

射电天文测量集第四版数据模型分析及对我国参与SKA的影响*

徐亦骏^{1,2} 王 锋^{1,2†}

(1 广州大学天体物理中心 广州 510006)

(2 广州大学国家天文科学数据中心大湾区分中心 广州 510006)

摘要 平方公里阵列射电望远镜(Square Kilometre Array, SKA)已进入建设与调试阶段,但海量数据的高效存储与管理仍是SKA科学数据处理和SKA区域中心(SKA Regional Center, SRC)建设中亟待解决的关键问题.系统调研了测量集(Measurement Set, MS)技术各个阶段的发展历程,各版本的技术特性,特别是在SKA应用中的局限性.重点分析了新一代测量集版本4(MSV4)所带来的技术创新及其对SKA数据管理和科学研究的潜在影响.分析表明,MSV4通过引入自描述、模块化处理和高效存储等特性,有望有效缓解SKA数据存储与处理的瓶颈问题,这一新的数据模型有望应用于下一代射电望远镜,对整个射电天文产生重要影响.现阶段应密切关注、掌握MSV4的技术发展趋势,提前布局相关技术储备,为开发、升级改造SKA区域中心科学应用管线和数据处理软件做好充分准备,以确保中国SKA科学研究的顺利开展并取得科学回报.

关键词 天文数据库, 仪器: 干涉仪, 方法: 数值, 数据分析

中图分类号: P164; **文献标识码**: A

1 引言

平方公里阵列射电望远镜(Square Kilometre Array, SKA)项目是一项由多国合作的大型国际科研项目,旨在建造世界上最为灵敏的射电望远镜阵列^[1].SKA将致力于回答宇宙最基本的重大科学问题,特别是关于第1代天体如何形成、星系形成与演化、暗能量性质、宇宙磁场、引力本质、生命分子与地外文明等.SKA项目由SKA天文台(SKA Observatory, SKAO)负责建设与运营.项目分为两个阶段,目前正在进行的是第1阶段,即SKA1,其中包括SKA1-Mid和SKA1-Low两个部分.SKA1-Mid位于南非,由197个碟形天线组成,

其中64个来自MeerKAT先导望远镜.SKA1-Low位于澳大利亚,由超过13万个低频偶极天线组成.SKA的建设已经进入SKA1 AA0.5阶段的调试,预期在2027年后进入科学观测阶段.

数据处理是SKA项目建设的巨大挑战之一.SKA项目每年产生的科学数据将超过700 PB^①.存储、管理并处理这些数据一直是科学家关注和研究的重点.在射电天文数据处理中,高效且准确的数据存储与高速访问方法是科学家开展研究并推动新科学发现的关键基础技术.

为解决射电天文数据的存储与管理问题,有多种通用数据存储模型被提出并得到应用.测量

2025-02-17收到原稿, 2025-04-07收到修改稿

*国家自然科学基金项目(12433012、12373097), 国家SKA专项(2020SKA0110300)资助

† fengwang@gzhu.edu.cn

① <https://www.skao.int/en/explore/big-data>

集数据(Measurement Set, MS)是射电干涉测量数据的常用存储格式^[2-3]. MS广泛应用于CASA (Common Astronomy Software Applications)等数据处理软件中^[4], 当前成熟的MS版本是第2版. FITS (Flexible Image Transport System)是一种通用的天文数据存储格式, 支持图像、光谱和表格数据的存储^[5]. 层次型数据格式第5代(Hierarchical Data Format 5, HDF5)是一种高效的数据存储格式, 适用于大规模数据的存储和管理, 近年来在射电天文学中逐渐得到应用^[6-7]. 此外, 还有根据

具体的望远镜出现的一些专有格式类型, 如阿塔卡马大型毫米波/亚毫米波阵列(Atacama Large Millimeter/submillimeter Array, ALMA)中使用的一种数据存储格式ASDM (ALMA Science Data Model), 专门用于存储原始观测数据和相关元数据^[8]. 为了深入分析和对比各种数据存储模型的特点, 我们从多个方面对比了测量集(MSV2)、传统的FITS格式以及在低频射电阵列(Low-Frequency Array, LOFAR)计划中曾被采用过的HDF5技术. 具体的对比明细见表1.

表 1 MSV2、FITS和HDF5数据模型比较
Table 1 Comparison of MSV2, FITS, and HDF5 data models

Feature	MSV2	FITS	HDF5
Data Structure	Table structure, includes multiple subtables, supports complex arrays	Simple, table-based	Hierarchical, supports complex arrays and groups
Data Storage Efficiency	Efficient, supports complex array storage	Less efficient, text and table-based	Efficient, supports binary storage
Data Processing Flexibility	Versatile, supports various data types and complex queries	Limited, basic table operations	Versatile, supports complex data structures and multiple data types
Compatibility	Integrated with Casacore, CASA, AIPS++, widely used in radio astronomy	Widely compatible, supported by almost all astronomy software	Integrated with various scientific computing tools, such as Python's h5py and MATLAB
Community Support	Strong, widely used in the radio astronomy community	Extremely strong, as the standard format for astronomical data	Strong, widely used in scientific computing
Scalability	Highly scalable, supports various data types and complex queries	Limited scalability, mainly supports basic table data	Highly scalable, supports various data types and complex structures
Data Integrity	High, supports complete metadata and data validation	Low, mainly relies on user-provided metadata, weak data validation	High, supports rich metadata and data validation
Data Access Speed	High, supports efficient indexing and queries	Low, text and table-based, slower access speed	High, supports efficient indexing and random access
File Size	Larger, supports complex arrays and metadata	Smaller, text and table-based	Larger, supports binary storage and complex data structures
Applicable Scenarios	Large-scale radio astronomy data processing and analysis	Small-scale astronomical data storage and exchange	Large-scale scientific data storage and processing

从对比中可以看出, FITS格式的元数据管理方式较为简单, 难以满足射电干涉测量数据的复杂需求. 对于射电天文学中常见的多维数据(如频谱、时间序列、空间分布)的扩展性不足, 难以有效表达复杂的数据关系. FITS格式虽然在天文学中广泛应用, 但其设计初衷更倾向于存储图像和光谱数据. FITS格式在处理高维度数据时效率较低. 射电干涉测量数据通常包含大量的可见度数据, 这些数据的维度包括时间、基线、频率和极化等. 以表格化方式存储的FITS格式数据在这种高维度数据的处理过程中, 会存在存储和访问效率较低的问题. 尤其是在处理大规模数据时, FITS格式数据的性能瓶颈尤为明显. 为了让FITS格式可以应用于射电干涉领域, 先后有UVFITS^[9]或FITS-IDI (FITS Interferometry Data Interchange)^[10]格式被提出, 但这些改进的FITS格式并没有彻底解决射电干涉数据存储的问题. 此外, 由于FITS文件是采用一个基于文本的文件头再加上二进制数据存储的单一文件存储方法, 并未充分考虑分布式计算环境的需求, 难以支持高效的并行处理和动态扩展.

HDF最初的设计目标是提供科学数据的“图形格式”标准. HDF数据格式具有高性能, 并通过透明压缩减小资源占用. HDF由非盈利组织开发维护. 从第4代开始在各学科尤其是天体物理领域流行起来. 由美国国家航空航天局(National Aeronautics and Space Administration, NASA)选定, 很多天文望远镜数据都通过HDF格式存储. HDF5格式虽然功能强大, 在LOFAR低频阵列中也有过试用, 但在射电天文学领域, 其生态系统支持不如测量集完善, 缺少针对射电数据的特殊性开展相应的设计, 也缺少了优化可能性.

总体来看, SKA、下一代甚大阵(Next Generation Very Large Array, ngVLA)^[11]等射电干涉阵数据的维度复杂. 可见度数据的维度包括时间、基线、频率和极化. 面对这样的数据存储要求, 测量集技术是相对可行的方法. 测量集自身就是针对射电天文数据存储提出的, 有非常明确的应用针对性. 因此, 基于测量集技术来开发下一代射电

望远镜的数据管理就成为一个非常合理的选择. 2024年底美国国家射电天文台、欧洲南方天文台、日本国家天文台和平方公里阵列天文台(SKAO)共同提出了测量集版本4 (Measurement Set Version 4, MSV4)数据模型^②, 目前已经进入了社区评估阶段. 尽快地深入分析、理解、掌握这一数据模型, 是开展SKA科学数据处理 (Science Data Processor, SDP)系统的研制和SKA区域中心 (SKA Regional Centres, SRC)项目建设至关重要的环节之一. 对我国科学家充分利用未来SKAO科学数据开展科学研究有重要的意义.

本文立足于对测量集数据模型的技术发展过程进行分析, 说明了MSV4数据模型推出的必要性及关键技术要点. 第2节, 介绍了MS数据模型的基本概念、技术发展与局限性. 第3节重点分析了MSV4数据模型的基本设计思想, 设计目标和相应的关键技术. 第4节讨论了在SKA数据处理中MSV4的应用, 对MSV4在SKA的SDP和SRC建设中的可用性进行了分析与展望. 最后一节进行了总结.

2 测量集早期版本技术发展及其局限性分析

测量集数据模型是一种专门为射电干涉测量数据设计的数据存储和管理标准, 旨在高效地存储和处理射电望远镜阵列生成的复杂观测数据. 其核心设计基于现代数据库技术和分层存储结构, 能够灵活地组织和管理可见度数据(Visibility)及其相关元数据. 测量集数据模型为射电干涉测量数据的存储、管理和分析提供了标准化和高效的解决方案. 在实际应用中, MSV1被提出后, 迅速成为射电天文数据存储与处理的标准, 并广泛地得到了射电天文主流软件的支持.

2.1 测量集V1和V2版本的核心技术分析

1996年, Wieringa和Cornwell提出测量集数据模型1.0版本^[2], 为射电数据的存储提供了一种通用的、适用于不同望远镜和项目的数据格式, 促进数据共享和协作研究. 在数据存储上, 采用表格结

^②Overview - xradio documentation. <https://xradio.readthedocs.io>.

构存储数据,支持快速查询和访问,并优化数据存储布局,减少冗余.同时,支持多波段、多偏振和复杂校准数据的存储,先后经过了不同版本的优化与改进,如测量集数据模型的2.0版本^[3],3.0 β 版本^[12].MSV2被当前主流的射电天文数据处理软件,如CASA^[13]、Astronomical Image-Processing System (AIPS++)^[14]等广泛支持.

MS文件的结构可以看作是一个分层的数据库,其中主表(MAIN表)是核心,其他表格如天线表、频谱表、观测日志表等通过外键与主表关联.

1、主表(MAIN Table)

主表是MS文件的核心,存储了所有与观测数据相关的信息.每一条记录对应一个时间点、一条基线和一个频率通道的可见度数据.主表的关键列包括(为与主表一致,列名都采用大写,粗体是强调这一单词表示主表中关键列的名字):

- **TIME**: 观测时间戳,通常以简化儒略日(Modified Julian Date, MJD)格式存储.
- **ANTENNA1**和**ANTENNA2**: 基线对应的两个天线索引,用于标识参与干涉的天线对.
- **FREQ**: 频率通道索引,指向SPECTRAL_WINDOW表中的具体频率值.
- **DATA**: 复数形式的可见度数据,通常以二维数组存储(极化 \times 频率通道).
- **FLAG**: 数据标志,用于标记无效或受干扰的数据点.

2、辅助表格

辅助表格用于存储与观测相关的元数据,例如天线位置、频率通道信息和观测日志等.这些表格通过外键与主表关联,形成一个完整的数据存储体系.例如:

- **ANTENNA表**: 存储每个天线的名称、位置和偏移信息.
- **SPECTRAL_WINDOW表**: 存储频率通道的具体值和宽度.
- **OBSERVATION表**: 存储观测的基本信息,如望远镜名称和观测者.

MSV1和MSV2的设计充分考虑了射电干涉测量数据的特性,利用了关系型数据库的技术特色,通过表间数据关联的方式实现了大数据存储.与同时期的其他天文数据存储标准相比,特别是

FITS,测量集数据模型有非常明显的进步,体现在以下几个方面:

1、测量集的核心设计理念是基于表格化数据存储,采用分层结构组织数据,以便支持多种观测模式和数据处理需求.这样的设计显然充分考虑了射电数据的特点.整个数据按逻辑关系分层存储.主表(MAIN表)存储核心的可见度数据,而辅助表格(如ANTENNA表、SPECTRAL_WINDOW表等)存储与观测相关的元数据.这种分层设计不仅提高了数据的组织效率,还使得数据的读取和查询更加灵活.例如,用户可以根据需要单独访问某一时间段的可见度数据,而无需加载整个数据集^[4].

2、支持多维数据的存储.射电干涉测量数据通常包含时间、频率、极化和基线等多个维度,MSV2通过将不同维度的数据分别存储在主表的列中,实现了对高维数据的高效管理.例如,DATA列存储复数形式的可见度数据,而FLAG列用于标记无效数据点.这种设计不仅减少了数据冗余,还便于针对特定维度的数据进行快速分析和处理.

3、扩展性良好.通过添加新的表格或列,用户可以轻松扩展MS文件以支持新的观测模式或数据处理需求.例如,在支持多波束观测时,可以通过在MAIN表中添加新的列来存储不同波束的数据.这种灵活性使得MSV2能够适应不断变化的科学需求.

应该肯定的是,MSV1/V2采用这样的设计后,解决了从90年代出现的多个射电干涉阵的数据存储问题.在SKA项目的诸多先导阵如默奇森大视场阵列望远镜、卡鲁阵列望远镜、LOFAR低频阵列中,MS数据都作为其数据存储格式的标准,并得到了射电天文界广泛地应用.

然而,深入分析MSV2,不难发现,受限于技术和MSV2提出的时代,MSV2的设计仍存在一些局限性,这使得MSV2完全不能满足SKA项目的建设需要,主要体现在以下几个方面:

1、MSV2的存储效率较低.由于MSV2采用基于表格的存储结构,数据以明文形式存储,未引入高效的数据压缩技术.在处理高时间分辨率、多

频率的观测数据时, MSV2文件的存储空间需求急剧增加, 导致存储和传输成本显著上升. 例如, 一次典型的射电干涉观测可能生成数TB的数据, 一个完整的观测会产生14 PB的数据量. MSV2文件的存储效率无法满足这样规模的数据量^[15], 对后端的存储系统和文件系统也带来了巨大的挑战.

2、MSV2的并行处理能力有限. SKA的诸多特性, 如高空间分辨率、高时间分辨率等导致的数据量迅速增长, 迫切需要有效的办法来予以解决, 迫切需要高性能计算技术、并行计算等手段的使用. 然而, MSV2的设计并未充分考虑并行读写需求, 导致其在现代分布式计算环境中的性能受限. 例如, 在多节点集群中同时访问MSV2文件时, I/O瓶颈会出现显著降低, 极大地限制了数据处理效率.

3、MSV2对复杂观测模式的支持不足. SKA提出了若干新的观测模式(如宽带频谱观测、多目标跟踪、干涉阵与单天线模式并存、子干涉阵观测等), 对数据存储格式提出了更高的要求. 然而, MSV2的设计较为固定, 难以灵活扩展以支持这些复杂模式. 例如, 在处理宽带频谱数据时, MSV2需要将数据分割为多个频率通道存储, 这不仅增加了数据管理的复杂性, 还可能引发数据一致性问题^[16].

2.2 MSV3的发展与局限性分析

为了推动SKA的建设, 2018年开始, SKAO曾在关键设计评估后开展了后续测量集标准的推动工作. 2019年MSV3工作组发布了一个测试版本, 即Measurement Set 3.0 β ^[12]. 这一测试版本通过引入显式键、数据版本管理、统一的数据列以及新增的子表和功能, 力求提高射电天文学数据存储和处理的效率和灵活性.

MSV3引入了对DATA (数据)、WEIGHT (权重)和FLAG (标记)数据变量的版本管理机制, 允许多个版本的数据共存, 而当前使用的版本中, DATA、WEIGHT和FLAG数据变量会被明确标识. 这种版本管理方式使得数据的更新和校准更加灵活, 而无需创建冗余的列, 如CORRECTED_DATA (相干数据)或FLAG_VERSION (标记版本)等. 软件实现可以根据需要智能地处理这些版

本, 避免不必要的的数据复制. FLOAT_DATA被DATA替换, 统一了单天线数据和干涉仪数据的存储方式. 软件实现可以根据需要对浮点数据进行高效存储, 并在请求时提供单精度复数数据. 这种改进使得数据存储更加灵活, 减少了因数据类型不同而带来的复杂性.

MSV3引入了多个新的子表, 如BEAM (波束)、EPHEMERIDES (历表)、INTERFEROMETER MODEL (干涉仪模型)和PHASED ARRAY (相位)等. 这些子表为处理复杂的观测数据提供了更多的灵活性和功能支持. 例如, EPHEMERIDES子表可以用于存储近地天体的轨道信息, 这对于处理移动天体的观测数据非常有用. MSV3还对一些功能进行了扩展, 如支持相控阵站基天线(PHASED ARRAY子表)和扫描信息(SCAN子表). 这些扩展功能使得MSV3能够更好地适应更多的现代射电天文学观测需求.

但从另一个方面来看, 尽管MSV3提供了更灵活的数据存储方式, 但在实际应用中, 这些新的存储方式并没有完全达到预期. 例如, 数据版本管理虽然提供了更多的灵活性, 但在处理大量数据时可能会导致性能下降. MSV3的设计要求软件实现更加复杂, 这可能导致开发和维护成本的增加. 例如, 数据版本管理需要软件能够智能地处理多个版本的数据, 而无需创建冗余的列, 这在实际实现中可能非常复杂. 在随后的用户与社区反馈中, 有一些用户认为某些新功能虽然理论上很有用, 但在实际应用中并不常用, 增加了不必要的复杂性. 同时, MSV3仍然没有能够充分适应射电天文观测需求的快速变化. 当涉及到多种复杂的观测模式情况下, MSV3虽然在一定程度上支持这些模式, 但其设计仍较为固定, 难以灵活扩展以适应新的观测技术.

上述的诸多因素决定了MSV3在社区评估中没有得到射电天文学家的认可, 最终成为了一个过渡性的版本, 并没有真正投入到实用. 但这其中的一些设计思想, 为后续版本的发展提供了参考.

3 MSV4的演进与关键技术分析

如前所述, 下一代射电干涉仪的出现—阿塔卡

马大毫米阵列宽带灵敏度升级(ALMA Wideband Sensitivity Upgrade, ALMA-WSU)、ngVLA和SKA的出现,将使天文数据量增加几个数量级,为了解决这些问题MSV4被提出,并初步在开源Python软件包XRADIO (Xarray Radio Astronomy Data IO)上实现.

MSV4的设计最重要的是体现在高效的数据存储与压缩、增强了并行处理能力以及灵活的扩展性.需要明确指出的是,MSV4还没有正式颁布,因此本文依据目前美国国家射电天文台(National Radio Astronomy Observatory, NRAO)完成的相关代码(XRADIO)、推出的软件开发包(Xarray-ms)和若干讨论文档进行了分析.可以预期的是,即使在MSV4正式分布时有可能有进一步的改进或完善,但整个MSV4的整体框架将不会有重大变更.

3.1 MSV4的特色

MSV4与MS的前序版本相比有重大变更,完全改变了整个底层的支撑技术,充分吸收了自21世纪以来由于云计算、分布式计算技术的发展出现的数据管理方面的最新成果.针对海量数据处理,特别是并行处理进行了大量的优化.主要特色如下:

1、自描述性:MSV4数据集是自描述的.这解决了长期以来MSV2的数据需要通过元数据来定义的麻烦.具体来说,观测得到的科学数据以表格形式组织,每个表格包含特定的数据类型(如MAIN表存储可见性数据,ANTENNA表存储天线信息).表格的列名和数据类型明确描述了数据的含义.元数据与科学数据存储在同一框架中,例如:TIME列存储时间戳.ANTENNA1和ANTENNA2列存储基线信息.数据的坐标(如时间、频率、基线)和维度信息被明确记录,便于用户理解数据的组织方式.

2、规则时间-频率网格:MSV4改变了基础数据的保存方法,数据由Xarray Datasets^[17]组成,这些Datasets包含了规则时间-频率网格上的多维数

组(ndarrays).与MSV2的表格(Table)形式不同,MSV4的数据结构更加规则,这使得数据的处理和分析变得更加简单和高效.与MSV2数据以表格形式存储相比,MSV4可以始终保持时间-频率网格的规则性.

3、高效的数据处理:MSV4通过利用Xarray的内部I/O例程和懒加载(Lazy load)机制,实现了高效的数据加载和处理.MSV4支持任意的分块数组(chunked array)类型,包括但不限于Dask^④等分布式框架的支持,这使得数据可以被任意地分块,从而支持并行计算.这彻底解决了MSV2支持的数据类型较为有限,且在数据分块和并行计算方面存在限制的问题.

4、不考虑向后兼容性:MSV4是一种全新的数据模型,并不提供向后兼容性.这意味着MSV4并不直接支持对MSV2数据模型的数据读写.MSV4放弃了原有束缚,重新定义了测量集数据模型.这是MSV4的整体性能有大幅度变化的根本原因,但也使得我们必须重新认识与掌握测量级数据模型,及其带来的数据储存格式兼容性等一系列问题.

5、多应用场景:MSV4能够满足射电天文领域的多种使用情况,如射电干涉测量、单天线观测、实时马赛克观测、星历表观测、异构天线甚长基线干涉、相控阵站和相控阵馈源等.

3.2 MSV4的多层数据架构

依托Xarray底层支持,MSV4的核心架构分为3个层次:数据存储层、数据处理层和元数据管理层.

1、在数据存储层,MSV4直接采用了Zarr^⑤和NetCDF (Network Common Data Form)^[18]来支持高效的多维数据存储和并行访问.Zarr是一种基于分块存储的数据格式,专为高效存储和访问大规模多维数组数据而设计.它采用分块(chunking)和压缩技术,能够显著减少存储空间并提高数据访问速度.Zarr格式支持多种后端存储系统(如本地文件系统、云存储等),并且具有高度可扩展性,

^③<https://github.com/casangi/xradio>

^④<https://dask.org>

^⑤<https://github.com/zarr-developers/zarr-python>

适合处理超大规模数据集. Zarr的另一个优势是其对并行读写的支持, 使其成为分布式计算环境中的理想选择. NetCDF是一种广泛使用的科学数据格式, 主要用于存储多维数组数据. NetCDF格式具有自描述性, 能够将数据与元数据(如单位、坐标轴信息等)一起存储, 便于数据的理解和共享. NetCDF支持高效的数据压缩和分块存储, 适合处理大规模科学数据集. 其跨平台兼容性和丰富的工具支持(如Python的netCDF4库使其成为科学研究和数据交换中的标准格式之一). 因此, 通过两个成熟的底层软件包的引入, MSV4有效地解决了天文数据的存储、新数据结构的设计以及分布式计算等问题. 同时, 利用Zarr和NetCDF对云的支持, 可实现对观测数据的云存储.

```
ms_xds.attrs['data_groups'] = {
    'base': {'correlated_data': 'VISIBILITY', 'flag': 'FLAG',
            'weight': 'WEIGHT', 'uvw': 'UVW'},
    'imaging': {'correlated_data': 'VISIBILITY_CORRECTED',
               'flag': 'FLAG', 'weight': 'WEIGHT_IMAGING', 'uvw': 'UVW'}
}
```

这其中“base”和“imaging”数据组共享相同的“flag”和“uvw”数据变量, 但有不同的“correlated_

2、在数据处理层, MSV4吸取了MSV3的经验, 引入了数据组(data groups)的概念, 允许用户在同一数据集中存储多个版本的可见度数据, 图1给出了MSV4的框架结构. 在这样的情况下, 可以在一个数据组中存储原始可见度数据, 在另一个数据组中存储经过校准的可见度数据. 这种设计不仅提高了数据的灵活性, 还便于用户对不同版本的数据进行比较和分析. 数据组功能对SKA数据处理带来了巨大的方便, 有望降低冗余数据量. 在实际数据处理中利用数据组字典作为主数据集(ms_xds)的属性存储, 可包含一个或多个数据组. 数据变量可以在数据组之间共享, 也可以是特定组独有. 举例来说, 下面描述了有一个“base”组和一个“imaging”组:

data”和“weight”数据变量. 在数据处理时, 用户可以通过指定对应的组名, 提取相应的数据.

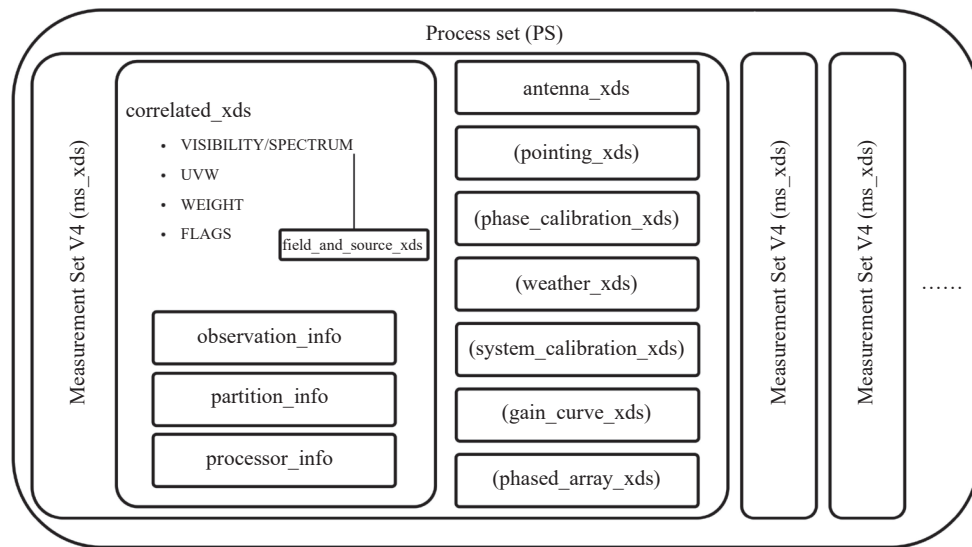


图1 MSV4的数据模式布局示意图. 可选数据集用圆括号表示. 数据变量用大写表示. 后缀“_xds”表示Xarray数据集, “_info”表示字典.

Fig. 1 MSV4 data mode layout diagram. Optional data sets are indicated with parentheses. Data variables are represented in uppercase. The suffix “_xds” denotes the Xarray data set, and “_info” denotes the dictionary.

3、在元数据管理层, MSV4充分利用了Xarray的优势, 采用了自描述的数据模型, 每个数据集都包含完整的元数据信息, 例如观测时间、频率范围和极化设置. 这种设计使得MSV4能够更灵活地支持多种观测模式, 并便于数据的共享和交换.

3.3 预期关键性能变化

依托现有技术资料, 我们可以进一步分析MSV4在存储、并行读写方面的一些性能变化(如表2给出了相关格式的性能对比). 从存储性能来看, MSV2自身不带压缩功能, 但一般可以通过GZIP等第3方压缩软件进行压缩. 而MSV4和

HDF5都有原生的压缩能力. 根据HDF5公开的数据, 在处理浮点数数组等科学数据时, HDF5的压缩比通常在1.5:1到2.5:1之间^[19]. 利用这个压缩比为参考, 再考虑到MSV4的冗余数据不再需要存储, 我们可以进一步估计MSV4的数据压缩.

从并行读写性能来看, 当前经典的Lustre等文件系统一般可以达到5–10 GB/s的性能. 因此, MSV2的并行读取主要是不同的数据表的并行读取, 最多也就能达到文件的最高性能. HDF5原生支持并行读写. 资料表明, HDF5并行读写时, 可以达到15–25 GB/s的性能. 综合来看, 预期MSV4的性能会与HDF5基本相当.

表 2 SKA数据的MSV2、MSV4和HDF5比较分析

Table 2 Comparative analysis of MSV2, MSV4, and HDF5 for SKA data management

Feature	MSV2	MSV4	HDF5
Storage Performance	Uses a multi-file directory structure with significant metadata overhead, resulting in lower efficiency. It employs third-party compression software with limited compression capabilities, typically achieving a compression ratio between 1.2:1 and 1.5:1. For 100 TB of data, compression reduces it to 66.7–83.3 TB.	Optimized for big data, with an efficient layout and improved performance. By significantly reducing redundant data, it is expected that 100 TB of data will result in an actual storage requirement of 33.3–50 TB.	Efficient when configured with block storage, making it suitable for large-scale datasets. Supports multiple compression methods, with 100 TB of data being compressed to 40–66.7 TB.
Parallel Read/Write Performance	Limited parallel I/O support, with throughput ranging from 5 to 10 GB/s. Parallel reads are supported across different tables, but not within the same data table.	Designed specifically for parallel I/O, with throughput ranging from 15 to 25 GB/s. Optimized versions could potentially reach 20–30 GB/s.	Supports parallel I/O through MPI-IO, with throughput ranging from 15 to 25 GB/s.

3.4 MSV4功能变化对现有软件的影响

SKA未来科学研究将完全依赖于SRC平台. SKA AA0.5即将结束, 后续的AA1、AA2阶段(SKA-Mid中有64碟形天线+MeerKAT 4个碟形天线, SKA-Low中有68个站)[®]将随即展开. SKA的早期科学研究工作也将会同步开展. 后续数据如果采用MSV4数据模型存储, 则原有的各种软件将完

全无法使用. 因此, 理解并掌握MSV4数据的变化、升级现有软件, 对后期开展SKA的数据处理和利用SKA开展科学研究是非常必要的.

1、如前所述, MSV4数据结构有了根本性的变化. 通过Xarray的自描述机制, 每次观测得到的数据, 如可见度数据会包含单次观测、光谱窗、极化设置、观测模式、处理器和每天线波束的信

[®]<https://www.skao.int/en/science-users/599/scientific-timeline>

息. 同时, 数据存储在有标有 n 维数组的数据集中, 这个并不是传统的数据库的表格. 在传统软件开发中, 利用python-casacore软件读取的可见度数据用户习惯于基于一个可见度数据就是一个Numpy的复数数组, 具体的每一个可见度数据的观测时间、所属基线等都需要通过开发者自己从其他数据表中来对齐. 但在MSV4中, 读入的数据是一个 n 维数组, 维数为时间 \times 基线 \times 频率 \times 偏振(行已被拆分为时间 \times 基线). 从具体的软件开发来说, MSV4对应的数据模式会更易于理解与分析, 虽然在内存开支上会有一些增加.

2、在MSV2中大多数“键(KEY)”过去都是采用隐式编号索引, 也就是在MSV2中简单地采用下标值来索引数据. 在MSV4中已改为描述性名称. 这一改变虽然可以不用修改原有代码, 但会显著提高代码的可读性, 提高代码质量. 同时允许在不重新索引的情况下选择子数据, 并使数据组合更加方便.

3、随着MSV4的升级, 数据描述的概念已被弃用, 取而代之的是spectral_window_name和polarization_setup. 考虑到射电干涉仪都是光谱仪, 每个通道的权重都是不同的, 因此WEIGHT重新定义为WEIGHT_SPECTRUM, WEIGHT数据变量的形状与VISIBILITY/SPECTRUM数据变量相同. 在MSV2中的场(FIELD)、源(SOURCE)和星历数据已合并为一个数据集. 天线和馈源数据已合并为一个数据集(MSV4的每个天线只能有一种馈源类型). 这些改动对后续数据处理软件对数据的读写都有显著影响. 此外, 与MSV2不同, MSV4使用JPL Horizons星历表来创建星历表模式.

4、为了降低数据量, 提高数据处理的可用性, 使用数据组对VISIBILITY/SPECTRUM、WEIGHT、UVW和FLAG数据变量进行版本控制.

4 MSV4对SKA项目的推动作用分析

4.1 MSV4数据模型的组织结构对SKA观测类型的支持

根据SKAO发布的数据管理政策^[20-21], 未来SKA的观测将包括多种项目类型, 如关键科学项目(Key Science Program, KSP)、首席科学家(Prin-

cipal Investigator, PI)项目、总干事自由时间项目等. 不同的观测项目类型涉及到不同的观测时间, 如KSP提供的观测时间将至少有5个周期以上, 而PI项目一般就在一个周期内完成. 不同的项目类型对数据管理提出了全新的要求. 在KSP项目没有完成前, 数据并不能释放, 其他团队成员的人也无权来访问处理这些KSP数据.

从当前的MSV4的数据组织分析来看, MSV4完全能够支持这样不同的观测类型和后续的处理, 在数据组织层面可以有效地对不同的观测类型数据进行管理. 从图1给出的MSV4的数据组织方式示意图也可以看出: 一次观测获得的MSV4的数据集可以包括多个子集, 包括correlated_xds (干涉数据)、antenna_xds (天线信息)、pointing_xds (指向信息)等. 每个子集都通过标签化的多维数组存储数据, VISIBILITY数据变量存储干涉可见度数据, 其维度包括时间、基线、频率和极化. MSV4支持动态扩展, 用户可以根据需要添加新的数据变量或元数据字段. 这意味着用户可以为新的观测模式添加特定的数据变量, 而无需修改现有的数据结构. 这种设计使得MSV4能够更灵活地适应SKA未来的观测需求.

从图1也可见, 除单次观测以外, MSV4支持了进程集(Processing Set, PS)的概念, 以方便组织和处理与单个观测相关的多个数据集. PS能将多个MSV4数据集组合在一起, 以便对它们进行集体处理. 这为管理和处理大量观测数据提供一个方便的框架. 通过对相关数据集进行分组, 处理集可以实现高效的数据选择、过滤和处理. 例如, 它允许用户以协调的方式对多个数据集进行操作, 确保一致性并降低数据处理的复杂性. 此外, 进程集还提供了一些方法来总结元数据、根据各种标准选择子集以及将数据导出到存储格式. 例如, 它可以通过summary方法生成数据集的概要信息, 包括数据集的名称、意图、极化方式、谱窗名称等. 这使得科学家能够快速了解PS中各个数据集的基本信息, 从而更好地进行数据管理和分析. 总体来看, 进程集对于SKA的KSP任务, 如需要长时间观测的宇宙再电离等十分有用, 将跨度长达多年或者数个月的观测数据进行有效的整合与管理, 保证了SKA的长周期观测需要.

4.2 MSV4在SKA数据处理中的可用性分析

如前所述,为了尽可能加快MSV4的开发,尽快满足NRAO和SKAO对数据处理的迫切需要. MSV4充分利用了Xarray数据集^⑦,采用多维数组和元数据标签的结构,取代了传统的表格化存储方式.这种设计使得MSV4能够更高效地处理大规模数据,并支持复杂的观测模式.这也意味着,MSV4的并行计算能力将取决于底层Xarray的性能.可以认为,在MSV3中所面临的问题,基本都在MSV4中得到了解决,其技术关键在于采用了基于Xarray数据集. Xarray带来了在性能、数据结构上的显著性变化,但也导致了MSV4与原来的数据存储方式不兼容.

在具体的计算中,以SKA连续管线为参考,预计1 h的观测可能会产生超过1 PB的可见度数据,最终产生TB量级的科学图像^⑧.以 32768×32768 像素的网格化过程内存开销为例,为了获得高精度的网格化结果,需要申请的网格内存接近74 GB.当成像尺度达到96 K,网格化所消耗的内存将达到667 GB.针对这一规模,相信Xarray能够在分布情况下完成上述规模的计算与结果保存,从而能够满足未来SKA数据的处理需要.

4.3 对SKA数据处理管线的影响与应对

随着SKA开始逐渐转入前期科学研究阶段,对我国科学家来说,迫切需要面对未来可能的MSV4数据提前开发各种数据预处理与处理管线.

从MSV4目前的发展趋势来看,SKAO最终推送到各国SRC的科学数据将很可能基于MSV4标准.考虑到MSV4不支持向后兼容,这意味着当前基于MSV2开发的各种应用后续将没有办法直接应用到MSV4上.为了避免这样的状况,我国科学家当前正在开发的各类数据处理程序需要对此有充分的认识,必须做好各种准备,以应对基础数据存储格式的改变.这一开发问题也是SKA国际各方合作团队共同面临的问题.在这方面,有些国家已经迅速开展了相关的前期研究工作.我们关注到,南非同仁开发了xarray-ms^⑨. xarray-ms在MSV2之上提出了测量集MSV4视图.它通过在MSV2数

据上使用MSV4 Xarray视图进行开发,这样可以在易于理解的数据上开发应用程序,然后无缝过渡到更新的格式.数据也可以通过Xarray的本地I/O例程导出到更新的格式(主要是Zarr).不过,对于软件开发人员来说,这两种格式的Xarray视图都是一样的.显然,这一模式对我们当前的软件与应用开发具有指导意义.

4.4 对SRC建设与运行的影响

根据SKA的建设要求,未来的科学数据都需要在各国SRC平台进行. SKA1-Low和SKA1-Mid观测获得的数据,经过SDP的初步处理,就会被推送到SRC进行后续处理.未来所有SKA科学研究工作都只能在SRC平台上进行.可以认为,MSV4的推出对于SRC的建设有重要的推动作用. MSV4有望进一步降低数据存储资源开销,降低SRC的建设成本.同时也有利于提升后续数据处理性能.对于各SRC间的远程数据传输与同步来说,数据量的下降意味着网络传输时间的下降,这对于SRC的运维来说是非常有意义的.

5 结论

随着国际射电天文界诸多天文台共同提出了MSV4版本草案,对测量集数据模型进行了全新的定义.这个版本经过社区评审和测试后,有望成为未来新一代射电干涉望远镜数据存储的标准格式.除了射电干涉阵以外,MSV4同时也可以支持单天线、相位阵等多种场景.做为基础数据模型,MSV4的推出将对整个射电天文领域产生重要的影响.我们必须尽早分析、了解、掌握MSV4的技术要点和功能特性,为后续开展数据处理奠定必要的基础.

本文给出了测量集文件的定义,系统介绍了其发展演进的过程,对测量集数据模型各版本的技术特点进行了分析.在此基础上,重点阐述了MSV4对我国参与SKA科学数据处理器、区域中心建设的影响.本文工作对后续我国建设SKA中的科学数据处理包、区域中心、开发数据应用处理系统的研发有借鉴与参考作用.

^⑦<https://github.com/pydata/xarray>

^⑧SKA技术文档SKA SDP Parametric Model Documentation <https://sdp-par-model.readthedocs.io>

^⑨<https://github.com/ratt-ru/xarray-ms>

参考文献

- [1] Dewdney P E, Hall P J, Schilizzi R T, et al. IIEEP, 2009, 97: 1482
- [2] Wieringa M H, Cornwell T J. Definition of Measurement Set: AIPS++ Note 191. Charlottesville, VA: National Radio Astronomy Observatory, 1996
- [3] Kembell A J, Wieringa M H. Measurement Set Definition Version 2.0. Charlottesville, VA: National Radio Astronomy Observatory, 2000
- [4] McMullin J P, Waters B, Schiebel D, et al. ASPC, 2007, 376: 127
- [5] Wells D C, Greisen E W, Harten R H. A&AS, 1981, 44: 363
- [6] Folk M, Heber G, Koziol Q, et al. Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases. New York: Association for Computing Machinery, 2011: 36–47
- [7] Anderson K, Alexov A, Bahren L, et al. ASPC, 2011, 442: 53
- [8] Viallefond F. ASPC, 2006, 351: 627
- [9] Greisen E W. AIPS FITS file format, AIPS Memo Series 117. Charlottesville, VA: National Radio Astronomy Observatory, 2012
- [10] Greisen E W. The FITS Interferometry Data Interchange Convention—Revised, AIPS Memo Series 114. Charlottesville, VA: National Radio Astronomy Observatory, 2011
- [11] Selina R J, Murphy E J, McKinnon M, et al. SPIE, 2018, 10700: 497
- [12] Dijkema T J, Golap K, Hiriart R, et al. Measurement Set definition version 3.0 β . Charlottesville, VA: National Radio Astronomy Observatory, 2019
- [13] CASA Team. *PASP*, 2022, 134: 114501
- [14] McMullin J P, Golap K, Myers S T. ASPC, 2004, 314: 468
- [15] Offringa A R, McKinley B, Hurley-Walker N, et al. *MNRAS*, 2014, 444: 606
- [16] Bhatnagar S, Rau U, Golap K. *ApJ*, 2013, 770: 91
- [17] Hoyer S, Hamman J. *JORS*, 2017, 5: 10
- [18] Rew R, Davis G. *ICGA*, 1990, 10: 76
- [19] Habib S, Morozov V, Frontiere N, et al. Communications of the ACM, 2016, 60: 94
- [20] Ball L. SKA Observatory Access Policy. SKAO-GOV-000088. Macclesfield, Cheshire: SKA Observatory, 2023
- [21] Breen S, Bolton R, Chrysostomou A. SKAO Science Data Products: A Summary. SKA-TEL-SKO-0001818. Macclesfield, Cheshire: SKA Observatory, 2021

Measurement Set Version 4 Data Model: Analysis and Relevance to China's SKA Participation

XU Yi-jun^{1,2} WANG Feng^{1,2}

(1 Center for Astrophysics, Guangzhou University, Guangzhou 510006)

(2 Great Bay Center of National Astronomical Science Data Center, Guangzhou University, Guangzhou 510006)

ABSTRACT The Square Kilometre Array (SKA) has entered the construction and commissioning phase. However, the storage and processing of its massive data remain one of the critical challenges that need to be addressed. The construction and operation of the SKA Regional Center (SRC) also heavily rely on efficient data storage and management solutions. To explore the key issues in SKA data storage and management in depth, this paper systematically investigates the development history of Measurement Set (MS) technology, the technical characteristics of its various versions, and their limitations in SKA applications. The paper focuses particularly on analyzing the technological innovations of the next-generation Measurement Set Version 4 (MSV4) and its potential impact on SKA data management and scientific research. The study indicates that MSV4, by introducing features such as self-description, modular architecture, and efficient storage, is expected to effectively alleviate the bottlenecks in SKA data storage and processing. If MSV4 becomes the standard for SKA data storage, it will significantly influence the construction of SRCs in China and SKA scientific research. Therefore, it is essential to closely monitor the technological development of MSV4 at this stage and to proactively prepare relevant technological reserves to ensure the successful development, upgrading, and transformation of SRC scientific application pipelines and data processing software for SKA scientific research.

Key words astronomical data bases, instrument: interferometer, methods: numerical, data analysis