ROOT I/O Status Developments

BNL April **2000**

René Brun/CERN

Ideal Persistency

No constraints on object model

Automatic schema evolution

Remote access

Automatic converters

transient

persistent

LAN
WAN

Granularity matching access patterns

Efficient storage compression

obj1;1, obj1;2,obj1;3
obj2;1, obj2;2

Machine independent format

ROOT I/O -- *Sequential/Flat*

Transient Object
is serialized
by the Streamer
No need for transient/persistent classes

Streamer

Object in memory

Object in memory

TMapFile
shared memory

TRFIOFile
RFIO daemon

ObjectGram
TBuffer

http

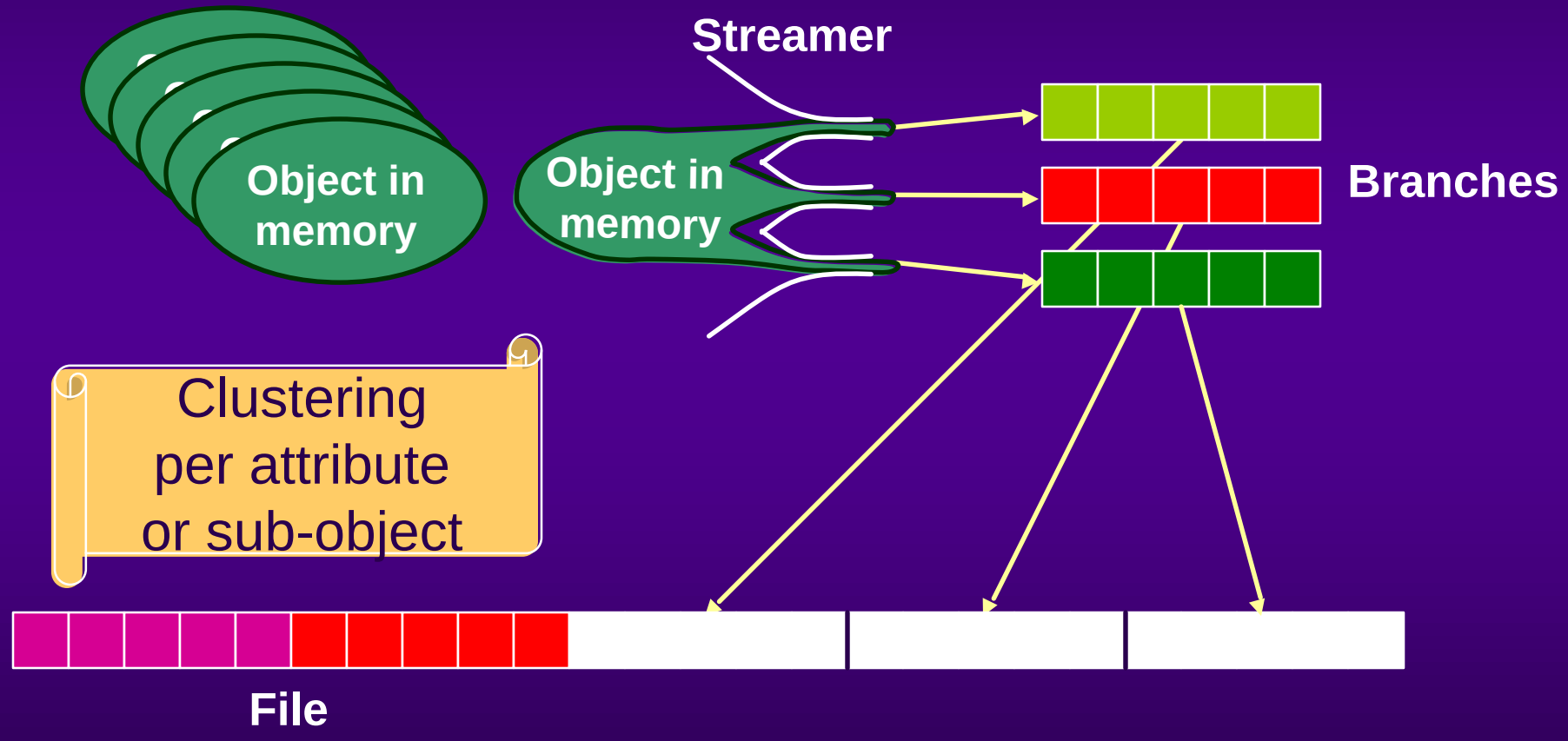
TWebFile
web server

sockets

TNetFile
rootd

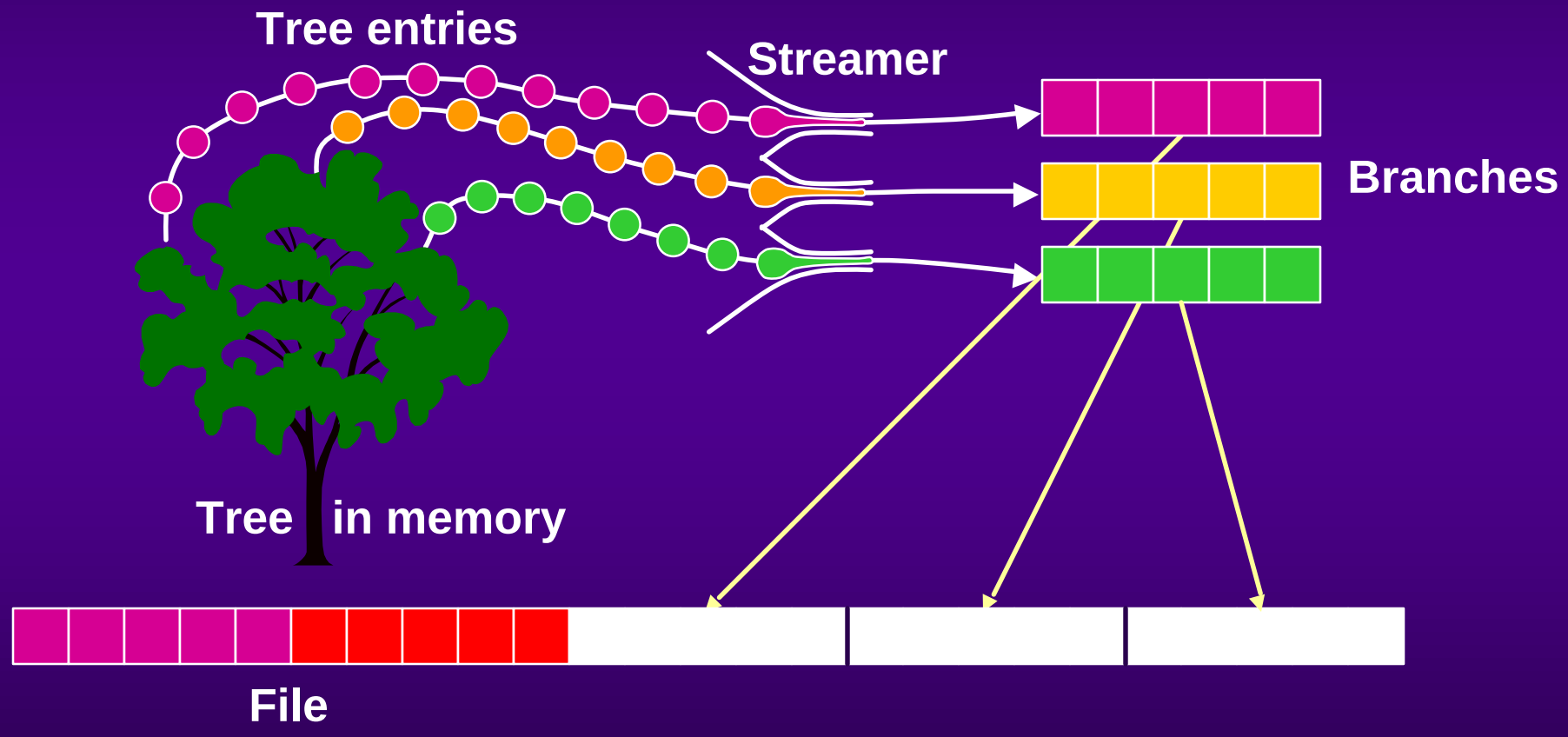
TFile

ROOT I/O -- *Split/Cluster*

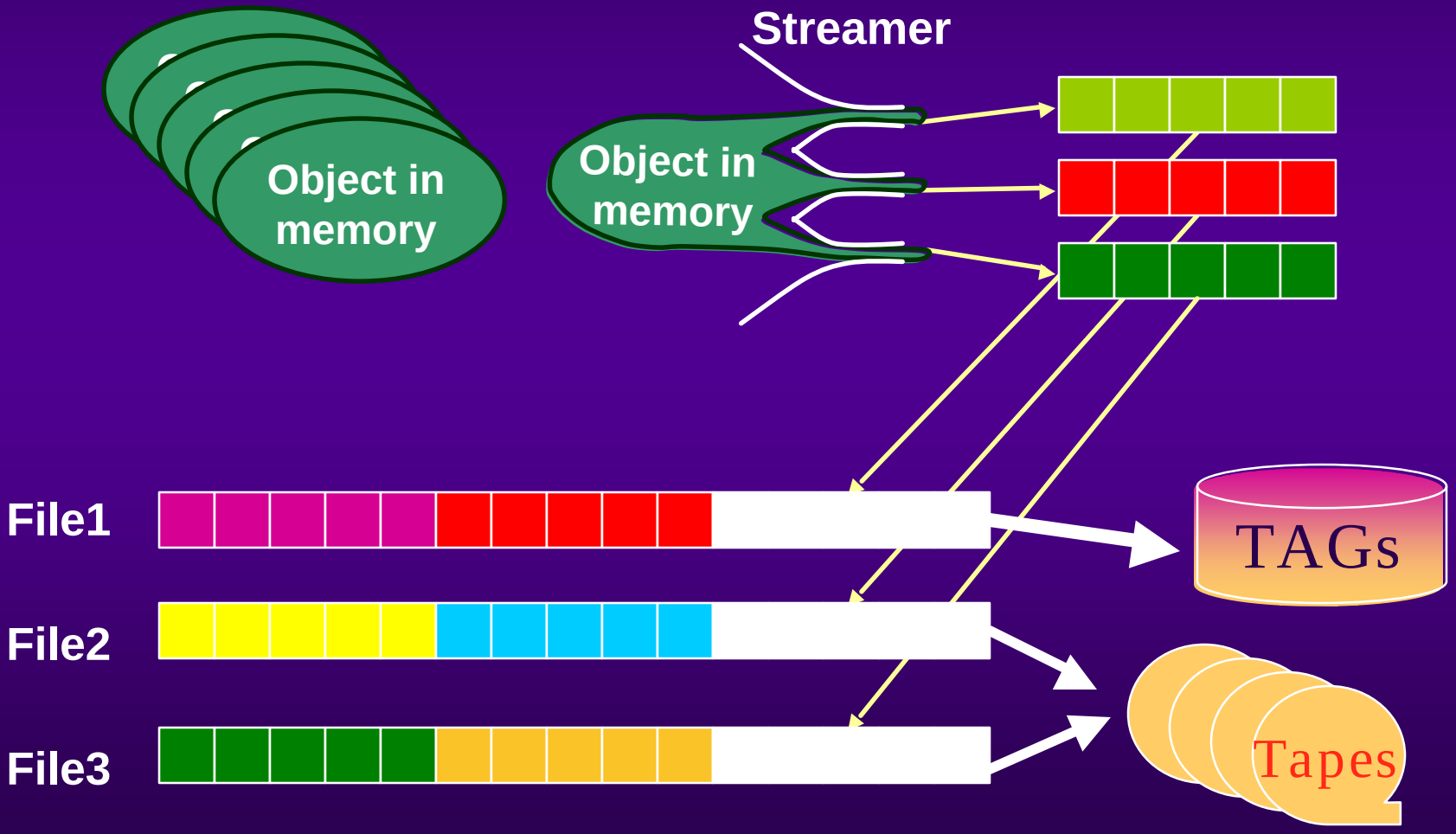


ROOT I/O -- *Split/Cluster*

Tree version



ROOT I/O - *Split - multifile*



Input/Output

- ByteCount implementation (set & check)
- Default for all Root classes in 2.23/12
- New option "+" in LinkDef to get it (important)
  - #pragma link C++ class TPad-;
  - #pragma link C++ class TPaveClass+;
- Preparation for self-describing object format
- The next slides describe the StreamerInfo system that will be included in Root version 2.24/xx

Code Generated automatically by rootcint

```
void TArrow::Streamer(TBuffer &R__b) {
   UInt_t R__s, R__c;

   if (R__b.IsReading()) {
      Version_t R__v = R__b.ReadVersion(&R__s, &R__c);
      TLine::Streamer(R__b);
      TAttFill::Streamer(R__b);
      R__b >> fAngle;
      R__b >> fArrowSize;
      fOption.Streamer(R__b);
      R__b.CheckByteCount(R__s, R__c, TArrow::IsA());
   } else {
      R__c = R__b.WriteVersion(TArrow::IsA(), kTRUE);
      TLine::Streamer(R__b);
      TAttFill::Streamer(R__b);

      R__b << fAngle;
      R__b << fArrowSize;
      fOption.Streamer(R__b);
      R__b.SetByteCount(R__c, kTRUE);
   }
}
```

I/O Object format descriptor

- We are now implementing a new facility to store the name and types of data written via Streamer.
- A new function const char*TObject::StreamerInfo returns a default value corresponding to the default Streamer function generated by rootcint.
- This function can be redefined in classes implementing manual Streamers
- The introduction of this new facility is backward compatible.

StreamerInfo example

- return "TNamed;TAttLine;fX;fY;fSize";
- return "TObject;fA;fB;Int_t a,b,c;Float_t x,y,z";
- In case of existing class data members (basic types or classes), just give the list of names.
- In case of new data, give the type followed by the list of basic types, or classes.
- Support for arrays: (rootcint upgrade)

```
Int_t       fN;
Float_t *fX ;    //[fN] x coordinates
Float_t *fY;     //[fN] y coordinates
```

StreamerInfo cntd

- The StreamerInfo is saved into a StreamerInfoList object that includes the StreamerInfos for all classes in a Root Tree or Root file in general.
- In case of a TTree, the StreamerInfoList is a new member of the class and saved with theTree header.
- In case of a file, the same list is a member of the TDirectory class and saved when the file is closed.
- The StreamerInfo can be used to read objects or Trees when the original user classes are not available.

StreamerInfo cntd2.

- The StreamerInfo will be used in future versions by:
  - the Root browsers
  - the TreeViewer
  - MakeClass
- It could also be used for automatic Streamers to provide an automatic class schema evolution support.
- We intend to test the performance of these automatic Streamers compared to the current Streamers.
- Could also be used by foreign readers.