

一种高性能多模式的内存数据库系统

陆鑫 王艳蓉 孙超 史豪杰

(南京南瑞继保工程技术有限公司 江苏 南京 211102)

摘要 随着智能电网和工业物联网的蓬勃发展,针对单一模式的数据库不能满足所有需求的问题,研发一种高性能多模式的内存数据库,以适应单应用、多应用、多节点、多现场等不同部署模式下的高性能数据访问要求。数据库以面向对象方式建模,提供面向对象接口、SQL 接口和键值访问接口,适应不同的编程模式和使用场景。该数据库广泛应用于电力、冶金、交通等多个领域,通过数千个不同规模系统的稳定运行验证其多模式的灵活性。

关键词 内存数据库 多模式 分布式 面向对象

中图分类号 TP311 **文献标识码** A **DOI**:10.3969/j.issn.1000-386x.2019.01.017

A HIGH PERFORMANCE AND MULTI-MODE MEMORY DATABASE SYSTEM

Lu Xin Wang Yanrong Sun Chao Shi Haojie

(NR Electric Engineering, Co., Ltd., Nanjing 211102, Jiangsu, China)

Abstract With the rapid growth of the smart grid and the industrial Internet of things, a single mode database cannot meet all the requirements. Therefore, a high performance and multi-mode memory database was developed to meet the requirements of high performance data access under different deployment modes such as single application, multi-application, multi-node, and multi-site. The database was modeled in an object-oriented way. It provided object-oriented interfaces, SQL interfaces and key value access interfaces, and adapted to different programming patterns and usage scenarios. The database is widely used in many fields, such as electric power, metallurgy and transportation. The multi-mode flexibility is verified by the stable operation of thousands of systems with different scales.

Keywords Memory database Multi-mode Distributed Object-oriented

0 引言

随着智能电网、工业物联网的蓬勃发展,实时数据库成为实时采集、实时控制、实时分析的重要支撑软件^[1-2]。同时云计算和大数据分析等技术推动了 No-SQL 数据库的发展,人们越来越认识到单一类型的关系数据库不能满足所有的任务^[3],发展了诸如基于 Key Value 的 Redis 数据库,基于 Document 存储的 MongoDB 数据库,基于列族存储的 HBase 数据库等等,针对特定问题的新的数据库技术层出不穷。

从数据库的规模来看,一方面,云端的大数据计算使得数据库越来越复杂,另一方面,随着边缘计算^[4]等

新的计算模型的提出,边缘计算的实时数据库也有精简的趋势。在此背景下,我们进一步发展了 Sophic-DB^[5],成为 SophicDB2 以适应多种计算模式下对数据库的要求。

本文根据近年以来该数据库在多个应用场景中的研发和应用,将其从一个分布式数据库发展为一个多模式的数据库。主要的场景包括:

(1) 应用嵌入式访问场景 运行于嵌入式装置的实时任务,可以消除与数据库服务器间的进程间通信开销,提升了访问效率。

(2) 数据库服务器场景 作为网络上的实时数据库服务,对应用进程提供位置透明的数据库服务。

(3) 数据库多现场场景 在系统级主备模式,如

异地双活,或高安全性架构,如电力自动化领域的安全区^[6]隔离,需要在不同物理位置部署数据库的可靠副本。

针对以上需求,本文提出了一种分层的多模式数据库架构,在统一存储引擎的支持下,使得其通过可配置的服务层,为不同的数据库使用方式提供一致的数据库接口,从而大大简化各类实时应用的开发。

1 技术架构

SophicDB2 数据库通过多模式支撑多种不同的应用场景,包括以下几个方面:

(1) 数据库部署和通信的多模式 数据库部署分为单应用模式、单机部署模式、多机部署模式和多现场部署模式。

相应的通信方式分为内存直接内存访问模式、本机访问模式、多机网络访问模式和多现场访问模式,访问透明性依次增强,性能依次降低。

(2) 编程接口的多模式 编程接口分为面向对象访问模式、层次访问模式、SQL 访问模式和键值访问接口,SQL 模式通用性好,面向对象接口编程效率和运行效率高。

为了最大化实现模块复用,数据库设计采用分层插件式设计以实现多种数据库使用模式的兼顾,如图 1 所示。两个维度上是正交的,包括本机-SQL 访问接口、多机-面向对象接口等,尤其需要指出的是只有面向对象接口支持直接内存访问模式,因此本数据库支持 10 种访问模式。

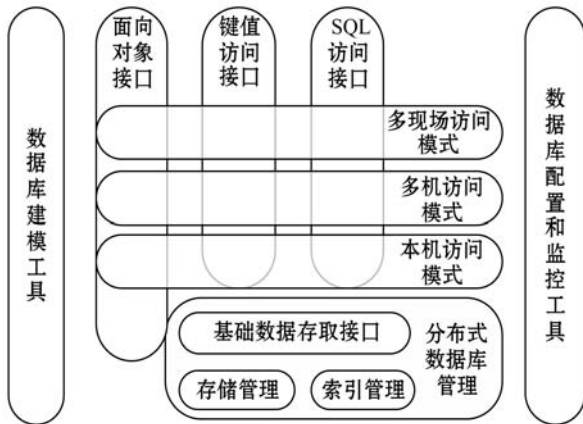


图 1 多模式数据库体系架构

2 关键技术

2.1 数据库建模

数据库的主要使用场景为能源和电力监控领域,

该领域的建模方式如 CIM 以层次对象建模为主^[7],例如变电站-电压等级-间隔-变压器-测点,这是一个层次化的包含关系。因此数据库建模的术语主要借用了对象关系术语,而非 ER 建模的方法。

(1) 继承 SophicDB2 支持类之间的继承关系,实现时从数据库的访问速度考虑,用基类的属性和关联扩充派生类,派生类既存储固有属性和关联,也存储从基类继承的属性和关联。

(2) 联系 CIM 中的联系分两类:简单联系和聚集。聚集是一种特殊形式的联系,它表示类之间的“整体—部分”关系,“整体”类由“部分”类组成 (consists of) 或包含 (contains) 部分类。

聚集关系又分成合成聚集和共享聚集,在合成聚集集中,“部分”有且仅有一个“整体”,一个合成聚集的“整体”拥有它的“部分”,“部分”不能脱离“整体”存在。

1) 1:1 非共生联系 SophicDB2 用 1:1 单向引用和 1:1 非共生双向引用实现简单联系。

(1) 1:1 单向引用 仅在引用发出端存有指向引用接收端的指针。

(2) 1:1 非共生双向引用 引用的两端均存有指向对端的指针,一端对象被删除不影响另一端对象的存在。

2) 共生联系 SophicDB2 用 1:1 共生双向引用、1:n 共生双向 Array 引用合成聚集。例如变电站删除了,下面的变压器也自然删除了。

(1) 1:1 共生双向引用 引用的两端均存有指向对端的指针,一端对象被删除,另一端与其共生的对象也不再存在。

(2) 1:n 共生双向 array 引用 上层对象(array 容器)有 n 个下层对象(array 成员),array 容器被删除后,array 的成员不再存在。

3) 1:n 非共生联系 SophicDB2 用 1:n 双向 Dlist 引用实现共享聚集。

• 1:n 非共生双向引用 上层对象(list 容器)有 n 个下层对象(list 成员),list 容器被删除后,list 成员仍存在。

以上关系在关系数据库中通过外键表示,但通过外键的查询本质上是一个查找过程,在采用 B+ 树做索引时,算法复杂度是 $O(\log_2 n)$ 。在本数据库中,通过一个间接指针实现,算法复杂度是 $O(1)$ 。尤其是在最常用的 1:n 关系中,例如一个变电站对象 (Substation) 包含多个开关对象 (Breaker),一个变压器所包含的所有开关都连续存储在一起,如图 2 所示。这对于

SCADA、FE 等这些常用的遍历操作,降低了 CPU 缓存换页和内存换页操作次数,实现了最高的访问效率。

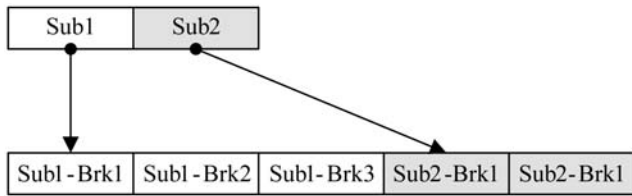


图2 共生 1:n 对象存储方式

2.2 数据库部署多模式

数据库支持四种部署模式,包括单应用模式、多应用模式、多机模式和多现场模式。

(1) 单应用模式 单应用模式的特点是应用逻辑和数据库在同一个进程中。

```
sophic_db_connection * pconn = sophicdb::
open( "/users/ems/sophic/db/
scada/scadamdl", SERVER_OPEN)
```

此时数据库直接以文件形式打开,数据库直接映射到应用进程的内存空间,数据访问的效率等效于内存。

(2) 多应用模式 多应用模式的特点是应用和数据库部署于同一个节点,但分属于不同的进程。

```
sophic_db_connection * pconn = sophicdb::
open( "sysnew", "scada", "scadamdl", CLIENT_OPEN)
```

此时应用通过 IPC 的共享内存方式访问数据库,应用间通过进程锁进行读写互斥。

(3) 多机模式 多机多应用模式的特点是多个应用和数据库部署于不同节点,并且多个数据库保持同步。

```
sophic_db_connection * pconn = sophicdb::
open( "sysnew", "scada", "scadamdl", CS_OPEN)
```

此时应用通过 TCP 报文访问数据库,由数据库服务器保证各副本的同步,并且多个副本通过读写分离的方式实现负载均衡。这是和大多数商用数据库如 MySQL 等类似的方式。

(4) 多现场模式 多现场模式的特点是应用和数据库部署于不同现场,现场间不能直接通信。

```
sophic_db_connection * pconn = sophicdb::
open( "sysnew", "scada", "scadamdl", CS_OPEN)
sophic_db_connection * pconn = sophicdb::
open( "site3", "scada", "scadamdl", CS_OPEN)
```

此时应用建立跨现场连接需通过第一个参数指定现场,默认 sysnew 参数为本现场,其他名称为指定现场,现场间通过符合特定要求的通信代理通信。

以上几种数据库部署方式对应了几种访问模式的对比如表 1 所示。

表1 四种数据库部署模式对比

数据库部署方式	访问模式			
	单应用模式	多应用模式	多机模式	多现场模式
数据存取方式	内存指针	本机 IPC 通信	直接 TCP 通信	分区网络代理
存取性能	极高	高	较高	较高
应用耦合性	高	中	低,位置无关	需指定网络分区
接口形式	直接内存操作 API	API,SQL	API,SQL	API,SQL
功能	单一,本进程专用	简单,可在本地进程间共享,少量并发	丰富,多副本支持分布式架构,大量并发	丰富,多副本跨现场同步
适合场景	资源有限的嵌入式装置,或性能要求极高的计算程序	嵌入式数据库,类似 SQLite 等的用法	通用数据库,类似于 MySQL, Oracle 等的用法	适合安全分区,异地主备等复杂场景

2.3 数据库接口多模式

关系数据库的 JDBC (ODBC) 接口是使用最广泛的数据库存取方式。同时在 Java 语言中我们使用对象关系映射 (ORM) 技术来解决关系数据库的关系模型和编程语言的面向对象模型的不匹配问题,如 Hibernate 框架^[8]等,其好处是编程时可用面向对象的方式访问数据库,而避免拼装繁琐的 SQL 语句,从而大大提高编程效率。对关系数据库来说,原生的是 SQL 接口,在此基础上派生了对象接口。

不同于关系数据库,实时监控系统的核心应用使用 C++ 开发,实时数据库本身以面向对象方式组织数据,系统生成和数据库定义一致的对象定义(.h 文件)提供面向对象 API。我们的内存数据库原生的是面向对象接口的,在此基础上派生了 SQL 接口。

1) 面向对象接口 在 Java 中一般使用反射 (reflection) 机制实现关系表字段到对象属性的映射,这存在一个序列化反序列化的过程。在实时数据库中,我们使用更直接的基于编译器的类型映射,从而节约字段查找和内存复制的开销。

(1) 从 OID 访问对象:

```
int ret = pconn -> get_object_by_oid ( oid_AnalogInput,
p_AnalogInput); //根据 OID 获取对象
cout << "名字:" << p_AnalogInput -> name << "地址:"
<< p_AnalogInput -> address << endl; //使用对象
```

(2) 从关系访问对象:

```
int ret = pconn -> get_relate_objects ( tid_AnalogInput, p_Sub-
```

```

station, p_AnalogInput, p_Substation -> ContainsAnalogInput);
//根据父对象 Substaion 获取子对象 AnalogInput
for(int i = 0; i < p_Substation -> ContainsAnalogInput. num;
i++) {
    cout << "名字:" << p_AnalogInput[ i ]. name << " 地址:"
<< p_AnalogInput[ i ]. address << endl; //遍历字对象
}

```

从 OID 和关系以面向对象方式访问数据可以大大提高编程效率。其他访问方式还有全表对象遍历,条件查询对象等,在此不再赘述。

2) SQL 接口 类似于关系库,数据库也提供 SQL 访问接口,以满足基于 Record 模式的数据访问。

```

int ret = pconn -> get_record_by_sql( "select name, address
from AnalogInput where oid = 2355677" , p_Record);

```

//根据 sql 获取 Record,oid 作为查询条件

相比于前述面向对象接口,其优点在于通用性好:

(1) 不依赖于数据库生成的对象定义文件,可基于数据库元数据。

(2) 封装为 JDBC(ODBC)后,开放性好。

缺点在于:

(1) 编程繁琐,需要拼 SQL,并没有字段类型等的编译器检查,编程效率低。

(2) 访问中间步骤较多,包括 SQL 解析执行,元数据的查询定位等。对实时应用来说开销较大。

3) 键值接口 类似于键值对数据库,数据库提供键值访问接口。不同的是,有三种键。

(1) 本表主键:如"1001001P",一个 Hash 索引或二叉树索引。

(2) 层次表主键结合:包括了其上级对象的主键,如"/雨花变/1001001P",这是一个二级索引,有时单表过大时,该层次索引效率更高。

(3) OID 主键:OID 是数据库内部对象的标识,应用程序可以存下来多次使用,这是一个间接地址描述符,查询效率最高,时间复杂度为 $O(1)$ 。

```

int ret = pconn -> get_object_by_key( "1001001P" ,
p_AnalogInput); //根据 key 获取对象
ret = pconn -> get_object_by_path( "/雨花变/1001001P" ,
p_AnalogInput); //根据层次表主键获取对象
ret = pconn -> get_object_by_oid( oid_AnalogInput, p_Analog-
Input); //根据 OID 获取对象

```

2.4 分布式多模式

1) 一主多读的负载均衡 数据库支持一主多副本的负载均衡模式,每个副本的数据都是完整的,应用首先判断本地是否有库,优先访问本地库,如图 3 所示。

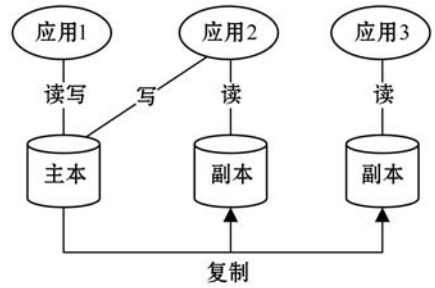


图 3 一主多读负载均衡

2) 多实例分布式负载均衡 第一种分布式负载均衡中各节点数据都是一致的,因此降低主本读的压力,对于超大规模的系统,例如 500 万点以上,单个库无法支撑,此时使用多实例分片的方式实现负载均衡,如图 4 所示。

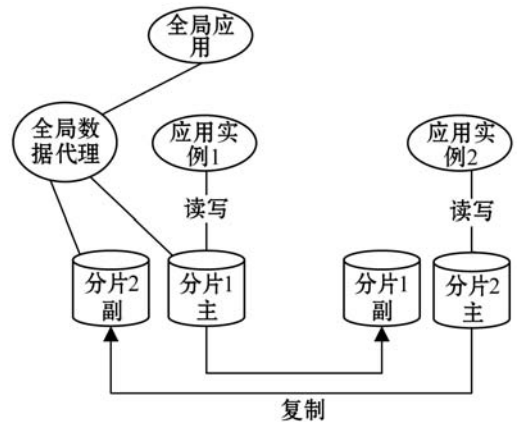


图 4 静态分片的多实例均衡

对实时应用来说,我们使用静态配置应用和数据亲和性的策略,例如 1~500 变电站及其包含对象放在分片 1 中,501~1000 变电站及其包含对象放在分片 2 中,如 SCADA 这样的实时处理程序和相关的分片部署在同一个节点,这样就保证每个测点处理时的时延基本一致。

另一方面,对如人机界面或统计分析这样的程序,实时性要求不高,但不希望了解每个测点在哪个分片上的应用。我们开发一个全局数据代理,对全局应用屏蔽了分片的细节,实现了数据的透明访问。

3) 多现场安全隔离 在电力领域,各应用是分安全区部署的,安全区之间不能建立直接的网络连接,此时必须通过代理连接,如图 5 所示。

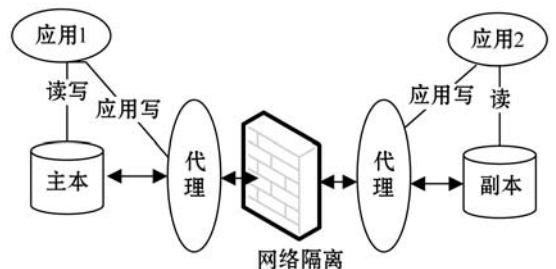


图 5 多现场安全隔离

代理的功能包括:

(1) 通信正常时,数据实时同步。

(2) 由于通信不可靠,管理数据库在通信异常时独立服务,异常恢复后可与主本同步。

(3) 基于安全考虑,一般不允许直接跨区的数据库写请求,但可以通过代理发给跨区的应用写。

(4) 记录所有操作的日志,以备安全审计。

3 应用实例

本文描述的多模式数据库应用于多个项目中,典型的应用的产品如下。

(1) 嵌入式装置 嵌入式装置的功能较为单一,资源比较有限。我们基于 SophicDB2 开发的智能终端采用单应用模式,实现了数据采集、存储、转发功能,相比于传统的远程终端单元(RTU),具备更多的功能和更好的扩展性,目前已有数百个应用案例。

(2) 厂站综合自动化 小型的电厂和变电站的综合自动化一般使用双机主备配置。我们基于 SophicDB2 开发的后台软件采用了多应用模式,包括了数据的采集、处理、控制和展示功能,目前已有数千个现场案例。

(3) 能量管理系统 大型系统,如电网的调度自动化系统、钢铁厂的能量管理系统、新能源集团的集中监控系统等,一般使用多机的集群系统,节点数可多达上百,并且需求不断增加,要求系统具备很高的可靠性和扩展性。我们基于 SophicDB2 开发的能量管理系统采用了多节点多实例模式,并开发了多实例的负载均衡机制,在常规的监控功能基础上,还开发了大量的高级分析应用,目前已有上百个应用案例。

(4) 地铁综合监控 对于类似地铁综合监控或地县配分级电网监控系统,其特点是正常时由上级系统监控,异常时下级系统也具备监控的能力,这就需要考虑数据在多个地点的独立工作和异常恢复机制。我们基于 SophicDB2 开发的地铁综合监控系统和地县配一体化电网监控系统采用了多现场模式,支持上下级异常时独立监控、故障恢复后自动同步等功能,目前已有数个应用案例。

4 结语

本文介绍的内存数据库提供了多种实时数据服务方式,并提供了面向对象 API、键值对和 SQL 三种接口。根据系统的不同规模和使用场景,归纳并实现了四种数据库部署模式,为从功能单一的嵌入式装置,到

复杂网络分区的多现场系统,提供了基本一致的应用编程接口,提升了软件复用程度。

在新的云计算和边缘计算框架下,SophicDB 仍有进一步发展的空间,例如作为微服务架构下的数据库^[9],最佳实践是一个服务一个数据库,通过建立微服务私有数据库和全局数据库(如 HBase)的映射关系,避免微服务和全局数据库的紧密耦合。也可以作为边缘计算装置的嵌入式数据库,可以继承现有的采集控制程序,这些方面仍可以继续研究,进一步扩展其使用场景。

参 考 文 献

- [1] 余广宏. 实时数据库技术的发展及应用效果[J]. 电子技术与软件工程,2016(20):185.
- [2] 吴庆曦,彭晖,王瑾,等. 电网调控集群分布式实时数据库的设计与关键技术[J]. 电力系统自动化,2017(22):89-95.
- [3] 覃雄派,王会举,李芙蓉,等. 数据管理技术的新格局[J]. 软件学报,2013,24(2):175-197.
- [4] 施巍松,孙辉,曹杰. 边缘计算:万物互联时代新型计算模型[J]. 计算机研究与发展,2017,54(5):907-924.
- [5] 崔昌栋,陆鑫,钱锋,等. SophicDB: 一个高性能分布式实时数据库系统[J]. 计算机应用与软件,2016,33(10):46-51.
- [6] 匡代璋,隰景宇,伍颖. 电力调度数据网安全防护设计及实现初探[J]. 中国新通信,2018,20(5):160.
- [7] 胡绍谦,李力,祁忠,等. 保护信息系统 IEC 61850 建模及 CIM 扩展的研究与应用[J]. 电力系统自动化,2016,40(6):119-125.
- [8] 张少应,程传旭. 基于 Hibernate 的对象关系映射研究[J]. 电子设计工程,2016,24(6):128-130.
- [9] 构建微服务体系结构的最佳实践[EB/OL]. [2016-09-19]. <http://www.infoq.com/cn/articles/best-practice-of-constructing-micro-service-system>.

(上接第 27 页)

- [9] Bittar A, Vitzilaios N I, Rutherford M J, et al. An integrated framework for cooperative ground and aerial vehicle missions utilizing Matlab and X-Plane[C]//2015 Annual IEEE Systems Conference (SysCon) Proceedings. IEEE, 2015: 495-500.
- [10] Seo Y J, Min C O, Lee D W. The Development and Flight Test of Auto-landing System Using X-Plane Simulator[C]//Proceedings of 2011 2nd International Conference on Innovative Computing and Communication and 2011 Asia-Pacific Conference on Information Technology and Ocean Engineering (CICC-ITOE 2011 V2), 2011: 4.