第11卷 第33期 2011年11月
1671—1815(2011)33-8357-04

科 学 技 术 与 工 程
Science Technology and Engineering

Vol. 11 No. 33 Nov. 2011

计算机技术

一种从数据库快速加载 mat 文件的方法

樊峰峰 张延园 林 奕 王慧文
(西北工业大学计算机学院,西安 710129)

摘 要 Matlab 和数据库技术广泛应用于大型工程应用系统中。存储在数据库中的 mat 文件数据须经过本地文件系统,由 Matlab 加载到工作空间中。每次文件数据加载过程附带两次文件 IO,降低了应用软件性能。提出基于内存的 mat 文件数据快速加载方法,将数据库中的文件数据下载到 Matlab 内存缓冲区中,在缓冲区中完成文件数据扫描和加载,消除了本地文件系统的介入,显著改善了应用软件系统性能。

关键词 Matlab 数据库 mat 文件 快速加载 MATIO 基于内存
中图法分类号 TP319; **文献标志码** A

MATLAB 是矩阵实验室(Matrix Laboratory)的简称,它提供了强大的科学计算能力及丰富的工具箱和函数库,被认为是工程应用系统中理想的计算引擎。面对工程应用系统中大量的数据文件及数据之间的复杂关系,简单的基于文件系统的数据组织与存储方式已经不能满足应用需求。数据库技术有效解决了计算机信息处理过程中大量数据的组织、存储、共享和冗余消除等问题,因此,在工程应用系统中,广泛采用数据库技术存储数据文件。

基于 Matlab 和数据库的工程应用系统中,存放在数据库中的 mat 数据文件须下载到本地文件系统中才能由 Matlab 加载。文件数据的每次加载和保存,都会附带两次文件 IO 操作,给应用系统的性能造成较大影响。那么,在此类应用中,如何消除文件系统造成的性能损失呢? 本文提出了一种 mat 文件数据快速加载方法。该方法从数据库中检索 mat 文件数据并下载到 Matlab 内存缓冲区中,在缓冲区完成 mat 文件数据分析和加载,彻底消除了本地文件系统介入 mat 文件数据加载过程,显著改善了工程应用系统性能。

1 相关研究工作

Matlab 仅仅提供了 mat 文件格式规范[1],而没有提供 mat 文件 IO 库的源代码。因此,无法明晰 mat 文件 IO 库的实现细节。MATIO[2] 是用于读写 mat 文件的标准 C 语言函数库,提供了解析 mat 文件格式的 API,且兼容最新版的 mat 文件格式,是 mat 文件 IO 库的开源实现。它与标准的 mat 文件 IO 库一样,采用基于文件系统的数据存储方式。而基于 Matlab 和数据库的工程应用系统中,采用现有技术和方法,都无法在数据加载过程中消除文件系统的介入。本文从文件数据传输过程入手,提出了基于内存的 mat 文件数据快速加载方法。该方法使用内存缓冲区替代文件系统作为 mat 文件数据临时存储介质,有效消除了文件系统的介入,显著提升了应用系统的数据加载速度。

2 基于文件系统的数据加载方法

图 1 显示了工程应用系统的软件架构与软件组件之间的数据通讯接口。数据库作为存储子系统,用于存储系统的数据文件;Matlab 作为系统的计算

2011 年 9 月 5 日收到,9 月 15 日修改
第一作者简介:樊峰峰(1986—),男,山西吕梁临县人,硕士研究生,研究方向:计算机软件与理论。E-mail:fanfengfeng043763@126.com。

引擎,用于完成数据计算与分析;文件系统则作为文件数据交换介质,图 1 中的着色部分显示了在数据文件加载过程中应用系统的数据文件传输路径。Matlab 的数据库工具箱采用 ODBC/JDBC 技术与数据库交换数据对象;而 Matlab 的 load 命令和 save 命令,用于在文件系统中加载和保存数据。

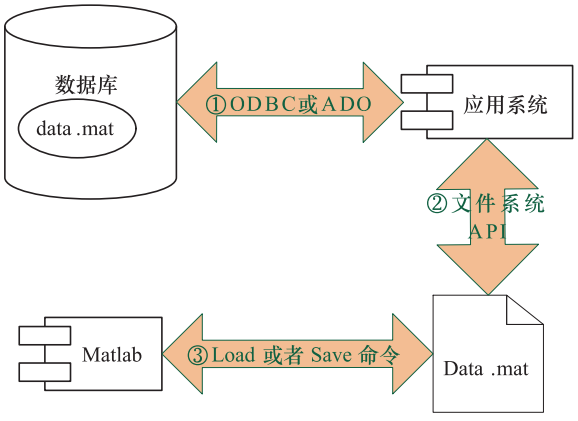

图 1 基于文件系统的数据加载方法软件结构图

以 data. mat 文件为例,分析工程应用系统中基于文件系统的数据加载方法,步骤如下:

1)从数据库中检索 data. mat 文件数据并下载到本地文件系统中;

2)调用 Matlab 的 load 命令将 data. mat 加载到 Matlab 计算引擎的工作空间中;

从图 1 不难看出, Matlab 与数据库子系统之间的每次数据交换都会附带产生两次磁盘 IO。而磁盘是计算机中速度较慢的部件,如果应用程序子系统之间中有频繁的数据交换,文件 IO 就会成为整个应用系统的性能瓶颈。

3 基于内存的数据文件加载方法

在计算机系统中,内存数据的存取速度远远快于磁盘系统,本文提出的基于内存的文件数据加载方法使用内存代替磁盘作为 Matlab 和数据库子系统之间的文件数据交换介质,在内存缓冲区中完成文件数据的分析和加载。文件数据传输过程和子系统通讯接口如图 2 所示。

同样以 data. mat 文件加载为例,分析工程应用系统中基于内存的数据文件加载方法,步骤如下:

1)从数据库中检索 data. mat 文件数据并下载到 Matlab 内存缓冲区中(图中 tmp 为待加载的数据文件字节流)。

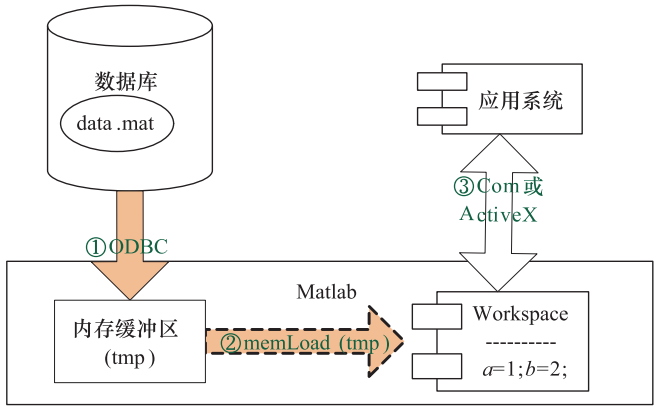

图 2 基于内存的数据加载方法软件结构图

2)扫描内存缓冲区中,解析出文件数据并加载到 Matlab 工作空间中;

图 2 中的 memLoad 实现了在内存缓冲区中文件数据扫描和加载的功能。

4 技术实现方案

基于内存的数据文件加载方法,在技术上分为三个部分:数据文件检索部分,数据文件解析部分和解析结果加载部分。

4.1 数据文件检索部分

Matlab 提供的数据库扩展工具箱采用 ODBC/JDBC 技术实现了与数据库的互联与数据交换。Matlab 与数据库之间的数据存取见文献[4]。

4.2 数据文件解析部分

Matlab 的 M 编程语言为解释性语言,对于非矩阵运算,执行效率较低下;而 Matlab 的 MEX 编译技术[4]能够将 *.C 源代码编译为可供 Matlab 调用的 *.mexw32 动态连接程序。因此,采用 C 语言编写用于分析缓冲区中 mat 文件数据的功能代码,然后经过 MEX 编译之后,生成由 Matlab 调用的动态连接程序。

4.3　结果加载部分

MEX 接口技术也提供了与 Matlab 交换数据对象的机制。缓冲区中 mat 格式字节流通过输入参数传递给解析函数。解析结果则通过 MEX 接口中 API(如 mexPutVariable 函数)加载到 Matlab 工作空间中。

内存缓冲区 mat 文件扫描和数据加载的函数调用层次结构源码部分如表 1 所示:

表 1　基于内存的数据加载方法部分源码

```
// memLoad. c#include "mex. h"
/* 从 pcurrent 到 pend 之间的缓冲区中解析数据变量 */
int
parseVariable(char * vname, const mxArray * pm,char * pcurrent, char * pend)
{
/* 按照 mat 数据格式扫描字节流,解析变量名和数据,
 * mat 格式解析代码见 MATIO 项目,在此省略……
 * 变量名赋给 vname,解析出的数据被动态构造成阵列
 * 赋给 mxArray,指针赋给 pm
 */
return 0;
}
void
mexFunction(int nlhs,mxArray * plhs[],int nrhs,const mxArray * prhs[])
{
/* 防御性的参数检查与异常处理代码省略…… */
const mxArray * cell_element_ptr;
//找到第一个阵列元素
cell_element_ptr = mxGetCell(prhs[0], 0);
char * pdata;
//pdata 指向二进制字节流首地址
pdata = (char *)mxGetData(cell_element_ptr);
mwSize length = 0;
//length 表示二进制字节流长度
length = mxGetNumberOfElements(cell_element_ptr);
//设置二进制字节流结尾指针
char * pend = pdata + length;
//设置二进制字节流当前指针
char * pcurrent = pdata;
char vname[1024];
const mxArray * pm;
while(pcurrent < pend)
{
//从 pcurrent 到 pend 之间字节流中解析一个变量
//vname 存放变量名,pm 存放 mxArray 数据矩阵指针
parseVariable(vname,pm,pcurrent,pend);
//将每个变量解析到 Matlab 工作空间
mexPutVariable("base", vname, pm);
//销毁 pm 所指向的动态创建的 mxArray 数据结构
mxDestroyArray(pm);
}
}
```

其中 mat 格式数据解析部分借鉴了 MATIO 项目中相关部分的功能代码。

5　实验数据论证

在实验中,分别采用基于文件系统和基于内存的数据文件加载方法对 1 M—50 M 的数据文件加载时间进行测试。其中基于文件系统的 mat 数据文件加载方案(即方案一)的数据曲线用蓝色标记;基于内存的 mat 数据文件加载方案(即方案二)的数据曲线用红色标记。实验数据曲线图如图 3 所示。

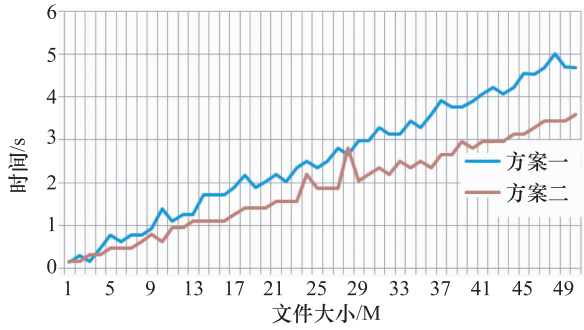

图 3　文件大小与数据加载时间关系图

实验数据论证:

由实验数据可以得出,采用基于内存的数据加载方式,对应用系统的数据文件加载性能有较显著的提升。

6　两种数据文件加载方法对比

基于文件系统的数据文件加载方法,采用了 ODBC/JDBC 技术、文件系统 API 和 Matlab 的文件 IO 库等常规技术的组合方案,是工程应用系统开发中采取的通用开发方法。

而基于内存的数据文件加载方法,则需要自行实现数据文件解析的相关功能,结合 Matlab 中 ODBC 技术和 MEX 接口等高级技术,构建完整的应用系统实现方案,适合性能关键的特殊工程应用系统。

基于文件系统的数据文件加载方法,具有开发难度较小,参考资料丰富,应用案例多等优点。然而对于性能关键的特殊应用系统,采用基于内存的数据文件加载方法等非常规技术手段,则能够满足特殊的性能要求。

7 展望与总结

本文介绍了基于内存的数据文件加载方法的基本原理和实现手段,该方法可用有效解决 Matlab 与数据库应用系统中数据文件加载性能低下的问题。相关实验结果表明:基于内存的数据文件加载方法其性能有较大提升。

目前,基于内存的数据文件加载方法完成了 mat 数据文件从数据库到 Matlab 工作空间的加载过程,其性能优势已经在实验中得到验证。笔者将继续研究快速保存 mat 数据文件到数据库中的方法,即把 Matlab 工作空间中的数据对象集在内存中以 mat 格式规范完成序列化,并将序列化之后的数据文件字节流保存到数据库中。

参 考 文 献

1 Mathworks Corporation. MAT-File Format, (2011-04) http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/matfile_format.pdf,2011-04

2 chulbe2lsu. matio. http://sourceforge.net/projects/matio/ ,2006

3 吴 迪, 刘 军, 徐 朋, 等. 基于 MATLAB 及数据库技术的实验数据检验及存取研究. 大连: 大连大学, 2010

4 Mathworks Corporation. MEX-files Guide, (2003-08-04) http://www.ee.columbia.edu/~marios/matlab/MEX-files% 20Guide% 201605.pdf.

A Method of Loading Mat Files Rapidly from Database

FAN Feng-feng, ZHANG Yan-yuan, LIN Yi, WANG Hui-wen

(Dept of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710129, P. R. China)

[**Abstract**] Matlab and database are widely used in engineering applications. The mat files stored in database must be transferred onto file-system from before being loaded by the Matlab. Two file operations, from database to file system and from file system to Matlab, get involved in the process of mat file loading, and cause the performance lost of applications. A memory-base loading method is proposed. By fetching file data from database into buffer of the Matlab, in which file data is analyzed and loaded into the Matlab's workspace, and getting rid of the unnecessary file IO operations, the performance of applications gets improved.

[**Key words**] Matlab database mat file fast loading MATIO memory-based