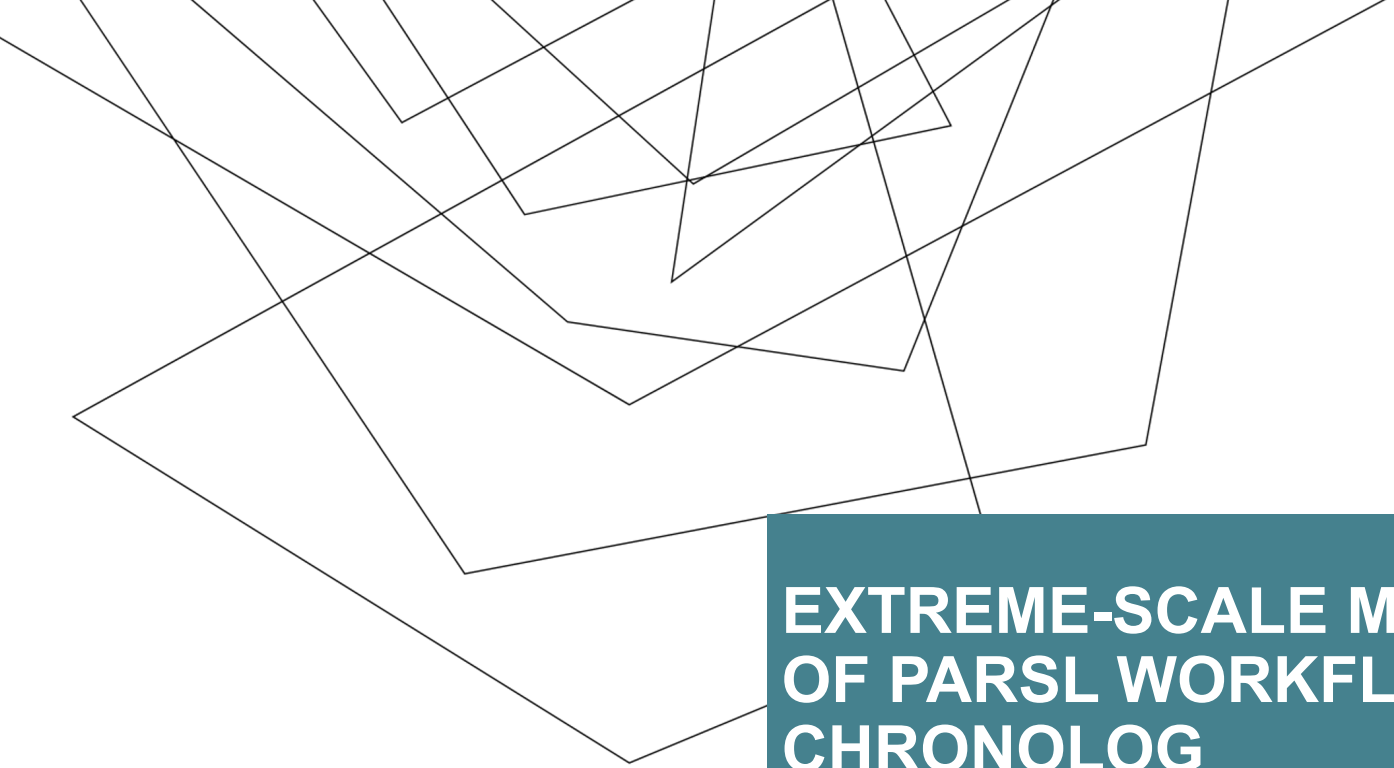


EXTREME-SCALE MONITORING OF PARSL WORKFLOWS WITH CHRONOLOG

ParslFest September 2024

ChronoLog

A High-Performance Storage Infrastructure for Activity and Log Workloads

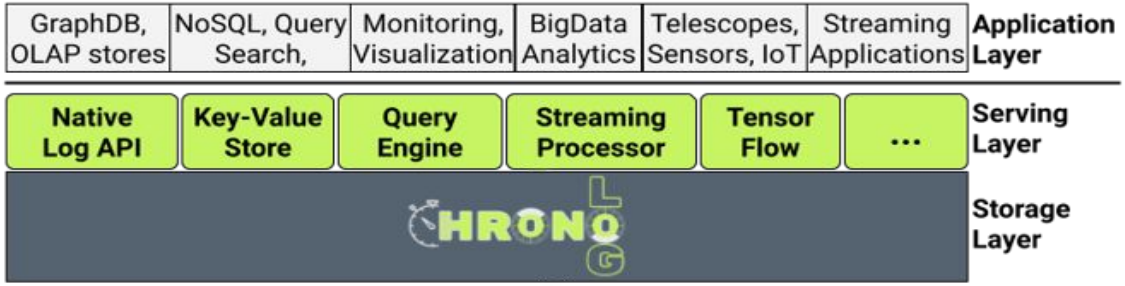
ILLINOIS INSTITUTE OF TECHNOLOGY

THE UNIVERSITY OF CHICAGO

Project Overview

ChronoLog is a distributed tiered log store ecosystem designed for the efficient recording, organization, storage, and processing of massive amounts of activity data generated by modern scientific and industrial applications.

Ecosystem



NSF Funded Project:
(NSF CSSI-2104013)

Dr. Xian-He Sun
Principal Investigator
Illinois Tech
sun@iit.edu

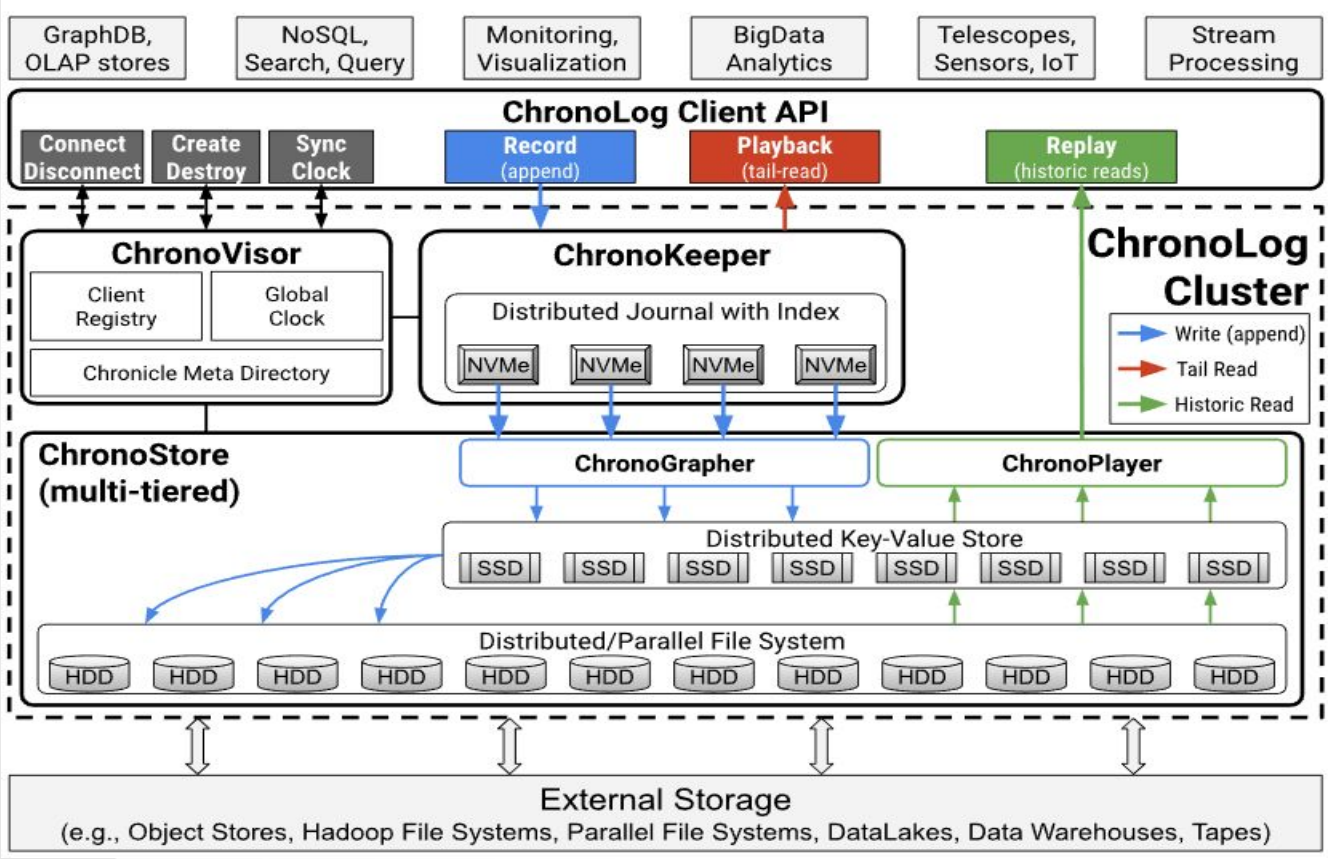
Dr. Anthony Kougkas
Co-Principal Investigator
Illinois Tech
akougkas@iit.edu

Dr. Kyle Chard
Co-Principal Investigator
University of Chicago
chard@uchicago.edu

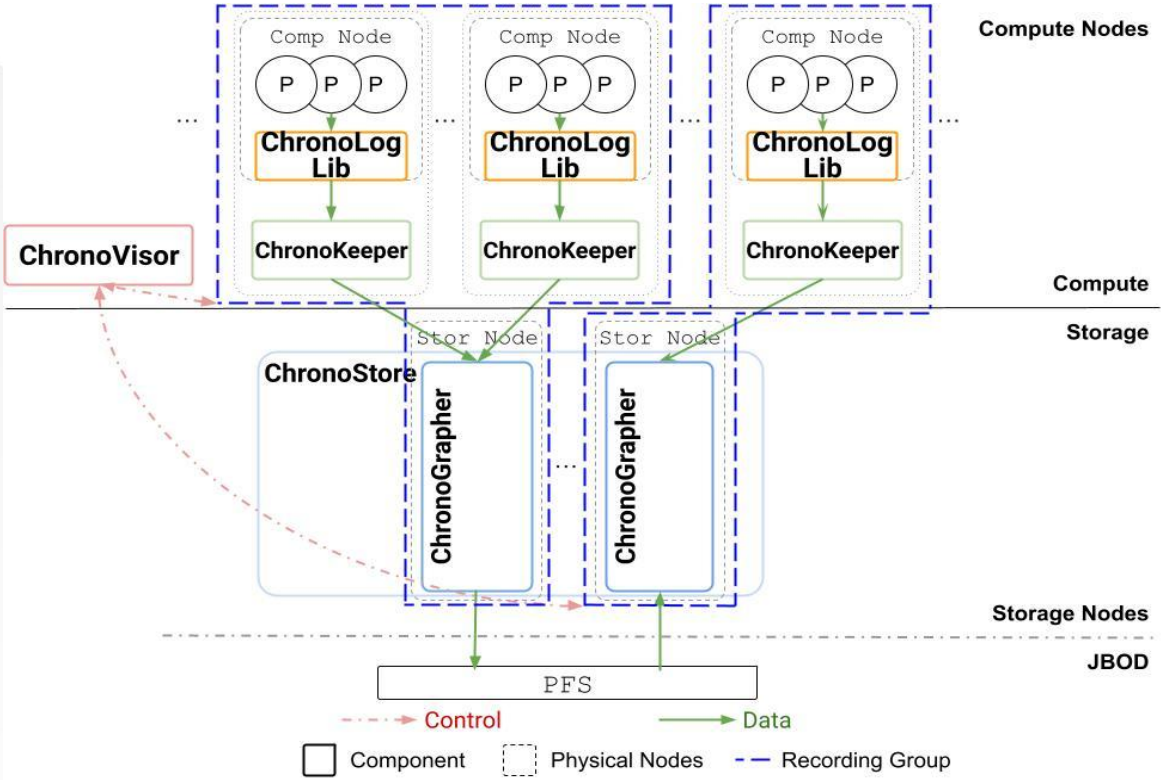
Website

Repository

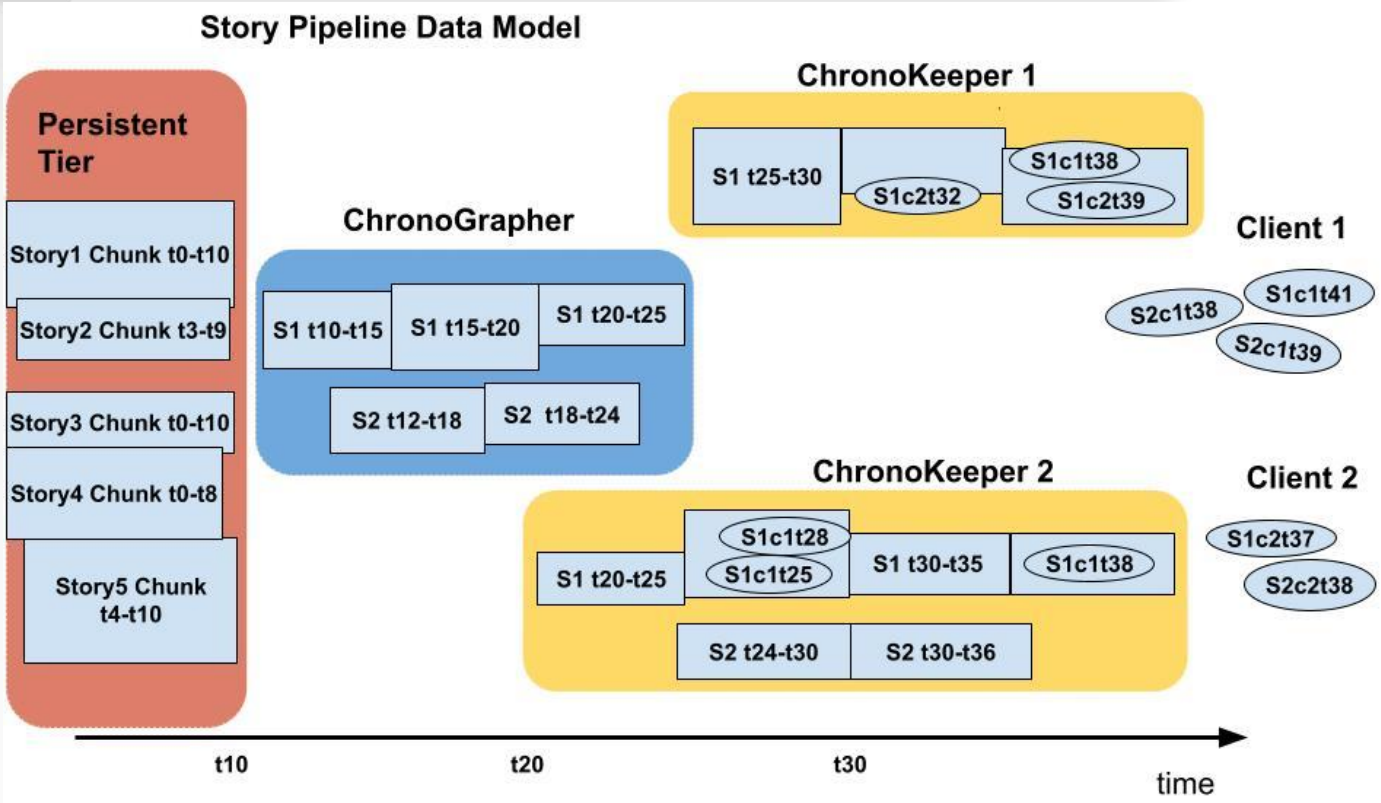
ChronoLog Framework Design



Deployment Model for ChronoLog Recording



Distributed Story Pipeline Data Model



Current Stage - ChronoLog Release 1.0.0

ChronoLog Server Side

1. **Components**
   ChronoVisor, ChronoKeeper, and ChronoGrapher processes
2. **Storage**
   3-tier distributed log recording system for efficient event stream management.
3. **Event Ordering**
   Ensures total log event ordering with client identifiers and event causality retained.
4. **Elasticity**
   Supports dynamic membership for ChronoKeeper and ChronoGrapher processes.

ChronoLog Client API

1. **Implementation**
   Multi-threaded C++ library for concurrent log event ingestion and distributed recording.
2. **Tools**
   Example client applications and a command-line admin tool for simulating logging workloads.
3. **Bindings**
   Python bindings for the C++ Client API.

Monitoring of Parsl Workflows with ChronoLog

Objectives

1. Integrate ChronoLog client API with Parsl DESC Monitoring system using and ChronoLog processes as a transport layer for Parsl DESC
2. Run various Parsl Workflows at exciting scale to collect benchmarking metrics for ChronoLog layer performance and use these to guide ChronoLog performance optimization efforts
3. Identify and implement new data extraction modules for ChronoLog Keeper and Grapher processes so that collected monitoring data can be recorded into storage solutions different from currently used Parsl DESC SQL database

WHERE DO WE GO NEXT?

- Solidify packaging and deployment procedures
- Complete HDF5 ChronoLog Storage layer implementation
- Add ChronoLog Client API for time range based historical data queries
- Implement minimal ChronoPlayer to facilitate time range based historical data retrieval

- Actively engage in conversations with potential clients to narrow down their specific requirements for collection, storage, and retrieval patterns of activity logs
- Start systematic collection of performance metrics generated through simulation of real client workloads
- Use performance metrics from client workloads and client requirements to prioritize development of new features and performance improvements

ChronoLog Project Repository and Documentation

https://github.com/grc-iit/ChronoLog

https://github.com/grc-iit/ChronoLog/wiki