

基于 CUDA 的并行雷达拼图算法研究¹

韩丰 高嵩 薛峰 李月安

国家气象中心, 北京 100081

摘 要

提 要: 雷达组网拼图算法是强对流天气短时临近预报系统(Severe Weather Automatic Nowcasting, SWAN)的重要基础方法之一。提高拼图算法的效率, 不仅可以提升现有 SWAN 临近算法序列的时效性, 也能更好地应用高分辨率雷达数据, 具有重要的实际意义。采用中央处理器(Central Processing Unit, CPU)和图形处理器(Graphics Processing Unit, GPU)混合架构设计并行雷达拼图算法, 其中 CPU 负责雷达数据的解析和调度 GPU 并行模块, GPU 负责大规模数据的并行计算。通过分析计算统一设备架构(Compute Unified Device Architecture, CUDA)算法的并行开销和拼图算法的特点, 提出并实现了 GPU 内存管理优化和数据交换流程优化方案, 提高了组网拼图算法的效率。对比试验结果表明, 基于 CUDA 的 GPU 并行拼图算法和 SWAN 中 30 线程并行的 CPU 算法相比, 在全国 1 km 和 500m 分辨率的拼图任务上, 加速比分别达到 3.52 和 6.82。综上, 基于 CUDA 的并行拼图算法不仅可以提高 SWAN 短临算法序列的时效性, 也为更高分辨率雷达资料的拼图提供了技术支持。

关键词: 组网拼图; GPU 并行; 计算统一设备架构(CUDA); 强对流天气短时临近预报系统(SWAN)

中图分类号: TP319

引 言

随着观测技术和预报业务需求的发展, 气象雷达的分辨率不断提高。2019 年, 中国气象局启动我国新一代天气雷达的双偏振改造和技术升级, 升级后的 S 波段雷达强度径向分辨率由 1 km 提升至 250m。同时, 为了满足精细化临近预报业务的需求, 北京市、江苏省、广东省等地区先后建立 X 波段双偏振雷达网并投入业务使用(张哲等, 2021; 苏永彦和刘黎萍等, 2022; Ryzhkov et al, 2005)。雷达径向分辨率的提升为数据的快速处理和客观算法提出了更高的效率要求。

雷达数据的三维组网拼图技术是雷达应用的重要基础方法之一(程元慧等, 2020; 张勇等, 2019)。在进行雷达组网时, 首先需要将回波从球坐标转换到等经纬度的直角坐标系下, 在通过最大值、均值、距离反比等方案, 在多部雷达的观测重叠区进行融合计算(肖艳姣和刘黎平, 2006)。拼图计算过程中, 涉及插值、坐标系转换、大规模数据网格等问题, 计算密集且数据吞吐量大, 是对计算资源需求量最大的雷达客观方法之一。以强对流短时临近预

¹基金项目: 国家重点研发计划(2022YFC3002904)、国家气象中心气象现代化专项(QXXDH202231)和中国气象局重点创新团队(CMA2022ZD07)资助

作者简介: 韩丰, 主要从事气象短临预报算法研发, E-mail: hanfeng@cma.gov.cn

收稿日期: 修回日期:

报系统 (Server Weather Automatic Nowcasting, SWAN) (韩丰和沃伟峰, 2018) 为例, 全国 1 km 雷达组网拼图算法是临近预报算法中耗时最长的算法, 且雷达拼图后续还有风暴识别和追踪算法 (Thunderstorm Identification, Tracking, Analysis and Nowcasting, TITAN)、外推等其他客观方法。所以提升雷达拼图的运算速度可以明显提升 SWAN 临近预报产品序列的时效性。此外, 考虑到雷达径向分辨率的提升, 将拼图的分辨率提升至 500m 时, 现有的拼图算法无法在一个观测周期 (6min) 内完成计算, 不具备业务应用能力。所以将更先进的并行计算技术应用于雷达组网拼图算法中, 一方面可以提升现有算法序列的时效性, 另一方面也可以更好地应用高分辨率雷达基数据, 具有重要的实际意义。

气象中常用中央处理器 (Central Porcessing Unit, CPU) 并行技术和基于 CPU 的分布式计算框架来提高运算能力。王志斌等 (2009) 通过使用共享存储并行编程 (open Multiple Processing, OpenMP) 并行技术和消息传递接口 (Message Passing Interface, MPI) 编程规范, 实现雷达拼图系统的两级并行。吴石磊等 (2012) 使用多线程技术对全国组网雷达估测降水系统进行效率优化, 通过调整计算负载, 可以将计算效率提升 5 倍。SWAN 系统设计实现了基于 OpenMP 的雷达组网拼图算法, 使用 30 个逻辑线程, 生成全国 1 km 分辨率三维拼图产品约需 2min。

经过多年的发展, CPU 性能提升已经逐渐达到瓶颈, 多核应用的成本-效益比不断下降, 仅仅通过挖掘 CPU 性能提升算法效率已经不能满足现有的需求。图形处理器 (Graphics Processing Unit, GPU) 作为一种新的计算资源, 由于其区别于 CPU 的硬件架构, 特别适合解决大规模数据的计算问题。同时 GPU 并行计算接口的快速发展, 也使得 GPU 在通用计算领域应用越来越广泛 (钱奇峰等, 2021; 黄骄文等, 2021)。

目前国内外也有一些使用 GPU 加速气象算法的研究成果。Vu et al (2013)、Mielikainen et al (2013)、Govett et al (2017)、Schär et al (2020)、王艺儒 (2020)、Wang et al (2021) 和肖洒等 (2019) 等学者将 GPU 并行技术应用于数值模式, 改写相关动力框架和计算模块, 在保持计算精度的同事, 显著提高模式的运行效率。Olander et al (2021) 介绍了 GPU 在台风定强深度学习模型中的应用。Rukmono et al (2018)、王兴等 (2015) 和胡荣华 (2015) 将 GPU 应用雷达数据处理和临近预报算法, 得到了 10 倍以上的加速比。

由于 GPU 程序主要使用图形应用程序接口 (Application Programming Interface, API) 进行编程, 如气象信息综合分析处理系统 (Meteorological Information Combine Analysis and Process System) MICAPS4 (高嵩等, 2017) 使用开放式图形语言 (Open Graphics Language, OpenGL) 开发气象数据渲染引擎。但是图形 API 并不适用于通用计算任务。因此比较主流的 GPU 并行算法一般通过一些标准化的并行编程接口进行实现。常用的并行计算接口有开放运算语言 (Open Computing Language, OpenCL) 如王兴等 (2015), OpenACC(Open Acceleration, 开源加速接口), 如 Govett et al (2017)、Schär et al (2020)、肖洒等 (2019), 计算统一设备架构 (Compute Unified Device Architecture, CUDA), 如 Wang et al (2021) 等。本文设计实现基于 CUDA 的雷达组网拼图算法, 并通过分析 CUDA 算法的并行开销和组网拼图算法的流程, 提出高效的 CPU+GPU 的并行拼图算法实现方案。通过和现有 SWAN 中的 CPU 并行组网拼图算法对比, 获得较好的性能优化和实际效果。

综上所述, 本文为了提高 SWAN 临近客观算法的时效性, 更好的应用高分辨率雷达资料, 将 GPU 并行技术应用于雷达组网拼图算法, 设计实现了基于 CUDA 的并行拼图算法。通过

算法性能分析，提出并实现了一种基于 GPU 内存管理和数据流程优化的方案。对比试验表明，基于 CUDA 的并行算法能获得超过 3.5 以上的加速比。

1. 基于 CUDA 的 GPU 并行计算模型

1.1 CUDA 编程模型

CUDA 是 NVIDIA 公司发布的通用并行计算平台和编程模型，它提供了 GPU 编程的简易接口，如初始化 GPU 空间，主机和 GPU 设备间的数据传输等。基于 CUDA 编程可以构建 GPU 计算的应用程序，而无需关注底层的图形 API。

CUDA 是一个异构编程模型，可以协同调度 CPU 和 GPU，共同完成一个计算任务。在 CUDA 编程模型中，计算资源被分为主机端（host）和设备端（device），其中主机端表示 CPU 及其内存，设备端表示 GPU 及其内存。CUDA 程序中既包含主机端模块，又包含设备端模块，它们分别在 CPU 和 GPU 上运行。

1.2 CUDA 并行算法流程

如前文所述，GPU 具有独立的内存系统，所以 CPU 在调度 GPU 进行并行计算时，首先需要将数据从 CPU 内存拷贝到 GPU 内存，最后再将计算结果拷贝回 CPU 内存。表 1 给出的是一个典型 CUDA 算法和 CPU 算法的执行流程。对比两者可以发现，CUDA 算法多了一次内存的分配，一次内存的释放和两次数据的搬运。这四次操作可以认为是 CUDA 并行算法所必须的并行开销。下文将通过组网拼图试验，按照表 1 中的步骤，分析基于 CUDA 的雷达组网拼图算法效率瓶颈，并给出优化方案。

表 1 CDUA 算法和 CPU 算法的执行流程对比

Table 1 Comparison of execution flow between CUDA algorithm and CPU algorithm

步骤	GPU 算法	CPU 算法
1	分配 CPU 内存，从文件载入数据	分配 CPU 内存，从文件载入数据
2	分配 GPU 内存	
3	将数据从 CPU 内存拷贝至 GPU 内存	
4	调用函数在 GPU 上完成指定计算	调用函数在 CPU 上完成指定计算
5	将计算从 GPU 内存拷贝至 CPU 内存	
6	释放 GPU 内存	
7	输出计算结果，释放 CPU 内存	输出计算结果，释放 CPU 内存

2. 基于 CUDA 的雷达组网拼图算法设计

2.1 基于 CPU 并行的雷达组网拼图算法设计

在雷达组网拼图算法中，首先需要每个单站分别进行三维离散化处理，最后将所有站的三维离散化结果拼接到一个大的数据网格中。在多雷达重叠部分，按照雷达融合策略进行取值，目前常用的策略为最大值法、距离反比权重法等。在 SWAN 中的 CPU 并行拼图算法和本文设计的 CUDA 并行算法中，均采用肖艳姣和刘黎平（2006）提出的三维离散化方案，并按照距离反比权重法进行重叠区融合。

SWAN 中目前使用的 CPU 并行拼图算法包括两个并行块。并行块 1：由于每个站雷达数

据都是独立离散化的，则设计并行方案，在多线程中并行处理不同站的数据；并行块 2：在离散化后数据组网拼图过程中，由于不同垂直层之间也是相互独立的，则设计并行方案，并行处理多层的数据拼接。并行块 1 和并行块 2 为顺序执行，即先完成全部站点的离散化之后，再进行多站拼图。

雷达三维离散化插值分为两步，首先计算出平面直角坐标系中点 $P(x,y,z)$ 在雷达站为中心的球坐标系中的位置 (r, θ, φ) ，随后将雷达基数据用双线性插值的方法插值到直角坐标系中。可以采用预先计算坐标转换查找表的方式，加快雷达数据空间插值的计算速度。当硬盘读取速度大于查找表计算速度时，优化是有效果的。

2.2 基于 CUDA 的雷达组网拼图算法设计

2.2.1 CUDA 并行粒度分析

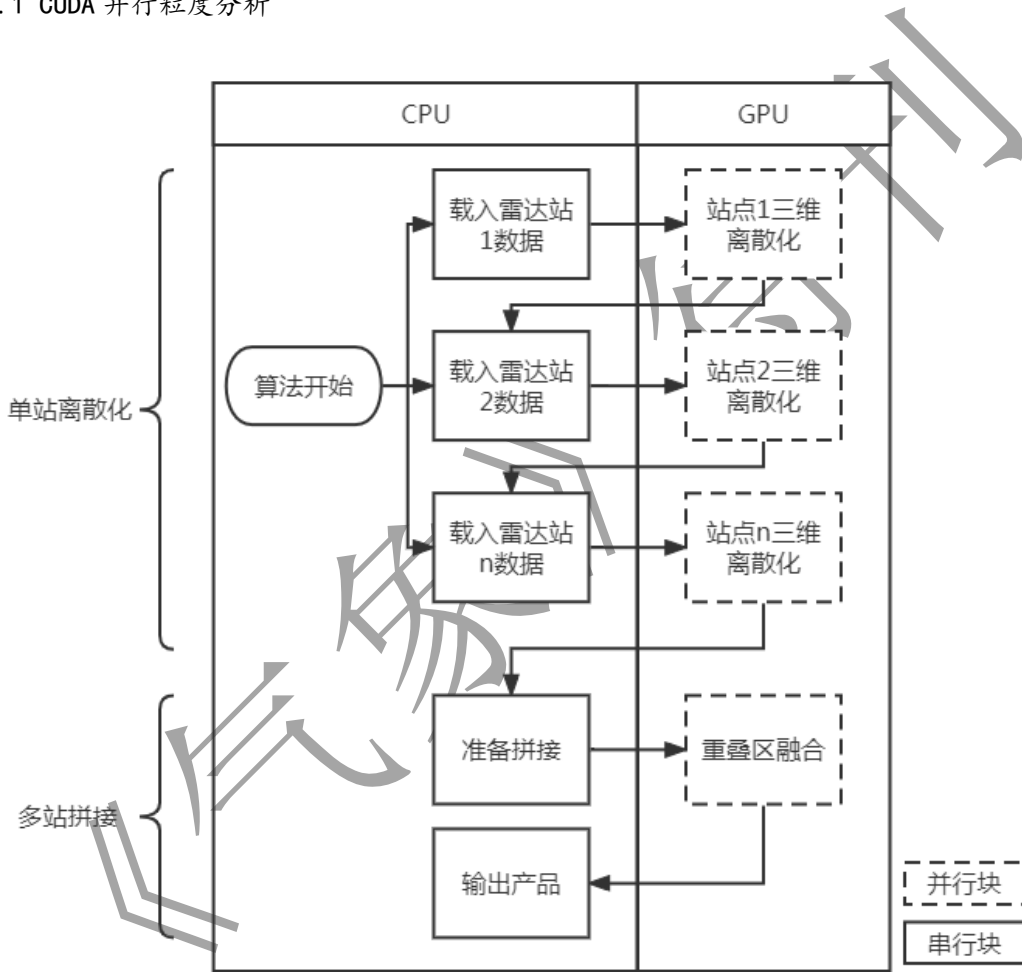


图 1 CUDA 并行组网拼图算法流程设计

Fig.1 Process design of radar mosaic algorithm based on CUDA

上述 CPU 并行拼图算法是合理而且有效的，主要是因为 CPU 一般只有几十个逻辑核心，且每个核心的计算能力很强，可以处理复杂的任务。将不同的站点和层次分配到单独的线程里面，既可以充分利用所有的核心，还能够充分发挥 CPU 核心的性能，减少频繁创建、销毁线程的开销。

但是 GPU 一般有几十流式多处理器（Stream Mutilprocessor），每个流式多处理器可以同时启动上千个 CUDA 计算单元，且每一个计算单元的计算能力较弱，不能完成一个站点的

三维离散化任务。所以要充分发挥 GPU 的并行性能，需要设计格点级的并行方案。

2.2.2 CUDA 并行拼图算法初步设计

图 1 给出了 CUDA 并行拼图算法初步设计流程图。算法采用 CPU+ GPU 并行混合的设计方案。其中 CPU 负责雷达基数据的解析和最终拼图产品的输出，GPU 负责每一步插值算法的并行计算。整个计算流程分为两个部分，单站的三维离散化插值和多站拼图。在单站三维离散化插值时，由 CPU 作为主要的调度程序，负责解压、解析雷达基数据，并调用 GPU 中的三维插值模块，实现并行计算。不同雷达站之间的离散化部分采用串行的方式，依次进行。在多站拼接阶段，依然使用 CPU 作为调度程序，调用 GPU 中的空间插值模块，实现多站拼图的并行计算。

2.2.3 效率分析

图 2 给出了 GPU 中单站三维离散化并行模块的执行流程。对照表 1，GPU 并行计算中包括内存管理，数据传输和计算模块三个部分。表 2 给出了一个单站三维离散化实例（1 km 分辨率）和多站拼图（213 个站）实例的运行时间分布。表中可以看出单站的三维离散化总计耗时 0.483s，考虑到多站雷达数据三维离散化之间采用串行方式，以表 2 的执行速度，处理全国大约 200 个站的数据大概需要 80~90 s，再算上 CPU 解析处理雷达基数据的时间，GPU 并行算法和 CPU 并行算法相比，没有明显的效率优势。由表 2 可以看出单站离散化过程中有超过 99%的运行时间都用在了内存管理和数据传输方面。所以从优化算法，提高效率的角度考虑，应提高内存管理和数据传输部分的效率。多站拼图并行模块的时间分布与此类似，不做赘述。

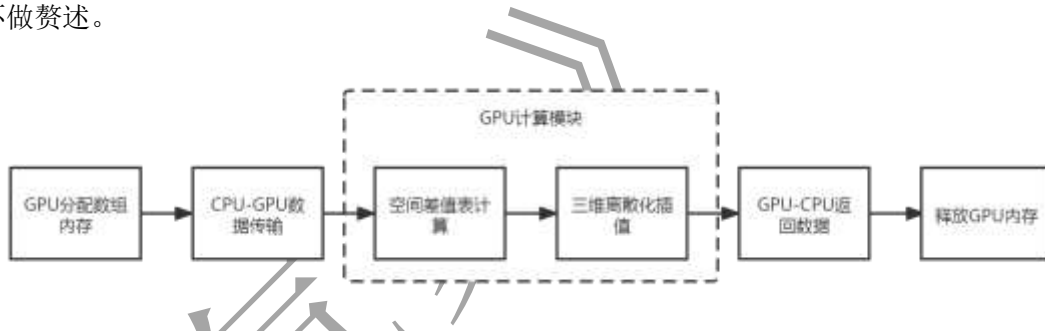


图 2 基于 CUDA 的单站三维离散化并行模块

Fig.2 Parallel module of radar archives discretization based on CUDA

表 2 基于 CUDA 的组网拼图算法中的耗时分配（单位:s）

Table 2 Time consuming in each step of radar mosaic algorithm based on CUDA (unit:s)

任务名称	分配空间	CPU-GPU 传输数据	GPU 计算模块	GPU-CPU 返回数据	释放空间	总计
单站离散化	0.269	0.003	<0.001	0.117	0.092	0.483
多站拼图	0.115	3.403	<0.001	1.203	0.193	4.914

2.2.4 改进方案

通过效率分析，本文从内存管理和数据传输两个方面提出 CUDA 组网拼图算法改进方案。

改进 1: 以计算代替存储。考虑到 GPU 计算的耗时要明显少于内存管理的耗时，尽可能

减少数据存储和设备间的数据拷贝，进而减少内存管理开销。通过表 2 可以看出，单站离散化中的空间插值表计算和三维离散化插值计算总计用时不到 0.001s。由于存储查找表需要在 GPU 中申请单独的大数组，其内存管理时间远远大于计算时间。由此，将空间查找表计算和三维离散化插值合并成一个并行计算模块，格点的坐标系转换关系直接存储在 GPU 中。通过测试，使用改进 1 后，分配空间的耗时由 0.269 s 下降至 0.202s。

改进 2：增加数据结构复用，避免重复申请和释放 GPU 内存空间。考虑到 CPU 以串行的方式调度 GPU 进行不同站的三维离散化计算，在不同站处理时，部分数据结构可以直接进行复用，减少 GPU 内存管理开销。在算法开始时，可以一次性申请全部内存空间，并在最后统一进行空间释放，减少内存管理的操作次数。通过测试，使用改进 2 后，200 个站的内存申请耗时 0.428s，内存释放耗时 0.192s，效率提升非常明显。

改进 3：以存储代替传输。组网拼图算法的第二步是将单站离散化之后的数据融合并接到拼图网格上，需要将单站离散化结果从单站模块搬运到多站拼图模块。此处考虑到由于两个并行模块都在 GPU 中完成，可以将单站离散化结果临时存储在 GPU 内存中，拼图模块直接从 GPU 内存读取相关数据进行拼图。使用改进 3 后，可以省去单站三维离散化模块的数据返回耗时和多站拼图模块的数据传输耗时。

表 3 给出了改进前后的 GPU 并行开销对比。在四个对比项目中，除了 CPU-GPU 传输数据部分是必要开销外，其余三部分都得到了明显优化，并行开销总计时间也得到了显著下降。

表 3 改进前后 GPU 组网算法并行开销对比（单位：s）

Table 3 Comparison of parallel overhead of GPU networking algorithms before and after optimization

based on CUDA (unit:s)					
任务名称	分配空间	CPU-GPU 传输数据	GPU-CPU 返回数据	释放空间	总计
改进前	53.221	0.604	25.621	17.991	97.437
改进后	0.766	0.622	1.988	0.489	3.865

2.2.5 基于 CUDA 的并行组网拼图算法设计

综上，图 3 给出了 CUDA 并行拼图算法的最终设计，图中 n 表示雷达站点数。和 2.2.2 节相比，最终设计增加了对于 GPU 内存的管理和 CPU-GPU 数据传输的优化，即上文提到的 3 个优化方案。GPU 完成单站三维离散化计算后，仅发送回调消息，通知 CPU 开始下一个雷达站数据的处理。拼图模块直接从 GPU 内存中读取数据，完成计算，最后将拼图产品返回给 CPU 端。通过数据流和内存管理的优化，组网算法一共只有 n 次 CPU-GPU 数据拷贝，1 次 GPU-CPU 的数据拷贝，1 次申请空间和 1 次释放空间。

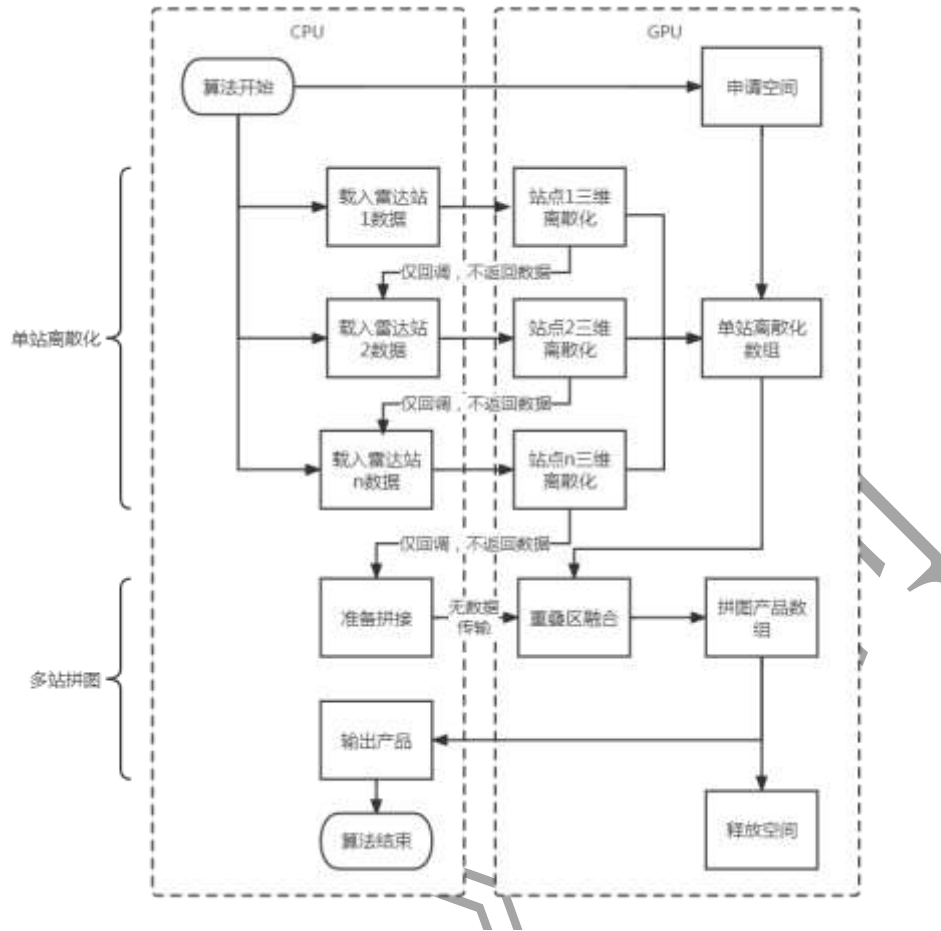


图 3 基于 CUDA 的并行拼图算法设计

Fig.3 Flow chart of of radar mosaic algorithm based on CUDA

3. 组网拼图算法对比试验

3.1 对比试验介绍

SWAN 实时业务逐 6 分钟生成全国 (70° E~135° E, 18° N~55° N) 三维 (垂直 20 层) 拼图产品, 每个时次大约处理 210 个雷达 (约 2GB) 的数据。以此业务场景为基础, 本文设计两组对照试验, 验证 CPU 并行和 GPU 平行拼图算法的效率。试验内容如表 4 所示, 其中 CPU 并行算法采用 SWAN 系统中的拼图算法 (30 线程并行)。试验在同一服务器上进行, 其中 CPU 为 E5-2690, 内存 64G, 显卡为 GTX1080, 显存 8G。耗时采用 5 次运算取均值的方法。

表 4 对比试验

Table 4 Comparison experiments of algorithm efficiency

试验名称	试验内容	算法	时间复杂度	空间复杂度
试验一	全国 1 km 雷达拼图	CUDA 并行	约 $O(4 \times 10^9)$	约 $O(2 \times 10^9)$
		CPU 并行		
试验二	全国 500m 雷达拼图	CUDA 并行	约 $O(16 \times 10^9)$	约 $O(8 \times 10^9)$
		CPU 并行		

3.2 运行效率分析

本文按照公式 1 进行算法运行效率分析

$$t_{total} = t_{decode} + t_{station} + t_{mosaic} \quad (1)$$

式中： t_{total} 表示算法的总运行时间， t_{decode} 表示解析全部雷达基数据的耗时， $t_{station}$ 表示全部单站完成三维离散化的耗时， t_{mosaic} 表示多站拼图的耗时。

使用加速比 S_G 定量评估 CUDA 并行算法加速效果

$$S_G = \frac{t_C}{t_G} \quad (2)$$

式中： t_C 表示 CPU 并行算法的运行时间， t_G 表示 CUDA 并行算法的运行时间。 S_G 越大，说明 CUDA 算法加速效果越好。

表 5 给出了算法效率试验结果。对比 t_{total} 可以看出，CUDA 并行算法的效率优势明显， S_G 分别达到 3.52 和 6.82。对比试验一、试验二可以看出，随着拼图分辨率的提高，CUDA 算法的效率优势也更加明显。这主要是由于在当前拼图的计算量下，CUDA 算法的耗时主要是内存管理和数据传输。当拼图分辨率提高时，GPU 内存管理和拼图产品的传输开销变大，并行计算部分的耗时还是仅占不到 10%。这说明 CUDA 并行算法具备应对更大规模计算的潜力。表中 $t_{decoder}$ 主要包括 CPU 解压和解析雷达基数据的时间，此步骤在四个试验中是基本一致的。从试验结果也可以看出，四个试验方案的耗时接近，符合预期。如果排除 t_{decode} ，仅考虑计算部分的加速比， S_G 分别达到 11.8 和 12.4。表中的试验二表明，基于 CUDA 的并行拼图算法可以在 40s 内完成全国 500m 分辨率拼图的运算，说明应用 GPU 并行技术之后，更高分辨率的雷达拼图具备一定的业务可行性。

表 5 算法效率试验结果（单位：s）

Table 5 The result of algorithm efficiency experiments (unit:s)

试验名称	算法	S_G	t_{total}	t_{decode}	$t_{station}$	t_{mosaic}
试验一	CUDA 并行	3.52	24.067	18.452	3.544	2.071
	CPU 并行		84.786	18.501	52.877	13.408
试验二	CUDA 并行	6.82	37.176	18.263	12.584	6.329
	CPU 并行		253.712	18.372	194.331	41.009

4. 结论

本文基于 CUDA 设计一种 CPU+GPU 并行的全国雷达组网拼图算法，其中 CPU 负责文件解析和流程调度，GPU 负责大规模的插值并行计算。在算法设计时，考虑到 GPU 特性和拼图算法特点，通过优化 GPU 内存管理和数据流程，减少 CUDA 算法的并行开销，和 CPU 并行算法（30 线程）相比，获得了超过 3.5 的加速比。

本文的试验表明，CUDA 并行拼图算法可以在 40s 内完成全国 500m 分辨率拼图的实时计算，为高分辨率雷达资料在业务中的应用提供了一种新的方式。本文设计的雷达组网拼图算法，可以代表着计算密集、数据吞吐量大的这一类气象算法的 CUDA 并行方案。此类算法

都可以通过本文类似的方法实现 CUAD 的并行化改造。

在今后的工作中,考虑将拼图算法和其前序的单站质控、后序的外推算法进行进一步整合,直接实现基于 GPU 的 SWAN 算法序列。此外,配合中国气象局雷达流传输业务改革,拼图算法可以直接对接雷达基数据数据流,通过在体扫观测结束前,预先加载部分数据的方式,进一步提高产品时效。

参 考 文 献

- 程元慧,傅佩玲,胡东明,等,2020.广州相控阵天气雷达组网方案设计及其观测试验[J].气象,46(6):823-836. Cheng Y H, Fu P L, Hu D M, et al, 2020. The Guangzhou phased-array radar networking scheme set-up and observation test[J]. Meteor Mon, 46(6): 823-836 (in Chinese).
- 高嵩,毕宝贵,李月安,等,2017. MICAPS4 预报业务系统建设进展与未来发展[J]. 应用气象学报,28(5):513-531. Gao S, Bi B G, Li Y A, et al, 2017. Implementation and development plan of MICAPS4[J]. J Appl Meteor Sci, 28(5): 513-531 (in Chinese).
- 韩丰,沃伟峰,2018. SWAN2.0 系统的设计与实现[J]. 应用气象学报,29(1):25-34. Han F, Wo W F, 2018. Design and implementation of SWAN2.0 platform[J]. J Appl Meteor Sci, 29(1): 25-34 (in Chinese).
- 胡荣华,2015. 并行计算在临近天气预报系统中的应用研究[D]. 广州:华南理工大学. Hu R H, 2015. Research on application of parallel computing in nowcasting system[D]. Guangzhou: South China University of Technology (in Chinese).
- 黄骄文,蔡荣辉,姚蓉,等,2021. 深度学习网络在降水相态判别和预报中的应用[J]. 气象,47(3):317-326. Huang J W, Cai R H, Yao R, et al, 2021. Application of deep learning method to discrimination and forecasting of precipitation type[J]. Meteor Mon, 47(3): 317-326 (in Chinese).
- 钱奇峰,王川,徐雅静,等,2021. 一种基于深度学习的台风强度估测技术[J]. 气象,47(5):601-608. Qian Q F, Wang C, Xu Y J, et al, 2021. A deep learning technique of typhoon intensity estimation[J]. Meteor Mon, 47(5): 601-608 (in Chinese).
- 苏永彦,刘黎平,2022. S 波段双偏振雷达和 X 波段相控阵天气雷达中气旋识别结果对比[J]. 气象,48(2):229-244. Su Y Y, Liu L P, 2022. Comparison of mesocyclone identification results between S-band dual polarization radar and X-band phased array weather radar[J]. Meteor Mon, 48(2): 229-244 (in Chinese).
- 王兴,王新,苗春生,等,2015. 基于 GPU 加速的雷暴追踪外推方法研究[J]. 南京师范大学学报(工程技术版),15(1):35-42. Wang X, Wang X, Miao C S, et al, 2015. Research on thunderstorm track and extrapolation based on GPU-acceleration[J]. Journal of Nanjing Normal University (Engineering and Technology Edition), 15(1): 35-42 (in Chinese).
- 王艺儒,2020. 闪电数值模式 GPU 并行化计算及其初步应用[D]. 南京 南京信息工程大学, Wang Y R, 2020. Parallel computation of lightning numerical mode GPU and its preliminary application[D]. Nanjing: Nanjing University of Information Science and Technology (in Chinese).
- 王志斌,陈波,万玉发,等,2009. 天气雷达资料实时并行处理方法[J]. 计算机工程,35(23):255-257. Wang Z B, Chen B, Wan Y F, et al, 2009. Weather radar data real-time parallel processing method[J]. Comput Eng, 35(23): 255-257 (in Chinese).
- 吴石磊,安虹,李小强,等,2012. 组网雷达估测降水系统并行化方案的设计与实现[J]. 计算机科学,39(3):271-275. Wu S L, An H, Li X Q, et al, 2012. Parallel program design and

-
- implementation on precipitation program of networking weather radar system[J]. *Comput Sci*, 39(3): 271-275 (in Chinese).
- 肖洒, 魏敏, 邓帅, 等, 2019. 基于 GPU-OpenACC 的气候模式加速优化研究[J]. *气象*, 45(7): 1001-1008. Xiao S, Wei M, Deng S, et al, 2019. Research on accelerated optimization of climate models based on GPU-OpenACC[J]. *Meteor Mon*, 45(7): 1001-1008 (in Chinese).
- 肖艳姣, 刘黎平, 2006. 新一代天气雷达网资料的三维格点化及拼图方法研究[J]. *气象学报*, 64(5): 647-657. Xiao Y J, Liu L P, 2006. Study of methods for interpolating data from weather radar network to 3-D grid and mosaics[J]. *Acta Meteor Sinica*, 64(5): 647-657 (in Chinese).
- 张勇, 吴胜刚, 张亚萍, 等, 2019. 基于 SWAN 雷达拼图产品在暴雨过程中的对流云降水识别及效果检验[J]. *气象*, 45(2): 180-190. Zhang Y, Wu S G, Zhang Y P, et al, 2019. Identification and effect verification of convective cloud precipitation in rainstorm processes based on SWAN mosaic products[J]. *Meteor Mon*, 45(2): 180-190 (in Chinese).
- 张哲, 戚友存, 朱自伟, 等, 2021. 深圳 S 波段与 X 波段双偏振雷达在定量降水估计中的应用[J]. *气象学报*, 79(5): 786-803. Zhang Z, Qi Y C, Zhu Z W, et al, 2021. Application of radar quantitative precipitation estimation using S-band and X-band polarimetric radars in Shenzhen[J]. *Acta Meteor Sin*, 79(5): 786-803 (in Chinese).
- Govett M, Rosinski J, Middlecoff J, et al, 2017. Parallelization and performance of the NIM weather model on CPU, GPU, and MIC processors[J]. *Bull Am Meteor Soc*, 98(10): 2201-2213.
- Mielikainen J, Huang B, Huang H L A, et al, 2013. Speeding up the computation of WRF double-moment 6-class microphysics scheme with GPU[J]. *J Atmos Oceanic Technol*, 30(12): 2896-2906.
- Olander T, Wimmers A, Velden C, et al, 2021. Investigation of machine learning using satellite-based advanced Dvorak technique analysis parameters to estimate tropical cyclone intensity[J]. *Wea Forecast*, 36(6): 2161-2186.
- Rukmono S A, Kistijantoro A I, Perdana R S, 2018. Optimizations of dual polarization FMCW weather radar signal processing on CUDA platform[C]//2018 5th International Conference on Data and Software Engineering (ICoDSE). Mataram, Indonesia: IEEE: 1-6.
- Ryzhkov A V, Schuur T J, Burgess D W, et al, 2005. The joint polarization experiment: Polarimetric rainfall measurements and hydrometeor classification[J]. *Bull Am Meteor Soc*, 86(6): 809-824.
- Schär C, Fuhrer O, Arteaga A, et al, 2020. Kilometer-scale climate models: prospects and challenges[J]. *Bull Am Meteor Soc*, 101(5): E567-E587.
- Vu V T, Cats G, Wolters L, 2013. Graphics processing unit optimizations for the dynamics of the HIRLAM weather forecast model[J]. *Concurr Comp Pract Exper*, 25(10): 1376-1393.
- Wang Z Z, Wang Y Z, Wang X C, et al, 2021. GPU-RRTMG_SW: accelerating a shortwave radiative transfer scheme on GPU[J]. *IEEE Access*, 9: 84231-84240.

Study of Method for Radar Networking Based on CUDA

HAN Feng, GAO Song, XUE Feng, LI Yue'an

Abstract: Radar networking method is one of the most important methods in Severe Weather Automatic Nowcasting (SWAN) radar applications. Improving the efficiency of the radar networking method, not only can improve the timeliness of the short-term algorithm sequence, and also can make good use of high-resolution radar data, which has important practical significance. With the development of GPU(Graphics Processing Unit) general computing interface, GPU has also been regarded as a powerful computing resource rather than display device for rendering and images. Therefore, this study proposes a new GPU parallel radar networking method based on CUDA(Compute Unified Device Architecture). The method is designed in a hybrid architecture of CPU(Central Processing Unit) and GPU, in which the CPU is for the decoding of radar data and scheduling the GPU parallel modules, and the GPU is for the parallel computing of large-scale data. By analyzing the parallel overhead of the CUDA and the characteristics of the radar networking method, a scheme of GPU memory management optimization and data exchange process simplification is proposed and implemented, which effectively improves the efficiency of the method. The comparative test results show that, compared with the CPU parallel algorithm in SWAN, the GPU parallel networking method based on CUDA achieves a speed-up ratio of 3.52 and 6.82, respectively, on the national puzzle tasks of 1km and 500 m resolution. To sum up, the parallel networking method based on CUDA can not only improve the timeliness of the short-term nowcasting methods sequence, but also provide technical support for the puzzle of higher resolution radar data.

Key Words: Graphics Processing Unit parallel; Compute Unified Device Architecture (CUDA); radar network method; severe weather automatic nowcasting (SWAN)