



RED ESPAÑOLA DE  
SUPERCOMPUTACIÓN



**Barcelona  
Supercomputing  
Center**  
*Centro Nacional de Supercomputación*

**Barcelona Supercomputing  
Center**

**Centro Nacional de  
Supercomputación**

*Seminario Computación Paralela*

Santander, 5 de noviembre de 2010

**Jorge Rodríguez**  
User Support, BSC



- **netCDF**

1. Introducción
2. Características
3. Evolución
4. Modelo de datos
5. API
6. Ejemplo

- **Parallel-netCDF**

1. Introducción
2. Acceso a ficheros
3. Caso real

- **Conclusiones**



## ¿Qué es netCDF?

- Conjunto de librerías software
- Formato científico de ficheros
- Primera versión en 1988
- Orientado a datos vectoriales
- Independiente de la plataforma
- API para manejo:
  - C
  - C++
  - Fortran
  - Java, MatLab





## Propiedades

Los datos netCDF son:

- **Autodescriptivos:** Contiene información acerca de los datos que contiene.
- **Portables:** Un archivo netCDF puede accederse por ordenadores que almacenen los números (reales o enteros) de diferente manera.
- **Escalables:** Se puede acceder a un pequeño conjunto de un gran dataset eficientemente.
- **Aragables:** Se puede añadir datos a estructuras existentes sin copiar ni redefinir todo el dataset.
- **Compartibles:** Un proceso escritor y varios lectores pueden acceder a los datos de un netCDF simultáneamente.
- **Archivables:** El acceso a versiones antiguas de netCDF está soportado.



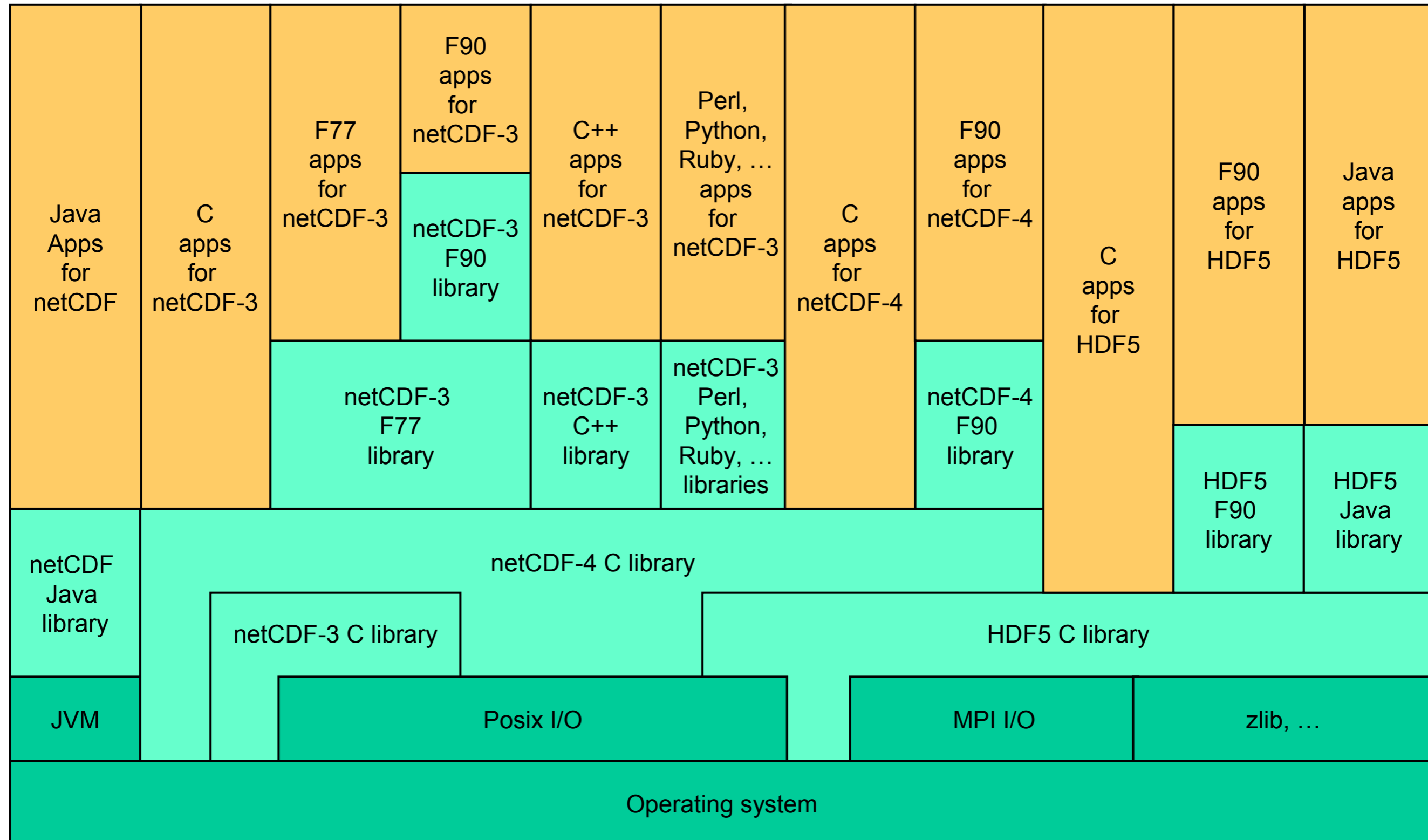
## Backward compatibility

- **Acceso a los datos:** Nuevas versiones de netCDF permitirán lectura y escritura de cualquier versión previa.
- **APIs y programas:** Código existente en C, Fortran o Java será soportado por nuevas versiones de netCDF (recompilación)



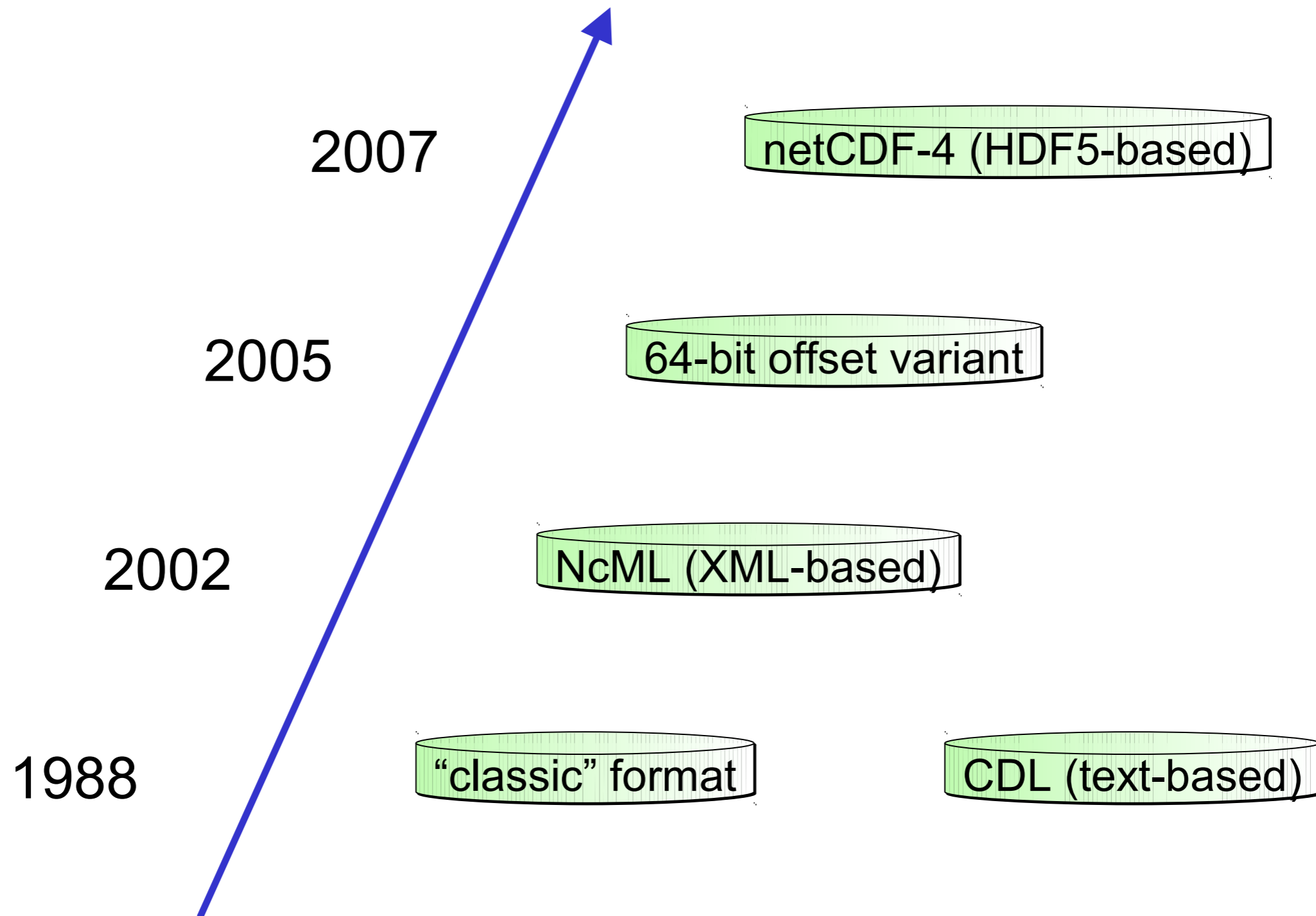


## ¿A qué nivel se encuentra netCDF?





## Evolución de los formatos netCDF





## Evolución de los formatos netCDF

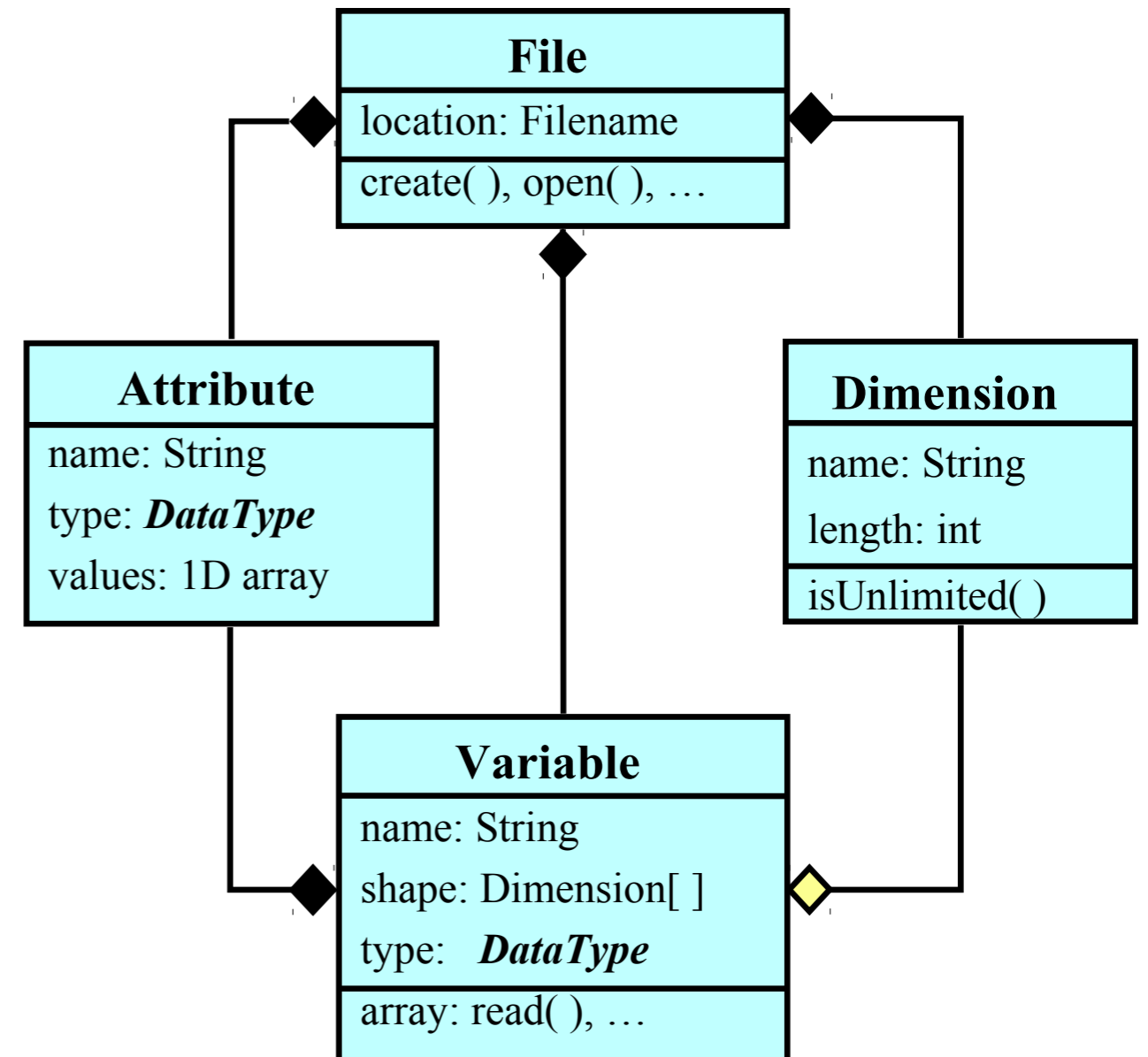
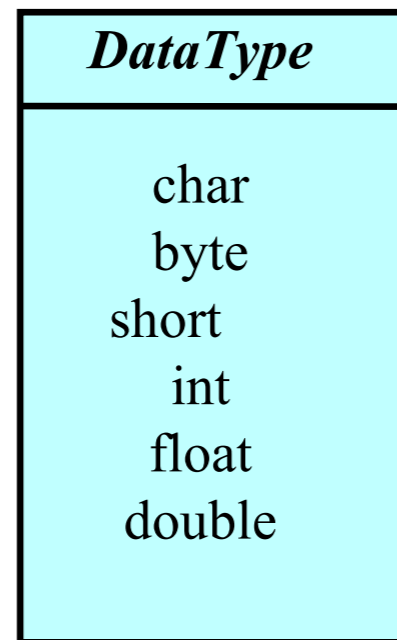
- **Classic format:** Formato binario original de netCDF. Es el formato por defecto para crear archivos. Soportado en la actualidad.
- **64-bit offset format:** Introducido en la versión 3.6.0 (2005). Este formato es similar al clásico pero con límites menos estrictos.
- **netCDF-4/HDF5 format:** Introducido en 2008, se trata de ficheros HDF5 que pueden ser accedidos por programas netCDF (y HDF5).





## Modelo de datos de netCDF

- **Classic format:**
  - Variables
  - Dimensiones
  - Atributos



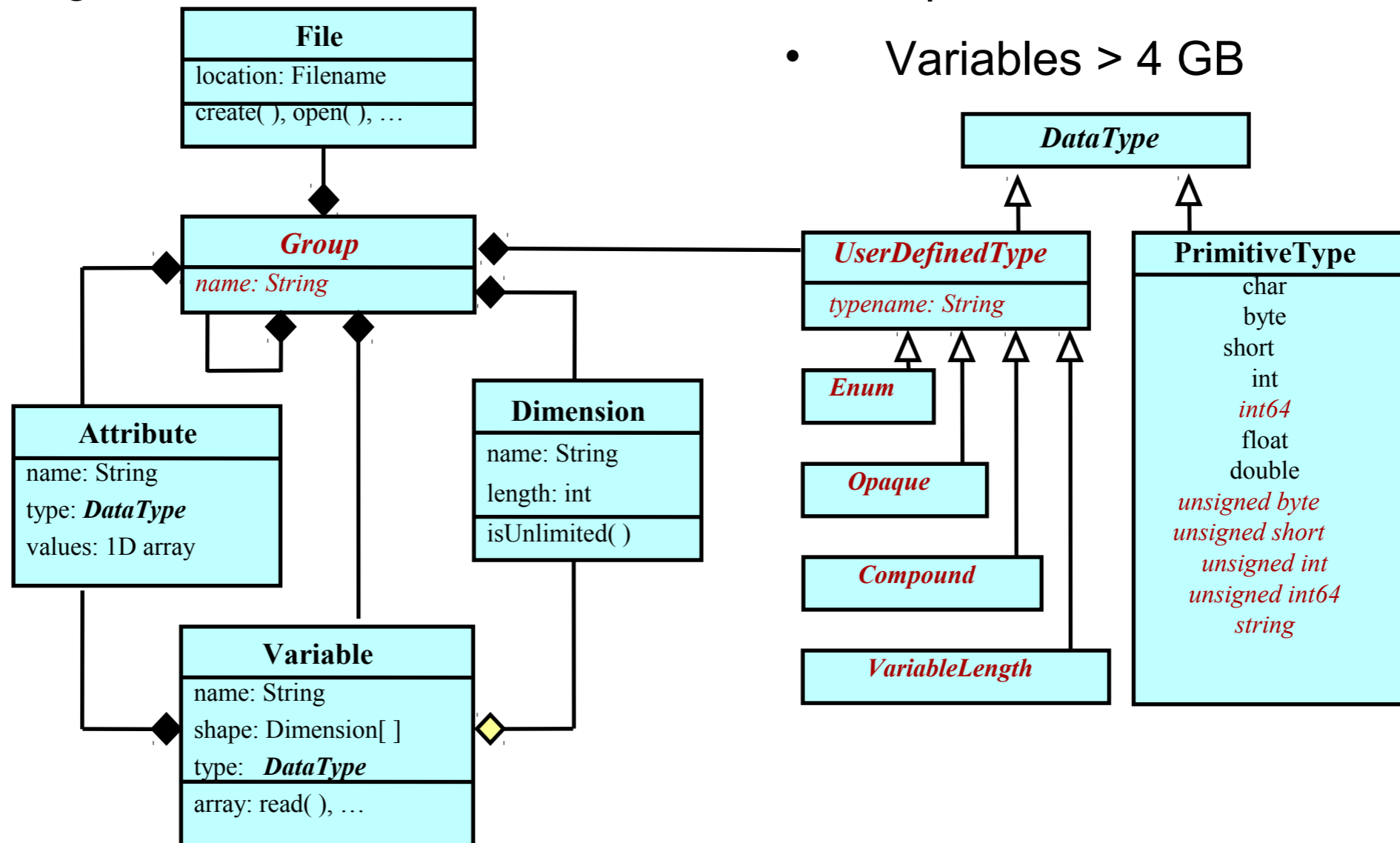


## Modelo de datos de netCDF

- **netCDF-4:**

- Dimensiones expandibles
- Strings

- Enteros de 64 bits
- Tipos enumerados
- Variables > 4 GB





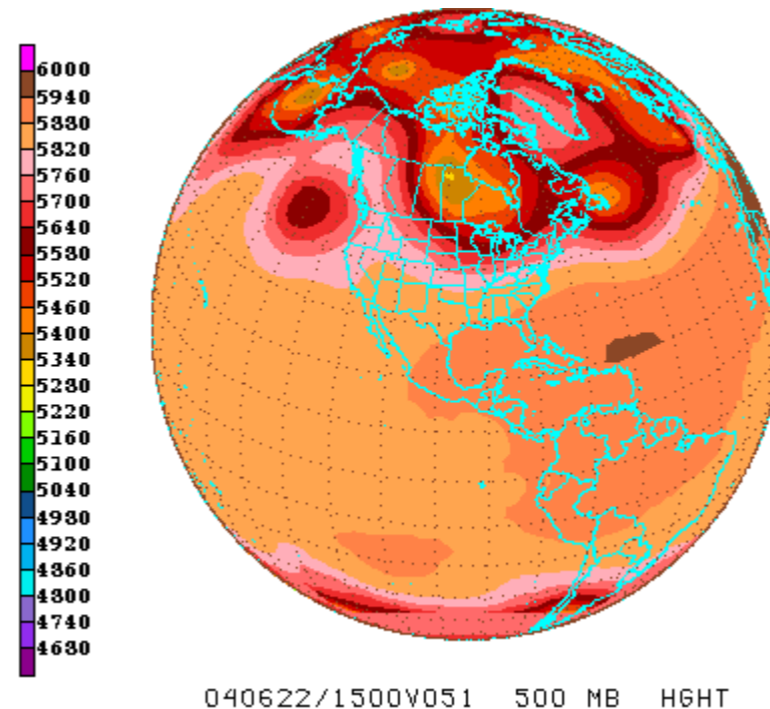
## Diferencias entre Classic Format y netCDF-4

- Escasez de estructuras de datos: arrays multidimensionales y listas
- Sin estructuras anidadas
- Vectores de caracteres
- Conjunto pequeño de datos numéricos
- Sin compresión de datos
- Modificación del esquema de datos muy ineficiente
- Ineficiencia al portar netCDF entre plataformas Big-endian y Little-endian



## API para diferentes lenguajes

- netCDF está desarrollado y mantenido en C
- APIs de C++, Fortran 77, Fortran 90, Perl: *wrapper* de la API de C
- La API para Java puede leer, pero no escribir archivos netCDF-4
- La API de C++ puede leer y escribir netCDF pero sólo en Classic Format.





## #include <netcdf.h>

- Crear un fichero netCDF:

```
nc_create("short_file.nc", NC_CLOBBER, &ncid);
```

- Abrir un archivo netCDF en modo lectura:

```
nc_open("mydata.nc", NC_NOWRITE, &ncid);
```

- Cerrar el fichero:

```
nc_close(ncid);
```

- Definir una variable:

```
nc_def_var(ncid, "temp", NC_FLOAT, NDIMS, dimids, &varid);
```

- Definir atributo a una variable:

```
nc_put_att_text(ncid, varid, "units", strlen(CELSIUS), CELSIUS);
```



```
#include <netcdf.h>
```

- Obtener el identificador de la variable “windspeed”:

```
nc_inq_varid(ncid, “windspeed”, &varid);
```

- Leer un conjunto de valores:

```
nc_get_vara_float(ncid, varid, start, count, valarray);
```

- Escribir un valor:

```
nc_put_var_float(ncid, varid, (float *)data);
```



## ¿Qué es Parallel-netCDF?

- Librería que ofrece E/S de alto rendimiento para ficheros netCDF
- Tradicionalmente, la lectura y escritura de netCDF se hace de modo serie
- Con Parallel-netCDF se utilizan funciones optimizadas de MPI-IO





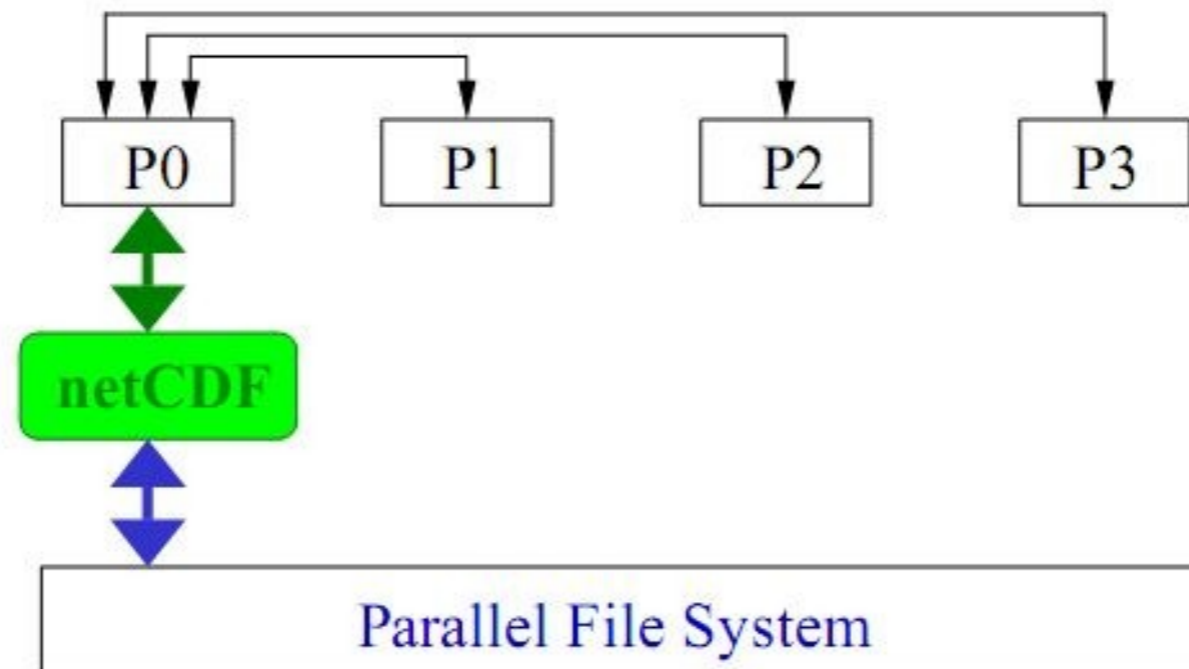
## ¿Por qué Parallel-netCDF?

- La mayoría de las aplicaciones científicas son paralelas
- Aumento significativo del volumen de datos
- Cuello de botella con E/S
- La escalabilidad necesita E/S en paralelo
- Tradicionalmente, la lectura y escritura de netCDF se hace de modo serie
- Con Parallel-netCDF se utilizan funciones ya optimizadas de MPI2





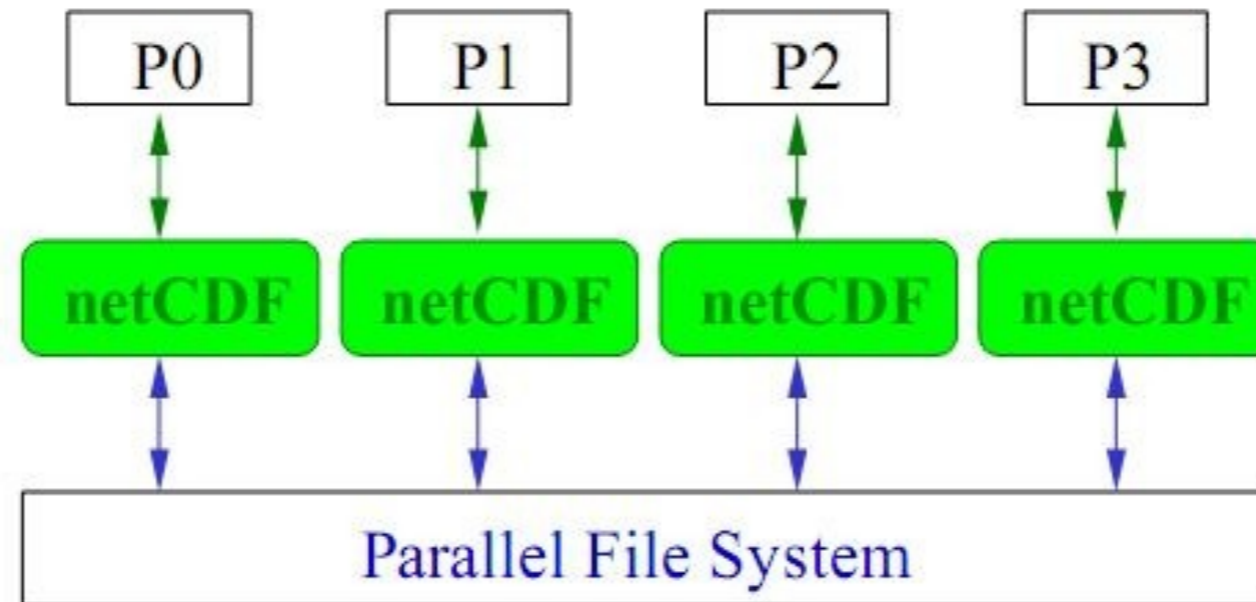
## Acceso a ficheros



- Acceso lento
- Movimiento de datos
- Cuello de botella con E/S
- Requisitos de memoria



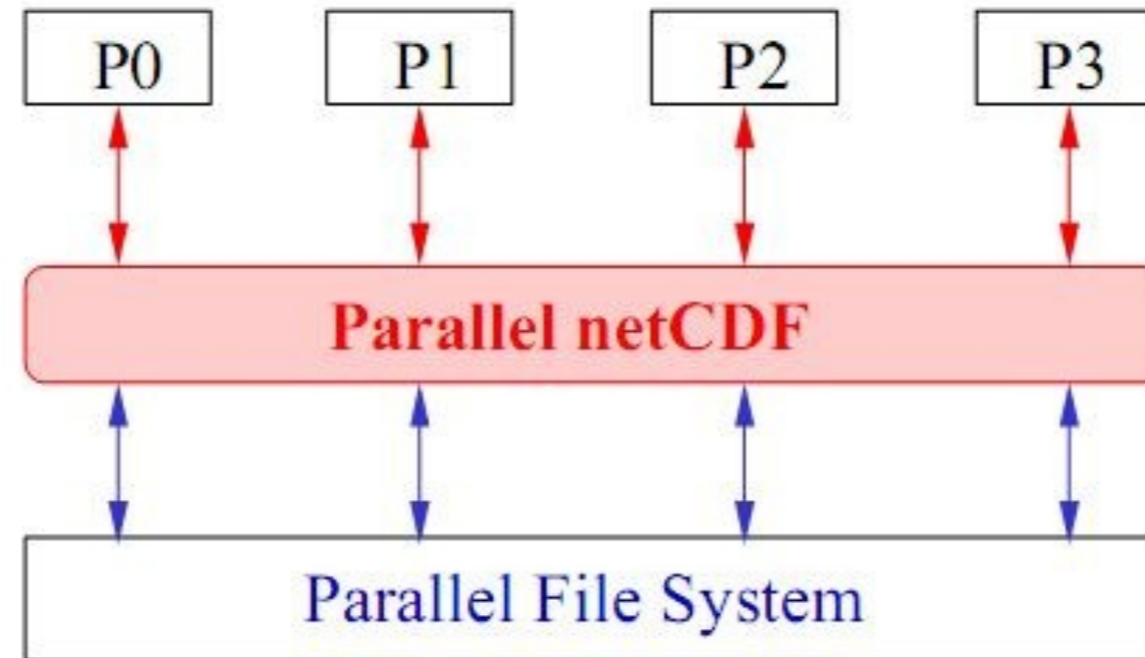
## Acceso a ficheros



- Numerosos archivos netCDF
- Acceso independiente



## Acceso a ficheros

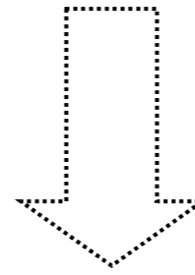


- Interfaz paralela entre procesos y sistema de ficheros
- E/S colectiva y cooperativa
- Gestión del formato netCDF

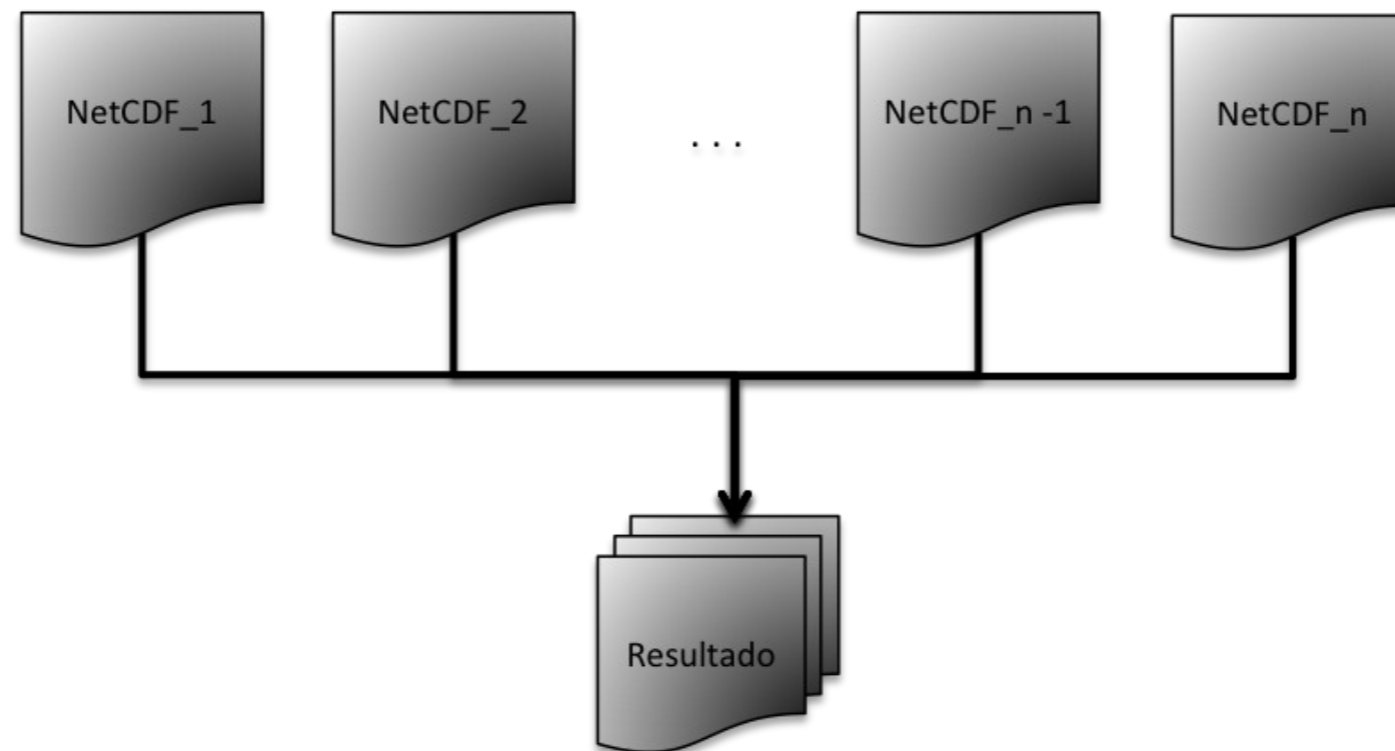


## Problema

- Tenemos 48 ficheros netCDF (~1 GB), y queremos unirlos en uno sólo.



## Fusión de ficheros netCDF en paralelo





## Solución

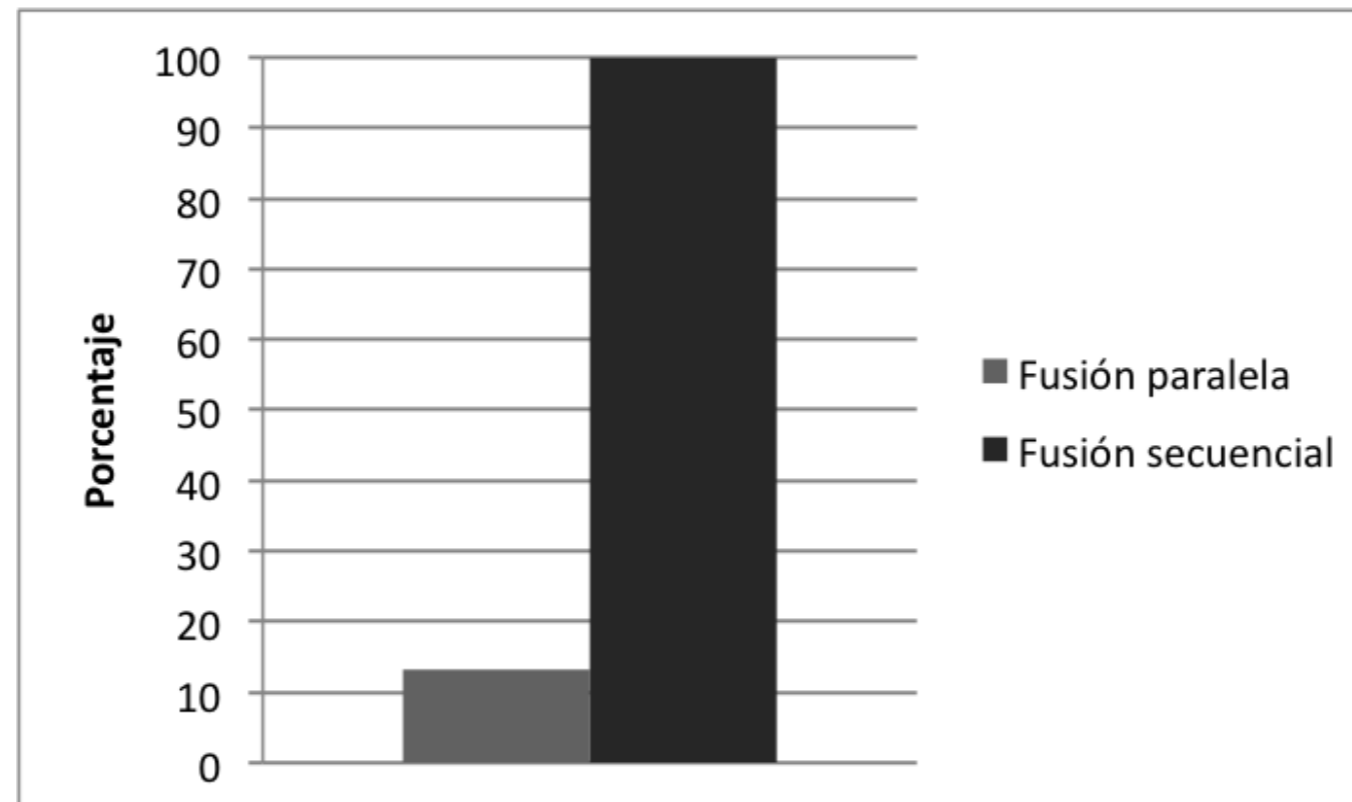
- Utilizamos Parallel-netCDF
- Cada proceso lee un fichero netCDF diferente
- Todos los procesos escriben cooperativamente, el mismo fichero resultado

```
sprintf(filename,"MET_BDY_3D_ %04d",mpirank);
ncmpi_open(comm,filename,NC_NOWRITE,MPI_INFO_NULL,&nc2_id);
...
for(vrbl=0;vrbl<VAR_len;vrbl++){
    ...
    ncmpi_get_var_float(nc2_id,varid2,VARIABLE);
    ncmpi_put_vara_float(ncid,varid[vrbl],start, count,VARIABLE);
}
```



## Análisis

- Con la versión secuencial previa: 53 minutos
- Con la nueva versión implementada: 7 minutos



- Se obtiene un speed-up de 7x



- NetCDF es un modelo de datos sencillo.
- El modelo y la API de NetCDF ofrecen un modo portable de almacenar y acceder datos multidimensionales.
- La extensión Parallel-netCDF permite una lectura y escritura de ficheros en paralelo muy eficiente.



Gracias por vuestra atención

