第 36 卷 第 3 期 计 算 机 工 程 2010 年 2 月
Vol.36 No.3 Computer Engineering February 2010
·软件技术与数据库· 文章编号：1000—3428(2010)03—0055—03 文献标识码：A 中图分类号：TP311.13

分布式过程实时数据集成方法及其实现

薛尧予，王建林，赵利强
(北京化工大学信息科学与技术学院，北京 100029)

摘 要：针对异构生产装置数据采集、集成和管理中的数据集成问题，提出一种分布式过程实时数据集成方法，给出系统体系结构和数据集成原理。采用 Hash-AVL 树的数据结构对生产数据进行描述，利用 XML 技术对实时数据及访问请求进行封装，实现统一的数据访问接口。该方法应用到某石化企业综合自动化系统，对 100 个数据点进行并发访问时，数据更新周期小于 3 s，结果证明了分布式数据集成方法可以满足对现场生产装置异构实时数据进行集成的要求。

关键词：实时数据；数据集成；XML 技术；Hash-AVL 树

Integration Method and Its Realization of Distributed Process Real-time Data

XUE Yao-yu, WANG Jian-lin, ZHAO Li-qiang
(College of Information Science & Technology, Beijing University of Chemical Technology, Beijing 100029)

【Abstract】Considering the data integration problem in data acquisition devices, integration and management of heterogeneous production, an integration method of distributed process real-time data is proposed meanwhile the system architecture and the principle of data integration are given. Hash-AVL tree is adopted to describe the production data. XML is applied in the method to unify data access interface. The method is used in the automation system of a petrochemical enterprise. For the 100 data points in the concurrent test, the updating period is less than 3 s. Results show that the method of real-time data integration can integrate heterogeneous real-time data and meet the demands of process industry.

【Key words】real-time data; data integration; XML technology; Hash-AVL tree

1 概述

很多流程工业企业存在多种生产装置并存的特点，同时流程工业对数据的实时性要求较高，这就给这些异构实时数据的集成和优化带来了很大的困难。目前在流程工业中生产装置数据集成所采用的方法主要有基于虚拟视图的方法、基于数据存储的方法和基于设备访问的方法[1]，但上述几种主流方法都具有各自的特点和适用范围。实时数据库的出现在一定程度上解决了常规数据存储方法对于实时数据集成方面的不足，但由于国内外相关产品种类较多、功能单一、成本较高，并不非常适合解决当前的种种问题；以 OPC(OLE for Process and Control)技术为代表的基于设备访问的方法具有很好的性能表现，实时性好，接口标准统一、开放，但因为技术发展等原因使得 OPC 设备在一定时期内还不能完全取代传统设备，需要找到一种方式作为过渡来替代它，在经济性较好的基础上解决实时数据集成的难题。因此，本文针对目前流程工业的现状研究过程实时数据集成方法，提出一种合理、可行且成本低的新思路以解决企业信息化道路上所面临的数据集成难题。

2 分布式实时数据集成系统架构

在流程工业中，为了实现生产装置异构数据的集成，分布式系统是一种非常好的选择。而采用分布式系统，异构是不可避免的。本文提出了一种新的基于分布式结构过程数据的集成方法，如图 1 所示。系统的主要功能可以分为 2 个部分：一部分是数据采集，即对生产异构装置的实时数据的采集；另一部分是统一接口，即对不同数据源不同数据格式的统一封装，使得上层用户可以以统一的方式进行访问。

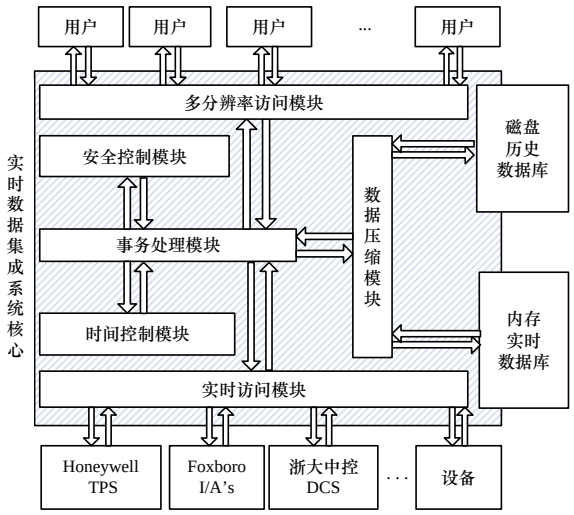

图 1 分布式数据集成系统架构

实时数据集成系统的核心在于将实时数据在设备及用户与核心接口模块的访问层中实现统一。这种方式屏蔽了装置的异构性，且涉及的层更少，降低系统的复杂性的同时有效

基金项目：国家自然科学基金资助项目(20676013)；北京市自然科学基金资助项目(4082022)
作者简介：薛尧予(1982－)，男，博士研究生，主研方向：企业综合自动化；王建林，教授、博士生导师；赵利强，博士研究生
收稿日期：2009-09-05 **E-mail：**wangjl@mail.buct.edu.cn

地提升了可靠性。由于在设备层与核心的接口模块中实现了数据的集成，因此，用户层有很好的灵活性以及扩展性，从而节省成本，降低应用开发的难度。

2.1 系统访问模块

用户通过多分辨率访问模块访问数据库，实时访问模块从设备读取实时数据。数据库分别为历史磁盘数据库和内存实时数据库，它们都是通过实时数据管理系统的接口软件来实现数据的存储、交换及访问。中间层即实时数据管理系统核心作为一个对象化和模块化模式的集合，统一完成数据的压缩、存储、访问等功能。

实时访问模块从异构装置采集数据，经事务处理模块、压缩模块存入内存实时数据库，用户可以由多分辨率访问模块访问实时数据(例如某一装置的实时曲线)，此实时数据即取自于内存实时数据库。对于历史数据，由数据压缩模块对内存实时数据库中的数据进行二次压缩存入磁盘历史数据库中。

2.2 数据压缩模块

数据压缩模块中的压缩算法对系统实时优化有至关重要的作用。文献[2]对化工实时数据采集、集中管理和压缩存储等问题进行了分析和研究，提出了一种增量型的 SDT 压缩算法，利用 SQL 数据库存储压缩数据，并利用 LZW 算法进行二次无损压缩，提高了存储的压缩效率。因此，压缩模块采用该文献中的压缩算法及策略。

3 基于Hash-AVL树的数据结构设计

3.1 Hash-AVL树

为了实现分布式异构实时数据的集成，如何设计数据结构来满足采用高效、快速的查询要求是一个关键的问题。Hash 表由于其速度快的优点在数据查询中有着广泛的应用[3]，只要能够较好地解决其自身的 Hash 冲突问题，便可以达到设计目标。对相同 Hash 值的数据查找若采用顺序查找方式，则效率非常低。目前数据库索引方法比较多，包括 B+-树、AVL 树和 SB-树等，但都不能很好地解决数据库的多键值查询问题。若采用 AVL 树来解决冲突，则对具有相同 Hash 值的数据的查找将变成 AVL 树的查找，比链式的顺序查找效率将提高很多。将 Hash 表和 AVL 树结合，采用复合结构能够兼顾 2 种数据结构的优点，具有非常高的查询效率。

AVL 树查找较快，但插入、删除时，由于要调整树的形态而效率较低。不过考虑到生产数据集成的实际情况，树的结构一旦形成，改动不会很频繁，改动的内容相对整个树来说也非常少。因此，采用 Hash-AVL 树能够很好地满足实际需要。用 C 语言结构体来描述 Hash-AVL 树，代码如下：

```
Typedef BINTREEBASENODE AVLTREENODE;
typedef struct HASH-AVLTREE_st {
AVLTREENODE **ppHash 表元;      /* 索引表指针 */
UINT uHash 表元 Count;              /* 索引表的大小 */
UINT uNodeCount;                  /* 表中实际节点的个数 */
UINT uCurHash 表元 No;    /* 当前要执行的 Hash 表元序号 */
AVLTREENODE *pCurEntry;
/* 当前 Hash 表元中下一个要执行的节点条目 */
} HASH-AVLTREE;
```

可见，Hash-AVL 树和 Hash 表的唯一区别就是 Hash-AVL 树中的每个 Hash 表元指向的是一棵 AVL 树。Hash-AVL 树的结构如图 2 所示。

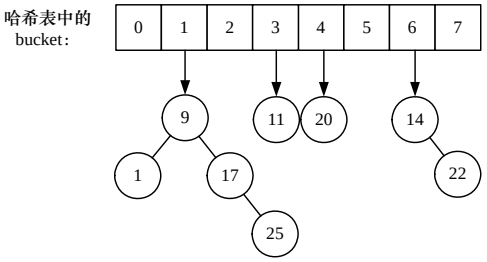

图 2 Hash-AVL 树的示意图

3.2 设计实现

Hash-AVL 树的基本的 XML 结构描述如下：

```
<hash_avl_tree_node id = n>
<avl_tree right_node = id, left_node = id, parent_node = id, node_count = n>balance</avl_tree>
<hash_table right_node = id, left_node = id, parent_node = id, node_count = n>balance</hash_table>
<node_data>data</node_date>
</hash_avl_tree_node>
```

XML 文档中包括了几个关键属性。其中，rigth_node 是指右节点的 id；left_node 为左节点的 id；parent_id 是父节点 id；node_count 为树的节点的个数。

此外，文档中还有 2 个关键字，balance 即为前文中提到的 AVL 树的平衡因子。data 是一个广义的概念，在这里代表数据。数据可以是实时数据，也可以是虚拟视图的数据，而这些数据都是以 XML 的方式进行统一封装的，易于集成。通过 XML 方式进行描述，其结构如下：

```
<Workstation Name=CY>
<Tag_ID id=TI-101>
<Location>192.168.168.1</Location>
<Constrain>every</Constrain>
<Period>2</Period>
<Value>248.062</Value>
<Unit>℃</Unit>
<Description>闪顶油温</Description>
<LowAlarm>100.0</LowAlarm>
<HighAlarm>500.0</HighAlarm>
</Tag_ID>
<Tag_ID id=TI-107>
<Location>192.168.168.1</Location>
<Constrain>every</Constrain>
<Period>1</Period>
<Value>112.617</Value>
<Unit>℃</Unit>
<Description>炉 101 对流段温度</Description>
<LowAlarm>100.0</LowAlarm>
<HighAlarm>500.0</HighAlarm>
</Tag_ID>
…
<Tag_ID id=TI-125>
…
</Tag_ID>
```

4 基于XML-RL的实时数据查询方法

实时数据查询技术在工业企业信息平台中具有广泛的用途。XML 技术标准的出现，使得能够实现各子系统数据的统一描述[4]。

XML-RL(XML Rulebased Query Language)是一种基于规则的 XML 查询语言。它以一种很自然的方式把 XML 文档看成是复杂对象数据模型。XML-RL 语言的查询语句由 2 个部分组成:

(1)查询子句，该部分是以规则为基础的路径表达式，被用来从 XML 文档提取数据。

(2)构造子句被用来构造查询结果。

结合实际需要，查询不仅需要返回正确的结果，还必须在一定的时间内完成，否则实时数据便失去其本来的意义。因此，对查询语句的结构进行扩展，在其原有结构的尾部加入实时性约束，其扩展后的结构描述如下:

querying [exp1[, exp2[, … [, expN]…]]] constructing [expC] {now|when|before|every} [time]

结构中扩展的部分主要是加入了几个关键字来对不同的查询请求进行实时约束。此外，time 是一个可选项，用来对几个关键字进行具体说明，它可以是一个时刻，也可以是一个时间间隔。下面对各个关键字逐一进行解释说明。

now——即收即发

说明: 对实时性要求最高的约束，当查询语句中有此关键字作为约束条件时，将会把该语句的优先级设定为最高，并将其插入到查询队列最靠前的位置。数据工作站接到此查询命令时立即返回满足查询要求的实时数据。因为是即时的发送，所以在这里不需要 time 项。

when——指定时刻发送

说明: 数据工作站接到此查询命令时，在请求时刻到达时准时返回满足查询要求的实时数据。这里 time 是指某一特定时刻。

before——指定时刻前发送

说明: 数据工作站接到此查询命令时，根据当前的闲忙情况，在请求时刻到达之前返回满足查询要求的实时数据。这里 time 是指某一特定时刻。

every——固定间隔时间发送

说明: 定义此种查询请求为优先级最低，当收到包含此关键字查询语句时，将此语句的优先级设定为最低，排入查询队列。此种查询方式属于一次请求多次应答的方式，在很大程度上提高了数据的查询效率。此语句执行时，数据工作站将会根据查询请求中设定的固定时间间隔，定期地向客户端发送查询的相关数据实时值。

因为此类操作是周期循环不间断的，为了保证其他操作的进行，需要在特定的情况下终止其操作。定义当其他 3 种约束的任何一种出现时，此类操作终止，直到空闲时，并且有新的该种类型查询请求出现时进行相应响应操作。在这里，time 是一个时间间隔，可以理解为传送数据的周期。

可以看出，所有数据查询请求可以分为 2 类: 一类是非周期任务，前 3 种实时约束的请求即为此种类型；另一类是周期任务，实时约束的关键字为“every”的请求属于此种类型。在实际的执行过程中，系统将整个查询语言分解为 2 个部分，一部分是基础的查询部分，另一部分是扩展的实时约束部分。查询部分仍然遵守 XML-RL 的规则，而实时约束部分则作为对其的约束，主要是为了确保数据不仅逻辑上正确而且能够满足用户对其的实时性要求。

5 系统应用

在某石化炼油厂生产数据实时监测与企业综合自动化系统项目中，利用本文所述的分布式过程实时数据集成方法实现了全厂生产异构数据集成。系统数据集成的实时性由多个因素所限制，对每个因素进行分解，分别研究各个因素对数据访问实时性的影响。对于装置访问，主要记录数据从 OPC 服务器到 OPC 客户端的传输延时；对于数据集成系统查询，主要记录 OPC 客户端更新内存中数据表以及系统处理带来的延时；对于网络传输，记录数据从工作站端发送到访问端接收之间的延时；对于上层访问，记录上层应用所带来的延时。数据访问实时性如表 1 所示。系统性能指标如表 2 所示。

表 1 数据访问实时性

限制因素	延时/ms	准确性
装置访问	不大于 500	准确
数据集成系统查询	不大于 500	准确
网络传输	不大于 1 000	准确
上层访问	不大于 1 000	准确

表 2 系统性能指标

并发连接数/个	数据更新周期/s	数据量/个
200	3	100

现场应用表明，本系统能够集成流程工业中生产装置异构实时数据，满足实际应用中对于数据实时性的要求，提供的数据准确有效，且运行可靠。而且这些数据量对于系统的运行效率影响不大，可以将数据点数量进行大幅度增加。

6 结束语

本文提出的分布式实时数据集成方法采用 Hash-AVL 树的数据结构描述、XML 数据访问封装，能够很好地实现生产装置异构实时数据的集成，有效地实现对生产装置异构实时数据的管理。

该方法为解决目前流程工业普遍存在的生产装置异构数据集成的难题提供了一种新的解决方法。

参考文献

[1] Lam Kam-Yiu, Kuo Tei-Wei, Lee T S H. Strategies for Resolving Inter-class Data Conflicts in Mixed Real-time Database Systems[J]. The Journal of Systems and Software, 2002, 61(1): 1-14.

[2] 赵利强, 于 涛, 王建林. 基于 SQL 数据库的过程数据压缩方法[J]. 计算机工程, 2008, 34(14): 58-62.

[3] 马如林, 蒋 华, 张庆霞. 一种哈希表快速查找的改进方法[J]. 计算机工程与科学, 2008, 30(9): 66-68.

[4] 张 晶, 张云生. 基于 XML 的实时数据一致性描述与查询处理[J]. 计算机工程, 2007, 33(10): 52-54.

编辑 顾逸斐