

art - Feature #15658

Milestone # 15372 (Closed): art multi-threading phase 1

Feature # 15374 (Closed): Modifying all framework-provided services to be thread safe.

Make TimeTracker service thread safe

02/24/2017 09:38 AM - Kyle Knoepfel

Status:	Closed	Start date:	02/24/2017
Priority:	Normal	Due date:	
Assignee:	Kyle Knoepfel	% Done:	100%
Category:		Estimated time:	8.00 hours
Target version:	2.07.01	Spent time:	4.00 hours
Scope:	Internal	SSI Package:	art
Experiment:	-		
Description			
Depends on cet::Ntuple, which is not yet thread safe. The TimeTracker service would need to be thread aware for calculating time differences, which are then inserted into the database. We should investigate if using CMSSW's StallMonitor, and suitably adjusting it to use SQLite, could be of benefit.			
Related issues:			
Blocked by cetlib - Feature #15672: Make SQLite cet::Ntuple facility thread safe		Closed	02/27/2017

History

#1 - 02/27/2017 10:06 AM - Kyle Knoepfel

- Blocked by Feature #15672: Make SQLite cet::Ntuple facility thread safe added

#2 - 02/28/2017 03:09 PM - Kyle Knoepfel

- % Done changed from 0 to 40

Removed TimeTracker header so that users cannot explicitly interact with the TimeTracker service. All that remains is to make the TimeTracker callbacks thread-safe, which relies inherently on cet::Ntuple.

#3 - 03/17/2017 02:35 PM - Kyle Knoepfel

- Status changed from Assigned to Resolved

- % Done changed from 40 to 100

Thread-safety achieved by representing time point data as an `std::vector<PerScheduleData>`, where the vector grows to the number of schedules specified by the user. Right now, there is a placeholder number of 1 schedule, which can be replaced by the necessary facility that communicates how many schedules have been configured by the user.

Implemented with commit [art:f4d403e7](#).

#4 - 04/27/2017 01:53 PM - Kyle Knoepfel

- Target version set to 1209

#5 - 05/25/2017 09:10 AM - Kyle Knoepfel

- Target version changed from 1209 to 2.07.01

#6 - 05/25/2017 09:20 AM - Kyle Knoepfel

- Status changed from Resolved to Closed