

# 电网图形数据管理 MongoDB 数据库的应用<sup>①</sup>

赵 越<sup>1</sup>, 李 培<sup>1</sup>, 王 震<sup>2</sup>, 张声圳<sup>2</sup>

<sup>1</sup>(国网江苏省电力公司扬州供电公司, 扬州 225000)

<sup>2</sup>(厦门亿力吉奥信息科技有限公司, 厦门 361008)

**摘 要:** 国家电网公司正推进资产全寿命周期管理体系建设, 电网 GIS 图形作为电网的信息化表征, 为了实现电网异动信息的全过程管理, 通过电网图形多时态多级管理机制, 同时引入 MongoDB 非关系型数据库, 提升了数据读写效率, 最终满足电网各业务部门海量数据信息化应用的性能需求。

**关键词:** 电网图形; 多时态多级; 分布式及分片; MongoDB

## Management Mechanism of Power Grid Graphics Based on MongoDB

ZHAO Yue<sup>1</sup>, LI Pei<sup>1</sup>, WANG Zhen<sup>2</sup>, ZHANG Sheng-Zhen<sup>2</sup>

<sup>1</sup>(State Grid Yangzhou Power Supply Company, Yangzhou 225000, China)

<sup>2</sup>(Xiamen Great Power GEO Information Technology Co. Ltd., Xiamen 361008, China)

**Abstract:** State Grid Corp is promoting the construction of the whole life cycle management system. It takes Grid GIS graphics as the information of the power grid. In order to achieve the whole process management of the grid's transaction information, multi temporal and multilevel management mechanism of power grid graphics based on MongoDB non relational database is introduced. The efficiency of data reading and writing is improved. Ultimately it meets the requirements of mass data information applications in the power grid.

**Key words:** grid graphics; multi temporal and multilevel; distributed and slicing; MongoDB

随着电网公司信息化建设的推进, 电网 GIS 作为电网设备资源管理的平台, 目前已实现了全电压等级的设备的信息录入, 大网省的设备数量以亿计; 同时其作为一线人员工作业务支撑平台, 并发量大, 业务操作复杂. 传统的关系型数据库难以提供超大规模的数据存储以及高并发的读写访问能力, 以至于随着图形数量等级的上升, 系统已无法及时响应高并发的复杂业务操作.

在此背景下, 亟需采取新的技术手段来提升海量数据大并发读写的效率. 非关系型数据库存储的广义定义, 不需要固定的表结构, 使其在大数据量高并发的读写操作有很高的效率. 除此之外, 一线业务人员对相关电网专题图的管理要求不一, 同时为了实现图形版本管理融入时态、空间、人物等基本特征, 构建多时态、多级数据管理模式, 实现电网图形从规划、设计、运行整个过程的实时跟踪, 满足各业务部门甚

至不同业务人员特定图形应用需求, 实现电网的全过程管理.

为了提高海量图形数据的处理效率, 本文提出基于 MongoDB 数据库进行电网图形的多时态多级分布式存储机制, 实现了电网从规划、设计、投运的全过程动态实时跟踪, 并根据业务部门及人员的不同图形需求实现多级管理, 同时利用 MongoDB 非关系型数据库大数据量高并发读取效率高的特点, 大大提升了图形应用效果.

### 1 多时态多级图形管理机制

空间、属性和时间是 GIS 的三个基本特征, 同时是其数据的三种基本数据成分. 传统 GIS 系统记录的只是不断变化的实际的某一“瞬间”特性. 当数据随时间变化时, 我们用新数据代替旧数据, 系统又称为世界的另一个“瞬间”<sup>[1]</sup>. 这种模式忽略了其时间特性, 但

<sup>①</sup> 收稿时间:2015-09-28;收到修改稿时间:2016-06-25 [doi: 10.15888/j.cnki.csa.005312]

是在另外一些领域，如电网领域其状态随时间的紧密关联，不同时刻电网所处的状态千变万化，采用传统电网图形管理方式仅仅是从版本角度进行管理，电网图形发生了变动即增加一个相应的图形版本，并未实现多维度上的时态化管理，为了满足国家电网公司多业务部门的应用需求，实现电网图形的全过程管理。本文提出多时态多级数据管理方式。

电网图形从规划设计、基建、运行的整个过程实时跟踪，记录不同时刻的电网状态变化，满足了各业务部门甚至不同业务人员特定图形应用需求，并实现了整个电网异动的全过程管理。具体管理机制如下：

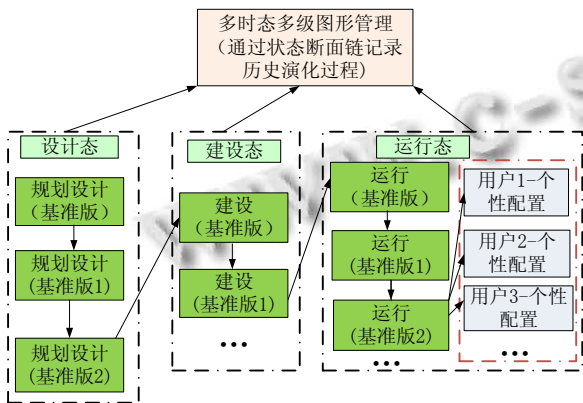


图1 多时态多级管理机制

根据电网的设计、建设和运行分别对应一个时态集，每个时态集合分别有一个基准线，在基准版本上进行迭代更新演化，同时只支持时态的推进不支持回退，即规划设计的最后一个基准版本演化成建设的基准版，建设的最后一个基准版本演化成运行态的基准版。同时在同一时态内，如在设计态基准线中，基准版与基准版1和基准版2的转化方式如下所示：

基准版本  $i+1$  只能基于基准版  $i$  的基础上进行“增量”滚动递归更新，同时记录每次基准版更新的所有记录以及修改人员。

同时，在运行态支持不同业务人员对图形版本的个性化设置，实现多级的管理方式。该展示方式是基于当前运行基准版本进行个性化选择展示的。

然而，随着电网的不断发展与壮大，电网设备数量特别是配网设备呈指数级增长，如江苏电网的图形数据量以亿计，如何在高效快速地实现图形相关功能满足用户的体验需求是亟需解决的问题。故本文进行了非结构化数据库技术的应用探索。

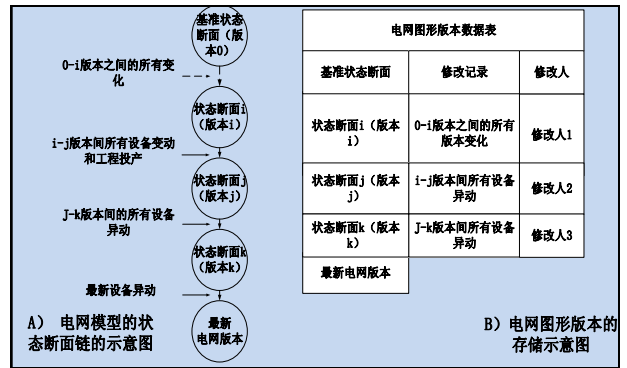


图2 同一时态下图形版本转化

## 2 基于MongoDB的分布式存储设计

### 2.1 非关系型数据库

传统关系型数据库具有高稳定性、使用简单，功能强大等优点<sup>[2]</sup>，随着智能电网的建设，电网数据以亿计，随着“大数据”技术的应用不断深入，对海量数据的分析与应用需求不断加深，然而传统的关系型数据库难以提供超大规模的数据存储以及高并发高速读写访问能力。

随着技术的发展，NoSQL 数据库应运而生，其不需要固定的表结构，通常也不存在连接操作。采用 key-Value 存储、文档型、xml 等方式存储数据模型。其具有易扩展性、大数据量高性能、灵活的数据模型等优点。而 MongoDB 作为非关系型数据库的典型代表，其支持自动分片且可以水平方向的数据库集群；同时支持丰富的查询表达式，可支持查询文档中内嵌的对象及数组，该特性在拓扑追溯分析时非常重要；其还具备全索引等优点。本项目选取 MongoDB 作为应用的数据库。

### 2.2 架构设计

分片是指将数据拆分，将其分散存在不同的机器上的过程，其作为 MongoDB 的扩展方式，其基本思想就是将集合切分成小块，再将这些块分散到若干片里面，每个片只负责总数据的一部分<sup>[3]</sup>。再通过路由进程 Mongos 记录数据的存储位置，应用通过连接它来正常发送请求。本文基于这种思想设计图形的多级多时态的分布式存储框架。

#### 2.2.1 分布式存储框架

① 服务器上分别部署 9 个 Mongo 节点 3 个配置服务节点(ConfigSvr)1 个路由节点 Mongos。

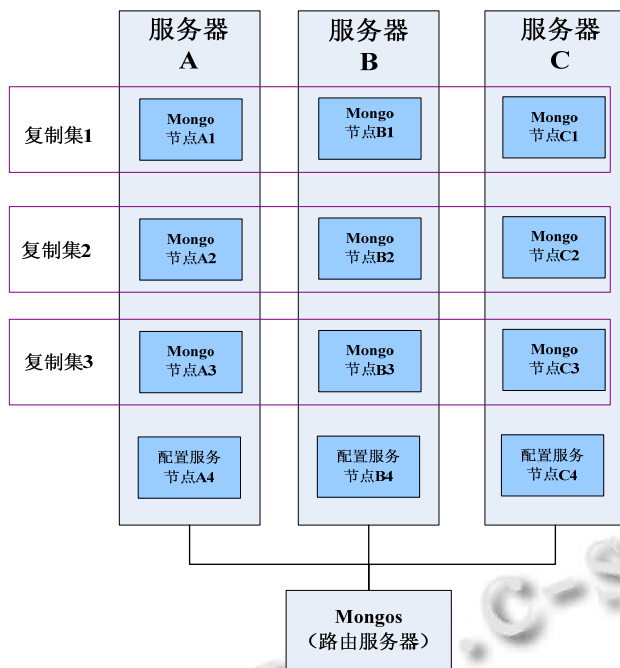


图 3 基于 Mongo 分布式存储框架图

② 不同服务器上面的节点 A1、B1、C1 构建成一个复制集 1 (Replica Sets)。复制集中的节点 A1、B1、C1 的存放的数据是相同的，能够相互复制，异步同步。构建复制集的作用是防止一个服务器或者一个节点不能工作导致数据丢失。

③ 配置服务器是用于构建记录分片(Shards)存储信息。数据集按照数据集数据范围或者使用哈希值进行数据分片存储。分片信息记录在配置服务器上，所有的配置服务器共享这些信息。

④ Mongos 就是 MongoDB 中的路由器进程，它路由所有请求，然后将结果聚合，其本身不存储数据或者配置信息(但会缓存配置服务器的信息)。

### 2.2.2 分片存储方式

由于基于范围的分片方式具有高效的范围查询，但是会导致数据在不同分片上的不均衡；基于哈希的分片方式使得所有分片数据分布均衡，但是查询时候会访问所有的分片，影响查询效率。本发明对数据的读写效率要求较高，故采取基于范围的分片方式。本项目的分配存储方式如下：

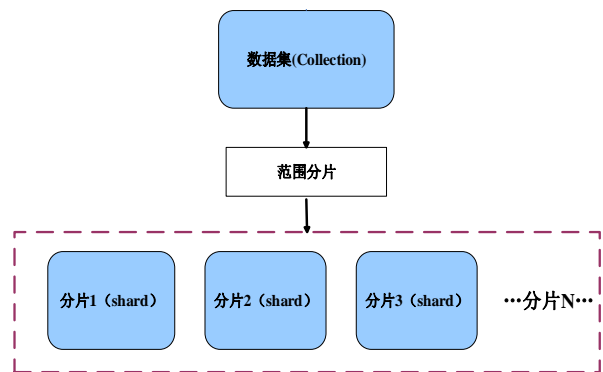


图 4 分片存储架构

## 2.3 存储设计

本管理机制主要涉及版本数据管理、图档数据结构以及设备关系结构三个大的方面，下面重点介绍这些数据的存储设计。

### 2.3.1 版本数据结构设计

通过实现对版本的递归转化以及同时态的多级管理，实现不同人员对专题图的不同需求以及版本的全过程管理。结构形式如下：

```

{
  TreeId //树节点编号，数字
  Code //节点对应编码(如馈线 ID)，字符串
  VersionId //版本编码，字符串
  VersionName //版本名称，字符串
  VersionType //版本类型，字符串
  VersionCode //版本编码，字符串
  UserName //版本更新人，字符串
  CreateTime //版本创建时间，字符串
  EditTime //版本更新时间，字符串
  DataSource //更新来源(PMS,GIS)，字符串
  SchematicType //图纸类型，字符串
  VersionState //版本状态，图档状态
  CoordinateOrigin //坐标原点，嵌套
{
  x //坐标点，数字
  y //坐标点，数字
}
}
    
```

下面以规划人员绘制的一个单线图版本为例，具体如下：

```

{
    
```

```
VersionId:"10000001",
  TreeId:"10102010101",
  Code:"321002001001",
  VersionType:"规划图纸",--规划图纸, 设计图纸,
运行图纸
  VersionName:"规划图纸单线图未发布版本 v003",
  VersionCode:"v003",
  UserName:"规划人员",
  CreateTime:"2015-07-29 09:04:31",
  EditTime:"2015-07-29 09:04:31",
  DataSource:"PMS",--PMS,GIS
  SchematicType:"单线图",--单线图, 网络图, 沿布
图
  VersionState:"未发布",

CoordinateOrigin:[{x:10.23423424,y:111:2423442343}]
}
```

### 2.3.2 图档数据结构设计

图档数据存储结构是存储机制的一个重点, 具体存储结构如下:

```
{
  VersionId //版本编号, 字符串
  GisFeatureType //设备类型(点, 线, 面, 标注),
字符串
  Fid //设备唯一编码, 字符串
  Fno //设备类型编码, 字符串
  FeatureFid //设备台账 ID, 字符串
  FeederFid //设备所属馈线 FID, 字符串
  OwnerFeatureFid //设备从属设备 FID, 字
符串
  IsContact //是否联络设备, 布尔值
  ModifyStatus //设备修改状态, 字符串
  RoleId
  ShapeComponents:[ // 设备 组 件 , 嵌 套
ShapeComponent
  {
    GisFeatureFid //设备 FID, 字符串
    GisFeatureFno //设备 FNO, 字符串
    Angle //角度, 数字
    SymbolSid //组件样式 ID,数字
    SchematicPoints:[ //坐标集, 嵌套
```

```
{x:10.23423424,y:111:2423442343},
{x:12.23423424,y:114:2423442343}
],
ShapeComponentType //组件类型, 字符串
}
],
LabelComponents:[// 设备 标 注 , 嵌 套
LabelComponent
{
  GisFeatureFid: //设备 FID, 字符串
  GisFeatureFno://设备 FNO, 字符串
  Angle: //角度, 数字
  SymbolSid: //组件样式 ID,数字
  SchematicPoints:[//坐标集, 嵌套 point
{x:10.23423424,y:111:2423442343},
{x:12.23423424,y:114:2423442343}
],
ShapeComponentType //组件类型, 字符串
SymbolText//标注内容, 字符串
TextStyle:{ //标注样式, 嵌套 TextStyle
  FontFamily //字体名称, 字符串
  FontSize //字体大小, 数字
  Fill //字体颜色, 字符串
}
}
],
GisAttribute:{ GisAttribute//设备的功能属性, 嵌套
  Name //功能位置名称, 字符串
  ShortName//短名称(杆塔), 字符串
  KGBH //运行编号, 字符串
  Length //长度, 数字
  EleStatus //带点状态, 字符串
  Basevoltage//基级电压, 字符串
  Zone //所属区域, 字符串
  Normalopen //开关状态, 字符串
  PoleId //所属杆塔 ID, 字符串
  YYHH //字符串
  Rates //字符串
  Pfixed //电压等级
  UserType //字符串
  OperatingCapacity //工作效率
```

```

        LineModel //线模型
        LineLength //线长
    }
}
2.3.3 设备关系存储结构
{
    Fid //设备 ID, 字符串
    RelationShipFid//拓扑对应的设备 ID, 字符串
    VersionId //版本 ID, 字符串
    RelationType//包含关系(连接关系), 字符串
},
{
    PrimaryFid //主设备 Fid, 字符串
    ThroughFid //经过的设备, 字符串
    AccessoryFid //附属设备, 字符串
    RelationType//包含关系(包含关系), 字符串
    VersionId //版本 ID, 字符串
}

```

通过以上的存储设计,同时通过设计、建设和运行态的管理机制,实现图形多级多时态管理。

### 3 与关系型数据库效率对比

基于本存储机制进行存储效率验证,通过基于 MongoDB 数据库的电网图形多时态多级的分布式存储机制进行千万级数据效率测试,同时对比传统关系型数据库 Oracle 的读写效率。

本次对比验证是基于以下条件:

- ① 基于 27000000 数量级数据,由于存储结构形式不同,数量上略有百条数据量的差别;
- ② 分页查询、插入数据以及指定查询条件相同,同时查询结果也需相同。

分别进行 1、10、100、500、1000、5000、10000、20000、30000、50000 次的分页查询、插入数据、指定查询操作,并对效率进行记录,具体结果如表 1。

表 1 读写操作效率记录表

数据总量	执行动作	执行次数	Mongo 测试结果(毫秒)	Oracle 测试结果(毫秒)
27000000(数量级)	分页查询	1	1	68
		10	2	102
		100	4	181
		500	15	912
		1000	28	1587
		5000	124	9242
		10000	289	19071
		20000	598	41287
		30000	926	73436
		50000	1236	144482
	插入	1	6	7

数据	10	10	64
	100	30	442
	500	73	1649
	1000	122	3621
	5000	644	16033
	10000	1238	34011
	20000	2493	80023
	30000	5100	126446
	50000	9064	时间远大于 5 分钟
指定查询	1	7	788
	10	2	463
	100	4	192
	500	15	853
	1000	25	1661
	5000	120	10209
	10000	271	
	20000	560	时间远大于 5 分钟
	30000	721	分钟
	50000	1398	

由以上验证结果表可知,在千万级别以上的数据量,基于 MongoDB 的分布式存储机制的分页查询、数据插入和指定条件查询的效率远高于 Oracle 数据,有的甚至提升了几十倍.由此得知,基于 MongoDB 数据库的电网图形多时态多级分布式存储机制可提升电网 GIS 的应用效率和效果。

### 4 结语

国家电网公司不断加强电网设备状态检修管理,实现设备管理向电网管理,不断推进“多时态的统一电网”<sup>[4]</sup>,但基于传统的关系型数据库在性能上有一定的效率瓶颈.本文基于 MongoDB 非关系型数据库的实现多时态多级的图形管理机制,同时与传统的关系型数据库的效率进行对比,结果表明非结构化数据库 MongoDB 在提升海量图形数据的处理效率上有很高的应用价值,同时可将其应用在其它领域.本文通过图形数据管理的 Mongo 应用提升专题图图形版本管理和应用效果,对支撑调度、营销和运检的实际业务有重大意义。

#### 参考文献

- 1 刘茂华.时态-基础地理信息数据库的版本化管理[硕士学位论文].阜新:辽宁工程技术大学,2005.
- 2 张华强.关系型数据库与 NoSQL 数据库.电脑知识与技术,2011,41(20):25-27.
- 3 霍多罗夫,迪洛尔夫,程显峰.MongoDB 权威指南.北京:北京邮电出版社,2011.
- 4 吴淑玮.多时态统一电网模型.计算机系统应用,2015,24(2): 244-247.