	1.9%（167） 是否引证：否
6	徐迎晓_13222010150_陈钦彦_基于iOS的数据收集客户端的设计与实现 徐迎晓 -《学术论文联合比对库》- 2016-03-26	1.7%（151） 是否引证：否
7	Google Protocol Buffer持久化框架分析_rotosix -《网络（http://blog.sina.com）》- 2013	1.7%（149） 是否引证：否
8	Hadoop 生态系统 -《网络（http://www.aiweibang）》- 2016	1.6%（140） 是否引证：否
9	021_GS132147C_姚飞 姚飞 -《学术论文联合比对库》- 2016-05-04	1.2%（106） 是否引证：否
10	SA15225048_罗鹏_1 罗鹏 -《学术论文联合比对库》- 2017-09-19	1.2%（103） 是否引证：否
11	智能电视个性化推荐系统的设计与实现 罗鹏 -《学术论文联合比对库》- 2017-09-19	1.2%（103） 是否引证：否
12	Spark SQL下Parquet内幕深度解密 - 囧芝麻的博客 - CSDN博客 -《网络（http://blog.csdn.net）》- 2017	1.2%（103） 是否引证：否
13	1401220045_李长亮_幼教校园管理系统的设计与实现_荆琦_552e0567efbcedc80b87d7d631eeb0e9_20171011231145 李长亮 -《学术论文联合比对库》- 2017-10-13	0.9%（80） 是否引证：否
14	幼教校园管理系统的设计与实现 -《学术论文联合比对库》- 2017-10-11	0.9%（80） 是否引证：否
15	陈晓佳_143520085211005_面向个性化电台的实时大数据分析系统设计与实现 陈晓佳 -《学术论文联合比对库》- 2016-04-18	0.9%（76） 是否引证：否
16	MapReduce容错机制的改进研究 吴慧城(导师：李肯立) -《湖南大学硕士论文》- 2014-05-15	0.5%（40） 是否引证：否
17	基于云计算技术的化合物相似性分析系统 李杰辉(导师：张亮) -《复旦大学硕士论文》- 2012-04-25	0.4%（35） 是否引证：否
18	张琦 张琦 -《学术论文联合比对库》- 2016-01-09	0.4%（32） 是否引证：否
19	云计算环境下资源调度策略的研究与实现 李振双(导师：张洪欣) -《北京邮电大学硕士论文》- 2014-01-06	0.3%（30） 是否引证：否
20	李振双_2011140567_云计算环境下资源调度策略的研究与实现 李振双 -《学术论文联合比对库》- 2014-01-15	0.3%（30） 是否引证：否
21	030121221142 -《学术论文联合比对库》- 2014-01-02	0.3%（30） 是否引证：否

	原文内容	相似内容来源
1	此处有 833 字相似 第3章时空矢量对象的索引设计及存储 3.1 矢量对象 本文遵循国际标准化组织制定的标准简单要素规范，进行相应矢量数据模型和结构的设计，简单要素规范以	201091303526472 -《学术论文联合比对库》- 2013-05-28（是否引证：否） 1.。本文遵循OGC国际标准化组织制定的标准简单要素规范（Simple Features Specification）进行相应矢量数据模型和结构的设计，简单要素规范以制定一个标准的基于ODBC API的SQL方案为目的，并使其能够支持简单地理要素集的存储、提取、查询、更新。简单要素模

制定一个标准的基于的方案为目的，并使其能够支持简单地理要素集的存储、提取、查询、更新。简单要素类和要素类的区别是要素类可以是点、线、面，也可以是复合模式，维护拓扑数据，可以生成拓扑类，而简单要素类是不含拓扑数据的，容量比要素类要小得多。简单要素类型有点、线、面和几何体集合。

1）点实体

点数据是一个零维的几何对象，记录几何信息点坐标（和属性代码多点是复合点类，是多个点的集合，如果这些复合点中任意两个点都是不相等的，那么这个复合点则可被称为是简单的。

2）线实体

记录两个或一系列采样点的坐标，并加属性代码。线实体又可分为曲线类、线串类、环类，其中的曲线类是一维的几何对象，通常是由顺序连接的两个或者两个以上的点构成的。简单要素规范中只定义了曲线的一个子线串类。且规定该线串中任意两个顶点之间必须线性差值，若曲线没有两次穿过同一个点，就可认为该曲线为简单的。若首尾相连，那么该曲线是闭合的；如果闭合同时又简单，则形成环、如果不闭合，则拓扑封闭，且其边界是其起始点和终点。若曲线各个顶点线性插值，则形成线串。每个连续的顶点对构成一个线段。多个线的集合还可构成复合曲线，当构成复合曲线的所有线简单时，这个复合曲线为简单的。若由多个线串构成复合曲线，则构成了复合线串。

3）面实体

面是二维的几何对象，通过记录面实体的边界来表现。简单标准规范定义

的简单面是具有一个外边界和零个或者多个内部边界。多边形类（是唯一可被实例化的简单平面。关于多边形可以定义为：多边形是一个拓扑封闭的面，其边界由一组线性环构成，且边界上任意两个环不能相交，其只能以相切的方式存在，且多边形内部是连接的点集。

4）复合多边形

复合多边形的每个子面都是多边形，任何两个复合多边形的子对象内部是不能交叉的，而且任意两个复合多边形其子对象的边界也是不能相交的，只允许在有限个点上相接。复合多边形不能有分割线和毛刺，且是内部闭合的。

说当前已有很多开源框架对于简单要素进行了实现，例如Java Topology Suite(JTS)包。JTS封装了包含点

型[ ]如图3.2所示：图 3.2 OGC简单要素模型简单要素类和要素类的区别是要素类可以是点、线、面，也可以是复

2.地理要素集的存储、提取、查询、更新。简单要素模型[ ]如图3.2所示：图 3.2 OGC简单要素模型简单要素类和要素类的区别是要素类可以是点、线、面，也可以是复合模式，维护拓扑数据，可以生成拓扑类，而简单要素类是不含拓扑数据的，容量比要素类要小得多。简单要素类型有点、线、面和几何体集合。1、点实体点数据是一个零维的几何对象，记录几何信息点坐标（x,y）和属性代码[ ]；多点是复合点类（MultiPoint），是多个点的集合，如果这些复合点中任意两个点都是不相等的，那么这个复合点则可被称为是简单的。2、线实体记录两个或一系列采样点的坐标，并加属性代码。线实体又可分为曲线类、线串类、环类，其中的曲线类是一维的几何对象，通常是由顺序连接的两个或者两个以上的点构成的。OGC的简单要素规范中只定义了曲线的一个子线串类。且规定该线串中任意两个顶点之间必须线性差值，若曲线没有两次穿过同一个点，就可认为该曲线为简单的。若首尾相连，那么该曲线是闭合的；如果闭合同时又简单，则形成环；如果不闭合，则拓扑封闭，且其边界是其起始点和终点。若曲线各个顶点线性插值，则形成线串。每个连续的顶点对构成一个线段。图 3.3 线串示意多个线的集合还可构成复合曲线，当构成复合曲线的所有线简单时，这个复合曲线为简单的。若由多个线串构成复合曲线，则构成了复合线串[ ]。3、面实体面是二维的几何对象，通过记录面实体的边界来表现。OGC简单标准规范定义的简单面是具有一个外边界和零个或者多个内部边界。多边形类（Polygon）是唯一可被实例化的简单平面。关于多边形可以定义为：多边形是一个拓扑封闭的面，其边界由一组线性环构成，且边界上任意两个环不能相交，其只能以相切的方式存在，且多边形内部是连接的点集。4、多多边形复合多边形的每个子面都是多边形，任何两个复合多边形的子对象内部是不能交叉的，而且任意两个复合多边形其子对象的边界也是不能相交的，只允许在有限个点上相接。复合多边形不能有分割线和毛刺，且是内部闭合的。可以通过图3.4有个更形象的认识：图 3.4 复合多边形对象示意3.2 常见空间索引概述空间索引一般包

基于Redis的矢量数据组织研究 张景云 -《南京师范大学硕士论文》- 2013-04-18（是否引证：是）

1.本文遵循OGC国际标准化组织制定的标准简单要素规范（Simple FeaturesSpeciifcation)进行相应矢量数据模型和结构的设计，简单要素规范以制定一个标准旳基于ODBCAW的SQL方案为目的，并使其能够支持简单地理要素集的存储、提取、查询、更新。简单要素模型【67】如图3.2所示：几何体 空间参考系统r~“?"" 'i ：i ^ |TS]] 11线|丨|

2.|线』形^环 |丨 U|多多边形|| ‘线串—^ 11 ^ 丨个图3.2 OGC简单要素模型简单要素类和要素类的区别是要素类可以是点、线、面，也可以是复合模式，维22第3章传统矢量数据组织与索引 护拓扑数据，可以生成拓扑类，而简单要素类是不含拓扑数据的，容量比

3.素类和要素类的区别是要素类可以是点、线、面，也可以是复合模式，维22第3章传统矢量数据组织与索引 护拓扑数据，可以生成拓扑类，而简单要素类是不含拓

		扑数据的，容量比要素类要小得多。简单要素类型有点、线、面和几何体集合。1、点实体点数据是一个零维的几何对象，记录几何信息点坐标（x,y)和属性代码[6S];多点是复合点类（Multipoint)，是多个点的集合，如果这些复合点中任意两个点都是不相等的，那么这个复合点则可被称为是简单的。2、线实体记录两个或一系列采样点的坐标，并加属性代码。线实体又可分为曲线类、线串类、环类，其中的曲线类是一维的几何对象，通常是由顺序连接的两个或者两个以上的点构成的。OGC的简单要素规范中只定义了曲线的一个子线串类。且规定该线串中任意两个顶点之间必须线性差值，若曲线没有两次穿过同一个点，就可认为该曲线为简单的。若首尾相连，那么该曲线是闭合的；如果闭合同时又简单，则形成环、如果不闭合，则拓扑封闭，且其边界是其起始点和终点。若曲线各个顶点线性插值，则形成线串。每个连续的顶点对构成一个线段。?^irt 入end^ s*tart / X)\?z交t衿rt (a>簡单线（b) ?简 4.t衿rt (a>簡单线（b) ?简单线（c>闭合简单线（d)闭合非简单线图3.3线串7K意多个线的集合还可构成复合曲线，当构成复合曲线的所有线简单时，这个复合曲线为简单的。若由多个线串构成复合曲线，则构成了复合线串[69]。3、面实体面是二维的几何对象，通过记录面实体的边界来表现。OGC简单标准规范定义的简单面是具有一个外边界和零个或者多个内部边界。多边形类（Polygon)是唯一可被实例化的简单平面。关于多边形可以定义为：多边形是一个拓扑封闭的面，其边界由一组线性环构成，且边界上任意两个环不能相交，其只能以相切的方式存在，且多边形内部是连接的点集。23第3章传统矢量数据组织与索引 4、多多边形复合多边形的每个子面都是多边形，任何两个复合多边形的子 5.只能以相切的方式存在，且多边形内部是连接的点集。23第3章传统矢量数据组织与索引 4、多多边形复合多边形的每个子面都是多边形，任何两个复合多边形的子对象内部是不能交叉的，而且任意两个复合多边形其子对象的边界也是不能相交的，只允许在有限个点上相接。复合多边形不能有分割线和毛刺，且是内部闭合的。可以通过图3.4有个更形象的认识：0>:肩漏(4)一个多边形构成（b)三个多边形构成（c)两个多边形构成
2	此处有 73 字相似 ，需要将矢量对象的所有信息转化为字节流来存储，这个过程称之为“序列化”。反过来，将字节流转化为矢量对象称之为“反序列化” 。Avro是Hadoop中的一个子项目，也是Apache中一个独立的项目，是一个数据序列化的系统，可以将数据结构或对象转化成便于存储或传输的格式。 它由Doug Cutting在2009年牵头开发，目的是用来支持数据密集型的应用，适合于远程或本地大规模数据的存储和交换	Hadoop 生态系统 - 《网络（http://www.aiweibang）》- （是否引证：否）
		1.1、Apache Zookeeper 2、Google Chubby 2.6.3 Apache Avro Apache Avro 是 Hadoop 中的一个子项目，也是 Apache 中的一个独立的项目，Avro 是一个基于二进制数据传输高性能的中间件。在 Hadoop 的其它项目中，例如 HBase，Hive 的 Client 端与服 2.oop 的其它项目中，例如 HBase，Hive 的 Client 端与服务端的数据传输也采用了这个工具。Avro 是一个数据序列化的系统，它可以将数据结构或对象转化成便于存储或传输的格式。Avro 设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。拥有一下特点： 丰富的数据结构类型
		Spark SQL下Parquet内幕深度解密 - 囧芝麻的博客 - CSDN博客 - 《网络（http://blog.csdn.net）》- （是否引证：否）
		1.处理框架 =>1.数据本身，2.数据的元数据，3.引擎 Parquet表现上是树状数据结构，内部有元数据的 Table， Avro是Hadoop中的一个子项目，也是

Apache中一个独立的项目，Avro是一个基于二进制数据传输高性能的中间件。在hadoop的其他项目中例如Hbase(Ref)和Hive(Ref)的Client

2.的其他项目中例如Hbase(Ref)和Hive(Ref)的Client端与服务端的数据传输也采用了这个工具。Avro是一个数据序列化的系统。Avro 可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。 完成数据存储本身 对象模型可以简单理

021_GS132147C_姚飞 姚飞 -《学术论文联合比对库》- 2016-05-04（是否引证：否）

1.型的结果数据定时倒入到oracle数据库中。下图是模型软件环境的层次结构图：图 19 软件环境的层次结构图图中的Avro是HADOOP中的一个子项目，也是Apache中一个独立的项目，Avro是一个基于二进制数据传输高性能的中间件。在HADOOP的其他项目中例如HBase(Ref)和HIVE(Ref)的Client

2.的其他项目中例如HBase(Ref)和HIVE(Ref)的Client端与服务端的数据传输也采用了这个工具。Avro是一个数据序列化的系统。Avro 可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。Avro有很多特点：丰富的数据结构类型

SA15225048_罗鹏_1 罗鹏 -《学术论文联合比对库》- 2017-09-19（是否引证：否）

1.action表示用户行为的编码（由0~1等表示点击和搜索）。经过ETL后的数据，会以avro格式保存在HDFS上Avro是Hadoop中的一个子项目，也是Apache中一个独立的项目，Avro是一个基于二进制数据传输高性能的中间件。在Hadoop的其他项目中例如HBase(Ref)和Hive(Ref)的Client

2.的其他项目中例如HBase(Ref)和Hive(Ref)的Client端与服务端的数据传输也采用了这个工具。Avro是一个数据序列化的系统。Avro可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。数据经过ETL之后，组织形式如下：

智能电视个性化推荐系统的设计与实现 罗鹏 -《学术论文联合比对库》- 2017-09-19（是否引证：否）

1.action表示用户行为的编码（由0~1等表示点击和搜索）。经过ETL后的数据，会以avro格式保存在HDFS上Avro是Hadoop中的一个子项目，也是Apache中一个独立的项目，Avro是一个基于二进制数据传输高性能的中间件。在Hadoop的其他项目中例如HBase(Ref)和Hive(Ref)的Client

2.的其他项目中例如HBase(Ref)和Hive(Ref)的Client端与服务端的数据传输也采用了这个工具。Avro是一个数据序列化的系统。Avro可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。数据经过ETL之后，组织形式如下：

幼教校园管理系统的设计与实现 -《学术论文联合比对库》- 2017-10-11（是否引证：否）

1.MessageDecoder类。代码示意如上图：定义好数据

		的解码类型以后，还要定义数据存储格式,这里使用Avro。Avro 是Hadoop中的一个子项目，是一个数据序列化系统[34]。Avro主要用在数据密集型应用中，适合于远程或本地大规模数据的存储和交换，可以将数据结构或对象转化成便于传输或存储的格式
		1401220045_李长亮_幼教校园管理系统的设计与实现_荆琦_552e0567efbcedc80b87d7d631eeb0e9_20171011231145 李长亮 -《学术论文联合比对库》- 2017-10-13（是否引证：否）
		1.essageDecoder 类。代码示意如上图：定义好数据的解码类型以后，还要定义数据存储格式,这里使用Avro。Avro 是Hadoop 中的一个子项目，是一个数据序列化系统[34]。Avro 主要用在数据密集型应用中，适合于远程或本地大规模数据的存储和交换，可以将数据结构或对象转化成便于传输或存储
		徐迎晓_13222010150_陈钦彦_基于iOS的数据收集客户端的设计与实现 徐迎晓 -《学术论文联合比对库》- 2016-03-26（是否引证：否）
		1.那些和Session定义相关的数据，必须在第一时间保存，严格确保其完整性。2.3 Apache Avro[13]Avro是Hadoop中的一个子项目，也是Apache中的一个独立项目。Avro是一个基于二进制数据传输的高性能中间件[14]。在Hadoop的其他项目中，例如HBase和Hive的客户端与服务器端之间
		陈晓佳_143520085211005_面向个性化电台的实时大数据分析系统设计与实现 陈晓佳 -《学术论文联合比对库》- 2016-04-18（是否引证：否）
		1.k的目标是传输到另外的Agent，所以采用Avro类型的Sink。Avro是一个基于二进制数据传输的高性能中间件，也称作是一个数据序列化的系统，可以将数据结构或对象转化成便于存储或传输的格式。相应的，接收什么类型的数据，Agent就需要配置什么类型的Source，所以，Agent2和Agent3采用Avro类型
		030121221142 -《学术论文联合比对库》- 2014-01-02（是否引证：否）
		1.p 实现各种机器学习和大规模数据挖掘算法库。（7）Avro：它是一个基于二进制数据传输高性能的中间件和数据序列化的系统。它可以将数据结构或对象转化成便于存储或传输的格式，适合于远程或本地大规模数据的存储和交换。（8）对于MapReduce和HDFS，本文作者将在接下来的章节中
		云计算环境下资源调度策略的研究与实现 李振双 -《北京邮电大学硕士论文》- 2014-01-06（是否引证：否）
		1.各种机器学习和大规模数据挖掘算法库。(7) Avro[2a]:它是一个基于二进制数据传输高性能的中间件和数据序列化的系统。它可以将数据结构或对象转化成便于存储或传输的格式，适合于远程或本地大规模数据的存储和交换。对于编程模型MapReduce和分布式文件系统HDFS，本文作者将在
		李振双_2011140567_云计算环境下资源调度策略的研究与实现 李振双 -《学术论文联合比对库》- 2014-01-15（是否引证：否）
		1.p实现各种机器学习和大规模数据挖掘算法库。(7) Avro[20]：它是一个基于二进制数据传输高性能的中间件和数据序列化的系统。它可以将数据结构或对象转化成便于存储或传输的格式，适合于远程或本地大规模数

		据的存储和交换。对于编程模型MapReduce和分布式文件系统HDFS，本文作者将在接下来
3	此处有 140 字相似 据序列化的系统，可以将数据结构或对象转化成便于存储或传输的格式。它由Doug Cutting在2009年牵头开发，目的是 用来支持数据密集型的应用，适合于远程或本地大规模数据的存储和交换。Avro是一个基于二进制数据传输高性能的中间件，是一种与编程语言无关的序列化系统，它提供了丰富的数据结构类型、快速可压缩的二进制数据格式、存储持久性数据的文件容器、远程过程调用（RPC）以及简单的动态语言结合功能。 当前除了Apache Avro系统外，Thrift, Protocol Buffers等系统也提供类似功能，但与其他系统	Java实现数据序列化工具Avro的例子 - 专注于大数据技术研究和应用 - 博客频道 - CSDN.NET - 《网络（http://blog.csdn.net）》- （是否引证：否） 1.1、Avro简介 Avro是一个数据序列化的系统。 它可以提供： 1)丰富的数据结构类型 2)快速可压缩的二进制数据形式 3)存储持久数据的文件容器 4)远程过程调用RPC 5)简单的动态语言结合功能，Avro和动态语言结合后，读写数据文件和使用RPC协议都不需要生成代码，而代码生成作为一种可选的优化只得在静态类型语言中 Avro序列化操作（1）：环境搭建和Schema处理 - hua245942641的专栏 - 博客频道 - CSDN.NET - 《网络（http://blog.csdn.net）》- （是否引证：否） 1.环境： Intellij 15.0.3 Maven avro 1.8.0 Avro是一个数据序列化系统。 它提供以下： 1 丰富的数据结构类型 2 快速可压缩的二进制数据形式 3 存储持久数据的文件容器 4 远程过程调用RPC 5 简单的动态语言结合功能，Avro和动态语言结合后，读写数据文件和使用RPC协议都不需要生成代码，而代码生成作为一种可选的优化只值得在静态类型语言 Hadoop 生态系统 - 《网络（http://www.aiweibang）》- （是否引证：否） 1.采用了这个工具。Avro 是一个数据序列化的系统，它可以将数据结构或对象转化成便于存储或传输的格式。Avro 设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。拥有一下特点： 丰富的数据结构类型 快速可压缩的二进制数据形式，对数据二进制序列化后可以节约数据存储空间和网络传输带宽 2.拥有一下特点： 丰富的数据结构类型 快速可压缩的二进制数据形式，对数据二进制序列化后可以节约数据存储空间和网络传输带宽 存储持久数据的文件容器 可以实现远程过程调用 RPC 简单的动态语言结合功能 相关链接地址，如下所示： 1、Apache Avro 另外，还有 Apache Curator，Twitter Elep Google Protocol Buffer持久化框架分析 _rotosix - 《网络（http://blog.sina.com）》- （是否引证：否） 1.入文件，编译器生成代码用来方便地生成RPC客户端和服务器通信的无缝跨编程语言。 1.3 Apache avro Avro是一个数据序列化的系统，它可以提供丰富的数据结构类型，快速可压缩的二进制数据形式，存储持久数据的文件容器，远程过程调用RPC。 Avro依赖于模式(Schema)。Avro数据的读写操作是很频繁的，而这些操作都需要使用模式，这样就减少写入每个数据 AVRO - u013061459的博客 - CSDN博客 - 《网络（http://blog.csdn.net）》- （是否引证：否） 1.Avro是一个数据序列化的系统。 功 能 丰富的数据结构类型 简介 它可以提供： 1 丰富的数据结构类型 2 快速可压缩的二进制数据形式 3 存储持久数据的文件容器 4 远程过程调用RPC 5 简单的动态语言结合功能，Avro和动态语言结合后，读写数据文件和使用RPC协议都不需要生成代码，而代码生成作为一种可选的优化只得在静态类型语言中

MapReduce容错机制的改进研究 吴慧城 -《湖南大学硕士论文》- 2014-05-15（是否引证：否）

1.式来对数据进行增删改查等。在生态系统中除了核心组件 HDFS 和 Map Reduce 外还包括：Avro：Avro 是一个基于二进制数据传输高性能的中间件，是一个数据序列化的系统。在 Hadoop 的其他项目中例如 HBase 和 Hive 的 Client 端与服务端的数据传输采用了这个工具

陈晓佳_143520085211005_面向个性化电台的实时大数据分析系统设计与实现 陈晓佳 -《学术论文联合比对库》- 2016-04-18（是否引证：否）

1.的是syslogtcp源的方式。而Agent1的Sink的目标是传输到另外的Agent，所以采用Avro类型的Sink。Avro是一个基于二进制数据传输的高性能中间件，也称作是一个数据序列化的系统，可以将数据结构或对象转化成便于存储或传输的格式。相应的，接收什么类型的数据，Agent就需要配置什么类型的Source，

基于云计算技术的化合物相似性分析系统 李杰辉 -《复旦大学硕士论文》- 2012-04-25（是否引证：否）

1.oop项目的核心，其它Hadoop子项目都是在Hadoop Common的基础上发展的。(2) Avro; Avro是一个基于二进制数据传输高性能中间件，在Hadoop的其他项目例如HBase和Hive的客户端与服务端的数据传输也采用了这个工具，它可以将数据进行序列化，适用于远

徐迎晓_13222010150_陈钦彦_基于iOS的数据收集客户端的设计与实现 徐迎晓 -《学术论文联合比对库》- 2016-03-26（是否引证：否）

1.的客户端与服务器端之间的数据传输也采用了这个工具。Avro可以将数据结构或对象转化成便于存储或传输的格式，它在设计之初就用来支持数据密集型的应用，适合于远程或本地大规模数据的存储和交换[15]。Avro提供了丰富的数据结构类型和快速可压缩的二进制数据形式，使得数据经过二进制序列化后能极大的节约数据存储空间和网络带宽；

021_GS132147C_姚飞 姚飞 -《学术论文联合比对库》- 2016-05-04（是否引证：否）

1.了这个工具。Avro是一个数据序列化的系统。Avro 可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。Avro有很多特点：丰富的数据结构类型；快速可压缩的二进制数据形式，对数据二进制序列化后可以节约数据存储空间和网络传输带宽；

Spark SQL下Parquet内幕深度解密 - 囧芝麻的博客 - CSDN博客 -《网络（http://blog.csdn.net）》-（是否引证：否）

1.了这个工具。Avro是一个数据序列化的系统。Avro 可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。 完成数据存储本身对象模型可以简单理解为内存中的数据表示，Avro,Thrift,Protocol Buffers,h

张琦 张琦 -《学术论文联合比对库》- 2016-01-09（是否引证：否）

1.ro：Avro是一个基于二进制数据传输高性能的中间件，可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远

		程或本地大规模数据的存储和交换。 MapReduce：MapReduce是一种编程模型，用于大规模数据集的并行运算。通过将数据处理逻辑定义成可分布式并行的 SA15225048_罗鹏_1 罗鹏 -《学术论文联合比对库》- 2017-09-19（是否引证：否） 1.用了这个工具。Avro是一个数据序列化的系统。Avro可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。数据经过ETL之后，组织形式如下：Uid Item-id action25aa3ww4210246769855e 智能电视个性化推荐系统的设计与实现 罗鹏 -《学术论文联合比对库》- 2017-09-19（是否引证：否） 1.用了这个工具。Avro是一个数据序列化的系统。Avro可以将数据结构或对象转化成便于存储或传输的格式。Avro设计之初就用来支持数据密集型应用，适合于远程或本地大规模数据的存储和交换。数据经过ETL之后，组织形式如下：Uid Item-id action25aa3ww4210246769855e
4	此处有 85 字相似 ro不需要生成代码。数据总是依赖于模式（schema），该模式允许在不生成代码、静态数据类型等情况下对数据进行完全处理。 Avro数据的读写操作非常频繁，而这些操作都需要使用模式，这样就减少了写入每个数据的开销，使得序列化过程快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用， 有助于构建通用数据处理系统和语言。 （2）无标记数据：由于在读取数据时存在模式，因此需要用数据编码的类型信息要少得多，	Google Protocol Buffer持久化框架分析_rotosix -《网络（http://blog.sina.com）》-（是否引证：否） 1.结构类型，快速可压缩的二进制数据形式，存储持久数据的文件容器，远程过程调用RPC。 Avro依赖于模式(Schema)。Avro数据的读写操作是很频繁的，而这些操作都需要使用模式，这样就减少写入每个数据资料的开销，使得序列化快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用。当Avro数据存储到文件中时，它的模式也随之存储，这样任何程序都可以对文件进行处理。如果需要以不同的模式读取数据，这也很 Java实现数据序列化工具Avro的例子 - 专注于大数据技术研究和应用 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》-（是否引证：否） 1.RPC协议都不需要生成代码，而代码生成作为一种可选的优化只得在静态类型语言中实现。 Avro依赖于模式(Schema)。Avro数据的读写操作是很频繁的，而这些操作都需要使用模式，这样就减少写入每个数据资料的开销，使得序列化快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用。 当Avro数据存储到文件中时，它的模式也随之存储，这样任何程序都可以对文件进行处理。如果需要以不同的模式读取数据，这也 Avro序列化操作（1）：环境搭建和Schema处理 - hua245942641的专栏 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》-（是否引证：否） 1.PC协议都不需要生成代码，而代码生成作为一种可选的优化只值得在静态类型语言中实现。 Avro依赖于模式(Schema)。Avro数据的读写操作是很频繁的，而这些操作都需要使用模式，这样就减少写入每个数据资料的开销，使得序列化快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用。 下面介绍如果使用avro进行序列化和反序列化的操作 前置条件：maven项目 1、在pom.xml中添加avro的依 AVRO - u013061459的博客 - CSDN博客 -《网络（http://blog.csdn.net）》-（是否引证：否）

		1.RPC协议都不需要生成代码，而代码生成作为一种可选的优化只得在静态类型语言中实现。 Avro依赖于模式(Schema)。Avro数据的读写操作是很频繁的，而这些操作都需要使用模式，这样就减少写入每个数据资料的开销，使得序列化快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用。 当Avro数据存储到文件中时，它的模式也随之存储，这样任何程序都可以对文件进行处理。如果需要以不同的模式读取数据，这也 徐迎晓_13222010150_陈钦彦_基于iOS的数据收集客户端的设计与实现 徐迎晓 -《学术论文联合比对库》- 2016-03-26（是否引证：否） 1.的实现，如C、C++、C#、Java。Avro的显著特征是它对于Schema的依赖，可以动态加载相关数据的Schema。Avro数据的读写操作很频繁，而这些操作使用的都是Schema，这样就减少写入每个数据文件的开销，使得序列化快速而又轻巧。这种数据及其Schema的描述也方便了动态脚本语言的使用。当Avro数据存储到文件中时，它的Schema也随之存储，这样任何程序就都可以对文件进行处理。 2.3.1Schema 幼教校园管理系统的设计与实现 -《学术论文联合比对库》- 2017-10-11（是否引证：否） 1.数据的存储和交换，可以将数据结构或对象转化成便于传输或存储的格式；Avro 的显著特征是：依赖于模式，动态加载相关数据的模式，这样就减少写入每个数据文件的开销，使得序列化快速而又轻巧[30]。Avro 模式是用广泛使用的 JSON 格式（一种轻量级的数据交换模式）定义的，这样与其他系统有很好的兼容性。每一条记录 1401220045_李长亮_幼教校园管理系统的设计与实现_荆琦_552e0567efbcedc80b87d7d631eeb0e9_20171011231145 李长亮 -《学术论文联合比对库》- 2017-10-13（是否引证：否） 1.数据的存储和交换，可以将数据结构或对象转化成便于传输或存储的格式；Avro 的显著特征是：依赖于模式，动态加载相关数据的模式，这样就减少写入每个数据文件的开销，使得序列化快速而又轻巧[30]。Avro 模式是用广泛使用的 JSON 格式（一种轻量级的数据交换模式）定义的，这样与其他系统有很好的兼容性。每一条
5	此处有 34 字相似 减少了写入每个数据的开销，使得序列化过程快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用，有助于构建通用 数据处理系统和语言。 （2）无标记数据：由于在读取数据时存在模式， 因此需要用数据编码的类型信息要少得多，从而导致较小的序列化大小，这有助于数据的压缩。 （3）不用手动分配字段ID：当模	Java实现数据序列化工具Avro的例子 - 专注于大数据技术研究和应用 - 博客频道 - CSDN.NET -《网络（http://blog.csdn.net）》-（是否引证：否） 1.型：Avro并不需要生成代码，模式和数据存放在一起，而模式使得整个数据的处理过程并不生成代码、静态数据类型等等。这方便了数据处理系统和语言的构造。 2)未标记的数据：由于读取数据的时候模式是已知的，那么需要和数据一起编码的类型信息就很少了，这样序列化的规模也就小了。 3)不需要用户指定字段号：即使模式改变，处理数据时 AVRO - u013061459的博客 - CSDN博客 -《网络（http://blog.csdn.net）》-（是否引证：否） 1.型：Avro并不需要生成代码，模式和数据存放在一起，而模式使得整个数据的处理过程并不生成代码、静态数据类型等等。这方便了数据处理系统和语言的构造。 2 未标记的数据：由于读取数据的时候模式是已知的，那么需要和数据一起编码的类型信息就很少了，这样

		序列化的规模也就小了。 3 不需要用户指定字段号：即使模式改变，处理数据时

指　标
疑似剽窃文字表述

1. 。Avro是Hadoop中的一个子项目，也是Apache中一个独立的项目，是一个数据序列化的系统，可以将数据结构或对象转化成便于存储或传输的格式。
2. Avro数据的读写操作非常频繁，而这些操作都需要使用模式，这样就减少了写入每个数据的开销，使得序列化过程快速而又轻巧。这种数据及其模式的自我描述方便于动态脚本语言的使用，

5. 第4章时空矢量对象的查询算法设计

总字数：8801

相似文献列表

去除本人已发表文献复制比：5.1%(452)　　文字复制比：5.1%(452)　　疑似剽窃观点：(0)

1	面向列存储模式的时空对象查询处理技术研究 史宗麟(导师：汤大权) -《国防科学技术大学硕士论文》- 2014-11-01	5.1%（452） 是否引证：否

原文内容	相似内容来源
1 此处有 35 字相似 e为5，图（6）中递归参数为6，由图可知，最大递归次数越大，近似的更加精确，所生成的Range更多。 图4-2 多边形 范围查询示意图 4.2 时空范围查询算法 时空范围查询可以定义为在 给定的时间范和空间区域（通常矩形，也可为不规则多边形）的条件下，从数据库中取出所有满足查询条件的时空矢量对象。对于点矢量	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.究基于“无表”结构面向列存储模式时空索引的时空范围查询和时空最邻近查询算法设计。3.4.1 时空范围查询算法设计时空范围查询可以描述为在特定时间段内或特定时刻，查询所有在特定的空间区域内的时空对象数据。其基本思想为：根据查询条件，将查询空间
2 此处有 47 字相似 angeEnd }，其中rangeStart，rangeEnd表示该段Range的起始、终止空间填充曲线编码值。 （2 ）connectHTable(tablename)表示连接列数据库中名为tablename的表 ；用于从HBase中查询结果。 （3）将查询时间范围Qtime按照时间粒度timeCoarseBin离散化，得到一个递	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.rt 值，也是最小值；Hexit 表示该段 Segment 的终止 Hilbert 值，也是最大值；（5）connect HTable(tablename)表示连接列数据库中名为 tablename 的表；（6）rowkey Range 表示 rowkey 的范围，由起始 rowkey 和终止 rowkey 构成；（
3 此处有 67 字相似 nge = (startrowkey , endrowkey);Scan = scan(rowkeyRange) ResultScanner=table.get Scanner(Scan);for each Result in ResultScan ner do:if Result in Qrange and Result in Qtime:Resultset.add	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.lename);（16）开始扫描 Scan=scan(rowkey Range);（17）获得扫描结果 Result Scanner=table.get Scanner(Scan);（18）foreach Result in Result Scanner（19） if Result in Qrange（20） 向查询结果列表中增加
4 此处有 47 字相似 部长度为4的倍数，其中rangeStart，rangeEnd表示该段QudrantRange的起始、终止编码值。	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.rt 值，也是最小值；Hexit 表示该段 Segment 的终止

	（2）connectHTable(tablename)表示连接列数据库中名为tablename的表；用于从HBase中查询结果。 （3）将查询时间Qtime按照时间粒度timeCoarseBin离散化，得到一个时间结	Hilbert 值，也是最大值；（5）connect HTable(tablename)表示连接列数据库中名为 tablename 的表；（6）rowkey Range 表示 rowkey 的范围，由起始 rowkey 和终止 rowkey 构成；（
5	此处有 67 字相似 nge = (startrowkey , endrowkey);Scan = scan(rowkeyRange) ResultScanner=table.get Scanner(Scan);for each Result in ResultScan ner do:if Result in Qrange and Result in Qtime:Resultset.add	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.lename);（16）开始扫描 Scan=scan(rowkey Range);（17）获得扫描结果 Result Scanner=table.get Scanner(Scan);（18）foreach Result in Result Scanner（19）if Result in Qrange（20）向查询结果列表中增加
6	此处有 122 字相似 算法 4.3.1 基本策略 时空最邻近查询可以描述为给定坐标和参数k，返回查询时间范围内的k个对象。其设计思路如下： 首先根据k值估算出一个以给定坐标为中心的网格的空间范围，然后在执行时空范围查询。如果返回结果超过k，则计算与给定坐标距离最近的k个对象，得到查询结果；如果返回结果不足k，则进一步扩大其邻域进行迭代查询，直到查询到的数据量达到为止。为了更好地描述 算法，本文首先定义三种距离。如图三对于空间点p(x1，x2)，q(y1,y2)，d1距离为两点之间的欧式距离，d2距离定	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.述为给定坐标和用户参数 K，返回时间段内或者某时刻距离坐标最近的 K 个对象。时空最邻近查询基本思想为：首先根据 K 值估算出一个以给定坐标为中心的正方形空间范围，然后在执行时空范围查询。如果返回结果超过 K，则计算与给定坐标距离最近的 K 个对象，得到查询结果；如果返回结果不足 K，则进一步扩大正方形的边长，迭代查询，直到查询到的数据量达到 K 为止。表 3.3 描述了此最邻近查询的算法。表 3.3 基于“无表”结构面向列存储模式的时空索引最邻近查询算法Alg
7	此处有 67 字相似 owkeyRange = cellToRange(curCell);Scan = scan(rowkeyRange) ResultScanner=table.get Scanner(Scan);for each Result in ResultScan ner do:Resultset.add(Result); //将计算结果添加至结果集end forif type is	面向列存储模式的时空对象查询处理技术研究_史宗麟 -《国防科学技术大学硕士论文》- 2014-11-01（是否引证：否） 1.lename);（16）开始扫描 Scan=scan(rowkey Range);（17）获得扫描结果 Result Scanner=table.get Scanner(Scan);（18）foreach Result in Result Scanner（19）if Result in Qrange（20）向查询结果列表中增加

指　标
疑似剽窃文字表述
1. 首先根据k值估算出一个以给定坐标为中心的网格的空间范围，然后在执行时空范围查询。如果返回结果超过k，则计算与给定坐标距离最近的k个对象，得到查询结果；如果返回结果不足k，则进一步扩大其邻域进行迭代查询，直到查询到的数据量达到为止。为了更好地描述

6. 第5章实验性能评价及结果分析　　总字数：2787

相似文献列表

去除本人已发表文献复制比：0%(0)　　文字复制比：0%(0)　　疑似剽窃观点：(0)

7. 第6章总结与展望　　总字数：2393

相似文献列表

去除本人已发表文献复制比：21.6%(518)　　文字复制比：21.6%(518)　　疑似剽窃观点：(0)

1	矢量大数据管理关键技术研究	9.2%（220）
	姚晓闯(导师：郧文聚;朱德海) -《中国农业大学博士论文》- 2017-05-01	是否引证：否
2	201321060644陈俊欣	9.0%（216）
	-《学术论文联合比对库》- 2016-03-21	是否引证：否
3	基于Hadoop的空间矢量数据的分布式存储与查询研究	8.7%（207）
	陈俊欣(导师：张凤荔) -《电子科技大学硕士论文》- 2016-03-18	是否引证：否
4	姚晓闯_B1311686_矢量大数据管理关键技术研究	7.9%（189）
	姚晓闯 -《学术论文联合比对库》- 2017-06-01	是否引证：否
5	面向异构资源集成的虚拟实验平台研究	2.5%（61）
	陈天赐(导师：王建新) -《中南大学硕士论文》- 2011-05-01	是否引证：否
6	基于网格的数据流聚类算法研究	1.8%（43）
	张丽(导师：姜保庆) -《河南大学硕士论文》- 2011-05-01	是否引证：否
7	SiC陶瓷/UHMWPE复合装甲弹道性能研究	1.4%（34）
	张友敏(导师：胡德安) -《湖南大学硕士论文》- 2018-04-20	是否引证：否

原文内容	相似内容来源
1 此处有 196 字相似 务器发出请求，得到周围八邻域的数据总数，而不获取数据。并通过总数与k-NN查询的k比较，决定是否进一步扩大网格大小。 6.2 展望 本文针对矢量大数据管理方面进行了较为全面的研巧工作，初步解决了大规模矢量数据的存储、索引、处理从及可视化等实际应用问题，并取得了一定的研巧成果，但由于作者的研究时间和精力有限，本文研究工作仍然存在着不足和需要改进的内容。目前，我认为还可以在以下几个方面继续开展相关研究，以推动大数据时代矢量数据管理基础理论和关键技术的快速发展。 1）对于线面对象，本文采取冗余备份的思想，但由于线对象一般是狭长的 形状，用至多4份的空间格网去覆盖线对象，仍会造成很多的空间浪费，需要进一步研究线对象的存储机	姚晓闯_B1311686_矢量大数据管理关键技术研究 姚晓闯 -《学术论文联合比对库》- 2017-06-01（是否引证：否） 1.他地图服务或系统的无缝集成，能够支撑大规模矢量数据地图瓦片的快速构建，解决了矢量大数据多尺度浏览查看的基本难题。研究展望本文针对矢量大数据管理方面进行了较为全面的研究工作，初步解决了大规模矢量数据的存储、索引、处理以及可视化等实际应用问题，并取得了一定的研究成果，但由于作者的研究时间和精力有限，本文研究工作仍然存在着不足和需要改进的内容。目前，我认为还可以在以下几个方面继续开展相关研究，以推动大数据时代矢量数据管理基础理论和关键技术的快速发展：（1）本文提出的矢量大数据云存储模型——GeoCSV数据模型基于NoSQL中最为常见的Key-Value存储模型进行扩展，尽管将面向对象 矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博士论文》- 2017-05-01（是否引证：否） 1.系统的无缝集成，能够支撑大规模矢量数据地图瓦片的快速构建，解决了矢量大数据多尺度浏览查看的基本难题。6.3研究展望本文针对矢量大数据管理方面进行了较为全面的研巧工作，初步解决了大规模矢量数据的存储、索引、处理从及可视化等实际应用问题，并取得了一定的研巧成果，但由于作者的研巧时间和精力有限，本文研巧工作仍然存在着不足和需要改进的内容。目前，我认为还可W在W下几个方面继续开展相关研巧，W推动大数据时代矢量数据管理基础理论和关键技术的快速发展：1) 本文提出的矢量大数据云存储模型一GeoCSV数据模型基于NoSQL中最为常见的Key-Value存储模型进斤扩 SiC陶瓷/UHMWPE复合装甲弹道性能研究 张友敏 -《湖南大学硕士论文》- 2018-04-20（是否引证：否） 1.实验、理论和数值模拟三个方面对 Si C 陶瓷/UHMWPE 复合装甲的抗弹机理进行了探究，得到了一定的结果；但是由于个人时间和能力有限，本文的研究工作及内容仍然存在很多不足之处和有待完善的地方，不能对其做更加深入的研究，后续还需要进一步加深开展工作。（1）本文主要侧重对陶瓷复合装甲的实验研究，不同几何
2 此处有 45 字相似 形状，用至多4份的空间格网去覆盖线对象，仍会造成很多的空间浪费，需要进一步研究线对象的存储机制。	基于Hadoop的空间矢量数据的分布式存储与查询研究 陈俊欣 -《电子科技大学硕士论文》- 2016-03-18（是否引证：否）

	提高查询效率。 2） 本文只完成了两类查询，即范围查询即k-NN查询。更多空间关系的查 询包括空间的连接查询， 以及复杂的拓扑关系判断，还需要进一步探索。 3）整个HBase在海量空间数据库的管理方面并没有传统关系数据库那么成	1.加智能一些，但是这涉及很复杂的计算逻辑，因此构建查询热点的地理散列的精度问题还需要进一步研究。（3）本文只完成了两类查询，更多空间关系的查询包括空间的连接查询（两个空间对象集合应用谓词（覆盖、距离、邻接等））以及复杂的拓扑关系判断，还需要进一步探索。（4 201321060644陈俊欣 -《学术论文联合比对库》- 2016-03-21（是否引证：否） 1.选择前缀方框时更加智能一些，但是这涉及很复杂的计算逻辑，因此构建查询热点的地理散列的精度问题还需要进一步研究。（3）本文只完成了两类查询，更多空间关系的查询包括空间的连接查询（两个空间对象集合应用谓词（覆盖、距离、邻接等））以及复杂的拓扑关系判断，还需要进一步探索。（4）整个HBase在海
3	此处有 172 字相似 机制。提高查询效率。 2）本文只完成了两类查询，即范围查询即k-NN查询。更多空间关系的查 询包括空间的连接查询， 以及复杂的拓扑关系判断，还需要进一步探索。 3）整个HBase在海量空间数据库的管理方面并没有传统关系数据库那么成 熟，包括对于数据分片的管理，对于操作的实时监控，以及用户的操作的权限管理，数据的复制与备份等等。探究HBase的空间数据处理引擎，不断丰富和提升其管理功能，为用户提供准确、快速的空间数据支持也是今后研究的一个方向。 致谢 在论文完成之际，谨向我给予帮助的老师、同学、朋友表示衷心的感谢。 首先，我要感谢范老师，虽然范老师长期在外出差	201321060644陈俊欣 -《学术论文联合比对库》- 2016-03-21（是否引证：否） 1.。（3）本文只完成了两类查询，更多空间关系的查询包括空间的连接查询（两个空间对象集合应用谓词（覆盖、距离、邻接等））以及复杂的拓扑关系判断，还需要进一步探索。（4）整个HBase在海量空间数据库的管理方面并没有传统关系数据库那么成熟，包括对于数据分片的管理，对于操作的实时监控，以及用户的操作的权限管理，数据的复制与备份等等。探究HBase的空间数据处理引擎HBaseGIS，不断丰富和提升HBaseGIS的管理功能，为用户提供准确、快速的空间数据支持也是今后研究的一个方向。致谢三年的研究生生活转瞬即逝，在很多老师和同学的帮助和关怀下，我愉快而充实地度过了自己的研究生生活。首先，在本文 基于Hadoop的空间矢量数据的分布式存储与查询研究 陈俊欣 -《电子科技大学硕士论文》- 2016-03-18（是否引证：否） 1.3）本文只完成了两类查询，更多空间关系的查询包括空间的连接查询（两个空间对象集合应用谓词（覆盖、距离、邻接等））以及复杂的拓扑关系判断，还需要进一步探索。（4）整个 HBase 在海量空间数据库的管理方面并没有传统关系数据库那么成熟，包括对于数据分片的管理，对于操作的实时监控，以及用户的操作的权限管理，数据的复制与备份等等。探究 HBase 的空间数据处理引擎 HBase GIS，不断丰富和提升 HBase GIS 的管理功能，为用户提供准确、快速的空间数据支持也是今后研 2.复制与备份等等。探究 HBase 的空间数据处理引擎 HBase GIS，不断丰富和提升 HBase GIS 的管理功能，为用户提供准确、快速的空间数据支持也是今后研究的一个方向。 致谢66致 谢三年的研究生生活转瞬即逝，在很多老师和同学的帮助和关怀下，我愉快而
4	此处有 44 字相似 空间数据处理引擎，不断丰富和提升其管理功能，为用户提供准确、快速的空间数据支持也是今后研究的一个方向。 致谢 在	基于网格的数据流聚类算法研究 张丽 -《河南大学硕士论文》- 2011-05-01（是否引证：否） 1. 43 致 谢 三年的研究生生活很快就要结束，这中间得到了老师和同学的帮助和关怀。在此论文完成之际，向给予我帮助的老师、同学、朋友表示衷心的感谢！ 首先衷心感谢我的导师姜保庆教授。在我攻读硕士期间，不论是学习和科研活动中，还是在日常生活中，姜老师都给予了我无微不至的关怀和照顾。 矢量大数据管理关键技术研究 姚晓闯 -《中国农业大学博

	论文完成之际，谨向我给予帮助的老师、同学、朋友表示衷心的感谢。 首先，我要感谢范老师， 虽然范老师长期在外出差，但范老师仍然给我提供了无微不至的关怀，对我的学习生活悉心指导。同时我要感谢关老师，在我论文的开题	士论文》- 2017-05-01（是否引证：否） 1.时光在巧，岁月如梭。转眼之间，四年的博±生活已经进入尾声，留恋之情悠然而生。值此论文搁笔之际，向所有曾经关屯、、帮助和支持我的老师、同学W及家人和朋友表示最诚挈的谢意！首先衷私感谢我的导师郎文聚教授。有幸成为郎老师的弟子，我感到十分荣幸。四年来，在整个研究生学习和生活上，无不得到那老师的悉屯、
5	此处有 61 字相似 谢导师吴教授。回顾在硕士三年的学习中，吴教授不仅提供了优越的学习环境，而且其严谨的治学态度和对我的严格要求促进我进步。两 年来，我在学习和科研上所取得的每一点成绩、每一点进步无不浸透着导师的心血。在此谨向尊敬的导师致以最诚挚的谢意。 还要感谢 在我论文写作过程中给予指导意见的关洪礼同学以及谌诞楠同学。在我论文开题与实验设计、论文写作当中，2位同门给了我莫大的帮助	面向异构资源集成的虚拟实验平台研究 陈天赐 -《中南大学硕士论文》- 2011-05-01（是否引证：否） 1.的开拓精神和在事业上锐意进取的执着精神都令我永生难忘。论文的选题、研究和撰写都倾注了王老师的无私关爱。几年来，我在学习和科研上所取得的每一点成绩、每一点进步无不浸透着导师的心血。在此谨向尊敬的导师致以最诚挚的谢意。 其次，感谢课题组的盛羽老师。在整个研究生的学习过程中，盛老师在课题研究中给予了我很多提示性的建议，给予了我莫大的启发，感谢

指　标
疑似剽窃观点
1. 目前，我认为还可以在以下几个方面继续开展相关研究，以推动大数据时代矢量数据管理基础理论和关键技术的快速发展。
疑似剽窃文字表述
1. 6.2 展望 本文针对矢量大数据管理方面进行了较为全面的研巧工作，初步解决了大规模矢量数据的存储、索引、处理从及可视化等实际应用问题，并取得了一定的研巧成果，但由于作者的研究时间和精力有限，本文研究工作仍然存在着不足和需要改进的内容。 2. 本文只完成了两类查询，即范围查询即k-NN查询。更多空间关系的查 询包括空间的连接查询，以及复杂的拓扑关系判断，还需要进一步探索。 3）整个HBase在海量空间数据库的管理方面并没有传统关系数据库那么成 熟，包括对于数据分片的管理，对于操作的实时监控，以及用户的操作的权限管理，数据的复制与备份等等。探究HBase的空间数据处理引擎，不断丰富和提升其管理功能，为用户提供准确、快速的空间数据支持也是今后研究的一个方向。 3. 论文完成之际，谨向我给予帮助的老师、同学、朋友表示衷心的感谢。 首先，我要感谢范老师， 4. 年来，我在学习和科研上所取得的每一点成绩、每一点进步无不浸透着导师的心血。在此谨向尊敬的导师致以最诚挚的谢意。 还要感谢

说明：1.总文字复制比：被检测论文总重合字数在总字数中所占的比例

2.去除引用文献复制比：去除系统识别为引用的文献后，计算出来的重合字数在总字数中所占的比例

3.去除本人已发表文献复制比：去除作者本人已发表文献后，计算出来的重合字数在总字数中所占的比例

4.单篇最大文字复制比：被检测文献与所有相似文献比对后，重合字数占总字数的比例最大的那一篇文献的文字复制比

5.指标是由系统根据《学术论文不端行为的界定标准》自动生成的

6.红色文字表示文字复制部分;绿色文字表示引用部分;棕灰色文字表示作者本人已发表文献部分

7.本报告单仅对您所选择比对资源范围内检测结果负责

amlc@cnki.net