

MongoDB 数据库在电力工程行业的应用

余建忠[✉]

(中国能源建设集团广东省电力设计研究院有限公司, 广东 广州 510663)

摘要: [目的] 为了贯彻落实国资委《关于加快推进国有企业数字化转型工作的通知》, 国有企业面临着数字化转型压力与挑战, 很多电力工程企业都在探索数字化转型的方向, 利用云大物移智等数字化技术, 解决工程项目全生命周期数据的管理是一个重要的前提条件。[方法] 针对电力工程项目全生命周期数据的大量性、多样性、复杂性, 分析了关系型数据库在大部分场景的不适用性, 采用 NoSQL 数据库 MongoDB 解决数据量大和数据多样性、复杂性的问题。[结果] 作者以 MongoDB 为核心数据库进行设计构建, 完成工程大数据平台建设, 实现了电力工程项目过程数据的灵活建模与管理, 并作为工程项目全过程的数字化产品的数据底座。[结论] 实践应用证明了 MongoDB 数据库相比于关系型数据库更加适合数据量大、数据多样性、复杂性的场景, 存储效率更高, 数据查询统计能力更强。

关键词: 电力; 工程大数据; 数据多样性; 关系型数据库; MongoDB; 数据建模; 灵活

中图分类号: TM6; TP311

文献标志码: A

文章编号: 2095-8676(2023)S1-0110-07

开放科学(资源服务)二维码:



Application of MongoDB Database in Power Engineering Industry

YU Jianzhong[✉]

(China Energy Engineering Group Guangdong Electric Power Design Institute Co., Ltd., Guangzhou 510663, Guangdong, China)

Abstract: [Introduction] In order to implement the "Notice on Accelerating the Digital Transformation of State-owned Enterprises" issued by the State-owned Assets Supervision and Administration Commission, state-owned enterprises are facing the pressure and challenge of digital transformation. Many power engineering enterprises are exploring the direction of digital transformation, and it is an important prerequisite to solve the management of engineering project life cycle data by using digital technologies such as cloud computing, big data, Internet of Things, mobile Internet, artificial intelligence. [Method] In view of the large volume, diversity and complexity of the whole life cycle data of power engineering projects, this paper analyzed the inapplicability of relational databases in most scenarios, and used the MongoDB, the NoSQL database, to solve the problems of large volume, diversity and complexity of data. [Result] The author designs and constructs engineering big data platform using MongoDB as the core database, which realizes the flexible modeling and management of the process data of power engineering projects, and serves as the data base of the digital products in the whole process of the engineering project. [Conclusion] The research has proved that MongoDB database is more suitable for scenarios with large data volume, data diversity, and complexity than relational database, with higher storage efficiency and stronger data query and statistical capabilities in practical applications.

Key words: electric power; engineering big data; data diversity; relational database; MongoDB; data modeling; flexible

2095-8676 © 2023 Energy China GEDI. Publishing services by Energy Observer Magazine Co., Ltd. on behalf of Energy China GEDI.

This is an open access article under the CC BY-NC license (<https://creativecommons.org/licenses/by-nc/4.0/>).

0 引言

随着信息化技术的快速发展, 云计算、大数据、物联网、移动互联网、人工智能等技术已经融入到我们工作和生活的方方面面, 带来了前所未有的便

利。电力行业作为我国重要的基础性产业, 积累了大量的设计、建造、运营的数据, 对这些数据进行组织管理, 形成基础数据库, 建立电力工程大数据中心, 是电力企业数字化转型的关键所在^[1]。

收稿日期: 2022-10-26 修回日期: 2022-12-17

基金项目: 2021年广东省自然资源厅海洋六大产业专项资金资助“海上风电场海洋环境立体监测网关键技术及装备产业化”(GDNRC[2021]37); 中国能建广东院科技项目“工程大数据技术及大数据平台研究”(EV06211W)

为了实现从电力工程设计到运维的“全产业链”数据赋能,不断把我们企业在电力能源领域积累的认知规律通过“数据+算力+算法”的模式嵌入到物理世界,实现“全面可观、精确可测、高度可控”,助力构建清洁、低碳、安全、高效的能源体系,为实现“双碳”目标贡献力量。以自主可控为原则,采用“大数据+物联网”的新一代数字化技术,围绕电力工程的规划、设计、建造、运营环节,对电力工程项目的共性需求进行抽象,打造电力行业的工程大数据中心。然而,工程项目全生命周期的数据不仅仅是复杂多样的,数据类型从结构化、半结构化到非结构化,数据来源不一、形式不一、标准不一,而且还是海量的,特别是运维阶段的数据。

面对工程中各种复杂多样且海量数据时,系统设计时需要以大数据技术和数据管理理论为基础,通过深入探索分析,采用多种数据库相融合,进而提升多类型复杂数据的管理和使用效率^[2]。本文主要介绍在电力工程行业的某些重要场景下,说明关系型数据库的不适用性,介绍基于 MongoDB 数据库进行设计建模的优势。

1 关系型数据库遇到的问题

在工程设计中,各专业使用不同的设计软件,如:PDMS、Revit、CAD 等,每个软件都有自己的元件库或族库,相同类型的元件有大部分相同的属性,少部分不同的属性,如手动蝶阀和电动蝶阀,它们的相同

属性有通径、外径、高度、长度等,电动蝶阀多了一个侧向长度的属性。不同类型的元件的属性差异非常大,有些类型的元件属性只有几个,而有些类型的元件属性甚至有十几个,甚至更多,如蝶阀属性只有五六个,而弯头的属性却有十二三个。并且在不同的设计软件中,相同类型的同一种元件,属性也有可能不同。同理,在运维期的设备也跟元件类似,不同的设备的属性不同,且属性的数量也不同。在 MongoDB 等 NoSQL 数据库还没出现之前,数据存储传统的方案一般都是采用基于关系型数据库进行设计,而采用关系数据库进行存储一般有两种较为常用的方案。

方案一:预先定义 1 张大宽表,也就是非常多列的数据表,这张表预留了多个字符串、整型、浮点型来存储不同的属性,预留列的数量等于设备或元件中最多属性的个数。每个属性单独存储在一列,由于不同设备或元件属性个数差别很大,导致该数据表是一个稀疏表,存储效率低,如图 1 所示,预留的字段很多是空的。最为关键的是难以保证不同设备或元件的同一类型的属性存储在同一列,如图 1,元件 A 的材料类型(类型:铝合金)存储在 str3 这一列,而元件 B 的材料类型(类型:钢)则存储在 str4 这一列,将无法实现根据材料类型对元件进行过滤查询和统计查询,更无法对预留列建立索引加快查询效率,只能通过全表扫描,在应用层面对数据进行查询和统计,导致效率低,复杂度高。

id	class_id	str1	str2	str3	str4	str5	int1	int2	int3	int4	int5	double1	double2	double3	double4	double5
15	5	ZJM01	16J601	不锈钢	(Null)	(Null)	3	(Null)	(Null)	(Null)	(Null)	2100	3000	(Null)	(Null)	(Null)
16	5	ZJM02	16J601	铝合金	(Null)	(Null)	5	(Null)	(Null)	(Null)	(Null)	1500	2200	(Null)	(Null)	(Null)
17	5	YM105	17J610-1	钢	(Null)	(Null)	5	(Null)	(Null)	(Null)	(Null)	2400	2700	(Null)	(Null)	(Null)
18	6	上悬SXC	15ZJ602	E03	铜	(Null)	6520	4	(Null)	(Null)	(Null)	900	2100	(Null)	(Null)	(Null)
19	6	上悬SXC2	15ZJ602	E03	铝合金	(Null)	3658	5	(Null)	(Null)	(Null)	1800	2100	(Null)	(Null)	(Null)
20	6	上悬SXC1	15ZJ602	E03	铝合金	(Null)	3658	5	(Null)	(Null)	(Null)	1800	2100	(Null)	(Null)	(Null)
21	7	节能GC	11CJ27	铝塑复合型材	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	(Null)	2100	(Null)	(Null)	(Null)	(Null)
22	8	PM1	17J610-1	不锈钢板	1.0-1.5 mm	1.5-2.0 mm	5	7686	4	3	12	8400	8400	12.23	41.22	65.2

图 1 稀疏大宽表

Fig. 1 Sparse large wide table

方案二:定义 2 张数据表,1 张为设备或元件基础表(只存储设备或元件的基础通用的共有属性),而另 1 张扩展数据表,采用按行存储方式存储每个设备或元件的其他属性,即一个属性存储在该表一行,每个设备或文件需要占用多行进行存储,如图 2

所示。该方案的缺点是扩展数据表的数据量很大,导致查询性能低,如一个电厂的精细化模型,假设构件的数量能达到 500 万个,一个构件(设备或元件)的属性平均为 20 个,则扩展数据表的需要 1 亿行。并且采用该方案无法通过 SQL 语言查询属性等于

某个条件的设备或者元件,如需要查询材料类型为铝合金的元件有哪些,与方案一类似,也只能通过全表扫描,上层应用进行过滤,性能非常低。

随着数字化相关技术的不断发展,数字孪生电厂或者变电站的三维模型越来越精细化,三维模型

的构件数量有些甚至能达到千万个,平均属性个数能达到 40~50 个,在这种级别数据量,关系型数据库难以支撑,或者需要花费很高的成本和牺牲性能,才能勉强实现,所以需要一种新型数据库来解决这个问题。

id	component_id	attr_name	attr_val	desc
1	54	type	ZJM01	(Null)
2	54	nationalStandardCode	16J601	(Null)
3	54	material	不锈钢	材质
4	54	width	2100	(Null)
5	54	height	3000	(Null)
6	58	type	节能GC	(Null)
7	58	nationalStandardCode	11CJ27	(Null)
8	58	material	铝塑复合型材	材质
9	58	width	1800	(Null)
10	58	height	2100	(Null)
11	58	fireRating	5	(Null)

图 2 按行存储的表

Fig. 2 Row-stores table

在电力工程造价管理的过程中,也会使用到大量的复杂多样的数据,并且需要分布式数据库,才能提升工程造价管理中数据分析、处理等方面工作的高效和准确^[3-5]。而关系型数据库的分布式方案成本较高,因为关系型数据库出现的比较早,当时需要处理的数据量很小,所以在设计之初并没有考虑分布式的情况,需要引入一些组件和中间件才能实现分布式部署,且复杂度大,所以同样也需要一种新型数据库来降低分布式部署成本。

2 MongoDB 数据库

2.1 简介

MongoDB 是 NoSQL 类型的文档数据库,是一个基于分布式文件存储的开源文档型数据库系统,是一个介于关系数据库和非关系数据库之间的产品,专为可扩展性、高性能以及高可用性而设计的数据库,是非关系型数据库中功能最丰富且最像关系型数据库的,它支持的数据结构非常松散,是类似 json 的 bson 格式,因此可以存储比较复杂的数据类型^[6-8]。它支持的查询语言非常强大,其语法有点类似于面向对象的查询语言,几乎可以实现类似关系数据库单表查询的绝大部分功能,而且还支持对数据建立索引,并提供了聚合、分片和负载均衡等功能,能够

通过简单地横向扩展来提高读写性能和大规模数据批处理^[9]。

2.2 特点

MongoDB 数据库的层次结构跟关系型数据库类似,其层次关系为:一套数据库管理软件可以包括多个数据库,而每个数据库下又可以包含多个集合,每个集合中又可以包含多个文档^[10-11],如图 3 所示。集合对应关系型数据库的表,一个文档对应关系型数据库的一行数据,如表 1 所示。

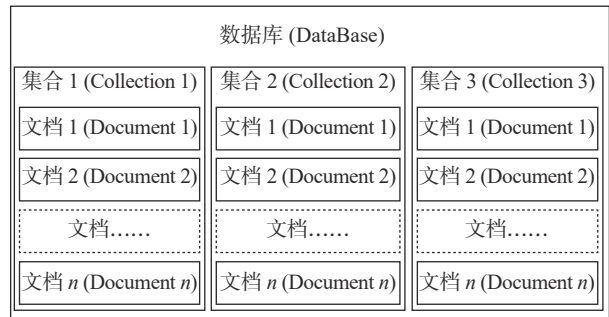


图 3 MongoDB 层次结构

Fig. 3 MongoDB hierarchy

MongoDB 数据库具有如下几个主要优点:

1) 高性能。存储引擎能够高效使用内存,减少 IO 次数,提供高性能的数据持久性,索引支持更快的查询^[12]。

表 1 数据库术语对比

Tab. 1 Database term comparison

MongoDB数据库	关系型数据库
Database(数据库)	Database(数据库)
Collection(集合)	Table(表)
Document(文档)	Row(行)
Field(字段)	Column(列)

2)高可用性。提供了副本集架构,通过类似主从(备)模式,提供自动故障转移和数据冗余。

3)高扩展性。通过数据分片实现方便实现水平扩展,突破单机的存储能力,提升数据的读写效率。

4)强大的查询功能。不仅支持关系型数据库的单表查询功能,还支持强大的聚合计算^[13]。

5)灵活的数据结构。不需要预先定义数据结构和数据类型,支持动态增加文档的字段^[14]。

当然 MongoDB 数据库也有一些局限性,不适合事务一致性很高的场景,如银行核心交易数据,以及其他涉及财务等相关场景,也不适合涉及多表复杂联查的场景^[15]。

3 数据模型设计

MongoDB 数据库不需要像关系型数据库那样必须先定义表结构才能进行数据的存储,并且 MongoDB 数据库同一个集合里面的每一文档都可以有不同的字段,所以 MongoDB 非常适合存储拥有不同属性的数据,如,不同的设备或元件只有部分属性相同,大部分属性是不同的,如图 4、图 5 所示。图中以“etattr_”开头是独有属性,它们有不同属性且属性个数相差很大,图 4 所示元件的是一个窗(一种元件),它拥有的属性只有 11 个,图 5 所示元件的是一个钢管(一种元件),它拥有属性却有 40 个,但它们都属于元件这个集合。图中其他不是以“etattr_”开头属性,是管理这个元件集合的管理属性或固有属性。

虽然 MongoDB 数据库不需要预先定义集合的数据结构,但是直接使用、或者随意使用将会导致数据混乱,如:同一种类型设备或元件的相同属性用不同的数据类型表示,或者采用不同的属性名,将会导致数据标准不统一、后期无法很好地对设备或元件进行查询分析。

为了解决这个问题,系统设计的时候必须为这些设备或元件关联到一个自定义的数据模板,模板

```
{
  "id": 5041280,
  "code": "A61B09C02D09E10F00BH8484",
  "entityUid": "basic_f455bb4870a7bb0275f3b2f12658b29",
  "entityType": 0,
  "parentId": 5033180,
  "classId": 579084,
  "topicName": "4245590/4590845/5033180/5041280",
  "etattr_classificationCode": "A61B09C02D09E10F00",
  "etattr_code": "A61B09C02D09E10F00BH8484",
  "etattr_glass Thickness": " ",
  "etattr_height": "8400.0",
  "etattr_material": "镀锌钢板或不锈钢板",
  "etattr_specsCode": "BH8484",
  "etattr_standardCode1": "E10",
  "etattr_standardCode2": "F00",
  "etattr_standardEncode1": "17J610-1",
  "etattr_type": "PMI",
  "etattr_width": "8400.0"
}
```

图 4 一个集合(窗)

Fig. 4 A set (window)

```
{
  "_id": 1921221,
  "code": "P1SPA11SI457x24",
  "entityUid": "pdms_c23b83a5b2d87703036cd5b86ec825f",
  "entityType": 0,
  "parentId": 1921205,
  "classId": 554706,
  "topicName": "554606/1468432/1920654/1921205/1921221",
  "etattr_BLCOUNT": 0,
  "etattr_BLDia": 0,
  "etattr_Cata_Name": "/JW-CATA",
  "etattr_Cata_Ref": "=21705/119708",
  "etattr_Cata_Seq": 303,
  "etattr_Cata_Type": "CATA",
  "etattr_Cate_Desc": "直管",
  "etattr_Cate_Name": "/JW/A178/TUBE-STD",
  "etattr_Cate_Ref": "=21705/120052",
  "etattr_Cate_Seq": 1,
  "etattr_Cate_Type": "CATE",
  "etattr_Category": "管道",
  "etattr_CategoryDesc": "直管(=21705/120052)",
  "etattr_CmprName": "/GEDI-TUBD-1-3154",
  "etattr_CmprRef": "=13412/137131",
  "etattr_Connection_type": "",
  "etattr_Cweight": 0.3154,
  "etattr_DbRef": "=24575/351",
  "etattr_DetrName": "/GTemplate-TUBE-STD-D4",
  "etattr_DetrRef": "=21609/64030",
  "etattr_Dtxrtext": "无缝钢管 ID457x24 ASTM A 335M",
  "etattr_Itemcode": "P1SPA11SI457x24",
  "etattr_Material": "A335 P92",
  "etattr_MatxName": "/SMTE-A335P92",
  "etattr_MatxRef": "=21609/1732",
  "etattr_Name": "无缝钢管",
  "etattr_ODID": "OD",
  "etattr_PointSetRef": "=21609/280",
  "etattr_Scom_Name": "/JW/A178/TUBE-STD-S:457",
  "etattr_Scom_Ref": "=21705/120083",
  "etattr_Scom_Seq": 16,
  "etattr_Scom_Type": "SCOM",
  "etattr_Sdth": 26.37,
  "etattr_Sect_Desc": "",
  "etattr_Sect_Name": "/JW-TUBE-SECT",
  "etattr_Sect_Ref": "=21705/119709",
  "etattr_Sect_Seq": 1,
  "etattr_Sect_Type": "SECT",
  "etattr_Stand": "ASTM A 335M",
  "etattr_Weight": 315.427
}
```

图 5 一个集合(钢管)

Fig. 5 A set (steel tube)

定义了相同类型的设备或元件的属性名、属性的数据类型,以保证数据标准统一,当然不同的设备或元件有不同的模板。这个模板其实就是元数据,定义数据的数据,如图 6 所示,是管接头(一种元件)的部分自定义属性。

系统设计的时候可以采用树结构模型对数据进行组织管理,树结构是多层次的组织结构,易于理解、容易管理、扩展。实体数据(是客观存在并可相互区别的事物,对现实世界中事物的抽象,可以是元件,可以是设备,也可以是其他东西,取决使用者对它定义)内置了一些固定属性用于维护基本信息,包括:内部唯一标识、外部唯一标识、数据模板标识、编码、父子关系、层级结构,这些属性的作用如表 2 所示。其中 topicName 是较为特殊的属性,记录了当前实体数据在树结构从根节点到该节点的位置信息,也就是这个实体在树结构中的层级信息,它由各个节点的_id 组成,以“/”分隔,并对该属性建立索引。它的作用是在查询分析时,能够快速地对匹配实体树上的某个节点下的实体数据。如需要统计某个节点下的各种设备或元件的数量,就可以使用如下的 SQL 语句进行查询:

```
SELECT class_id,count(*) FROM entity WHERE
topicName like '178/463/%' GROUP BY class_id
```

除了上面固定属性外,其他属性都属于数据模板中自定义的属性,这些属性存储在 MongoDB 中以固定“etattr_”作为前缀,如图 4 或图 5 中以“etattr_”前缀的都属于自定义的属性,这个前缀是系统自动加上,对于使用者是不可见的。系统可以通过开放接口的方式,给其他的应用通过数据模板来自定义具体的实体数据的不同属性,通过数据模板维护实体数据的属性,保证相同设备或元件具有相同的属性名、相同属性类型,避免数据不统一的问题。

相较于关系型数据库,自定义属性除了支持基本数据类型(字符串、整型、浮点型、布尔型、时间类型)之外,还支持基本数据类型数组(Array)、复合数据类型、复合数据类型数组。这些复杂属性的支持在设备树管理中非常有用的,如可以对设备进行分组管理,特别一个设备可以属于多个组的场景,如图 7 所示,分组属性 etattr_tags 是一个字符串数组,这个设备有多个组的标签,对“red”组或者“green”组的设备进行个数统计时,可以使用如下语句:

```
SELECT count(*) FROM entity WHERE etattr_
tags = "red"
```

```
SELECT count(*) FROM entity WHERE etattr_
tags = "green"
```

有些读者可能好奇为什么可以通过 SQL 语句

排序	数据类型	关键字	属性代码	属性名称	属性分类
1	浮点数	PARAM1	主管外径	主管外径	尺寸属性
2	浮点数	PARAM2	分支管外径	分支管外径	尺寸属性
3	浮点数	PARAM3	分支管长度	分支管长度	尺寸属性
4	字符串	PARAM4	CONN TYPE	连接类型	尺寸属性
5	字符串	Name	Name	名称	元件材料属性
6	字符串	Spec	Spec	规格	元件材料属性
7	字符串	Material	Material	材料	元件材料属性
8	字符串	Weight	Weight	重量	元件材料属性
9	字符串	Category	Category	类别	PDMS属性
10	字符串	CategoryDesc	CategoryDesc	类别描述	PDMS属性
11	字符串	Connection_type	Connection_type	连接方式	设计属性
12	字符串	Applicable_medium	Applicable_medium	适用介质	设计属性
13	字符串	Nominalpressure	Nominalpressure	公称压力	设计属性
14	字符串	Applicabletemperature	Applicabletemperature	适用温度	设计属性

图 6 元数据

Fig. 6 Metadata

表 2 固定属性含义
Tab. 2 Fixed property meaning

字段	字段含义
_id	自增主键, 内部唯一标识
code	编码或者名称
entityUuid	外部唯一标识/KKS码, 对接外部系统
entityType	树节点的类型, 管理节点或实体节点
parentId	父节点的主键
classId	数据模板的id
topicName	树上的层次结构

```
{
  "_id": 1094760,
  "code": "1号风机",
  "entityuuid": "bigd_a44cc985a04f4475bdd9a8aa9bca2ab5",
  "entityType": 0,
  "parentId": 1094758,
  "classid": 520886,
  "topicName": "19478/1094758/1094760",
  "etattr workTime": "2022-04-07",
  "etattr lossPower": 5.5,
  "etattr tags": [
    "red",
    "green"
  ]
}
```

图 7 一个集合(风机)
Fig. 7 A set (WTG)

对 MongoDB 数据库进行查询, 因为开发人员或者数据分析人员对 SQL 语言比较熟悉, 系统设计的时候可以提供一个通过 SQL 语句查询的功能, 它能够将常用的单表查询的 SQL 语句解析转换为 MongoDB 数据库的查询语言, 便于相关人员对实体数据的统计分析。

4 结论

基于 MongoDB 的数据模型虽然有很大的灵活性, 但容易引起数据不统一, 系统设计的时候可以采用自定义模板对其属性进行约束, 既保证了在使用过程中的灵活性, 也保证了数据统一, 为用户提供灵活的实体建模功能。

MongoDB 数据库作为 NoSQL 类型的文档型数据库, 跟其他类型 NoSQL 数据库在互联网行业中应用非常广泛, 其中包括电商、游戏、社交、物流等行业, 但在一些传统行业技术栈的使用相对滞后, 通过对 MongoDB 数据库的学习和研究, 结合电力工程行业的数据特点, 引入 MongoDB 数据库作为系统的核心数据库是一个非常好的选择, 特别适合解决部分

场景下关系型数据库的存储效率低、无法灵活建模、数据量大查询效率低、分布式成本高的问题。

参考文献:

- [1] 张计宏. 大数据在工程建设项目全过程投资管控中的应用研究 [J]. 建筑经济, 2022, 43(增刊1): 38-42. DOI: 10.14181/j.cnki.1002-851x.2022S10038.
- [2] 李元琳. 大数据时代工程项目管理新视角 [J]. 科技创新与应用, 2022, 12(23): 153-157. DOI: 10.19981/j.cnki.1002-851x.2022S10038.
- [3] 吴雁, 吴孝华. 大数据时代BIM技术在工程造价管理中的应用探析 [J]. 中国市场, 2022(19): 73-75. DOI: 10.13939/j.cnki.zgsc.2022.19.073.
- [4] 窦晓薇. 基于大数据环境下工程造价的管理对策 [J]. 中华建设, 2022(9): 27-28.
- [5] 杨超, 陈健, 刘光. 基于云平台的电力大数据业务处理技术 [J]. 电气应用, 2022, 41(8): 76-81.
- [6] 郭远威. MongoDB核心原理与实践 [M]. 北京: 电子工业出版社, 2022.
- [7] 王辉, 傅康平, 侯毅, 等. 一种面向MongoDB的数据库统一访问接口设计与实现 [J]. 电子技术与软件工程, 2021(1): 171-173.
- [8] 王萍. 基于MongoDB的外卖订单业务实现 [J]. 信息与电脑, 2021, 33(11): 110-113. DOI: 10.3969/j.issn.1003-9767.2021.11.034.
- [9] 叶思斯, 林志达, 郭献彬, 等. 基于MongoDB的配置管理平台应用研究 [J]. 系统仿真技术, 2021, 17(4): 253-258. DOI: 10.16812/j.cnki.cn31-1945.2021.04.008.

- simulation technology, 2021, 17(4): 253-258. DOI: 10.16812/j.cnki.cn31-1945.2021.04.008.
- [10] 白洁, 武佳丽, 余啟旺, 等. 基于MongoDB的非关系型数据库的设计与应用 [J]. 湖北师范大学学报(自然科学版), 2022, 42(2): 79-82. DOI: 10.3969/j.issn.2096-3149.2022.02.012.
- BAI J, WU J L, YU Q W, et al. Design and application of non relational database based on MongoDB [J]. Journal of Hubei normal university (natural science), 2022, 42(2): 79-82. DOI: 10.3969/j.issn.2096-3149.2022.02.012.
- [11] 刘瑜, 刘胜松. NoSQL数据库入门与实践: 基于MongoDB、Redis [M]. 北京: 中国水利水电出版社, 2018.
- LIU Y, LIU S S. Introduction and practice of NoSQL database: based on MongoDB, Redis [M]. Beijing: China Water & Power Press, 2018.
- [12] 许娜, 耿恒高, 徐传鹏, 等. 基于MongoDB的地震勘探数据管理系统的设计与实现 [J]. 实验室研究与探索, 2022, 41(2): 251-260. DOI: 10.19927/j.cnki.syyt.2022.02.054.
- XU N, GENG H G, XU C P, et al. Design and implementation of seismic exploration data management system based on MongoDB [J]. Research and exploration in laboratory, 2022, 41(2): 251-260. DOI: 10.19927/j.cnki.syyt.2022.02.054.
- [13] 马亮亮, 钟闰禄. 一种基于MongoDB的企业内容管理系统实现 [J]. 电脑编程技巧与维护, 2021(12): 110-112. DOI: 10.16184/j.cnki.comprg.2021.12.040.
- MA L L, ZHONG R L. Implementation of an enterprise content management system based on MongoDB [J]. Computer programming skills & maintenance, 2021(12): 110-112. DOI: 10.16184/j.cnki.comprg.2021.12.040.
- [14] 凌敏, 王骥. 基于Netty和MongoDB的车联网Web系统 [J]. 测绘与空间地理信息, 2021, 44(10): 55-58,62. DOI: 10.3969/j.issn.1672-5867.2021.10.015.
- LING M, WANG J. Internet of vehicles (IOV) Web system based on Netty and MongoDB [J]. Geomatics & spatial information technology, 2021, 44(10): 55-58,62. DOI: 10.3969/j.issn.1672-5867.2021.10.015.
- [15] 张雯杰, 蔡佳玲. MongoDB从入门到商业实战 [M]. 北京: 电子工业出版社, 2019.
- ZHANG W J, CAI J L. MongoDB from entry to business practice [M]. Beijing: Publishing House of Electronics Industry, 2019.

作者简介:



余建忠

余建忠 (通信作者)

1990-, 男, 工程师, 硕士, 主要从事计算机、大数据研究工作(e-mail)yujianzhong@gedi.com.cn。

项目简介:

项目名称 “海上风电场海洋环境立体监测网关键技术及装备产业化”(2021年广东省自然资源厅海洋六大产业专项资金海上风电专题)(GDNRC[2021]37)

承担单位 中国能源建设集团广东省电力设计研究院有限公司、彩虹鱼科技(广东)有限公司、暨南大学、中广核阳江海上风力发电有限公司

项目概述 本项目以海上风电场海洋环境立体监测网关键技术及装备产业化为总体目标,着力研究解决成套设计方案、关键装备研制、大数据集成平台研制、水上与水下无线远距离传输等问题。

主要创新点 (1)形成高效的大数据集成平台;(2)建立“海、陆、空”三位一体的海洋生态环境立体监测体系;(3)开发新型海洋生态圈相关生物监测识别原理和算法模型;(4)构建稳定的海上风电场立体无线专网通信网络;(5)首次对风电场内的环境与生物之间的关系和影响进行原位实验和评估。

(编辑 赵琪)