

HDF数据格式及其IDL读写

激光雷达探测技术研究室

<http://lidar.aiofm.ac.cn>



概述

文件格式是指计算机存储和处理数据的方式，有的文件格式被设计用于储存特殊的数据。比如图像文件中的文件格式，多媒体文件格式等等。

在遥感与大气探测研究中我们经常使用到的文件格式有HDF格式，NetCDF格式。

HDF数据

简介

HDF(Hierarchical Data Formats)数据格式由美国国家高级计算机应用中心NCSA（National Center for Supercomputing Application）开发，中文名称为层次型数据格式，以满足不同群体的科学家在不同工程项目领域之需要。**HDF**可以存储不同类型的图像和数码数据的文件格式，并且可以在不同类型的机器上传输，同时还有统一处理这种文件格式的函数库。



实际问题

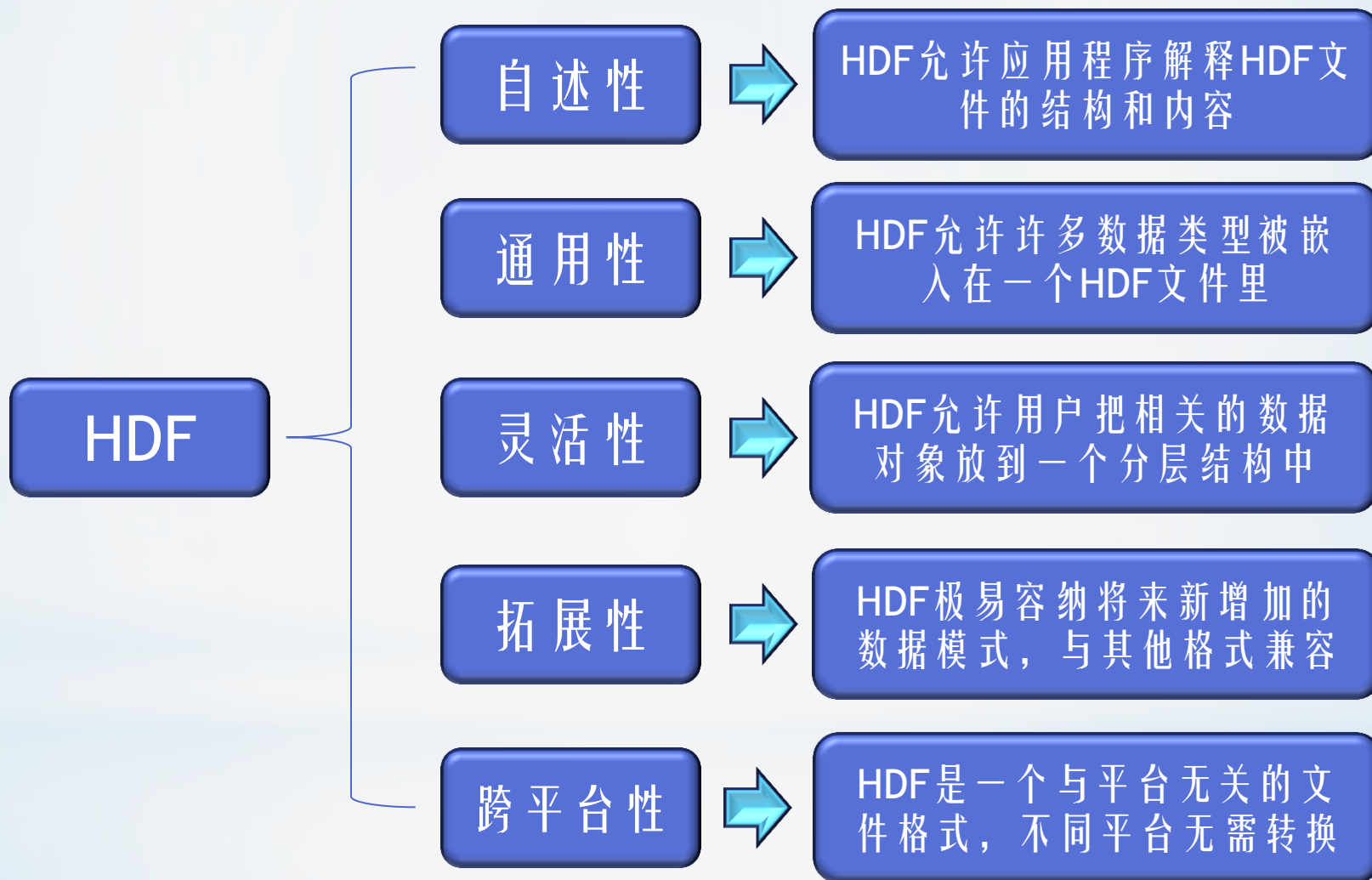
HDF产生的背景是人们通常在不同机器上建立、处理数据。在处理数据过程中，除了原始数据信息外，还会产生大量的结果、辅助、说明等信息，这些信息由于具有不同的数据格式，所以往往被存于不同的文件中。这样，在数据共享过程中，不得不利用各种软件将其打包，进行传输。即便如此，也难免会出现遗漏或出现错误现象，造成许多不必要的麻烦。



实际问题

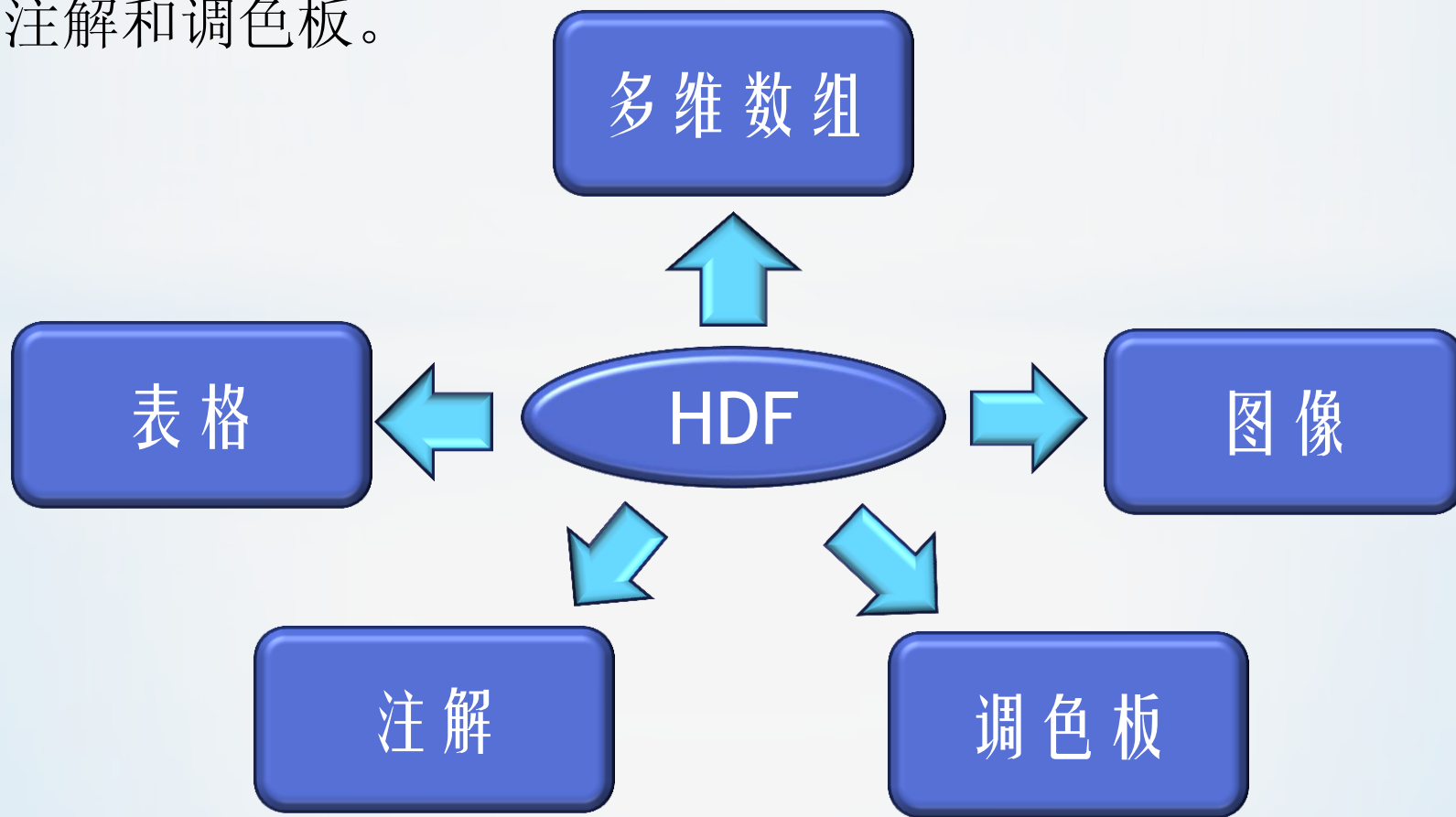
为了解决这些问题，HDF提供了一种“**总体目录结构**”。它为程序提供一种机制，使它能够直接从嵌套的文件中获得信息；可以将不同类型的数据源存于同一个文件中，而这些数据源又可以同时包含其数据信息和其他相关信息；对常用数据集的格式和描述标准化；鼓励所有机器和程序使用标准数据格式，产生包括具体数据的文件。

HDF数据



HDF数据

HDF提供大量的数据模式，包括多维数组、表格、图像、注解和调色板。





发展历史

HDF的最初产生于20世纪80年代，到现在已经具有两个不同的产品。从HDF1到HDF4的各个版本在本质上是一致的，因而HDF4可以兼容早期的版本。HDF5推出于1998年，相较于以前的HDF文件，可以说是一种全新的文件格式。它与HDF4只在概念上一脉相承，而在数据结构的组织上却截然迥异。HDF5的产生与发展反映了HDF在不断适应现代计算机发展和数据处理日益庞大复杂的要求。



发展历史

HDF5紧跟时代变化，也为其自身注入活力，使它被愈来愈多的领域采纳运用。许多遥感影像采取HDF格式存取，一些大的图像处理软件也开始提供接口读取HDF文件。HDF强大的机制适应了遥感影像的特点，能够有条不紊、完备地保存遥感影像的属性和空间信息数据，同时使查询和提取相关数据也很方便容易。



HDF4与HDF5的优缺点对比

HDF4文件由文件头，数据描述符块和数据元素组成，后两者组成数据对象。数据描述符块由若干描述符组成，它们由标识符、参照符、数据偏移量、数据长度等组成。标识符和参照数组合在一起唯一确定一个数据对象。



HDF4与HDF5的优缺点对比

HDF4不能存储多于2万个复杂对象，文件大小不能超过2G字节，其数据结构不能完全包含它的内容。随着对象的增多，数据类型也受到限制，库代码过时，过于琐碎，不能有效执行并行I/O，难于运用到线程程序中，HDF5不但能处理更多对象，存储更大的文件，支持并行I/O，线程和具备现代操作系统与应用程序所要求的其他特性。而且数据模型变得更简单，概括性更强。



HDF4与HDF5的优缺点对比

HDF5 格式运用了HDF4 和AIO 文件的某些关键思想, 它比HDF4 的自描述性更强, 它由一个超级块(super block) 、 B 树结点(B-tree node) 、 对象头(object header) 、 集合 (collection) 、 局部堆(local heaps) 和自由空间(free space) 组成。



未来发展趋势

HDF 是一种功能强大, 广泛运用于科学领域的文件格式。研究它的组织结构特别是HDF5 的组织结构对于处理和管理地理信息系统的海量图形数据和属性数据具有一定的借鉴作用。掌握和运用NCSA提供的API 提取影像数据, 可以节省时间, 提高程序编写效率。因此, HDF将会得到很广泛的应用。而HDF5由于前面所述很多优点, 已经可以在未来取代HDF4。

HDF数据



使用IDL读HDF文件

名称	功能
<code>hdf_sd_start()</code>	打开一个SDS模式的HDF文件
<code>hdf_sd_end</code>	关闭一个SDS模式的HDF文件
<code>hdf_sd_nametoindex()</code>	返回变量索引
<code>hdf_sd_select()</code>	返回变量标识符
<code>hdf_sd_getdata</code>	读取变量数据
<code>hdf_sd_endaccess</code>	结束到一个变量的通道
<code>hdf_sd_attrinfind()</code>	返回索引属性

HDF数据

名称	功能
<code>hdf_sd_attrinfo</code>	读取全局属性数据
<code>hdf_sd_fileinfo</code>	返回文件信息
<code>hdf_sd_getinfo</code>	返回变量信息
<code>hdf_sd_start ()</code>	创建一个HDF文件
<code>hdf_sd_create ()</code>	创建一个变量
<code>hdf_sd_dimgetid ()</code>	创建一个维度
<code>hdf_sd_dimset</code>	设置维度信息
<code>hdf_sd_adddata</code>	写入变量数据
<code>hdf_sd_attrset</code>	写入属性数据

HDF数据



从HDF文件中读取一个变量

HDF文件中的变量由名称引用，例如，在HDF文件 EarthProe5_31_99.hdf中，从变量“TOTAL_OZONE”。读取一个全球大气臭氧层数据的数组：

```
IDL>hdfid = hdf_sd_start('EarthProbes5_31_99.hdf')
```

```
IDL>index = hdf_sd_nametoindex(hdfid, 'TOTAL_OZONE')
```

```
IDL>varid = hdf_sd_select(hdfid,index)
```

```
IDL>hdf_sd_getdata, varid, data
```

```
IDL>hdf_sd_endaccess, varid
```

```
IDL>hdf_sd_end, hdfid
```

```
IDL>tvsc1, data
```

HDF数据



从HDF文件中读取一个属性

HDF中的属性由名称引用。例如，在HDF文件Manassas.hdf中读取属性valid_range以及海拔变量CELO: LELEVATION:

```
IDL>hdfid = hdf_sd_start('Manassas.hdf')
IDL>index = hdf_sd_nametoindex(hdfid, 'CELO:ELEVATION')
IDL>varid = hdf_sd_select(hdfid, index)
IDL>attindex = hdf_sd_attrfind(varid, 'valid_range')
IDL>hdf_sd_attrinfo, varid, attindex, data = attvalue
IDL>hdf_sd_endaccess, varid
IDL>print, attvalue
```

HDF数据



从HDF文件中读取一个属性

HDF中的属性由名称引用。例如，在HDF文件Manassas.hdf中读取属性valid_range以及海拔变量CELO: LELEVATION:

```
IDL>hdfid = hdf_sd_start('Manassas.hdf')
```

```
IDL>index = hdf_sd_nametoindex(hdfid, 'CELO:ELEVATION')
```

```
IDL>varid = hdf_sd_select(hdfid,index)
```

```
IDL>attindex = hdf_sd_attrfind(varid, 'valid_range')
```

```
IDL>hdf_sd_attrinfo, varid, attindex, data = attvalue
```

```
IDL>hdf_sd_endaccess, varid
```

```
IDL>print, attvalue
```



从HDF文件中读取一个属性

在该例中，调用函数`hdf_sd_nametoindex`获取变量CELO:
ELEVATION的索引，并使用`hdf_select`返回变量标识符`varid`。属性
`valid_range`的索引由`hdf_sd_attrfind`返回。最后，调用`hdf_sd_attrinfo`
读取属性数据。



解开HDF文件中的内容

前面所有HDF文件的例子都假定已经知道了一个特定的变量或属性在给定的文件中存在。但是，用户有时可能需要从一个预先不知道内容的HDF文件中读取数据。在这种情况下，首先要做的事探测文件内部包含哪些内容。然后才能决定哪一部分是你需要读取的。为了解开HDF文件中的内容，可以调用调查程序来决定文件中包含的所有变量和属性的名称、类型和大小。以函数`hdf_sd_vadir.pro`为例，当给定一个文件的标识符`hdfid`后，将返回一个字符串数组，该数组包含文件中的所有变量的名称。如果文件中没有变量，则返回空字符串。

HDF数据



解开HDF文件中的内容

```
FUNCTION HDF_SD_VARDIR, HDFID
```

```
;-Check arguments
```

```
If(n_params() ne 1)then $
```

```
    message, 'Usage: RESULT = HDF_SD_VARDIR(HDFID)'
```

```
If(n_elements(hdfid) eq 0)then $
```

```
    messsage, 'HDFID is undefined'
```

```
;-Set default return value
```

```
varnames= “
```

HDF数据



解开HDF文件中的内容

;-Get file information

hdf_sd_fileinfo, hdfid, nvars, ngatts

;-if variables were found, get variable names

if (nvars gt 0) then begin

varnames = strarr(nvars)

for index = 0L, nvars - 1L do begin

varid = hdf_sd_select(hdfid, index)

hdf_sd_getinfo, varid, name = name

HDF数据



解开HDF文件中的内容

```
hdf_sd_endaccess, varid
```

```
varnames[index] = name
```

```
endfor
```

```
endif
```

```
;-Return the result
```

```
return, varnames
```

```
END
```



解开HDF文件中的内容

在该例中，当检测完输入参数后，调用函数`hdf_sd_fileinfo`返回包含文件中变量和全局属性的数目，如果变量的数目大于0，则创建字符串数组`varnames`来存放变量名称。然后开始一个循环操作，调用`hdf_sd_select`来选择一个变量，并用`hdf_sd_getinfo`获取变量的名称。同时使用`hdf_sd_endaccess`关闭通向变量的路径。每个变量的名称存储在数组`varnames`中。最后，返回结果。

HDF数据



写入HDF文件

若将变量写入到HDF文件中，则必须先定义该变量的名称、大小和类型，然后再写入需要的数据。在下面的例子中，将例子图像ctscan.dat的内容写入到一个HDF文件中。首先，从ctscan.dat中读取数据：

```
IDL>file=filepath('ctscan.dat',subdir='examples/data')
```

```
IDL>openr,lun,file,/get_lun
```

```
IDL>data=bytarr(256,256)
```

```
IDL>readu,lun,data
```

```
IDL>free_lun,lun
```

HDF数据



写入HDF文件

然后创建一个新的HDF文件，定义变量属性和一个全局属性写入文件中，并关闭文件：

```
IDL>hdfid = hdf_sd_start('ctscan.hdf',/create)
```

```
IDL>varid = hdf_sd_create(hdfid,'image',[256,256],/byte)
```

```
IDL>dimid = hdf_sd_dimgetid(varid,0)
```

```
IDL>hdf_sd_dimset, dimid, name='xdim'
```

```
IDL>dimid=hdf_sd_dimgetid(varid,1)
```

```
IDL>hdf_sd_dimset, dimid, name='ydim'
```

HDF数据



写入HDF文件

最后将数据以及一个变量属性和一个全局属性写入文件中，并关闭文件：

```
IDL>hdf_sd_adddata, varid, data
```

```
IDL>hdf_sd_attrset, varid, 'long_name', $
```

```
IDL> 'CT scan from IDL examples'
```

```
IDL>hdf_sd_attrset, hdfid, 'creation_data', systime()
```

```
IDL>hdf_sd_endaccess, varid
```

```
IDL>hdf_sd_end, hdfid
```

HDF数据



写入HDF文件

在该例中，文件通过调用函数hdf_sd_start创建。名为image的变量由函数hdf_sd_create创建，该变量为字节型 256×256 数组（如果在调用hdf_sd_create时没有设置数据类型关键字，如/byte，则默认情况下的数据类型为浮点型）。维度通过调用hdf_sd_dimgetid和hdf_sd_dimset创建。而数据通过调用hdf_sd_adddata写入。同时，加入变量属性long_name以描述image的内容，加入全局属性creation_date以记录文件创建的日期和时间。

HDF数据



坐标变量

我们先创建了一个包含大气温度垂直分布的新文件，该内容在另一张讲述NetCDF的PPT中也有涉及：

```
IDL>hdfid = hdf_sd_start('profile.hdf',/create)
```

```
IDL>varid = hdf_sd_create(hdfid, 'Temperature', [10],/float)
```

```
IDL>dimid = hdf_sd_dimgetid(varid,0)
```

```
IDL>hdf_sd_dimset, dimid, name='Pressure'
```

```
IDL>data = 300.0-2.5*findgen(10)
```

```
IDL>hdf_sd_adddata, varid, data
```

```
IDL>hdf_sd_attrset, varid, 'units', 'K'
```



坐标变量

为了检测一个给定的变量是否为坐标变量，可以调用 `hdf_sd_iscoordvar` 函数。如果变量为坐标变量，则传递时变量的标识符 `hdf_sd_iscoordvar` 返回真（1），否则返回假（0）。

HDF数据



坐标变量

然后创建一个名为**Pressure**的坐标变量，其维度仍使用**Pressure**的维度。该变量用来存储每个大气高度的实际大气压值：

```
IDL>varid = hdf_sd_create(hdfid, 'Pressure', [10],/float)
```

```
IDL>dimid = hdf_sd_dimgetid(varid,0)
```

```
IDL>hdf_sd_dimset, dimid, name='Pressure'
```

```
IDL>hdf_sd_adddata, varid, data
```

```
IDL>hdf_sd_attrset, varid, 'units', 'hPa'
```

```
IDL>hdf_sd_endaccess, varid
```

```
IDL>hdf_sd_end, hdfid
```

谢谢！

如有问题可与孙路联系，电子邮件：louissun2009@gmail.com