

基于 ESMF 的 POP 海洋模式组件化设计与并行方法研究^①

叶 斯^{②*} 栾钟治^{③*} 谭 超* 王兰宁** 钱德沛*

(* 北京航空航天大学中德软件技术联合研究所 北京 100191)

(** 北京师范大学全球变化与地球系统科学研究院 北京 100875)

摘 要 针对目前海洋模式开发存在的移植、更新和相同功能代码重用困难的问题,对海洋模式的功能分解及组件化封装进行了研究,设计了一套基于地球系统建模框架(ESMF)的组件化与并行化方法,并将其应用在并行海洋模式 POP 上,构建了一种新的组件化海洋模式。对新模式的测试结果表明,上述方法能在保证模式正确性的基础上实现模式的复用性与可维护性,并且能在一定程度上提高模式的运行效率。

关键词 并行海洋模式(POP),地球系统建模框架(ESMF),海洋模式,组件化

0 引 言

近年来,随着气候模式研究的不断深入,模式的规模日益庞大。采用紧耦合方式设计的模式由于功能之间的调用关系复杂,数据在全局范围内共享,要进行模式的更新,往往需要对许多相关的函数进行修改,这极大地增加了模式的维护与更新负担。同时,随着高性能计算需求的增加,模式所依赖的软硬件环境也越来越复杂。在这样的情况下,气候科学家往往需要花费大量精力来了解整个模式的并行架构以及高性能计算机的软硬件细节,并对底层代码做大量的工作,才能开发出符合模式架构的物理模型,这很大程度上影响了模式的创新与发展。特别是当模式研究涉及到更多学科的交叉与更多研究机构的合作时,这种情况会变得更加明显。因此,如何减少气候科学家在模式开发上的学习负担,提高用户代码的模式代码的可重用性、可替换性和并行性能,是急需解决的问题。

目前上述问题的解决方法是通过统一的接口与底层架构,实现标准的组件化模式,使用户可以专注于自身领域的研究,快速将研究成果以组件的方式实现。但是由于模式的发展先于模式框架研究,要将整个模式研究成果转向组件化需要花费大量的时

间与精力,当前研究重点主要集中在分量模式之间的组件化与耦合设计上,通过统一的接口来耦合各个气候领域的研究成果,从而把模式更新与维护的工作分散到各个模式开发小组当中。但分量模式组件的规模仍然非常庞大,并且组件内部有关并行的实现也没有遵循统一的底层标准,这极大地阻碍了不同模式间的相同功能代码重用,气候科学家要实现自己所关注的物理模型,仍然需要深入了解分量模式的设计架构。因此,本文尝试将组件化的构建方式深入到模式内部,将单个模式构建成标准组件的集合。本研究以当前广泛使用的并行海洋模式 POP(parallel ocean program)^[1,2] 为对象,借助地球系统建模框架(Earth System Modeling Framework, ESMF)^[3] 的标准接口及并行框架,构建了一种标准的组件化海洋模式,基于此,设计了一套针对海洋模式的组件化和并行化方法,并对其进行了测试与分析。POP 是一种典型的使用紧耦合的开发方法实现的并行海洋模式,是目前使用最广泛的公用地球系统模式(community earth system model, CESM)^[4] 的组成部分,具有很好的用户基础以及社区活跃度。通过对 POP 进行组件化封装与设计,可提高 POP 代码的可重用性、可移植性和并行性能,能有效促进海洋模式的研究发展。

① 863 计划(2010AA012404)和科技部国际合作(2009DFA12110)资助项目。

② 男,1986 年生,硕士;研究方向:高性能计算;E-mail: yesshi06@yahoo.com.cn

③ 通讯作者,E-mail: rick710055@263.net

(收稿日期:2011-09-27)

1 相关工作

早在 20 世纪 80 年代,气候科学家就意识到了统一的接口与框架对模式研究的巨大推动作用,并因此产生了许多相关的气候框架并得到了应用。其中使用比较广泛的有灵活建模系统(flexible modeling system, FMS)^[5]、天气研究与预报模型(weather research and forecast model, WRFM)^[6]、地球建模系统(goddard earth modeling system, GEMS)^[7]、通用组件架构(common component architecture, CCA)^[8]、模型耦合工具包(model coupling toolkit, MCT)^[9]等。这些框架或是针对一类应用提供解决方案,如 WRFM;或是专注于模式某一方面的需求,如用于系统建模的 FMS 以及 GEMS,用于模式耦合的 MCT;或是面向更一般化的高性能应用,如 CCA。与这些早期的气候框架相比,地球系统建模框架(ESMF)是一个更全面更规范化的框架。它统一了各种已有的专用软件框架和工具包,并对其进行标准化和扩展,形成了一组标准的组件接口和完整的气候开发并行框架。相比于同类型的欧洲委员会资助的 PRISM^[10]框架,ESMF 更关注组件构造的基础结构及用于耦合组件的标准接口,能够很好地满足本文设计组件化模式的需求。

随着气候框架的出现,许多模式也开始朝着组件化的方向发展。美国国家地球物理流体动力学实验室(Geophysical Fluid Dynamics Laboratory, GFDL)开发的模块化海洋模式(modular ocean model, MOM)^[11]就是一个典型的例子,由于模式中的所有分量模式均由 GFDL 自身开发,因此各分量模式的底层接口基本统一。同时 MOM 这种模块化结构的优势在于可以在不改变原模式的并行计算以及数据通信代码的基础上,根据新的计算环境修改模式内部过程,补充新的程序。但是, MOM 并没有提供一个统一的软件底层架构,因此面对不同的计算环境,用户仍然需要进行底层代码的编写和维护工作。随着模式的发展,模式的组件化应用也越来越广泛。公用地球系统模式 CESM 就为其下的每个分量模式设计了 ESMF 的标准调用接口,使每个分量模式都构成标准的组件,以提高分量模式的耦合能力。此外,美国空军气象局的太空天气建模系统(space weather modeling system, SWMS)将其中的太阳风模式 HAFv2 与电离层预测模式 IFM 封装成模式组件进行耦合,显著改善了模式的数据吞吐率以及计算

效率^[12]。美国海军实验室的海洋大气耦合中尺度预测系统(coupled ocean/atmosphere mesoscale prediction system, COAMPS)通过组件化技术来解决海洋大气之间的耦合问题^[13]。美国国防部的战场环境研究所(Battlespace Environment Institute, BEI)也开始进行各分量模式间的组件化耦合^[14]。在海洋模式方面, Campin 等实现了一个多尺度模拟的高分辨率 2 维非静力过程模式,并将其封装成一个单独的组件耦合到大规模的流体静力海洋模式中^[15]。这些模式虽然都使用了组件化的方式来进行各个分量模式的耦合,但并没有将组件化设计深入到模式内部,使得组件仍然具有规模庞大,内部代码难以复用和移植的缺点。目前真正在模式内部使用标准框架来进行组件化封装的只有最新的地球观测系统模式(Goddard Earth Observing System Model, version 5, GEOS-5)^[16]。GEOS-5 是当前分辨率最高的气候模式系统,它集成了美国国家航天局(NASA)、美国国防部、美国国家大气与海洋管理局以及各大机构与高校的气候研究成果。由于系统结构复杂,分量模式众多,因此 GEOS-5 从设计之初就采用了组件化的技术,将 30 多个不同的遵循 ESMF 规范的组件耦合在一起形成一个完整的系统,该模式不仅在上层采用了统一的耦合接口,内部也实现了分层次的组件化,是第一种实现了内部组件化的模式。但是系统的内部组件化主要实现在大气环流分量模式中,其他分量模式仍然是以整体组件的方式耦合在系统当中。

目前,大部分的气候模式的组件化设计与研究都基本围绕模式间的耦合开展,将组件结构深入到参数化水平,特别是实现针对海洋模式的这种组件化,几乎没有相关工作。因此,本文以 POP 为对象,借助 ESMF 框架的标准接口以及并行框架构建了一种组件化的海洋模式,将组件结构深入到海洋模式的参数化水平,并通过模式组件特性的分析,提出了一套适用于组件化海洋模式的并行化方法。

2 ESMF

2.1 ESMF 总体架构

ESMF 架构分为两层,上层定义耦合接口,底层定义基础架构类,通过这两层架构共同封装与具体学科相关的代码,形成标准化的组件。具体来说,ESMF 的上层架构构建于模式科学代码之上,包含程序驱动以及组件类,负责组件的执行以及组件间

的数据通信;底层架构位于组件内部,包含了代表数值网格和物理区域的数据结构,此外还包括优化的、便于使用的工具集,负责组件内部并行通信、错误处理、时间管理等功能。上层架构以及底层架构可以结合使用也可以单独使用,如只使用上层架构来耦合不同实现的标准组件,也可以只使用底层架构提供的数据结构以及优化工具类构造科学应用。本文在设计时将引用其上层架构作为海洋模式组件的接口标准,同时基于其底层架构设计模式的并行逻辑。

2.2 ESMF 上层架构

ESMF 的上层架构是组件的交互基础。它通过统一的接口来耦合模式的各个功能组件。上层架构将模式组件分成网格组件与耦合器组件两类,网格组件负责模式具体功能或物理过程的实现,模式具体的逻辑都封装在网格组件当中;耦合器组件负责网格组件间的数据交互,包括组件数据传输、组件间的网格插值等。通过这两类组件的结合,共同构成完整的气候模式应用。

2.3 ESMF 底层资源管理

ESMF 底层架构负责定义标准的数据类型和管理模式的计算资源。它把资源的集合抽象为一个虚拟机(virtual machine, VM)类。VM 类屏蔽了计算机硬件以及操作系统的差异,把所有的资源都统一封装为持久执行线程(persistent execution threads, PET)。VM 类内保存所有 PET 的拓扑描述,在进行 PET 间的通信时,对通信通道进行筛选,以提供最好的通信性能。PET 是 ESMF 中资源单位,根据分配的处理单元(processing element, PE)以及配置的不同既可以代表进程也可以代表线程。具体的逻辑结构如图 1 所示。

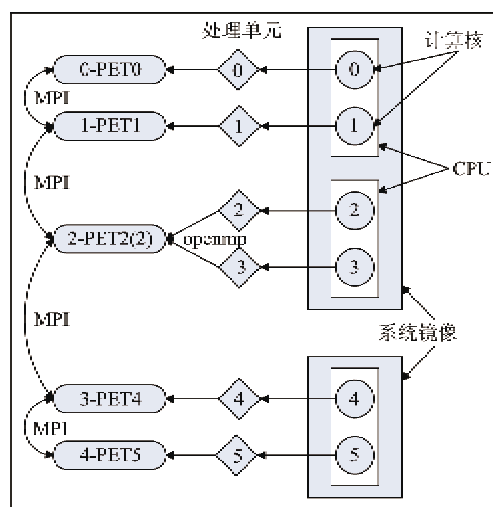


图 1 VM 逻辑结构图

从图 1 中可见,VM 下的计算资源首先形成操作系统实例镜像,然后实例镜像内的每一个核都是一个单独的处理单元。处理单元按照需求分配到具体的 PET 内。其中 PET2 被分配了 2 个处理单元,则该 PET 可以通过 openmp 等并行语言来实现共享内存式的并行。同时,各个 PET 间可以通过多点接口(multipoint interface, MPI)进行通信,形成混合并行的结构。通过这种设计,ESMF 的底层资源框架能有效地屏蔽计算机软硬件上的差异,提供统一的计算资源以及良好的通信性能。因此本文在设计上使用了 ESMF 的底层框架来对计算资源进行管理。

3 POP 海洋模式的组件化设计与并行实现

有别于传统的模块紧耦合设计的模式,本文借助 ESMF 来对 POP 海洋模式进行重新设计。新的模式采用分层拓扑的构造方式,将整个程序的功能进行分解,形成功能子组件的集合,同时模式采用自顶向下的方式来分配计算资源,使每个功能子组件都形成独立的并行计算区域,最终实现一种新的组件化海洋模式。下面分组件化的模式设计、模式的并行实现、组件内的混合并行设计三个方面来介绍模式的实现。

3.1 POP 海洋模式的组件化构建

POP 是一种并行化的海洋数值模拟模式,它的计算可以使用三维浅水波的原始动力学方程组来描述^[2]。模式方程在时间离散时,分为 2 维正压外模态(barotropic)以及 3 维斜压的内模态(baroclinic)。在进行并行计算时,POP 把整个网格划分为多个子网格,每个进程计算属于自己的子网格区域,每计算一步就进行并行通信,根据其他进程计算结果更新自身信息。其中 2 维正压求解一个表面压力的椭圆方程,并行通信包括边界通信以及全局归约,主要的通信开销为全局归约,通信比较密集。3 维斜压使用显示差分离散来求解斜压速率与轨迹。模块计算密集,主要通信开销为边界通信。除了计算部分,POP 还有与大气、海冰等外部状态交互的强迫场部分,通过这种外部交互来更新自身状态。原 POP 的基本结构如图 2 所示,整体形成顺序结构,各个函数与数据在整个模式内共享,程序之间耦合非常紧密。

为了实现组件化海洋模式设计,本文将整个模式进行分解,形成结构上独立的组件。整个模式分解后形成如图 3 所示的分层拓扑结构。与原 POP 相比,组件内的数据与函数不再在整个模式内共享。

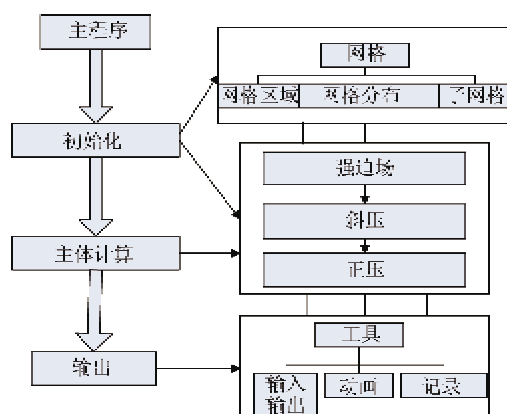


图 2 原 POP 结构图

所有组件所需的输入数据都集中封装在标准的输入数据类中,通过耦合器传入。模式的组件设计遵循 ESMF 的接口规范,但在组件结构上做了进一步的扩展,设计了一个 5 层的组件概念模型,通过统一组件的上层接口以及底层数据结构,保证中间层用户代码的功能实现。5 层组件模式从上到下分别为:接口层、模式层、物理区域网格层、底层并行以及第三方类库。如图 4 所示。

■ 接口层是组件的对外接口,由组件类、输入数据类、输出数据类以及时间类组成。其中组件类是当前组件的实例,包含了模式计算的网格信息以及

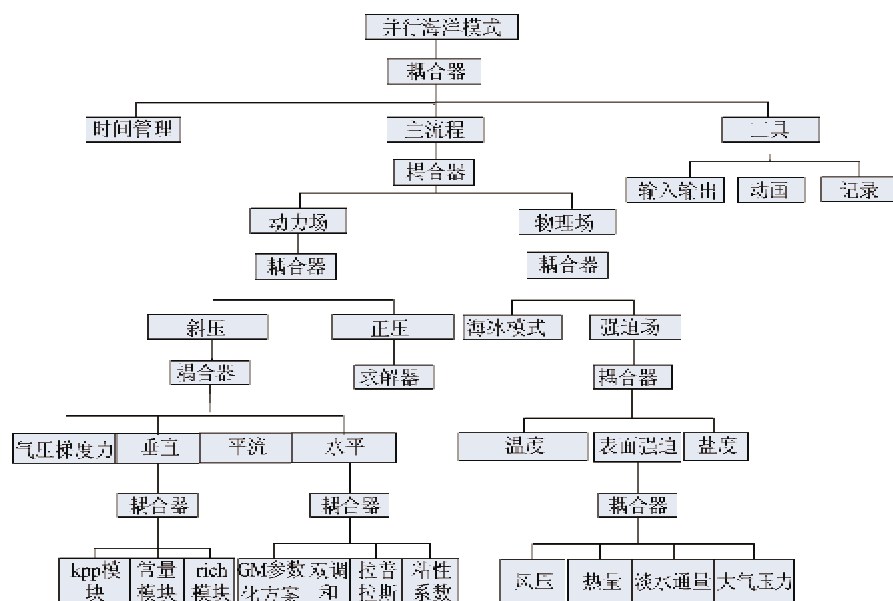


图 3 组件化海洋模式结构图

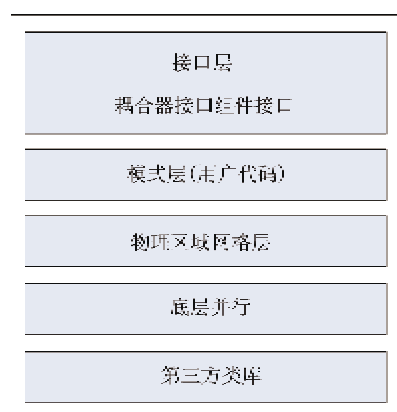


图 4 5 层组件概念模型

■ 模式层是组件内具体的计算逻辑,其中既包括气候模式的科学代码也包括分层拓扑中的下层组件的调用逻辑。POP 海洋模式的代码就封装在组件的模式层内。

■ 物理区域网格层主要由网格相关的物理量组成,如表面大气压力、淡水通量等。在程序中这些物理量的数据往往以多维数组的形式保存。本文对这些物理量进行了进一步的封装,将这些物理量数据与具体的 ESMF 网格相关联,使其带有具体的网格剖分信息,既可以为模式层的计算提供底层数据结构支持,也可以直接调用 ESMF 中提供的并行算法。

■ 组件底层由 ESMF 标准的虚拟机 (VM) 类构成,VM 屏蔽了计算机软硬件上的差异,向组件提供统一的计算资源 PET,并与物理区域网格层结合共同为组件提供并行支持。

■ 最后一层则是提供应用支持的网络通用数据

当前组件拥有的计算资源。输入与输出数据类则包含了整个组件输入输出数据。组件类以及输入输出数据类都由分层拓扑结构中的上层组件创建,并在调用下层组件时分配合适的资源以及数据。

格式(network common data form, NetCDF) 等第三方类库。

在组件化海洋模式运行时, POP 组件是整个模式的入口, 负责创建整个模式的网格, 包括网格属性设置(如全球网格中经度方向采用循环边界、纬度方向采用开边界或闭边界), 根据输入参数剖分网格并向每个进程分配计算任务。除了入口组件 POP 以外, 其他所有组件都需要向上层组件提供具体的方法调用。具体调用方法设计如表 1 所示。

表 1 组件可供调用方法表

组件	方法	方法描述
主流程	step_run	执行单个时间步的运算
物理场	physics_run	获取模式的强迫输入
动力场	dynamic_run	根据模式强迫输入执行模式计算
斜压	driver	计算斜压内模态的速度与轨迹
斜压	correct_adjust	调整轨迹
正压	driver	计算正压外模态的表面压力与速度
海冰	formation	计算冰形成
强迫场	forcing_get	获取模式表面和内部强迫数据
气压梯度力_grad	gradp	计算气压梯度力
水平	ceoffs	选择参数化方法计算混合系数
水平	vdifft	计算示踪物的垂直扩散
水平	vdiffr	计算动量的垂直扩散
水平	impvmixt	计算示踪物的隐式垂直混合
水平	correct	进行隐式垂直混合的校正
水平	impvmixr	计算动量的隐式垂直混合
平流	advu	计算平流动量
平流	advr	计算示踪物水平对流
垂直	hdiffu	计算动量的水平扩散
垂直	hdiffr	计算示踪物水平扩散
温度	get_pt	获取海洋的位温
表面强迫	get_surface	获取海洋表面强迫
盐度	get_salt	获取海洋的盐度
Kpp 模块	vmix_kpp	使用 kpp 混合方案计算垂直混合系数
常量模块	vmix_const	使用固定值作为垂直混合系数
Rich 模块	vmix_rich	使用 Richardson 参数化方案计算垂直混合系数
Gm 方案	hmix_gm	使用 GentMcWilliams 参数化方案计算水平混合

(续)

双调和	hdiffu_del4	计算动量的双调和水平扩散
双调和	hdiffr_del4	计算示踪物的双调和水平扩散
拉普拉斯	hdiffu_del2	计算动量的拉普拉斯水平扩散
拉普拉斯	hdiffr_del2	计算示踪物的拉普拉斯水平扩散
粘性系数	hmix_aniso	计算动量方程中的粘性系数
风压	get_wind	获取海洋表面风压值
热量	get_heat	获取海洋表面热通量
淡水通量	get_flux	获取海洋表面淡水通量
大气压力	get_pressure	获取海洋表面大气压力

分层拓扑中、下层组件的方法只允许上层组件调用, 数据的使用与传输也只能通过上层组件来进行。如 wind 组件中的海洋表面风压值要传输到斜压组件, 中间就需要经过表面强迫、强迫场、物理场、主流程、动力场个组件(如图 3 所示), 这需要一些额外的传输时间, 但由此可以带来组件的独立性以及整个模式的高可扩展性。同时由于每一层组件的设计实现只与其上层组件相关, 因此要进行组件更新和扩展时, 只需了解上一层组件的调用细节(接口数、输出输出数据)就可以按照标准接口设计新的组件加入到模式当中。

本文的组件化方法主要改变的是程序的组织结构, 使每个组件成为独立的功能模块。只要模式的运行平台支持 ESMF, 就能够不修改源代码而直接进行模式的部署和运行。这在一定程度上提高了模式系统的可移植性。

同时, 科学家在使用组件化模式开展研究的过程中, 只需要专注于具体组件内部的物理过程和机理, 不需要理解如并行架构和高性能计算机的软硬件结构等模式系统的底层技术细节。这使气候科学家可以把精力从模式程序的开发转移到模式本身的发展上来。

本节介绍了组件化海洋模式单个组件的设计模型以及整个模式总体结构, 下一节将基于本节的内容来介绍组件的并行实现。

3.2 组件化海洋模式的并行方法设计与实现

当前主流气候模式的并行方案都是对网格进行剖分, 每个进程计算网格的一部分, 最终获得整个网格的计算结果, POP 海洋模式也遵循这一设计思路。本文在对 POP 进行组件化设计时, 将这种并行方案

进行了细化,把并行设计从整个模式细化到单个组件当中。

由于并行设计的细化,每个组件都有属于自己的 VM 类,原来基于 MPI 实现的针对整个模式的并行代码已经不再适用于单个组件。针对这种情况,组件内的并行设计通常可以有两种解决方案:通过物理区域类进行以及通过网格数组类进行。例如,在原 POP 模式中,边界交换的通信过程通过 MPI 分别接收其他进程发送的边界信息来完成,而在组件化海洋模式中,要执行边界交换的数据只需要调用 ArrayHalo 或者 FiledHalo 方法就可以实现整个通信过程。一般来说,模式并行所需要的方法大部分能通过调用类库来实现。而对于特殊的并行逻辑,组件还可以通过 VM 类的并行调用来实现,这实际上是一种类似 MPI 的调用接口,可以进行进程间的点对点的通信。如正压组件中的全局归约由于没有相关的接口设计,本文根据原 POP 的并行逻辑,用 VM 类代替了原程序中的 MPI 接口调用,重新实现了全局归约的逻辑。由于 VM 类屏蔽了底层的软硬件细节,提供了一种统一的并行资源 PET,因此,使用 VM 类实现的并行逻辑也具有非常高的可移植性。

此外,这种分层拓扑的模式组件架构也提供了一种灵活的组件资源分配方式。由于组件化后的海洋模式形成了自顶向下的分层拓扑结构,同时并行资源也会在初始化时自顶向下分配到各个组件中,从而形成每个组件自身的并行通信域。通过这种资源分配方式,模式可以很方便地调配每个组件的并行数,同时并行从整个模式细化到组件也降低了并行设计的难度。当然这需要以增加网格同步的时间开销为代价。具体如图 5 所示。

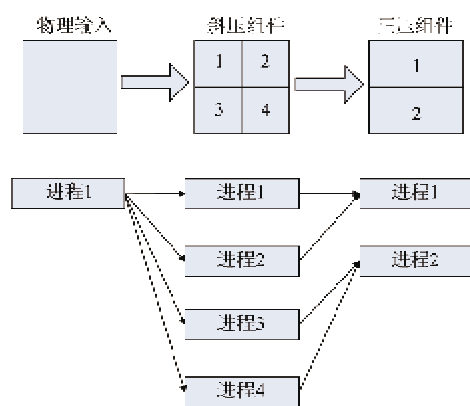


图 5 基于网格划分的 4 进程模式图

图 5 中分配了 1 个进程给物理场组件获取整个网格的强迫输入。当物理场组件计算完毕后,获得

的数据会在耦合器传输数据时同步到所有进程当中。这样,当程序执行到斜压计算时,每个进程都会有一份物理场的输入数据。当 4 个斜压进程计算完各自部分之后,在耦合器传输时需要进行汇总同步,再分配到 2 个进程正压计算中,以此保证组件的网格数据一致。

由于不同并行度的组件需要进行额外的网格同步,因此在进行组件设计时需要注意并行度的改变所带来的影响。本文组件化海洋模式的并行主要集中在动力场组件,包括计算密集的斜压组件以及通信密集的正压组件。其中正压组件是通信密集组件,通信开销以全局归约为主,需要大量的小数据块的通信,因此在并行增加到一定规模后反而会带来性能的下降。研究表明,当正压所计算的网格局部块小于 80×96 时,正压的计算量就已经不足以屏蔽并行通信带来的消息延迟^[17]。因此,针对不同的组件情况,分配正压与斜压不同的计算资源,在高并行环境下,能很好地减少整个模式由于通信而造成的性能损失。本文具体的资源分配设计分为 2 步:首先在正压与斜压组件创建时通过组件的 VM 类来指定组件的并行域,然后在耦合器中创建对应的组件间网格同步调用,包括从斜压到正压以及正压到斜压的数据同步。在具体运行时,斜压组件先通过 VM 类获得整个动力场的计算进程执行计算。计算完成后,数据进入耦合器,通过网格同步方法分配到正压所使用的进程中,正压使用部分进程执行计算。计算结果数据再通过耦合器同步到动力场的所有进程中,完成整个分配过程。

与原 POP 模式基于 MPI 的代码相比,本文的并行实现设计在每个组件的内部,从而保证每个组件的并行独立性。与原模式类似,用户仍然可以根据模式的划分确定并行进程数,每个组件都运行在该并行进程集内,并可随着并行集的扩展进行调整。因此,对模式进行组件化并不会影响模式的并行扩展性。

3.3 组件的混合并行设计

网格划分方式对海洋模式的运算性能有着显著影响。小的网格划分能有效剔除陡地点,减少计算负载,但同时过小的网格也会造成每个进程的計算量的下降和并行通信的增加。POP 对此采用了一种局部网格技术,由一个进程负责多个子网格的计算,细化到本文的组件化海洋模式中,就是每个进程内的组件执行时计算多个子网格。此时,在组件内对每个子网格的计算实行线程级并行能有效提高组件的计算效率。首先在组件初始化时定义每个组件进

程所使用的处理单元数(如图 1 所示),然后再通过 openmp 等并行语言来设计组件内并行计算。本文在斜压组件上使用了 openmp 并行,线程并行区主要包括轨迹更新以及动量方程计算两个子网格的计算循环。但需要注意的是,与一般的 openmp 设计相比,在并行区域内调用子组件运行时,由于子组件需要一个统一的数据结构来封装所有的输入,因此该结构内既包含了各线程的私有数据也包含了各线程共享数据,在多线程运行时会产生数据冲突。为了解决此问题,本文为每个线程中的子组件构造了一个数据结构,如图 6 所示。

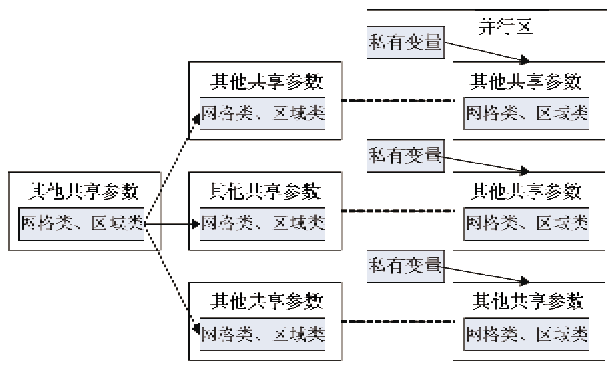


图 6 并行数据结构处理图

具体处理过程如下:在进入并行区之前,先构造只包含共享数据的数据结构,然后以该结构为对象,用浅拷贝的方式为每个线程构造一个相同的副本,使各个副本内的网格数组类以及物理区域类实际都指向原数据结构下的内容,保证每个线程对网格数组类以及物理区域类的共享;进入并行区之前从原数据结构中抽出的私有数据,在进入并行区后以纯拷贝方式添加到前述构造的副本中。计算完毕后,再将线程计算的数据汇总,销毁拷贝的副本。通过这种处理可以保证组件内的线程级并行的数据一致性。此外,组件采用动态线程分配,允许多个线程并发请求并行区的子网格计算任务,直到计算完毕,这种方式能实现组件内线程的有效利用,提高组件计算效率。

4 组件化海洋模式的测试与分析

为了测试组件化海洋模式的正确性与性能,准备了以下的实验环境:16 个节点的计算集群,使用 PBS 作为作业管理器。每个节点的配置如下:4 核 Intel(R) Xeon(R) CPU 2.40GHz × 2, 24G 内存, 292G SAS 硬盘, RedHat 5.4 操作系统, 节点开启了

超线程,有 8 个物理核与 8 个虚拟核,可以同时运行 16 个进程。本次测试一共分为 3 组,分别测试组件化海洋模式的正确性和并行性能、组件资源分配以及组件混合并行的效果。测试使用的模式输入为 brenchX1 320 × 384 的网格,POP 源程序版本为 2.0.1, MPI 实现使用 openmpi-1.4.1, ESMF 版本为 ESMF_5.2.0, Fortran 编译器使用 ifort-10.1.022, 编译优化开关统一为 -O2。

第一组实验的目的是测试海洋模式实现组件化后的正确性和程序性能。测试样例一共模拟 30 天,每天 25 步。在该组测试下,正压与斜压采用相同的资源数,模式统一采用 20 × 24 的区域剖分,形成 256 个子网格。正确性验证通过原 POP 模式和组件化海洋模式的诊断输出文件对比得出结果。经过测试,我们得到如表 2 所示的正确性验证结果和如图 7 所示的运行性能对比结果。

表 2 模式测试结果表

并行度	原 POP 运算时间(s)	组件模式运算时间(s)	输出文件对比
1	7482	7856	正确
2	4042	4243	正确
4	1981	2083	正确
8	1191	1250	正确
16	1082	1130	正确
32	1238	1242	正确
64	1257	1251	正确
128	1534	1506	正确
256	2582	2453	正确

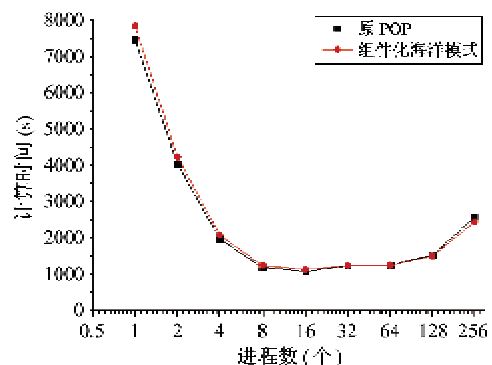


图 7 原 POP 与组件化海洋模式性能比较图

从图 7 可以看出,组件化海洋模式与原 POP 的计算效率非常接近。在并行度大于 64 之后,组件化海洋模式的计算效率超过并略高于原 POP 模式。通过使用 gprof 分析模式运行过程,产生以上结果的

主要原因在于:一方面,组件化海洋模式需要额外的时间来进行组件间的数据传输;另一方面,由于组件化海洋模式在处理组件内并行时采用了调用接口和类库的方式,减少了并行通信量,在并行度高的情况下相比原 POP 模式有较少的并行通信开销。因此,将海洋模式进行组件化并不会降低模式的运行效率,但能带来的模式的高可复用性、高维护性的优点,降低对程序进行 2 次开发的难度。

第二组实验的目的是测试组件化海洋模式的资源分配效果。通过图 7 可以发现 POP 海洋模式在 16 个进程并行开始就处于倒加速状态,计算性能随着节点数的增加不断下降。这说明,从 16 进程并行开始,并行通信的开销已经超过了并行计算带来的效率提升。特别是在 128 到 256 并行时,性能的下降非常迅速。通过对程序的分析可以发现,模式的通信开销很大一部分在正压模块,并且从 Jones 等人的研究^[17]可知,使每个正压进程计算的网格接近 80X96,能很好地保持计算量与通信量的平衡。为此我们在动力场模块中对并行资源进行了选择分配,将正压计算的进程限制在单个节点的 16 个进程以减少通信对性能的影响。测试结果如表 3 和图 8 所示。

表 3 资源分配前后的比较效果

总进程数/正压进程	32/16	64/16	128/16	256/16
分配前的计算时间	1242	1251	1506	2453
分配后的计算时间	1193	1203	1339	1983
加速比	1.04	1.04	1.13	1.24

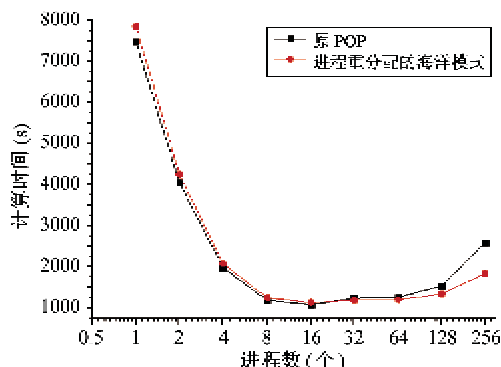


图 8 进程重分配后的模式性能比较图

根据上面的测试结果,我们发现这种分配方式在 32 和 64 并行的时候,效果并不明显。这是由于这种资源分配方式需要付出额外的网格同步开销,而 32 和 64 并行时,正压组件的性能下降并不快。

但是在并行度达到 128 和 256 时,这种分配方式的优势就能体现出来,通过控制并行通信量的增加提高了模式的运行效率。

第三组实验测试组件化海洋模式的混合并行性能。在 MPI 并行的基础上,进一步给计算量最大的斜压组件添加 openmp 并行,测试混合并行对组件化模式的影响。每个进程在并行区分出 2 个线程进行并行计算,具体的混合并行效果如表 4 所示。从表 4 的测试结果可以看出,在斜压组件引入 openmp 并行后,程序在各个并行度的计算效率都有提高。特别是在低并行度时,程序受并行通信的影响较少,效率提高明显。但需要注意的是,随着并行度的增加,通信量在增加,而组件内进程的計算量则是在不断减少,混合并行所带来的效率提升是不断下降的。因此在实际应用时,需要考虑并行规模的影响。

表 4 组件混合并行效率表

并行数 /局部 块数	2/128	4/64	8/32	16/16	32/8	64/4	128/2
并行前	4243	2083	1250	1130	1193	1203	1339
并行后	2831	1485	932	918	1028	1074	1263
加速比	1.5	1.4	1.34	1.23	1.16	1.12	1.06

5 结 论

本文以当前广泛使用的并行海洋模式 POP 为对象,借助 ESMF 的标准接口以及并行框架,构建了一种标准的组件化海洋模式。在此基础上,总结了一套针对海洋模式的组件化与并行化方法,并研究了其在 MPI 并行以及混合并行中的性能表现。测试结果表明,组件化海洋模式很好地保持了原模式的功能,同时基于组件的并行资源分配能在一定程度上解决模式计算不平衡导致的在高并行下性能下降的问题,提高模式运行效率。同时,混合并行的测试结果表明,通过多线程的数据结构构建,在线程并行区内调用子组件计算是有效可行的。总的来说,基于 ESMF 的模式组件化和并行化方法能够在提高气候模式的可移植性和可维护性的基础上,满足气候模式的并行需求。

目前本文的 POP 模式组件已经集成到面向地球系统模式研究的高性能计算支撑软件系统中,为气候模式的应用开发提供了支持。下一步工作将在更多其他类型的模式中应用本文提出的组件化方法,验证该方法在其他类型模式中的效果。

参考文献

- [1] Smith R, Gent P. Reference manual for the Parallel Ocean Program. <http://climate.lanl.gov/Models/POP/>, 2002
- [2] Los Alamos National Laboratory. Parallel Ocean Program (POP) User Guide. <http://climate.lanl.gov/Models/POP/>. 2003
- [3] Hill C, DeLuca C, Balaji V, et al. The architecture of the Earth System Modeling Framework. *Computing in Science and Engineering*, 2004. 6(1):18-28
- [4] Vertenstein M, Middleton A, Feddema D, et al. Community Climate System Model CESM1.0.3 User Guide. <http://www.cesm.ucar.edu/>. National Center of Atmospheric Research, 2010
- [5] Balaji V. FMS: the GFDL Flexible Modeling System. <http://www.gfdl.noaa.gov/fms/>, GFDL, 2005
- [6] Michalakes J G, Chen S, Dudhia J, et al. Development of a next generation regional weather research and forecast model. In: Proceedings of the 9th ECMWF Workshop on Use of High-performance Computing in Meteorology. World Scientific Press, 2001. 65-75
- [7] Rood R B. The GEOS-3 Data Assimilation System Architectural Design. Greenbelt, Maryland, USA: NASA Goddard Space Flight Center, 1997
- [8] Bernholdt D E, Allan B A, Armstrong R, et al. A component architecture for high-performance scientific computing. *International Journal for High Performance Computing Applications, ACTS Collection Special Issue*, 2006, 20(2):163-202
- [9] Larson J, Jacob R, Foster I, et al. The Model Coupling Toolkit. In: Proceedings of the International Conference on Computational Sciences, Heidelberg, Germany, 2001. 185-194
- [10] Guilyardi E, Budich R, Brasseur G, et al. PRISM System Specifications Handbook-Version 1.0. <http://prism.enes.org/Publications/index.php>, 2005
- [11] Stephen M Griffies, Matthew J Harrison, Ronald C Pacanowski, et al. A Technical Guide to MOM4. http://gcmd.nasa.gov/records/NOMADS_MOM4_BETA_MODEL.html, 2009
- [12] Reich J P, Fry C D, Eccles J V, et al. The space weather modeling system: an ESMF compliant solar wind and ionospheric forecast system. In: Proceedings of the American Geophysical Union, Fall Meeting, San Francisco, USA, 2008. 258-263
- [13] Daniel A Geisler, SAIC, San Diego, et al. Transitioning coupled atmospheric and ocean prediction using the Earth System Modeling Framework (ESMF) into operations. In: Proceedings of the Joint between the 9th Conference on Coastal Atmospheric and Oceanic Prediction and Processes and the 17th Conference on Air Sea Interaction, Annapolis, USA, 2010. 166-172
- [14] Cheng H, Cheng J C, Hunter R M, et al. Two-way coupled watershed-nearshore modeling using DBuilder and ESMF. In: Proceedings of the American Geophysical Union, Fall Meeting, San Francisco, USA, 2008. 131-143
- [15] Campin J, Hill C, Marshall J, et al. Superparameterization in Ocean Modeling Using General Multiscale Techniques: A Deep- Convection Case Study Employing ESMF. In: Proceedings of the American Geophysical Union, Fall Meeting, San Francisco, USA, 2008. 233-238
- [16] da Silva A, Suarez M J, Trayanov A, et al. Use of ESMF in the GEOS-5 modeling and assimilation system. In: Proceedings of the American Geophysical Union, Fall Meeting, San Francisco, USA, 2008. 161-169
- [17] Jones P W, Worley P H, Yoshida Y, et al. Practical performance portability in the Parallel Ocean Program (POP). *Concurrency and Computation: Practice and Experience*, 2005, 7(10):1317-1327

Research on componentized design and parallelization towards POP ocean model using ESMF

Ye Si^{*}, Luan Zhongzhi^{*}, Tan Chao^{*}, Wang Lanning^{**}, Qian Depei^{*}

(^{*} Sino-German Joint Software Institute, Beihang University, Beijing 100191)

(^{**} College of Global Change and Earth System Science, Beijing Normal University, Beijing 100875)

Abstract

In order to address the problem that it is difficult to port and utilize existing abundant reusable modules during the development of a new model due to the tight coupling among different modules in a contemporary ocean model, the authors decomposed the parallel ocean model of parallel ocean program (POP) into a set of scientific procedures, and then encapsulated the implementation codes into reusable components with the Earth System Modeling Framework (ESMF). In addition, the ESMF parallelism was applied to the new component-based POP, which can help to further enhance the efficiency. The experiment results showed that the proposed approach, on the premise of correctness, improved the reusability and maintainability of the model significantly, as well as its performance.

Key words: parallel ocean program (POP), Earth System Modeling Framework (ESMF), ocean model, componentized