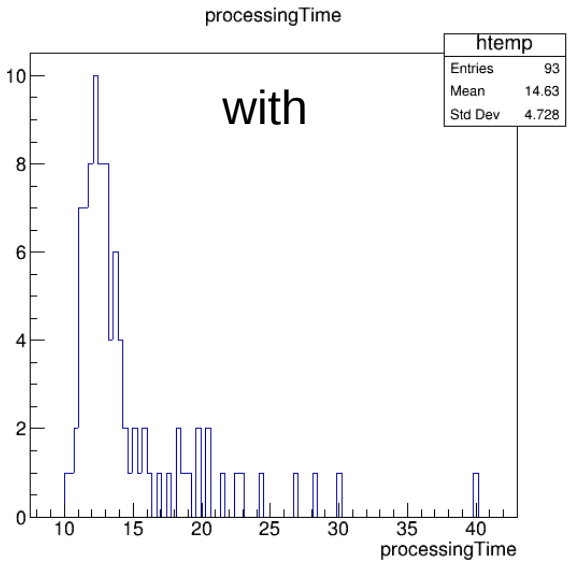
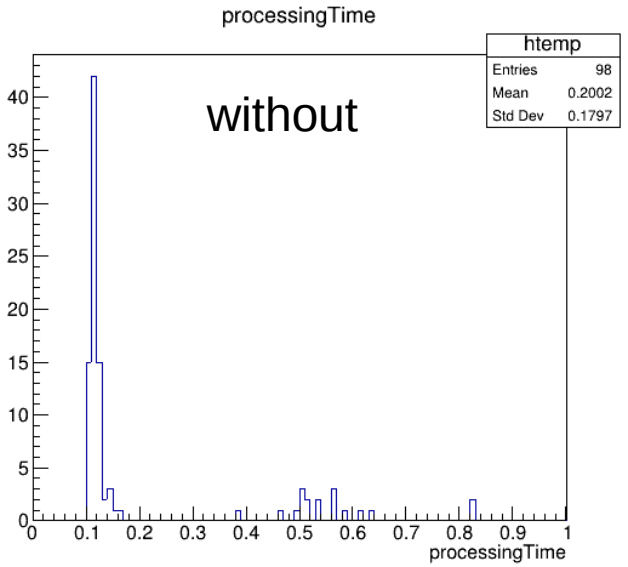
Increase Run Time due to Boolean Solids





ECAL1:

```
    for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 20; j++) {
//      OlgaPart_sub = new G4SubtractionSolid("olga_sub", OlgaPart_sub, HoleBox, 0,
//      G4ThreeVector(-4*2*OlgaModuleSize + i*2*OlgaModuleSize + OlgaModuleSize, -10*2*OlgaModuleSize + j*2*OlgaModuleSize + OlgaModuleSize, 0));
    }
    }
    for (int i = 0; i < 22*2; i++) {//0-22*2
    for (int j = 0; j < 38*2; j++) { //0-38*2
//      MainzPart_sub = new G4SubtractionSolid("mainz_sub", MainzPart_sub, HoleBox, 0,
//      G4ThreeVector(-11*2*MainzModuleSize + i*MainzModuleSize + MainzModuleSize/4, -19*2*MainzModuleSize + j*MainzModuleSize + MainzModuleSize/4, 0)
    }
    }
```

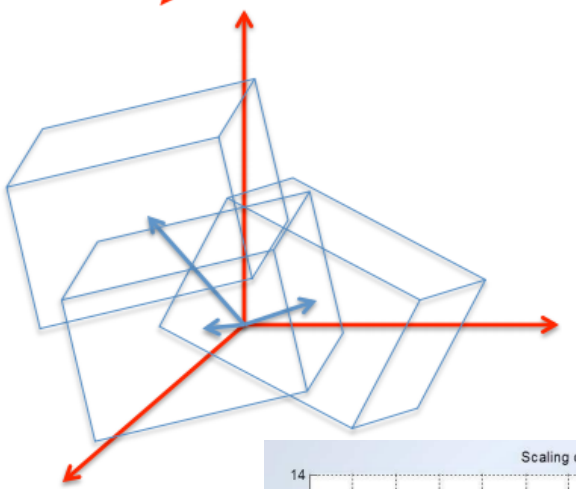
ECAL2:

```
  G4ThreeVector(-32*2*ModuleSize + ModuleSize, -24*2*ModuleSize + ModuleSize, 0));
    for (int i = 0; i < 64; i++) { //0-64
    for (int j = 0; j < 48; j++) { //0-48
//      fiberPlate_sub = new G4SubtractionSolid("fiber_sub", fiberPlate_sub, fiberHoleBox, 0,
//      G4ThreeVector(-32*2*ModuleSize + i*2*ModuleSize + ModuleSize, -24*2*ModuleSize + j*2*ModuleSize + ModuleSize, 0));
    }
    }
```

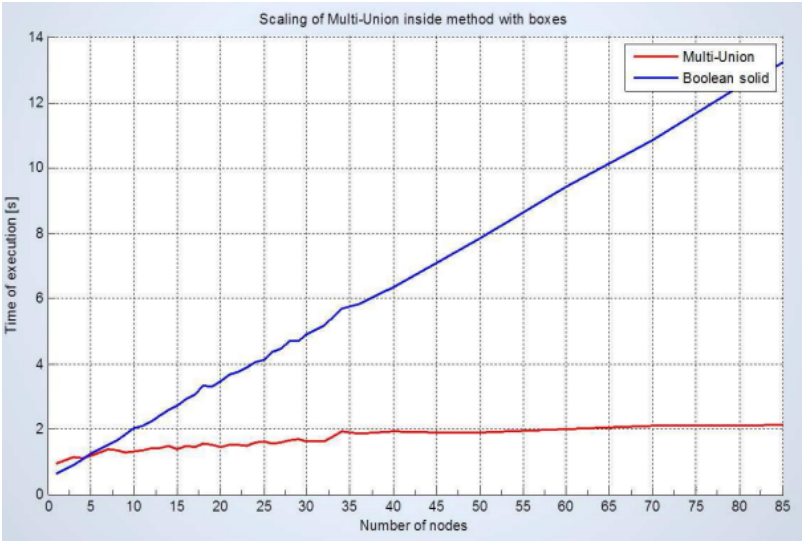
Multi-Union vs. Boolean Solids


```
G4MultiUnion* munion_solid = new G4MultiUnion("UnitedBoxes");

for( int i=0 ; I < nNode ; i++)
{
  G4Box* aBox = new G4Box(…);
  G4ThreeVector pos = G4ThreeVector(…);
  G4RotationMatrix rot = G4ThreeVector(…);
  G4Transform3D tr = G4Transform3D(rot, pos);
  munion_solid -> AddNode( *aBox, tr );
}

munion_solid -> Voxelize();
```

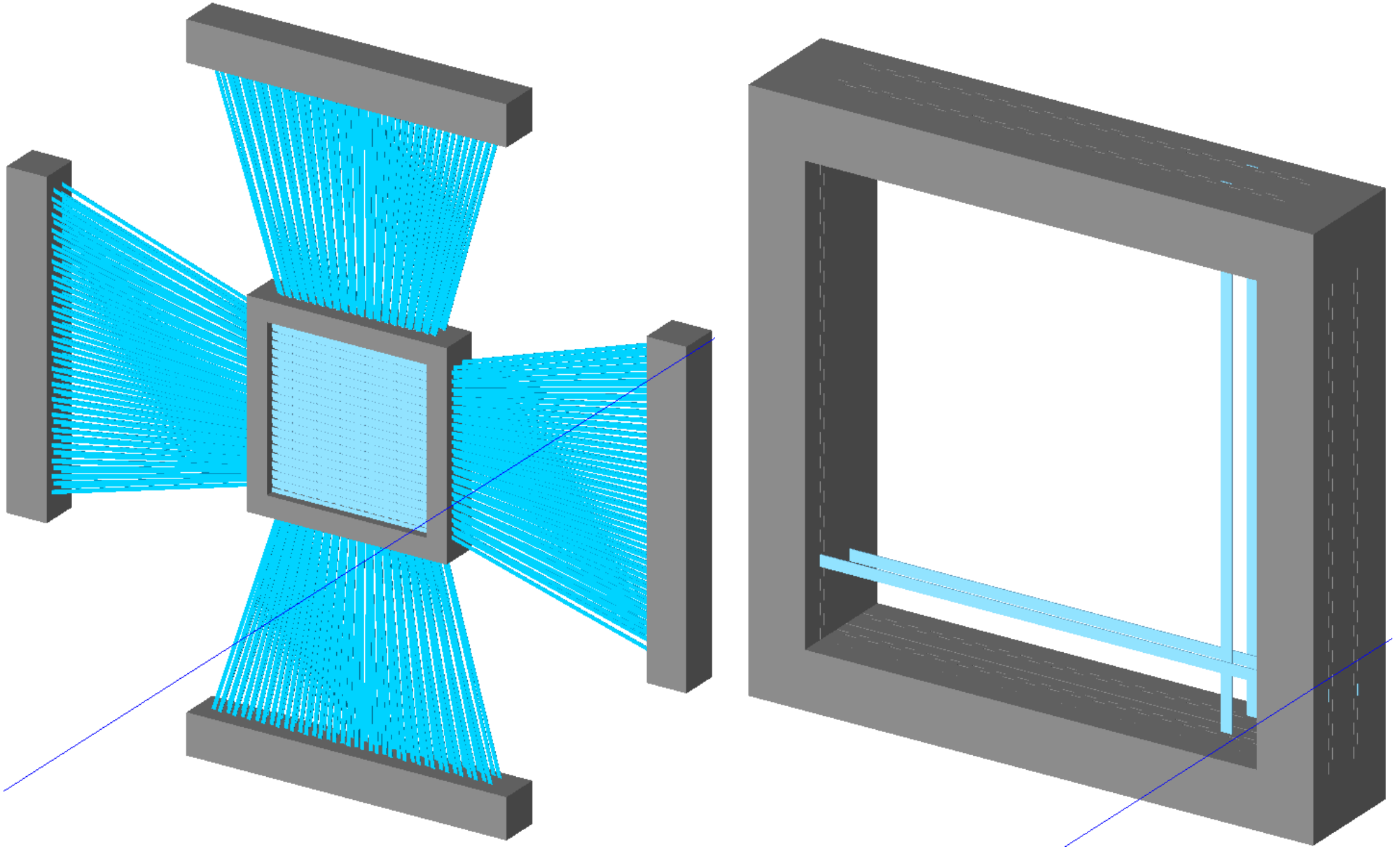
New in v10.4



Note : G4MultiUnion is a solid. Use it to create a logical volume.



Example: PRM Fiber Tracker



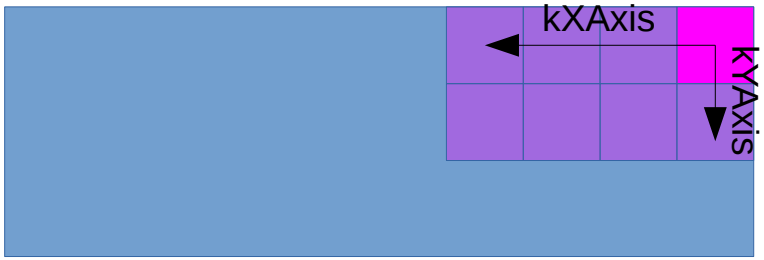
Example: PRM Fiber Tracker

Currently implemented:
For batch simulation – G4SubtractionSolid (very slow if visualized (visible))
For visualization – G4MultiUnion (not supported by GDML ROOT – CORAL etc.!)

Why working with G4SubtractionSolid for FT? - No idea yet, maybe no interactions or smaller number?

→For ECAL: G4Replica (used for instance in PixelSilicon Module)→ but: check run time!

```
// row
G4Box *rowSolid = new G4Box("row",(pixelSizeX * numberPixelX) / 2., pixelSizeY / 2., thickness / 2.);
G4LogicalVolume *rowLogicalVolume = new G4LogicalVolume(rowSolid, materials->air_noOptical, "row");
rowLogicalVolume->SetVisAttributes(colour->invisible);
new G4PVReplica("pixel", pixelLogicalVolume, rowLogicalVolume, kXAxis, numberPixelX, pixelSizeX);

// sensor layer - columns
G4Box *sensorSolid = new G4Box("Sensor",(pixelSizeX * numberPixelX) / 2., (pixelSizeY * numberPixelY) / 2., thickness / 2.);
G4LogicalVolume *sensorLogicalVolume = new G4LogicalVolume(sensorSolid,materials->air_noOptical, "Sensor");
sensorLogicalVolume->SetVisAttributes(colour->green);
new G4PVReplica("row", rowLogicalVolume, sensorLogicalVolume, kYAxis, numberPixelY, pixelSizeY);

// Place modules
for( int indexModuleY = 0; indexModuleY < numberModulesY; ++indexModuleY )
{
    for( int indexModuleX = 0; indexModuleX < numberModulesX; ++indexModuleX )
    {
        G4ThreeVector modulePosition = G4ThreeVector( (-1.0) * (numberModulesX-1) * moduleSizeX/2. + indexModuleX * moduleSizeX,(-
        modules.push_back(new G4PVPlacement(nullptr, modulePosition, sensorLogicalVolume, sensorLogicalVolume->GetName()+"_"+intTc
    }
}
```



Pixel Silicon Module
(1024x512 pixel)
Single pixel not
visible!