



射电脉冲星搜索的优化方法

韦建文¹, 张晨飞¹, 张仲莉^{2,3*}, 余婷^{2,4}, 林新华¹, 安涛²

1. 上海交通大学网络信息中心, 上海 200240;

2. 中国科学院上海天文台, 射电天文重点实验室, 上海 200030;

3. 上海引力波探测前沿科学中心, 上海 200240;

4. 中国科学院大学天文与空间科学学院, 北京 100049

*联系人, E-mail: zzl@shao.ac.cn

收稿日期: 2022-06-29; 接受日期: 2022-11-03; 网络出版日期: 2023-01-11

中国SKA专项(编号: 2020SKA0120200)、国家重点研发计划(编号: 2018YFA0404603)、国家自然科学基金(编号: 12041301, 11873079)和中国科学院青年创新促进会项目(编号: 2021258)资助项目

摘要 天文学计算通常具有数据量大、计算量多的特点. 平方公里阵列射电望远镜(Square Kilometre Array, SKA)等装置的建设和高性能计算平台的发展促进了天文学与高性能计算之间的联系. 脉冲星搜索是SKA的主要科学方向之一. 本文介绍了一种基于OpenMP多线程和Multiprocessing多进程技术加速脉冲星搜索管线的方案, 提出了一种解决负载不平衡问题的方法, 并成功地将优化后的管线安装于中国SKA区域中心原型机的x86和ARM计算节点. 在默奇森大视场阵列望远镜(Murchison Widefield Array, MWA)的3个脉冲星搜索样例上的测试显示, 与原始单核心管线相比, 优化后的管线在x86和ARM节点上分别获得10.4–12.2倍和24.5–25.8倍的加速, 并行效率达到37.1%–43.6%和24.5%–26.9%. 并且ARM节点比x86节点的计算快1.1–1.3倍, 显示出国产计算设备在SKA数据处理方面的巨大潜力. 这项应用优化成果, 近期将用于加速MWA南天快速两米巡天(Southern-sky MWA Rapid Two-metre, SMART)项目的脉冲星搜寻工作.

关键词 平方公里阵列, 脉冲星, 脉冲星搜索, 高性能计算, 并行优化

PACS: 07.05.-t, 07.05.Bx, 07.05.Kf, 95.85.-e, 95.85.Bh

1 引言

脉冲星是具有强磁场的快速旋转中子星, 它于1967年在低频射电观测中首次被发现^[1], 在时域天文学、致密天体结构、星际介质模型以及引力波探测等领域有着重要价值. 脉冲星的典型半径约为10–12 km, 质量约为1.4倍太阳质量 $M_{\odot}$ ^[2]. 观测研究显示, 脉冲星的辐射源于磁极, 其磁轴和自转轴存在夹角. 当

辐射束扫过地球的时候, 能够在地面捕捉脉冲星信号, 对其进行观测和研究.

脉冲星在天文学和物理学上的巨大应用价值是建立在大样本基础上的, 因此脉冲星的搜寻工作是发挥脉冲星价值的基础. 脉冲星搜索已经在全球范围内展开, 迄今为止已发现超过3300颗射电脉冲星(ATNF目录^[3]). 它们大部分(约70%)分布在银道面 $|b| < 10^{\circ}$ 的范围内, 只有一小部分分布在高银

引用格式: 韦建文, 张晨飞, 张仲莉, 等. 射电脉冲星搜索的优化方法. 中国科学: 物理学 力学 天文学, 2023, 53: 229506
Wei J W, Zhang C F, Zhang Z L, et al. Parallel optimization of the pulsar search pipeline (in Chinese). Sci Sin-Phys Mech Astron, 2023, 53: 229506, doi: [10.1360/SSPMA-2022-0264](https://doi.org/10.1360/SSPMA-2022-0264)

纬地区,这是由于我们观测到的脉冲星主要在银河系内,并且与脉冲星是由超新星爆炸产生的理论是一致的. 国外知名的脉冲星巡天项目包括美国Arecibo望远镜的银道面L波段馈源阵列脉冲星巡天(the Pulsar Arecibo L-Band Feed Array, PALFA)^[4]和澳大利亚Parkes望远镜的高时间分辨率宇宙脉冲星巡天(High Time Resolution Universe, HTRU)^[5,6]等. 我国自主研制的500米口径球面射电望远镜“天眼”(Five-hundred-meter Aperture Spherical radio Telescope, FAST)^[7-9],至今已发现约500颗新脉冲星^[10-14],主要来自FAST银道面脉冲星巡天(The FAST Galactic Plane Pulsar Snapshot survey, GPPS)^{[12]1)}、FAST多科学目标同时扫描巡天(Commensal Radio Astronomy FAST Survey, CRAFTS)^{[13]2)}和球状星团脉冲星的搜寻^{[14]3)}等巡天项目.

同时,一些巡天项目在极低的频率(通常指350 MHz以下)也发现了脉冲星. 如美国的Green Bank北天盖巡天项目(Green Bank Northern Celestial Cap, GBNCC)^[15]利用Green Bank射电望远镜在350 MHz对南纬40°以北天区进行全面搜寻,发现了195颗脉冲星⁴⁾. Arecibo在327 MHz的巡天,自2010年起,已发现了96颗脉冲星^{[16]5)}. 荷兰低频阵列射电望远镜(Low Frequency Array, LOFAR)^[17]的相干波束全天巡天(The LOFAR Tied-Array All-Sky Survey, LOTASS)在更低的135 MHz频率上发现了73颗脉冲星^[18,19].

正在建设的平方公里阵列(Square Kilometre Array, SKA)射电望远镜有望在一定程度上克服单天线射电望远镜灵敏度和巡天效率难以提高的缺点,获得前所未有的高灵敏度、大视场和高巡天速度^[20,21]. 其第一阶段项目(SKA1,约占整个规模的10%)预期能够发现18000多颗脉冲星,这个数字超过了目前已发现的5倍^[22]. 然而,搜寻更多的脉冲星需要更长的积分时间,在此过程中将产生海量观测数据. 在SKA建成之前,已经有许多SKA的探路者设备投入使用,脉冲星搜索过程数据量大的特点已经可以从中间预见. 例如位于西澳大利亚的默奇森大视场阵列望远镜(Murchison

Widefield Array, MWA)^[23],是SKA-Low的先导项目,其视场包括具有最密集脉冲星分布的银河系中心区域. MWA的电压捕获系统(VCS)^[24]覆盖MWA 70–300 MHz的频率范围,在30.72 MHz的总带宽上细分了3072个频率通道,系统具备10 kHz的频率分辨率. 在0.1 ms的采样间隔下,每小时将产生28 TB观测数据. 当SKA1完成后,数据生成速率最高可达每秒TB量级^[25]. 巨大的数据量带来了计算上的挑战,也带来了科学发现的新机遇,仅MWA项目预期将发现约230颗新的脉冲星^[26].

除了观测时间与观测数据量的增长,脉冲星的加速搜寻算法的计算量也会随着观测时间呈立方增长的趋势(如文献[22]中图1所示). SKA1生成的海量数据将需要至少300 PFlops浮点运算能力^[21],这与撰稿时世界上最快的超级计算机的计算能力处于同一量级(Fugaku超级计算机的峰值计算能力为442 PFlops^[27]). 考虑到算法效率,实际数据处理的算力需求将大大超过这个理论估计.

顺利完成海量数据的处理和存储是SKA正常运行的前提,因此SKA主要成员国正计划建立各自的SKA区域中心(SKA Regional Centre, SRC),执行科学数据深度分析和数据产品长期存储的任务,并且支撑全球科学用户的工作^[28]. 分布式SRC将形成一个全球的协作和协调网络,负责SKA的科学数据产品^[28]的共享与流通. 全球SRC网络的设计、开发、运营和维护符合SKA天文台需求和国家/地区的战略发展.

上海天文台正牵头推进中国SKA区域中心原型系统(China SKA Regional Centre Prototype, CSRC-P)建设,并且已建成世界第一台SRC原型机^{[29,30]6)}及其软件系统^[31]. SKA区域中心建成后,将以MWA为基础,开拓和澳大利亚SKA探路者(Australian SKA Pathfinder, ASKAP)、南非台地高原阵列望远镜(More Karoo Array Telescope, MeerKAT)、荷兰LOFAR等望远镜阵列的合作,开展SKA先导项目的科学预研和技术发展,研究内容包括低频射电连续谱巡天观测^[32],

1) <http://zmtt.bao.ac.cn/GPPS/GPPSnewPSR.html>.

2) http://groups.bao.ac.cn/ism/CRAFTS/202203/t20220310_683697.html.

3) <https://fast.bao.ac.cn/cms/article/65/>.

4) <http://astro.phys.wvu.edu/GBNCC/>.

5) <http://www.naic.edu/~deneva/drift-search/>.

6) <https://www.skatelescope.org/news/china-ratifies-skao-convention/>.

以及本文介绍的射电脉冲星搜索加速研究等. 在全面建设SRC之前, 原型机的研制和测试至关重要, 将承担SKA先导和探路望远镜的数据处理技术验证、压力测试、流程开发和科学数据分析的任务. CSRC-P配备了多核处理器、全闪存并行文件系统和100 G高速IB网络系统, 为改进脉冲星搜索管线等SKA科学应用提供了测试平台. CSRC-P采用混合异构计算架构, 同时配置了使用x86和ARM⁷⁾处理器的计算节点, 是第一个使用ARM架构的SKA数据处理系统. 正如2019年的SKA工程会议⁸⁾上所展示的, ARM计算节点的基准测试显示了出色的性能.

如何让脉冲星搜索的计算任务在高性能计算机上高效执行, 是一项有挑战的工作. 由Ransom^[33,34]开发的PRESTO开源程序, 同Lorimer^[35]开发的SIGPROC和Barr^[36]开发的Peasoup一样, 是当前应用范围较广的脉冲星搜索和观测数据分析管线路程序. PRESTO的最初设计目的, 是从对球状星团的长时间观测中有效地搜索毫秒脉冲双星^[33], 后来则被广泛地应用于大天区的脉冲星巡天搜寻, 如Arecibo的PALFA巡天^[4]、LOFAR的LOTAAS巡天^[18]和FAST的GPPS巡天^[12]等. 目前, 已利用PRESTO搜索到了600多颗脉冲星, 其中包括230多颗毫秒脉冲星和毫秒脉冲双星^[34].

研究人员在CPU, GPU, FPGA等多种计算设备上分析和优化脉冲星搜索管线路程序. 其中, 文献^[37]测试了CPU超线程对PRESTO运行性能影响, 结果显示禁用超线程能缩短运行时间, 这与高性能计算机“关闭超线程”的默认设置一致. 文献^[38]在CPU上设计了PRESTO并行计算架构, 相比串行版本在单机8核心上实现了1.7倍加速. 文献^[39]将PRESTO的消色散步骤移植到GPU, 相比串行版本和MPI并行版本, 分别获得了约120倍和60倍的加速. 文献^[40]使用OpenCL语言编写程序驱动FPGA完成频域加速搜索操作, 在单位功耗性能上优于GPU和CPU.

本文将介绍如何在CSRC-P的多核心计算节点上, 针对MWA-VCS的三个非相干叠加数据样例, 优化PRESTO (V2.1版本)脉冲星搜索管线的执行速度, 测试科学结果的一致性. 采用此数据进行测试, 一

方面是为了加速持续进行的南天MWA非相干巡天搜寻^[41], 另一方面则可以为MWA的南天快速两米巡天(Southern-sky MWA Rapid Two-metre, SMART)⁹⁾做准备. 本文主要贡献包括:

(1) 优化脉冲星搜索管线. 优化后的管线相比原来的单核心版本, 在x86节点(28个CPU核心)上加速了10.4–12.2倍, 并行效率为37.1%–43.6%; 在ARM节点(96个CPU核心)上加速了24.5–25.8倍, 并行效率为24.5%–26.9%.

(2) 验证ARM计算节点可用性. 本文在x86和ARM计算节点上使用相同的并行优化方法, 展示了优化方法从x86到ARM的可移植性. 由于脉冲星搜索任务的大规模并行性, ARM节点得益于每个节点更多的核心数和内存带宽, 因此比x86节点快1.1–1.3倍, 展现了ARM处理器在SKA数据处理任务中的巨大潜力.

(3) 分析脉冲星搜索管线对文件系统、内存系统的压力. 通过对比优化前后管线各步骤的读写内存和文件系统的峰值速率, 进一步揭示了管线在浮点运算之外的计算资源需求, 为进一步优化管线应用、规划和扩容区域中心原型机提供了依据.

文章结构如下: 第2节介绍本文所采用的脉冲星搜寻管线, 第3节介绍优化方法, 第4节评估管线优化后的性能并剖析管线的资源需求特性, 第5节对比本文与相关研究工作, 第6节总结全文.

2 脉冲星搜索流程

使用PRESTO搜索脉冲星的典型管线如图1所示, 主要由6步组成, 包括:

(1) 消除射频干扰(Remove Narrowband Radio Frequency Interference); (2) 消色散(De-dispersion); (3) 快速傅里叶变换及消除红噪声(FFT & Remove Red-noise); (4) 频域加速搜索(Accelerate Search); (5) 候选体折叠(Candidate Folding); (6) 单脉冲搜索(Single Pulse Search). 后续章节将逐一介绍这些操作, 以及参数设置的注意事项.

7) ARM architecture. https://en.wikipedia.org/wiki/ARM_architecture.

8) <https://indico.skatelescope.org/event/551/>.

9) Bhat N D R, Swainston N A, McSweeney S J, et al. The southern-sky MWA rapid two-metre (SMART) pulsar survey: System overview, pulsar census, and first pulsar discoveries. Publ Astron Soc Aust, 2022 (in preparation).

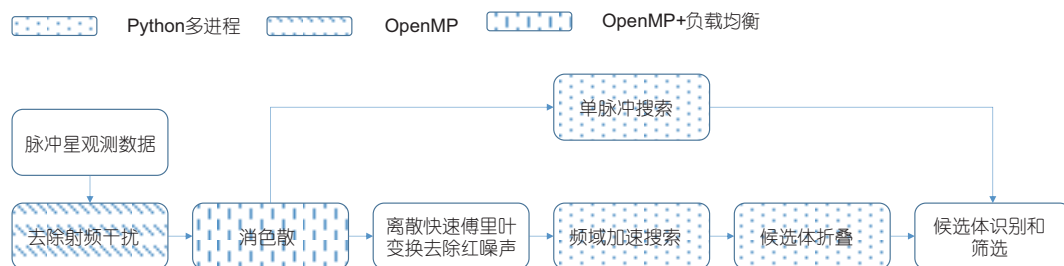


图1 脉冲星搜索管线和分步优化方法

Figure 1 The flowchart of the pulsar search pipeline.

2.1 消除射频干扰

射频干扰主要来源于地面射电源和空间卫星, 干扰信号的强度可以达到真实脉冲星信号的上万倍, 会很大程度影响对于弱脉冲星周期信号的探测, 并大幅增加后续流程的工作量, 从而影响脉冲星信号搜寻的效率.

搜索管线使用PRESTO提供的`rfifind`的程序识别和去除窄带射频干扰和持续时间较短的宽带射频干扰. `rfifind`将时域分为等间隔区间, 用这个区间去扫描整个数据, 搜寻和标记射频干扰, 记录在标记文件`.mask`中. 本文使用的脉冲星搜索流程, 时域区间的间隔设置为12 s.

另一个需要消除的是周期性干扰信号. 由于色散量(Dispersion Measure, DM)为 0 pc cm^{-3} 的信号主要为射频干扰信号, 管线对数据进行一次DM-0搜索, 找出零色散信号, 并记录到`.bird`文件中, 从而消除此类干扰.

2.2 消色散

由于不同频率的电磁波在星际介质中的传播速度存在差异, 因而导致脉冲星信号到达射电望远镜存在时间差, 这一现象称之为脉冲星信号的色散. 不同的脉冲星距离不同, 经过的星际介质不同, 因此色散各异. 色散导致脉冲星信号展宽. 脉冲展宽受射电望远镜观测频率、带宽和脉冲星信号自身色散量共同影响. 观测频率越低、观测带宽越大, 以及脉冲星信号自身的色散量越大, 脉冲展宽越大. 因此需要对观测数据进行消色散处理.

PRESTO所提供的`prepdata`和`prepsubband`程序可以分别对一个色散和一系列的色散进行消色散处理, 我们根据需要选择了后者. `prepsubband`首先去

除“消除射频干扰”步骤标记的干扰信号, 然后根据特定的色散量DM, 在时域上对不同通道组的信号进行延迟处理, 最后将信号在频率轴向相加, 生成固定频宽上的一系列不同色散量的时间序列, 用于后续搜索.

2.3 快速傅里叶变换及消除红噪声

不同色散量的时域数据需要变换到频域, 为下一步“频域加速搜索”做准备. 搜索管线使用`realfft`程序进行快速傅里叶变换(Fast Fourier Transform, FFT), 生成频域数据文件`.fft`. 此外, 由于环境中无处不在的粒子布朗运动, 导致频域信号低频端升高, 混入红噪声. 红噪声主要对周期P超过100 ms的脉冲星造成影响, 导致约30%的长周期脉冲星无法在频域被搜寻到^[42]. 管线使用PRESTO提供的`rednoise_cmd`程序移除信号中的红噪声.

2.4 频域加速搜索

在频域中, 一颗真实脉冲星的信号并非只在一个频率处出现, 而是表现为一个基频和一系列谐波. 为了充分利用谐波, 使用“谐波叠加技术”来加强脉冲星的频域信号, 并获取脉冲星的候选体.

搜索管线使用PRESTO提供的`accelsearch`程序实现频域加速搜索, 该程序可同时用于搜索孤立脉冲星和双星系统中的脉冲星. 其中, `numharm`参数控制最高谐波叠加次数. 理论上, 叠加次数越高对搜索越有利, 但同时运算量也会成倍增加. 在本文中, 叠加次数设置为16. `accelsearch`中的`zmax`参数控制搜索参数空间中由双星系统轨道运动引起的最大频域漂移量, 通过它可以计算出脉冲星的平均加速度 a (主要为双星

轨道加速度), 如下所示:

$$a = zc/(T^2 f_0). \quad (1)$$

其中, c 为光速, T 为观测时长, f_0 为脉冲星的自转频率. $z_{\max}$ 可以设值的范围是 0–1200, 设定值越大, 运算耗时越长. 本文使用的管线将 $z_{\max}$ 设置为 200, 旨在涵盖大部分毫秒脉冲星 8 倍谐波的搜寻¹⁰⁾. 考虑到优化流程的主要目的是脉冲星巡天搜寻, 我们不考虑极度消耗计算资源的 Jerk 搜寻模式, $w_{\max}$ 参数设置为 0.

2.5 候选体折叠

由“频域加速搜索”筛选出的脉冲星候选体, 需要在搜索到的色散量 DM 和周期 P 上, 将其与时间序列折叠, 为进一步分析脉冲轮廓和确定脉冲星类型做准备. 由于设置了最大频域偏移量 ($z_{\max}$) 不为零, 对于某些脉冲星双星候选体而言, 还需要考虑其双星轨道加速度带来的影响. 因此, 搜索管线使用 PRESTO 提供的 `prepfold` 程序折叠脉冲轮廓. 根据候选体列表中的 DM 值, 利用 `accelfile` 参数选择对应的 `.cand` 文件进行折叠. 本流程则对不同候选体进行多进程的折叠操作, 缩短折叠总时长.

2.6 单脉冲搜索

部分脉冲星的脉冲信号不具有稳定性, 难以通过频域搜索方法发现, 需要在消色散处理后的时域数据上搜索单脉冲信号. 比如具有零脉冲的脉冲星 (Nulling Pulsar), 这类脉冲星会在一段时间内没有脉冲信号; 又比如旋转射电暂现源 (Rotating Radio Transient, RRAT), 每隔一段时间才发出一个或几个脉冲, 它们通常采用时域上的单脉冲搜索方法. 值得一提的是, PRESTO 的单脉冲星搜寻也可以用于搜索快速射电暴 (Fast Radio Burst, FRB) 这种尚未被完全理解的天体辐射现象. 搜索管线使用 PRESTO 提供的 `single_pulse_search.py` 脚本, 获得不同色散量时间序列中较明显的单脉冲信号.

3 脉冲星搜索并行优化

3.1 脉冲星搜索管线热点分析与优化策略

搜索管线由“驱动脚本”和“可执行程序”两部分组成. 驱动脚本通常由 Bash 或 Python 脚本语言编写, 用于设置程序参数 (如 CPU 核心数、搜索范围等) 后, 启动 PRESTO 二进制可执行程序完成计算任务. 本文所用搜索管线的驱动程序和可执行程序如表 1 所示, 部分脚本和程序展现了并行优化潜力, 少部分初步实现了并行.

本文在 x86 和 ARM 节点上以单核心运行搜索管线, 绘制各步骤耗时占比. 如图 2(a) 和 (b) 所示, 管线在两种计算节点上具有类似的热点分布. 其中 5 个计算步骤 (消除射频干扰、消色散、加速搜索、候选体折叠以及单脉冲搜索) 运行耗时高, 是剖析优化的重点目标. 而快速傅里叶变换和消除红噪声耗时少, 本文暂不优化.

CSRC-P 提供了 x86 CPU, ARM CPU, GPU 等多种计算资源, 为搜索管线提供了多种可选的优化方法, 包括: (1) OpenMP^[43] 多线程并行: 通过在源代码中添加并行制导语句, 将同一个共享地址空间上的工作任务划分给多个核心并行执行. 研究人员可以从最耗时的循环代码入手, 逐步提高制导语句的覆盖范围, 从而渐进地将串行程序改写为并行程序. OpenMP 已被标准化, GCC, LLVM, Intel 等编译器均支持. (2) Multiprocessing 多进程并行: 任务并行的一种实现形式, 每个进程独立负责一部分计算任务, 任务之间没有依赖. 脚本语言常用的多进程库和工具包括用 Python 的 multiprocessing 库^[44]、GNU Parallel 等. (3) MPI 多节点并行: 基于消息传递模型的多核心多节点并行编程方法, 各进程通过通信传递数据, 独立求解一部分计算问题^[45]. 遵循开放标准实现的 MPI 库有 MPICH, OpenMPI 和 Intel MPI 等. (4) GPU, FPGA 异构并行: 使用 GPU, FPGA 等异构计算资源上的特殊计算单元求解问题的一种并行技术. 利用 GPU 上的大规模并行浮点计算单元、FPGA 上的可定制逻辑门阵列, 可获得比 CPU 更高的单位功耗处理能力^[46]. GPU 和 FPGA 缺少 CPU 上的通用开放可移植编程标准, 从 CPU 向 GPU 和 FPGA 移植程序需要改写程序, 常用的

10) https://github.com/scottransom/presto/blob/master/docs/PRESTO_search_tutorial.pdf.

表 1 针对热点函数的并行优化策略

Table 1 Parallel optimization methods against hotspot functions

步骤	驱动脚本	可执行程序	驱动脚本		可执行程序		优化策略
			编程语言	已并行	编程语言	已并行	
消除射频干扰	rfifind.py	rfifind	Python	否	C	是	OpenMP并行
消色散	prepsubband.py	prepsubband	Python	否	C	是	OpenMP+负载均衡
频域加速搜索	accelsearch.py	accelsearch	Python	否	C	否	多进程并行
候选体折叠	prepfold.py	prepfold	Python	否	C	是	多进程并行
单脉冲搜索	—	single_pulse_search.py	—	—	Python	否	多进程并行

编程工具包括CUDA, ROCm和OpenCL等. (5) 负载均衡: 计算过程中计算核心的负载差异会导致运行时间

差异, 所有核心在同步时必须等待运行最慢的核心, 这将拖慢整体并行效率. 因此在并行优化中, 均衡各核心的任务量, 是非常有必要的.

我们综合考虑了代码特性、任务优先级和技术难度, 决定采用OpenMP多线程、Multiprocessing多进程和负载均衡三种方法, 投入2个人月, 完成第一版优化代码. 异构并行和多节点并行则留到下一阶段考虑. 理由简述如下: 第一, 代码功能完整性. 需要交付一个功能完整、能直接替代原串行版PRESTO使用的并行版本. 相比使用MPI重新实现一个分布式版本, 或者是在GPU, FPGA上从零开始移植全部功能, 在现有PRESTO逐步增加并行代码, 更容易实现“用法不变, 速度提升”的目标. 第二, 任务紧迫性. 我们希望尽快在原型机上完成这一批MWA-VCS数据处理, 与合作方交流前期科学成果以明确下一阶段目标. 从代码特点评估工作量, 实现CPU并行版本是比较可行的方案. 第三, CSRC-P的算力分配. 按浮点能力计算, CSRC-P约80%计算能力来自CPU (含x86和ARM), 剩余20%计算能力来自GPU. 要尽快完成数据处理任务, 需要充分利用CPU部分的算力. 第四, 管线的内存容量需求. 管线运行的内存需求大约在16–256 GB不等, 主流CPU服务器较容易满足这样的需求, 而以NVIDIA V100为代表的主流GPU加速卡显存容量只有32–64 GB, 运行完整管线还需重新设计显存分配方法, 因而增加了优化难度.

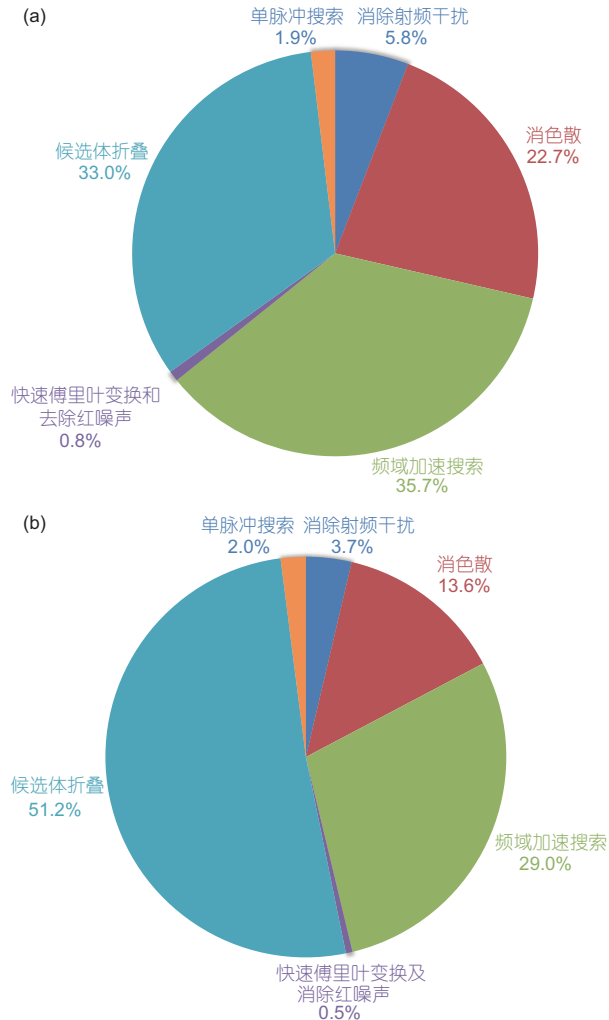


图 2 脉冲星搜索管线样例1在x86节点上的热点分析(a)和在ARM节点上的热点分析(b)

Figure 2 The hotspots of Pulsar Search Pipeline (a) on x86 platform and (b) on ARM platform.

3.2 使用OpenMP优化消除射频干扰

使用OpenMP优化PRESTO的射频干扰消除程序rfifind.

如补充材料中“原始的窄带射频干扰消除”算法所示, 源代码rfifind.c主要完成搜索射频干扰和写入.mask文件这两步操作, 耗时分别占95%和5%. 这

个未优化的算法在进行搜索射频干扰时,是在分割后的时间区间上逐个通道串行执行的,即处理器单个核心处理完第一个通道的数据后,再处理第二个通道的数据.该方法没有利用不同通道的数据独立性,无法发挥处理器的多核性能.

为了利用多核处理器加速搜索射频干扰,在遍历所有通道的for循环前添加了OpenMP制导语句“pragma omp parallel for”,优化后的流程如补充材料中的“优化后的窄带射频干扰消除”算法.制导语句将for循环内的工作量均匀分配给多个不同的线程并行执行.在默认OpenMP策略下,工作量分配方案如下:将for循环的区间按照线程数均分,然后令每个线程负责处理与其id对应的区间.

3.3 使用OpenMP和负载均衡方法优化消色散

本文为PRESTO的消色散程序prepsubband添加OpenMP多线程并行支持,每个消色散子任务使用的OpenMP线程数根据估算的任务量动态分配,以缩短各个子任务执行的时间差异.

消色操作的热点代码位于“prepsubband.c”约1019–1039行,其流程如补充材料中的“原始的消色散流程”算法所示.这段代码首先读取一条没有窄带射频干扰的数据并生成预处理过的数据,然后使用上一条数据与当前数据做消色散.由于第一次运行时不存在上一条数据,因此第一条数据会被使用两次.

为这段代码添加并行支持,需要消除数据依赖.为此,展开热点代码的循环,重新设计计算流程.如补充材料中的“并行后的消色散流程(未做负载均衡)”算法所示,待处理的 M 个通道被均匀分配给 N 个CPU核心,每个CPU核心独立处理各自的 $\frac{M}{N}$ 个通道数据.算法循环分为两层,第一层循环预处理数据,并将它们加载进数组中;第二层循环读取预处理的数据,并进行消色散的操作.

以上并行方法实现了各个CPU核心处理通道数(即任务数)的均衡,但并没有实现处理时间的均衡,会导致整个消色散流程的运行耗时受制于处理最慢的通道批次.图3展示了运行时间不均衡现象:每个核心分配了数量相等的消色散任务,但任务耗时300–1500 s不等.这是因为每个消色散任务的输入参数差

异导致了耗时差异.表2列举了几个典型消色散任务的参数和耗时,可见,Nsteps(需要执行的步数)越大,nsb(消色散中将总带宽均分的组数)越大,任务越耗时.根据现有经验¹¹⁾,由nsb和Nsteps估算消色散任务运行时间,估算法方法如表3所示.可见,对于同一个消色散任务,增加OpenMP线程数能够缩短任务运行时间.

本文基于消色散任务的运行耗时估算,调整每个消色散任务的OpenMP线程数,从而实现处理时间的均衡.负载均衡的OpenMP线程数分配流程如补充材料中的“消色散流程负载均衡线程数分配方法”算法所示,简述如下:

- (1) 输入: 该批次的任务总数 C (通常取CPU核心数的一半或四分之一),每个任务初始OpenMP线程数为1.
- (2) 根据命令耗时的预测值,记 T 为所有任务耗时的最大值.
- (3) 如果当前使用的线程数小于等于节点上CPU核心数,找出最耗时的任务,为该任务线程数加1.否则,合并两个耗时最短的任务,并将线程数设置为原来两个任务中线程数的较大值.
- (4) 更新 T 并与原值进行比较,验证是否满足收敛条件;若满足收敛条件且当前使用的线程数大于等于节点上CPU核心数,则算法结束,并输出结果;否则返回前一步.
- (5) 输出: 该批次执行的任务总数 C 以及每个任务的线程数.

3.4 使用多进程优化频域加速搜索、候选体折叠和单脉冲搜索

频域加速搜索、候选体折叠和单脉冲搜索这三个步骤具有相似的计算模式:使用Python脚本逐个读入、处理、写回数据文件.由于数据文件之间没有依赖关系,可以使用Multiprocessing多任务并行来执行.本文使用Python的multiprocessing库在以上三个步骤中实现多任务并行,优化后的流程如算法A3所示.优化后的Python脚本把所有待处理的文件集平均分成 N 份,每个文件子集分配给一个Python进程在一个CPU核心上处理.在多任务并行模式下,充分利用处理器核心的关

11) https://github.com/NickSwainston/mwa_search/tree/master/scripts/mwa_search/lfdplan.py.

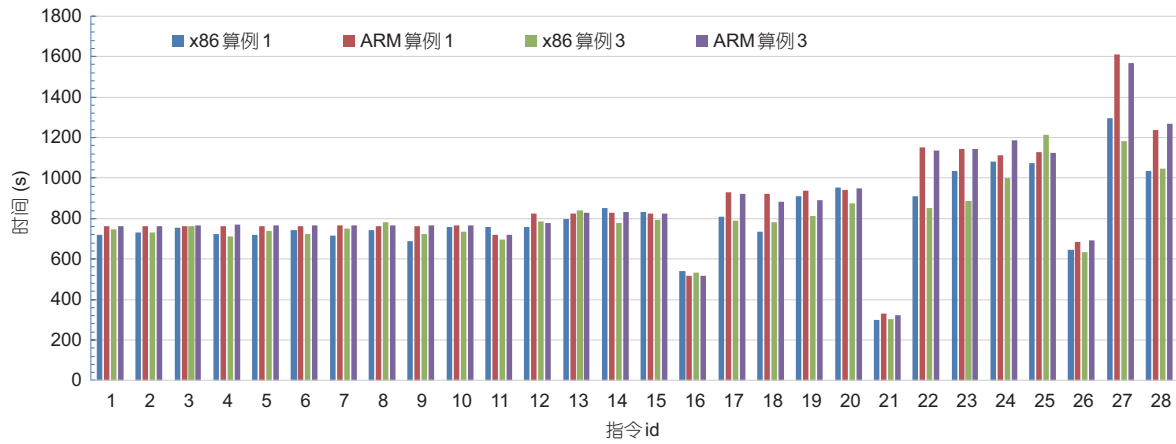


图 3 消色散操作在各核心上负载不均导致运行时间差异

Figure 3 Work imbalance leads to runtime variation on CPU cores of x86 and ARM.

 表 2 低频MWA非相干叠加数据的消色散方案^{a)}

 Table 2 The De-dispersion scheme for MWA incoherent summed data at low frequency^{a)}

Low-DM ($\text{cm}^{-3} \text{ pc}$)	High-DM ($\text{cm}^{-3} \text{ pc}$)	dDM ($\text{cm}^{-3} \text{ pc}$)	Nsteps	Downsamp	nsub	耗时 (s)
1.0	22.9	0.02	1093	1	4	0.1
22.9	45.7	0.05	457	2	8	0.2
45.7	91.5	0.11	415	4	16	0.4
91.5	183.0	0.21	435	8	32	0.8
183.0	250.0	0.43	115	16	64	1.6

a) 消色散方案应随着观测频率、带宽等因素相应调整, 但该步骤的优化方案不变.

表 3 每条prepsubband命令在给定nsub, Nsteps及OpenMP线程数(N)下的预估运行时间

Table 3 The assumed runtime of each prepsubband command given nsub, Nsteps and the number of OpenMP threads

	1线程	N线程($N > 1$)
nsub = 4	$700 \times \frac{N_{\text{step}}}{100}$	$1.25 \times \frac{700 \times \frac{N_{\text{step}}}{100}}{N}$
nsub = 8	$800 \times \frac{N_{\text{step}}}{100}$	$1.25 \times \frac{800 \times \frac{N_{\text{step}}}{100}}{N}$
nsub = 16	$900 \times \frac{N_{\text{step}}}{100}$	$1.25 \times \frac{900 \times \frac{N_{\text{step}}}{100}}{N}$
nsub = 32	$1000 \times \frac{N_{\text{step}}}{100}$	$1.25 \times \frac{1000 \times \frac{N_{\text{step}}}{100}}{N}$
nsub = 64	$1200 \times \frac{N_{\text{step}}}{100}$	$1.25 \times \frac{1200 \times \frac{N_{\text{step}}}{100}}{N}$

键是可并行执行的任务数要大于处理器核心数. 加速搜索和单脉冲星搜索的可并行执行的任务数高达几千, 候选体折叠流程的数据并行度达到一百以上, 这些流程都能充分利用x86和ARM节点上的处理器核心.

4 性能测试

我们使用3个典型脉冲星搜索样例, 在CSRC-

P的x86和ARM计算节点测试优化前后的运行时间, 以加速比和并行效率评估优化效果, 分析优化后对内存和文件系统的压力, 并验证脉冲星搜索结果的一致性, 最后总结优化效果, 简要评价ARM计算节点对科学应用的适用性.

性能测试在CSRC-P的x86和ARM计算节点上进行, 这两种节点的软硬件配置如表4所示, 测试中设置的主要PRESTO参数如表5所示.

测试所用样例如表6所示, 这些样例来自非相干模式下中心频率为185 MHz的MWA-VCS观测. MWA-VCS的非相干模式具有约600平方度的大可视天区, 被广泛用于SMART巡天前期的大面积浅层普查. 在2019年, 文献[41]中就发表了利用50个MWA-VCS观测的非相干叠加数据, 在旁瓣中找到的两颗北天脉冲星.

优化前的管线使用单核心运行; 优化后的管线使用节点所有核心运行, 即x86节点上使用28核心, ARM节点上使用96核心.

表 4 CSRC-P中的x86和ARM计算节点

Table 4 x86 and ARM compute nodes in CSRC-P

	x86节点	ARM节点
处理器型号	2×Xeon Gold 6132	2×Kunpeng 920
处理器核心	2×14核心@2.6 GHz	2×48核心@2.5 GHz
内存通道	2×6通道DDR4-2666	2×8通道DDR4-2666
内存容量	384 GB	512 GB
内存带宽	119.2 GB/s	170.6 GB/s
浮点性能	1536 Gflops	384 Gflops
PCI-E通道数	2×48条Gen3通道	2×48条Gen4通道
PCI-E总带宽	96 GB/s	192 GB/s
操作系统	CentOS 7	CentOS 7
内核版本	3.10.0	4.14.0
Python版本	2.7	2.7
gcc版本	8.3	8.3
PRESTO版本	2.1	2.1

表 5 脉冲星搜索管线的主要参数

Table 5 Parameters in pulsar search pipeline

参数	取值
谐波叠加上限numharm	16
频域最大偏移zmax	200
候选体折叠做图点数	128

加速比、并行效率被用于评估优化效果. 加速比 S 定义如下, 用于衡量优化前后耗时缩短的程度:

$$S = \frac{t_{\text{origin}}}{t_{\text{opt}}}, \quad (2)$$

其中, t_{origin} 是优化前的单核心运行时间, t_{opt} 是优化后多核心运行时间.

并行效率 E 定义如下, 用于衡量并行程序利用多核心的效率(上限为100%), N 是程序使用的核心数:

$$E = \frac{t_{\text{origin}}}{t_{\text{opt}} \times N}. \quad (3)$$

4.1 消除射频干扰优化性能测试

消除射频干扰的优化效果如图4所示, 优化后的多核心版本在x86和ARM节点上, 相比优化前的单核心版本, 分别加速了8.7–13.4倍和10.2–18.8倍. 按照优化后使用全部28和96个计算核心计算, 该步骤优化后在x86和ARM上的并行效率分别达到31.1%–47.6%和10.6%–19.6%.

4.2 消色散优化性能测试

消色散使用了OpenMP并行和负载均衡两种优化方法, 效果如图5所示. 相比优化前的单核心版本, 仅使用OpenMP并行优化后的消色散流程在x86和ARM节

表 6 非相干模式下的MWA-VCS观测脉冲星搜寻测试样例

Table 6 The MWA-VCS observations in incoherent mode for pulsar search tests

Case	ObsID	R.A.(J2000)	Dec.(J2000)	Duration (s)	Pulsar searched
1	1088850560	13:20:07.2000	−26:37:12.0000	3535	–
2	1145367872	11:34:17.2956	−33:25:06.9706	3613	J1116–4122
3	1115381072	10:10:08.1228	+10:39:45.9377	4868	J0953+0755

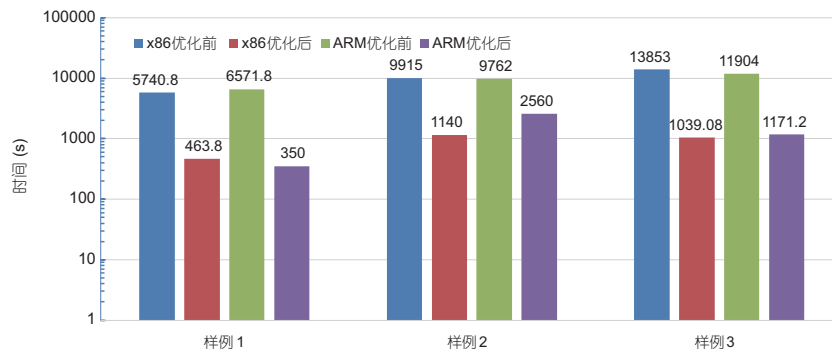


图 4 x86及ARM节点的消除射频干扰优化性能测试

Figure 4 The performance of remove narrow-band RFI on x86 and ARM single nodes.

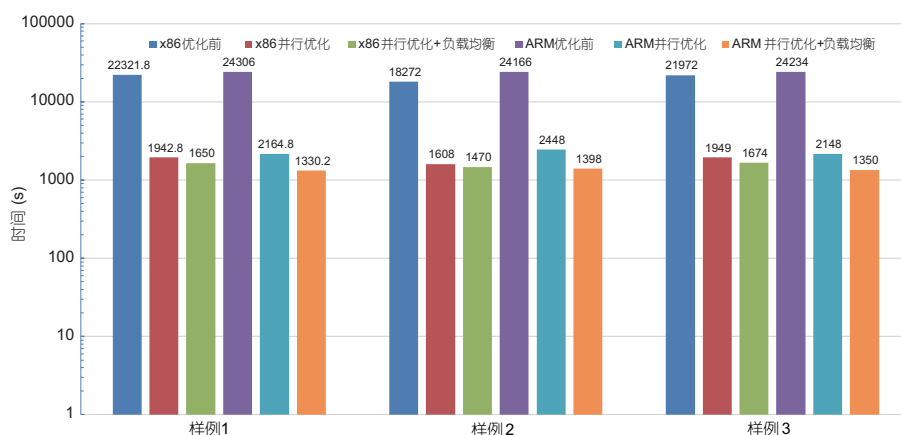


图 5 x86和ARM节点上的消色散优化性能测试

Figure 5 The performance of De-dispersion on x86 and ARM single nodes.

点分别加速了8.7–11.5倍和9.9–11.3倍, 并行效率分别为31.0%–41.0%和10.3%–11.8%. 在OpenMP并行优化基础上加入负载均衡优化, 加速比进一步提高到12.4–13.5倍和17.2–18.3倍, 并行效率则提高到44%–48.2%和17.9%–19.1%.

每个x86单节点有28个核心, 每个批次的消色散任务数设置为14; 每个任务的OpenMP线程是1–4, 由线程分配算法决定. 每个ARM节点有96核心, 每个批次的消色散任务数是28, 每个任务的OpenMP线程数是2–10.

4.3 频域加速搜索、候选体折叠、单脉冲搜索的优化性能测试

频域加速搜索的优化效果如图6所示. 优化后的多核心频域加速搜索流程在x86和ARM节点上, 相比优化前的单核心版本, 分别加速了23.0–25.4倍和67.4–72.7倍, 并行效率分别达到82.0%–91.5%和70.2%–82.9%.

候选体折叠的优化效果如图7所示. 优化后的候选体折叠流程在x86和ARM节点上, 相比优化前的单核心版本, 分别加速了8.6–9.8倍和32.1–41.5倍, 并行效率分别达到30.7%–34.9%和33.4%–43.2%.

单脉冲搜索的优化效果如图8所示. 优化后的单脉冲搜索流程在x86和ARM节点上, 相比优化前的单核心版本, 分别加速了20.8–22.1倍和47.0–82.0倍, 并行效率分别达到74.3%–78.8%和48.9%–85.5%.

4.4 整体优化效果

脉冲星搜索流程的整体优化效果如图9所示, 相比优化前的单核心版本, 整个流程在x86和ARM节点上分别加速了10.4–12.2倍和24.5–25.8倍, 并行效率分别达到37.1%–43.6%和25.5%–26.9%. 图10以样例1展示了两种计算节点上优化前后各步骤的加速比和运行时间. 加速搜索和单脉冲搜索, 优化后的加速比及并行效率优于候选体折叠. ARM节点在加速搜索、候选体折叠、单脉冲搜索这三个步骤的加速比, 是x86节点的3倍以上. ARM节点优化后每个步骤的运行时间均小于x86节点, 整体运行时间也小于后者.

4.5 优化后对内存系统、文件系统的影响

我们使用GNU Perf测量管线优化前后各步骤读写内存和文件系统的峰值速率, 以样例1为例分析应用优化后对计算资源要求的变化.

管线优化前后各步骤读写内存的峰值速率如图11所示, 横线标注了每节点内存带宽上限. 管线优化后, 由单核心变成多核心同时处理数据, 读写内存的峰值速率相应增加. 加速搜索、候选体折叠和单脉冲搜索内存读写速率的增长倍数略高于加速比, 优化后的访存速率不到上限的一半, 内存带宽在这三个步骤有富余. 这三个步骤表现为“计算密集”, 加速比和并行效率较高, 增加并行度和核心数能显著缩短运行时间. 消除射频干扰、消色散两个步骤优化后的访存速率接近上限. 这两个步骤表现为“访存密集”, 加速比和并

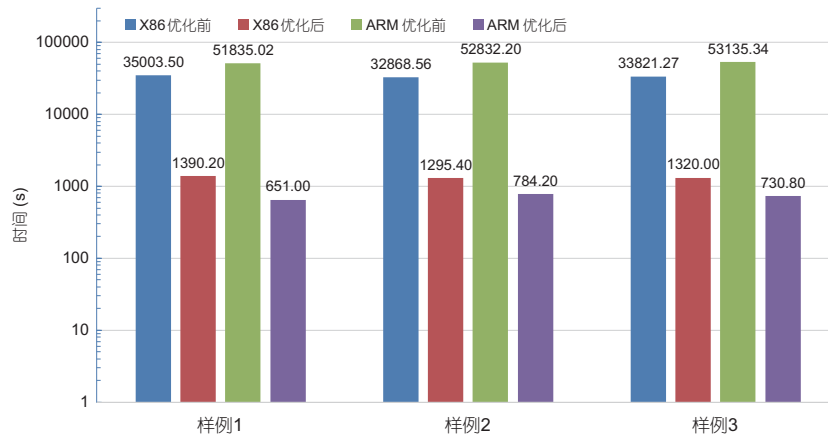


图 6 x86和ARM节点上的加速搜索优化性能测试

Figure 6 The performance of accelerate search on x86 and ARM single nodes.

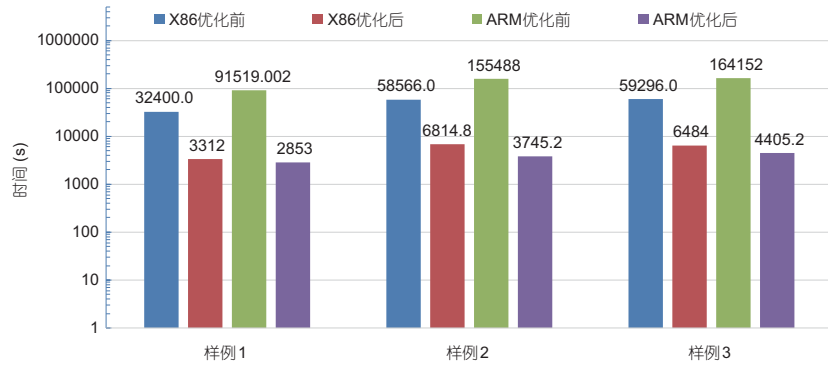


图 7 x86及ARM节点上的候选体折叠优化性能测试

Figure 7 The performance of candidate folding on x86 and ARM single nodes.

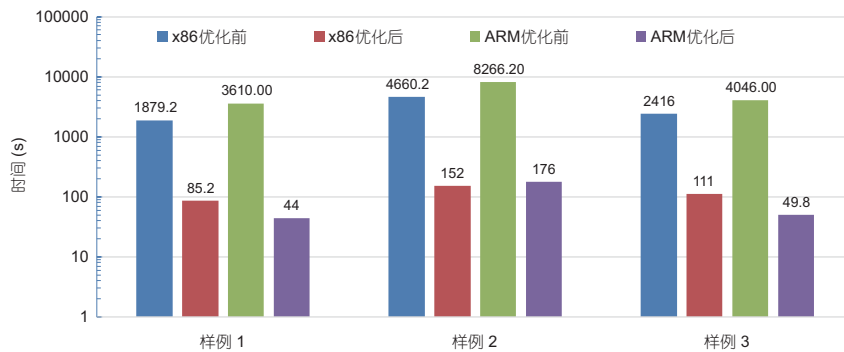


图 8 x86和ARM节点上的单脉冲搜索优化性能测试

Figure 8 The performance of single pulse search on x86 and ARM single nodes.

行效率较低, 增加并行度和核心数不能显著缩短运行时间.

管线优化前后各步骤读写文件系统的峰值速率如图12所示, 横线标注了单节点文件系统的带宽上

限7.6 GB/s. 优化前, 各步骤串行读写文件系统. 优化后, 各步骤读写文件系统的速率增量与读写模式相关, 均未超过文件系统带宽上限. 其中, 消除射频干扰步骤多线程并行读取和写入多个文件, 在x86和ARM计

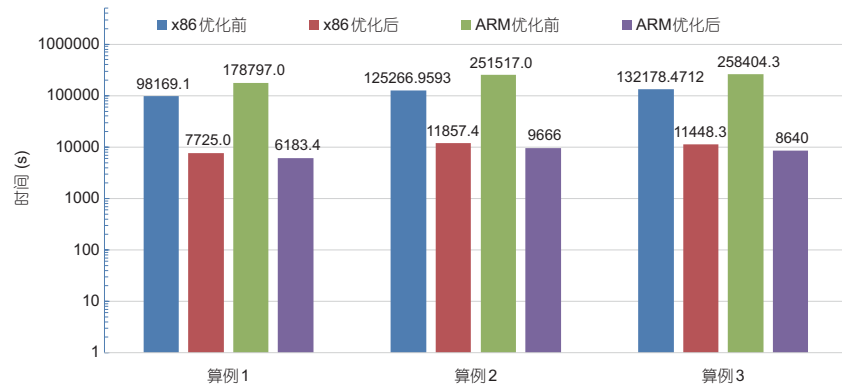


图 9 x86和ARM节点上脉冲星搜索管线的整体优化效果

Figure 9 Overall performance of the whole pipeline on x86 and ARM single nodes.

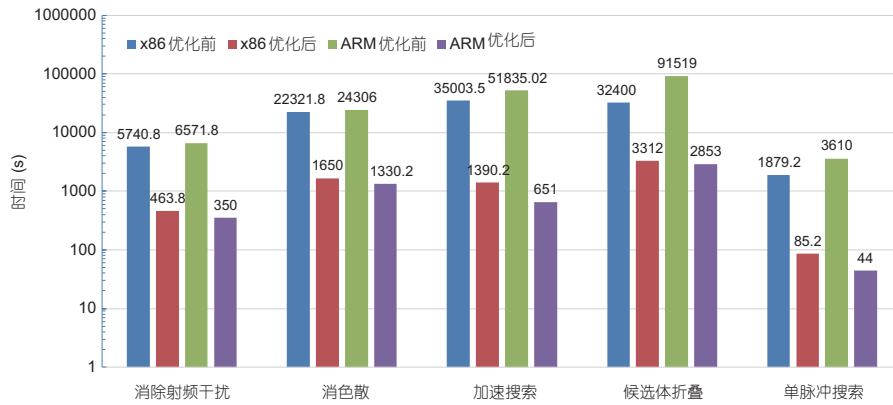


图 10 x86和ARM节点上的各步骤优化效果(以样例1为例)

Figure 10 Performance of individual steps on x86 and ARM single nodes using case No. 1.

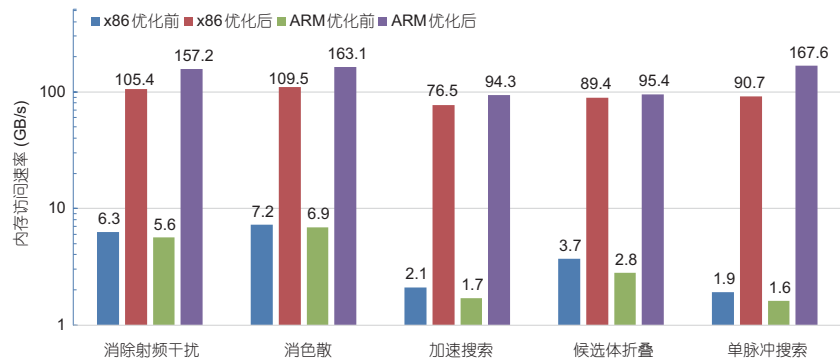


图 11 x86和ARM节点上脉冲星搜索管线的各步骤读写内存的峰值速率(以样例1为例)

Figure 11 Peak memory bandwidth of individual steps on x86 and ARM single nodes using case No. 1.

算节点上读写速率分别增加6.3和7.2倍; 消色散步骤的优化不涉及文件系统读写, 优化后读写速率几乎不变; 频域加速搜索、候选体折叠和单脉冲搜索

步这三个骤优化后, 会以多进程并行读取和写入多个文件, 在x86和ARM计算节点上读写速率分别提高了19.7和22.5倍、19.7和22.1倍、11.8和12.7倍。

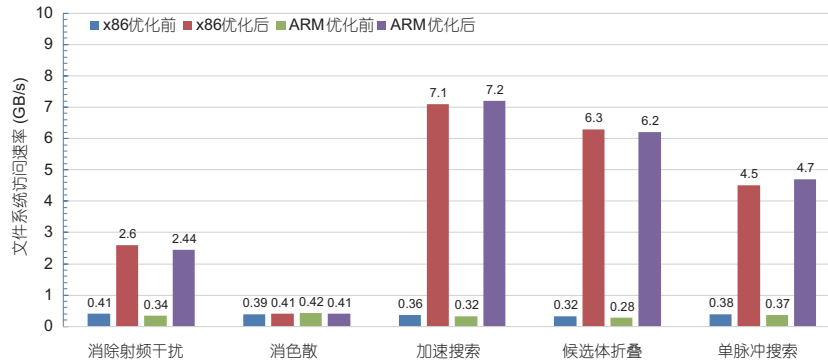


图 12 x86和ARM节点上脉冲星搜索管线的各步骤读写文件系统的峰值速率(以样例1为例)

Figure 12 Peak file system bandwidth of individual steps on x86 and ARM single nodes using case No. 1.

4.6 ARM计算节点对科学应用的适应性

本文基于两种计算节点在脉冲星搜索任务中的表现和其他文献中的测试结果, 结合部署CSRP-C平台的经验, 初步归纳ARM处理器对科学应用的适应程度。

在基础生态环境方面, ARM节点易用程度与x86节点相当. 本次实验使用的CSRC-P x86和ARM计算节点, 在节点体积、额定功耗、售价上接近. 本文在ARM节点上部署了操作系统、KVM虚拟化工具、编译器、作业调度系统、Lustre并行文件系统、MPI并行库、常用服务组件(网页服务器、网页代理、域名解析服务器等)等, 安装和使用方法与x86一致. ARM可以作为替换x86服务器的备选方案之一。

微基准测试和科学应用测试表明, 在整数运算、大内存带宽的场景中, ARM节点表现优于同代x86节点; 而在浮点计算场景中, x86优于ARM. 文献[47]使用GTC-P科学应用对比了x86和ARM处理器, 在名为charge的访存密集步骤中, ARM优于x86; 而在名为push的计算密集步骤中, x86优于ARM. 此外, ARM先于x86支持新一代PCI-E Gen4标准, 可扩容安装200G IB高速网卡, 扩容后文件系统带宽翻倍, 能更好地支持SKA等数据密集应用。

ARM用于科学计算的主要瓶颈是缺少高质量的商用科学工具与计算库, 而其开源替代产品的优化程度也和x86有差距. 科学计算工具MATLAB、集成开发环境Intel Parallel Studio、核心数学库MKL等软件尚未有功能完备的替代产品供ARM使用. 本文使用的开源编译器套件GCC, 其OpenMP库在ARM上创建销毁线程的时间开销比x86更大. 我们初步推测这个库尚未

对ARM微架构做充分优化, 并将进一步分析两种处理器在OpenMP并行效率上的差异。

4.7 一致性验证

本文比较了优化前后脉冲星搜索管线生成的脉冲星候选体信息列表candslist.txt, 对比了候选体数量、色散和周期. 优化前后的结果完全一致, 这说明优化后的代码能够替代原始版本, 用于SKA脉冲星搜索任务中。

5 相关工作

文献[38]设计了任务并行的分布式脉冲星搜索架构, 在单机8核心上实现了1.7倍加速, 并行效率约21%. 得益于对搜索流程函数一级的代码优化, 本文在28核心上实现了10.4倍加速, 并行效率约37%. 我们后续将借鉴该工作增加新一级并行. 文献[39]在GPU上优化搜索流程, 消色散步骤获得了约120倍加速, 优于本文在该步骤获得的13.5–18.3倍加速. 下一步, 计划使用OpenACC编译器, 添加制导语句, 渐进地把关键计算步骤移植到GPU, 进一步缩短运行时间. 文献[40]使用FPGA芯片搜索脉冲星, 完成任务所需功耗低于CPU和GPU设备. FPGA芯片需要使用OpenCL语言重新编写程序, 与CPU协同工作的模式仍处于探索阶段, 更适合于低功耗或实时性要求高的使用场景. 我们也将继续关注FPGA的技术进展, 评估观测站点等功耗受限场景下, 计算设备和优化方案的选型。

6 总结

本文使用OpenMP多线程和多进程并行方法,将原本单核心运行的脉冲星搜索程序移植到多核平台.并且设计了一个负载均衡算法,解决了消色散流程中的负载不平衡问题,进一步缩短了运行时间.本文针对多核架构的优化方法对x86和ARM节点均适用.我们选择MWA-VCS在中心频率为185 MHz上的三个观测的非相干叠加数据进行了性能以及一致性测试,相比优化前的单核心代码,优化后的代码在x86和ARM节点上的运行时间分别缩短了10.4–12.2倍和24.5–25.8倍,并行效率分别为37.1%–43.6%和24.5%–26.9%,且运行结果与优化前一致.

并行优化后的脉冲星搜索管线已经部署在CSRC-P供国内外SKA研究人员使用,这将有效促进脉冲星研究的发展.同时,数据处理管线的性能优化不仅解决了脉冲星搜索管线的耗时问题,还可以扩展到其他应用管线的性能优化,对于中国SKA区域中心的天文研究有很大帮助 [41, 48–54].

近几年内,我们的工作将集中在PRESTO管线的向量化优化、多机并行和GPU移植上,将大量使用MWA-VCS的SMART巡天波束合成数据进行更大规模的实验,以验证ARM, GPU等新型处理器的适用性.对于即将到来的SKA1数据,我们将做好充分的准备,迎接脉冲星搜寻的挑战.届时, CSRC-P平台将发挥更重要的作用.

致谢 本研究使用了中国SKA区域中心原型机的资源.

补充材料

本文的补充材料见网络版physcn.scichina.com. 补充材料为作者提供的原始数据,作者对其学术质量和内容负责.

参考文献

- Hewish A, Bell S J, Pilkington J D H, et al. Observation of a rapidly pulsating radio source. *Nature*, 1968, 217: 709–713
- Sturrock P A. A model of pulsars. *Astrophys J*, 1971, 164: 529
- Manchester R N, Hobbs G B, Teoh A, et al. The Australia telescope national facility pulsar catalogue. *Astron J*, 2005, 129: 1993–2006
- Cordes J M, Freire P C C, Lorimer D R, et al. Arecibo pulsar survey using ALFA. I. Survey strategy and first discoveries. *Astrophys J*, 2006, 637: 446–455
- Keith M J, Jameson A, van Straten W, et al. The high time resolution universe pulsar survey—I. System configuration and initial discoveries. *Mon Not R Astron Soc*, 2010, 409: 619–627
- Barr E D, Champion D J, Kramer M, et al. The northern high time resolution universe pulsar survey—I. Setup and initial discoveries. *Mon Not R Astron Soc*, 2013, 435: 2234–2245
- Nan R, Li D I, Jin C, et al. The Five-hundred-meter Aperture Spherical radio Telescope (FAST) project. *Int J Mod Phys D*, 2011, 20: 989–1024
- Jiang P, Tang N Y, Hou L G, et al. The fundamental performance of FAST with 19-beam receiver at L band. *Res Astron Astrophys*, 2020, 20: 064
- Qian L, Yao R, Sun J, et al. FAST: Its scientific achievements and prospects. *Innovation*, 2020, 1: 100053
- Qian L, Pan Z C, Li D, et al. The first pulsar discovered by FAST. *Sci China-Phys Mech Astron*, 2019, 62: 959508
- Cameron A D, Li D, Hobbs G, et al. An in-depth investigation of 11 pulsars discovered by FAST. *Mon Not R Astron Soc*, 2020, 495: 3515–3530
- Han J L, Wang C, Wang P F, et al. The FAST galactic plane pulsar snapshot survey: I. Project design and pulsar discoveries? *Res Astron Astrophys*, 2021, 21: 107
- Li D, Wang P, Qian L, et al. FAST in space: Considerations for a multibeam, multipurpose survey using China's 500-m aperture spherical radio telescope (FAST). *IEEE Microw*, 2018, 19: 112–119
- Leibowitz E. The orbit and nature of the semidetached stellar companion of the Sgr A* supermassive black hole. *Astrophys J*, 2021, 915: 2
- Stovall K, Lynch R S, Ransom S M, et al. The green bank northern celestial cap pulsar survey. I. Survey description, data analysis, and initial results. *Astrophys J*, 2014, 791: 67
- Deneva J S, Stovall K, McLaughlin M A, et al. New discoveries from the arecibo 327 MHz drift pulsar survey radio transient search. *Astrophys J*, 2016, 821: 10

- 17 van Haarlem M P, Wise M W, Gunst A W, et al. LOFAR: The low-frequency array. *Astron Astrophys*, 2013, 556: A2
- 18 Sanidas S, Cooper S, Bassa C G, et al. The LOFAR tied-array all-sky survey (LOTAAS): Survey overview and initial pulsar discoveries. *Astron Astrophys*, 2019, 626: A104
- 19 Tan C M, Bassa C G, Cooper S, et al. The LOFAR tied-array all-sky survey: Timing of 21 pulsars including the first binary pulsar discovered with LOFAR. *Mon Not R Astron Soc*, 2020, 492: 5878–5896
- 20 Dewdney P E, Hall P J, Schilizzi R T, et al. The Square Kilometre Array. *Proc IEEE*, 2009, 97: 1482–1496
- 21 An T. Science opportunities and challenges associated with SKA big data. *Sci China-Phys Mech Astron*, 2019, 62: 989531
- 22 Keane E, Bhattacharyya B, Kramer M, et al. A cosmic census of radio pulsars with the SKA. In: *Proceedings of Advancing Astrophysics with the Square Kilometre Array*. Giardini Naxos, 2014
- 23 Tingay S J, Goeke R, Bowman J D, et al. The murchison widefield array: The Square Kilometre Array precursor at low radio frequencies. *Publ Astron Soc Aust*, 2013, 30: e007
- 24 Tremblay S E, Ord S M, Bhat N D R, et al. The high time and frequency resolution capabilities of the murchison widefield array. *Publ Astron Soc Aust*, 2015, 32: e005
- 25 Dewdney P E, Turner W, Millenaar R, et al. SKA1: Design Baseline Description. SKA-TEL-SKO-0001075-v1, Internal SKA Document, 2019
- 26 Beardsley A P, Johnston-Hollitt M, Trott C M, et al. Science with the murchison widefield array: Phase I results and phase II opportunities. *Publ Astron Soc Aust*, 2019, 36: e050
- 27 Yamamura S, Akizuki Y, Sekiguchi H, et al. A64FX: 52-core processor designed for the 442PetaFLOPS supercomputer Fugaku. In: *Proceedings of the IEEE International Solid-State Circuits Conference*. San Francisco, 2022. 352–354
- 28 Quinn P, van Haarlem M, An T, et al. SKA Regional Centres—A White Paper by the SKA Regional Centre Steering Committee. V1.0, Internal SKA Document, 2020
- 29 An T, Wu X P, Hong X. SKA data take centre stage in China. *Nat Astron*, 2019, 3: 1030
- 30 An T, Wu X, Lao B Q, et al. Status and progress of China SKA Regional Centre prototype. *Sci China-Phys Mech Astron*, 2022, 65: 129501
- 31 Lao B Q, Zhang Y K, An T, et al. Software platform on China SKA Regional Center prototype system (in Chinese). *Sci Sin-Phys Mech Astron*, 2023, 53: 229507 [劳保强, 张迎康, 安涛, 等. 中国SKA区域中心原型系统——软件平台. 中国科学: 物理学 力学 天文学, 2023, 53: 229507]
- 32 Wei J W, Zhang C F, Lao B Q, et al. Optimization of parallel processing of the Square Kilometre Array low-frequency imaging pipeline (in Chinese). *Sci Sin-Phys Mech Astron*, 2023, 53: 229503 [韦建文, 张晨飞, 劳保强, 等. SKA低频成像管线并行优化. 中国科学: 物理学 力学 天文学, 2023, 53: 229503]
- 33 Ransom S M. New Search Techniques for Binary Pulsars. Dissertation for Doctoral Degree. Cambridge: Harvard University, 2001
- 34 Ransom S M. PRESTO: Pulsar exploration and search toolkit. Astrophysics Source Code Library, 2011
- 35 Lorimer D R. SIGPROC: Pulsar signal processing programs. Astrophysics Source Code Library, 2011
- 36 Barr E. Peasoup: C++/CUDA GPU pulsar searching library. Astrophysics Source Code Library, 2020
- 37 Yu Q Y, Pan Z C, Qian L, et al. A PRESTO-based parallel pulsar search pipeline used for FAST drift scan data. *Res Astron Astrophys*, 2020, 20: 091
- 38 Zhang X. Pulsar Search Acceleration Based on Distributed Computing and Research and Implementation of Feature Extraction and Identification of Pulsar Candidates (in Chinese). Dissertation for Master's Degree. Guiyang: Guizhou Normal University, 2018 [张翔. 基于分布式计算的脉冲星搜索加速及脉冲星候选体特征提取和识别方法研究与实现. 硕士学位论文. 贵阳: 贵州师范大学, 2018]
- 39 You S P, Wang P, Yu X H, et al. A GPU based single-pulse search pipeline (GSP) with database and its application to the commensal radio astronomy FAST survey (CRAFTS). *Res Astron Astrophys*, 2021, 21: 314
- 40 Wang H, Thiagaraj P, Sinnen O. FPGA-based acceleration of FT convolution for pulsar search using OpenCL. *ACM Trans Reconfig Technol Syst*, 2018, 11: 1–25
- 41 Gong H Y, Zhang Z L, Xue M Y, et al. Search and detection of northern pulsars in the side lobes of the murchison wide-field array (in Chinese). *Sci Sin-Phys Mech Astron*, 2020, 50: 109501 [龚宏宇, 张仲莉, 薛梦瑶, 等. 利用默奇森大视场阵列旁瓣探寻北天脉冲星. 中国科学: 物理学 力学 天文学, 2020, 50: 109501]
- 42 Lazarus P, Brazier A, Hessels J W T, et al. Arecibo pulsar survey using alfa. IV. Mock spectrometer data analysis, survey sensitivity, and the discovery of 40 pulsars. *Astrophys J*, 2015, 812: 81
- 43 Chandra R, Menon R, Dagum L, et al. *Parallel Programming in OpenMP*. Burlington: Morgan Kaufmann, 2001
- 44 Palach J. *Parallel Programming with Python*. Birmingham: Packt Publishing Ltd., 2014
- 45 Gabriel E, Fagg G E, Bosilca G, et al. Open MPI: Goals, concept, and design of a next generation MPI implementation. In: *Proceedings of the European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting*. Budapest, 2004. 97–104

- 46 Vestias M, Neto H. Trends of CPU, GPU and FPGA for high-performance computing. In: Proceedings of the 24th International Conference on Field Programmable Logic and Applications (FPL). Munich, 2014. 1–6
- 47 Wang Y C, Chen J K, Li B R, et al. An empirical study of hpc workloads on huawei kunpeng 916 processor. In: Proceedings of the 25th International Conference on Parallel and Distributed Systems. Tianjin, 2019. 360–367
- 48 Lao B Q, Tao A, Ang Y, et al. Research on parallel algorithms for uv-faceting imaging. *Chin Astron Astrophys*, 2019, 43: 424–443
- 49 Lao B Q, An T, Yu A, et al. Parallel implementation of w-projection wide-field imaging. *Sci Bull*, 2019, 64: 586–594
- 50 An T, Mohan P, Zhang Y, et al. Evolving parsec-scale radio structure in the most distant blazar known. *Nat Commun*, 2020, 11: 143
- 51 Gu J, Wang J. Direct parameter inference from global EoR signal with Bayesian statistics. *Mon Not R Astron Soc*, 2020, 492: 4080–4096
- 52 Wang R, Wicenec A, An T. SKA shakes hands with summit. *Sci Bull*, 2020, 65: 337–339
- 53 Wang Y, Tuntsov A, Murphy T, et al. ASKAP observations of multiple rapid scintillators reveal a degrees-long plasma filament. arXiv: [2101.06048](https://arxiv.org/abs/2101.06048)
- 54 Lao B Q, An T, Wang A, et al. Artificial intelligence for celestial object census: the latest technology meets the oldest science. *Sci Bull*, 2021, 66: 2145–2147

Parallel optimization of the pulsar search pipeline

WEI JianWen¹, ZHANG ChenFei¹, ZHANG ZhongLi^{2,3*}, YU Ting^{2,4}, LIN James¹ & AN Tao²

¹Network & Information Center, Shanghai Jiao Tong University, Shanghai 200240, China;

²Key Laboratory of Radio Astronomy, Shanghai Astronomical Observatory, Chinese Academy of Sciences, Shanghai 200030, China;

³Shanghai Frontiers Science Center of Gravitational Wave Detection, Shanghai 200240, China;

⁴School of Astronomy and Space Science, University of Chinese Academy of Sciences, Beijing 100049, China

The connection between astronomy and high-performance computing is becoming stronger with the development of innovative observing facilities such as the Square Kilometre Array (SKA) and the proposed innovative platform for big data and high-performance computing. Astronomical computation is characterized by a large data volume and massive parallelism, particularly for pulsar search, a leading scientific direction of the SKA. In this study, we present an approach for accelerating the pulsar search pipeline based on OpenMP and multiprocessing techniques. We propose a method for solving the load imbalance problem and have successfully installed the pipeline on x86 and ARM compute nodes on the China SKA Regional Centre prototype (CSRC-P). The performance evaluation from tests on the Murchison Widefield Array (MWA) VCS observations shows that our optimization method works well on the x86 and ARM nodes, improving the relative speedup by factors of 10.4–12.2 and 24.5–25.8, respectively, compared with the original single-thread approach. The ARM platform was 1.1–1.3 times faster than the x86 platform in the tested cases, showing its great potential for SKA data processing. This optimized pulsar search pipeline deployed on the CSRC-P will be used for the pulsar survey of the southern-sky MWA rapid two-meter program for various scientific goals, including pulsar timing arrays for gravitational wave detection.

Square Kilometre Array, pulsar, pulsar search, high performance computing, parallel optimization

PACS: 07.05.-t, 07.05.Bx, 07.05.Kf, 95.85.-e, 95.85.Bh

doi: [10.1360/SSPMA-2022-0264](https://doi.org/10.1360/SSPMA-2022-0264)