基于 PostGIS 的遥感数据服务系统检索研究

陈文韬[1,2]，特日根[1,2,3]

(1.长光卫星技术有限公司,吉林 长春 130000;2.吉林省卫星遥感应用技术重点实验室,吉林 长春 130000;
3.中国科学院长春光学精密机械与物理研究所,吉林 长春 130000)

摘　要: 在遥感数据服务系统中,为了快速高效地检索出用户需要的遥感数据信息,常使用 PostgreSQL 开源数据库的空间数据扩展 PostGIS,模拟遥感数据服务系统的使用场景。从建立数据库索引、构建查询语句等方面入手,分析并设计了更高效、更适合遥感数据服务系统的数据库索引建立策略,证明了在遥感数据服务系统中,同时建立 GIST 空间索引和非空间属性索引更有效。

关键词: 遥感数据服务系统; PostgreSQL; PostGIS; 数据库索引; GIST; 空间索引

中图分类号: P237　　**文献标识码:** A　　**文章编号:** 1672-5867(2021)07-0078-06

Research on Retrieval of Remote Sensing Data Service System Based on PostGISs

CHEN Wentao[1,2], TE Rigen[1,2,3]

(1.Changguang Satellite Technology Co., Ltd., Changchun 130000, China; 2.Key Laboratory of Satellite Remote Sensing Application Technology of Jilin Province, Changchun 130000, China; 3.Changchun Institute of Optics, Fine Mechanics, and Physics, Changchun 130000, China)

Abstract: In the remote sensing data service system, in order to quickly and efficiently retrieve the remote sensing data information required by the user, it often simulates the usage scenario of the remote sensing data service system using PostGIS with the spatial data extension of the PostgreSQL open-source database. A simulation experiment is employed to conduct retrieval process in remote sensing data services. The simulation includes analysis of multiple query conditions and multiple query scales. Starting from the establishment of database indexes and query statements, the analysis and design of a more efficient and suitable database index establishment strategy for remote sensing data service system are carried out. It is more effective to establish GIST spatial index and non-spatial attribute index at the same time in the remote sensing data service system.

Key words: remote sensing data service system; PostgreSQL; PostGIS; database index; GIST; spatial index

0 引　言

遥感数据服务系统是通过互联网为用户提供遥感数据查询、浏览和下载等服务的服务系统。同时,随着遥感卫星技术的不断提升,遥感数据量正以几何倍数增加[1-4],而有效地存储和管理海量遥感数据,需要一个适合的空间数据库作为支撑[5-6]。影响空间数据库检索效率的原因包括:数据库在存储空间信息时所选用的属性类型;空间属性和非空间属性的索引方式;空间数据关系的描述方式。

PostgreSQL 是一个开源的关系型数据库,支持大部分 SQL 标准,同时也具有事务、子查询、多版本并行控制、数据完整性检查等特性[7-8]。而仅通过关系数据库的数据类型,不足以满足全部业务需求,通过将复杂的数据类型作为对象放入关系数据库中,并提供索引机制和简单的操作,则可以满足服务的业务需求[9]。PostGIS 是 PostgreSQL 的一个扩展,其遵循 OpenGIS 规范,提供空间对象建立、空间数据索引、空间数据操作函数和空间数据操作符等功能[10-12]。

在遥感数据服务系统中,数据库查询形式可分为属

收稿日期: 2020-08-17

基金项目: 国家重点研发计划(2018YFB1004605);吉林省科技计划——科技创新中心项目(20180623058TC)资助

作者简介: 陈文韬(1994-),男,黑龙江佳木斯人,助理研究员,硕士,2019 年毕业于吉林大学计算机技术专业,主要从事后端研发、遥感大数据等方面的工作。

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

性查询形式和空间关系查询形式，前者不涉及空间对象，而后者涉及空间对象。本文在上述两种查询形式共存的条件下，通过使用 PostGIS，分析并设计了遥感数据服务系统数据库索引的建立策略。

1　PostgreSQL 以及 PostGIS 的使用

1.1　索引的使用

索引有助于更快地获取或更新数据库信息，然而在创建索引过程中，会出现锁表现象，导致数据库写入性能下降。同时，索引的建立也会增加数据库所占的存储空间，因此，需要合理地创建索引。

PostgreSQL 支持的索引类型有: B-Tree、Hash、GIST、SP-GIST、GIN 和 BRIN。其中 B-Tree 索引一般建立在顺序存储的数据列中，如时间、工号等，也可以将多个属性建立成一个复合 B-Tree 索引。而 GIST 索引是一种平衡树结构[13-14]，与 B-Tree 索引相比，GIST 索引更适合多维数据类型和集合数据类型，同时支持更多操作符[15-16]，而且，GIST 索引也更适合构建空间索引。GIST 索引的缺点是创建时间长、占用空间大，因此，在插入操作较为频繁的业务中使用会消耗更多资源。

1.2　PostGIS 空间关系

在 PostGIS 中一般用空间谓词来描述两个空间对象的关系，PostGIS 中的空间谓词函数包括 st_intersects、st_contains、st_crosses、st_disjoint 等。有时这些典型的空间谓词不足以描述空间过滤的条件，就需要使用九交关系模型描述一些复杂的空间关系[17]。根据 OpenGIS Simple Features Implementation Specification for SQL 规范[18]，对给定的几何对象 a，分别用 I(a) 、B(a) 、E(a) 表示 a 的内部、边界、外部，那么两个几何对象 a 和 b 的相交矩阵见表 1。

表 1　相交矩阵

Tab.1　Intersection matrix

	内部	边界	外部
内部	dim(I(a) ∩I(b))	dim(I(a) ∩B(b))	dim(I(a) ∩E(b))
边界	dim(B(a) ∩I(b))	dim(B(a) ∩B(b))	dim(B(a) ∩E(b))
外部	dim(E(a) ∩I(b))	dim(E(a) ∩B(b))	dim(E(a) ∩E(b))

dim(a) 表示 a 的维度，其中 dim(a) ∈{0，1，2，T，F，*}，并且满足式(1) 。

$$\begin{cases}\dim(a)=0\Leftrightarrow a\text{ 是点}\\ \dim(a)=1\Leftrightarrow a\text{ 是线}\\ \dim(a)=T\Leftrightarrow\dim(a)\in\{0,1,2\}\\ \dim(a)=F\Leftrightarrow a\text{ 不存在}\\ \dim(a)=*\Leftrightarrow a\text{ 不考虑}\end{cases}\quad(1)$$

将相交矩阵按照行优先形式转化为一个字符串，该字符串作为 st_relate 函数的第三个参数，可以表示任意两个空间对象间的关系。

由相交矩阵可得出，代表相离(disjoint) 的相交矩阵可以表示为字符串“FF* FF＊＊＊＊”，而相交关系(intersects) 可以表示为“不相离”。

PostGIS 中还提供了一些交集运算符，如 &&、&&& 等，交集运算符可以充分利用空间索引，在空间检索条件构建过程中使用交集运算符可以提高索引的使用效率。

2　遥感信息检索方法

2.1　实验环境与前提

实验使用的操作环境是 64 位 Windows7 操作系统，数据库版本为 64 位 PostgreSQL 9.6.14，PostGIS 版本是2.5.2。

为了精确地计算检索时间，本文将遥感数据以文件形式存储，而在数据库中仅存储遥感数据的路径信息以及查询相关的空间和非空间数据。使用的空间对象是多边形(POLYGON) 对象，坐标系为 WGS84，存储类型为 geometry。实验中可以通过 st_geomfromgeojson 函数将 geojson 描述的几何对象转化为 geometry 对象。

实验中创建表的语句如语句 1 和语句 2 所示。

语句 1

```
create table search_test_table(
id serial PRIMARY KEY,
product_id varchar( 128) ,
satellite varchar( 32) ,
time timestamp,
geo geometry( POLYGON,4326)  not null
) ;
```

语句 2

```
create table condition_test_table(
geojson text PRIMARY KEY,
geo geometry( POLYGON,4326)  not null
) ;
```

实验数据共有 10 718 764 条数据。表 search_test_table 用于存放查询相关的空间和非空间数据，其中 id 为 32 位自增序列主键，product_id 和 satellite 为两个非时间属性数据，分别表示产品号和卫星名称，time 是一个时间属性数据，表示拍摄时间，geo 则是空间属性数据，表示遥感数据的地理位置信息。

表 condition_test_table 则在 geo 字段中存放了额外的空间对象，并以此作为检索条件，为了模拟用户的检索环境，该表中只有一条记录。

2.2　构建查询语句

本文中空间检索使用的查询语句如语句 3 所示。

语句 3

```
SELECT
A.*
FROM
```

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

```
search_test_table AS A JOIN
condition_test_table AS B
ON
st_intersects ( A.geo, B.geo );
```

其功能是联结两表,利用 PostGIS 的空间谓词函数筛选出符合条件的数据。实际查询过程中,condition_test_table 表可以由输入空间对象参数取代,并用 WHERE 语句连接由空间谓词函数组成的条件。同时可以使用 st_intersects、st_contains、st_crosses 等空间谓词函数,支撑不同业务场景的检索需要,对于更为复杂的场景也可以直接使用 st_relate 函数。

本文以 st_intersects 为例,探索 SQL 语句在检索中的合理构建方式。在构建查询语句之前,为了保证查询的基本效率,要对 search_test_table 表建立索引。首先使用语句 4 构建 geo 列中 GIST 空间索引,并将 search_test_table_geo_idx 命名为索引 1,其创建时间是 187.270 s,占用空间大小是 940 MB。

语句 4

```
CREATE INDEX search_test_table_geo_idx
ON search_test_table
USING GIST ( geo);
```

PostgreSQL 所有索引的使用都由优化器统一完成,同时,使用 EXPLAIN 命令可以获取优化器给出的查询计划,并得到检索过程中索引的使用方式。

如果在只建立索引 1 的条件下执行语句 3,数据库优化器在查询过程中会使用索引 1 进行检索。其中 st_intersects 函数的检索过程中会通过自动添加 A.geo&&B.geo 交集运算来加速检索效率,而使用 st_relate 函数则不会自动添加交集运算。

根据 2.2 节可知,st_intersects(A.geo, B.geo) 与 not st_relate (A.geo, B.geo,'FF* FF****') 等价,因此,通过实验对这两种方式执行过程中的执行计划和执行时间进行统计,其执行语句分别为语句 5 和语句 6,其结果如图 1 和表 2 所示。

语句 5

```
SELECT
A.*
FROM
search_test_table AS A JOIN
condition_test_table AS B
ON
not st_relate ( A.geo, B.geo,'FF* FF****');
```

语句 6

```
SELECT
A.*
FROM
search_test_table AS A JOIN
condition_test_table AS B
ON
A.geo && B.geo AND
not st_relate ( A.geo, B.geo,'FF* FF****');
```

在分析查询时间时,第一次查询会产生数据库结果缓存,缓存对查询效率有很大的提升。所以实验中每个查询都清理三次缓存,并在每次清理缓存后再查询三次,求出每次清理缓存后第一次查询的平均时间、第二次查询的平均时间以及第三次查询的平均时间,时间单位为 s。

由表 2 可以看出语句 5 在查询计划中没有使用空间索引,查询效率很低,st_relate 与交集运算符并用的语句 6 检索使用了空间索引,但检索效率仍然不如语句 3。

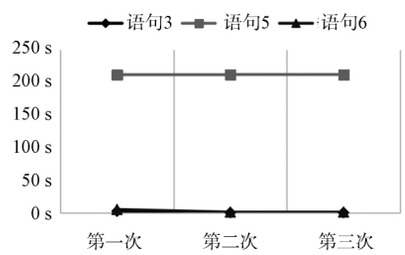

图 1 使用 st_intersects 函数、st_relate 函数的查询时间

Fig.1 Query time of using st_intersects and st_relate functions

表 2 使用 st_intersects 函数、st_relate 函数的查询计划和查询时间

Tab.2 Query plan and time of using st_intersects and st_relate functions

查询语句	使用索引	结果条数	第一次	第二次	第三次
语句 3	索引 1	10 089	2.229	0.096	0.093
语句 5	无	10 089	211.771	212.043	211.653
语句 6	索引 1	10 089	4.857	0.827	0.843

2.3 建立索引

在实际检索过程中,查询条件不仅包括对空间属性的筛选,还包括对时间属性等其他非空间属性的筛选和排序。因此,设计实验,通过语句 7 和语句 8 分别以卫星名称和拍摄时间两个属性建立索引 2 和索引 3,并通过语句 9、语句 10 及语句 11 测试其检索效率。其中,索引 2 单次创建时间是 26.730 s,占用空间大小是 230 MB,索引 3 单次创建时间是 28.909 s,占用空间大小是 238 MB。

语句 7

```
CREATE INDEX search_test_table_satellite_idx
ON search_test_table USING btree (
satellite
);
```

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

语句 8

```
CREATE INDEX search_test_table_time_idx
ON search_test_table USING btree (
time DESC NULLS LAST
);
```

语句 9

```
SELECT
A.*
FROM
search_test_table AS A JOIN
condition_test_table AS B
ON
st_intersects ( A.geo, B.geo )
WHERE
satellite = 'JL101A' AND
time > '2016-01-01'
ORDER BY
time DESC;
```

语句 10

```
SELECT
A.*
FROM
search_test_table AS A,
condition_test_table AS B
WHERE
satellite = 'JL101A';
```

语句 11

```
SELECT
A.*
FROM
search_test_table AS A ,
condition_test_table AS B
WHERE
time > '2019-01-01'
ORDER BY
time DESC;
```

语句 12

```
SELECT
A.*
FROM
search_test_table AS A ,
condition_test_table AS B
WHERE
time > '2016-01-01'
ORDER BY
time DESC;
```

如果改变 condition_test_table **表中的** geo **空间对象，会导致查询产生不同规模的查询结果，三种不同空间对象的** geojson **表达式见表** 3。**在后续实验中会选择不同结构序号的空间对象构建查询条件来表示不同的查询规模，这些结构不会影响非空间查询的查询规模。**

表 3　3 种不同空间对象的 geojson 表达式

Tab.3　Geojson expression of three different spatial objects

结构序号	geojson
1	{"type":"Polygon","coordinates":[[[125.5078125,44.087585028245165],[125.5078125,43.83452678223683],[125.15625,43.83452678223683],[125.15625,44.087585028245165],[125.5078125,44.087585028245165]]]}
2	{"type":"Polygon","coordinates":[[[123.530273,46.407564],[121.552734,45.79817],[122.255859,44.056012],[125.947266,41.14557],[130.03418,42.55308],[131.132813,44.245199],[123.530273,46.407564]]]}
3	{"type":"Polygon","coordinates":[[[122.46108,51.629511],[88.346421,37.459277],[114.020339,25.190952],[129.143332,45.351646],[122.46108,51.629511]]]}

在同时存在多个索引的情况下，优化器会为不同的查询语句或不同的查询规模给出不同的查询计划，如语句 9 **在** 3 **种不同规模查询条件下，优化器均只使用了索引** 1。**而针对语句** 10 **和语句** 11，**优化器则使用不同的索引。同时，因语句** 12 **查询结果条数过多而不使用任何索引，其具体查询计划和查询时间如图** 2 **和表** 4 **所示。同时由表** 4 **可知，查询结果的规模越大，时间开销就越大。**

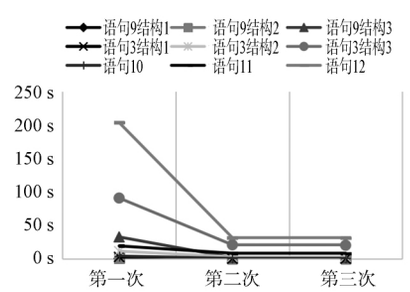

图 2　使用多个索引的查询时间

Fig.2　Query time of using multiple indexes

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net

表 4 使用多个索引的查询计划和查询时间

Tab.4 Query plan and time of using multiple indexes

查询语句	结构序号	使用索引	结果条数	第一次	第二次	第三次
语句 9	1	索引 1	187	0.165	0.026	0.017
语句 9	2	索引 1	371 6	0.853	0.220	0.228
语句 9	3	索引 1	14 486	32.509	1.577	1.547
语句 3	1	索引 1	10 089	2.560	0.121	0.096
语句 3	2	索引 1	157 819	11.256	2.657	2.650
语句 3	3	索引 1	1 052 569	90.796	20.693	20.035
语句 10	无影响	索引 2	125 757	3.817	0.505	0.515
语句 11	无影响	索引 3	1 189 656	19.064	8.114	7.686
语句 12	无影响	无	5 474 654	34.919	31.401	30.988

如果通过语句 13 建立一个 B-Tree 多值索引并命名为索引 4，则索引 4 的属性是有顺序的，在索引 4 的三列属性中，satellite 是第一列属性。索引 4 单次创建时间是 93.028 s，占用空间大小是 1 767 MB。使用语句 9 和索引 4 的具体查询计划和查询时间如图 3 和表 5 所示。

语句 13

```
CREATE INDEX search_test_table_complex_idx
ON search_test_table USING btree (
satellite,
time DESC NULLS LAST,
geo
);
```

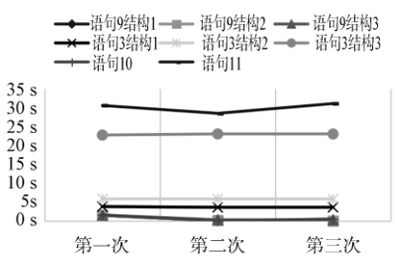

图 3 使用索引 4 的查询时间

Fig.3 Query time of using index 4

表 5 使用索引 4 的查询计划和查询时间

Tab.5 Query plan and time of using index 4

查询语句	结构序号	使用索引	结果条数	第一次	第二次	第三次
语句 9	1	索引 4	187	1.449	0.076	0.061
语句 9	2	索引 4	3 716	1.448	0.127	0.103
语句 9	3	索引 4	14 486	1.558	0.331	0.293
语句 3	1	无	10 089	3.809	3.608	3.632
语句 3	2	无	157 819	5.848	5.839	5.809
语句 3	3	无	1 052 569	22.687	22.980	22.967
语句 10	无影响	索引 4	125 757	1.604	0.573	0.559
语句 11	无影响	无	1 189 656	30.484	28.410	30.996

表 5 结合表 4 中结果表明，在查询条件包含索引第一列属性时，优化器会使用 B-Tree 多值索引进行检索。当第一列属性是如卫星名称的散列值时，在检索结果数较多的情况下，使用 B-Tree 多值索引的检索效率比使用空间索引速度更快。如果查询条件中不包含第一列属性，优化器将会使用全表扫描。综上所述 B-Tree 多值索引在使用过程中对查询条件有很大的限制，只有查询条件包含索引第一列属性时，优化器才会给出使用 B-Tree 多值索引的查询计划。

如果使用语句 14 建立 GIST 多值索引并命名为索引 5，索引 5 单次创建时间为 1 145.252 s，占用空间大小约 1 031 MB，使用索引 5 的查询计划和查询时间如图 4 和表 6 所示。

语句 14

```
CREATE INDEX search_test_table_complex_gist_idx
ON search_test_table USING btree (
satellite,
time,
geo
);
```

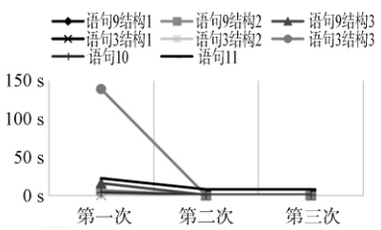

图 4 使用索引 5 的查询时间

Fig.4 Query time of using index 5

表 6 使用索引 5 的查询计划和查询时间

Tab.6 Query plan and time of using index 5

查询语句	结构序号	使用索引	结果条数	第一次	第二次	第三次
语句 9	1	索引 5	187	3.309	0.056	0.031
语句 9	2	索引 5	3716	6.442	0.079	0.095
语句 9	3	索引 5	14 486	15.713	0.288	0.284
语句 3	1	索引 5	10 089	55.484	0.571	0.535
语句 3	2	索引 5	157 819	91.557	3.295	3.312
语句 3	3	索引 5	1 052 569	139.122	21.089	20.693
语句 10	无影响	索引 5	125 757	3.828	0.520	0.522
语句 11	无影响	索引 5	1 189 656	22.168	7.690	7.512

表 6 表明使用 GSIT 多值索引时，只要条件中包含索引中存在的列，优化器就会使用索引进行检索。

综上所述，GIST 多值索引在单个条件的查询中检索性能不如单值索引，而在检索结果集较大且查询中包含多个条件的情况下效率优于单值索引。

3 结束语

由上述实验可知，若表中某一属性的值较为分散，且需要频繁使用，则需建立针对该属性的 B-Tree 索引。同时，在满足业务需要的情况下，查询条件的设计应避免产生大量结果来保证检索效率。而且检索结果的数据传输也会影响缓存的使用，在构建遥感信息服务过程中应该给用户提供范围更明确的查询条件。B-Tree 多值索引对检索效率优化有限，且依赖固定模式的检索条件。在检索条件不合适的情况下优化器就会给出全表扫描计划，而 GIST 多值索引使用条件更宽泛，同时比 B-Tree 多值索

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved. http://www.cnki.net

引占用空间更小。但其建立过程消耗时间更多,还会降低插入操作的效率,并且检索效率只会在特定条件下有提升,因此,在遥感数据服务系统中应以建立单值索引为主,以 GIST 多值索引为辅。

参考文献:

[1] ZHANG W,WANG L,LIU D,et al.Towards building a multi-datacenter infrastructure for massive remote sensing image processing [J].Concurrency and Computation,2013,25(12):1 798-1 812.

[2] KUCUK A,HAMDI S M,AYDIN B,et al. PG-TRAJECTORY: A PostgreSQL/PostGIS based data model for spatiotemporal trajectories [C] // IEEE International Conferences on Big Data and Cloud Computing, New York: IEEE,2016:81-88.

[3] 戴芹,刘建波,刘士彬.海量卫星遥感数据共享的关键技术 [J].计算机工程,2008,34(6):283-285.

[4] WANG L,MA Y,ZOMAYA A Y,et al.A parallel file system with application-aware data layout policies for massive remote sensing image processing in digital earth [J].IEEE Transactions on Parallel and Distributed Systems,2015,26(6):1 497-1 508.

[5] 曹学诚,刘周周.基于 PostGIS 的高分一号卫星影像管理与 Web 应用研究 [J].遥感技术与应用,2016,31(2):285-289.

[6] WANG B,LU X,ZHENG X,et al.Semantic descriptions of high-resolution remote sensing images [J].IEEE Geoence and Remote Sensing Letters,2019,16(8):1 274-1 278.

[7] 盛凯,刘忠,周德超.基于 PostGIS 的历史航迹重演分析系统设计与开发 [J].海军工程大学学报,2017,29(5):108-112.

[8] 赵芸,黄丙湖,吕瑞,等.基于 Web 的农情遥感信息服务系统设计 [J].湖北农业科学,2019,58(4):78-81.

[9] 王密,龚健雅.基于扩展关系数据库的遥感影像数据库管理系统的研究与实现 [J].测绘信息与工程,2002(5):1-3.

[10] 林媛媛.基于 PostgreSQL 与 PostGIS 的空间数据库设计及应用研究 [D].南昌: 江西理工大学,2014.

[11] 梁明,罗荣,胡最.基于 Lucene 和 PostGIS 的地图搜索研究 [J].测绘通报,2014(11):42-45.

[12] 钟远军,李照,林澍哲,等.基于 PostGIS 的地名数据库设计与应用研究 [J].测绘与空间地理信息,2011,34(3):100-103.

[13] EITABAKH M Y,ELTARRAS R,AREF W G. Space-partitioning trees in PostgreSQL: realization and performance [C].Proceedings of the 22nd International Conference on Data Engineering, New York: IEEE,2006.

[14] 高艳,胡启平.数据库通用索引树 GIST 的研究 [J].计算机应用,2003(6):66-68.

[15] FLEITES F C,CHEN S C.Efficient content-based multimedia retrieval using novel indexing structure in PostgreSQL [C] // IEEE International Symposium on Multimedia,New York: IEEE,2013.

[16] 郭龙江,李建中.空间数据库的索引技术 [J].黑龙江大学自然科学学报,2005(3):288-293.

[17] SHEN J,ZHOU T,CHEN M.A 27-intersection model for representing detailed topological relations between spatial objects in two-dimensional space [J].ISPRS International Journal of Geo Information,2017,6(2):37.

[18] 钟志农,景宁,李军.空间查询语言中拓扑关系的定义 [J].国防科技大学学报,2003(5):58-62.

[19] 蔡维华,马乐,王华,等.基于 IGES 文件的舰船航行性能数值仿真模型 [J].计算机应用与软件,2012,29(2):192 - 194,236.

[编辑: 刘莉鑫]

(上接第 77 页)

[15] 周连章,黄荣辉.中国西北干旱、半干旱区感热的年代际变化特征及其与中国夏季降水的关系 [J].大气科学,2008,32(6):1 276-1 288.

[16] MCHEE T B,DOESKEN N J,HLEIST J.The relationship of drought frequency and duration to time scales [C] // Proceedings of the 8th Conference on Applied Climatology. Boston: American Meteorological Society,1993:179-183.

[17] GUNMAN N B.Comparing the pahner drought index and the standardized precipitation index [J].Journal of the American Water Resources Association, 1998, 34(1):113-121.

[18] 李树岩,刘荣花,马志红.基于降水距平的黄淮平原夏玉米干旱评估指标研究 [J].干旱地区农业研究,2012,30(3):252-251.

[19] 张调风,张勃,刘秀丽,等.基于 CI 指数的甘肃省黄土高原地区气象干旱的变化趋势分析 [J].冰川冻土,2012,34(5):1 076-1 083.

[20] 周丹,张勃,任培贵,等.基于标准化降水蒸散指数的陕西省近 50 a 干旱特征分析 [J].自然资源学报,2014,29(4):677-688.

[21] 马永强.宁东能源化工基地生态承载力评价及预警研究 [D].银川: 宁夏大学,2019.

[22] 马斌,徐志友,卜崇德,等.宁东能源化工基地水土保持生态环境动态监测分析 [J].中国水利,2011(14):53-56.

[23] 罗成科,毕江涛,肖国举,等.宁东基地不同工业园区周边土壤重金属污染特征及其评价 [J].生态环境学报,2017,26(7):1 221-1 227.

[24] 李风军,冯晓秀,陆桂琴,等.宁东能源化工基地生态环境脆弱性评价研 [J].生态科学,2014,33(5):1 017-1 022.

[25] 秦丽杰,袁帅.四平市降水量与农业干旱研究 [J].东北师大学报(自然科学版),2005,37(3):99-102.

[26] 卫捷,马柱国.Palmer 干旱指数、地表湿润指数与降水距平的比较 [J].地理学报,2003,58(S1):117-124.

[27] 曾波.湖南省农业干旱空间分布研究 [D].长沙: 中南林业科技大学,2014.

[28] 刘昭媛.宁东基地生态文明建设可持续发展路径研究 [D].银川: 宁夏大学,2015.

[编辑: 刘莉鑫]

(C)1994-2021 China Academic Journal Electronic Publishing House. All rights reserved.　http://www.cnki.net