

# 基于 COM+ 技术的储罐计量

郁汉琪<sup>1</sup>, 徐德洪<sup>2</sup>

(1. 南京工程学院工程实训中心, 南京 211167; 2. 南京工业职业技术学院机械系, 南京 210016)

**摘要:** 储罐计量是计量管理的核心, 计量算法的软件机制的实现尤为重要。该文提出了基于 COM+ 技术开发计量软件的策略, 利用基于哈希表的方法优化数据, 采取了 3 层架构部署软件方案, 解决了早期计量软件存在的实时性不强、复用性较差的问题。实践表明了该计量软件的有效性。

**关键词:** 罐容表; COM+; 3 层架构; 储罐计量

## Calculation of Storage Tank Based on COM + Technology

YU Han-qi<sup>1</sup>, XU De-hong<sup>2</sup>

(1. Engineering Training Center, Nanjing Institute of Technology, Nanjing 211167;

2. Department of Mechanism, Nanjing Institute of Industry Technology, Nanjing 210016)

**【Abstract】** The calculation of storage tank is the core of measurement management. It is important to implement the software mechanism for calculation algorithm. This paper puts forward the strategy based on COM+ technology. Data are optimized by the method based on Hash table. The scheme of software deployment on three-tier structure is put forward, which makes up for defects of the previous software, such as bad real-time and reusability. Practices show the calculation software is effective.

**【Key words】** volume table for tank; COM+; three-tier structure; calculation of storage tank

当今, 精细的生产、油品计量管理成为了炼油企业生产与储运管理的重要课题, 储罐计量是计量管理的核心。储罐计量是炼油企业生产流程的重要组成部分, 储罐管理和罐量计算的准确性直接关系到计量的工作效率。储罐计量包括油罐的油尺、水尺、温度、压力、密度数据。根据罐容表和罐的基本参数计算储罐的存油量。罐容表结构、罐的类型和物料种类决定了储罐计量的计算模型, 罐的检定单位所采用的检定方法不同, 导致了罐容表结构的多样化。

目前, 储罐计量的软件分为单机版和网络版。单机版已无法适应多用户在线使用的需求, 并且无法实现罐容表的统一管理和维护。网络版是基于 B/S 结构的, 满足了多用户的需要, 但是受制于网络, 网页的刷新速度难以满足计量数据的实时性需求。尤为重要的是, 企业在线运行的许多软件需要实时的计量数据, 网络版不能满足软件的复用性需求, 无法与这些软件交互数据。

### 1 储罐计量计算模型及其实现策略

储罐计量的基础数据是依据某种检定方法生成的罐容表: (1) 根据罐容表, 按照罐的类别, 生成视体积( $V_t$ ); (2) 根据物料类别和国标 GB1885 有关表, 生成标准体积( $V_{20}$ ); (3) 综合罐类别和物料类别, 生成罐量(含液相量和气相量)。

#### 1.1 罐容表的产生

以立罐为例, 检定单位测量罐的圈板直径、各圈板高度及其板厚、罐底量、浮顶量及其参照高度, 经过一定的数据处理方法, 生成罐的基本参数表、厘米表、小数表及其静压力修正表。例如采用几何法计算底量的算法为

$$V_{底} = \frac{\pi d^2}{4} \times \left\{ \frac{1}{3k} \sum_1^n H_0(h, B_0, B_{基准}) + \frac{7}{6k} \sum_1^n H_1(h, B_1, B_{基准}) + \frac{1}{k} \sum_1^n H_2(h, B_2, B_{基准}) + \dots + \frac{1}{k} \sum_1^n H_{m-1}(h, B_{m-1}, B_{基准}) + \frac{1}{2k} \sum_1^n H_m(h, B_m, B_{基准}) \right\} \times 10^{-6}$$

其中,  $B_0, B_1, \dots, B_m$  为各测量点标高;  $B_{基准}$  为下计量基准点标高;  $H(h, B, B_{基准})$  为自定义函数, 即

$$H(h, B, B_{基准}) = \begin{cases} B+h-B_{基准} & (B+h > B_{基准}) \\ 0 & (B+h \leq B_{基准}) \end{cases}$$

计量的计算流程见图 1。

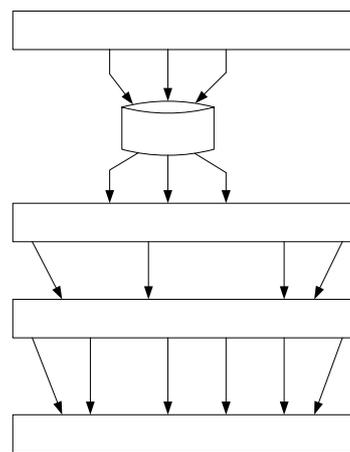


图 1 计量的计算流程

#### 1.2 油品(VCF)方法计算 $V_{20}$

$V_{20}$  的计算公式为  $V_{20} = V_t * V_{cf}$ 。关键是求解石油体积温度系数  $V_{cf}$ 。  $V_{cf}$  根据标准密度 ( $P_{20}$ ), 通过双线性插值法查 GB1885-83 中的石油体积温度系数表得到分产品部分、原油

**作者简介:** 郁汉琪(1961 -), 男, 副教授、高级工程师, 主研方向: 自动化控制, 信息集成; 徐德洪, 硕士研究生、讲师

**收稿日期:** 2007-05-28 **E-mail:** hanqiyu2003@yahoo.com.cn

和润滑油 3 种表数据。P20 根据视密度( $P_t$ ),通过双线性插值法查GB1885-83 中的石油视密度换算表得到。

### 1.2.1 双线性插值法查表计算

无论是石油体积温度系数表,还是石油视密度换算表,都是二维表,计算方法均采用双线性插值法。表 1 为石油视密度换算表的一部分。当温度为 32.7、视密度为 0.7545g/cm<sup>3</sup>时,密度尾数修正值为

$$\frac{0.7642 - 0.7632}{0.7550 - 0.7540} * (0.7545 - 0.7540) = 0.0005 \text{ g/cm}^3$$

温度尾数修正值(近似处理)为

$$\frac{0.7649 - 0.7642}{33.0 - 32.0} * (32.7 - 32.0) = 0.0005 \text{ g/cm}^3$$

查表得到,当温度为 32.7、视密度为 0.7545g/cm<sup>3</sup>时,表载数据为 0.763 2 g/cm<sup>3</sup>,由此得到标准密度(20 下)为 0.7632+0.0005+0.0005=0.7642g/cm<sup>3</sup>

表 1 石油视密度换算表(部分)

温度( )	视密度/(g/cm <sup>3</sup> )	密度P20/(g/cm <sup>3</sup> )
31.0	0.754 0	--
32.0	0.754 0	0.763 2
33.0	0.754 0	--
31.0	0.755 0	0.763 4
32.0	0.755 0	0.764 2
33.0	0.755 0	0.764 9
31.0	0.756 0	--
32.0	0.756 0	0.765 2
33.0	0.756 0	--

### 1.2.2 基于哈希表的 V20 的算法实现

分析表 1,根据  $P_t$ 和温度  $t$ 计算  $P_{20}$  必须查表得到  $\{P_t, t_1: P_{20}; P_t, t_2: P_{20}\}$  数据,再根据双线性插值法计算。另外,此二维表有 672 行温度,数据从 -18 ~ 150,间隔为 0.25;有 212 列视密度数据。油品分润滑油、原油、产品 3 个部分,每一部分表结构相似于表 1。

本文采用基于哈希表(图 2 所示)的算法压缩、提取数据,软件设计成动态链接库 DLL(sybdll.dll),供第 3 方软件调用。

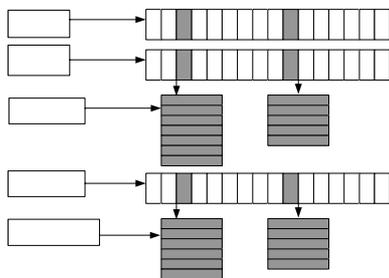


图 2 基于哈希表的内存结构

本文设计成 3 个哈希表,基于 C++ 类来实现:Check 用的是 HashSet 类,Value 和 compitable 使用的是 HashMap 类。算法具体描述为:使用 HashSet 录入数据优化方法,使用 HashMap 2 个功能<sup>[1]</sup>:(1)录入 Value,即原始数据;(2)根据映射规则及 Value,根据表号及其温度产生索引。

表 1 的数据压缩方法为:哈希表 Key 为温度数据(不难析出,其具有规律性和唯一性),将每列的 4 位小数点精度的浮点数转换为 ASCII 码字符,构成字符串,再将此字符串的 ASCII 码变成 1 和 0 序列,这样原字符串压缩成,如...0 + 0 的个数 + 1 + 1 的个数+...

采用 VC++ 开发的动态链接库 DLL<sup>[2]</sup>的接口函数有 2 个,分别为:

```
(1)extern "C" _declspec(dllexport) double ptsuanp20(double pt, double t1, int oiltype)
```

其中,  $pt$  为视密度;  $t1$  为视温度;计算标准密度为  $p20$ ;  $oiltype$

为油品类型;  $oiltype=0$  表示产品部分;  $oiltype=1$  表示为润滑油;  $oiltype=2$  表示为原油;  $ptsuanp20$  为标准密度  $P_{20}$ 。

```
(2)extern "C" _declspec(dllexport) double p20suanvcf(double p20, double t2, int oiltype)
```

其中,  $p20$  表示标准密度;  $t2$  表示油温;  $p20suanvcf$  为体积系数  $vcf$  值。

本方案和早期的量算软件的测试的对比见表 2。结果表明,采用本文的方案性能较强。

表 2 几种方案的对比

方法	程序运行数据来源	文件大小(6 个表)/MB	V20(以产品部分为例)时间/ ms	维护性
采用 excel 存储数据方式	硬盘	23.511	8 256	维护数据
采用关系数据库(FoxPro)方式	硬盘	5.168	196	维护数据
本文方案	内存	1.46	null	维护程序代码

### 1.3 罐量的计算

根据罐类型,内浮顶立罐、外浮顶立罐、球罐、槽车的罐量计算不同,其中,球罐的计算量为液相量和气相量之和;根据物料类型,油品、化工品、低温乙烯、丙烯、酸、碱的罐量计算也不同。本文的软件采用关系数据库存储分类码及其分类码对应算法公式方案,新的计算类型出现时,只需要在数据库定义分类码及其算法的正则表达式即可,通过用户界面即可实现,不需要更改软件代码,使得软件具有较强的配置性和复用性。

## 2 COM+ 技术的研究

COM+ 倡导了一种新的概念,它把 COM 组件<sup>[3]</sup>软件提升到应用层而不再是底层的软件结构,它通过操作系统的各种支持,使组件对象模型建立在应用层上,把所有组件的底层细节留给操作系统。COM+ 集成了 COM, DCOM 和 MTS(microsoft transaction server)技术,目前 COM+ 技术已成为 Windows DNA(distributed Internet application architecture)策略的核心。

### 2.1 COM+ 的基本结构

COM+ 不仅继承了 COM, DCOM, MTS 的许多特性,同时也新增了一些服务,比如负载平衡、内存数据库、事件模型、队列服务等。COM+ 新增的服务为 COM+ 应用提供了很强的功能,建立在 COM+ 基础上的应用程序可以直接利用这些服务而获得良好的企业应用特性,基本结构见图 3。

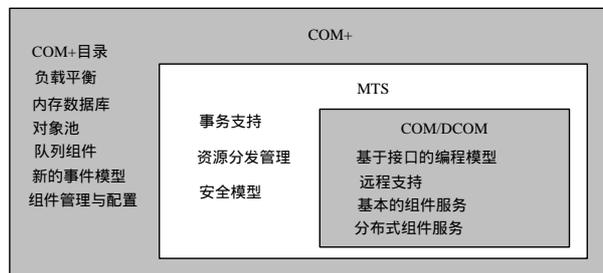


图 3 COM+ 的基本结构

### 2.2 COM+ 的应用开发

和 MTS 组件一样,COM+ 也要求对象必须以动态链接库(DLL)组件的形式提供。在 VC++ 中可以通过 ATL COM AppWizard 创建。在编译时刻程序员可以在代码中使用一些说明性的语句来设置 COM+ 组件的属性,比如 CLSID、ProgID、线程模型以及双接口等,如果不指定这些属性,编译器将使用缺省值。编译器的结构见图 4。

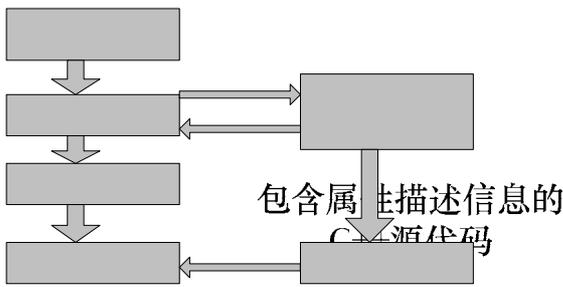


图4 编译器的结构

### 3 基于 COM+的储罐算量软件的总体方案

该软件应用于分布式环境，采用基于COM+技术的3层结构部署方案<sup>[4]</sup>，即表现层、业务层和数据层，见图5。

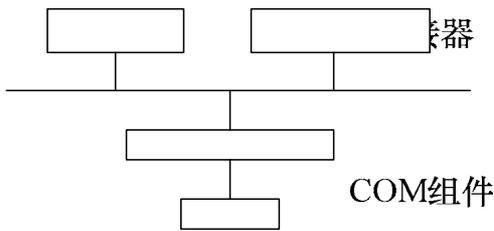


图5 3层结构部署

#### 3.1 数据层实现

数据层存储数据，有关系数据库和二进制文件2种。由于罐容表、罐基本信息、管理信息等内容各企业不同，因此设计好的数据库会将这类数据存储。石油体积温度系数表及其密度换算表的数据由国标产生，数据基本不变，封装在DLL中，方法在上面已阐述。

#### 3.2 业务逻辑层实现

业务逻辑层即为储罐算量的核心，采用VC++开发，基于COM+技术实现。图6为算量的计算流程(图1)开发的COM+组件逻辑<sup>[5]</sup>。

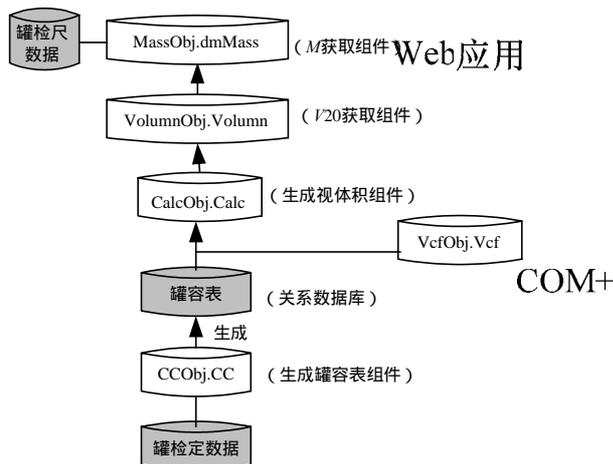


图6 COM+组件逻辑

如 MassObj.dmMass 组件的函数 M 接口定义为：

```
HRESULT M([in] BSTR * sGH, [in] Double fOLevel, [in] Double fWLevel, [in] Double fMD, [in] Double fTy, [in] Double fTq, [in] Double fPq, [out] Double fV20, [out] Double fMy, [out] Double fMq, [out,retval] BSTR * pbstrResult)
```

其中，fV20为计算出的标准体积V<sub>20</sub>；fMy为计算出的液相量；fMq为计算出的气相量；pbstrResult为计算成功标志位。

#### 3.3 表现层实现

表现层也称为用户界面，即为调用储罐算量的第3方软件，如Web应用软件、客户端软件及其IFIX等组态软件。

以IFIX组态软件的VBA脚本调用MassObj.dmMass组件(分布式环境)举例，调用代码如下：

```
Dim obj As Object, ProgID As String, ss As Variant
ProgID = "MassObj.dmMass"
```

```
Set obj = CreateObject(ProgID)
```

属性信息  
bSucceed)

可以看出，表现层调用代码编制比较方便。实际上，COM+组件的安装部署在分布式环境下通过Windows系统完成，既方便又确定。

#### 3.4 方案主要优点

(1)真正的异步通信。COM+底层提供了队列组件服务，这使客户端程序和算量组件有可能在不同的时间点上协同工作，COM+应用不需增加代码就可以获得这样的特性。

(2)事件服务。新的事件机制使事件源和事件接收方实现事件功能更加灵活，利用系统服务类型信息。

(3)可伸缩性。动态负载平衡以及内存数据库、对象池等系统服务都为COM+的可伸缩性提供了技术基础，COM+的可伸缩性原理上与多层结构的可伸缩特性一致。

(4)可管理性和可配置性。管理和配置是应用系统开发完成后的行为，在软件维护成本不断增加的今天，组件策略将有助于软件厂商和用户减少这方面的投入。

(5)高度灵活性。采用算法对应代码的正则表达式，使得在增加算法时，只需在数据库组态对应内容，不需要更改组件代码即可实现计算。

(6)计算的高效性。采用基于哈希表的优化数据算法及其COM+技术部署软件，使得算量具有高效性。测试结果表明，在分布式环境下，计算球罐物料为原油所需时间仅为120ms，完全能满足在线计算的实时性要求。

### 4 结束语 客户端软件应用

COM+技术标志着Microsoft的组件技术达到了一个新的高度，它不再局限于一台机器上的桌面系统，它把目标指向了一个更为广阔的企业内部网，甚至指向Internet。COM+与多层结构模型以及Windows操作系统为企业应用或Web应用提供了一套完整的解决方案。部署于服务器端，解决了单机版和网络版的缺陷，适应了现代化计量信息管理的需求。以后，将重点考虑数据压缩的稳定性，安装COM+组件时其对系统的依赖性、安全性问题。

#### 参考文献

- 1 复钢强, 范强, 谢冬青. SPKI/SDSI 名字证书缩减闭包算法及改进[J]. 计算机工程与应用, 2005, 41(9).
- 2 孙鑫. Visual C++深入详解[M]. 北京: 电子工业出版社, 2006.
- 3 韩敏, 李天昊, 孟华, 等. 基于COM的湿地地理信息系统的设计[J]. 计算机工程, 2003, 29(15).
- 4 刘薇. 基于COM+技术的三层架构模型在大型信息系统中的应用[J]. 武汉理工大学学报(信息与管理工程版), 2004, 26(4).
- 5 余英, 梁刚. Visual C++实践与提高-COM和COM+篇[M]. 北京: 中国铁道出版社, 2001.