实战 Matlab 之 并行程序设计

刘 维 编著

登录http://www.iLoveMatlab.cn/注册
用户名，验证密码后即可与作者在线交流

卡号：2011001679984
密码：

北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

实战 Matlab 之

并行程序设计

刘 维 编著

北京航空航天大学出版社

内 容 简 介

本书对基于 Matlab 的并行程序设计的原理进行了深入的剖析，并结合各章给出的大量实例对基于 Matlab 的并行计算程序设计方法和技巧给出了详细的说明。通过阅读和学习本书的内容，读者可以掌握基于多种平台（多核、多处理器、集群和 GPU 等），利用多项技术（Matlab 并行计算工具箱、多线程 MEX 文件、Open MP 和 GPU 等），学习理解 Matlab 并行程序设计的原理、方法和技巧。全书共分 10 章：第 1 章为 Matlab 开发环境和程序设计基础；第 2 章为利用 parfor 对 for 循环进行并行；第 3 章为 SPMD 并行结构；第 4 章为其他 Matlab 并行结构；第 5 章为 Matlab 并行计算数据类型；第 6 章为 Matlab 通用并行程序设计；第 7 章为 MDCE 配置；第 8 章为创建多线程 MEX 文件；第 9 章为在 Matlab 中应用 OpenMP 进行并行计算；第 10 章为利用 GPU 并行执行 Matlab 程序。书中附录共包括三个部分，即 MEX 文件基础知识、用户配置项和 Matlab 并行计算常用概念说明。

书中所有的源代码均可在出版社网站的下载中心和 Matlab 中文论坛（www. iLoveMatlab. cn）中下载。除特别说明之外，其开发和编译环境均为 Matlab 2010 与 Visual C++ 2010。本书的阅读对象包括大中专院校学生以及利用 Matlab 开发并行程序的人员。

图书在版编目(CIP)数据

实战 Matlab 之并行程序设计 / 刘维编著. -- 北京 : 北京航空航天大学出版社，2012.3

ISBN 978-7-5124-0597-4

Ⅰ. ①实… Ⅱ. ①刘… Ⅲ. ①计算机辅助计算—软件包，Matlab②并行程序—程序设计 Ⅳ. ①TP391.75②TP311.11

中国版本图书馆 CIP 数据核字(2011)第 193435 号

实战 Matlab 之并行程序设计

刘 维 编著

责任编辑 胡 敏

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) http://www.buaapress.com.cn

发行部电话：(010)82317024 传真：(010)82328026

读者信箱：bhpress@263.net 邮购电话：(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

*

开本：787×1 092 1/16 印张：18.75 字数：480 千字

2012 年 3 月第 1 版 2012 年 3 月第 1 次印刷 印数：5 000 册

ISBN 978-7-5124-0597-4 定价：35.00 元

前　言

随着多核和集群技术的快速发展，并行程序设计成为提高数值计算效率的主流技术之一。目前，常用的小型并行计算平台可大致分为两类：一类是由多核和多处理器构建的单计算机平台；另一类是由多个计算机构成的集群(Cluster)系统。单个计算机上的多核或多 CPU 之间可以通过共享内存进行数据交互；多个计算机构成的集群通过网络进行数据通信。常用的并行计算技术包括多线程技术、基于共享内存的 OpenMP 技术、基于集群系统的 MPI 技术等。但无论是基于多线程的并行计算技术，还是基于 OpenMP 的并行计算技术，以及基于 MPI 的并行计算技术，都需要用户处理大量的与并行算法无关的技术细节。此外，这些并行计算技术本身并不提供高效的算法库，与数值计算的关联是松散的。

Matlab 已经成为数值计算领域的主流工具，它提供了大量高效的数值计算模块和丰富的数据显示模式，便于用户进行快速算法研究和科学建模仿真。因此，如果 Matlab 支持多核、多处理器和集群并行计算的话，那么对于科学研究人员和工程师来说，无疑是一个新的福音。在 Matlab 2009 之后，Matlab 推出了并行计算工具箱(Parallel Computing Toolbox，PCT)和并行计算服务(Distributed Computing Server，DCS)，通过 PCT 和 DCS 用户可以实现基于多核平台、多处理器平台和集群平台的多种并行计算任务。利用 PCT 和 DCS，用户无需关心多核、多处理器以及集群之间的底层数据通信，而是更多地将主要精力专注于并行算法的设计，同时可以充分利用 Matlab 提供的数值计算模块和数据显示功能，高效便捷地完成并行计算任务。

PCT 除了支持通用处理器构建的多核、多处理器和集群平台之外，还增加了对 GPU(Graphics Processing Unit)的支持。GPU 最早主要应用在图形计算领域，近年来 GPU 在通用计算领域的发展迅速。在 Matlab 中，可以通过 PCT、MEX 文件等多种方式利用 GPU 完成数据处理功能。

在 Matlab 中，并行程序开发与普通程序开发的差别很大，因此在开始应用 Matlab 并行计算功能之前，读者首先需要考虑以下几个问题。

(1) 并行计算的平台

平台是并行计算的载体,也是并行计算技术选择的最主要的依据。如果读者只有一个单核单处理器的计算机,那么最终用 Matlab 并行计算很难带来任何计算效率上的提高。如果读者拥有一台多核或多处理器的计算机,那么采用 Matlab 并行计算工具箱就可以满足一般的并行计算要求(Matlab 2010b 的并行计算工具箱最多可支持 8 个 worker,或者说可利用 8 个核的计算能力)。如果读者拥有多台计算机组成的集群,则需要利用 PCT 和 DCS 共同完成并行计算任务。如果读者拥有 GPU 处理器,则可以利用 PCT、MEX 文件等技术通过用 GPU 完成并行计算任务。

(2) 并行计算的复杂程度

Matlab 并行计算工具箱为常用的并行计算问题提供了相对快捷和简单的解决方案。比如,利用 parfor 可以对 for 循环进行并行处理,利用 SPMD 可以对单个程序多组数据的情况进行并行处理,因此读者在分析待解决的并行计算问题时,应重点考虑待解决并行计算问题的复杂程度。如果并行计算问题可以直接采用 parfor 或 SPMD 进行处理(或者可以分解为类似的问题),则应尽量选择 Matlab 并行计算工具箱内置的并行结构,这样可以大大提高用户解决问题的效率。如果面临的并行计算问题比较复杂,则可以采用更为通用的解决方案(比如创建 parallel job 或 distributed job),甚至可以创建 mex 文件采用多线程和 OpenMP 并行技术。

(3) 并行计算的数据通信问题

并行计算的目的主要有两个,一是提高计算效率,二是提高计算机的利用率。提高计算效率比较容易理解,某个问题如果采用单个 CPU 计算需要 10 小时,那么采用 10 个 CPU 进行计算可能只需要 1.5 小时就可以完成。在有些情况下,即使单个 CPU 的计算效率可以满足需求,但是如果图形界面和计算模块放在一个 CPU 上执行,将会对图形界面的操作性产生影响,从而影响用户使用软件的体验。此时,如果配置双 CPU 或多核的计算机,可以采用一个 CPU 或处理器核(下文统称处理单元)完成计算,另外一个处理单元完成图形界面显示或用户响应。这只是利用并行计算提高计算机利用率的一个例子。在实际应用中,并行计算不是总能带来令人满意的效果。在极端情况下,并行计算的效率反而会低于单处理单元的计算效率。因为并行计算本身也需要占用系统资源。在目前的处理体系中,多

线程或多进程是并行计算的基础。虽然 Matlab 并行计算工具箱已经隐藏了多线程或多进程操作，用户在程序设计的层面不用关心线程或进程操作，但是用户不能忽略并行计算带来的系统资源消耗。并行计算占用系统资源主要体现在两个方面：一是准备并行计算的必要操作，比如创建线程、创建进程、创建任务、并行计算启动前准备数据等；二是并行计算过程中各个并行任务之间的数据通信。并行计算运行之前准备引入的系统开销是一次性的，相对容易估算，但并行任务之间的数据通信需要引起特别的重视。在设计并行计算方案的时候，应尽量做到各并行任务之间的数据通信量最小化。特别是在网络速度较慢的集群计算平台上，任务之间的数据通信问题往往是提高并行效率的主要瓶颈。

上述三个问题也是本书重点讲解的三个问题。在并行计算平台方面，本书重点介绍了基于多核、多处理器、集群系统和 GPU 平台 Matlab 并行计算程序的开发方法。根据并行计算的复杂度方面，本书由简入难，从 Matlab 基本的并行结构逐步深入到 Matlab 并行计算数据类型，Matlab 通用并行程序设计，Matlab 中应用多线程技术、Matlab 中应用 OpenMP 基础和 Matlab 中应用 GPU 并行处理技术等方面的内容。考虑到不同的计算平台和不同的并行计算技术涉及的数据通信问题差异较大，因而将数据通信问题的讲解和说明分散到各章节中进行说明。

本书对基于 Matlab 的并行计算程序设计的原理进行了深入的剖析，对基于 Matlab 的并行计算程序设计方法和技巧给出了详细的说明，并在各章节给出了大量实例。通过阅读和学习本书的内容，读者可以掌握基于多种平台（多核、多处理器、集群和 GPU 等），利用多项技术（Matlab 并行计算工具箱、多线程 MEX 文件、OpenMP 和 GPU 等），完成 Matlab 并行程序设计的原理、方法和技巧。全书共分 10 章，其中：第 1 章为 Matlab 开发环境和程序设计基础；第 2 章为利用 parfor 对 for 循环进行并行；第 3 章为 SPMD 并行结构；第 4 章为其他 Matlab 并行结构；第 5 章为 Matlab 并行计算数据类型；第 6 章为 Matlab 通用并行程序设计；第 7 章为 MDCE 配置；第 8 章为创建多线程 MEX 文件；第 9 章为在 Matlab 中应用 OpenMP 进行并行计算；第 10 章为利用 GPU 并行执行 Matlab 程序。附录共包括三个部分，其中：附录 A 为 MEX 文件基础知识；附录 B 为 Matlab 并行计算配置项；附录 C 为 Matlab 并行计算常用概念说明。

本书的阅读对象包括大中专院校学生以及利用 Matlab 开发并行程序的科学家和工程师。书中对 Matlab 并行计算相关基础知识均进行了说明，因此读者可

以循序渐进地完成本书的阅读和学习。如果读者熟悉 Matlab 语言和 C/C++语言，将有助于理解和掌握本书的内容。

本书是较早对 Matlab 并行计算进行系统介绍的图书之一，书稿和书中实例虽经作者仔细斟酌和详细测试，但仍不免有错误或不足之处。如果读者阅读过程发现书中文字或内容存在不尽如人意之处，还望不吝指出。针对本书，北京航空航天大学出版社和 Matlab 中文论坛(http://www.iLoveMatlab.cn/)特别提供了读者与作者在线交流的平台(http://www.ilovematlab.cn/forum-212-1.html)，我希望借助这个平台实现与广大读者面对面的交流，解决大家在阅读此书的过程中遇到的问题，分享彼此的学习经验，从而达到共同进步。

刘 维

2011 年 10 月 5 日夜

目　　录

第 1 章　Matlab 开发环境和程序设计基础

1.1　本章导读

作为后续章节的基础，本章介绍了 Matlab 开发环境和程序设计的基础知识，重点针对 Matlab 环境、Matlab 语言基础、Matlab 数据类型和 Matlab 数据显示函数进行了说明。

- ✦ Matlab 环境是包括各种工具和语言的开发环境，重点介绍了命令行、代码编辑器、工作空间、历史命令、数据导入、自动生成代码、代码提示和帮助等方面的内容。
- ✦ Matlab 语言是一种基于矩阵运算的脚本语言，重点介绍了构成 Matlab 语言的基本元素、主要控制结构、运行 Matlab 语言的脚本（文件）和函数（文件），利用向量运算优化 Matlab 程序的方法和面向对象设计基础。Matlab 并行程序设计建立在 Matlab 串行程序设计的基础之上，熟练掌握和应用 Matlab 语言非常必要。
- ✦ Matlab 变量只有一种数据类型即阵列，但 Matlab 阵列类型非常丰富。既有较为简单的、存储单一类型数据的数值阵列、字符阵列、逻辑阵列，又有较为复杂、可以存储不同类型数据的结构体阵列和元组阵列，同时，还有可以存储 Matlab 函数的函数句柄阵列。掌握和熟悉 Matlab 数据类型相关知识和操作是完成 Matlab 并行程序设计的关键之一。
- ✦ 除数据处理之外，数据可视化是 Matlab 具备的另外一项重要功能。在利用 Matlab 语言开发的各种应用中，数据可视化函数的应用非常普遍。因此，对常用的数据可视化函数和技术进行了简要说明。

1.2　Matlab 环境

Matlab 的前身是一组由 Fortran 语言编写的矩阵计算函数库。调用 Fortran 或 C 语言函数库的一个缺点是函数接口不统一，数据类型复杂。比如同样是矩阵求和操作，整型和浮点型需要调用不同的函数。再比如，将矩阵和常量相加在逻辑上与矩阵和矩阵相加相同，但需要调用不同的函数接口。Matlab 的出现很好地解决了这一问题。在 Matlab 中只有一种数据类型，即阵列。同时，Matlab 将各种逻辑上相同的矩阵操作的函数接口通过合理的软件设计进行封装，极大地方便了用户的使用。

发展到现在这个阶段，Matlab 已经变成一个非常庞大的应用系统。通常说的 Matlab 是一个笼统的概念，利用 Matlab 解决某一问题，一般可以有如下两种含义。

- ✦ 通过图形化操作，利用 Matlab 环境提供的工具或应用解决某一问题。
- ✦ 通过 Matlab 编程语言直接开发或调用其他开发者编写的程序（工具箱）解决某一问题。

为了避免混淆，对 Matlab 环境和 Matlab 语言分别进行说明（后续章节均遵循这一说明）。

1) Matlab 环境指 Matlab 软件可提供的所有辅助程序设计的工具和应用的总称。

Matlab 环境是开发、运行和调试 Matlab 代码的主要场所，Matlab 环境提供的与程序设计密切相关的工具和应用主要包括：代码编辑器(Editor)、命令行窗口(Command)、工作空间(Workspace)窗口、代码分析器(Profiler)。随着 Matlab 的不断升级换代，Matlab 环境可提供的功能越来越丰富，越来越强大。用户不用编写代码，只通过界面操作就可以完成相当多的功能。

2) Matlab 语言指可以在 Matlab 环境中执行、符合 Matlab 环境语法要求的计算机语言。

Matlab 语言有时简称为 M 语言。Matlab 语言可以直接输入到命令行窗口中执行，也可以保存在扩展名.m 的文本文件中通过 Matlab 命令行窗口执行，或者由 Matlab 提供的 pcode 函数编译为扩展名为.p 的加密文件通过 Matlab 命令行窗口执行。在 Matlab 环境中，可以混合使用 M 语言、C/C++语言、Fortran 语言等完成数值计算和数据可视化功能，通过多语言混合程序设计，可以增强和扩展 Matlab 的功能。非 M 语言一般被编译为 MEX 文件，MEX 文件在 Matlab 环境中的调用方式与 M 语言的函数相同。

1.2.1 命令行窗口

在介绍命令行窗口之前，首先明晰一个概念。在 Matlab 程序设计中，命令和语句的界限比较模糊。在本书中，凡是通过命令行窗口直接输入的代码统称为命令，凡是通过代码编辑器或者其他文本编辑器保存在*.m 文件中的代码统称为程序语句或代码。

命令行窗口可能是用户最先使用、打交道最多的 Matlab 工具窗口之一。原则上，用户在 Matlab 命令行窗口中通过输入指令可以完成 Matlab 环境支持的所有功能。所有的 Matlab 代码或命令均可以通过命令行窗口执行。通过 Matlab 命令行窗口可以执行单条命令、多条命令以及符合 Matlab 语法要求的任何程序代码。

比如：

```
≫a=1;
≫b=2;
≫c=a+b
c=
     3
≫
```

上述命令中，“≫”为 Matlab 命令行窗口命令输入提示符。

上述代码段中，语句 a=1 和 b=2 表示对变量 a 和变量 b 赋值，c=a+b 表示执行变量 a 和变量 b 的求和操作。每条命令或语句后面的“;”号用于控制命令的输出结果。如果命令或语句后面带“;”，则不输出命令或语句的执行结果；反之，则输出命令或语句的输出结果。从上述代码段中可以看出，M 语言应用非常灵活，用户不需要编译程序，就可以单条执行语句或命令。

Matlab 命令的种类繁多，具有强大功能。在 Windows 操作系统中，在 Matlab 命令行窗口中甚至可以通过“dos”命令直接执行 dos 指令。比如，可以通过 dos 命令 hostname 查看当前工作机的主机名，在 Matlab 中的执行方式如下所示。

```
≫dos('hostname');
THINK
```

```
≫
```

或者通过 dos 命令执行 Windows 关机命令(下述第一条命令是通知 Windows 1 小时后关机，第二条命令是取消执行 Windows 关机命令)。

```
≫dos('shutdown /s /f -t 3600');
≫dos('shutdown /a');
```

1.2.2　代码编辑器

在 Matlab 命令行窗口中执行单条语句适用于简单的应用。对于复杂的应用，如果也通过单条语句的方式执行的话，就费时费力了。用户可以创建脚本(script)文件或函数(function)文件解决将这一问题。编辑脚本文件或函数文件的工具即代码编辑器 Editor。启动代码编辑器有两种方式，一种是通过 edit 命令启动，另外一种是通过选择 File→New→Script 或 Function 菜单项启动。启动后的代码编辑器界面如图 1-1 所示，用户可以在代码编辑器中完成程序代码的编辑、调试和性能分析等操作。

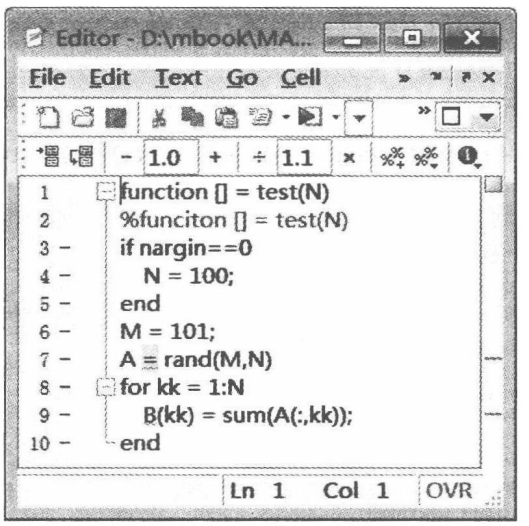

图 1-1　Matlab Editor 窗口

Matlab 代码编辑器的代码编辑功能与普通文本编辑器的类似，不同的是前者增加了自动注释(Ctrl+R 快捷键和 Ctrl+T 快捷键)、代码错误自动提示等功能。除此之外，采用代码编辑器可以完成的最常用功能为代码调试和代码性能分析，下面分别进行说明。

(1) 代码调试(主要功能查看 Debug 菜单)

通过 F12 快捷键可以在当前光标所在处插入一个断点，这样通过快捷键 F5 执行当前函数文件或脚本文件时，程序会在断点处停止，用户可以通过命令行窗口或工作空间查看 Matlab 变量的内容，完成程序调试功能。在断点处，通过 F10 或 F5 快捷键继续执行后续 Matlab 代码。

(2) 代码性能分析

通过代码编辑器可以直接启动 Matlab 代码分析器(Profiler)。通过代码分析器，用户可

以对 M 程序中耗时较长的语句进行分析，找到程序中执行效率比较低的部分，然后再采用针对性的手段解决这一问题。

通过选择 Tools→Open Profiler 菜单项打开代码分析器(Profiler)，在代码分析器中输入待分析的函数或命令(这里采用 test 函数作为分析对象，如图 1－1 所示)，然后按 Enter 键执行此命令。待命令执行完毕后，Matlab 代码分析器会生成分析结果。如图 1－2 和图 1－3 所示，其中图 1－2 展示了 test 函数执行的信息概要，图 1－3 展示了 test 函数各行代码性能分析的结果。

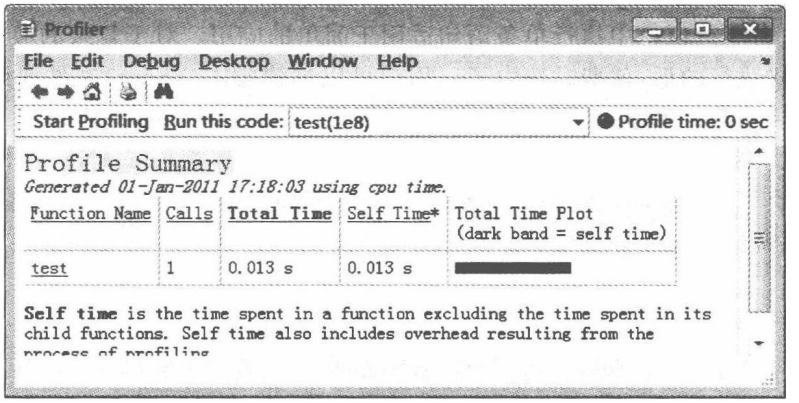

图 1－2 Matlab Profiler 窗口—test 函数执行的信息概要

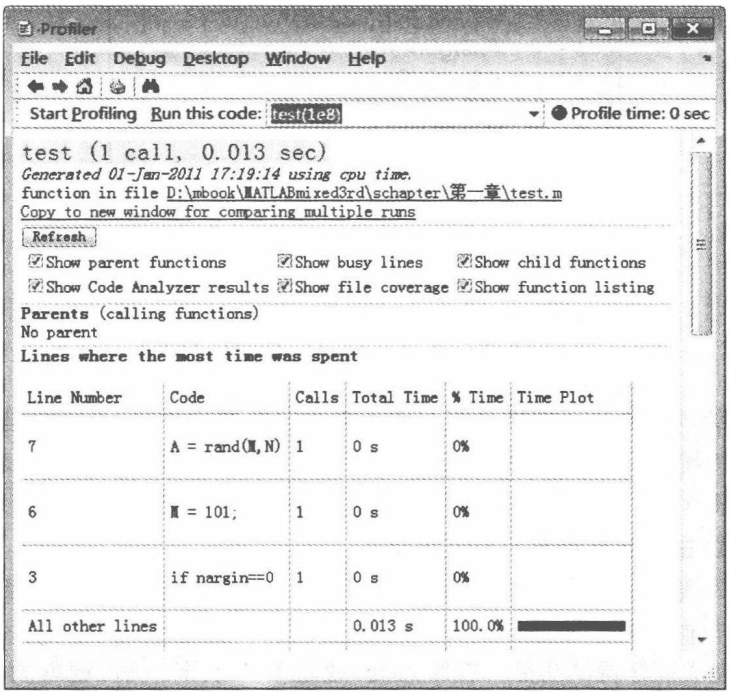

图 1－3 Matlab Profiler 窗口—test 函数各行代码性能分析的结果