

分布式气象大数据快显技术的设计与实现

雷鸣

(天津市气象信息中心 天津 300074)

摘要 随着气象数据不断迈向海量的大数据范畴,原有省级 CIMISS 已不能满足实际需求。为了解决系统实时查询性能低下、动态扩展能力不足等问题,基于 GBase、MySQL、MongoDB、HBase 等不同的数据库,针对不同的气象数据特征进行分库存储,构建大数据分布式存储数据中心,并利用并行计算技术针对数据服务显示功能进行增速。实践证明,该方案不但能与 CIMISS 无缝衔接,而且还极大提升数据的查询显示速度,且并发响应能力强,时间跨度越大查询优势则越明显。与 CIMISS 相比,整体系统查询速度平均提升了 860%,能够有效地满足气象业务和科研的服务需求。

关键词 数据共享 CIMISS 分布式数据库 MUSIC 关系数据库

中图分类号 TP399 **文献标志码** A **DOI**:10.3969/j.issn.1000-386x.2023.02.003

DESIGN AND IMPLEMENTATION OF DISTRIBUTED METEOROLOGICAL BIG DATA QUICK DISPLAY TECHNOLOGY

Lei Ming

(Tianjin Meteorological Information Center, Tianjin 300074, China)

Abstract With the meteorological data constantly moving into the category of massive big data, the original provincial CIMISS has been unable to meet the actual needs. In order to solve the problems of low real-time query performance and insufficient dynamic expansion ability of the system, based on different databases such as GBase, MySQL, MongoDB and HBase, the distributed big data storage environment center was built according to different meteorological data characteristics. The parallel computing technology was used to increase the data service display function. The practice shows that the scheme can not only connect with CIMISS seamlessly, but also greatly improve the query and display speed of data, and has strong concurrent response ability. The larger the time span is, the more obvious the query advantage is. Compared with CIMISS, the overall system query speed is improved by an average of 860%, which can effectively meet the service requirements of meteorological business and scientific research.

Keywords Data sharing CIMISS Distributed database MUSIC Relational database management system(RDBMS)

0 引言

随着气象业务高速扩展,数据的种类与数据量不断增长。而与此同时,针对气象数据的服务性能和响应速度的要求却越来越高。但目前省级全国综合气象信息共享系统(CIMISS),却是 2009 年由国家气象信息中心负责组织建设,集数据收集、分发、处理、存储和

共享于一体。2013 年,该系统推广部署在全国各省级气象数据中心,并获得良好应用。但随着设备老化和技术的落后,目前其数据处理能力已经明显无法满足要求^[1-3]。但省级部门却无权针对该系统进行改造。

为了进一步提升气象数据服务的敏捷性和存储动态扩展的需求。同时又能够与 CIMISS 进行无缝衔接,在充分参考相关行业在解决海量数据查询的成功方案基础上^[4-8],利用分布式技术^[9-11],构建满足省级特

色需求的数据服务中心。同时,利用 CIMISS 的气象数据统一服务 MUSIC 接口(Meteorological Unified Service Interface Community)^[12-14],打通多系统之间的壁垒,屏蔽异构环境,提供统一的对外数据服务功能。

1 系统架构设计

通过引入系统面向的四类用户,结合整个平台进行管理和监控运行监管体系和平台建设,并遵循气象信息标准化体系规范,针对系统的六个层次分别进行了细化与分析,其具体技术架构如图 1 所示。

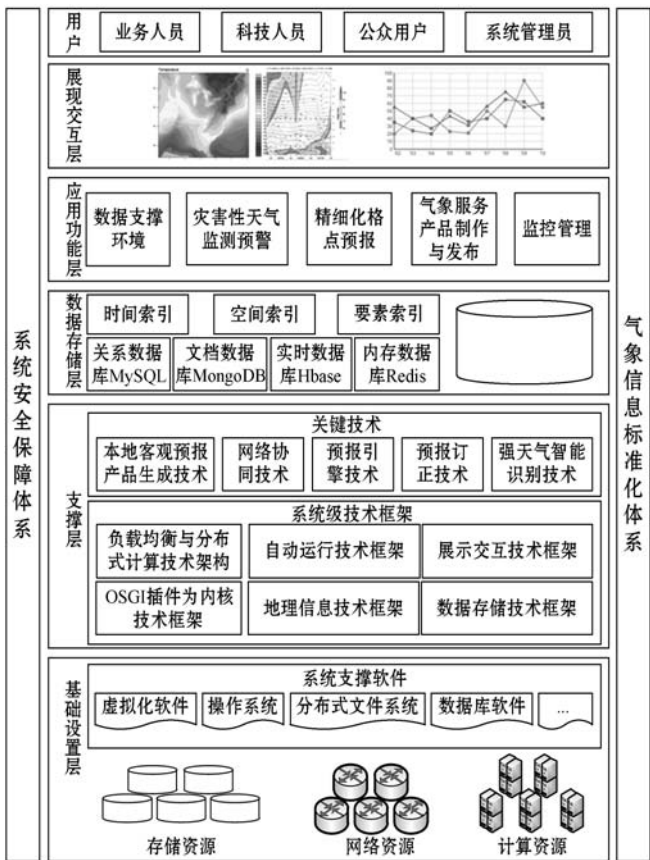


图 1 系统总体框架设计

天津省级数据服务中心的总体设计机构共分为 5 层:展现交互层、应用功能层、数据存储层、基础支撑层与硬件层。

展现交互层:该层是天津气象大数据共享平台软件进行交互的入口,该层主要负责接收用户提交的输入请求,通过后端的接口层对业务逻辑层进行访问,从而获得、并向用户输出可视化响应。

应用功能层:应用功能层则负责接收前端用户的输入请求,并以业务逻辑过程能够理解的方式将其转化。同时,根据特定的业务逻辑向数据层有序地发送数据请求,并将返回的数据层数据进行解释和组合,形成用户所需的信息,最终再返回到展现交互层。这一

层在整个应用软件系统里,是业务逻辑处理与实现的核心。

应用功能层采用基于组件化架构思想进行设计,即将天津气象大数据共享平台软件的业务功能单元封装成各个相对独立又互相联系的功能组件,通过支撑层的调度控制,各功能组件相互配合,协作完成系统的各项任务。

数据存储层:该层针对气象数据进行管理,并向应用服务层提供开放式访问的标准化接口。该层负责提供访问位于持久化容器中数据的功能,以及涉及从持久化介质中写入数据或者读取数据的工作。

基于 HBase 存储半结构化混合数据,基于 MySQL 存储结构化观测和预报资料,基于 MongoDB 存储非结构化数据,数据存储层为系统提供对缓存进行管理的功能,在此基础上,分别对数据库和文件库的进行统一的接口封装,为应用功能层提供统一基于时间索引、空间索引和要素索引的大数据 SQL 查询器。

天津气象大数据共享平台软件为加快上层访问数据存储层数据/文件的访问速度,在数据库与文件库物理存储基础上使用缓存机制。

支撑层:支撑层描述了实现天津气象大数据共享平台软件所使用的技术框架和所采用的关键技术,为应用功能层各个业务组件、功能模块起到支撑与组织的作用。支撑层包括两部分:系统级技术框架及关键技术。

系统级技术框架描述支撑整个系统应用功能所使用技术架构,主要包括:展示交互技术框架、自动运行技术框架、地理信息技术框架、OSGI 插件微内核技术框架、负载均衡与分布式计算框架和数据存储技术框架。

关键技术指构建系统级技术框架中所采用的技术,主要包括本地客观预报产品生成技术、网络协同技术等。

2 可扩展性设计决策

为更好适应未来气象预报业务发展,系统需具有可扩展性与开放性。可扩展性具体表现为业务可扩展性,可动态加载气象预报业务算法。

对业务扩展性需求,采用的设计决策是:将具体的算法封装成动态链接库,与具体的业务逻辑相分离,算法可替换,参数可配置,业务流程可配置。业务流程配置如图 2 所示。

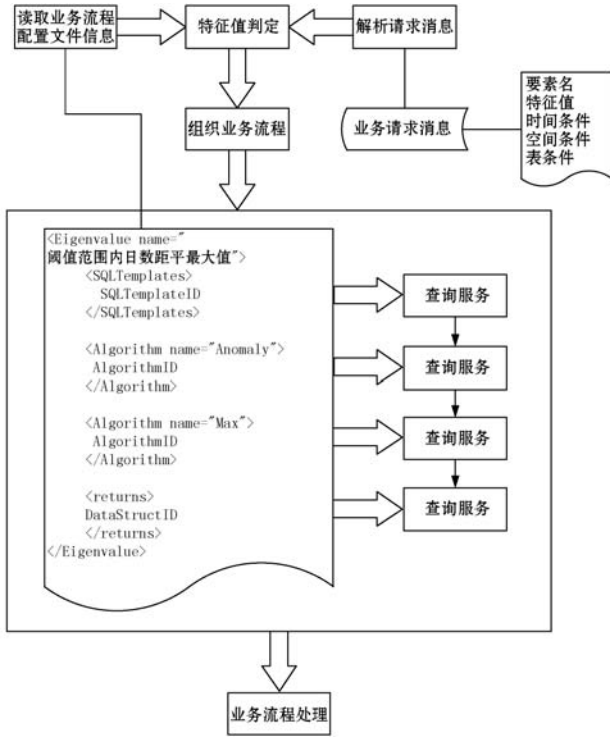


图2 业务流程配置示意说明

针对系统的扩展性需求,采用设计决策是:业务逻辑及公共服务层与展示层之间功能分离,模块之间松耦合,分别部署在不同的物理节点,可重用业务逻辑及公共服务层。

3 系统存储设计

3.1 系统存储架构设计

为了做到数据与应用分离,使用户对后台变动无感。基于 MUSIC 服务接口,将全部数据库打通,形成一个逻辑上统一的数据服务中心,对外透过 API 接口提供服务。整个存储设计架构如图 3 所示。

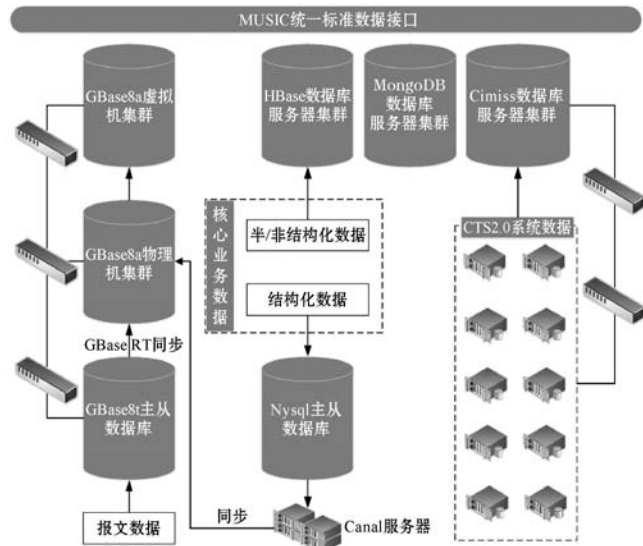


图3 系统存储架构设计图

3.2 数据库存储配置

系统涉及到多种数据库,如 MySQL、MongedB、HBase 等。为进一步提升系统响应速度,特别针对各数据库分别进行了优化处理。限于篇幅,仅以 MySQL 优化为例:

MySQL 提供了一些存储分配参数,例如:数据库的大小、锁的数目,以及使用的缓冲区大小等。但这些分配参数的默认值不能达到天津省级数据服务中心的功能需求。为进一步优化数据库性能,根据天津省级数据服务中心的特点,针对系统的配置参数进行了如下调整:

```
[mysqld]
server-id = 115
basedir = /mysqldata
datadir = /mysqldata/data
log-bin = mysql-bin
log-bin-index = mysql-bin.index
sync-binlog = 1
max-binlog-size = 200M
expire_logs_day = 3
skip-host-cache
skip-name-resolve
innodb_buffer_pool_size = 40G
innodb_log_buffer_size = 32M
max_connections = 1000
event_scheduler = ON
```

4 并行加速算法

为了进一步提升数据服务的速度,尤其是涉及到图形渲染等高密度计算场景时,本文利用并行计算技术进行数据显示增速^[15],如:针对格点数据等值线提取、色斑渲染等功能。天津省级数据服务中心所使用的 WebGIS 以及其他需要渲染计算等方面,均采用了并行运算。其具体策略为:结合 OpenMP 的 CPU 并行计算以及基于 OpenCL 的 GPU 并行计算技术,将其应用与气象格点数据计算处理和气象信息图形显示分析中,提高基于 WebGL 地图渲染的执行效率和运算速度。

4.1 基于 OpenMP 的 CPU 并行算法

CPU 并行框架底层基于 OpenMP 编程框架,透过使用计算机的多线程多核处理机制,提供了对 CPU 并行算法的高层抽象描述,以及线程粒度的控制和负载均衡,具体工作流程如图 4 所示。

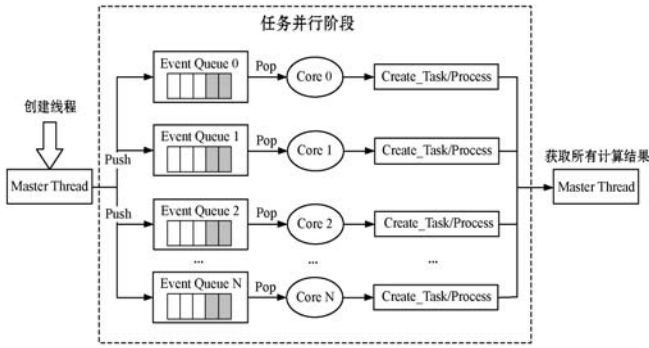


图 4 并行计算工作流程示意图

OpenMP 的算法设计基于如下阿姆达尔定律的最小化串行代码原则进行:

$$S = \frac{1}{1 - a + \frac{a}{n}} \quad (1)$$

式中: a 是并行计算部分占用的比例大小, n 则是并行处理部分的结点个数,即处理器个数。可以看到,当没有串行,而仅有并行,即 $1 - a = 0$ 时,最大加速比 $S = n$;而当仅有串行,并不存在并行,即 $a = 0$ 时,其最小加速比 $S = 1$;最大加速比的上限发生在 $n \rightarrow \infty$ 时,此时,极限加速比 $S \rightarrow 1/(1 - a)$ 。为了使速度最大化,在并行处理节点个数固定的情况下,应尽可能地提升并行计算部分所占的比例。

OpenMP 中的执行模型采用的是 fork-join,其中, fork 的含义表示唤醒已有线程或者创建线程,而 join 则代表了多线程的会合。当 Fork-join 执行模型最初执行时,它仅有一个在运行的线程:即“主线程”(Master Thread)。而当需进行并行计算时,系统则会由主线程派生出新线程来执行并行任务。而此时,主线程与派生线程将会在并行执行阶段一同协同工作。当并行运算完结之后,派生线程会自动阻塞或退出,而不再继续执行,控制流程将会回到单独的主线程中。当系统涉及高密度数据计算的时候,如:数据渲染和数据时间插值等操作,计算将会由 CPU 转到 GPU 中进行处理。

4.2 基于 OpenCL 的 GPU 并行算法

GPU 并行计算基于开放运算语言(Open Computing Language, OpenCL),利用 GPU 强大的浮点数计算能力,辅助 CPU 完成大规模的并行计算任务。OpenCL 可运行在多种不同的平台之上(Windows、Linux、Unix 等),通过对不同平台底层的抽象与封装,屏蔽了相异平台底层的不同设计,并对应用层提供了统一的接口服务。

而 GPU 渲染常采用构建三角形带和 LOD(多细节层次)技术^[16]以减少 GPU 固有的渲染数据量,则其渲

染的数据量公式如下:

$$A = \sum_{y=1}^m (k_y N_y)^x k_b^{-x} (k_f I n)^x \quad (2)$$

式中: m 则是场景模型的总数; k_y 是 N_y 缩减的比例; N_y 则是单个模型在场景中的总顶点数; x 是反射次数; k_b 是在并行架构下顶点的冗余度系数(在 2-3 之间); k_f 是材质种类所固有的反射系数; I 是平均光强值; n 是场景光源的个数。

设 t_y 为同一线程块处理面片集合的平均时间, A_b 为同一批次处理的渲染数据量, T 为处理渲染数据消耗的总时间,则有:

$$T = \frac{\sum_{y=1}^m t_y (k_y N_y)^x k_b^{-x} (k_f I n)^x}{A_b} \quad (3)$$

整体 OpenCL 并行计算框架的架构设计由设备、上下文环境、程序、内核、内存对象、命令队列六个部分组成,具体组成如图 5 所示。

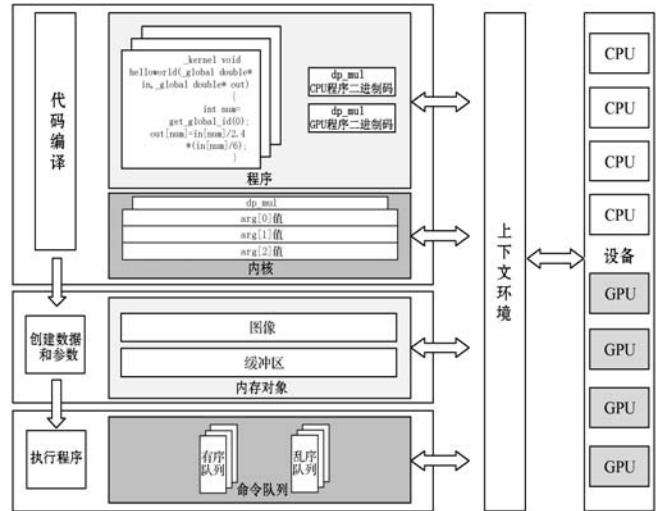


图 5 并行计算系统流程图

设备:它是并行计算框架中的计算单元,一个 GPU 或者 CPU 将会对应一个设备。而设备通过命令队列,获取自己需要执行的计算指令。

上下文环境:上下文是一个抽象的容器,是整个并行计算框架的纽带,上下文环境管理在设备上的列中的有序队列与无序队列。只有在一个上下文环境上的系统的设备才能彼此交流工作。

程序:这是所有代码的集合,包含核函数和其他库。OpenCl 是一个动态编译的语言,代码编译后生成一个中间文件(可根据需要实现为虚拟机代码或者汇编代码),在使用时连接进入程序读入处理器。

内核:这是在设备上运行的核函数及其参数组。为了进一步提升计算速度,在其中特别使用了单指令多数据流技术。这是一种采用一个控制器来控制多个处理器,同时对一组数据(又称“数据矢量”)里面的每

一个各自进行相同操作,从而达到空间上并行性的技术。

内存对象:包括图像和缓冲区,并行计算需要在不同设备上使用的内存,内存对象由上下文创建,从而实现上下文管理的多个设备,能够将内存对象中的数据进行共享。

命令队列:这是上下文环境给每个设备提交的指令序列,通过命令队列,上下文环境将需要执行的指令,发送到每个设备上。在顺序执行命令队列里(默认),命令将会按照接收的顺序压入到命令队列中。乱序队列允许 OpenCL 在实现时重排命令以便高效地执行。如果使用乱序队列,须指定依赖关系以确保正确地执行顺序。

OpenCl 执行分为三个阶段,第一个阶段进行代码编译,创建上下文环境以及命令队列,生成内核与程序,并完成设备的初始化;第二个阶段创建参数和数据,上下文环境创建内存对象,并将计算需要使用的数据写入其中;第三个阶段上下文环境将需要执行的计算指令发送到命令队列中,并等待设备计算完成。设备计算完成之后,上下文环境读取内核对象处理的结果(存放在内存对象中),并释放资源。

5 实践与测试

为大数据分布式存储数据中心的数据服务能力,特别对其进行了综合测试。测试使用的硬件配置为:CPU i5-3470 3.20 GHz,内存 4 GB,支持谷歌内核的主流浏览器,如 Chrome。获得的系统测试指标如表 1、表 2 所示。

表 1 基于 WebGIS 的自动站显示指标

产品内容	站点个数	要素个数	耗时/s
自动站站点	140	7	0.520

表 2 基于 WebGIS 的数据渲染显示指标

产品内容	格点个数	格距/(°)	耗时/s
卫星云图	2 000 × 1 500	0.05	0.625
天气雷达组网	920 × 770	0.01	0.620
色斑图	156 × 167	0.09	0.350
风场	156 × 167	0.09	0.700
等值线	156 × 167	0.09	2.200

为了进一步获得系统的查询性能,特别针对页面响应速度进行了测试,获得如表 3 所示的各项数据查询结果。

表 3 页面查询响应速度表

查询类型	网格数	分辨率 /km	响应时间/s		
			传输	显示	总计
自动站 单时效	320 × 250	5	0.880	0.217	1.1
雷达 单时效	680 × 880	3	0.640	0.475	1.1
卫星 单时效	850 × 1 160	5	0.512	0.722	1.2
模式 单时效	340 × 464	12.5	1.6	0.460	2.1

其中,查询类型为常规产品中的典型查询项目,而响应时间中的显示则为第 1 次的显示耗时。页面的响应速度比基于 CIMISS 的速度平均提升了 860%。下面展示了几类不同的查询显示效果图。

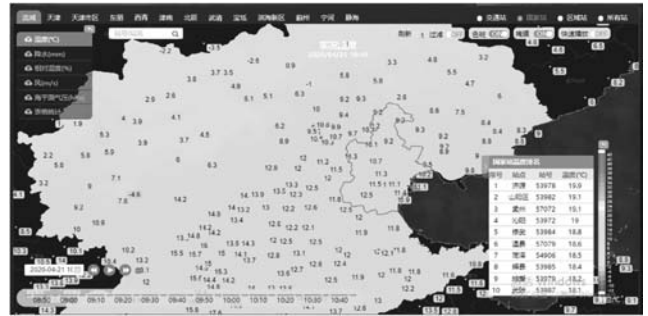


图 6 自动站实况温度查询效果图

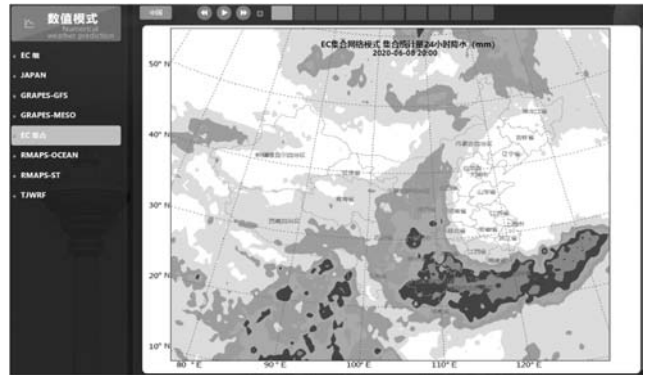


图 7 EC 集合统计量:24 小时降水

6 结语

本文在不改变现有业务系统和系统架构的基础上,基于 MUSIC 接口,利用分布式和并行计算技术构建了满足省局气象需求的大数据环境中心,提高了气象数据显示分析过程中各种复杂的气象算法运算的效率,并对系统中整个业务流程执行效率和图形显示分析的人机交互体验也有较大的提升,为气象数据的高质量服务,提供了一个可行的解决方案。

参考文献

- [1] 周晓丹. 基于 Oracle RAC 平台的海量数据 DML 处理性能 [D]. 厦门:厦门大学,2007.

计等问题还需要解决。

参 考 文 献

- [1] 苗清. 如何运用 AI 与大数据打造优质智能安防报警[J]. 中国安防, 2019(12):94-97.
- [2] 郝爽, 李国良, 冯建华, 等. 结构化数据清洗技术综述[J]. 清华大学学报(自然科学版), 2018, 58(12):1037-1050.
- [3] 陈俊月, 郝文宁, 张紫萱, 等. 基于改进句子相似度的释义识别研究[J]. 计算机工程, 2020, 46(9):76-82.
- [4] Niewiarowski A. Short text similarity algorithm based on the edit distance and thesaurus [J]. Technical Transactions. Fundamental Sciences, 2016, 113:159-173.
- [5] Palkowski M, Bielecki W. Parallel tiled codes implementing the smith-waterman alignment algorithm for two and three sequences[J]. Journal of Computational Biology: A Journal of Computational Molecular Cell Biology, 2018, 25(10):1106-1119.
- [6] Jing C, Zhang L. The review of the acceleration of Smith-Waterman algorithm by using CUDA-enable GPU[C]//2015 3rd International Conference on Machinery, Materials and Information Technology Applications, 2015:758-766.
- [7] 王常武, 韩菁华, 张付志. 一种相似重复元数据记录检测方法[J]. 计算机工程, 2009, 35(21):85-87.
- [8] 陈亮, 杜璐, 胡康. 基于分块和滑窗技术的相似重复记录检测算法研究[J]. 计算机应用与软件, 2019, 36(4):262-267.
- [9] 吕国俊, 曹建军, 郑奇斌, 等. 基于多目标蚁群优化的单类支持向量机相似重复记录检测[J]. 兵工学报, 2020, 41(2):324-331.
- [10] 张攀. 面向重复记录检测的数据清洗算法的研究[D]. 西安:西安电子科技大学, 2018.
- [11] 孟祥逢, 鲁汉榕, 郭玲. 基于遗传神经网络的相似重复记录检测方法[J]. 计算机工程与设计, 2010, 31(7):1550-1553.
- [12] Gu J, Wang Z, Kuen J, et al. Recent advances in convolutional neural networks[J]. Pattern Recognition, 2018, 77:354-377.
- [13] 李丽华, 胡小龙. 基于深度学习的文本情感分析[J]. 湖北大学学报(自然科学版), 2020, 42(2):142-149.
- [14] 李卫疆, 伊靖. 基于扩展特征矩阵和双层卷积神经网络的微博文本情感分类[J]. 计算机应用与软件, 2019, 36(12):150-155.
- [15] 宋岩, 刘汉永, 宁向南, 等. 基于层次特征提取的文本分类研究[J]. 计算机应用与软件, 2020, 37(2):68-72, 77.
- [16] 肖琳, 陈博理, 黄鑫, 等. 基于标签语义注意力的多标签文本分类[J]. 软件学报, 2020, 31(4):1079-1089.
- [17] 张璞, 陈超, 陈韬, 等. 两分类器融合的中文微博用户性别分类方法[J]. 计算机工程与设计, 2019, 40(1):268-272.
- [18] Fukushima K. Neocognitron: A hierarchical neural network capable of visual pattern recognition[J]. Neural Networks, 1988, 1(2):119-130.
- [19] Lecun Y, Bottou L. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.

(上接第 16 页)

- [2] 周晓丹, 冯少荣, 薛永生. Oracle 10g RAC 核心技术研究与分析[J]. 计算机工程, 2007, 33(7):53-55.
- [3] 姜召凤. Oracle RAC 数据库缓存优化方法研究[D]. 大连:大连海事大学, 2009.
- [4] 杨东日, 陈跃, 刘姝祎. 面向健康大数据快速读写的存储系统设计[J]. 计算机工程与设计, 2018, 39(10):3063-3067.
- [5] 李佳, 徐胜超. 基于云计算的智能电网大数据处理平台[J]. 计算机工程与设计, 2018, 39(10):3073-3079.
- [6] 赵康, 杨余旺. 基于 Hadoop 的物联云监控系统的设计与实现[J]. 计算机与数字工程, 2019, 47(7):1738-1742.
- [7] 李斌, 郭景维, 彭骞. 面向大数据存储的 HBase 二级索引设计[J]. 计算技术与自动化, 2019, 38(2):124-129.
- [8] 邱超, 许金涛, 元晓华. 基于大数据技术的水情云数据中心设计与研究[J]. 浙江大学学报, 2019, 46(1):92-100.
- [9] 宋智, 徐晓莉, 张常亮, 等. 应用分布式存储技术优化省级 CIMISS 数据服务能力[J]. 热气象科技, 2019, 47(3):433-438.
- [10] 吴燕波, 薛琴, 向大为, 等. 云平台下的 NoSQL 分布式大数据存储技术与应用[J]. 现代电子技术, 2016, 39(9):44-47.
- [11] 李波, 杜景林, 李正方. 基于 SOA 的气象数据共享平台研究[J]. 电子设计工程, 2019, 27(4):25-29.
- [12] 曾行吉, 任晓炜, 宋瑶, 等. 微服务在气象数据服务中的应用研究[J]. 气象研究与应用, 2019, 40(1):80-83.
- [13] 赵芳, 熊安元, 张小纓, 等. 全国综合气象信息共享平台架构设计技术特征[J]. 应用气象学报, 2017, 28(6):750-757.
- [14] 曾行吉, 李涛, 詹利群, 等. 基于 MUSIC 的特色数据与产品回写 CIMISS 方法研究[J]. 气象研究与应用, 2018, 39(1):111-114.
- [15] 丁晓刚, 鲍广宇, 胥秀峰. 一种基于多核并行计算的目标分配算法设计[J]. 指挥控制与仿真, 2014, 36(5):97-99, 127.
- [16] 李白云, 赵春霞. GPU 实时构建四叉树的快速地形渲染算法[J]. 计算机辅助设计与图形学报, 2010, 22(12):2259-2264.