

气象业务现代化

应用 OpenMP 在 Linux 下实现 MM5 多线程运行

杨 森 周小珊

(中国气象局沈阳大气环境研究所,110016)

提 要

介绍了如何在 Linux 操作系统下,应用 OpenMP 编译选项实现 MM5 的多线程(并行)运行。笔者从用户使用 OpenMP 时可能出现的问题,到寻求解决问题的思路,到问题的最后解决,进行了说明。同时,介绍了使用 OpenMP 的优点。

关键词: OpenMP 多线程运行 Linux MM5 模式

引 言

中尺度数值预报模式 MM5 目前已广泛应用于各级气象部门的科研和业务工作,随着 MM5V3 版本中 Linux 操作系统下运行选项的出现,MM5 已经可以顺利的在 PC 机上运行,这对暂时无法购置巨型机和工作站的单位和科研人员使用 MM5 模式提供了很大的方便。一般用户在使用 MM5 时多选择单 CPU,因此,仅下载 MM5.tar.gz 即可。能够使用多 CPU 进行并行计算的用户,一般选择 MM5 中的并行计算版本 MPP,即下载

MPP.tar.gz,在 mpi 环境下运行 MPP 模式。事实上,在一般的 MM5 模式(非 MPP)编程中,已经考虑了模式的并行计算,在 Fortran 源程序中,都有并行计算指导语句来实现共享内存多线程计算。随着计算机技术的发展,微机双 CPU、多 CPU 的使用将越来越广泛。因此,一般用户在其使用的 MM5 模式中,只需要在 configure.user 里的编译选项里,选择并行计算的 OpenMP 选项(-mp -Mnosgimp),即可在 Linux 操作系统下实现 MM5 的多线程运行,而无须使用 MPP 版

本。这里介绍一下 OpenMP 的使用方式。

1 问题的提出

MM5 主模式的 configure.user 中,存在 Linux 下应用 pgf 编译器编译 mm5 的 OpenMP 并行版本的选项:

```
FCFLAGS=-I$(LIBINCLUDE) -O2 -Mcray = pointer -tp p6 -pc 32 -Mnoframe -byteswapio -mp -Mnosgimp
```

但用户在选择该选项,编译后,试运行时,会遇到如下错误信息:

segmentation fault (段错误)

因此,程序无法继续执行。

笔者使用的计算机配置为:PⅢ 800MHz 双处理器,256 MB 内存,RedHat Linux 6.2,PGI 3.1。

2 解决的方法

首先,在 www.google.com 上查找关键字“mm5 segmentation fault”,可以在网址上 www.mmm.ucar.edu/mm5/mm5v2/mm5v2-PC.html 找到答案。原来是堆栈太小。原文为:

If you are getting strange error when running the model (such as 'Segmentation fault'), it may be due to lack of stack size. (在运行模式时如果遇到奇怪的错误(如 Segmentation fault),可能是由于堆栈不足造成的。)

按照其给的解决方法,使用户的堆栈使用大小不受限制,执行“ulimit -s unlimited”。但重新运行 MM5 后仍出现上面的错误提示,问题仍未解决。

随后,笔者阅读了 pgf90 的运行选项说明,考虑在 pgf90 的编译选项上增加 -Mchkstk(检查堆栈)的选项,这样在重新编译后运行时,出现了如下错误提示:

```
** ERROR: in routine solve there is a stack overflow: thread 0, max 8180KB, used 1660KB, request 10230168B
```

译文: ** 错误:在子程序 solve 中出现堆栈溢出:线程 0,最大 8180KB,已用

1660KB,需要 10230168B。

PGI 的用户指南对此条信息作了说明: PGI 编译器使用 linuxThreads 软件包实现 Linux 上的 SMP 并行。由于 Linux 本身的 linuxThreads 软件包或许与 PGI 编译器存在着不兼容的可能,所以 PGI 自带了此函数库位于 \$PGI/linux86/lib/libpthread.a。在缺省情况下,此版本的函数库会在系统函数库调用之前被加载。此函数库只在编译时选择 -Mconcur 或 -mp 选项时会被用到,也就是说以上出现的最大 8M 的限制是这个函数的问题。

为了寻找解决方法,笔者在互联网上继续搜索“pgi stack size”,最后还是在 MM5 的 HELPDESK 上找到如下信息:可以从网上下载 linuxThreads 的源程序,自己修改其最大的堆栈大小,再重新编译。原文如下:

Subject: MM5 on dual-processed Linux boxes

If you want to do symmetric multiprocessing (not MPI), you will need to increase the default stack size and recompile the libpthread.a library (this is part of the gnu C library).

即:在 Linux 双处理器系统上的 MM5,如果要使用对称多处理器运算(而非 MPI),你需要增加缺省的堆栈大小,并重新编译 libpthread.a 函数库(为 gnu C 函数库的一部分)。

由于笔者的计算机安装的 Linux 版本为 redhat6.2,其上的 glibc 为 2.1.2 版,因此按照其提示的 ftp 站点

ftp://prep.ai.mit.edu/pub/gnu/glibc
下载对应的软件包:

glibc-2.1.2.tar.gz,

glibc-linuxthreads-2.1.1.tar.gz,

glibc-crypt-2.1.tar.gz。

解开上述打包文件后,修改 linuxthreads 目录下的 internals.h:

① 修改其中定义的宏 STACK_SIZE

STACK_SIZE 原来的大小为 $(2 * 1024 * 1024)$,即为 2M 字节,因为此数值大小必须为 2 的正整数次幂,将其改为 $(64 * 1024 * 1024)$ 。然后,重新编译。

② 编译后将在 linuxthreads 目录下的 3 个文件, libpthread.a, libpthread.so, libpthread.so.0 拷贝到 \$PGI/linux86/lib 目录下替换原有文件。

③ 再重新编译 mm5,运行时仍会遇到上面的错误提示:

```
* * ERROR: in routine solve there is a
stack overflow: thread 0, max 8180KB, used
1660KB, request 10230168B
```

这时,执行“ulimit -s unlimited”使得用户的堆栈大小不受限制,这样程序就可以正常运行了。

但这时程序只有一个线程。

④ 通过设置环境变量 export OMP_NUM_THREADS=2,即可以使用 2 个线程进行计算了。

3 OpenMP 在使用上的优点

① 在 Linux 操作系统下使用同一台计算机 PIII 800MHz 双处理器, 256 MB 内存,对同一个个例,分别用 MPP 模式和在 MM5 中选择 OpenMP,进行计算,并对比计算速度。结果表明,利用 OpenMP 在 Linux 下实现 MM5 多线程运行,具有略高于使用 MPP 并行计算的速度。

② 用户只需下载一套 MM5,不用考虑 MPP,就可以根据自己机器的情况,很方便地选择使用单 CPU 或多 CPU。

③ 非常重要的一点是:MM5(非 MPP)的 OpenMP 编程时,由于 OpenMP 的 fork-join 结构,仅在计算循环语句时考虑并行计算,因此,不熟悉并行编程的用户也可以根据自己的需要对原代码进行修改,同时可以在任何位置随时输出自己需要的变量。而 MPP 由于采用 MPI 的将消息传递编程方法的标准化、规范化编程,编程难度大,不易于用户随意改动,因此用户如果想在程序中实现一些自己的想法,则较困难。

4 体会

(1) MM5 是一种普遍使用的中尺度模式,其在各种平台上都有众多的用户。在遇到问题时可以借鉴其他人的经验。

(2) Linux 是源代码开放的操作系统,其下的函数库也都有源代码可供下载,在遇到限制时可以自行修改,使其满足使用的需要。

(3) 大力利用互联网资源共享的优势,是日常工作中解决一些问题的行之有效的方法。但是互联网上的资源浩如烟海,这时就要借助搜索器的优势,在本文中就利用 www.google.com 进行了两次搜索,由于进行了多关键字的组合选取,搜索结果的第一个相关内容就是所想要的信息。

Using OpenMP to realize Multi-threads run of MM5 under Linux

Yang Sen Zhou Xiaoshan

(Shenyang Atmosphere Environmental Institute 110016)

Abstract

How to use OpenMP compiler option to realize Multi-threads (parallel processing) run of MM5 under Linux is introduced. The problem which may be presented when using OpenMP is discusses. The advantage of using OpenMP is also presented.

Key Words: OpenMP multi-threads processing Linux MM5