

Proxy classes for logical & physical volumes

J. Apostolakis

SFT R&D Dev Meeting Tuesday 28 April 2020



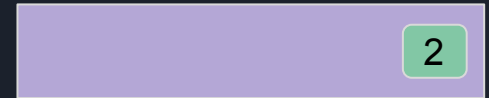
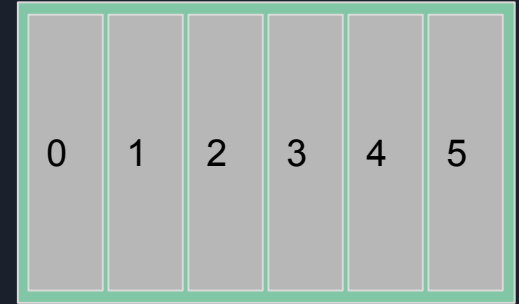
Overview

Key objects in G4 geometry modified during navigation - even in sequential mode

- Replicas have Replica Number updated with the slice number
- In a Parameterised Volume many attributes can change - user code gives
 - Solid type and parameters (e.g. box length, width, ...)
 - Material (pointer)
- Others must be derived (if needed): Material Cuts Couple

Multi-threading adds state that differs between events / threads / tasks

- Sensitive Detector (thread / task)
- Classes with state of track integration in field (Field, FieldManager, ...)



G4Replica



Geant4 MT design

For MT we split some object's state into 'invariant' and 'changeable' parts:

- G4LVState= part of G4LogicalVolume that changes per task / thread
- G4PVState= part of G4PhysicalVolume that changes
- ..

The original object has an index (92nd LogicalVolume) into per-thread data structure (vector). The pointer to G4(LV)Data is in TLS storage.

Accessing a track's logical volume needed the thread index

- Modulo some recent optimisation

G4LogicalVolume



G4LogicalVolume



92

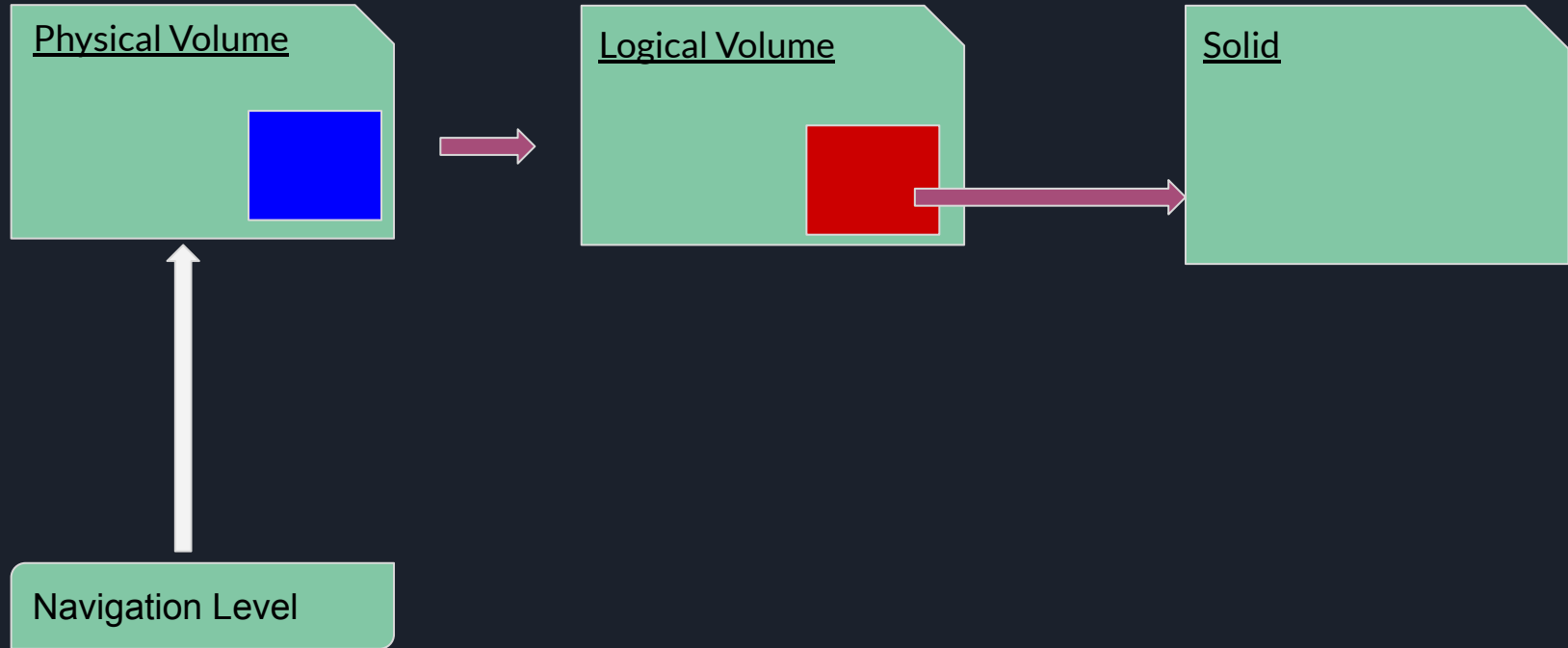




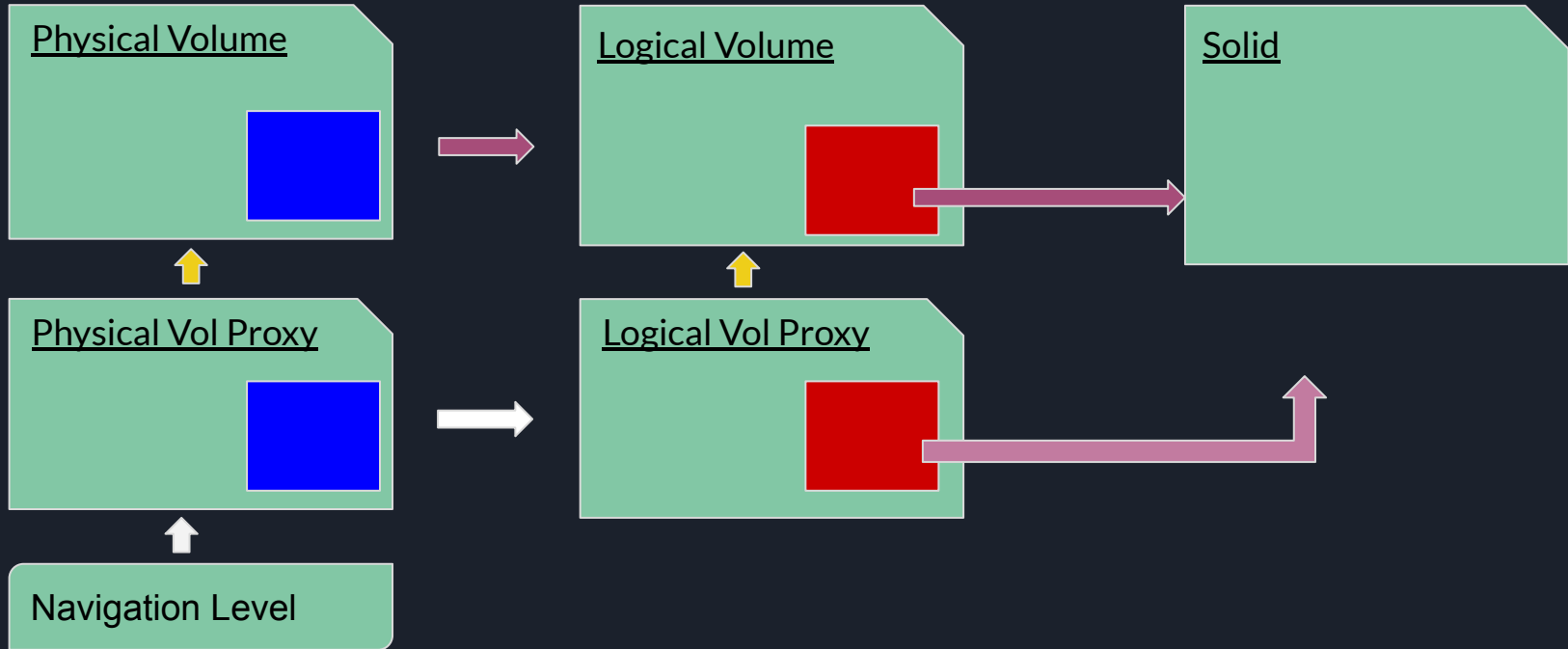
Understanding the problems

- 01 Sequential Geant4 used to mix invariant data (read-only) with changable data in the same cache line
- 02 Geant4-MT separates the invariant data in memory, but introduces extra indirection(s) and the need to access the thread index in order to retrieve (some) key data
- 03 The Geant4 API provides access to all the G4 geometry classes for the user to retrieve any relevant information during tracking -- in addition to our preferred class `G4TouchableHistory`

Current design



New idea: proxies to hold changing data





Typical information sought during tracking

Obtain properties

```
touchableHistory->GetPhysicalVolume()->GetLogicalVolume()->GetSolid() or GetMaterial()
```

```
G4VPhysicalVolume pv= touchableHistory->GetPhysicalVolume();
```

```
G4int repNo= pv->GetReplicaNumber();
```

Identity checks (in user's Sensitive Detector and Action classes)

```
if( touchableHistory->GetPhysicalVolume()->GetLogicalVolume() == targetLogical )
```

```
if( pv== targetPhysical )
```



Goals of prototype

Implement proxy classes for Logical and Physical Volume and integrate them into the G4 Geometry modeller.

Evaluate whether such a significant redesign (refinement/revision) can

- Eliminate or reduce the overhead due to TLS in the Geant4 Geometry during tracking
- Be achieved with a minimal impact on user code
- Provide (modest) CPU performance improvement for cases with TLS overhead when G4 is used with shared libraries & TLS
- Evaluate whether the code becomes more 'fragile' after the changes.



Status of prototype

Created proxy classes for Logical and Physical Volume, than respond to all* their methods

Create a Touchable History that uses the proxies in place of PhysicalVolume ptr

Revised G4Navigator: update the information in proxies (keeping 'live updated' TLS state too.)

Calls to TouchableHistory's access methods now check between proxy & 'live updated' geom.

Creating 'handle' class to address 'identity' use case

Replace {touchable,step,track} GetVolume() with Get Physical Volume/ Handle (underway)

Check that G4 library, key tests, examples compile / run correctly - for now with lots of changes.

Extensive testing ... then benchmarking.