

地球系统模式耦合器

C-Coupler2

清华大学地球系统科学系
地球系统数值模拟教育部重点实验室



目录

简介	1
主要功能特点	2
与其他耦合器的功能特点对比	4
软件测试	6
版权说明	8
源码下载与技术支持	8
未来工作	9

简介

C-Coupler (Community Coupler) 是一个由我国独立自主发展的地球系统模式耦合器系列，其研发始于 2010 年，目标是为各种耦合模式的系统集成提供灵活、易用且功能丰富的耦合器。为了达成这一目标，第二个版本即 C-Coupler2 不仅继承了上一版本即 C-Coupler1 在耦合架构（如图 1 所示）与静态三维耦合方面所取得的创新性，还具备了一系列新功能特点，主要包括通用且灵活易用的耦合配置接口、对同一可执行文件内或同一分量模式内耦合的支持、灵活的自动耦合生成功能、动态三维耦合功能、非阻塞数据传输、对便捷实现增量耦合和模式嵌套的支持、对耦合模式进行调试的支持、以及自适应重启动功能。

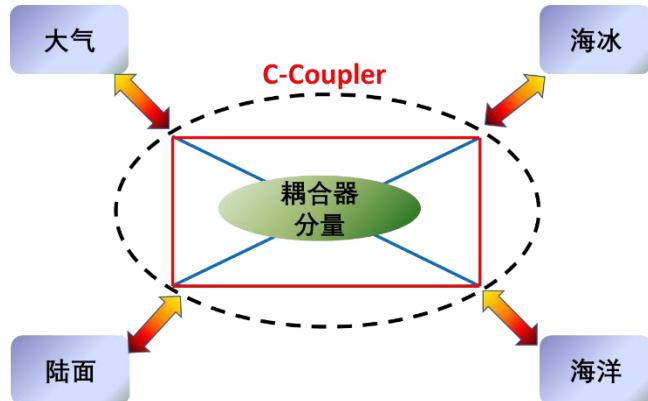


图 1 C-Coupler 系列的耦合架构。该架构的主要含义是：分量模式程序只需告诉 C-Coupler 其能提供的耦合变量和所需的耦合变量，而无需指定所需耦合变量的具体来源。这一架构将使得（1）C-Coupler2 具有通用性，能用于各种耦合模式的构建；（2）一个分量模式可在不同耦合模式中保持相同的代码版本。

C-Coupler2 的软件结构如图 2 所示，其由耦合配置接口、耦合生成器和耦合功能模块三部分组成。C-Coupler2 的源代码总量约为 40000 行，全部由清华大学 C-Coupler 团队自主研发，采用的编程语言为 C++ 和 Fortran，其中 C++ 程序约 36700 行，主要用于实现各种耦合功能，而 Fortran 程序约 3300 行，用于实现面向模式的应用程序接口。C-Coupler2 属于软件库，其不单独占用处理器资源，模式通过其 Fortran 应用程序接口及 XML 格式的配置文件来使用各种耦合器功能。C-Coupler2 具有近 80 个 Fortran 应用程序接口，以提高设置耦合配置的灵活性。C-Coupler2 是一个并行耦合器，采用了 MPI (Message Passing Interface) 来实现并行化，不仅能支持分量模式之间的并行耦合过程，还保证了在不同并行度设置下耦合过程的结果完全不变。

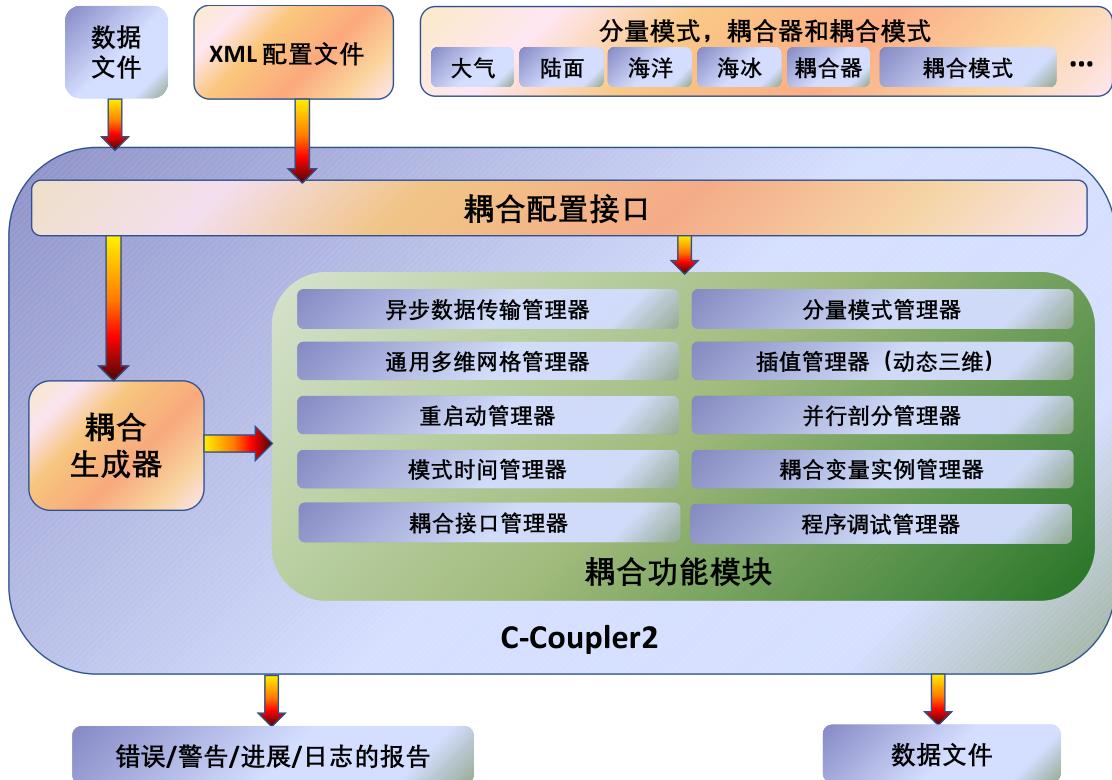


图 2 C-Coupler2 的软件结构，由耦合配置接口、耦合生成器和耦合功能模块三部分组成。

主要功能特点

与 C-Coupler1 甚至国际上其他耦合器相比，C-Coupler2 具有以下新功能：

- 1) 通用、灵活、易用的耦合配置接口，其有机结合了一系列应用程序接口和一系列采用 XML 格式的配置文件。其使用户可以灵活方便地指定或修改耦合配置，包括耦合模式中的分量模式、每个分量模式的时间步长、模式网格、网格上的并行剖分、耦合频率、模式之间的耦合延迟、耦合变量及其数据类型、耦合连接关系、耦合生成、重启设置等。插值配置也是可灵活修改：既可使用由其他插值软件（如 SCRIP, ESMF, YAC, CoR 等）生成的插值权重文件，也可使用由 C-Coupler2 自动并行生成的插值权重。
- 2) 对同一可执行文件内或同一分量模式内耦合的支持。相互耦合的分量模式可在相同或不同的可执行文件中，并且可以完全或部分共享一组 MPI 进程。同一分量模式中的不同过程（如动力过程与物理过程）也可以通过 C-Coupler2 来实现耦合。
- 3) 灵活的自动耦合生成功能。耦合生成器可以自动检测到耦合模式中的分量模式，并在给定的分量模式子集中检测潜在的耦合连接，为每个耦合连接自动生成耦合程序流程。耦合程序流程可包含数据传输、数据插值、数据类型转换和数据平均等操作。一个耦合模式可以进行多次耦合生成，每次可对任意一组分量模式进行耦合生成。
- 4) 动态三维耦合功能。这一功能可以便捷实现耦合变量在两个三维网格之间的耦合，其中三维网格的垂直坐标值可以随模式积分动态改变。

- 5) 非阻塞式数据传输。其分别采用了 MPI 单边通信（即 MPI_put 和 MPI_get）和 MPI 双边通信（MPI_Isend 和 MPI_Irecv）而得以实现（用户可以根据实际需要选择实现方式），以尽可能减少潜在的死锁，实现数据传输与模式计算之间的有效重叠，并可支持耦合延迟的灵活设置（对于两个相互耦合的分量模式，它们在进行耦合交换时的模式时间可以不同）。
- 6) 对增量耦合和模式嵌套的支持。对于一个采用任何耦合器构建的已有耦合模式，其可被用作 C-Coupler2 的分量模式，而已有耦合模式的一个分量模式可被用作其子分量模式；在实现一个已有耦合模式与其他模式的耦合时，仅需用 C-Coupler2 实现新增加的模式耦合过程，而无需修改已有模式中原有的模式耦合过程。在无需大量修改模式代码的前提下，C-Coupler2 就能便捷实现一个区域模式（可以是单独分量模式或使用了其他耦合器的耦合模式）到其自己或其他模式中的嵌套，并可实现嵌套系统中不同网格区域之间的进行积分，以提高模式的并行效率。因此，C-Coupler2 既可以直接耦合分量模式，也可以耦合已有耦合模式或自嵌套模式，可与其他耦合器配合使用，以便捷构建更大的耦合模式（如图 3 所示）。
- 7) 对软件调试的支持。C-Coupler2 源码中有超过 1700 个错误诊断，以尽早提醒用户在构建或使用耦合模式过程中的潜在风险，并指导用户修正模式代码和配置文件中的错误。C-Coupler2 的每个应用程序接口都有一个名为“annotation”的字符串输入参数，其能使用户便捷找到错误对应的模式源代码。
- 8) 自适应重启动功能。C-Coupler2 自动保证了在一个耦合连接的耦合延迟大于耦合周期情况下重启动的正确性，即耦合延迟设置的变化不会导致耦合模式无法正确重启动。此外，C-Coupler2 还能为模式自身的非耦合变量的重启动提供服务。

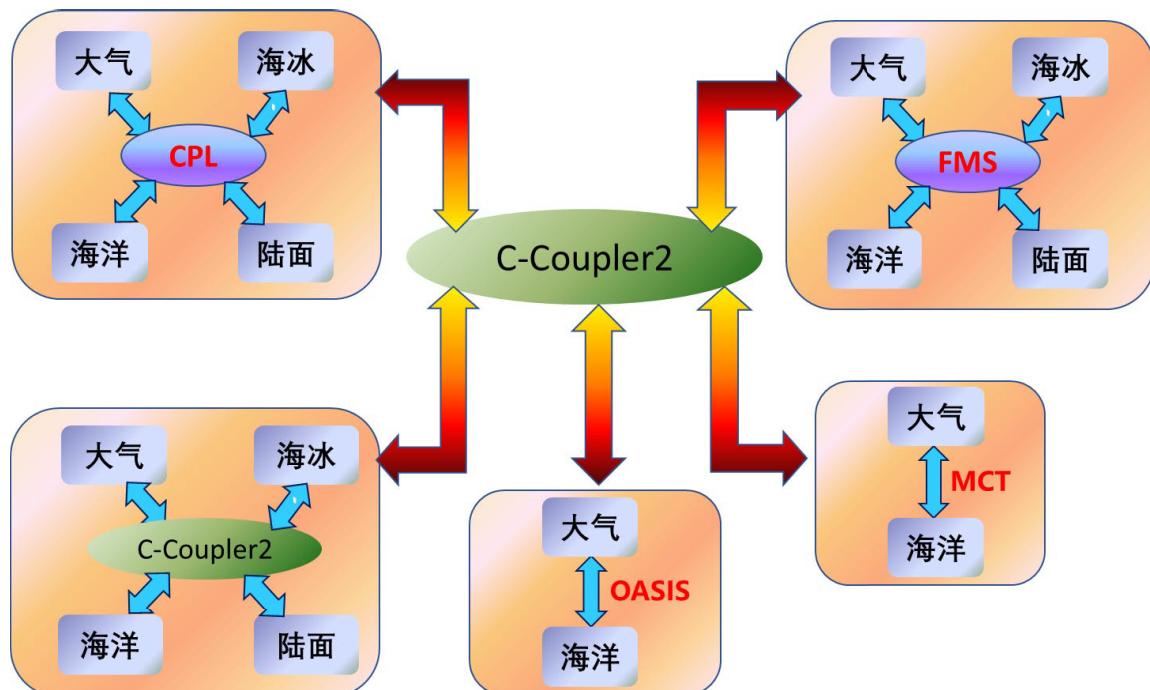


图 3 C-Coupler2 便捷增量耦合架构：既可以直接耦合分量模式，也可以耦合已有耦合模式和自嵌套模式，可与其他耦合器配合使用。

与其他耦合器的功能特点对比

表 1 对 C-Coupler2 与国内外最新的三个耦合器即 CPL7、C-Coupler1 和 OASIS3-MCT_3.0 进行了功能特点对比。表 1 已列出四个耦合器覆盖的主要功能特点。由于数据插值与数据传递是耦合器的两项最本功能，而这四个耦合器都予以支持，因此表 1 未列出。

表 1 G Coupler2、CPL7、C-Coupler1 和 OASIS3-MCT_3.0 的功能特点对比

耦合器功能特点	CPL7 (2010)	C-Coupler1(2014)	OASIS3-MCT_3.0 (2015)	C-Coupler2 (2018)
耦合配置接口	CESM 的专用耦合器，不具备耦合配置接口	完全基于 ASCII 配置文件的耦合配置接口，灵活性很低、通用性较低、可读性不佳	结合了应用程序接口和 ASCII 配置文件的耦合配置接口，灵活性较好，但通用性较低、可读性不佳	有机结合了应用程序接口和 XML 配置文件的耦合配置接口，灵活性好、通用性好、可读性较好
动态三维耦合	二维耦合器，静态三维耦合功能不完善	三维耦合器，仅具有静态三维耦合功能	二维耦合器，静态三维耦合功能不完善	三维耦合器，具有静态和动态三维耦合功能
自动耦合生成	不支持	不支持	仅有唯一的全局自动耦合生成；不能自动发现潜在耦合连接	多次局部或一次全局自动耦合生成；能自动发现潜在耦合连接
增量耦合与模式嵌套	不支持	不支持	除耦合配置接口和自动耦合生成外，无其他支持	除耦合配置接口和自动耦合生成外，能把已有耦合模式（或自嵌套模式）当作分量模式，以便捷实现其与其他模式的耦合；支持同一系统内的多个同类分量模式的便捷耦合集成
同一可执行文件	支持，但依赖于耦合器	不支持（仅支持不同可执行文件）	支持	支持

内或同一分量模式内耦合	合模式 CESM 的驱动程序	行文件间的耦合)	
插值权重的在线生成	不支持	在线串行计算二维和三维插值权重	在线并行计算二维和三维插值权重 (包括垂直插值权重的动态计算)
非阻塞式数据传输	不支持	基于 MPI 双边通信实现的非阻塞式数据传输，在通信过程中存在死锁的风险，影响耦合模式的稳定性	非阻塞式数据传输，提高了设置耦合延迟的灵活性；其中基于 MPI 单边通信的实现方式进一步降低了死锁风险
耦合程序调试功能	不支持	耦合配置信息的正确性检查；各进程独立的日志文件	耦合配置信息的正确性检查；各进程独立的日志文件，及其对模式的开放使用；错误的精确定位功能
并行性能	支持并行数据传输与插值	支持并行数据传输与插值	支持并行数据传输与插值
重启动功能	支持耦合变量的重启	支持耦合变量和模式变量的重启	除支持耦合变量的重启之外，增加了耦合延迟灵活设置的自适应重启支持
通量计算算法集成	基于 CESM 驱动程序嵌入了通量计算算法；因受限于 CESM 驱动程序，可扩展性与灵活性差	提供了耦合集成通量算法的接口，但因要求开发算法 Fortran 程序的 C++ 接口程序而难以使用，灵活性差，且安全性低，无法用于自动耦合生成	耦合配置接口、自动耦合生成功能和对同一分量模式内耦合的支持使得通量计算算法可被当作分量模式使用，这为在后续版本中研制易用、灵活性好且安全性能好的通量计算算法集成与应用接口奠定了坚实基础

软件测试

软件测试对提高 C-Coupler2 的可靠性有极其重要的意义。C-Coupler2 的开源释放版本经过了以下几方面的测试：

- 1) 在进行软件测试时，C-Coupler2 源码中所有错误诊断均处于开启状态。
- 2) 设计了同时含有模式耦合与嵌套的理想模式，并基于其开展了上千项针对耦合配置文件和各应用程序接口及其输入参数的功能单元测试。
- 3) 基于多个应用测试任务（如表 2 所示），开展了上百项具体测试。如表 3 所示，应用测试任务覆盖了 C-Coupler2 的所有主要功能和多种计算机平台。

表 2 C-Coupler2 的应用测试任务

序号	测试任务	任务说明	测试结果
1	基于 CIESM 耦合架构升级的测试	采用 C-Coupler2 来完成 CIESM 中分量模式之间的耦合连接与数据插值，实现耦合架构的升级。	在使用 C-Coupler2 前后，CIESM 的模拟结果完全不变。
2	基于 GAMIL 与 CESM 框架便捷耦合的测试	采用 C-Coupler2 来实现大气环流模式 GAMIL 与地球系统模式 CESM 框架之间的便捷耦合。	便捷耦合版本与原有嵌入式耦合版本（GAMIL 被拆分为若干子程序后被 CESM 驱动程序直接驱动的版本）可保持完全相同的结果。
3	基于 BCC-CSM 耦合器分量并行化升级的测试	采用 C-Coupler2 来替代气候系统模式 BCC-CSM 原有耦合器 CPL5 的插值与通信功能，并实现原有串行耦合器的并行化。	实现了 BCC-CSM 耦合器分量的并行化升级。BCC-CSM 使用 C-Coupler2 前后的结果保持完全不变，耦合器分量采用不同并行度时的运行结果也完全相同。
4	基于 GAMIL2 与 GEOS-Chem 模块化耦合的测试	采用 C-Coupler2 来实现大气环流模式 GAMIL2 与大气化学模式 GEOS-Chem 的模块化三维耦合。	通过检查动态变化的垂直插值权重、对比发送端与接收端的三维耦合变量，确认动态三维耦合结果的正确。
5	基于 MPAS 与 Wavewatch 耦合的测试	采用 C-Coupler2 来实现大气环流模式 MPAS 与海浪模式 Wavewatch 的耦合。	通过对比发送端与接收端的耦合变量，确认模式耦合过程的结果正确。

表 3 表 2 中应用测试任务对 C-Coupler2 主要功能（①灵活、通用且易用的耦合配置接口，②动态三维耦合功能，③自动耦合生成功能，④对增量耦合与模式嵌套的支持，⑤对同一可执行文件内或同一分量模式内耦合的支持，⑥插值权重的在线生成，⑦非阻塞式数据传输，⑧耦合程序调试功能，⑨并行耦合能力，⑩重启功能）的覆盖情况，及测试所用到的计算机环境。

序号	任务名称	C-Coupler2 的主要功能支持										测试用的计算环境			
		①	②	③	④	⑤	⑥	⑦	⑧	⑨	⑩	处理器	操作系统	编译器	MPI 库
1	基于 CIESM 耦合架构升级的测试	√	-	√	-	√	-	√	√	√	√	Intel X86; SW26010;	Linux	Intel 商用编译器；神威太湖之光专用编译器	Intel MPI; MPICH;
2	基于 GAMIL 与 CESM 框架便捷耦合的测试	√	-	√	√	√	-	√	√	√	√	Intel X86	Linux	Intel 商用编译器	Intel MPI
3	基于 BCC-CSM 耦合器分量并行化升级的测试	√	-	√	-	-	-	√	√	√	√	Intel X86; IBM POWER	Linux; AIX	Intel 商用编译器；IBM 编译器	Intel MPI; IBM MPI
4	基于 GAMIL2 与 GEOS-Chem 模块化耦合的测试	√	√	√	-	-	√	√	√	√	√	Intel X86	Linux	Intel 商用编译器；GNU 开源编译器	Intel MPI; MPICH
5	基于 MPAS 与 Wavewatch 耦合的测试	√	-	√	-	√	-	√	√	√	√	Intel X86	Linux	GNU 开源编译器	MPICH OpenMPI

版权说明

C-Coupler2 的版权为清华大学所有，作者包括刘利、张诚、李锐喆和王斌。C-Coupler2 为公开的开源软件，除任何商业活动（如商业使用、商业销售和商业服务等）外，用户在满足如下规定的情况下，可以免费访问、复制、使用、修改、改进和分发 C-Coupler2 及其任何衍生品和支持文档：

- 1) C-Coupler2 源码中的完整版权说明需要出现在 C-Coupler2 的所有副本、衍生作品（如使用了 C-Coupler2 的耦合模式）和支持文档中（如耦合模式的文档或版权说明等）。
- 2) 在用户对 C-Coupler2 进行修改后，并将修改后版本给予他人时（无论是以源代码形式还是二进制形式），其中需要附带对 C-Coupler2 具体修改的说明。

C-Coupler2 是由清华大学及作者以现状 ("as is") 提供的。清华大学及作者，对于任何因行使本许可协议或者使用本软件所产生的任何直接性、间接性、偶发性、特殊性、惩罚性或任何结果的损害，不承担任何责任。清华大学及作者没有义务但会尽量向用户提供支持、咨询、培训和任何形式的关于该软件的使用、操作和性能方面的协助，没有义务但会尽量为用户提供更新、修订、新版本或问题修复。

源码下载与技术支持

C-Coupler2 源代码的释放采用了 Git 软件来进行版本管理，下载源代码的步骤如下：

- 1) 采用浏览器打开页面 <https://github.com/C-Coupler-Group/c-coupler-lib>。
- 2) 页面打开后，采用如下两种方式之一进行源代码的下载：
 - a) 点击主页上的“release”后，进入下载页面，可下载 C-Coupler2 的所有公开历史版本。请注意，下载到的“zip”或“tar.gz”文件将不包含 C-Coupler2 的 Git 版本管理信息。
 - b) 点击主页上的“clone or download”后，可将获得通过 Git 客户端下载源代码的地址。然后在 Git 客户端，采用相关命令，下载到带有 Git 版本管理信息的源代码。

用户可采用以下方式获得 C-Coupler2 最新版本：

- 1) 采用浏览器打开页面 <https://github.com/C-Coupler-Group/c-coupler-lib> 后，点击主页上的“release”，以查看并下载最新的 C-Coupler2 版本。
- 2) 当采用了 Git 客户端方式下载时，可采用命令“git pull”来获取到最新释放的 C-Coupler2 版本。

我们强烈推荐用户点击页面 <https://github.com/C-Coupler-Group/c-coupler-lib> 上的“watch”，并进行相应操作后完成对 C-Coupler2 这一项目的关注，这样在 C-

Coupler2 有版本更新时，用户将能自动收到邮件通知。

用户在使用 C-Coupler2 过程中，当发现 C-Coupler2 存在潜在 bug 时，建议用户首先确认当前所使用的是 C-Coupler2 最新释放版本，当确认 C-Coupler2 最新释放版本存在潜在 bug 时，可采用页面 <https://github.com/C-Coupler-Group/c-coupler-lib> 上的“issues”功能提交对 bug 的描述，后面我们将会尽快反馈。也欢迎用户就潜在 bug 直接联系我们（liuli-cess@mail.tsinghua.edu.cn、liruijhe@mail.tsinghua.edu.cn 或 zhangc-cess@mail.tsinghua.edu.cn）。对于首次发现 C-Coupler2 某个 bug 并提供了有效测试用例的用户，我们将为每个 bug 提供 1000-1500 元奖励。

当用户在使用 C-Coupler2 过程中遇到了困难时而需要技术帮助时，也欢迎联系我们。

未来工作

C-Coupler2 一方面在一系列功能特点上比原有 C-Coupler1 有了明显改进，另一方面其为 C-Coupler 的未来发展定义了有机结合了应用程序接口和配置文件的耦合配置接口规范，C-Coupler 的后续版本将服从这一规范，并尽量保证向后兼容性（例如将一个耦合模式的耦合器从 C-Coupler2 升级为 C-Coupler3 后，耦合模式可以直接成功编译后运行起来）。C-Coupler 后续版本的研制将首先考虑以下两方面需求：

- 1) 面向通量计算算法的耦合集功能，其中将支持针对通量计算算法的自动耦合生成。通量计算算法的耦合集成从一开始就是 C-Coupler 的发展目标之一。

C-Coupler1 提供了集成通量算法的程序接口与使用通量计算算法的耦合配置接口，但因要求开发算法 Fortran 程序的 C++ 接口程序而难以使用，灵活性差，且安全性低，且无法用于自动耦合生成。因此，C-Coupler 的后续版本无法使用 C-Coupler1 的相关实现来提供通量计算算法的耦合集功能。C-Coupler2 的耦合配置接口、自动耦合生成功能和对同一分量模式内耦合的支持使得通量计算算法可被当作分量模式使用，这为在 C-Coupler 后续版本中研制易用、灵活性好且安全性好的通量计算算法集成与应用接口奠定了坚实基础。

- 2) 面向模式超高分辨率与高性能计算机众核架构的并行支撑。随着科学与技术的快速发展，模式的分辨率不断提升，已经朝着水平方向全球公里级、区域百米级的超高分辨率方向发展，而高性能计算机也朝着众核架构发展，其中同一个处理器及同一个计算节点内的处理器核越来越多。如何支持超高分辨率分量模式在高性能计算机众核架构下采用高并行度时的高效并行，以及在高并行度情况下如何实现分量模式间的高效并行耦合，这将 C-Coupler 未来的发展要重点解决的问题。

