

GRAPES 模式软件框架与实现*

伍湘君¹⁾²⁾ 金之雁¹⁾ 黄丽萍¹⁾ 陈德辉¹⁾

¹⁾(中国气象科学研究院,北京 100081)

²⁾(国防科技大学计算机学院,长沙 410073)

摘 要

GRAPES(Global and Regional Assimilation and PrEdiction System)是中国气象科学研究院数值预报研究中心自主开发的新一代静力/非静力多尺度通用数值预报模式。该模式采用标准化、模块化软件体系结构,并严格按照软件工程要求完成系统开发,包括程序的并行计算。初步试算结果表明:GRAPES 模式软件框架的设计和实现符合模式发展的要求,为中国数值天气预报系统的可持续发展奠定了良好的基础。

关键词: GRAPES 数值预报 标准化设计 多尺度通用 并行计算

引 言

GRAPES(Global and Regional Assimilation and PrEdiction System) ^{①②}是一个以多尺度通用动力模式为核心的、以统一软件编程标准为平台的新一代数值预报模式系统。该模式不仅采用了多尺度统一模式的设计思想,在模式软件设计上也采用了模块化、标准化等现代软件设计思想,目的是为了使 GRAPES 的研究开发能够吸纳国内外数值预报的优秀成果,方便科学家开展联合工作和交流。

作为一个大型数值预报软件,GRAPES 模式的开发体现了软件工程的工作模式^[2]。在以往的模式开发过程中,不同模式的软件结构之间互不相同,甚至同一模式内部也同时存在多种不同的编程风格,使用不同的接口协议。这种开发方式使得各模式之间的成果很难交流,也阻碍了让更多的科学家参与到模式的研发中来。为了实现 GRAPES 模式软件系统标准化、模块化和软件开发的可靠性、可用性和可扩展性,为软件开发的控制和管理提供保障,缩短软件开发周期,减少程序错误,提高模式可读性和可移植性,为 GRAPES 数值预报系统的可持续发展奠定基础,GRAPES 程序系统在开发初期即完成了《中国气象数值预报创新软件编程标准》^②的编写,并严格按照《编程标准》的要求,完成 GRAPES 软件系统各阶段的任务,包括科学文档的编写、程序结构的设计、数据流程的确定。在 GRAPES 系统软件编写过程中,注意保持程序良好的编程格式和程序结构,并严格遵循软件包编写“可插拔”的思想,保证软件包的独立性。

* 国家十五科技攻关重点项目“中国气象数值预报创新研究”资助。

2004-03-25 收到,2004-06-10 收到修改稿。

① 陈德辉主持撰写,《新一代多尺度数值预报系统(区域)》技术手册(内部交流)。

② 中国气象科学研究院数值预报研究中心编写,《中国气象数值预报系统软件编程标准》(内部交流)。

GRAPES 软件程序设计的目标包括:①保证模式系统软件的通用性,使 GRAPES 模式软件框架可适合于不同的应用以及计算机平台;②使用 Fortran 90 的模块化编程语言,确保模式各软件包的独立性;③提供多层次并行计算功能,使模式能在不同结构计算机平台上高效运行;④提供与国际上常用数值预报资料的接口,使模式输入输出方便灵活。

1 GRAPES 模式的软件框架

GRAPES 模式是采用半隐式半拉格朗日方案的有限区/全球统一模式, WRF(The Weather Research and Forecast) ^③模式则是一个采用欧拉方案的有限区模式。按照中国气象科学研究院数值预报研究中心与 WRF 研究开发项目组关于 GRAPES 模式与 WRF 模式协同开发工作协议, GRAPES 模式与 WRF 模式采用共同的软件框架、共同的编程规范、相同的软件模块接口,以便于两模式系统的高度兼容,双方可以方便地利用对方的研究成果。为此, GRAPES 模式对原来的软件结构进行适当地调整,采用与 WRF 模式共用的软件框架,这包括 3 个方面的内容,即:3 层结构设计、变量注册工具以及程序并行计算实现时需要调用的 1 个外部函数包^[4~5]。通过保持这 3 个方面的一致,使得 GRAPES 模式可以方便地与 WRF 模式交流,并大大加快了 GRAPES 模式的软件开发。可以说,先进的动力框架内核、标准化的程序设计,再加上良好的软件实现方法,保证了 GRAPES 模式能够实现最初的设计目标。

1.1 软件框架

GRAPES 的软件框架与 WRF 的软件框架一致,均采用 3 层结构形式,如图 1 所示。

在图 1 中, GRAPES 模式软件框架分为 3 层,即驱动层、中间层和模式层。软件框架顶层为程序驱动层,驱动层程序与具体的科学计算无关,它处理与计算环境、计算结构有关的所有问题,包括计算区域的设定、变量内存空间的分配与释放、并行计算区域的划分、积分参数的设定、初始资料的读入等。在 GRAPES 的驱动层中有两个非常重要的程序模块,它们是 module_configure.F 和 module_domain.F。module_configure.F 模块定义了模式配置参数记录类型,并给出所有与模式配置参数有关的全局变量定义和子程序; module_domain.F 模块则给出了模式预报区域的类型定义,以及有关预报区域处理的子程序。

驱动层的数据信息通过中间层传递给模式层进行各种物理计算。在 GRAPES 中,中间层的代表程序是 solve_grapes,它控制 GRAPES 模式一步积分内的所有处理过程,将模式驱动层给出的域(domain)变量分离成模式的预报变量。在分布式并行计算环境中,中

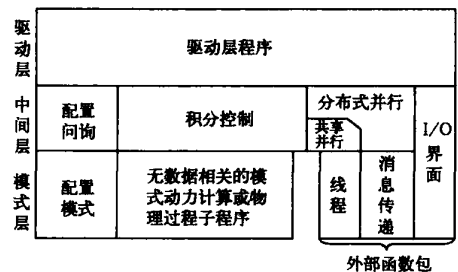


图 1 GRAPES 模式软件框架

③ Weather Research and Forecast Model 1.2: Software Design and Implementation. WRF 技术手册.

间层同时完成不同节点间由于模式层计算而产生的变量内存边界信息交换,这也就是我们常说的重叠(halo)区交换。中间层之下为模式层,模式层程序不涉及并行计算的有关问题。

模式层是模式计算的核心层,包括具体的数学物理计算,如模式线性及非线性项计算、上游点计算、赫姆霍兹(helmholtz)方程求解、各种物理过程计算等等。到达模式层的变量应是具体的、具有各种物理含义的变量,这可以保证科学家按照自己的思路编写程序,而无需顾及问题之外的内容。模式层程序的计算数组大小由驱动层设定,在分布式并行计算方式下,其计算区间是预报区域的一个子区间。在模式层的同一计算模块内,各子区间的物理量之间应保证计算无相关。另外,部分 I/O 操作和 GRAPES 模式提供的一个底层并行实现函数包也处于模式层中,如线程和消息传递。在 1.3 节中,我们将介绍有关 GRAPES 共享内存并行计算和分布式并行计算的相关内容。

图 2 是 GRAPES 模式软件的一个计算流程示意图。如图所示,程序运行由驱动层开始,首先进行程序的初始化(包括并行初始化)操作,包括:各程序模块的初始化设置(子程序 init_modules);名表(namelist)控制变量的读入(子程序 initial_config);预报域的定义、计算区间的划分(并行计算)、变量空间的分配(这 3 个部分均由子程序 alloc_and_configure_domain 调用完成);初始资料的输入/输出(I/O,子程序 init_grapesio)。完成初始化操作之后,模式启动积分(integrate)计算,将需积分计算的物理变量从模式框架中剥离出来(子程序 solve_interface,solve_grapes),并传递给模式层。在 GRAPES 模式的 1 个积分时间步内,完成模式动力框架积分计算和诸如陆面、辐射、积云降水、微物理计算等物理过程的计算,同时完成积分时间递增。积分循环在子程序 integrate 内控制,该子程序同时控制程序的历史数据与后处理数据输出频率。

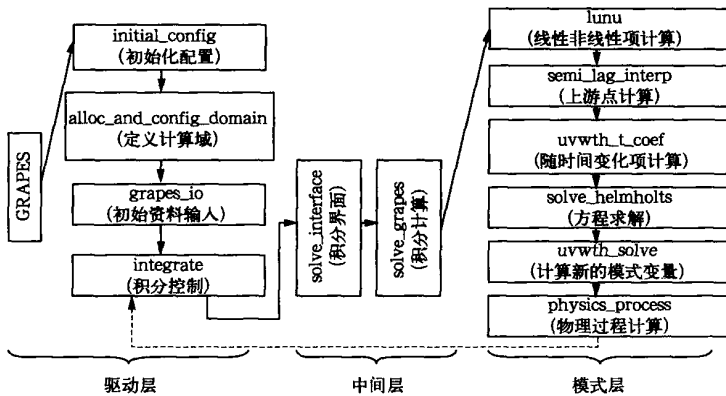


图 2 GRAPES 模式程序流程示意图

需要强调的是,GRAPES 模式软件程序中,有两个数据派生类型非常重要,它们就是上面驱动层提到 module_domain.F 模块定义的域(domain)和 module_configure.F 模块定义的模式配置记录(model_config_rec)。Domain 定义了 GRAPES 模式运行的所有操作对象,如风速、温度、高度等,model_config_rec 则记录了模式运行需要的所有控制变量值,如 domain 域的经纬度范围、格点大小、物理参数选项等。可以说,GRAPES 模式的运行就是

围绕着这两个数据对象进行的。

1.2 变量注册表(Registry)机制

GRAPES 模式的变量管理方式与 WRF 相同,即采用变量注册的方式完成模式变量的登录。

Registry 变量注册分为两部分,即注册表和注册程序。注册程序完成将注册表登记注册的模式变量转变为模式程序中可处理的形式,体现在外部即生成若干以 inc 为后缀的文件。这些文件包括模式变量定义、变量空间分配、变量输入输出等模式变量操作;运行控制变量(namelist 变量)定义、缺省值和 I/O 控制;过程哑元、过程哑元说明、调用子过程实元等哑实结合方式。其他还包括:同一积分步内变量(II 变量)说明、物理过程参数传递、并行 halo 区的通讯交换等等。可以说,变量注册程序为模式各部分的接口建立了一个方便的工作平台,程序设计者只要将模式需要的变量按变量注册表的要求填写,即可完成以往程序编写过程中因变量增减而带来的大量工作。

图 3 是 GRAPES 模式利用注册机进行变量注册的一个简单工作流程示意图。从图中也可看到,模式变量注册之后,形成若干程序内含文件,这些文件包括:模式状态变量说明类(状态变量定义、分配、过程实函数、过程哑元、过程哑元说明),模式控制变量说明类(namelist 变量定义、namelist 变量缺省值定义、namelist 变量输入/输出、namelist 变量到状态变量的复制),II 变量定义类、变量输入/输出操作类、物理过程常数赋值定义类、并行通讯交换定义类等等。这些内含文件利用计算机系统提供的预编译功能,以“包含(include)”方式插入程序中,大大方便了程序的协调管理,减少了程序出错机会。

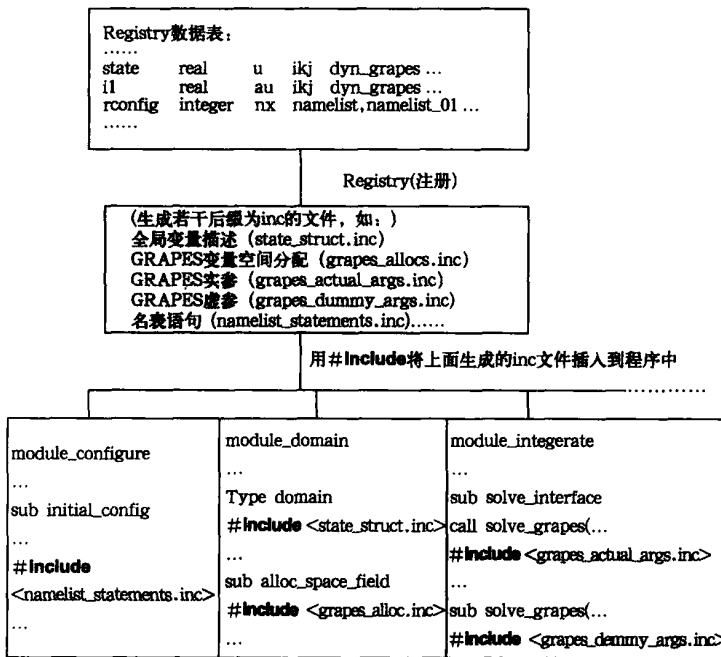


图 3 GRAPES 变量注册工作流程示意图

注册表除了进行模式变量的登记外,另一作用是定义模式并行计算中需要进行信息交换的 halo 区域,这将在 1.3 节作说明。

通过模式注册,GRAPES 程序很好地将软件框架的 3 层结构衔接起来,使模式预报域的建立(module_domain)和模式配置记录的设置(module_configure)从模式层的计算中分离出来,极大方便了模式设计人员的程序编写。对于一个只关心模式层计算的科学家而言,当需要增加一个物理全局量时,只需在 Registry 表中增加相应的变量名,通过变量注册,即可在中间层(如 GRAPES 的 solve_grapes 子程序)得到该变量的定义,通过与模式层子程序的虚实结合,进入模式层计算。

1.3 并行计算

对于一个大型数值预报模式而言,并行计算已成为其必需的结构特征。为此,GRAPES 模式从程序框架做起,统一设计模式的并行计算方案,为不同使用提供选择^[6]:一是针对分布式内存计算机系统而设计的区域分块,即将需要进行计算的整个区域按照计算机节点个数做区域划分,划分得到的子区域即称之为一个块(patch),每个计算机节点上只完成本节点 patch 的计算,且只存储相应的 patch + halo 数组空间(图 4),当计算涉及到其他 patch 区域时,必须事先进行相应 halo 区的信息交换。这也就是通常所说的多任务并行计算;另一种是针对共享内存计算机系统完成的瓦片(tile)计算,也就是多线程并行计算。tile 是在 patch 内按本节点可用线程数进行区域划分,因此同一节点上不同 tile 间内存共享,只是计算区间不同(图 4)。两种并行计算方案相辅相成,既可单独使用,亦可混合使用。但对研究模式科学问题的科学家而言,GRAPES 的并行计算方案仅在程序框架体现,因此模式并行计算对模式层是透明的。

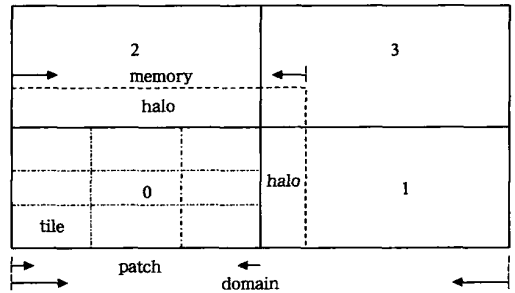


图 4 GRAPES 的并行计算区域划分

GRAPES 并行计算程序由 MPI + Fortran 90 + c 完成^[7]。在底层,GRAPES 并行计算利用 WRF 提供的一个消息通讯外部函数库 RSL 以及此函数库上层若干应用通讯过程,如:并行程序初始化,并行计算结束,并行区域划分,数据分发和收集,数据广播等。这些函数过程结合模式注册表 Registry 中变量 halo 区交换定义,为 GRAPES 模式软件并行实现提供了方便,使程序并行化工作能够侧重于方案的正确性检验而减少了程序开发的繁琐。Registry 中 halo 区信息交换定义方式如下:

```
halo name npts: u, v, ...
```

其中“halo”为关键词,“name”为本次信息交换的命名,“npts”是数组进行信息交换的大小,可以是 8, 24, 48, 80, 120, 分别表示交换 1 圈、2 圈、3 圈、4 圈和 5 圈。“u、v...”是需要进行 halo 区信息交换的变量名。

GRAPES 区域模式并行计算对预报区域做二维水平划分,即并行分区在 X 和 Y(通常所说的经向和纬向)方向上进行。图 4 反映了 GRAPES 区域模式并行计算方案的划

分,利用 RSL 库提供的功能,划分可以是有胖瘦的。这种计算负载的分配可根据实际计算环境中计算负载平衡来考虑^[8-9]。

由于每个处理节点在并行计算时只保证本节点计算变量的更新,即得到 patch 范围内变量新值,因此当模式下一步计算需要用到 halo 区变量值时,必须进行 halo 区变量值的更新。如图 4 所示例中 0 号节点需要与 1,2,3 号节点交换 halo 区变量信息。这只需在变量注册表 Registry 中用 halo 关键词注册需要进行信息交换的变量名即可,halo 交换允许设定交换不超出变量存储空间范围内任意 halo 区的大小。

1.4 I/O

目前,GRAPES 模式的输入/输出采用比较简单的方式。模式的初始值和侧边界由模式的标准初始化提供,模式可以按要求输出模式场和经过后处理的等压面场预报结果。

2 初步计算结果及计算性能测试

截至 2004 年 10 月,GRAPES 区域模式软件设计已基本完成,并发布了 GRAPES V2.1 版本。GRAPES V2.1 模式版本在多种计算机平台上进行了试算,这些平台包括 alpha 工作站、pc-cluster 微机机群、IBM 计算机以及国产巨型计算机等。在这些机器上的试算结果表明,GRAPES V2.1 模式版本具有良好的可移植性,在各种并行计算机上计算稳定、有效。

图 5 给出的是 GRAPES V2.1 模式版本以 2003 年 7 月 8 日 T213 资料为初始场,在 pc-cluster 微机机群上串行、并行预报 24 h 结果比较。该 pc-cluster 是中国气象科学研究院数值预报中心科研人员自主搭建的一个微机集群,共装有 32 个节点,每个节点 1 个 CPU,2 G 内存。从图 5 可以看到,GRAPES 利用 8 个 CPU 并行计算 24 h 预报结果与串行计算结果一致。

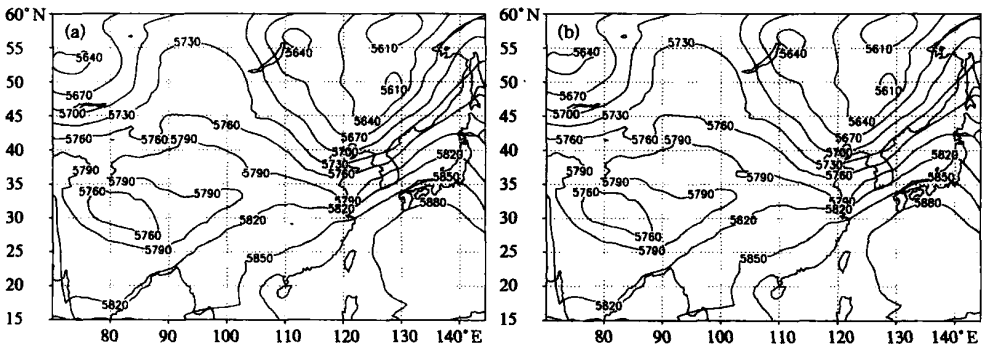


图 5 GRAPES 模式串行、并行积分 24 h 500 hPa 高度场比较(单位:gpm)
(a) 1 个 CPU,(b) 8 个 CPU

图 6 给出了 GRAPES V2.1 模式在 pc-cluster 上运行中国区域,分辨率为 20 km × 20 km,时间积分步长取 600 s,进行 24 h 积分计算时在不同 CPU 数下的计算时间(墙钟

时间)比较,由于机器内存量的限制,没能得到 1 个 CPU 和 2 个 CPU 的计算时间。图 7 是 GRAPES 模式在中国气象局 IBM 计算机上运行中国区域,分辨率为 $20 \text{ km} \times 20 \text{ km}$,时间步长为 120 s ,积分 24 h 使用不同 CPU 数的墙钟时间的比较。

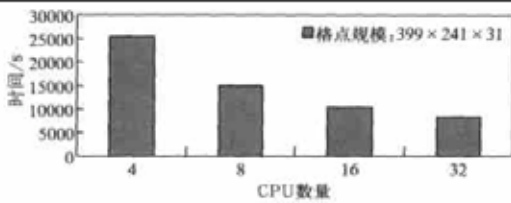


图 6 GRAPES 模式在 pc-cluster 计算时间

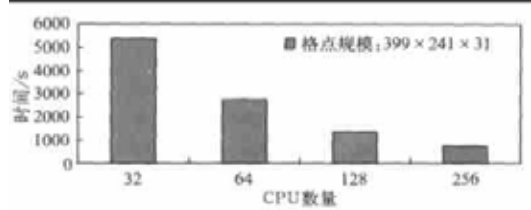


图 7 GRAPES 模式在 IBM 计算机的计算时间

由上面的结果可以看到: ① GRAPES V2.1 并行计算结果稳定有效; ② GRAPES V2.1 模式并行计算大大改善了计算的墙钟时间,保证预报及时可用; ③ 当计算规模增大到超出机器内存限制时,单节点串行计算已无法完成,但通过采用并行计算方式,GRAPES V2.1 模式可以方便地通过增加 CPU 数量达到减少单机内存需求的目的,从而使科研人员可以充分利用现有的机器资源,进行更大范围或作更精细分辨率的数值试验。

3 小 结

GRAPES 的软件框架为其未来发展奠定了一个良好的工作基础,程序 3 层结构思想和模式变量注册机制,使得对程序的修改和维护都十分方便,降低了模式开发完成后进行模式改进的难度。其标准化、模块化的程序设计框架不仅有利于模式在不同计算机平台的实现,而且为模式本身的发展提供了方便。从初步测试结果也可看到模式并行计算是有效和可靠的,但我们认为就模式计算而言,下面几点是可改进的:

- (1) 模式某些核心算法,如 hel mhotz 方程求解算法的进一步优化。
- (2) 模式 I/O 接口有待完善。
- (3) 共享内存并行计算(omp)有待改进。

作为一个以发展网格距离从 1 km 到 100 km 区域与全球模式统一、天气预报模式与短期气候预测模式通用、研究与业务通用为目标的格点模式,未来 GRAPES 的发展还将经历长久的过程。在这一发展过程中,GRAPES 模式系统将得到不断的完善,其模式软件框架也将更标准,程序性能得到进一步的提高,并最终实现成为一个具有中国自主知识产权的先进的数值预报系统。

参 考 文 献

- 1 陈德辉,杨学胜,张红亮,等. 多尺度非静力通用模式框架的设计策略. 应用气象学报,2003,14(4): 452 ~ 461.
- 2 Eric J Braude. 面向对象的软件工程. 北京:电子工业出版社,2003.
- 3 Michalakes J, Dudhia J, Gill D, et al. Design of A Next-generation Regional Weather Research and Forecast Model, Towards Teracomputing. Proceedings of the eighth ECMWF workshop on the use of parallel processors in meteorology.

- World Scientific, River Edge, New Jersey, 1999. 177 ~ 123 .
- 4 John G Michalakes, Michael McAttee, Jerry Wegiel. Software infrastructure for the Weather Research and Forecast model. [http:// www . wrf- model . org](http://www.wrf-model.org) .
- 5 Michalakes J, Canfield T, Nanjundiah R, et al. Parallel Implementation, Validation and Performance of MM5, Coming of Age. Proceedings of the sixth ECMWF workshop on the use of parallel processors in meteorology. World Scientific, River Edge, New Jersey, 1995. 266 ~ 276 .
- 6 金之雁, 王鼎兴. 一种适用于有限差分模式的负载均衡区域分解方法. 气象学报, 2002, 60(2): 188 ~ 193 .
- 7 都志辉编著. 高性能计算并行编程技术. 北京: 清华大学出版社, 2001 .
- 8 金之雁, 王鼎兴. 一种在异构系统中实现负载均衡的方法. 应用气象学报, 2003, 14(4): 410 ~ 418 .
- 9 金之雁. 异构系统动态负载均衡技术研究: [博士学位论文]. 北京: 清华大学博士论文, 2003 .

THE SOFTWARE FRAME WORK AND APPLICATION OF GRAPES MODEL

Wu Xiangjun¹⁾²⁾ Jin Zhiyan¹⁾ Huang Liping¹⁾ Chen Dehui¹⁾

¹⁾ (Chinese Academy of Meteorological Sciences, Beijing 100081)

²⁾ (Computer school, National University of Defence Science & Technology, Changsha 410073)

Abstract

GRAPES (Global and Regional Assimilation and PrEdiction System) is a new-generation general hydrostatic/ non-hydrostatic, multi-scale numerical model of China developed by Research Center for Numerical Meteorological Prediction of CAMS (Chinese Academy of Meteorological Sciences) of CMA. The model adopts a structure of standardized and module-based software and is developed in accordance with the strict requirement of software engineering, including the program-operated parallel calculation. Preliminary calculations show that the design and application of the model agree on the pre-requirements, thereby laying a good foundation for the sustainable development of the numerical prediction system of the country. Several approaches in GRAPES model: standardization, module and multi-scale community. According to the principles of software engineering, the system and its parallel computing are developed. Results of experiments show that both the design of model framework and the implementation of GRAPES model can meet the requirements of model processing. It's asserted that this framework plays an important role in GRAPES development.

Key words: GRAPES Numerical forecast Standardize design Multi-scale community
Parallel computing