Program Listing for File dfft_collector.hpp

Return to documentation for file (include/holoscan/core/resources/gxf/dfft_collector.hpp)

/* * SPDX-FileCopyrightText: Copyright (c) 2023 NVIDIA CORPORATION & AFFILIATES. All rights reserved. * SPDX-License-Identifier: Apache-2.0 * * Licensed under the Apache License, Version 2.0 (the "License"); * you may not use this file except in compliance with the License. * You may obtain a copy of the License at * * http://www.apache.org/licenses/LICENSE-2.0 * * Unless required by applicable law or agreed to in writing, software * distributed under the License is distributed on an "AS IS" BASIS, * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. * See the License for the specific language governing permissions and * limitations under the License. */ #ifndef HOLOSCAN_CORE_RESOURCES_GXF_DFFT_COLLECTOR_HPP #define HOLOSCAN_CORE_RESOURCES_GXF_DFFT_COLLECTOR_HPP #include <map> #include <set> #include <utility> #include "gxf/std/clock.hpp" #include "gxf/std/monitor.hpp" #include "holoscan/core/dataflow_tracker.hpp" namespace holoscan { class DFFTCollector : public nvidia::gxf::Monitor { public: gxf_result_t on_execute_abi(gxf_uid_t eid, uint64_t timestamp, gxf_result_t code) override; void add_leaf_op(holoscan::Operator* op); void add_root_op(holoscan::Operator* op); void data_flow_tracker(holoscan::DataFlowTracker* d); int num_root_ops() { return root_ops_.size(); } int num_leaf_ops() { return leaf_ops_.size(); } private: holoscan::DataFlowTracker* data_flow_tracker_ = nullptr; std::map<int64_t, holoscan::Operator*> leaf_ops_; std::map<int64_t, int64_t> leaf_last_execution_count_; std::map<int64_t, holoscan::Operator*> root_ops_; }; } // *namespace holoscan* #endif/* HOLOSCAN_CORE_RESOURCES_GXF_DFFT_COLLECTOR_HPP */