

HLT framework comparison

Athena

Single-threaded

HLT-specific “steering” layer

- Schedules algorithms
- Makes trigger decisions

HLT-specific algorithm class for Rol-based reconstruction

AthenaMT

Multi-threaded

Common scheduler for HLT and offline algorithms

Common algorithm class for HLT and offline

- Facilitates code-sharing

Trigger decisions made by menu algorithms

Rol data stored using **EventViews**

- Can be accessed or manipulated by any algorithm
- HLT-specific information stored as event data

Status and plans

All components of the **AthenaMT** HLT workflow described here have a working prototype

Currently being combined in a demonstrator for the full workflow

- Menu algorithm workflow exists with dummy algorithms
- Migrating real algorithms to use **DataHandles**
- Adding **EventView** manipulation to the menu algorithms

Aiming for a first implementation using a limited set of algorithms by the end of this year

Will add algorithms to the menu as they are migrated over the following years, with the [plan to have a complete implementation ready for 2019](#)

Currently the HLT uses

- ~150 algorithms (connected by ~2000 “chains” defining data flow and selection)
- ~400 tools (configurable sub-components of algorithms)
- ~100 services (globally accessible interfaces to I/O, configuration, etc.)

Some components will be simplified or replaced, but the rest must be migrated