

中国科学院高能物理研究所
Institute of High Energy Physics
Chinese Academy of Sciences



国家高能物理科学数据中心
National HEP Data Center



高能所计算中心
IHEP Computing Center

第二十届全国科学计算与信息化会议

基于Daisy的 天文卫星科学数据处理软件框架

王爽(wangshuang@ihep.ac.cn), 张红梅, 胡誉 等

中国科学院高能物理研究所

2023年7月12日 西宁



目录



一、背景

二、软件框架介绍

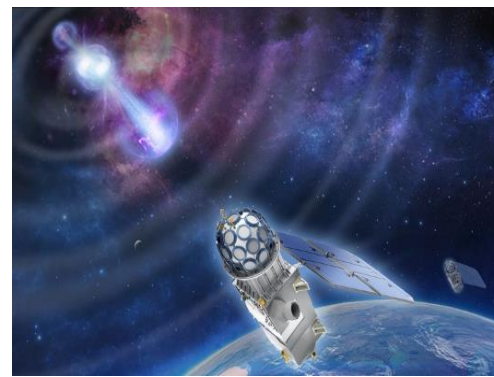
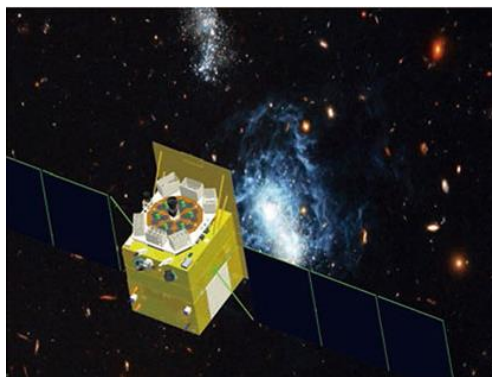
三、进展情况

四、总结



卫星数据海量性

- 计算机技术、通信技术、卫星导航技术等迅速发展
- 卫星种类数量增多、探测器精度越来越高
- 天文卫星：观测和研究宇宙中高能天体和高能宇宙现象 HXMT
- 数据量达到TB级甚至PB级



GECAM

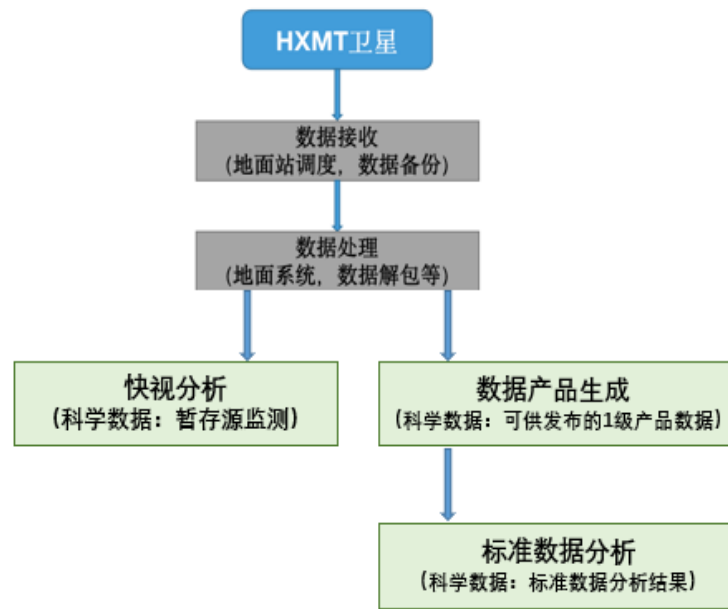


eXTP



卫星数据处理

- 数据处理过程具有**复杂性**和**专业性**，数据处理步骤具有一定**重复性**
- 数据处理需求**多样性**
- 传统人工处理方式缺点
 - 耗费人力和时间成本
 - 降低数据处理效率



HXMT卫星数据处理流程

亟需实现天文卫星数据处理流程的标准化和自动化



目录



国家高能物理科学数据中心
National HEP Data Center



高能物理计算中心
HEP Computing Center

一、背景

二、软件框架介绍

三、进展情况

四、总结



● 基本概念



流水线思维理念

- 软件为流水线，数据为产品
- **问题分解**：模块化设计和开发，便于各成员间的分工和合作
- **有机整体**：通过框架实现不同模块的组合或替换，保证前后步骤的正确衔接



团队协作形式研发

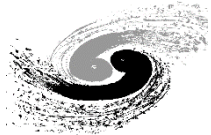
- 开发人员实现较复杂、底层的功能，科学家不关注软件技术细节
- 一定周期内，保持稳定的软件接口、统一的开发规范，便于软件的发展和**维护**



核心功能

- **数据管理**
 - 数据格式及存取方法定义
 - 内存数据管理
- **任务执行的流程控制（工作流）**
 - 算法加载
 - 模块执行顺序
- **公共服务功能模块**
 - 支撑框架中任务执行的公共服务
 - 基础库
- **用户接口**
 - 命令行
 - UI接口

软件框架决定着整个软件系统的实现、使用方式、性能和可靠性等

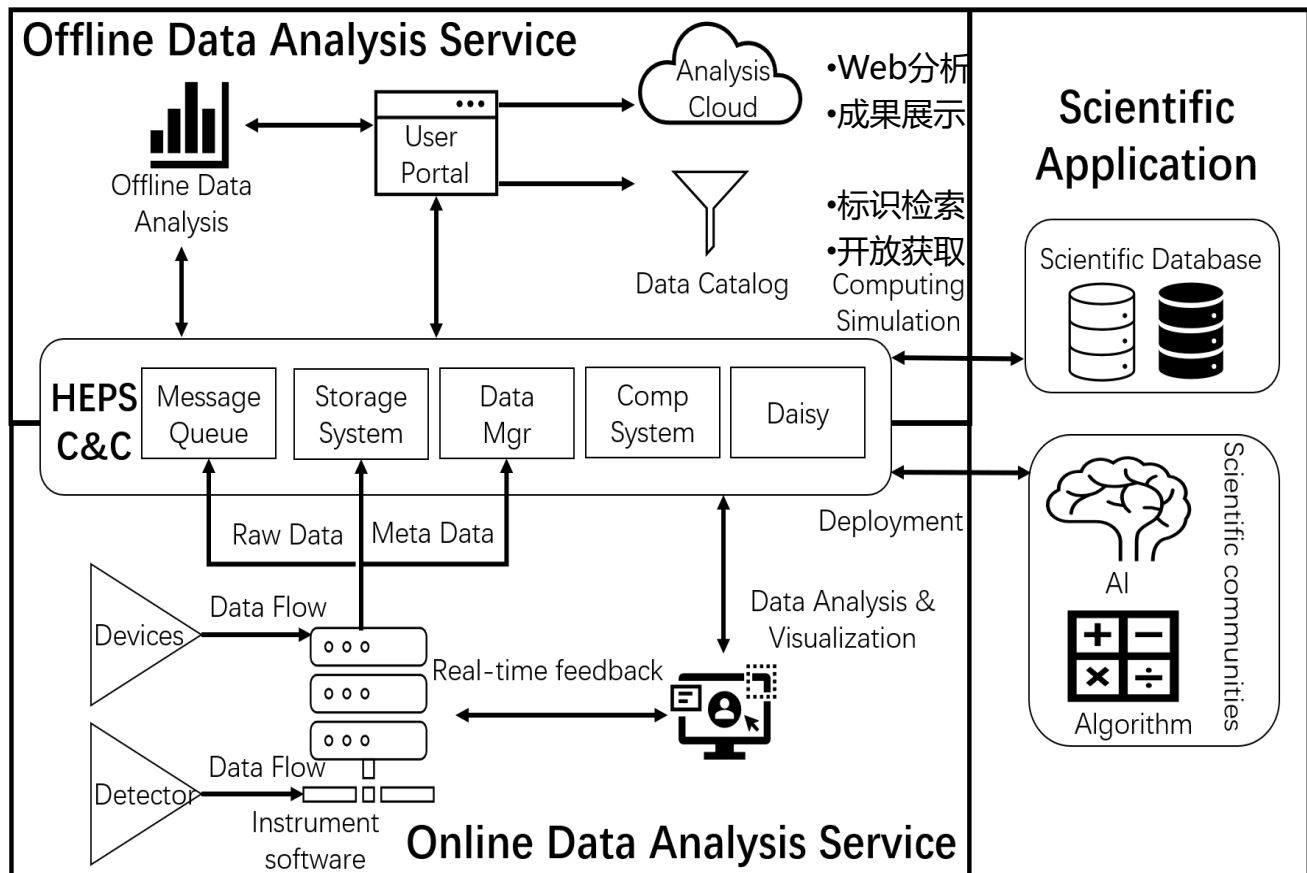


数据处理软件框架介绍



● Daisy框架

目的：形成一个通用的、具有良好扩展性的**基础软件架构**，集成多种方法学算法和工具，屏蔽计算架构的复杂性和计算资源的多样性，为上层应用软件和用户提供统一的调用接口，并在此基础上开发数据可视化和分析桌面等通用组件。



● 在线数据处理和分析

- 数据产品生成
- 文件/流 (全部/部分)

● 离线和远程数据分析

- 规约, 重建, 建模和模拟
- 庞大的数据容量 (数据托管, 算力)

● 计算基础设施

- HPC 集群 (Spark/Slurm)
- 单个工作站 (虚拟机)
- 接入方式: JupyterHub (notebook app)
- 远程桌面 (traditional app),

● 软件部署和容器化

- 容器封装软件环境
- 方便移植、复用软件环境, 复现分析结果

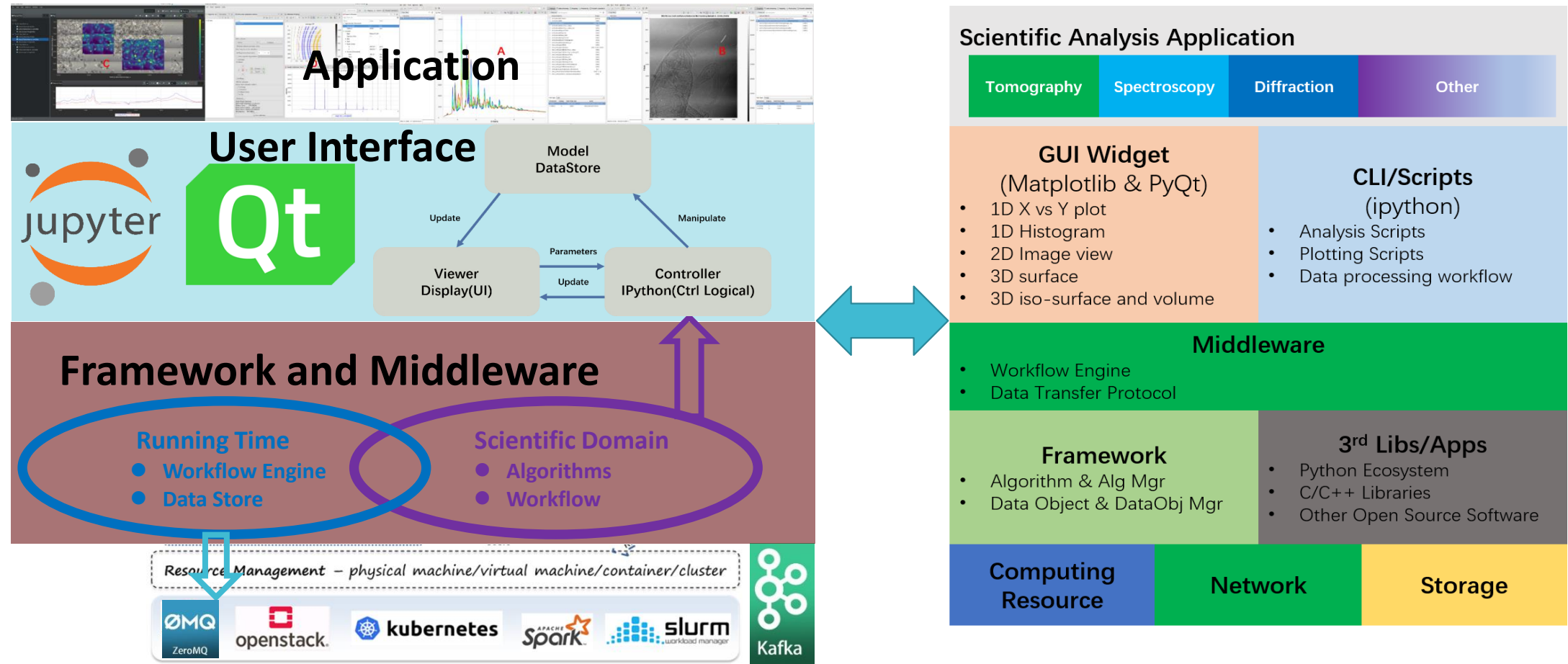


数据处理软件框架介绍

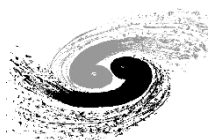


● Daisy框架整体架构

从下向上分为四层，分别是基础设施层，领域层，用户界面层和应用层



对科学家隐藏内部细节，使其专注于数据处理业务过程，并通过简单的接口降低使用难度



数据处理软件框架介绍

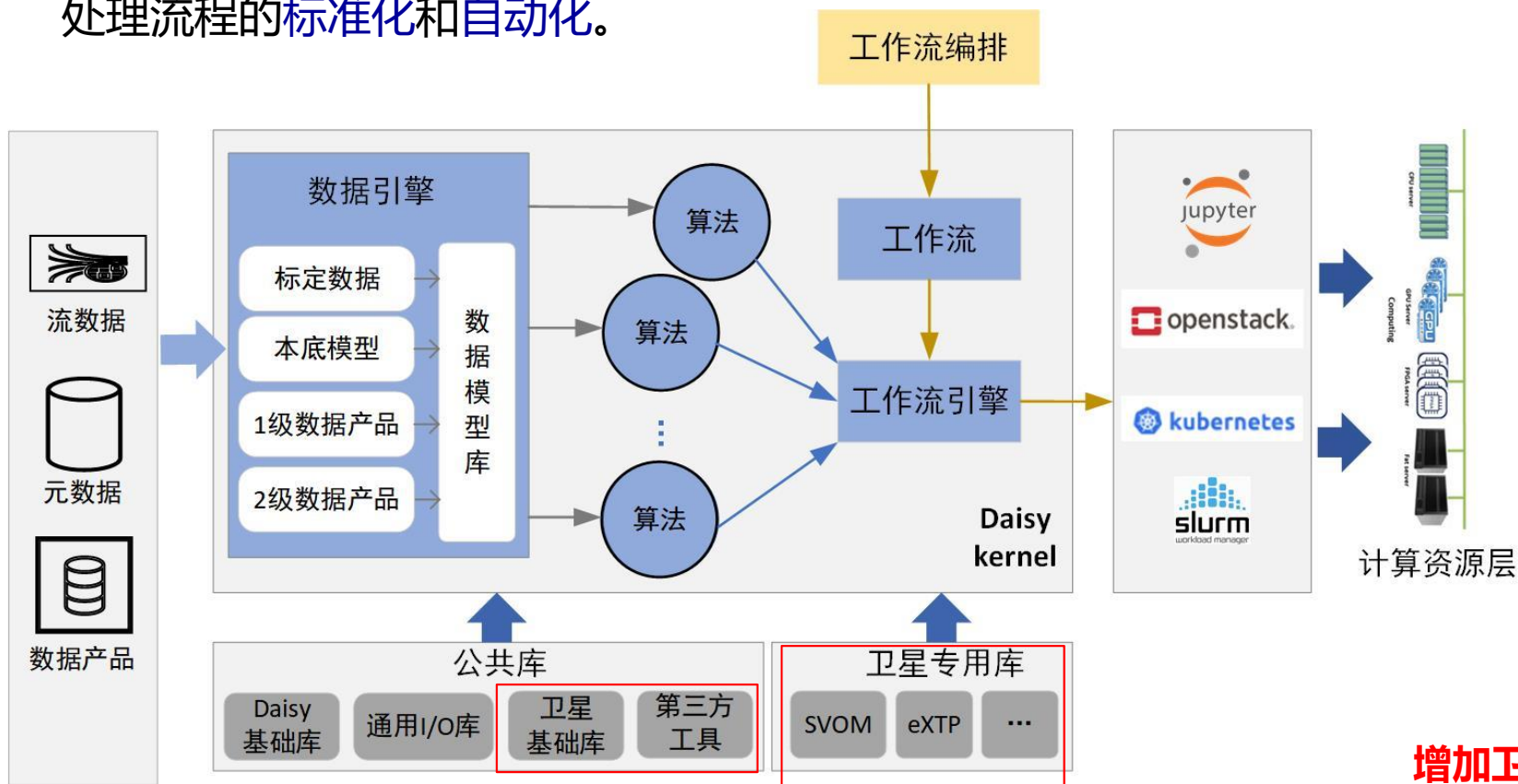


国家高能物理科学数据中心
National HEP Data Center



● 逻辑结构

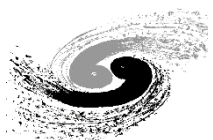
结合天文卫星特点，同时借鉴Heasoft设计理念，在Daisy软件框架上进行扩展和衍生，以实现卫星科学数据处理流程的**标准化**和**自动化**。



核心模块:

- **数据模型库:** 管理框架中的数据对象，使数据对象能被工作流中的算法调用。
- **算法:** 框架中的最小单元，即具体的数据处理模块。
- **工作流:** 一系列调用算法的序列。
- **工作流引擎:** 管理工作流执行过程中的运行环境，是算法、工作流、底层计算资源的桥梁。

**增加卫星基础库、卫星专用库和第三方工具
研发工作流调度系统**

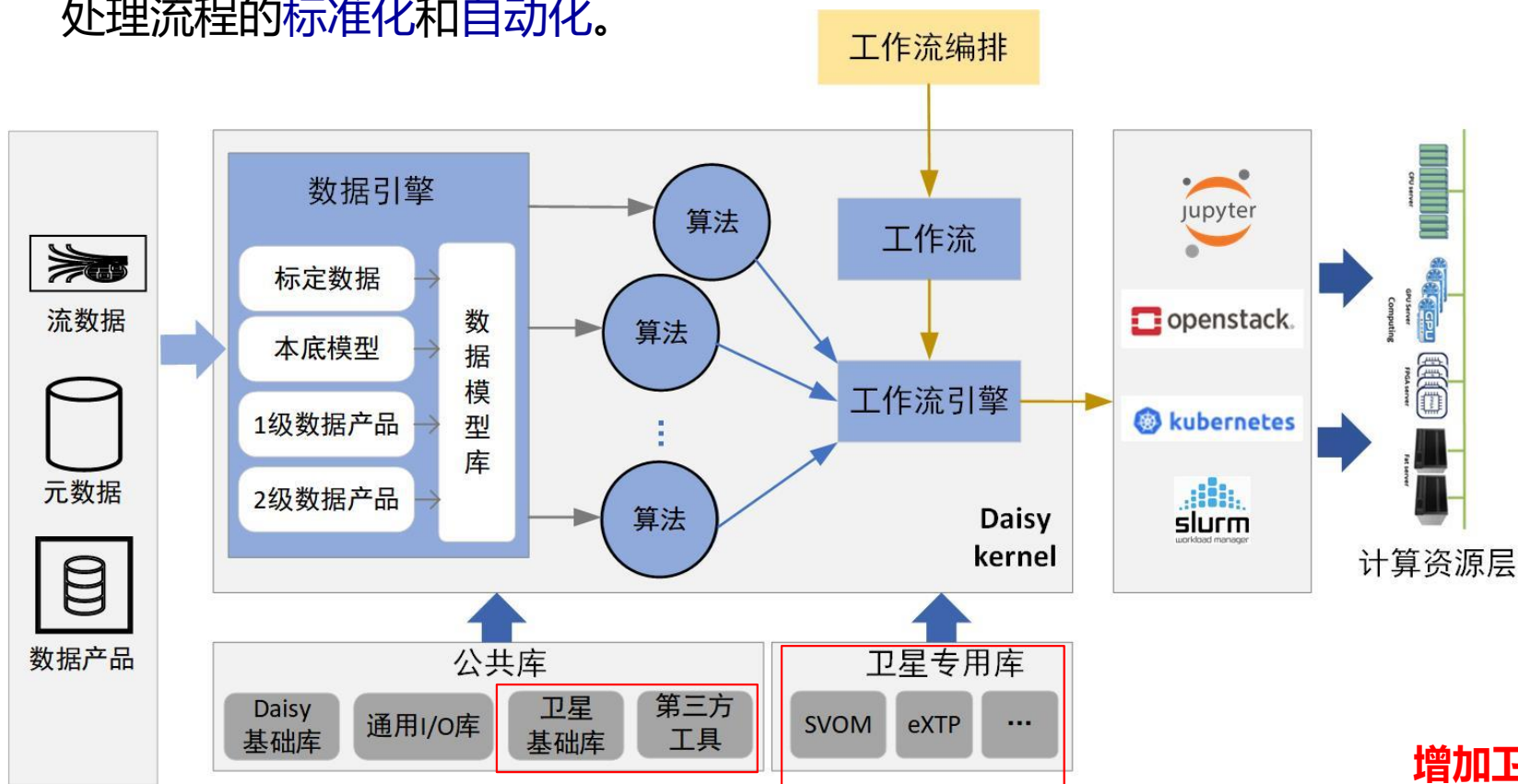


数据处理软件框架介绍



● 逻辑结构

结合天文卫星特点，同时借鉴Heasoft设计理念，在Daisy软件框架上进行扩展和衍生，以实现卫星科学数据处理流程的**标准化**和**自动化**。

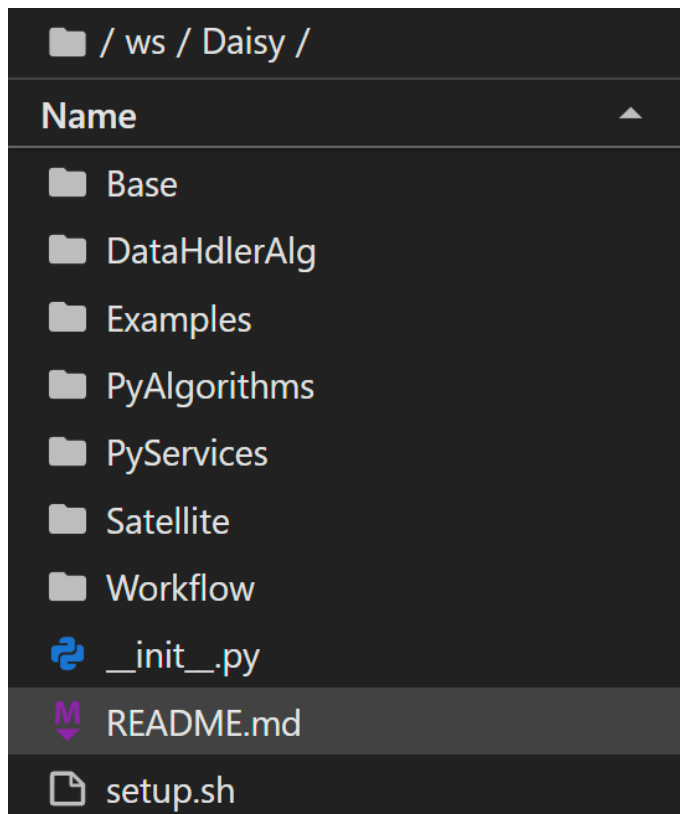


- 控制逻辑和执行逻辑分离
- 执行模块和数据模型分离
- 外部代码可以被集成进入执行逻辑

**增加卫星基础库、卫星专用库和第三方工具
研发 workflow 调度系统**



● 代码结构



代码组织结构

文件目录:

- **Base**: Daisy框架的核心基类, 包括datastore, algorithm, workflow, workflow engine以及service等
- **DataHdlerAlg**: 不同数据类型的I/O实现, 如FITS, HDF5
- **PyAlgorithms**: 数据处理算法, 如光变曲线、计数谱。
- **Satellite**: 卫星库, 包括基础库和专用库
- **Workflow**: 工作流, 数据处理流程的完整实现
- **PyServices**: 外部公共服务, 如条件数据库查询等
- **Examples**: 用户的开发示例

集成到框架中的算法, 须实现`initialize()`、`execute()` 和 `finalize()` 三个方法, 对应算法初始化、算法执行和算法销毁三个阶段。



目录



一、背景

二、软件框架介绍

三、进展情况

四、总结



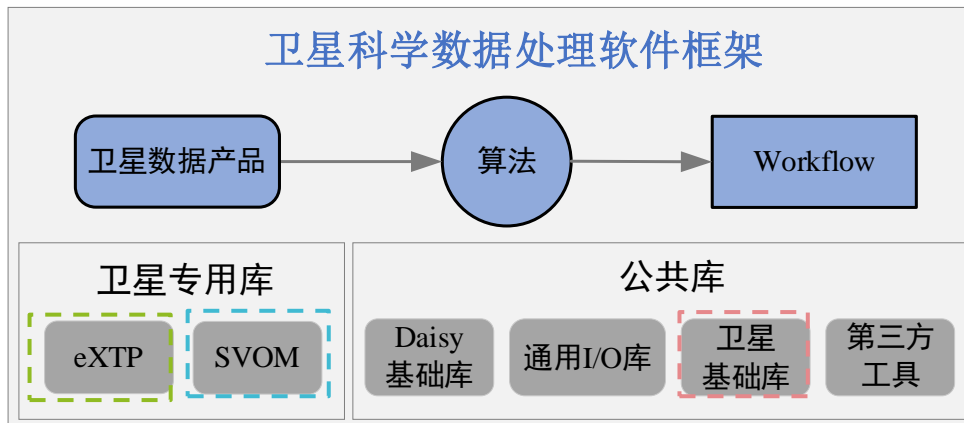
● 代码集成



集成两个卫星数据处理算法

- 数据模板生成工具 (tsv-json-fits) → 卫星基础库
- SVOM (光变曲线CLC、计数谱CSP) → 卫星专用库 SVOM
- eXTP (数据切割算法) → 卫星专用库 eXTP

卫星科学数据处理软件框架



```

+ / ... / Satellite / PublicTools /
Name Last Modified
grmtools 23 days ago
_init_.py 23 days ago
event.py 23 days ago
FitsTools.py 23 days ago
io.py 23 days ago
lightcurve.py 23 days ago
spectrum.py 23 days ago
TemplateGenerate_buildJson.py 6 minutes ago

TemplateGenerate_buildJson.py
1 from __future__ import print_function
2 import Daisy
3 import csv, json
4 from astropy.io import fits
5 from astropy.table import Table
6
7 class TemplateGenerateJson(Daisy.Base.DaisyAlg):
8     def __init__(self, name):
9         super().__init__(name)
10    def initialize(self):
11        self.LogInfo("initialized, 初始化, 开始生成json文件")
12        return True
13    def execute(self, input_tsv, output_json, eventfile, eventjsonfile, hefile, hejson, outputfile):
14        self.LogInfo("execute, 执行, 建立primary_header json文件")
15        self.json_build(input_tsv, output_json, delimiter='\t', encoding='UTF-8')
16        self.LogInfo("execute, 执行, 成功构建primary_header json文件")
17        self.LogInfo("开始生成primary_hdu header")
18        header = self.header_build(output_json)
19        primaryhdu = self.primary_hdu(header)
20        self.LogInfo("成功生成primary_hdu header")
21        self.LogInfo("开始生成event header json文件")
22        self.json_build(eventfile, eventjsonfile, delimiter='\t', encoding='UTF-8')
23        event_header = self.header_build(eventjsonfile)
24        self.LogInfo("成功生成event header json文件")
25        self.LogInfo("开始生成he data json文件")
26        self.json_build(hefile, hejson, delimiter='\t', encoding='UTF-8')
27        table = self.empty_table_build(hejson)
28        hedatahdu = self.hdu_generate(table, header, 'hedata')
29        self.fitsbuilder(outputfile, primaryhdu, hedatahdu)
30        self.LogInfo("成功生成fits文件")
31        return True

```

数据模板生成工具

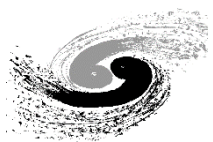
```

+ / ... / Satellite / eXTP /
Name Last Modified
_init_.py 23 days ago
AlgSplitData.py 4 minutes ago

AlgSplitData.py
1 from __future__ import print_function
2 import Daisy
3 import csv, json
4 from astropy.io import fits
5 import numpy as np
6 from sys import getsizeof as getsize
7
8
9 class AlgSplitData(Daisy.Base.DaisyAlg):
10    def __init__(self, name):
11        super().__init__(name)
12
13    def initialize(self):
14        self.LogInfo("initialized, split data")
15        return True
16
17    @profile
18    def execute(self, indir, outdir, outfilename, eventslist):
19
20        with open(eventslist) as f:#
21            file=f.readlines()
22            file=[indir+filename.strip() for filename in file]
23
24            hdulist=fits.open(file[0])
25
26            priheader=hdulist[0].header
27            mergedata=hdulist[1].data
28            mergegti=hdulist[2].data
29

```

数据分割算法

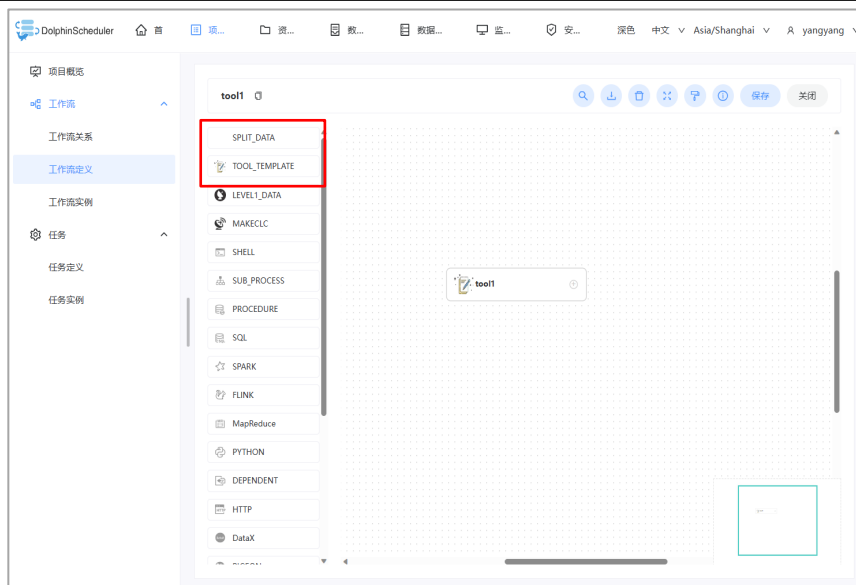


● Workflow

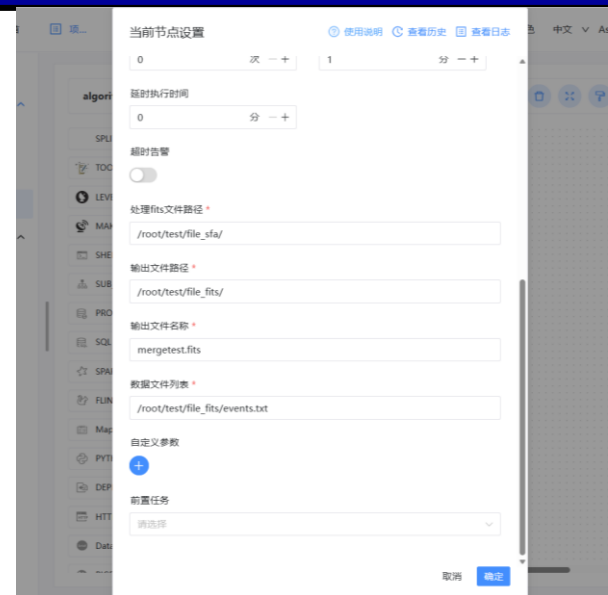


workflows 开发

- 基于DolphinScheduler研发
- 完成组件在工作流平台上的注册
 - 算法库：数据分割算法
 - 工具库：数据模板生成工具
- 通过拖拽任务组件，实现用户对卫星数据处理流程的编排和定义
- 通过监控系统，实现用户对卫星数据处理流程的监控
 - 查看任务执行状态：成功、失败、未执行
 - 查看日志



用户自定义数据处理流程



用户配置页面 - 数据分割算法



工作流调度系统监控页面



● 用户启动方式



命令行/终端terminal

- 登录 <https://sdcscompute.ihep.ac.cn/jupyterhub> 为用户分配容器和资源
 - ✓ 统一认证账号密码
 - ✓ 选择Astronomy环境启动
- 配置环境
 - ✓ `source Daisy/setup.sh`
 - ✓ `source /opt/setup.sh`
 - ✓ `export PYTHONPATH="/opt/sniper/SniperInstall/python:/opt/sniper/SniperInstall/lib:/opt/sniper/SniperInstall/pylib:/opt/conda/lib/"`

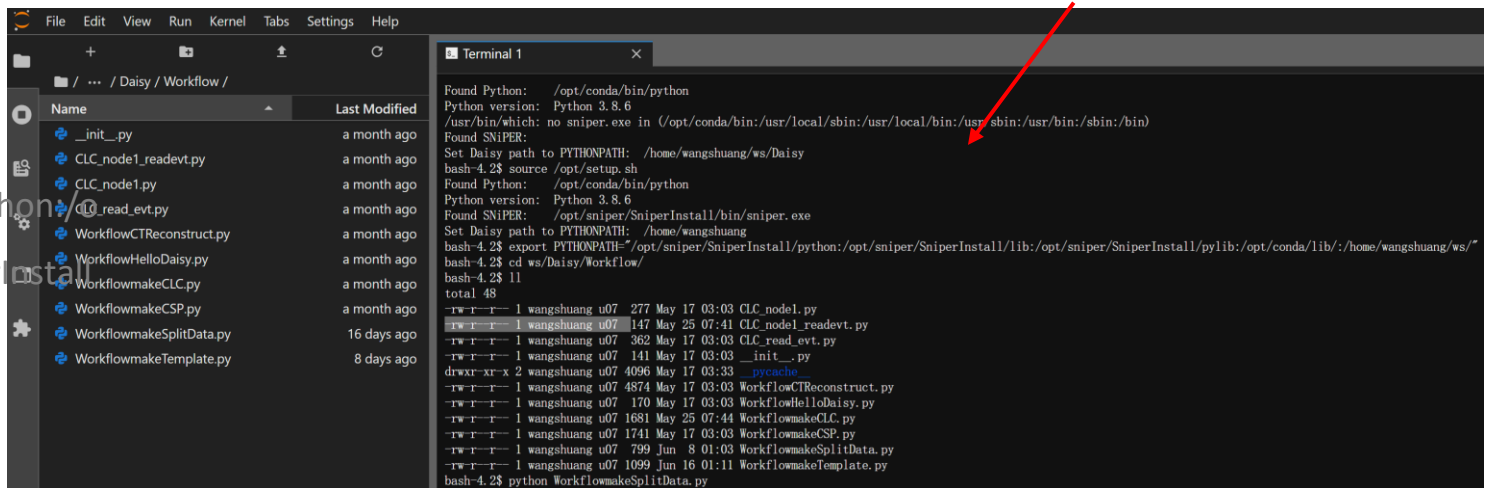
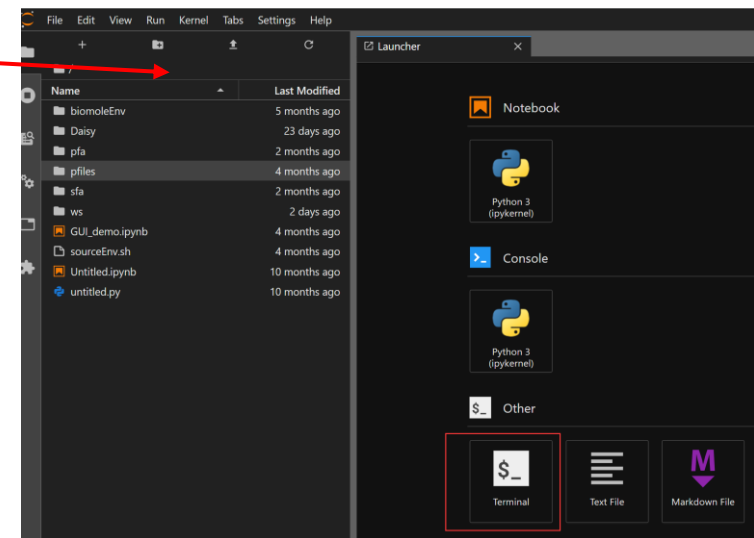
启动已选择的分析环境

应用分析环境列表

物理分析环境	
<input type="radio"/>	HXMT data analysis HXMT interactive data analysis service. pandas, numpy, matplotlib, ipywidgets appmode requests h5py plotly astropy PyERFA scipy ipypdateime.
<input type="radio"/>	Astronomy Astronomy

光源分析环境

开发者环境





● 用户启动方式



workflows平台

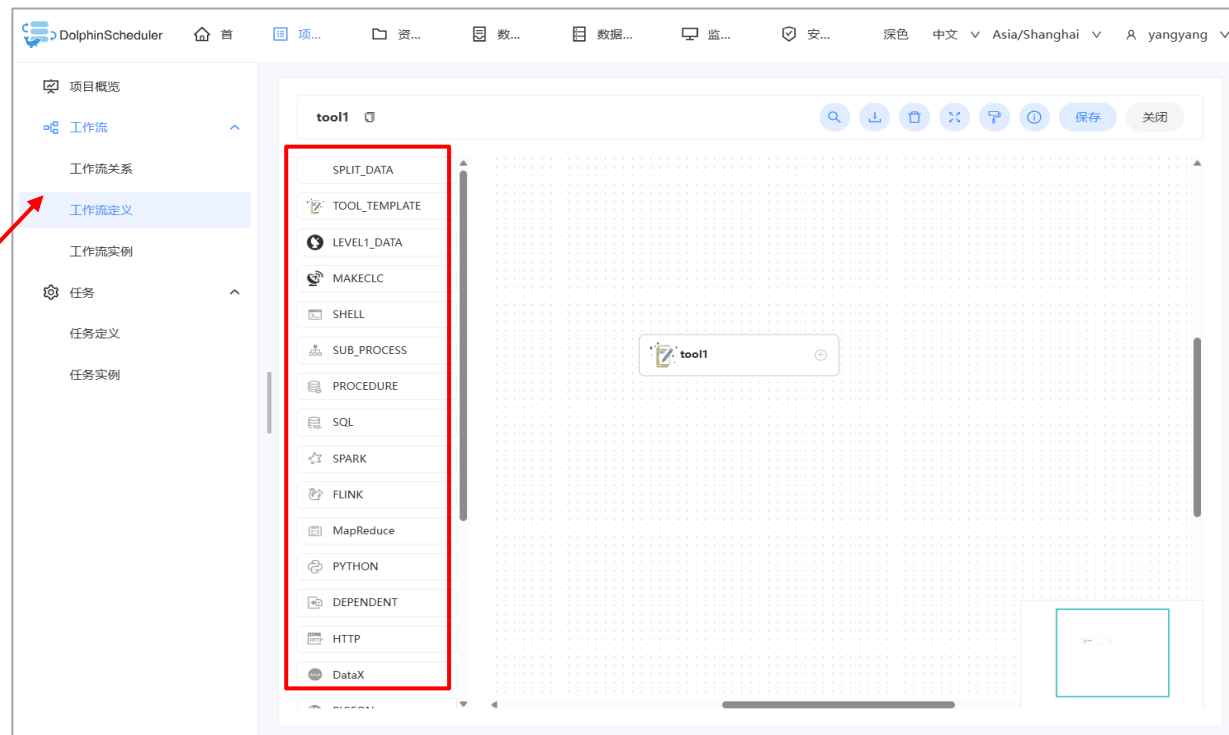
- 管理员admin新建用户，并设置权限
 - ✓ 创建项目
 - ✓ 创建工作流
 - ✓ 编辑工作流

DolphinScheduler

用户名
wangshuang

密码
.....

登录



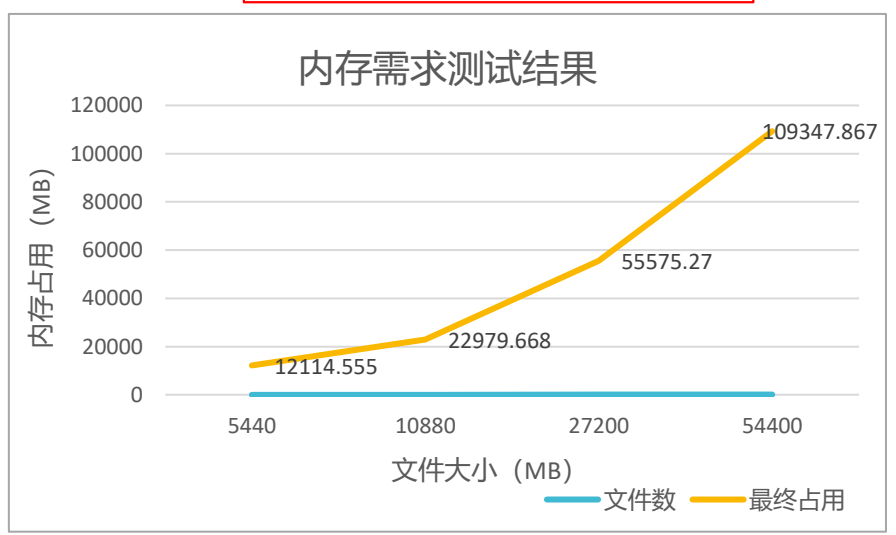


● 内存需求测试



数据分割算法

- 能谱测量X射线聚焦望远镜阵列 (Spectroscopic Focusing Array, SFA)
每个文件~544MB
- 增加数据排序
 - ✓ TIME: 接收光子时间
 - ✓ gti: 符合观测条件的观测时间



```
def execute(self, indir, outdir, outfilename, eventslist):
    with open(eventslist) as f:#
        file=f.readlines()
    file=[indir+filename.strip() for filename in file]
    hdulist=fits.open(file[0])
    priheader=hdulist[0].header
    mergedata=hdulist[1].data
    mergegti=hdulist[2].data
    dataheader=hdulist[1].header
    gtiheader=hdulist[2].header
    for i in range(1, len(file)):
        hdulist=fits.open(file[i])
        data=hdulist[1].data
        gti=hdulist[2].data
        mergedata=np.append(mergedata,data)
        mergegti=np.append(mergegti,gti)
    exposure=0
    for i in range(len(mergegti)):
        exposure=exposure+(mergegti[i][1]-mergegti[i][0])
    mergedata=np.sort(mergedata,order='TIME')
    mergegti=np.sort(mergegti,order='START')
    dataheader['EXPOSURE']=exposure
    priheader['EXPOSURE']=exposure
    prihdu=fits.PrimaryHDU(header=priheader)
    datahdu = fits.BinTableHDU.from_columns(mergedata,header=dataheader)
    gtihdu=fits.BinTableHDU.from_columns(mergegti,header=gtiheader)
    allhdulist=fits.HDUList([prihdu,datahdu,gtihdu])
    allhdulist.writeto(outdir+outfilename,overwrite=True)
    return True
```

文件数	10	20	50	100
内存/文件	2.23	2.11	2.04	2.01



目录



一、背景

二、软件框架介绍

三、软件框架进展

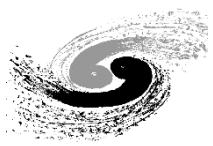
四、总结



总结



- 该框架在天文卫星数据处理应用场景中是可行的
- 初步完成多个卫星数据处理算法和工具集成
 - 数据处理算法：
 - ✓ svom: 光变曲线产品CLC、计数谱产品CSP
 - ✓ eXTP: sfa数据分割算法
 - 工具：
 - ✓ FITS文件读取和保存
 - ✓ eXTP: FITS模板生成
- 下一步, 将继续集成eXTP卫星数据处理算法 (光变曲线和能谱), 并进行全流程测试



谢谢!