



数据同步复制技术 在控制河段通行指挥系统中的应用

赖珍宝¹, 钟 丽¹, 李永辉², 梁 山²

(1. 长江泸州航道局, 四川 泸州 646000; 2. 重庆大学自动化学院, 重庆 400044)

摘要: 针对控制河段船舶通行指挥系统中多信号台数据与服务器需要实时同步的问题, 提出利用 SQL Server 事务复制技术实现多信号台的数据库与服务器数据库上数据的实时同步, 并基于 SQL Server 数据库提供的 SqlDependency 类通过 C# 编程对整个同步过程进行监视, 实现主控端服务器能够知道何时接收到新的船舶 AIS 信息。

关键词: 控制河段; SQL 数据同步; 事务复制; 同步监视

中图分类号: U 617

文献标志码: A

文章编号: 1002-4972(2014)11-0024-04

Application of data synch replication technology in intelligent command system of controlled waterway

LAI Zhen-bao¹, ZHONG Li¹, LI Yong-hui², LIANG Shan²

(1. Luzhou Waterway Bureau of Yangtze River, Luzhou 646000, China; 2. College of Automation, Chongqing University, Chongqing 400044, China)

Abstract: In view of the problem of data synch between signal stations and server on intelligent command system of controlled waterway, this paper proposes using the transactional replication technology of SQL Server to realize data synch. Furthermore, the process of synchronous monitoring program based on SqlDependency class is created for the server knowing when ships' AIS messages were synchronized.

Keywords: controlled waterway; data synchronization on SQL; transactional replication; synchronous monitor

长江干线航道在我国内河航运中扮演着越来越重要的角色, 但是随着经济的发展, 船舶通航数量剧增, 加之内河航道自然条件恶劣、信号台通行指挥效率低下, 使得通航能力无法满足实际的需求, 这就制约了社会经济的发展。日前已经在长江上游控制河段部署完成“控制河段船舶通行智能指挥系统”, 该系统正向多信号台信息共享协同指挥发展, 迫切需要完善多信号台与服务器间的数据进行实时同步。

部署在信号台的控制河段船舶通行智能指挥系统不间断地接收行驶在弯曲狭窄控制河段中的船舶发送的 AIS (船舶自动识别系统, automatic identification system)^[1]信息被存储于信号台的数据

库中。当多个信号台间需要进行协同指挥时需要各个信号台将采集到的数据实时同步到服务器以实现信息共享, 但现有技术条件下信号台数据库中的数据尚不能主动将信息推送到服务器的数据库上, 这就可能导致新接收的信息不能得到及时处理, 出现不能正确掌控船舶的行驶状况、指挥失误等问题^[2], 为解决上述问题, 需利用数据同步技术, 把新接收到的 AIS 信息推送到服务器上进行处理。

文献[3]提出了一种“推拉式”分布式数据复制方案, 该方法虽然可以实现数据的同步, 但是该同步方案具有较长的时间间隔, 不能满足“控制河段船舶通行智能指挥系统”中对数据同步实

收稿日期: 2014-09-09

作者简介: 赖珍宝 (1965—), 男, 高级工程师, 从事航道工程、科技信息管理工作。

时性的需求; 文献[4]提出了选择数据库同步备份解决的 3 种方案, 即数据库镜像、数据库复制以及数据库日志传送, 该文仅从理论角度进行了简单阐述, 且采用该方法仅能够实现数据的同步而无法得知同步发生的具体时间和具体内容, 也不能在整个同步过程中随时获取需要进行处理的数据。

针对上述问题, 提出利用 SQL Server 事务复制技术来实现控制河段通行指挥系统中多信号台数据库与主机服务器数据库上数据的实时同步, 并基于 SQL Server 提供的 SqlDependency 的类结合 C#编程达到对整个同步过程进行监视, 当船舶发送的 AIS 数据发生变化时, 监视程序能够监视到数据库中数据的变化, 并能够把发生变化的数据发送给应用程序 (如电子航道图、百度地图), 从而实现对发生变化的数据进行及时处理。

1 控制河段多信号台数据同步方案

1.1 数据同步

位于控制河段中的信号台存储行驶在控制河段中的船舶 AIS 信息, 根据实际需求要把客户端 (位于斗笠子信号台) 数据库的数据与主控端服务器数据库数据进行同步, 要同步的数据存储于客户端信号台的数据库 “KZHD” 中, 其表为 “tb_dshipinfo”, 表中存储的是解码后的 AIS 信息, 主要用于记录船舶的 MMSI (maritime mobile service identity, 海上移动通信业务标识) 号、行驶速度、经纬度等通行信息, 存储形式如图 1 所示。

id		MMSI		navigationstatus		rot	speed	posaccuracy	longitude	latitude	cog	truehead	datetime
1	18257	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 01:58:00.000
2	18258	4133897	未定义值(U)	0	0	0	-1	-1	-1	-1	0	0	2013-08-26 01:58:22.000
3	18259	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 01:58:42.000
4	18260	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 01:59:00.000
5	18261	4133897	未定义值(U)	0	0	0	-1	-1	-1	-1	0	0	2013-08-26 01:59:22.000
6	18262	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 01:59:42.000
7	18263	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 02:00:02.000
8	18264	4133897	未定义值(U)	0	0	0	0	-1	-1	-1	0	0	2013-08-26 02:00:42.000
9	18265	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 02:01:02.000
10	18266	4133897	未定义值(U)	0	0	0	0	-1	-1	-1	0	0	2013-08-26 02:01:22.000
11	18267	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 02:02:00.000
12	18268	4133897	未定义值(U)	0	0	0	0	-1	-1	-1	0	0	2013-08-26 02:02:22.000
13	18269	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 02:02:42.000
14	18270	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 02:03:02.000
15	18271	4133897	未定义值(U)	0	0	0	1	105.557271666667	28.882631666667	0	0	0	2013-08-26 02:04:00.000

图 1 AIS 解码后数据

事务复制技术可以满足 “控制河段船舶通行智能指挥系统” 实时同步船舶 AIS 数据的需求。

事务复制是先把发布服务器数据库中的初始快照发送到各订阅服务器上, 然后复制监视器监视 SQL Server 上执行的 INSERT、DELETE、

UPDATE 数据修改语句, 并对其进行标识。事务日志阅读代理从事务日志中捕获出版条目中数据的变化, 并将引起这些变化的事务拷贝到分发服务器的分发数据库中^[5]。之后由分发服务器将这些事务传递到订阅服务器上, 在订阅服务器上顺序执行, 从而保证复制数据的一致性。由于事务的数据量远小于它所操作数据库的数据量, 所以采用事务复制方式还能够有效地减轻网络的通信负担。事务复制流程见图 2。

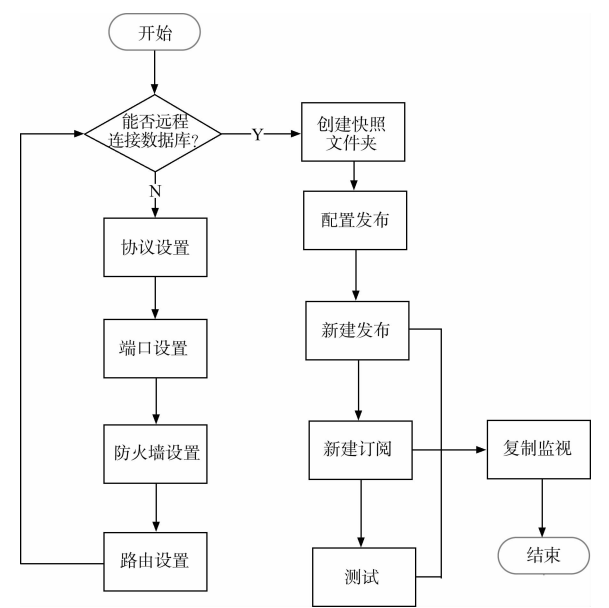


图 2 事务复制技术流程

按照图 2 所示的流程图对客户端服务器以及主控端服务器数据库进行相关配置, 配置完成之后主控端服务器便更新有客户端服务器数据库中的数据, 可以通过复制监视器查看同步状态、事务传递记录等信息 (图 3)。

所选会话中的操作 (U):	操作时间
已在 40498 毫秒内将快照从 "unc:\I-E-PC\JHARGAD\JHARGAD_P\BLSR\20140428205040\" 子文件夹中传...	2014/4/28 21:00:46
正在表 "[db1].[tb_dshipinfo]" 上创建主键索引	2014/4/28 21:00:46
已将前缀大容量复制服务器 "tb_dshipinfo" 中 (35040 行)	2014/4/28 21:00:46
正在将前缀大容量复制服务器 "tb_dshipinfo"	2014/4/28 21:00:44
已将前缀大容量复制服务器 "tb_dshipinfo" 中 (63728 行)	2014/4/28 21:00:44
正在将前缀大容量复制服务器 "tb_dshipinfo"	2014/4/28 21:00:41
已将前缀大容量复制服务器 "tb_dshipinfo" 中 (63728 行)	2014/4/28 21:00:41
正在将前缀大容量复制服务器 "tb_dshipinfo"	2014/4/28 21:00:39
已将前缀大容量复制服务器 "tb_dshipinfo" 中 (63728 行)	2014/4/28 21:00:39
正在将前缀大容量复制服务器 "tb_dshipinfo"	2014/4/28 21:00:36
已将前缀大容量复制服务器 "tb_dshipinfo" 中 (63728 行)	2014/4/28 21:00:36
正在将前缀大容量复制服务器 "tb_dshipinfo"	2014/4/28 21:00:34
已将前缀大容量复制服务器 "tb_dshipinfo" 中 (63721 行)	2014/4/28 21:00:34

图 3 复制监视器

1.2 可视化监视

数据源是一直变化的, 同步过程是一直进行的, 通过复制监视器可以查看当前事务以及同步

的状态,但是不能直观看到何时同步了何种数据,故需要对 SQL Server 复制过程进行监视,当数据库中有数据变化时能够知道数据库中发生了何种变化,方便对其变化的数据进行处理。

常用的利用 C#监控 SQL Server 数据变化的方法有两种^[6-7]:利用 DML (数据操纵语言, data manipulation language) 触发器实现监视;利用数据库中的 SqlDependency 类对数据库进行监视。

DML 触发器是一种特殊类型的存储过程,不能由用户直接调用,触发器只能对其创建之后的数据的操作进行控制,之前已存在的数据则不能控制;SqlDependency 工作原理如下:当被监视的数据库中数据发生变化时,SqlDependency 会自动触发 OnChange 事件来通知应用程序,从而达到让系统自动更新数据的目的。

根据系统的实际需求,利用数据库自带 SqlDependency 功能,采用 C#编程对数据库进行监视:启用 Service Broker 监听服务;调用 SqlDependency. Start (String strConnectionString) 方法,创建实例,并对数据的变化进行处理。

2 同步结果及分析

客户端服务器中 KZHD 数据库中要进行同步的数据如图 4 所示,其中主要包括船舶 MMSI 号、

船舶经纬度、船舶航行速度、船舶行驶方向与正北方向夹角、AIS 数据发送的时间,数据总条数约为 20 万条。

id	MMSI	navigationalstatus	rot	speed	posaccuracy	longitude	latitude	cog	truehead	datetime
1	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:55:28.000
2	413818747	未定义(默认)	0	0	0	105.609273333333	28.9391466666667	0	480	2013-08-22 11:55:28.000
3	413972093	未定义(默认)	0	11.6	0	105.604051666667	28.9450466666667	168.4	511	2013-08-22 11:56:34.000
4	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:41.000
5	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:01.000
6	413972093	未定义(默认)	0	11.5	0	105.604541666667	28.9434933333333	158	511	2013-08-22 11:56:03.000
7	4133897	未定义(默认)	0	0	0	-1	-1	0	0	2013-08-22 11:56:17.000
8	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:21.000
9	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:42.000
10	41372093	未定义(默认)	0	0	0	105.532016666667	28.9040333333333	92.5	511	2013-08-22 11:56:56.000
11	41372093	未定义(默认)	0	4	0	105.594081666667	28.9570833333333	282.1	511	2013-08-22 11:57:00.000
12	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:57:01.000
13	413972093	未定义(默认)	0	11	0	105.606073333333	28.9460866666667	148	511	2013-08-22 11:57:03.000
14	4133897	未定义(默认)	0	0	0	-1	-1	0	0	2013-08-22 11:57:21.000
15	413972093	未定义(默认)	0	10.8	0	105.607	28.9340333333333	150.8	511	2013-08-22 11:57:33.000
16	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:57:41.000
17	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:58:01.000
18	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:58:21.000
19	413818747	未定义(默认)	0	0	0	105.609273333333	28.9391466666667	0	480	2013-08-22 11:58:27.000

图 4 客户端服务器 KZHD 数据库中要同步的数据

同步进行之后主机服务器中的数据库中数据如图 5 所示。

id	MMSI	navigationalstatus	rot	speed	posaccuracy	longitude	latitude	cog	truehead	datetime
1	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:55:28.000
2	413818747	未定义(默认)	0	0	0	105.609273333333	28.9391466666667	0	480	2013-08-22 11:55:28.000
3	413972093	未定义(默认)	0	11.6	0	105.604051666667	28.9450466666667	168.4	511	2013-08-22 11:56:34.000
4	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:41.000
5	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:01.000
6	413972093	未定义(默认)	0	11.5	0	105.604541666667	28.9434933333333	158	511	2013-08-22 11:56:03.000
7	4133897	未定义(默认)	0	0	0	-1	-1	0	0	2013-08-22 11:56:17.000
8	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:21.000
9	4133897	未定义(默认)	0	0	1	105.557258333333	28.8826083333333	0	0	2013-08-22 11:56:42.000

图 5 主机服务器数据库中同步的数据

通过对比可以发现两个数据库中的数据是一致的,达到了同步的目的。为了查看同步过程,在客户端服务器数据库中输 入一串数据,数据如图 6 所示。

```
SQLQuery4.sql - YD...tor (54) 正在执行... SQLQuery3.sql - YD...ministrator (51)* SQLQuery2.sql - Y...dministrator (58) SQLQuery1.sql - YD...ministrator (56)*
set identity_insert dbo.tb_dshipinfo ON
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000001,413389712, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000002,413940754, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000003,413356742, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000004,413786416, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000005,413865432, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000006,413894646, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000007,413565654, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000008,413656434, 0,0,1,105,28,0,0,2014-08-22);
insert into dbo.tb_dshipinfo (id, MMSI, rot, speed, posaccuracy, longitude, latitude, cog, truehead, datetime) values (2000009,413389788, 0,0,1,105,28,0,0,2014-08-22);
```

图 6 测试数据

查看主机服务器数据库,可以看到新接收的船舶 AIS 数据已经出现,如图 7 所示。

结果		消息									
id	MMSI	navigationalstatus	rot	speed	posaccuracy	longitude	latitude	cog	truehead	datetime	
991929	20000011	123456781	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991930	20000012	123456782	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991931	20000013	123456783	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991932	20000014	123456784	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991933	20000015	123456785	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991934	20000016	123456786	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991935	20000017	123456787	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991936	20000018	123456788	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000
991937	20000019	123456789	NULL	0	0	1	105	28	0	0	2014-04-22 00:00:00.000

图 7 主机服务器数据

同时,利用 C#编写的监控程序可进行可视化监视,如图 8 所示。

MMSI	longitude	latitude	cog	datetime
123456781	105	28	0	2014-08-22
123456782	105	28	0	2014-08-22
123456783	105	28	0	2014-08-22
123456784	105	28	0	2014-08-22
123456785	105	28	0	2014-08-22
123456786	105	28	0	2014-08-22
123456787	105	28	0	2014-08-22
123456788	105	28	0	2014-08-22
123456789	105	28	0	2014-08-22

图 8 可视化监视

同步“KZHD”中表“tb_dshipinfo”的数据耗时 10 s 左右, 在客户端服务器数据库中输人一串数据对新接收的数据进行同步耗时和监视测试, 结果耗时 2 s 左右。鉴于该表中约 20 多万条数据, 数据量较大, 故可以认为实现了实时同步, 并且利用 C#对数据变化进行可视化监视, 每当有数据发生变化时变化通知显示数据的变化。

3 结语

目前控制河段通行指挥系统向多信号台信息共享协同指挥发展, 迫切需要完善多信号台与服务器间的数据进行实时同步。本文针对这一问题, 在考虑同步数据容量、同步耗时的基础上提出利用 SQL Server 事务复制技术对信号台数据库和服务 器数据库进行数据同步, 订阅服务器按时间顺序执行发布服务器上发送的更新数据请求, 并将此请求更新在服务器数据库上, 从而实现数据同步。同时, 针对控制河段通行指挥系统需要监视同步数据内容的要求, 启用 Service Broker 监听服务, 利用 C#进行编程, 实现对同步过程的监视。

测试结果表明, 所采用的方法可有效对数据进行实时同步并监视其同步过程, 满足控制河段通行指挥系统的要求。

参考文献:

[1] 刘畅. 船舶自动识别系统 (AIS) 关键技术研究[D]. 大连: 大连海事大学, 2013.

[2] 靳智, 梁山, 曹芳平. 基于自动识别系统的长江控制河段船舶视觉伺服跟踪方法[J]. 计算机应用, 2011, 31 (12): 3 414-3 417.

[3] 高振清. 分布式数据库数据复制技术研究[J]. 延安职业技术学院学报, 2013, 27(5): 100-101.

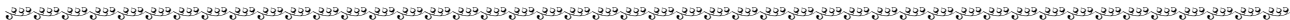
[4] 孙广. 通过数据库复制功能实现数据库同步备份[J]. 科研发展, 2012(22): 97-99.

[5] özsü M T, Valdúriez P. Principles of distributed database systems[M]. Berlin: Springer, 2011.

[6] 常捷. 利用 CLR 实现 SQL Server 2005 复杂字符串格式的验证[J]. 计算机与现代化, 2010 (10): 186-188.

[7] 陈建华. 一种基于 C#的 SQL 服务器安全的监控方法[J]. 重庆工商大学学报: 自然科学版, 2012, 29(5): 73-77.

(本文编辑 武亚庆)



· 消 息 ·

广航局参建的广州港出海航道三期工程通过国家竣工验收

10 月 17 日, 广东省交通运输厅组织召开广州港出海航道三期工程竣工验收会议。会上, 与会专家一致同意该工程通过国家竣工验收。

广州港出海航道三期工程按 10 万吨级集装箱船不乘潮单向通航、兼顾 12 万吨级散货船乘潮单向通航、5 万吨级集装箱船不乘潮双向通航的标准建设, 工程范围南起珠江口外隘洲岛南侧的天然水深区, 北至广州港南沙作业区中部挖入式港池口门北边线, 航道长度约 71. 84 km, 有效宽度 243 m, 设计底高程 - 17. 0 m。广航局共承担本工程 4 个标段的施工任务, 工期为 2006 年 7 月—2013 年 5 月。

该项目的建成、运营不仅将促进广州港本身的发展, 而且也将带动广大腹地社会经济和水上运输的发展, 进一步巩固广州作为华南航运枢纽的地位。

(摘编自《中国交通建设网》)