

# 实时数据库主动计算的设计与实现

王耀飞, 李林, 康卫, 徐新国  
(华北计算机系统工程研究所, 北京 100083)

**摘要:** 针对工业生产过程的优化控制和历史数据查询中的计算需求, 结合实时数据库数据采集和数据存储, 设计并实现了实时数据库的主动计算。在实时数据库中使用主动计算, 既可以实时获取计算结果值进行实时的优化控制, 也可以在后续条件查询中直接获取计算结果, 从而大大提高查询效率。

**关键词:** 实时数据库; 主动计算; 条件触发; 时间触发

中图分类号: TP311 文献标识码: A 文章编号: 0258-7998(2014)08-0133-03

## Research and implementation of proactive computing for the real-time database

Wang Yaofei, Li Lin, Kang Wei, Xu Xinguo  
(National Computer System Engineering Research Institute of China, Beijing 100083, China)

**Abstract:** For the computing needs of optimization control and historical query in the industrial, this paper proposes the real-time database proactive computing, which is combined with real-time data acquisition and storage. Using proactive computing in the real-time database, we can not only obtain the results in real time for control, but also obtain the results directly in the historical query, which greatly improving the efficiency of historical query.

**Key words:** real-time database; proactive computing; condition triggered; time triggered

实时数据库作为工业控制系统的核心, 肩负着数据采集、报警、转换和压缩存储等重任, 为工业现场的稳定运行提供了统一的数据平台<sup>[1]</sup>。然而, 在实时数据库的应用场景中, 不仅需要实时给出工厂运行时从数据采集点采集到的实时数据, 而且需要及时准确地对机组生产运行的实时状态参数进行筛选、计算、分析, 形成设备、系统、机组和全厂的性能参数<sup>[2]</sup>; 同时, 在工厂后期决策中, 历史数据的查询与各种参数的计算既消耗网络带宽, 又非常耗时, 具有一定的滞后性<sup>[3]</sup>。

针对上述情况, 在实时数据库中引入主动计算, 它吸取了主动实时数据库<sup>[4]</sup>的设计思想, 自动、实时地满足工厂在生产控制中的计算需求。在历史数据的查询和计算时, 只要条件满足, 无需查询大量的原始数据, 直接查询计算结果即可, 为工厂的经济运行提供高效的数据支撑。

### 1 研究现状

随着实时数据库理论的日益成熟和完善, 实时数据库产品也逐渐发展完善, 有些产品中逐渐包含了主动计算的相关内容, 例如浙大中控的 ESP-iSYS 实时数据库和美国 OSI 公司的 PI(Plant Information System)。

ESP-iSYS 实时数据库中内嵌了用于实时计算的虚

拟机引擎, 可以轻松添加虚拟的计算位号, 支持在位号读、写和定时等多种触发机制, 从而提供了强大的实时计算和实时信息处理的功能。而且, ESP-iSYS 提供了支持实时计算业务的函数库, 通过一行或数行 VB 语法的脚本, 就可以实现丰富的业务功能。

PI 实时数据库中包含一个高级计算引擎 ACE(Advanced Computing Engine)。ACE 由三大基本组件组成, 分别是 ACE Wizard、ACE Manager 和 ACE Scheduler。其中, ACE Wizard 是 Visual Basic 插件, 可帮助用户快速、高效地创建 ACE 模块; ACE Manager 允许用户监控并更改 ACE 模块的各种属性; ACE Scheduler 作为一种 NT 服务运行, 可及时执行 ACE 模块并处理更新和异常行为。另外, PI ACE 为工厂运转和企业评估中的方方面面开发了标准的计算公式, 可跨越时区和地域的界限, 快速解决问题。

### 2 系统设计

#### 2.1 数学描述

##### 2.1.1 整体公式

在实时数据库的主动计算中, 为了统一调度和管理各种可能用到的计算, 用以下公式表示一个计算:

$$Y=f(X, K, S, R, C) \quad (1)$$

其中,  $f$  代表主动计算的计算函数,  $Y$  代表计算结果,  $X$  代表计算变量,  $K$  代表常量,  $S$  代表触发方式,  $R$  代表结果处理方式,  $C$  代表触发公式所满足的逻辑条件, 可以为空。通过这样的描述, 使每个计算都统一到式(1)中, 每个计算都至少包含上述的 5 个要素, 这样可以极大地方便后续计算的处理。

### 2.1.2 输入

$$X = D || Y' \quad (2)$$

输入变量既可以是实时数据库中的实时数据 (用  $D$  表示), 也可以是主动计算的计算结果 (用  $Y'$  表示)。

### 2.1.3 触发方式

$$S = T || U \quad (3)$$

触发方式  $S$  包括固定时间周期触发  $T$  和点更新触发  $U$  两种, 如式(3)所示。而点更新触发  $U$  包括单点更新触发 (Single)、多点“或”的关系触发 (Or) 和多点“与”的关系触发 (And) 3 种, 如式(4)所示:

$$U = \text{Single} || \text{Or} || \text{And} \quad (4)$$

### 2.1.4 结果处理方式

对于式(1)中的结果处理方式  $R$ , 包括实时反馈 (RT-Feedback); 作为中间结果参与到其他计算公式中 (Intermediate), 即式(3)中的  $Y'$ ; 存储到历史数据中 (Store)。如式(5)所示:

$$R = \text{RTFeedback} || \text{Intermediate} || \text{Store} \quad (5)$$

### 2.1.5 查询

$$Y = f(C) \quad (6)$$

其中,  $Y$  表示查询结果,  $f$  表示查询函数,  $C$  表示查询条件。对于满足主动计算的查询条件, 则直接查询主动计算的结果; 对于不满足主动计算的查询条件, 则查询原始数据, 经计算后返回计算结果。

## 2.2 系统结构

实时数据库主动计算的系统结构如图 1 所示。其运行流程如下: (1) 计算公式和触发方式等通过公式管理器输入到实时数据库中; (2) 通过预处理器将公式、触发方式等进行初始化; (3) 启动数据更新器、条件触发器和公式计算器, 三者并行执行, 完成主动计算功能并将计算结果存入磁盘; (4) 当用户需要统计、计算、决策时, 无需查询大量的原始历史数据进行计算, 直接查询已经计算好的数据即可。

## 3 系统实现

系统实现主要分为公式管理器、预处理器、数据更新器、条件触发器和公式计算器 5 部分。

### 3.1 公式管理器

用户通过计算公式管理器实现公式的输入、查询、删除、更改等操作。每个公式的数据结构如下:

```
typedef struct active_computer_data
{
    int cpID; //主动计算结果标识
    int ScheduleMode; //触发方式
```

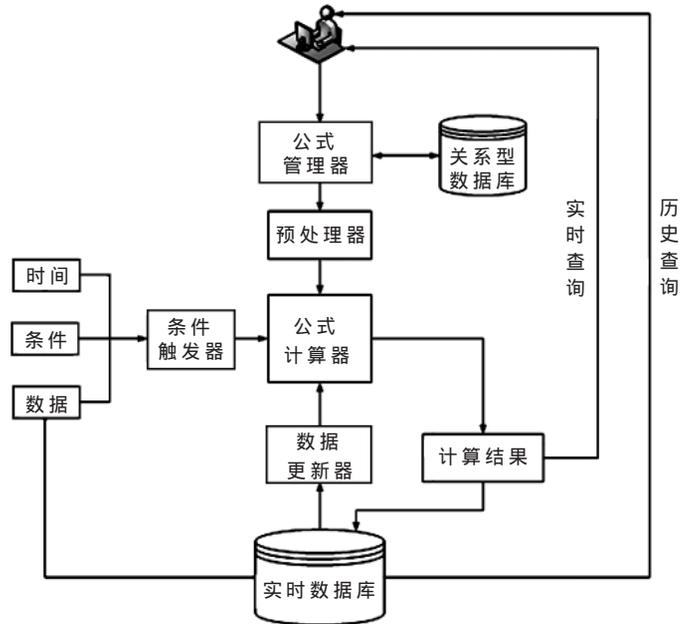


图 1 实时数据库主动计算系统结构图

```
char Expression[256]; //计算表达式
int ScheduleData[32]; //调度信息
int SaveResult; //结果处理方式
}ExpData;
```

其中,  $cpID$  是主动计算结果在实时数据库中的唯一标识;  $ScheduleMode$  包含单点触发、多点触发和固定周期触发;  $Expression$  中, 实时变量使用点号表示, 例如  $_8_$  表示点号为 8 的点参与运算;  $ScheduleData$  表示调度信息, 当触发方式为单点或者多点触发时,  $ScheduleData[0]$  存放表达式中参与运算的点的总数, 从  $ScheduleData[1]$  开始顺序存放点 ID; 当触发方式为固定周期触发时,  $ScheduleData[0]$  存放的是周期, 单位是  $s$ ;  $SaveResult$  存放的是结果处理方式。公式管理器对公式的所有操作实际上是对上述结构的操作。

### 3.2 预处理器

如图 2 所示, 预处理器读取用户输入的每个计算表达式, 将表达式中的所有变量点的 ID 记录于数组  $SchedulePtoPID$  中, 变量的个数记为  $N$ 。建立大小为  $M \times N$  的位图表  $Schedule\_bitmap$ , 其中  $M$  表示公式的个数, 初始

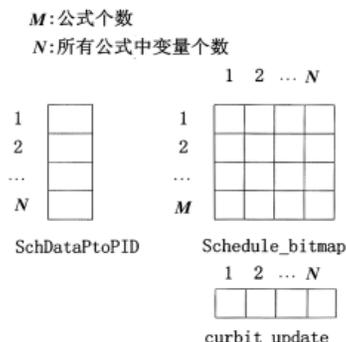


图 2 预处理器的映射操作

化 Schedule\_bitmap, 使得每个公式中变量点所对应的位为 1, 其余位均为 0, 为条件触发的条件判断做准备。

### 3.3 数据更新器

主动计算中的计算变量点直接从实时数据库的内存中获取, 数据更新器的作用是扫描实时数据库的数据采集单元, 若有计算变量点的更新, 则将更新位图 curbit\_update 的相应位置 1, 供条件触发器进行相应的条件判断。

### 3.4 条件触发器

这是主动计算的核心部分, 包括两种方式, 分别是点更新事件触发和固定时间周期触发。

#### 3.4.1 点更新事件触发

点更新事件触发包括单点更新触发、多点或关系触发和多点与关系触发。由于单点更新触发实质上是一种多点或关系触发, 所以在处理点更新触发时, 只需要处理多点或关系触发和多点与关系触发。两种触发方式都是用点更新位图 curbit\_update 与原始位图 Schedule\_bitmap 进行逐项与操作, 如图 3 所示, 然后进行各自的判断。

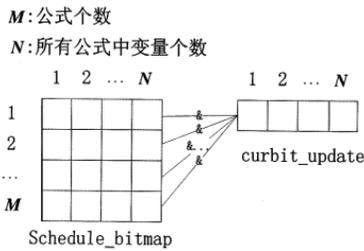


图 3 点更新事件触发示意图

多点或关系触发, 具体表现是更新位图 curbit\_update 与原始位图 Schedule\_bitmap 进行与操作后, 若结果不为 0, 则触发多点或关系的计算, 将计算表达式加入计算队列。

多点与关系触发具体表现是更新位图 curbit\_update 与原始位图 Schedule\_bitmap 进行与操作后, 若结果为 Schedule\_bitmap 对应项, 则触发多点与关系的计算, 将计算表达式加入计算队列。

#### 3.4.2 固定时间周期触发

对于固定时间周期触发方式, 为了减少资源消耗, 本文设计了一个时间刷新器刷新所有需要固定时间周期触发的计算表达式, 这样有利于统一操作<sup>[5]</sup>。

如图 4 所示为固定时间周期触发的时域图, 横坐标表示时间, 两个纵坐标之间表示固定时间周期触发的一个执行周期。由图可知, 固定时间周期触发的一个执行周期包含刷新、执行和休眠。刷新过程在每个时间周期扫描所有公式, 将对应公式的时间记录减 1; 执行周期扫描所有公式, 将时间记录为 0 的公式添加到计算表达式执行队列, 并将时间记录重新置为公式的执行周期。

### 3.5 公式计算器

公式计算器扫描计算表达式执行队列, 若队列不为空, 则逐个取出公式, 根据计算公式和给定的计算变量计

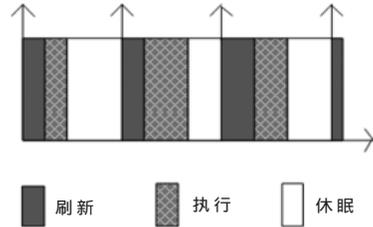


图 4 固定时间周期触发的时域图

算出用户需要的结果并存入磁盘, 方便用户的后续查询。

## 4 实验及结果

实时数据库主动计算都是在 Linux 下开发的。开发完成后, 在一台主频 2.93 GHz、内存 2.0 GB 的主机上, 模拟火电厂 SIS 系统中的性能计算来对主动计算进行功能测试和性能测试。功能测试主要内容是测试主动计算的各个驱动方式以及各种公式的计算是否正确。性能测试主要从查询时间和资源消耗两方面来说明, 测试条件查询数目分别为 5 个级别(0、50、100、150、200)情况下的系统资源消耗和查询时间。资源消耗情况如表 1 所示, 在测试查询时间时, 所有的主动计算驱动均为按照固定周期驱动, 主动计算公式均为两个点相加的简单运算, 然后对比有主动计算后期查询时间和无主动计算的后期查询与运算时间。时间对比如图 5 所示。

表 1 主动计算内存消耗

历史条件查询数目	0	50	100	150	200
内存消耗/KB	252	328	433	543	647

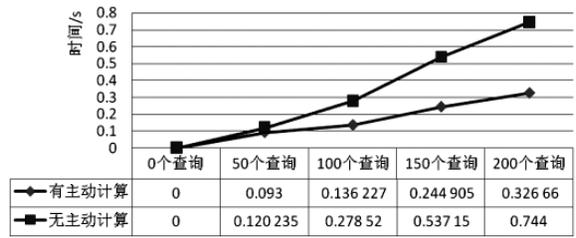


图 5 有无主动计算查询时间对比图

通过功能测试, 分别验证了主动计算的采集点更新驱动和时间周期驱动的功能, 满足主动计算的需求。在性能上, 主动计算对内存的消耗很小, 对实时数据库的其他功能影响不大; 后续查询中添加了主动计算后, 查询计算结果的时间明显小于无主动计算的情况, 大大提高了企业对于计算结果的查询速度。

主动计算系统最大的优点在于在几乎不影响实时数据库性能的同时, 可以实时主动计算出工厂所需的数据提供给用户, 在历史数据查询与计算中解决了数据后期处理的滞后性, 可有效提高工厂决断效率, 适用于火电厂、造纸厂、制药厂等所有使用实时数据库的行业。

### 参考文献

[1] 徐新国, 朱廷劭, 康卫, 等. 基于数据挖掘的工业控制系统防危机制研究[J]. 电子技术应用, 2012, 38(5): 87-90.

(下转第 139 页)

~~~~~  
(上接第 135 页)

- [2] 钱笑宇,张彦武.工业实时数据库的研究和设计[J].计算机工程,2005,31(1):98-99,132.
- [3] 李健, Ji Guorui, 焦嵩鸣, 等.一种火电机组实时性能计算引擎的设计[J].仪器仪表与分析监测,2008(3):11-13.
- [4] 李红娇.主动实时数据库管理系统的主动机制[D].武汉:华中科技大学,2002.
- [5] 郭丽娟,刘双与,张激,等.基于时间触发的高可靠性实时系统架构[J].计算机工程,2006,32(4):272-274.

(收稿日期:2014-04-18)

---

作者简介:

王耀飞,男,1988年生,硕士研究生,主要研究方向:实时数据库。

李林,男,1970年生,高级工程师,主要研究方向:工业自动化、工控安全。

徐新国,男,1966年生,博士,研究员,主要研究方向:实时控制、工控安全。