

基于 CDC 机制的数据仓库实时数据更新方法研究

谭光玮 武 彤

(贵州大学计算机科学与技术学院 贵阳 550025)

摘 要 分析了某特定应用系统的数据仓库实时决策需求,确定了需要实时更新到数据仓库的数据库表。对几种实时更新数据的方案进行了比较和权衡,经过综合考量,设计了使用基于读取和分析数据库日志的 CDC 机制来捕获变更数据,然后在数据加载程序中设定周期,循环地将捕获到的变更数据放入中间数据集并批量加载到数据仓库中的实时数据更新方案。该方案经过实验验证可以满足实时更新数据的需求,并且更新数据的过程不会影响源系统的事务处理,适用于此应用系统。

关键词 动态数据仓库,实时更新,变更数据捕获,数据加载

中图分类号 TP311 文献标识码 A

Study on Method of Data Warehouse Real-time Data Updating Based on Mechanism of CDC

TAN Guang-wei WU Tong

(School of Computer Science and Technology, Guizhou University, Guiyang 550025, China)

Abstract This article analysed the real-time decision requirement of the data warehouse of a specific application system, identified database tables which need to be updated in real-time to data warehouse. Then compared several overall plan of real time data loading, after comprehensive consideration, we designed a method of real time data loading—using the CDC mechanism to acquire the changed data, and then circularly load the changed data to data warehouse via data loading program. After experimental verification, this method can meet the requirement of real time data loading, while doesn't have much influence over the Database Transaction. So it's suitable to be applied to the system.

Keywords Real-time data warehouse, Real-time updating, Changed data capture, Data loading

动态数据仓库在传统数据仓库的基础上增加了对实时决策的支持,为了满足实时决策的要求,动态数据仓库中的一部分数据需要最新的实时数据,因此动态数据仓库的实时数据更新问题是目前企业进行实时决策时必须面对的问题。

本文基于实际使用中的某企业生产线质量控制系统的数据库,分析该系统实时决策的需求,对几种实时更新数据的方案进行权衡后做出选择,然后具体地设计并实现了所选择的方案,并将其应用于生产线质量控制系统中,最后通过实验验证了所设计的实时数据更新方案的可行性。

1 研究背景及需求

某电视机生产线质量控制系统分为数据采集及 OLTP 系统、OLAP 及决策分析系统。数据采集及 OLTP 系统的主要目的是管理电视机基本信息,并从生产线的数个数据采集点采集电视机生产中的故障信息,存入事务处理系统对应的数据库 PQACS 的电视机故障信息记录表 faultstatistics 中; OLAP 及决策分析系统的分析处理基于数据仓库 HXDW,它一般用于 OLAP 等战略分析,例如一段时间内某种机型出现某种故障数目(时间+故障现象+机型)(切块分析,钻取分析,按时间维度(月,季度,年)钻取),在 HXDW 中有故障统计细节表 basicinfo,其数据来源于数据库 PQACS 中的电视机故障信息记录表 faultstatistics。对于该系统有一系列的基于数据仓库 HXDW 的实时查询分析需求,如,某一类型的电

视机产品每天出现故障数量超过给定阈值时需要立即检查生产线上的生产情况,因为同一故障原因出现的故障数量超过给定阈值时,需要立即检查原材料质量、线上生产工人状态等,这时就需要将故障记录信息表 faultstatistics 中的新数据实时地加载进数据仓库中的故障统计表 basicinfo 中,根据企业对实时处理的要求,需要在 10 秒内在数据仓库中获得最新采集到的故障信息,因此这里的实时加载也是指在满足需求的前提下尽量缩短延时。

图 1 为整个系统的组成部分及其各部分所对应的作业, PQACS 数据库用于数据库操作等事务处理,从多个数据采集点获得最新的故障信息。HXDW 数据仓库在用于 OLAP 等战略分析之外,也需要满足一些实时分析需求[1-3]。

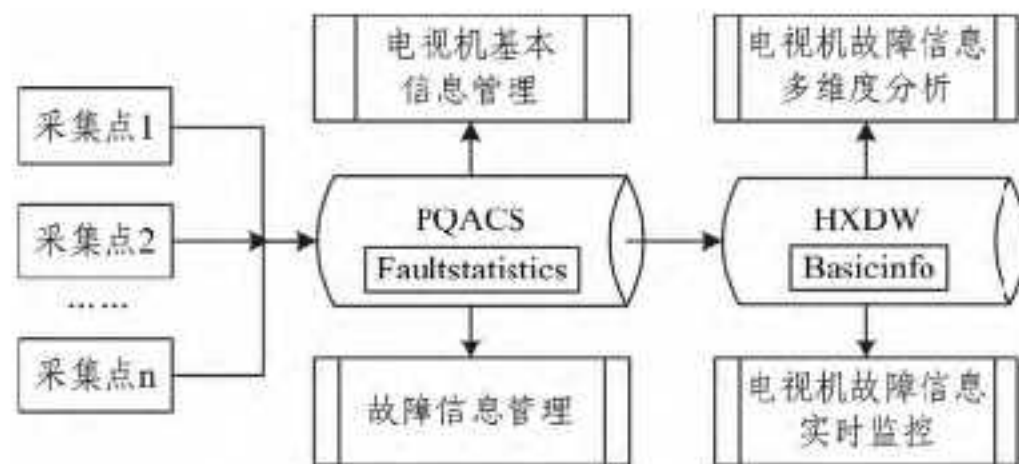


图 1 电视机生产线质量控制系统的组成

2 可选择的实时更新方案

选用的实时更新方案,需要满足以下基本准则[4,5]:

本文受贵州省自然科学基金项目,动态数据仓库的数据加载技术研究(黔科合 J 字[2013]2115 号)资助。

谭光玮(1989—),男,硕士生,主要研究方向为数据库技术,E-mail:troybrown@163.com;武彤(1964—),女,硕士,教授,CCF 会员,主要研究方向为数据仓库技术、OLAP、数据挖掘,E-mail:wtxx@citiz.net。

- (1) 不影响源业务系统的事务处理;
- (2) 实时性要求高的故障记录信息的变更数据应做到接近零延时地加载到数据仓库中。

目前在动态数据仓库实时更新的研究中有以下几个方案可供选择^[6,7]:

(1) 比较数据库表 `faultstatistics` 的前后快照的差分得到变更数据,之后加载这部分变更数据至数据仓库细节表 `basicinfo` 中。这种方式在执行过程中需要将 `faultstatistics` 中变化前后的所有数据放到内存中,由于 `faultstatistics` 中数据量很大,因此其空间代价相当高,所以不予考虑。

(2) 在数据库表 `faultstatistics` 上设置触发器,当有新的故障信息时,触发其中的存储过程,将新的故障信息插入到数据仓库细节表 `basicinfo` 中。这种方式的实时性非常好,但是每个触发操作都会占用数据库资源,因此这种方式在短时间内新的故障信息数量不大的时候效果很好,但是在系统的应用中,高峰时期数个采集点会同时高频率地采集故障信息,每秒会有很多新的故障信息进入数据库中,这时频繁地触发操作会严重影响数据库的事务处理性能。

(3) 利用数据库的事务日志得到最新的故障信息,将它们放入一个中间的结果集中,循环地查询该结果集,从而将最新的故障信息插入到数据仓库中。这种方式得到的变更数据来源于事务日志,异步地读取和分析日志不会影响数据库的事务处理,设置适当的查询周期也可以满足系统的实时需求。因此针对电视机生产线质量控制系统,这种方式最适合。

下面将叙述所选择方案的功能模块和实现过程。

3 实时更新方案的具体实现

首先需要利用事务日志得到最新的故障信息,生产线质量控制系统的数据库存储在 `sql server` 中,它提供了一种变更数据捕获机制(changed data capture,简称 CDC),这种机制通过异步读取、分析选定数据库的事务日志,实时地得到选定数据库表中的变更数据,可以利用这个机制得到想要的最新故障信息,步骤如下^[8-10]。

1. 开启数据库级别的 CDC(这里为数据采集及 OLTP 系统对应的数据库 PQACS)

```
USE PQACS
GO
EXEC sys.sp_cdc-enable-db
GO
```

2. 对存储故障记录信息的表 `faultstatistics` 开启表级的 CDC

```
USE PQACS;
GO
EXECUTE sys.sp_cdc-enable-table
    @source-schema=N'dbo'
    ,@source-name=N'FaultStatistics'
    ,@role-name=null
    ,@capture-instance=DEFAULT
GO
```

在这一步之后,建立一个名为 `FaultStatistics-CT` 的表,在建立此表后,所有对 `faultstatistics` 所做的更改都会记录在这个表里,此表在结构上保持了被追踪记录的表 `faultstatistics` 的所有字段,并增加了用于分辨操作类型的 `operation` 字段,根据该字段不同的值可以分辨出对 `faultstatistics` 所做的操作的类别(增为 2,删为 1,改为 3(旧值)、4(新值)),也就可以从这个表中得到想要的最新故障记录信息。在 `faultstatist-`

`tics-et` 表里得到最新的故障记录信息之后,就可以根据不同的操作类别将这些信息组织在程序中设置的结果集中,然后将这些信息更新到数据仓库对应的细节表 `basicinfo` 中^[11,12]。

数据加载程序部分核心代码段如下:

```
public static void Main(string[] args)
{
    Timer tmr=new Timer(new TimerCallback(dataload),null,1000,
        5000);
    tmr.Dispose();
}
```

这里使用 `timer` 控件控制程序执行,程序开始运行 1000 毫秒(1 秒)后开始执行 `dataload` 方法,之后每隔 5 秒循环执行 `dataload` 方法。对数据的加载都写在 `dataload` 方法中,`data-load` 方法的实现代码概要如下。

```
public static void dataload(object sender)
{
    cmd.CommandText="select * from dbo-FaultStatistics-CT
        where ID>@lastcheckID and operation=2";
    .....

    SqlBulkCopy bulkcopy=new SqlBulkCopy(connstr);
    bulkcopy.DestinationTableName="basicinfo";
    bulkcopy.WriteToServer(table);

    cmd.CommandText="select ID from dbo-FaultStatistics-CT
        where ID>@lastcheckID and operation=1";
    Executenonquery();

    cmd.CommandText="select * from dbo-FaultStatistics-CT
        where ID>@lastcheckID and operation=4";
    Executenonquery();
}
```

在 `dataload` 方法中,根据 `operation` 字段的不同,对应不同的处理流程,其值为 1 时,根据主键 ID 删除 `basicinfo` 中对应的故障信息;其值为 4 时,将 `FaultStatistics-CT` 表中相应的更改数据更新到 `basicinfo` 对应的故障信息行;其值为 2 时,是新插入的故障信息,将其加入到中间数据集中,再载入数据仓库,由于在高峰时期,多个采集点不断采集新的故障信息,短时间内可能会在 `FaultStatistics-CT` 表中产生大量新的故障信息,一条一条地插入故障信息到数据仓库细节表 `basicinfo` 中会耗费大量时间,所以这里采用 `sqlbulkcopy` 进行批量加载。

图 2 为数据库中变更数据从被捕获到实时加载到数据仓库的全过程的主要流程。

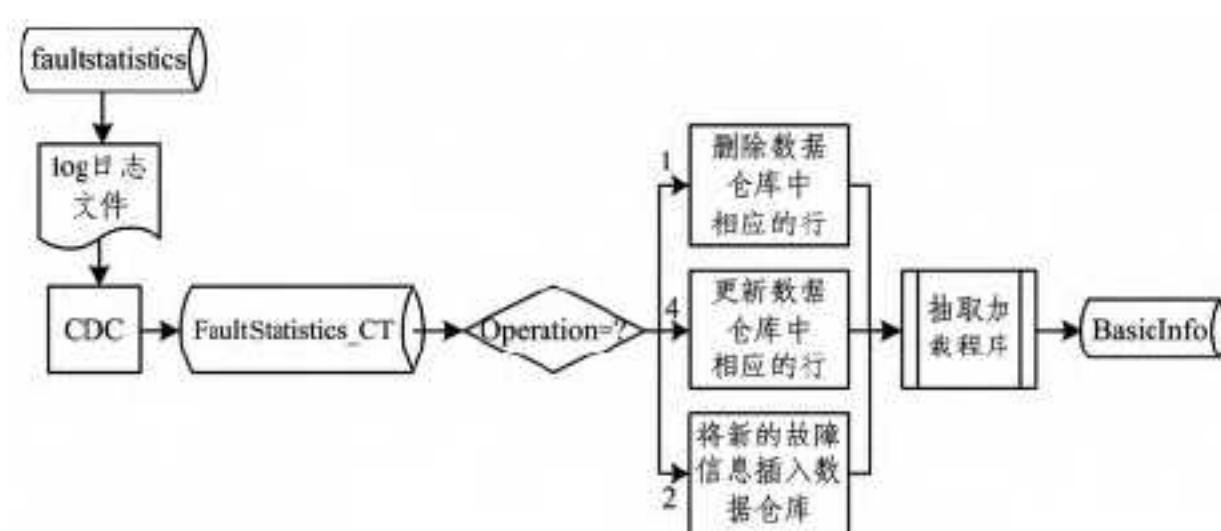


图 2 实时数据更新的主要流程

在上面的步骤中,不可避免地需要对 PQACS 数据库的 `FaultStatistics-CT` 进行查询来得到最新的故障信息,而在对数据库的故障记录表 `faultstatistics` 开启 CDC 之后,就会将 `faultstatistics` 中的变化全部记录在对应的变更记录表 `FaultStatistics-CT` 中。在信息采集的高峰时段,表中的数据量在短时间内就会变得很大,而这也影响查询该表的时间和性能,因此需要定期清除表中已经加载到数据仓库中的变更数据。系统要求的是 10s 内得到数据库中的变更数据,上面的

加载程序设定在 5s 循环查询执行一次。在 CDC 机制中,当对指定数据库 PQACS 的表开启 CDC 之后,会在数据库服务器代理当中创建两个作业 cdc. PQACS—eapture 和 cdc. PQACS—cleanup, 第一个作业负责读取和解析数据库事务日志得到变更数据,后一个作业负责定时清空变更数据,默认的数据清理时间是每天一次,在系统数据量快速增长的同时想提高查询速度,这显然是不符合要求的,因此这里调用如下存储过程设置 CDC 中 cdc. PQACS—cleanup 作业的数据定期清理时间。

```
Exec sys. sp_cdc_change_job
@job-type=N'cleanup',
@retention=2
go
```

这样就将 FaultStatistics—CT 的数据清理时间设置为了 2 分钟一次。

4 实验验证

4.1 实验 1: 测试更新数据的实时性

本文提出的实时数据更新方案利用基于读取和分析日志的 CDC 机制得到目标数据库的变更数据,将其装入对应的变更数据表中,再循环执行 dataload 方法将变更数据加载到数据仓库中,因此 dataload 方法的执行时间是决定此更新方案实时性的关键,故在此测试 dataload 方法执行一次的时间。

在查询数据库语句之前插入 DateTime starttime=DateTime. Now 以标记执行方法前的时间,在批量插入语句之后、方法体的结尾,插入 TimeSpan usingtime=DateTime. Now-starttime; 计算方法执行所耗费的时间,并把这个时间打印出来。

```
public static void dataload(object sender)
{
    DateTime starttime=DateTime. Now
    cmd. CommandText = " select * from dbo -FaultStatistics -CT
    where ID>@lastcheckID and operation=2 ";
    .....
    SqlBulkCopy bulkcopy=new SqlBulkCopy( connstr);
    bulkcopy. DestinationTableName= " basicinfo ";
    bulkcopy. WriteToServer( table);
    .....
    TimeSpan usingtime=DateTime. Now-starttime;
    Console. Writeline(usingtime. ToString());
}
```

方法执行一次后,显示其一次执行的时间如图 3 所示。



图 3 dataload 方法执行一次的时间

如图 3 所示,执行一次 dataload 方法,只需要 0.39 秒左右,这样循环 5 秒执行一次 dataload 方法不仅可以满足系统对于数据实时性的需求,还给了数据库充分的释放资源、重新组织连接池的时间。

4.2 实验 2: 测试实时更新数据是否会对源系统的数据库操作有影响

在 CDC 机制和数据加载程序开启时,通过生产线质量控制系统的 OLTP 子系统对 PQACS 数据库进行数据库中电视机信息的查询操作,测试查询的响应时间,通过查看操作时产生的 HTTP 请求和响应报文的信息,可得到事务处理的时间。

由图 4 可见,同时对数据库开启 CDC 机制并且运行数据

加载程序时,此次生产线质量控制系统的 OLTP 处理反应时间为 2.937 秒,源系统的性能并未受到严重影响,可正常操作。

Performance	Timings	Status Codes	Errors
Description	Value	Units	
Elapsed Time	2.937	seconds	
Network Round Trips	1		
Downloaded Data	9703	bytes	
Uploaded Data	546	bytes	

图 4 响应报文中的处理时间

由以上两个实验可以得出,本实时数据更新方法可以在几乎不影响数据库事务处理性能的前提下,完成对需要实时加载到数据仓库的数据的近实时加载,适合应用于此生产线质量控制系统的。

结束语 生产线质量控制系统中有时决策的需求,因此数据仓库中的一部分细节表需要实时更新数据,以达到实时决策。在实时更新数据时不仅需要满足系统的实时要求,也不能影响数据库的事务处理性能。

本文在综合权衡后,设计了使用基于读取和分析数据库日志的 CDC 机制来捕获变更数据,然后在数据加载程序中设定周期,循环地将捕获到的变更数据放入中间数据集并批量加载到数据仓库中的实时数据更新方案。经过实验验证,这种方法可以满足实时更新数据的需求,也不会因为实时更新数据过多而影响源系统的数据库事务处理。本文的实验是在有限数量的采集点的情况下进行的,下一步将在整个生产线的全部采集点同时进行新的故障信息采集的情况下来验证此方法是否仍然有效且不影响源数据库,并根据结果进一步改进加载数据的算法。

参 考 文 献

- [1] 黄帆. 对提高企业级数据仓库数据即时性的研究[D]. 上海: 上海交通大学, 2009: 22-40
- [2] Inmon W H. Building the data warehouse[M]. New York: John Wiley & Sons, 1996
- [3] 杨乐. 数据仓库中实时抽取机制的研究与实现[D]. 北京: 北京邮电大学, 2007: 23-32
- [4] 徐春艳. 面向实时数据仓库的 ETL 研究[D]. 南京: 南京航空航天大学, 2007: 18-35
- [5] 肖裕洪. 实时数据仓库关键技术的研究与实现[D]. 广州: 华南理工大学, 2011: 13-20
- [6] 徐富亮, 周祖德. 变化数据捕获技术研究[J]. 武汉理工大学学报, 信息与管理工程版, 2009, 31(5): 740-743
- [7] 林子雨, 杨冬青, 宋国杰, 等. 实时主动数据仓库中的变化数据捕捉研究综述[J]. 计算机研究与发展, 2007, 44(23): 447-451
- [8] Shi Jin-gang, Bao Yu-bin, Leng Fang-ling, et al. Study on log based change data capture and handling mechanism in real-time data warehouse[C]// Proceedings of 2008 International Conference on Computer Science and Software Engineering. Wuhan, 2008: 478-481
- [9] 邹先霞, 贾维嘉, 潘久辉. 基于数据库日志的变化数据捕获研究[J]. 小型微型计算机系统, 2012, 33(3): 531-536
- [10] 陈振. 基于日志分析的 SQL Server 数据库变更数据捕获方法的研究与实现[D]. 广州: 暨南大学, 2010: 23-46
- [11] 舒琦. ETL 过程优化与增量数据抽取的研究[D]. 长沙: 湖南大学, 2011: 34-56
- [12] Itamar Ankorian. Changed data capture-efficient ETL for real-time BI[OL]. <http://www.dmreview.com/article-sub.cfm?articleId=1016326>