

伺服测试数据格式转换算法研究

谢伟, 张蕾, 兰天, 胡涟, 邱子涵

(北京精密机电控制设备研究所, 北京 100076)

摘要: 测试数据及其格式转换是计算机测控领域的一项重要研究内容, 对于伺服测试尤其如此; 伺服种类繁多, 不同种类测试数据差别很大, 需要格式转换; 某种类伺服测试软件采用测试与数据处理分开的框架结构, 数据处理采用已成熟应用的数据平台, 测试子系统全新设计; 两者底层数据格式不同, 测试子系统设计了数据转换模块进行格式转换; 但现有数据格式转换效率较低, 不能满足该种类快速判读以确定伺服产品状态的需求; 对伺服测试软件生成的原始数据格式、待转换的物理量数据格式、数据格式转换逻辑进行了研究, 提出了优化代码内存使用效率的方法, 设计了新的转换算法, 使转换效率有了质的提高; 并设计了全新可视化的数据格式转换工具在其他种类中推广应用; 由于该方法具有普遍性, 对类似问题具有借鉴意义。

关键词: 伺服测试数据; 数据格式转换算法; 内存使用; 转换效率

Research on Servo Test Data Format Conversion Algorithm

Xie Wei, Zhang Lei, Lan Tian, Hu Lian, Qiu Zihan

(Beijing Research Institute of Precise Mechanical and Electronic Control Equipment, Beijing 100076, China)

Abstract: Test Data and its data format conversion are an important aspect of computer test and control area, also to servo test area. There are so many servo types and so many test data intertype differences that the data format conversion is needed. Some servo test software selects the test and data processing separate frame structure, data processing subsystem uses the mature application data platform, test subsystem is redesigned. The two subsystems have different low data formats, the test subsystem has a special data format conversion module. But now it has low efficiency on data format conversion, and can't satisfy servo type needs for quickly justifying data to confirm servo products status. Research on the original data type, the final physical data format, and the data conversion logic, propose a new method on optimizing code memory usage efficiency, design a new conversion algorithm, thus has greatly improved the efficiency of converting works. And a new visual data format conversion tool has been designed and used in other servo types. The method has the common meaning, so analogous problems can be solved like it.

Keywords: servo test data; data format conversion algorithm; memory usage; conversion efficiency

0 引言

伺服机构系统是火箭控制系统的执行机构, 是火箭控制回路的重要环节。伺服机构的性能直接关系到火箭的控制稳定性, 因此伺服机构的测试非常重要^[1]。在伺服测试中软件设计是测试设备研制的一项关键内容; 目前所知, 软件系统一般采用测试、数据处理分开的框架结构^[2]。为了提高实时性和存储效率, 测试软件一般采用二进制形式存储测试原始数据; 数据处理软件要输出用户可读形式的文本数据; 另外不同种类测试数据要求也差别很大, 因此数据格式转换不可避免^[3]。

本文以某种类伺服测试软件为背景论述了测试原始数据转换成物理量格式所遇到的问题, 分析原因、研究大数据量情况下内存操作的优化方法、提出新的数据转换算法。

1 概述

出于数据存储实时性要求以及存储数据文件不能过大等条件的限制, 伺服测试软件在测试过程中生成的原始数据文件采用二进制格式, 后缀为. da。该格式的文件为内存数据结构硬

拷贝形式, 未经格式转换就直接将内存数据映射到数据文件上进行存储, 具有物理空间占用低的优点。在减少数据存储空间的同时, 极大缩短数据存储时间, 提高了测试效率。但是该类型的数据文件只能通过开发专门的数据格式转换软件, 将数据转换为文本格式(物理量数据格式是其中一种, 直接表示通道具有物理意义的数据内容)后, 才能用文本编辑器等其他常用数据分析和查看工具查看和分析。

在日常使用过程中软件操作人员, 如伺服总体人员和调试记录人员, 却经常需要将原始数据文件. da转换为. txt格式的文本文件^[4]进行数据查看和数据分析。

目前使用的. da到. txt格式的数据格式转换工具有两种: 一种为伺服测试软件本身自带的功能, 通过软件界面对转换文件逐个操作; 一种为控制台形式操作, 需要使用者使用操作系统的命令行操作, 输入相关转换命令完成数据格式转换, 操作不方便。另外, 关键问题是目前的转换算法执行效率很低, 特别是在进行大数据量转换时, 需要花费数个小时时间才能完成, 转换效率非常低。这不仅增加了数据使用人员的操作负担、影响他们进行快速的数据比对分析, 也降低工作效率, 影响了伺服异常情况下进行原因分析和定位工作的正常开展。

2 现有数据转换过程分析

以某种类伺服测试软件为基础, 对现有数据处理的数据格式转换过程进行梳理, 分析了其逻辑过程及实现手段, 重点考察了其内存操作情况。现有算法的数据转换过程分为以下几个

收稿日期:2016-01-09; 修回日期:2016-02-23。

基金项目:新一代运载火箭项目(C0100181001)。

作者简介:谢伟(1974-),男,山东济宁人,硕士研究生,高级工程师,伺服地面软件总设计师助理,主要从事航天伺服测试软件方向的研究。

基本步骤, 其内存利用情况如图 1 所示。

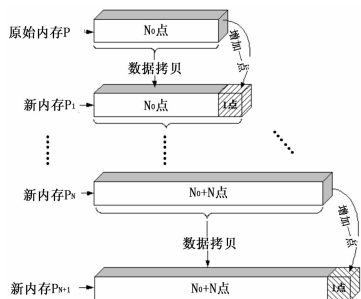


图 1 现有数据转换过程内存利用情况

- 1) 读取二进制文件到内存;
- 2) 开辟一块大内存 P;
- 3) 逐个转换每个二进制数据为字符串类型;
- 4) 将每个转换过的字符串追加到 P 的有效字符串尾;
- 5) 如果 P 大小不足以容纳新加入的字符串, 开辟更大快内存 P1, 将 P 内数据拷贝过去的同时, 将新加入的字符串追加到 P1 的有效字符串尾;
- 6) 若还有数据未转换, 继续对 P1 判断; 如果 P1 大小不足以容纳新加入的字符串, 开辟更大快内存 P2, 将 P1 内数据拷贝过去的同时, 将新加入的字符串追加到 P2 的有效字符串尾;
- 7) 如此循环执行。转换完毕后, 存储 Pn 数据到文本文件, 流程结束。

3 数据格式转换影响因素分析

在数据格式转换过程中, CPU 除了执行每个数据的物理量转换公式 (简单的线性公式或者复杂的多项式) 外, 主要就是内存和数据转字符串操作^[5]。具体说, 由 .da 格式的数据转换为 .txt 格式的数据, 进行的主要操作为将大量 float 类型数据转换为一个字符串型长数据, 其间要进行大量字符串合并操作。通过深入分析数据格式转换过程, 得出主要影响转换效率的 3 个因素:

- 1) 转换算法所用的循环方式;
- 2) 转换时所用的字符串操作函数;
- 3) 转换的数据量。

以下就 3 个因素逐一进行影响域分析。

3.1 转换算法所用的循环方式

现有转换算法用到了通道数和总数据量两层循环来遍历所有待转换点, 其中通道数是外循环, 总数据量是内循环, 通过将内外循环顺序颠倒, 进行实测, 发现对转换时间基本没有影响, 故其为影响的次要因素。

3.2 转换时所用的字符串操作函数

在 Windows 平台下基于 C++ 的非嵌入式软件开发一般采用 Visual C++ 系列或 C++ Builder 平台。现有转换算法用到了 MFC 类库的 CString 类进行数据格式转换和字符串合并, 具体用到的操作函数为数据格式转换函数 Format () 和字符串合并函数 operate += ()。通过在 3.1 的两层循环中插入计时函数来实时监测转换时间, 发现: 随着待合并数据量的不断增加, 数据格式转换函数 Format () 在进行一个新字符串转换时所用时间基本不变, 而字符串合并函数 operate += () 在进行一个新字符串合并时所需时间越来越长。即, 算法

起始时合并一个字符串和过程中合并一个字符串所用时间并不相等, 而是呈现出时间不断增长趋势。故 operate += () 应为影响的主要因素。C++ Builder 平台下 VCL 类库的 TString 类也有类似问题。

3.3 转换的数据量大小

通过在 3.1 的两层循环中插入计时函数实时监测转换时间, 发现: 随着待合并数据量不断增加, 合并一个新字符串所需时间越来越长。即, 算法起始时合并一个字符串和过程中合并一个字符串所用时间并不相等, 而是呈现出了不断增长的趋势。通过对比发现: (下述数据基于某种类伺服测试软件的测试工况取得) 转换 52 个通道每通道 4 000 点所用时间为 44 秒, 而转换 52 个通道每通道 16 000 点所用时间为 6 分钟, 转换 52 个通道每通道 100 000 点所用的时间为 5 小时。随着转换数据量增加, 转换时间并没有呈现线性增长, 而是呈现指数级更快增长。图 2 为转换时间与数据量关系对照图。

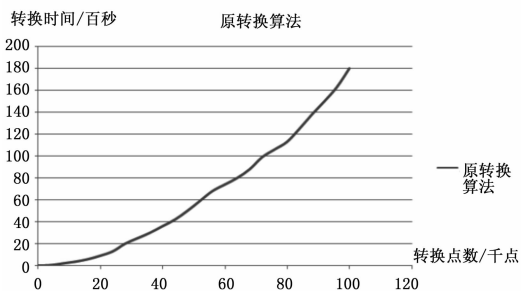


图 2 转换时间与数据量关系对照

可以看出, 待转换数据量也为影响的主要因素。但是由于在实际应用中待转换数据量为必须完成的任务, 不可更改, 该因素予以去除。只能从前面两个因素来进行考虑。即应该设计出和数据量大小成线性关系的新转换算法。

4 新转换算法设计

4.1 算法分析与改进

通过分析上述影响因素发现, 影响转换效率的主要因素为 CString 类的 operate += () 函数, 进一步分析该函数发现, 由于该函数为 CString 类的成员函数, 该函数在合并一个新的字符串时所进行的操作异常复杂, 主要包括以下几步:

- 1) 判断新字符串是否为空, 以及新串长度;
- 2) 如不为空, 则根据新串长度来计算自身空间是否足够容纳新串, 如果可以则直接追加到自身串尾;
- 3) 如果自身空间不足, 则需要根据自身算法新建一个新内存空间, 该空间大小除了能够容纳原有字符串和新字符串长度之和外, 还应该有一定余量, 以便于再次合并新串时, 不需要反复新建内存空间, 然后将原有字符串拷贝过去, 再将新字符串追加到串尾。

分析以上步骤不难发现: 随着待合并数据不断追加, 需要不断开辟新的内存空间, 进行原有字符串和新字符串的拷贝操作; 而且这些操作随着数据的增加越来越频繁, 最终导致转换时间非线性指数级增长。

基于主要影响因素分析, 设计了新字符串合并方法, 保证转换时间的线性增长。主要改进为: 抛开 CString 类, 通过直接内存操作方式, 一次性开辟足够大内存空间, 进行拷贝、转换和存储字符串数据。

具体实现为：

1) 预先开辟一块足够大内存空间 P, 该空间大于全部待转换字符串长度之和, 足够容纳所有待转换数据量。由于通道数和每通道个数已知, 只需再确定每个待转换字符串的最大长度;

2) 设置一个动态指针 p, 该指针始终指向 P 中有效字符串尾部, 初始时指向 P 起始位置。当需要合并一个新字符串时, 只需在 p 的位置进行新字符串合并操作即可。合并完成后, p 后移新字符串长度的位置。以此循环执行, 即可完成所有新字符串合并操作;

3) 所有数据合并完成后, 重新计算 P 中有效字符串总长度, 即为转换完成的字符串总长度, 将该部分内容存储为 .txt 格式的文本文件。

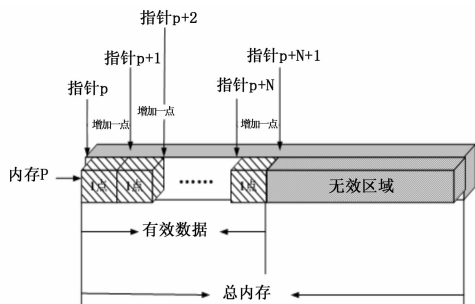


图 3 新算法数据转换过程

该方法通过预先分配足够大内存空间 P, 每次操作都针对同一块内存区域, 避免了反复开辟、删除、拷贝内存数据带来的时间开销, 保证了转换时间线性增长。同时, 通过直接操作内存, 在最短时间内完成一个新字符串的拷贝工作。

4.2 改进效果验证

通过实际操作验证, 采用新的转换算法, 转换 52 个通道每通道 4 000 点所用时间为 0.1 秒, 而转换 52 个通道每通道 16 000 点所用时间也为 0.4 秒, 转换 52 个通道每通道 100 000 点所用的时间为 2.5 秒。转换时间大大减少, 用时甚至可以忽略不计, 再也不是快速获取。txt 文本数据的制约因素。随着转换数据量的增加, 转换时间表现出了线性增长的趋势, 转换进度变为线性可控。下图 4 为优化后的转换时间与数据量关系对照图, 转换效率提高 260 000 倍, 特别是在转换超大数据量方面, 极大缩短了转换时间。优化前后数据转换时间对比如下图 5 所示。

4.3 改进措施固化

除进行源代码级的算法改进, 同时设计了人机交互界面^[6]工具协助使用者进行格式转换操作。该界面能够帮助使用者选择待转换数据, 并实时显示转换进度, 避免使用命令行转换形式带来的不便和操作失误; 同时增加操作的便利性, 转换状态实时显示以方便用户查看, 不再是未知的等待过程。

(上接第 243 页)

参考文献:

[1] 搜搜百科: 微信公众平台 [EB/OL]. <http://baike.soso.com/v58378222.htm>.

[2] 百度百科: 微信公众平台 [EB/OL]. <http://baike.baidu.com/view/9212662.htm>.

[3] 观察: 微信公众平台开发的两种方向 [EB/OL]. <http://mobile.51cto.com/comment-388760.htm>.

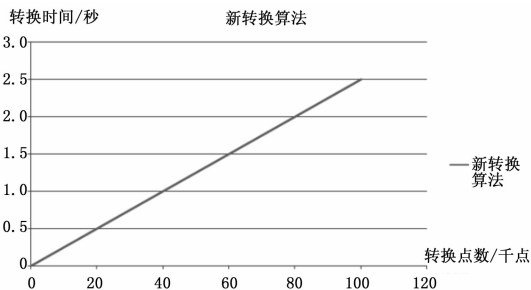


图 4 优化后的转换时间与数据量关系对照

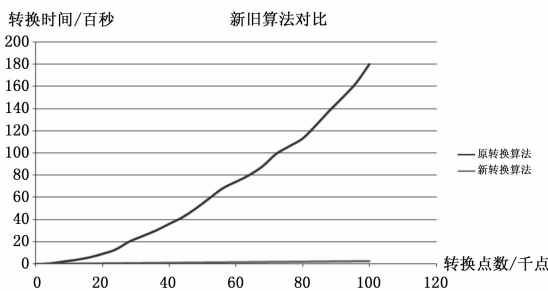


图 5 优化前后数据转换用时对比

5 结论

在伺服测试数据格式转换中某种类伺服测试软件遇到了大数据量时转换效率低下的问题, 通过对数据格式转换过程分析, 设计了新的转换算法, 优化了算法内存使用方式, 提高了转换效率。该成果已成功应用于某种类伺服系统单元测试仪、总体人员客户端及试验室数据分析客户端; 在其它种类伺服测试中, 如 CZ-5、CZ-7 运载种类也得到广泛应用。通过开发界面化的转换工具, 将数据转换工作进行了前移, 直接部署到数据客户端, 为使用者提供便捷、高效的数据转换服务, 提高了生产效率。随着应用的深入, 提出了大批量数据批处理需求, 这是以后需要完善的地方。

参考文献:

[1] 张志成, 张合新. 基于 PXI 总线的导弹伺服机构测试系统 [J]. 计算机测量与控制, 2006, 14 (12): 1657-1659.

[2] 谢伟等. 伺服机构测试软件系统的发展及新技术的应用 [J]. 流体传动与控制, 2008 (5): 61-65.

[3] 苏利娜等. 电子测量仪器自动记录数据格式转换程序设计 [J]. 测绘科学, 2011, 36 (4): 186-190.

[4] 张可迪等. 一种嵌入式操作系统运行时验证方法 [J]. 计算机工程与科学, 2014, 36 (5): 900-905.

[5] 权建军, 等. 嵌入式系统中数据与字符串的转换及应用 [J]. 单片机与嵌入式系统应用, 2012 (4): 71-72.

[6] 魏瑞华等. 紫外火焰探测器测试系统人机交互界面的设计 [J]. 西安工程大学学报, 2013, 27 (1): 65-68.

[4] 易伟. 微信公众平台搭建与开发揭秘 [M]. 北京: 机械工业出版社, 2013.

[5] 杨尚琴. 基于微信平台的公共图书馆个性化参考咨询服务研究 [J]. 图书馆学刊, 2015, 10: 61-63.

[6] 微信公众平台 API 文档 [EB/OL]. <http://mp.weixin.qq.com/wiki/home/index.html>.

[7] 微信公众平台官方 [EB/OL]. <http://mp.weixin.qq.com>.