

Conditions data access in Athena

Nonspecialist view

Vakho Tsulaia (LBNL)

ATLAS Software TIM
LBL, November 9, 2015



Conditions data access in Athena

- Reading of conditions data in Athena is managed by the **IOVDbSvc**
 - None of the developers of the IOVDbSvc is participating in this meeting
- The IOVDbSvc is plugged in the mechanism of providing data objects to the clients via **Detector Store**. Other components involved in this mechanism:
 - StoreGate
 - ProxyProviderSvc & AddressProviders
 - IOVSvc & IOVSvcTool
- Clients need to interact only with objects in the Detector Store
 - The rest is happening behind the scene
- Finally, to make our lives more interesting, this mechanism is **based on incidents and callbacks**

Client's perspective

- Typical scenario:
 - A client declares that it is interested in some conditions object by **registering a callback**
 - StoreGate **key** for this object **may or may not be equal to the COOL Folder name**
 - When IOV boundary gets crossed for this object, the **IOVSvcTool triggers the callback**
 - Inside the callback the client **just retrieves the object from the Detector Store**
 - Conditions data gets often copied over into the client's private data structures
- Conditions client can be a **Tool**, a **Service** or an **Algorithm**

IOVDbSvc

- IOVDbSvc is one of the **AddressProviders** managed by the **ProxyProviderSvc**
- A list of **COOL Folders (and tags)** is provided to the IOVDbSvc at configuration time
- During **initialization**:
 - IOVDbSvc creates a container (map) of **IOVDbFolder** objects
 - For each folder in the map
 - IOVDbSvc reads **folder metadata** from the Conditions database
 - Sets up **cache length**
 - Creates **TransientAddress** and adds it to StoreGate

IOVDbSvc (contd.)

- updateAddress() for the given Transient Address
 - If necessary reloads the cache: **triggers reading of data from Conditions DB**
 - Loads the cache for the given IOVTime. Makes a range around this time according to the caching policy
 - Gets OpaqueAddress and sets it to TransientAddress
 - Sets IOVRange to the IOVSvc
- The calls to IOVDbSvc::updateAddress() are originated from the **incident handler of the IOVSvcTool**
 - **BeginRun.** For all folders
 - **BeginEvent.** When IOV boundary is crossed