DOI:10.16515/j.cnki.32-1722/n.2019.02.010

基于维度建模的岗位画像系统

窦如林,虞李凯,张 凯

(金陵科技学院软件工程学院,江苏 南京 211169)

摘 要:截至2018年,全球网民总数已经达36亿,超过全球总人口的50%。互联网使用率不断增长的同时,产生数据的能力已经远远超过我们处理数据的能力。为了从互联网中获取更多的商业价值,介绍了基于维度建模的大数据分析技术,并利用爬虫技术获取网上的岗位信息,再利用大数据分析技术,形成岗位画像。

关键词:大数据;维度建模;画像;爬虫

中图分类号:TP18;TP311 **文献标识码**:A **文章编号**:1672-755X(2019)02-0044-05

Research on Application Scenarios of Dimensional Modeling on Post Portraits

DOU Ru-lin, YU Li-kai, ZHANG Kai

(Jinling Institute of Technology, Nanjing 211169, China)

Abstract: As of 2018, the total number of Internet users has reached 3.6 billion, more than 50% of the global population. While Internet usage continues to grow, the ability to generate data has far exceeded our ability to process data. In order to obtain more commercial value from the Internet, this paper introduces big data analysis technology based on dimensional modeling, and uses crawler technology to obtain online post information, and then use big data analysis technology to form post portraits.

Key words: big data; dimensional modeling; portrait; crawler

大数据的发展正在全面展开,各种新技术纷纷涌现,不仅使人们获取数据变得较为容易,还可以让人们根据不同的应用需求,采用对应的数据分析方法,为企业创造商业价值。针对大数据的应用,本文介绍设计的一套大数据处理系统,可以爬取目前热门的在线招聘网站,获取企业人才招聘的需求条件,利用大数据技术等前沿技术进行智能分析,让广大高校在校学生了解社会各类岗位对人才技能的需求。

1 大数据处理的现状及对岗位画像的影响

基于大数据的岗位画像系统是在爬取了大量的求职网站的招聘信息后,通过运用机器学习算法,进行大量的迭代计算后得出关于一个岗位信息的精准描述。旨在帮助企业明确招聘需求,确定精准的人才战略和人才培养体系,提升招人效率。然而由于大量迭代计算的存在,以及画像系统后期需要提供交互式查询的功能,在画像系统完成之后,交互式查询的时间延迟仍然很高。

目前Apache Hadoop已经成为大数据领域基础且重要的技术,其诞生到现在已有10余年。最初用于简单的分布式存储底层实现,然后使用Map-Reduce编程模型来进行分布式计算。而如今其在交互式

收稿日期:2019-04-21

基金项目:江苏省高等学校大学生创新创业训练计划项目(201813573047X);江苏现代教育技术研究课题(2017-R-54608)

作者简介:窦如林(1979—),男,江苏南京人,高级实验师,硕士,主要从事计算机网络、大数据研究。

分析、多维分析、人工智能甚至机器学习方面取得了很大进展。但是传统架构仅支持垂直扩展，通过扩展单机处理能力，来增加系统的算力，但这种方式很快就会达到极限[1]。

此外，分钟级查询响应离交互式分析的用户需求相差甚远。通常情况下，系统在使用过程中，当用户获取分析结果后，会根据情况更改查询参数，重新进行数据分析。类似的多轮参数调整，分析结果需要数小时甚至几天才能完成，效率较为低下。这是因为大规模并行处理和列存储不会改变查询问题本身的时间复杂度，同时，提升算力和存储速度也无法改变查询时间随数据量线性增长的现实[2]。假设 1 min 内查询了 1 亿条记录，那么至少需要 100 min 才能完成 100 亿条记录的查询。虽然可以使用大量的优化技术来缩短查询时间，如更加快的存储设备、更高压缩效率的算法等，但一般来说，查询性能和时间线性关系是无法改变的。即使大规模的并行处理可以用数十倍、百倍的规模来扩展计算集群，以此得到一个强大的算力，但购买和部署这样的计算集群并不容易，同时存在较高的硬件购置和运营成本。

2　关键技术

基于目前大数据处理的现状，通过无限制扩展单机的计算能力来提升数据处理能力是不现实的，但是可以通过空间换时间的思路来达到类似的效果，本文研究的系统引入维度建模[3]。

对于一个给定的数据模型，一般可以先对其所有维度进行组合。例如，对于 n 个维度来说，组合的可能性一共有 2^n 种。对于每一维度组合，取出度量来做聚合计算，将运算的结果存储成一个物化的视图，称为 Cuboid。那么全部维度组合的 Cuboid 作为一个整体，就称为 Cube。简单来说，一个 Cube 也就是许多按维度聚合的物化视图的一个集合[4]。

例如，一个电商的销售数据集中，包括时间（Time）、商品（Item）、地点（Location）和供应商（Supplier）四个维度，度量为销售额（GMV）。那么，所有维度的组合有 $2^4=16$ 种，比如一维度（1D）的组合有［Time］、［Item］、［Location］、［Supplier］4 种；二维度（2D）的组合有［Time，Item］、［Time，Location］等 6 种；三维度（3D）的组合有 4 种；零维度（0D）和四维度（4D）的组合各有 1 种，总共就有 16 种组合[5]。详细维度展开如图 1 所示。

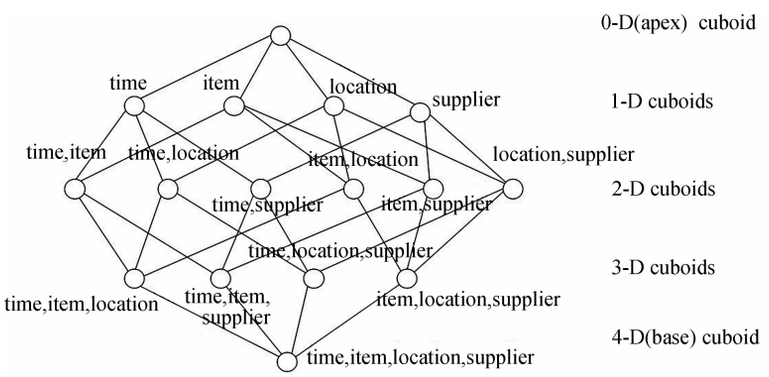


图 1　维度组合模型

如果需要计算 Cuboid，也就是按维度来聚合销售额，用 SQL 语句来表述 Cuboid［Time，Location］，得出 SQL 语句如下：

select Time，Location，Sum(GMV)as GMV from Sales group by Time，Location

如果把计算的结果存储为物化的视图，那么所有 Cuboid 物化视图的总称就是 Cube。

Cube 预计算系统利用计算结果加快查询速度，其工作原理是对数据模型进行立方体预计算[6]，具体工作流程如下：首先，定义一个数据模型，指定维度以及度量；其次，预计算 Cube，计算所有 Cuboid 并保存为物化视图；最后，执行查询时，读取 Cube，生成查询结果[6]。

因为预先计算系统的查询过程并不扫描原始记录，但 pre-calculates 等复杂的操作，例如聚合等会扫描原始记录，并使用预计算结果来执行查询，与 non-pre-computed 查询技术相比，一般快 1～2 个数量级

的速度，非常大的数据集效果会更明显，当数据集达到千亿甚至上万亿时，预计算系统的查询速度较非预计算技术的快近千倍。

3 系统架构与实现

为获取互联网上的招聘岗位数据，本文设计了分布式爬虫，其难点在于如何保证系统的可靠性，即高可用性。由于爬虫是24 h工作，且具有多个节点，因此某个爬虫节点挂掉是较平常的情况，本文基于Paxos协议设计了如下的爬虫系统(图2)。

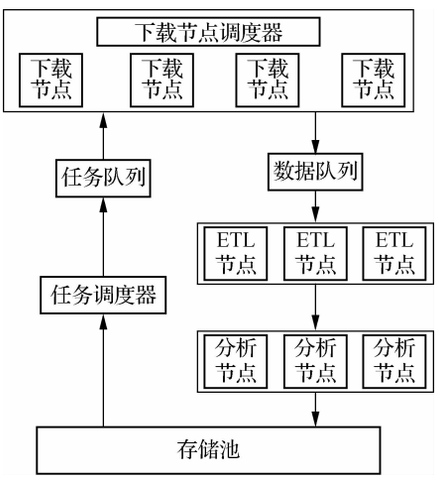


图2 分布式爬虫架构

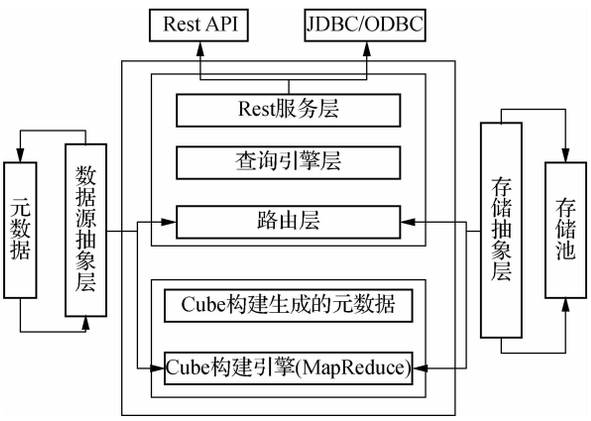


图3 预计算系统架构

从图2看出，系统设计有若干个爬虫下载节点，用于爬取互联网上各大招聘网站的数据。为了保证爬虫下载节点的高可用性，会设置一个爬虫下载的任务调度器，负责监控下载节点的心跳，一旦出现节点宕机，可以立即报告错误，并且进行重启操作。下载节点下载的数据会存到一个消息队列中，下层有ETL节点负责对数据进行清洗，规整后交给分析节点，对数据做去重处理，并在分类后存到存储池对应的分析库里面。系统中的任务调度器，主要负责在任务队列中推送待爬取的URL。

预计算系统主要是为大数据处理提供预计算服务，基于维度建模理论，是一种以空间换时间的思路，整个系统作为大数据分析的核心模块，底层依赖于HDFS，并且对上层应用提供Rest API和JDBC/ODBC接口，架构图如图3所示。

本系统构建在分布式计算平台(Hadoop)上，利用MapReduce的并行处理能力和可扩展基础架构，有效地、可伸缩地处理大规模的数据。同时作为OLAP引擎的系统，包括了很多功能，例如：从数据源获取源数据，基于MapReduce构建多维立方体，利用HBase的列存的特性分布式存储多维数据集数据，提供标准的SQL查询和解析，对查询语句进行优化等。同时支持ODBC、JDBC驱动和RESTAPI等多模块的设计，采用可插拔、灵活的架构，使系统允许更多数据源访问和其他技术作为存储引擎。本模块主要包含以下核心组件：

1) 元数据管理。包括表结构同步、模型设计、立方体设计、数据采样分析。支持维度优化技术，例如层次维度、关节维度和可导出维度，以避免多维数据集数据的扩展。支持多种编码字典算法，实现高效的数据压缩和存储效率。

2) 任务引擎。用于向Hadoop平台提交Cube构建任务，支持多种构建机制，如全表构建、增量构建和流构建，并支持IO优化方法，如自动立方体合并，充分利用了MapReduce计算框架的计算能力。

3) 存储节点。预先计算关系表的源数据，将其存储在密钥数据库HBase中，并充分利用HBase的高效过滤和并行处理技术，以并行计算模式检索数据，支持查询逻辑压缩存储节点。$O(N)$到$O(1)$的计算复杂度降低了数据检索问题。

4) 查询节点。构建在Apache Calcite语法解析器之上，支持JDBC、ODBC、REST等多种协议及接

口，支持大多数 SQL 函数，同时提供自定义计算函数实现，可与主流 BI 工具（如 PowerBI）完美配合。

5）Web 管理端。提供用户友好界面，来向导驱动的模型构建，拥有直观的任务监控和警报以及用户权限管理。

本文重构了预计算的系统体系结构，将数据源、构建引擎和存储引擎的三个主要依赖关系抽象为接口，Hive、MapReduce 和 HBase 采用默认实现。深度用户可以根据需求进行二次开发，用更合适的技术替换一个或多个深度用户，这也为预计算系统技术跟上时代的步伐奠定了基础。如果未来更先进的分布式计算技术取代了 MapReduce，或者更高效的存储系统超越了 HBase，预计算的系统可以以更低的成本取代子系统。从而确保预计算系统的先进性以及可扩展体系结构的额外的灵活性。

本文设计的岗位画像系统是基于上述大数据处理技术、维度建模、预计算技术和爬虫技术的一个大数据处理和分析的系统。首先，系统对爬虫爬取到的数据进行 ETL，再进行维度建模后放到预计算系统中做 Cube 预计算处理，最后基于 Cube 进行分析，使得交互式查询的速度下降到亚秒级别。系统展现的功能模块如图 4 所示。

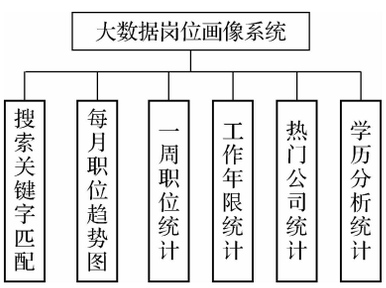


图 4　岗位画像系统功能模块

从上图看出，系统用户从 Web 页面登录，进入岗位画像系统之后，用户可以看到爬虫对于默认关键字的 6 个统计信息图标，可以自己手动输入感兴趣的关键字，查看其统计信息。该系统还提供了一周职位统计、工作年限统计、热门公司统计和学历统计分析。分析界面如图 5 所示。

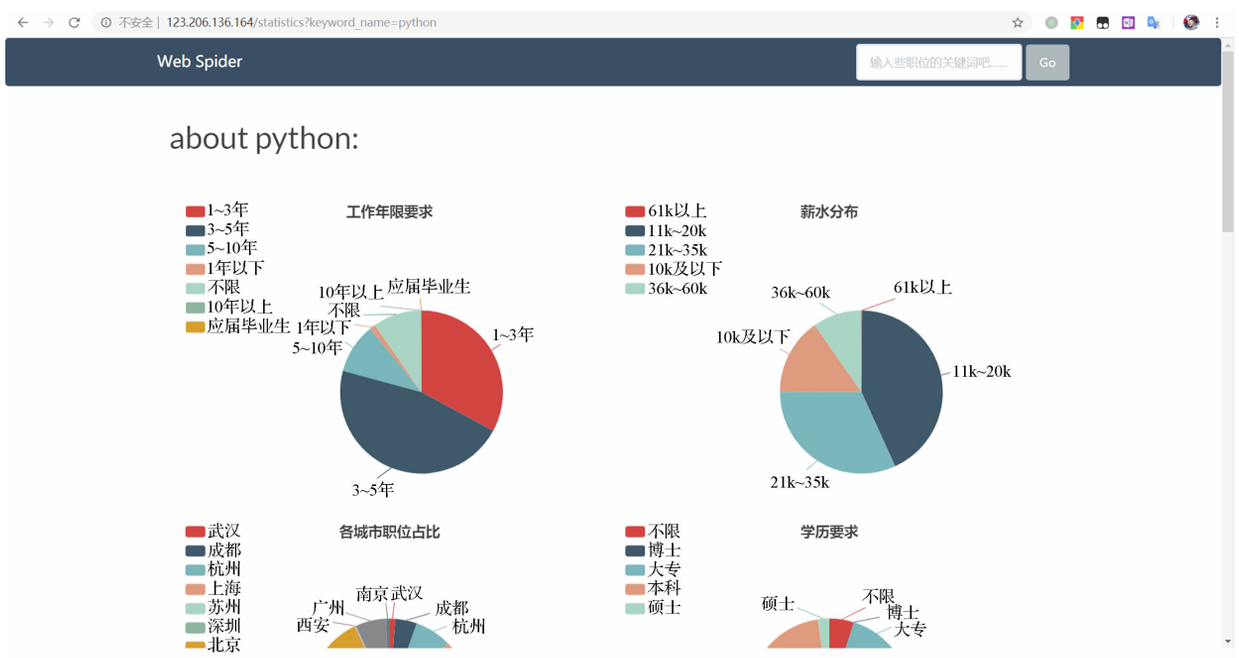


图 5　岗位分析界面

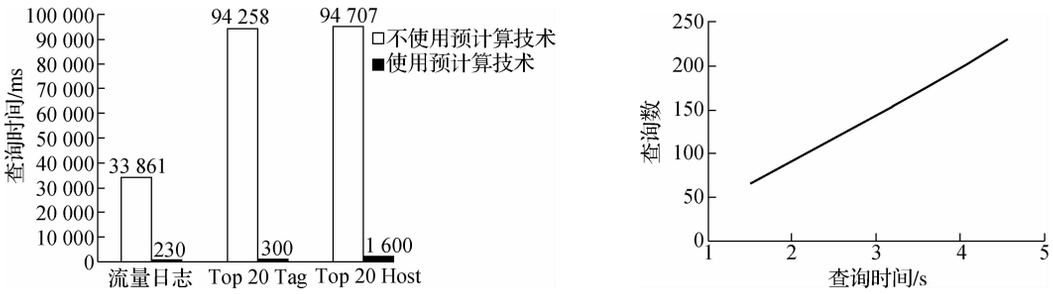
系统默认以 python 为关键字，进行职位信息统计分析，支持从不同的维度对一个职位进行分析，包括但不限于工作年限要求、薪水分布、各城市职位占比和学历要求等。

4　结果与分析

大数据分析的岗位画像系统在使用 Cube 预计算级数后，具有良好的可扩展性和高吞吐量，同时保持高速响应。其性能对比可见图 6。

图 6(a)是使用预计算技术的查询速度与不使用预计算技术的查询速度的比较。可以看出，在这三个

测试查询中，使用预计算技术比不使用预计算技术分别快147倍、314倍和59倍。同时，图6(b)显示了预计算系统的吞吐量及其可扩展性。仅使用一个预计算系统的实例，预计算的系统每秒可处理近70个查询，远高于通常每秒20个查询的水平。



(a)使用不同技术时的查询时间　　(b)吞吐量

图6　性能分析图

这主要是因为预计算减少了查询所需的计算量，允许预计算的系统在相同的硬件配置下携带更多的并发查询[7]。由此画像系统在构建出来之后，可以提供近实时的交互式查询，基本可以解决系统查询高延迟的问题。

5　结　语

维度建模并不是近期的新理论，在大数据到来之前的OLAP与OLTP分析中，由于数据量并不十分庞大，传统的分析技术可以满足用户需求。当信息系统拥有巨大的数据量，并对查询响应时间有秒级要求，就要考虑使用维度建模来进行预计算，可以结合场景考虑使用Druid或Kylin这种OLAP类型的大数据预计算系统[8]。相信维度建模作为目前大数据预计算技术中的重要组成部分，将会发挥不可替代的作用。

在人工智能时代，利用大数据处理相关技术对数据进行爬取、挖掘、分析，并通过智能筛选清洗出有实用价值的数据，为企业招聘和高校人才培养做重要参考。

参考文献：

[1] 李伟，孙新杰，武晋民. 基于Hadoop的大数据分析平台构建研究[J]. 信息通信，2017(8)：247－248

[2] 范小春. 智慧校园环境下高校大数据治理及应用策略[J]. 金陵科技学院学报，2018(4)：48－51

[3] 钟华. 科学大数据智能分析软件的现状与趋势[J]. 中国科学院院刊，2018(8)：812－817

[4] Haughey T. Is Dimensional Modeling One of the Great Con Jobs in Data Management History Parts 1 & 2[EB/OL]. (2004－03－08)[2019－03－17]. http://xueshu.baidu.com/usercenter/paper/show?paperid=4970e0a5ecb97f58932c3523d358d7c3&site=xueshu_se

[5] Ross M. Design Tip #105 Snowflakes，Outriggers，and Bridges. Kimball Group[EB/OL]. (2008－09－03)[2019－03－02]. https://www.kimballgroup.com/2008/09/design-tip-105-snowflakes-outriggers-and-bridges

[6] Rainardi V. The Main Weakness of Snowflake Schemas[EB/OL]. (2012－07－16)[2019－03－17]. https://dwbi1.wordpress.com/2012/07/16/the-main-weakness-of-snowflake-schemas

[7] Rainardi V. When To Snowflake. Data Warehousing，BI and Data Science[EB/OL]. (2011－05－11)[2019－03－16]. https://dwbi1.wordpress.com/2011/03/11/when-to-snowflake

[8] Rainardi V. Star Schema or Snowflake. Data Warehousing，BI and Data Science[EB/OL]. (2012－03－13)[2019－03－16]. https://dwbi1.wordpress.com/2012/03/13/star-schema-or-snowflake

(责任编辑：湛　江)