

FlowVR 的 Scilab ToolBox 的设计与实现

唐 剑,赵光恒,刘 薇,陈福恩

TANG Jian,ZHAO Guang-heng,LIU Wei,CHEN Fu-en

中国科学院 光电研究院,北京 100080

The Academy of Opto-Electronics,Chinese Academy of Sciences,Beijing 100080,China

TANG Jian,ZHAO Guang-heng,LIU Wei,et al.Design and implementation of Scilab ToolBox for FlowVR.Computer Engineering and Applications,2008,44(24):95-97.

Abstract: FlowVR is a data-flow oriented distributed framework for VR systems^[1],which has many advantages for implementing large scale distributed scientific analysis systems except for that it has no mathematical modeling functionality.Scilab has numerous features for modeling and simulation,which enlightens us to build high performance data analysis system with the collaboration of both FlowVR and Scilab.This article presents the design and implementation of a Scilab toolbox that functions as an interface between FlowVR and Scilab because there is no interface between them.We realize data exchange between modules that run under FlowVR framework and Scilab by means of a well developed Scilab programming interface.With the help of scientific computing functionalities of Scilab,it becomes easier to develop data analysis system with FlowVR.

Key words: interface;ToolBox;Scilab;FlowVR

摘 要:FlowVR 是为 VR 提供的支持数据流的分布式框架^[1],对于构建功能强大的分布式科学数据分析系统也有很大的优势,但是 FlowVR 本身没有数学建模功能。Scilab 具有强大的数学建模和仿真功能。结合 FlowVR 和 Scilab 的优点,可以构建高性能的科学数据分析系统,以应用于各种领域。文章介绍了 FlowVR 的 Scilab Toolbox 的设计与实现。该系统通过开发 Scilab 的函数编程接口,实现了 FlowVR 框架下各个模块与 Scilab 的数据交互,利用后者的科学计算功能,能够为 FlowVR 开发数据分析系统提供强大的建模工具。

关键词:接口;工具箱;Scilab;FlowVR

DOI:10.3778/j.issn.1002-8331.2008.24.027 **文章编号:**1002-8331(2008)24-0095-03 **文献标识码:**A **中图分类号:**TP311

1 引言

随着航天事业的高速发展,空间系统仿真平台变得越来越重要,要完成各种空间任务仿真,高性能的数据分析系统就必不可少。

Scilab 是由法国国立信息与自动化研究院 INRIA 开发的科学计算软件,提供类似于 Matlab 的强大科学计算功能。其特点是为科学计算提供了一个开放式的编程环境,便于定制,提供多个操作系统版本,语法与 Matlab 的相似,初学者学习周期短。

FlowVR 是一种基于并行分布式构架^[2],用于虚拟现实和科学可视化系统的开源软件框架。其特点是面向数据流的分布式的模块化开发,不同模块通过 FlowVR 网络交换数据。在保证分布式构架下的高性能的同时,有效地降低了各种分布式仿真系统的开发难度。

FlowVR 的标准模块是基于 C/C++ 的,其开发难度和复杂度相对较高。对于进行大量数值计算的模块尤为如此,而 Scilab 在此领域具有非常明显的优点。开发 Scilab 与 FlowVR 的工具

箱 Toolbox,对于简化代码量,提升系统扩展性、适应性等方面都有突出的意义。

2 总体框架设计

本文开发的 Scilab 工具箱 Toolbox 称为 ScilabToolBox,分为三个部分:(1)FlowVR 标准模块 FModule1,此模块提供一系列接口函数供使用者调用;(2)Scilab 接口工具箱 ScilabToolBox,其作用是 Scilab 和 FlowVR 的桥梁,构建 Scilab 类型嵌入函数,调用 FModule1 模块的成员函数、数据等;(3)Scilab 脚本是用户编写自定义的脚本,调用 ScilabToolBox 提供的 Scilab 类型的函数。

3 功能模块设计

如图 1 所示,FlowVR 标准模块 FModule1 由开发者用 C++ 编写,遵循 FlowVR 模块开发规范;由于 Scilab 接口编程只能用 C 实现,ScilabToolBox 的 c2cpp 组件由 C++ 编写,其作用是

作者简介:唐剑(1980-),男,硕士研究生在读,主要研究方向为计算机仿真技术;赵光恒(1965-),男,研究员,博士生导师,主要研究方向为飞行器动力学、控制与仿真,计算机控制与应用,系统仿真测试技术等;刘薇(1977-),女,工学博士,主要从事计算机控制与应用,飞行器有效载荷运控模式研究及仿真研究等;陈福恩(1962-),男,工学博士,研究员,硕士生导师,主要研究方向为飞行器设计领域,计算机应用等。

收稿日期:2007-10-23 **修回日期:**2008-02-22

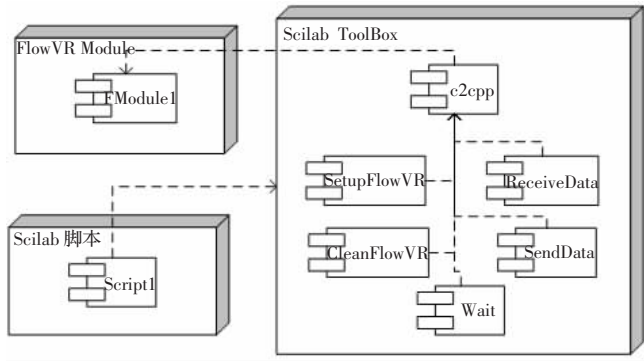


图1 ScilabToolBox 总体结构图

提供 C 调用 C++类成员函数的一种方法。剩下的 5 个组件分别是 SetupFlowVR, ReceiveData, SendData, CleanFlowVR, Wait, 都由开发者遵循 Scilab 接口开发规范实现。Scilab 脚本由用户实现, 详细说明如表 1 所示。

4 接口设计

4.1 语言接口—Scilab 调用 C

根据 Scilab 的规定, 其接口程序只能用 C 实现, 并且需要遵循相对严格的格式和调用约定。开发的基本步骤如下:

- (1)按照 Scilab 接口规范编写 c 源代码, 编译到.o 目标文件;
- (2)编辑一个.sce 配置文件, 用于描述该模块的可执行文件位置, 在 Scilab 脚本中调用函数名称等信息, 通过 Scilab 提供的 ilib_build 函数编译上述信息到一个.so 文件;
- (3)Scilab 调用此函数之前, 调用 addinter 函数将上述.so 文件加载到 Scilab 环境中;
- (4)编写 Scilab 脚本, 调用上述函数。以 ScilabToolBox 中的 ReceiveData 组件为例, 在 Scilab 脚本中代码为:

```
[RevData, state] = ReceiveData();
```

上述函数接口调用完成之后将得到两个返回值, 一个为接收到的数据, 是 Scilab 类型的矩阵; 另一个是调用是否成功状态值。

Scilab 运行时内部数据, 比如矩阵、函数参数等等, 保存在类似于堆栈类型的指定内存区域^[9]。接口代码需要访问上述数据, 是通过调用 Scilab 提供的函数实现的。比如要得到输入参数, 首先通过调用 GetRhsVar 得到目标数据在上述堆栈中绝对位置, 将其保存在一个静态 int 变量中, 比如定义为 m。之后通过调用 Scilab 提供的函数 stk(m)得到上述区域的指针, 然后即可按照正常的 C 程序对其进行读写操作。需要注意的是, 不同的数据类型需要不同的 stk 函数, 比如 int 类型调用 istk(), char 型调用 sstk(), double 类型调用 stk()。

对于上述步骤的第一步, 需要建立一个单独的 c 文件, 包含一个提供给 Scilab 调用的 C 函数, 其输出为 int 类型, 输入为 char* 类型字符串指针, 函数名称自定, 定义如下:

```
int int_ReceiveData(char *fname)
```

int_ReceiveData() 首先调用 CheckRhs 和 CheckLhs 函数检查 Scilab 脚本调用参数个数合法性。然后通过 GetRhsVar 等函数获取输入的参数堆栈位置, 对返回给脚本的数据调用 CreateVar 在堆栈分配区域, 之后将上述区域指针通过 stk() 得到并送入 c2cpp 对应的 ReceiveData 函数, 进而调用 FModule1 的 IReceiveData 函数得到数据, 数据保存到上述指针指向的内存区域后返回; 最后调用 LhsVar 指定每一个返回数据在 Scilab 内部数据堆栈中的绝对位置。

上述步骤的简化版本流程图如图 2:

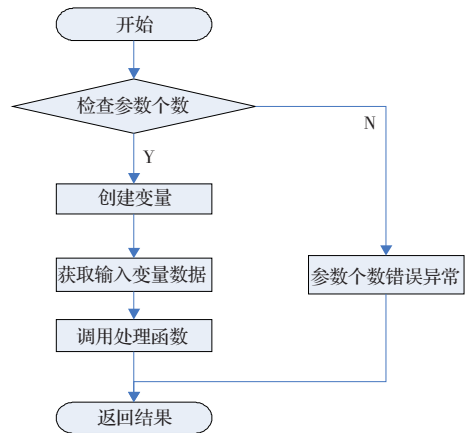


图2 接口实现简化流程图

4.2 语言接口—C 调用 C++成员函数

由于 Scilab 接口只能用 C 实现, 而 FlowVR 模块通过 C++类实现, 其接口以成员函数的形式存在。因此需要 C 调用 C++函数的一种方法, 即 c2cpp 组件的功能, 内部通过声明 extern 代码块实现^[9]。

以上述 ReceiveData 函数为例, 简化版源代码如下:

```
#include "FModule.h"
FModule fm;
extern "C" {
    int ReceiveData(double *pdData, int *pnState){
        fm.IReceiveData(pdData, pnState);
    }
}
```

4.3 Scilab 脚本实例

一个典型的 FlowVR 模块需要首先初始化模块参数, 然后

表 1 功能模块详细说明表

模块/组件	语言	实现者	功能说明
FlowVR 标准模块	FModule1	C++ 开发者	实现 FlowVR 标准模块基本功能, 提供若干接口供 ScilabToolBox 调用
	c2cpp	C++ 开发者	提供 C 调用 C++成员函数功能
	ReceiveData	C 开发者	通过 c2cpp 调用 FModule1 的 IReceiveData 接口函数, 实现数据接收功能
ScilabToolBox	SendData	C 开发者	通过 c2cpp 调用 FModule1 的 ISendData 接口函数, 实现数据接收功能
	SetupFlowVR	C 开发者	通过 c2cpp 调用 FModule1 的 ISetupFlowVR 函数, 实现 FlowVR 模块初始化
	CleanFlowVR	C 开发者	通过 c2cpp 调用 FModule1 的 ICleanFlowVR 函数, 实现 FlowVR 模块退出时资源释放
	Wait	C 开发者	通过 c2cpp 调用 FModule1 的 IWait 接口函数, 实现 FlowVR 模块与管理者同步
Scilab 脚本	Script1	Scilab 用户	通过调用 ScilabToolBox 用 C 语言实现的 5 个模块, 实现不同应用

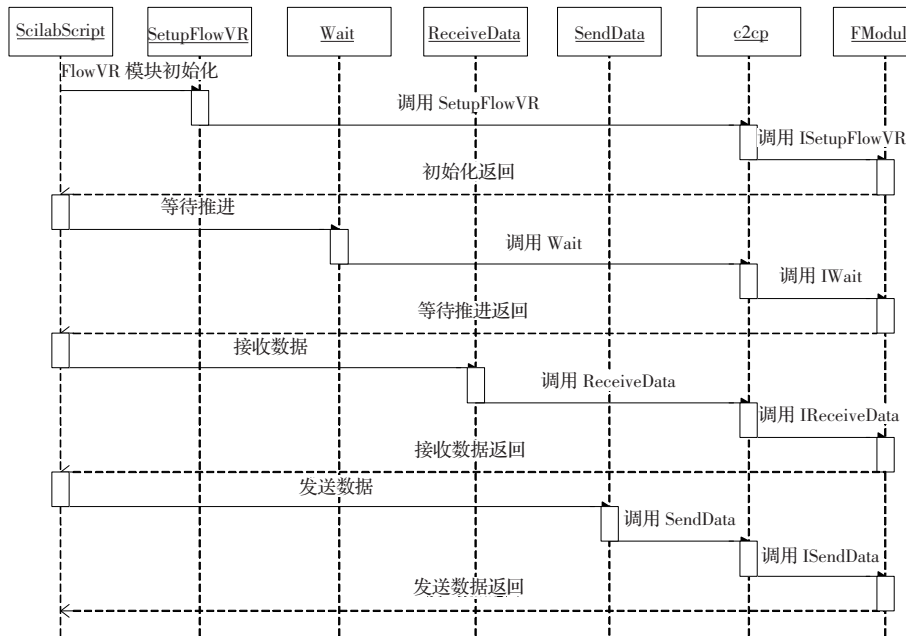


图 3 单次循环顺序图

调用 wait()函数进入仿真循环,退出时还需要释放对应资源等等。Scilab 脚本开始运行后,通过调用 ScilabToolBox 对应的接口函数执行上述步骤。下面给出了一个典型脚本简化代码:

```

// test.sce
state = SetupFlowVR()
if state <> -1 then,
    wait_state = Wait();
    while wait_state > 0
        [Data, state] = ReceiveData()
        // 数据位于 Data 中,在此计算
        if state == 0
            [row, col] = size(Data);
            SendData(Data, col);
        end;
        wait_state = Wait();
    end;
end;
end;

```

5 运行机制

5.1 嵌入 FlowVR 环境

对于正常的 FlowVR 项目,每个模块通过 C++实现,编译之后会在/bin/目录下面生成对应的可执行文件,运行时 FlowVR 管理者通过一定机制调用对应文件。为了实现前文所述与 ScilabToolBox 的对接, FModule1 模块更改为一个 shell 脚本文件,覆盖上述/bin/目录中的同名可执行文件,则运行时 FlowVR 将调用 Shell 脚本实现的 FModule1。

上述 Shell 脚本的关键是通过调用 Scilab 启动参数,如 scilab -nw -f test.sce,意为指定 Scilab 以无窗口模式启动并运行 test.sce 脚本,运行的控制权即交给了 test.sce 脚本。下面

给出了 Shell 文件的典型内容:

```

#!/bin/sh
/usr/local/scilab/scilab-4.0/bin/scilab -nw -f
/usr/local/scilab/scilab-4.0/examples/test/test.sce

```

5.2 模块相互调用关系

图 3 通过一个单次循环的顺序图进一步说明 ScilabToolBox 与其他组件之间在运行时的相互调用关系。

6 结论

本文通过开发 ScilabToolBox 工具箱,有效地结合 FlowVR 面向数据流的分布式构架,实现了某分布式数据分析系统的数据计算与 Scilab 脚本交互。本设计具有较强灵活性、易用性,同时在 FlowVR 的分布式构架下,能够确保较好的性能可扩展性,有效地提高了分析系统开发效率。

参考文献:

- [1] Arcila T, Allard J, Menier C, et al. FlowVR: a framework for distributed virtual reality applications[J]. Journees de l'AFRV, 2006, Rocquencourt: 1-4.
- [2] Allard J, Gouranton V, Lecointre L, et al. FlowVR: a middleware for large scale virtual reality[C]//Applications, Euro-Par'04, Pisa, Italia, August 2004.
- [3] Paulo S, Motta Pires, Rogers D A. Free/open software: an alternative for engineering students[C]//32nd ASEE/IEEE Frontiers in Education Conference, Boston, MA, November 6-9, 2002.
- [4] 胡包钢, 赵星, 康孟珍. 科学计算自由软件——SCILAB 教程[M]. 北京: 清华大学出版社, 2003.
- [5] C++Primer 中文版[M]. 潘爱民, 张丽, 译. 3 版. 北京: 中国电力出版社, 2002.