

船舶信息系统的数据库设计

王堪美^{1,2} 胡晓惠¹ 周文亚¹ 高 骞¹

¹(中国科学院软件研究所天基综合信息系统重点实验室 北京 100190)

²(中国科学院大学 北京 100049)

摘 要 随着船舶各个分系统之间数据交互要求日益增加、历史数据爆炸性增加,实现数据高效的分发以及海量数据存储日益重要。在分析数据分发服务 DDS(Data Distribution Service)规范的基础上,搭建船舶信息数据分发系统,同时提出基于 RDB 和 NoSQL(Not Only SQL)的联合存储方案。实现船舶信息系统的数据库高效、实时发布,以及海量历史数据的有效存储、高效查询。通过搭建船舶仿真系统测试船舶信息系统的数据库设计系统及其存储方案,实验结果表明将 DDS 技术和数据联合存储方案应用于船舶信息系统是可行的。

关键词 数据库设计 非关系型数据库 发布/订阅 船舶信息系统

中图分类号 TP3 文献标识码 A DOI:10.3969/j.issn.1000-386x.2014.04.005

SHIP INFORMATION SYSTEM DATA DISTRIBUTION AND STORAGE DESIGN

Wang Kanmei^{1,2} Hu Xiaohui¹ Zhou Wenya¹ Gao Qian¹

¹(Science and Technology on Integrated Information System Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)

²(University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract With the gradual increase of data exchange demands among various subsystems on the ship and the explosive increase of historical data, it becomes gradually important to realize data efficient distribution and mass data storage. On the basis of analyzing DDS specifications, the paper builds ship information data distribution system, at the meantime proposes an RDB-and NoSQL-based joint storage solution. It realizes ship information system's data's efficient and real-time distribution as well as mass historic data's effective storage and efficient querying. By establishing ship simulation system it tests ship information system's data distribution system and its storage solution. Experiment result show that it is feasible to apply DDS technology and data joint storage solution to ship information system.

Keywords Database design NoSQL Publish/Subscribe Ship information system

0 引 言

船舶信息系统是利用计算机、通信、网络、自动控制等技术,将相对独立的各个分系统纳入到统一的信息网络中,实现信息集成、集监测、控制、管理于一体的网络化、信息化综合系统^[1-3]。由于船舶平台的各分系统采用不同的监控技术、通信标准和来自不同的制造商的监控设备,给平台综合管理系统的信息集成设计带来了巨大的困难。而且随着实时性的需求越来越高,在信息集成的同时,必须保证各个子系统之间数据实时交互。目前,船舶信息系统采用的客户机/服务器模型(如 CORBA^[4])和消息队列模型(如 JMS^[5])等模型虽然可以实现系统内部之间的互连、互通和互操作,但针对船舶信息系统中越来越高的实时性应用需求,仍存在通信的实时性、快捷性和灵活性问题。为了满足分布式实时应用的通信需求,对象管理组 OMG 公布了全球第一个用于发布/订阅^[6,7]数据库设计服务 DDS^[8-10]的新规范。DDS 为各种不同类型的分布式应用提供了数据库通信模型,能促使数据库实时、高效地分发。

同时日益复杂的船舶信息系统,伴随着通信分系统的扩充、通信节点的增加,通信数据量日益加剧,历史数据量成爆炸性趋势增加,这些数据资源对存储系统的存储空间、存储可靠性、数

据访问性能等方面的要求越来越高。目前,船舶信息系统主要采用关系型数据库(如 SQL Server、DB2、Oracle 等),随着数据库爆炸性的增长,这些传统的基于关系型的数据库的存储系统已经远远不能满足船舶网络综合信息系统对数据存储和管理的需求,因此,非关系型的数据库 NoSQL^[11-13]现在成了一个极其热门的新领域,发展非常迅速。

针对船舶系统集成中通信的实时、高效性问题和海量历史存储问题,本文在对 DDS 进行深入分析和研究的基础上,设计了以数据为中心的船舶信息系统数据库设计服务模型,并结合 NoSQL 数据存储,提出一种海量历史数据的存储、管理方法。最后搭建了平台仿真系统实现了基于 DDS 的船舶信息数据库设计系统及其存储方案,满足了信息集成系统的通信实时、高效性需求和海量历史数据的有效存储、高效查询。

1 基于 OpenDDS 的船舶信息系统数据库设计

1.1 船舶信息系统数据库设计服务

通过深入分析 OMG 发布的数据库设计服务(DDS)规范,以开

收稿日期:2012-10-31。王堪美,硕士,主研领域:信息系统集成。
胡晓惠,研究员。周文亚,工程师。高骞,高工。

源 OpenDDS^[14] 产品为基础,对其进行优化和封装后,设计了一种以数据为中心的适合船舶信息系统的分布式实时通信的数据分发服务模型,满足船舶信息系统的数据分发需求,将采用不同监控技术、通信标准和来自不同制造商的监控设备的船舶分系统(如航行、电站、损管、推进)集成起来,并实现了数据的高效、实时分发。

OpenDDS 中以数据为中心的发布/订阅模型构建了一个共享的全局数据空间的概念,负责完成数据的发布、订阅,以及相关服务质量的控制保证等。任何对该空间中的数据感兴趣的应用程序都可以通过接口接入:向这个数据空间提供信息的应用程序通过声明意图可成为“发布者”,从数据空间中获取数据的应用程序通过声明订阅意图和感兴趣主题成为“订阅者”。每当发布者将新数据发送到全局数据空间,全局数据空间就会将感兴趣的订阅者与之进行关联,把信息传送给所有感兴趣的订阅者。

船舶信息系统数据分发模型如图 1 所示。船舶信息系统中节点的角色主要分为 3 部分:(1) 发布各个分系统数据的数据源节点;(2) 订阅各个分系统数据的订阅节点;(3) 管理数据交互的监测节点。一个节点也可以起到 3 个角色的作用,即可以在发布、订阅数据的同时监测交互过程。根据角色需求,数据分发服务中间件主要提供 3 种接口:发布接口、订阅接口和监测接口。

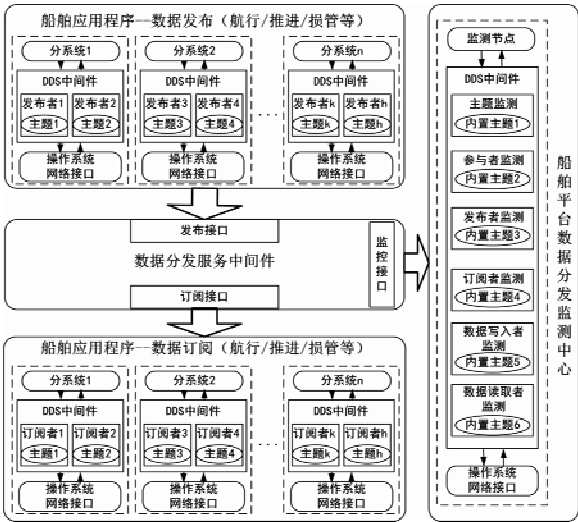


图 1 船舶数据分发服务模型

负责发布数据的船舶应用程序即数据发布者,为不同分系统的数据构建主题,然后通过发布接口建立实体对象将数据推入数据分发服务中间件中;负责订阅数据的船舶应用程序即数据订阅者,通过订阅接口建立实体对象,到数据分发服务中间件中订阅感兴趣的主体;负责监测交互过程的节点,直接到数据分发服务中间件中订阅内置的主题,当有主题、数据发布者、数据订阅者等通信实体进入数据分发服务中间件时,监测节点就可以监听到消息、获得交互信息。

1.2 船舶分系统数据发布和订阅流程

在上述构建的船舶信息系统中,各个分系统的数据发布和订阅的流程如图 2 所示。数据分发服务中间件的数据模型由“主题 (Topic)”和“类型 (DataType)”标识。“主题”是惟一标志某种数据的标志符。“类型”则提供了中间件如何操纵这些数据所需的结构信息,同时也允许中间件提供一定的类型保护措施。

发布数据的应用需要注册声明为“发布者”,注册时需要声明其发布的数据类型、主题号,以及提供的服务质量 (QoS),然后针对每个数据类型创建相应的数据写入者,将数据推送到中间件中。类似的,获取数据的应用则注册声明成为一个“订阅者”,注册时需要描述其需要的数据类型、主题号、以及要求的服务质量,然后针对每个数据类型创建相应的数据读取者来获取数据。中间件根据双方的数据类型、主题号以及服务质量来进行关联,若匹配则建立数据链路,这样每次发布者将新的数据对象推送到中间件,中间件将把数据正确地分发给所有的订阅者。

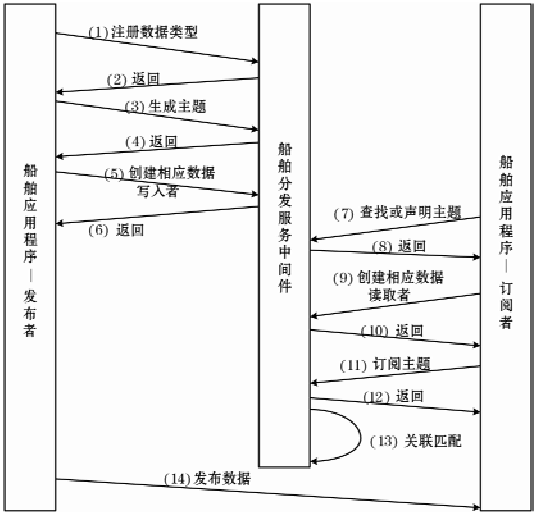


图 2 数据发布和订阅流程

具体步骤如下所示:(1) 发布者在数据分发服务中间件上注册数据类型;(2) 返回;(3) 发布者通知数据分发服务中间件生成主题,数据分发服务中间件根据数据类型定义主题并设置服务质量;(4) 返回;(5) 发布者根据数据类型创建相应的数据写入者 (DataWriter),每个数据写入者只专注于应用程序的一种数据类型;(6) 返回;(7) 订阅者向中间件发出请求,直接声明感兴趣的主体或在中间件中查找;(8) 返回;(9) 订阅者根据数据类型创建相应的数据读取者 (DataReader),每个数据读取者只专注于应用程序的一种数据类型;(10) 返回;(11) 订阅者订阅主题,中间件保存订阅信息;(12) 返回;(13) 中间件根据发布和订阅同一主题的双方的 QoS 进行关联:如果发布者提供的 QoS 满足订阅者要求的 QoS,匹配成功,进入下一步;否则不进行数据传递过程;(14) 关联的发布者和订阅者之间直接进行数据发布和传递。

1.3 数据分发服务的通信过程监测

在数据分发服务的通信过程中,发布者和订阅者是以通信实体的形式参与,并且每个实体都有相应的 QoS 绑定。这些通信实体包括:主题 (Topic)、参与者 (Participant)、发布者 (Publisher)、订阅者 (Subscriber)、数据写入者 (DataWriter) 和数据读取者 (DataReader)。为了实时监控通信过程中实体的状态及其变化,基于开源 OpenDDS 在全局数据空间中封装了监控接口,对通信实体及其交互关系、状态进行实时监控。对于每种通信实体,数据分发服务中间都有相应的内置主题,具体如下:

BuiltInTopic_Topic:主题的内置主题,发布或订阅的数据的标志符。

BuiltInTopic_Participant:参与者的内置主题,发布者和订阅者都是以参与者实体进行通信。

BuiltInTopic_Publisher:发布者的内置主题,发布者是一个负责分发数据的对象,它可以发布不同类型的数据。

BuiltInTopic_DataWriter:数据写入者的内置主题,数据写入者是发布者的一个类型化的接入者。

BuiltInTopic_Subscriber:订阅者的内置主题,订阅者是负责接收已发布的数据并使接收方的应用程序能够获得此数据,它可以接收并分派不同类型的数据。

BuiltInTopic_DataReader:数据读取者的内置主题,数据读取者是订阅者的一个类型化的读取者。

负责监测的应用直接订阅这些内置主题,订阅流程同普通主题的订阅流程。每当有参与者、主题、发布者、订阅者、数据写入者或数据读取者进入船舶分发服务中间件时,订阅这些内置主题的订阅者会收到相应通信实体的实体信息数据及相应的服务质量,如果是发布者/订阅者,则可以收到与其关联的订阅者/发布者信息。

2 海量历史数据的存储设计

为了将订阅到的数据进行数据持久化以保留历史数据以待查询、分析,针对订阅的历史数据随着时间推移以爆炸性的趋势增长,设计关系型数据库(RDB)和非关系型数据库(NoSQL)相结合的混合分布式存储架构 HybridDB:当日的数据存储在传统的关系型数据库中,以便快速查询;历史数据存储在非关系型数据库中,便于海量数据的持久保存、高效查询,以及系统和数据的易扩展。存储系统的设计如图 3 所示。

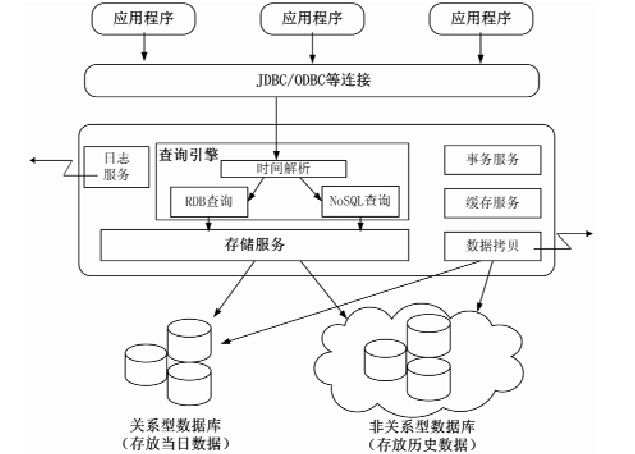


图 3 HybridDB 存储设计

应用程序查询数据时,具体查询步骤如下所示:

(1) 用户的查询首先被传到查询引擎模块,查询引擎对查询语句进行时间解析:如是当日日期,进入关系型数据库查询;如是历史信息,进入非关系型数据库查询。

(2) 查询引擎解析完后,调用存储服务与数据库进行查询交互,获取数据,返回给用户。

HybridDB 包含传统的数据库服务:日志服务、事物服务和缓存服务,并且包含数据拷贝模块。数据拷贝模块负责历史数据的拷贝加载工作,即定期将关系型数据库中的数据加载到非关系型数据库中。为了提高海量数据的加载效率和性能,数据

拷贝模块采用分布式 MapReduce 算法。MapReduce 算法的计算框架和逻辑如图 4 所示。MapReduce 算法是一个框架模型,主要是通过简单的接口来实现自动的并行化和大规模的分布式计算,通过使用 MapReduce 模型接口实现在大量普通的 PC 机上高性能计算。计算过程由称为 Map 和 Reduce 两部分用户程序组成,然后利用框架在计算机集群上面根据需求运行多个程序实例来处理各个子任务,然后再对结果进行归并。当调用 MapReduce 函数,具体计算过程如下所示。

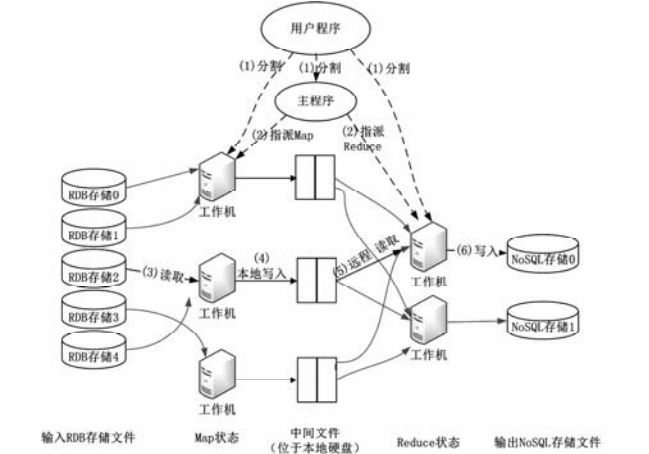


图 4 数据加载算法

(1) 用户程序首先调用 MapReduce 库将输入的 RDB 存储分成 M 个数据片度,然后用户程序在机群中创建大量的程序副本。

(2) 这些程序副本中有一个特殊的程序:主程序。副本中其它的程序都是工作进程(运行在工作机上),由主程序分配任务。有 M 个 Map 任务和 R 个 Reduce 任务将被分配,主程序将一个 Map 任务或 Reduce 任务分配给一个工作进程。

(3) 被分配了 Map 任务的工作进程将读取 RDB 存储片度的数据,从输入的数据片段中解析出 key/value 对,然后把 key/value 对传递给定义的 Map 函数,由 Map 函数生成并输出中间 key/value 对,并存储存储在本地磁盘中的中间文件中。

(4) 中间文件在本地磁盘的存储位置被传递回主程序,主程序负责把这些存储位置传送给 Reduce 工作进程。

(5) 当 Reduce 进程收到这些存储位置信息后,使用 RPC 从 Map 进程所在的磁盘上读取这些数据。当 Reduce 进程读取了所有的中间数据后,通过对 key 进行排序后使得具有相同 key 值的数据聚合在一起。

(6) Reduce 进程遍历排序后的中间数据,对于每一个唯一的中间 key 值,Reduce 进程将这个 key 值和它相关的中间 value 值的集合传递给 Reduce 函数。Reduce 函数的输出被追加到所属分区的 NoSQL 存储片区中。

3 系统仿真实验验证

为了验证 DDS 技术和分布式存储技术在船舶信息系统上的应用可行性,本文以船舶信息系统中的电力分系统、损管分系统、船舶分系统和舰长终端系统为例,搭建了一套基于 DDS 和分布式存储的船舶信息仿真系统。系统仿真实现如图 5 所示。

(下转第 72 页)

shop on Wireless, Ad Hoc, and Sensor Networks. Taiwan, China; [s. n.], 2005; 3-8.

- [7] 陈星舟, 廖明宏, 林建华. 基于粒子群优化的无线传感器网络节点定位改进[J]. 计算机应用, 2010, 30(7): 1736-1738.
- [8] 王晓乐, 徐家品. 基于粒子群优化算法的 WSNs 节点定位研究[J]. 计算机应用, 2009, 29(2): 494-496.
- [9] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [10] 尹呈, 郭观七, 李文彬, 等. 基于自适应学习的多目标粒子群优化算法[J]. 计算机应用研究, 2012, 29(9): 3232-3235.
- [11] Ratnaweera A, Hlagamuge S K, Watson H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.
- [12] 潘正君, 康立山, 陈毓屏. 演化计算[M]. 北京: 清华大学出版社, 1998.

(上接第 21 页)

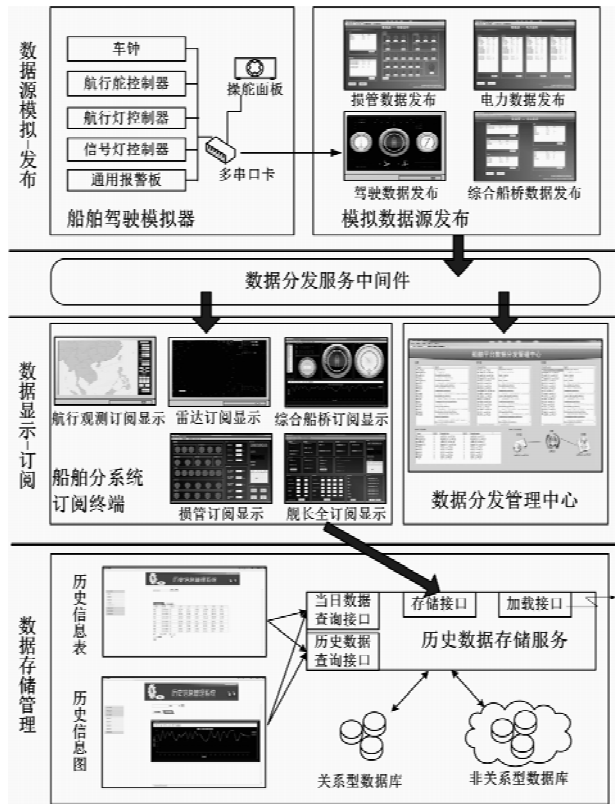


图5 系统仿真实现

“数据源模拟-发布”是船舶信息系统的数据发布方。驾驶数据通过仿真控制台上的驾驶模拟器硬件设备模拟产生, 电力分系统、损管分系统和船桥分系统中的数据通过程序模拟产生。应用程序采集、解析模拟的数据, 为不同数据构建相应主题, 将数据发送到数据分发服务中间件中。

“数据显示-订阅”是船舶信息系统的数据订阅者。他们依角色登录, 订购各自感兴趣的数据。航行观测终端只订阅航行数据; 雷达观测终端只订阅雷达数据; 综合船桥终端订阅船桥分系统发布的所有设备数据, 包括 GPS、测深仪、电罗经、计程仪和风速风向仪设备的数据; 损管观测终端订阅损管分系统发布的所有设备数据, 包括吃水、综合消防、火警探测设备的数据。其

中舰长终端可以灵活地订阅各个分系统的数据, 对全部数据进行监测和统计。

“数据分发管理中心”监控发布方和订阅方的参与实体及其状态变化。包括主题、发布者、订阅者实体及其服务质量信息, 以及发布者和订阅者的匹配关系。

“数据存储管理”负责实时数据和历史数据的存储和管理, 当日的数据存储到关系型数据库中, 历史数据存储到非关系型数据库中。这里的关系型数据库采用 SQL Server, 非关系型数据库采用 MongoDB, 是由 5 台虚拟机组成的机群进行存储和管理的。每日的 23:00 将当日的数据从 SQL Server 导入到 MongoDB 中。

利用上述的数据分发服务平台和混合存储数据库方案, 实现了船舶信息系统的集成, 各个系统之间能够灵活的通过发布/订阅进行数据分发, 消除了系统之间的紧密耦合性; 实现了海量历史数据的持久存储、高效查询, 以及数据存储系统的易扩展。

4 结 语

OMG 发布的数据分发服务规范专门为高性能数据分发而设计。它完全以数据为中心进行数据传输, 降低了系统集成时各个子系统的耦合度, 特别适合于现代船舶信息系统, 实现了船舶分系统的灵活扩展、数据按需高效订阅; 同时采用分布式混合存储方案, 完成对海量数据的存储和管理操作, 为船舶信息系统提供全面的数据共享、数据传输访问以及数据存储管理的存储服务。

参 考 文 献

- [1] 黄龙水, 孙允标. 舰艇综合平台管理体系基本结构分析[J]. 舰船电子工程, 2004, 24(6): 8-10.
- [2] 熊瑛, 许建. 船舶综合平台管理系统概念设计[J]. 船海工程, 2009, 38(5): 36-40.
- [3] 吕云飞, 王旋, 张军. 基于 DDS 技术的舰船平台综合管理系统[J]. 舰船科学技术, 2011, 33(5): 47-62.
- [4] OMG—Object Management Group. Common Object Request Broker Architecture: Core Specification, Version 3.0.3[S]. March, 2004.
- [5] Sun Microsystems. Java Message Service[S]. v1.1 SUN Specification, 2002.
- [6] Eugster P T, Felber P A, Guerraoui R, et al. The many faces of publish/subscribe[J]. ACM computing Surveys, 2003, 35(2): 114-131.
- [7] Corsaro A, Querzoni L, Scipioni S, et al. Quality of Service in Publish/Subscribe Middleware[J]. Global Data Management, 2003, 8(2): 79-97.
- [8] OMG—Object Management Group. Data Distribution Service(DDS) for Real-Time System[S]. v1.2 OMG Document, 2007.
- [9] 卢传富, 蔡志明, 夏学知. 数据分发服务体系结构的研究[J]. 计算机与数字工程, 2008(5): 67-69.
- [10] 杨俊. 基于 QoS 请求调度策略的 DDS 系统研究与实现[D]. 南京: 东南大学, 2010.
- [11] 姚林, 张永库. NoSQL 的分布式存储与扩展解决方法[J]. 计算机工程, 2012, 38(6): 40-42.
- [12] 毕洪宇. 利用 NoSQL 构建高性能全文检索系统[J]. 计算机与现代化, 2012(3): 122-127.
- [13] Open DDS[EB/OL]. <http://www.opendds.org>.
- [14] OCI—Object Computing, Inc. OpenDDS Developer's Guide, Version 2.3[M]. June 2009.