

# 从构建到封装：HPC-AI 融合下磁约束聚变集成建模中数据 FAIR 的可追溯性研究

Monday, 25 August 2025 15:00 (20 minutes)

科学数据的管理正从单纯的数据存储与共享，向确保研究过程和结果的**可追溯性** (provenance) 和**重复性** (reproducibility) 演进。FAIR 原则 [1] (可发现、可访问、可互操作、可重用) 为这一目标提供了指导框架。然而，正如 FAIR4RS[2] 所强调的，将这些原则应用于“研究软件”比应用于静态数据更具挑战性，因为软件是动态的、可执行的，并且其行为高度依赖于其运行环境。

在复杂大科学工程磁约束聚变研究中，由多个物理模拟程序耦合的集成建模是探索复杂聚变发生过程的核心。数据结果的可靠性就要求我们首先保证集成计算过程的 FAIR。因此，我们开发了用于管理于磁约束聚变集成建模流程软件环境管理的工具，FyDev[3]，保证**软件的整个生命周期** (获取、构建、测试、部署、调用) 进行标准化、自动化和可追溯化。但是，随着 AI/ML 的发展，集成建模中的部分传统物理程序逐渐被替代模型、数据驱动推理模型、基于 GPU 的求解算法等代替，IM 经历着传统高性能计算 (HPC) 向融合人工智能 (AI) 的范式转变，这给软件环境的可追溯性 (provenance) 与 FAIR (可发现、可访问、互操作、可重用) 管理带来了严峻挑战。

传统的 HPC 软件依赖于系统级安装 (如 EasyBuild[4])，环境稳定且易于追踪；而 AI 应用则依赖于动态、易变的 Python 生态和容器化技术。前者是系统级的构建，后者是应用级的封装。完全基于传统包管理、构建工具，很难同时维护 AI 应用的环境。两种模型运行环境的割裂，必然导致计算过程的 provenance 链条在边界处断裂。进一步，虽然业界从虚拟化/云原生的领域衍生出很多成熟的工具，提供对基于 slurm 的传统 HPC 和基于 K8S 的容器化环境的统一管理，但是，这些工具大都构架庞大，服务繁琐，彻底改变用户的使用模式，不考虑实际应用场景、用户使用习惯的生搬硬套，很难彻底解决问题。**磁约束聚变集成模拟 (IM) 中日益增加的 HPC 与 AI 的融合，是一种“自底向上”、由传统 HPC 科学计算环境“演化”而来的需求，而非“自顶向下”、由云计算/虚拟化平台“嫁接”过去的模式。**

因此，我们在 FyDev 之前版本的基础上，将“软件环境的构建过程形式化、自动化和可追溯化”这一普适理念，从 HPC 软件的“构建”扩展到融合 AI 应用“封装”。我们引入“计算单元” (Computational Unit, CU) 的概念，将传统的 HPC 程序和容器化的 AI 应用统一抽象为可管理的实体。为此，我们在 CU\_description 文件中增加了 container 字段，使其能同时描述 HPC 依赖和容器化环境。构建系统 (Build System) 也随之演进：对于 HPC 组件，它调用 EasyBuild 进行编译；对于 AI 组件，它根据 container 字段描述的内容自动生成 Dockerfile 并构建镜像。最终的软件栈 (Software Stack) 不再仅仅是二进制文件目录，而是一个包含本地 HPC 软件和容器镜像仓库的混合体。

我们以磁约束聚变集成建模中典型的“平衡-输运”耦合工作流为例验证了该框架：首先使用 EFIT (HPC 程序) 反演等离子体平衡，然后将结果作为输运程序 TORAX (基于 JAX 加速的 GPU 优化代码) 的输入，同时调用一个数据驱动的代理模型 QLKNN (AI 应用，运行在容器内，依赖驱动版本和 python 环境) 来预测芯部输运剖面。在 FyDev 中，提供标准化的程序调用 API，通过基于配置文件的描述，使用 Slurm，自动化地协调了 HPC 作业的和容器的启动，并追踪了数据传递的过程，实现了该工作流的可重复部署。通过统一的 @pid 和完整的 provenance 记录，有效弥合了 HPC 与 AI 之间的管理鸿沟，为融合时代的科学计算提供了端到端的 FAIR 管理解决方案。

[1] Wilkinson, M. D. et al. The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* 3, 160018 (2016).

[2] Barker, M. et al. Introducing the fair principles for research software. *Sci. Data* 9, 1–6 (2022).

[3] Liu, X., Yu, Z. & Xiang, N. Building an integrated modeling and analysis software environment that follows FAIR principles. Preprint at <https://doi.org/10.48550/arXiv.2310.12345> (2023).

[4] Hoste, K., Timmerman, J., Georges, A. & De Weirdt, S. EasyBuild: Building Software with Ease. In *SC Companion*, 572–582 (2012).

## Summary

本研究针对磁约束聚变集成建模中 HPC 与 AI 融合带来的软件环境管理挑战，提出并实现了 FyDev 框架的演进。传统的 HPC 环境依赖系统级构建，而 AI 应用则依赖动态的容器化技术，两者构建方法的割裂导致在一个集成的计算过程中的结果可追溯性 (provenance) 中断。为此，我们引入“计算单元” (Computational Unit, CU) 的统一抽象概念，并扩展 CU\_description 文件以同时描述 HPC 依赖和容器化环境。FyDev 的构建系统据此可自动化地调用 EasyBuild 编译 HPC 组件，并生成 Dockerfile 构建 AI 容器镜像，形成混合软件栈。通过一个典型的“平衡-输运”耦合工作流 (EFIT-HPC → TORAX-JAX

→ QLKNN-容器化 AI) 的验证, FyDev 利用标准化 API 和 Slurm 调度, 实现了 HPC 作业与容器化 AI 应用的自动化协调与数据传递追踪。该框架为融合时代的科学计算提供了从构建到执行的端到端可追溯、可重复的 FAIR 管理解决方案

**Primary author:** 刘, 晓娟 (Institute of plasma physics, Chinese Academy of Sciences)

**Co-author:** Mr YU, ZHI

**Presenter:** 刘, 晓娟 (Institute of plasma physics, Chinese Academy of Sciences)

**Session Classification:** 科学数据存储与管理

**Track Classification:** 科学数据存储与管理技术