# WorkVisualizer

## A User Support Tool to Facilitate General Computing Using Heterogeneous Architectures at Scale

Pierre Pébay - Barcelona Supercomputing Center (BSC-CNS) - NexGen Analytics (NGA)

# Bio

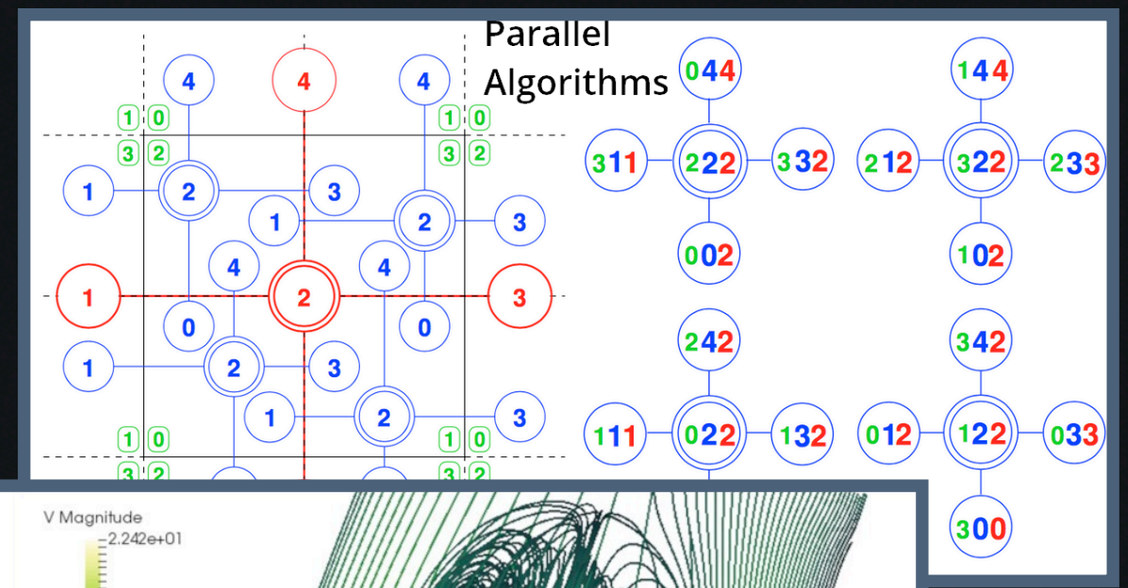
- MSc ISIMA, France
- Currently: PhD at Barcelona Supercomputing Center (BSC-CNS) in computational fluid dynamics
- Worked three years at NexGen Analytics
- Latest publications: ML-assisted decentralized load-balancing algorithms

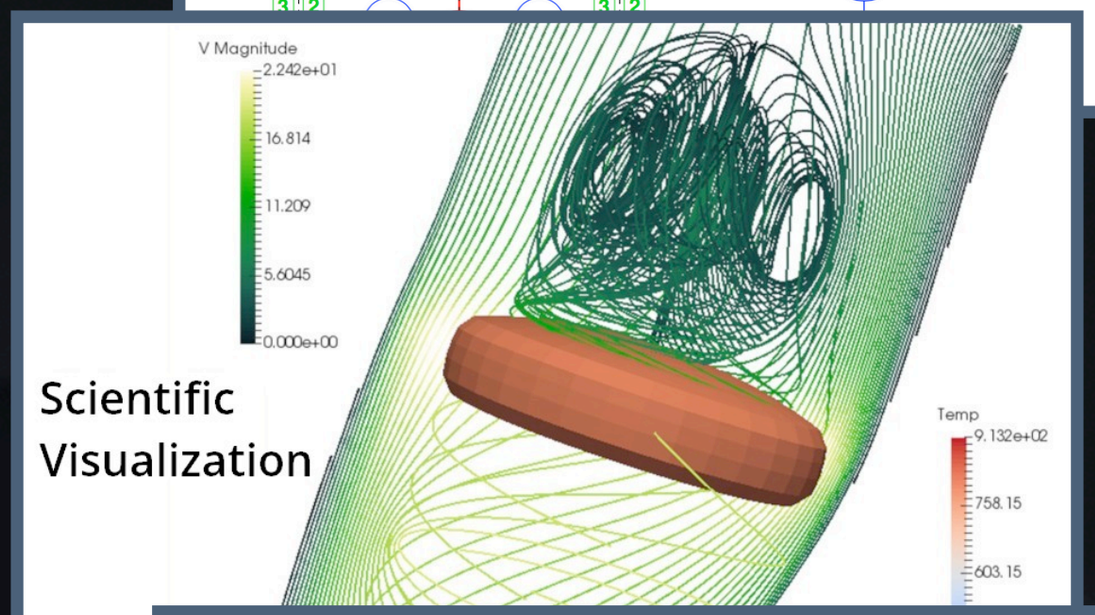

NexGen Analytics

- Office in North Carolina, USA
- Mission: support R&D projects
- Private and public clients
- Partnership with Sandia National Laboratories
- OVIS/LDMS

# Our HPC simulation background

- Multiple physics simulation codes: electromagnetics, solid mechanics, fluid dynamics
- Parallel programming paradigms, large-scale HPC
- Scalable load-balancing




Computational Statistics

$$\widetilde{M}_p = \sum_{i=1}^{n_{\mathcal{A}}} (x_i - \widetilde{x}_W)^p + \sum_{i=n_{\mathcal{A}}+1}^{n} (x_i - \widetilde{x}_W)^p$$

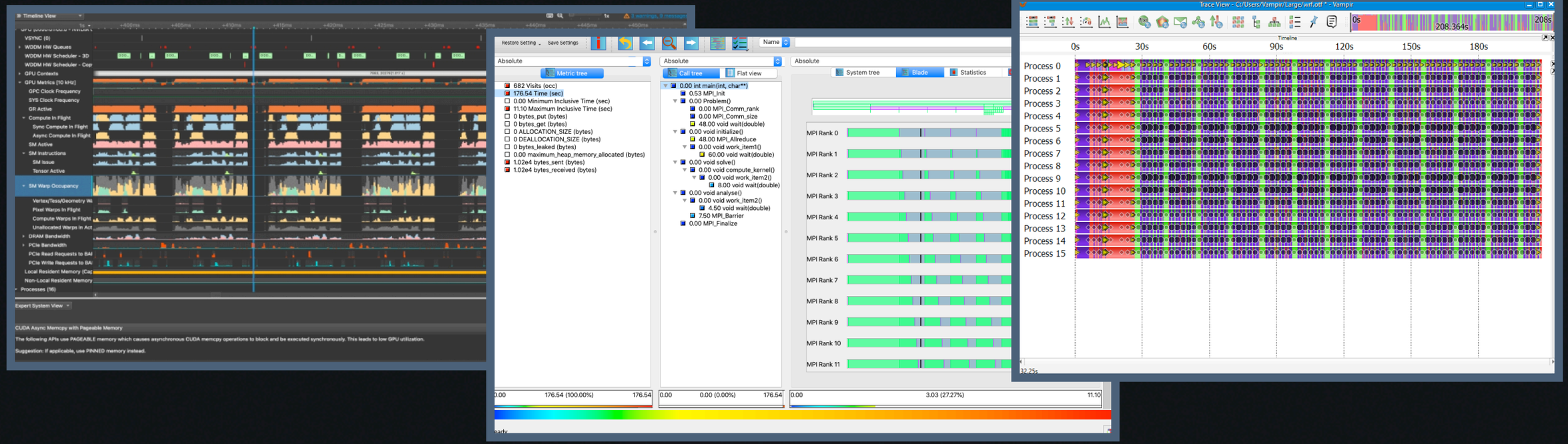
$$= \sum_{i=1}^{n_{\mathcal{A}}} \left( x_i - \widetilde{x}_{W,\mathcal{A}} - \frac{W_{\mathcal{B}}}{W} \delta_{\mathcal{B},\mathcal{A}} \right)^p + \sum_{i=n_{\mathcal{A}}+1}^{n} \left( x_i - \widetilde{x}_{W,\mathcal{B}} + \frac{W_{\mathcal{A}}}{W} \delta_{\mathcal{B},\mathcal{A}} \right)^p$$

$$= \sum_{k=0}^{p} \binom{p}{k} \delta_{\mathcal{B},\mathcal{A}}^k \left[ \widetilde{M}_{p-k}^{\mathcal{A}} \left( \frac{-W_{\mathcal{B}}}{W} \right)^k + \widetilde{M}_{p-k}^{\mathcal{B}} \left( \frac{W_{\mathcal{A}}}{W} \right)^k \right]$$

# New to fusion – Opportunity

- Fusion is new to us - we are learning
- Here to learn
- Show a potentially interesting tool
- Share technical approach to handling large-scale data and creating visually pleasing tools

# Market gap: profiling/tooling in HPC




- Why existing tools may fall short: complexity, steep learning curve, fragmented workflows
- No analysis
- Gap in the profiler/tooling market: easy to use, modern, quick, low additional work, helps improve HPC code performance

# SBIR Funding Context

Department Of Energy Small Business Innovation Research

- Awarded **DOE SBIR Phase I** last year to build an MVP and demonstrate feasibility
- Successfully completed Phase I: validated approach, proved technical viability
- Recently awarded **Phase II** (just last month) → major expansion of scope

U.S. DEPARTMENT of ENERGY | Office of Science

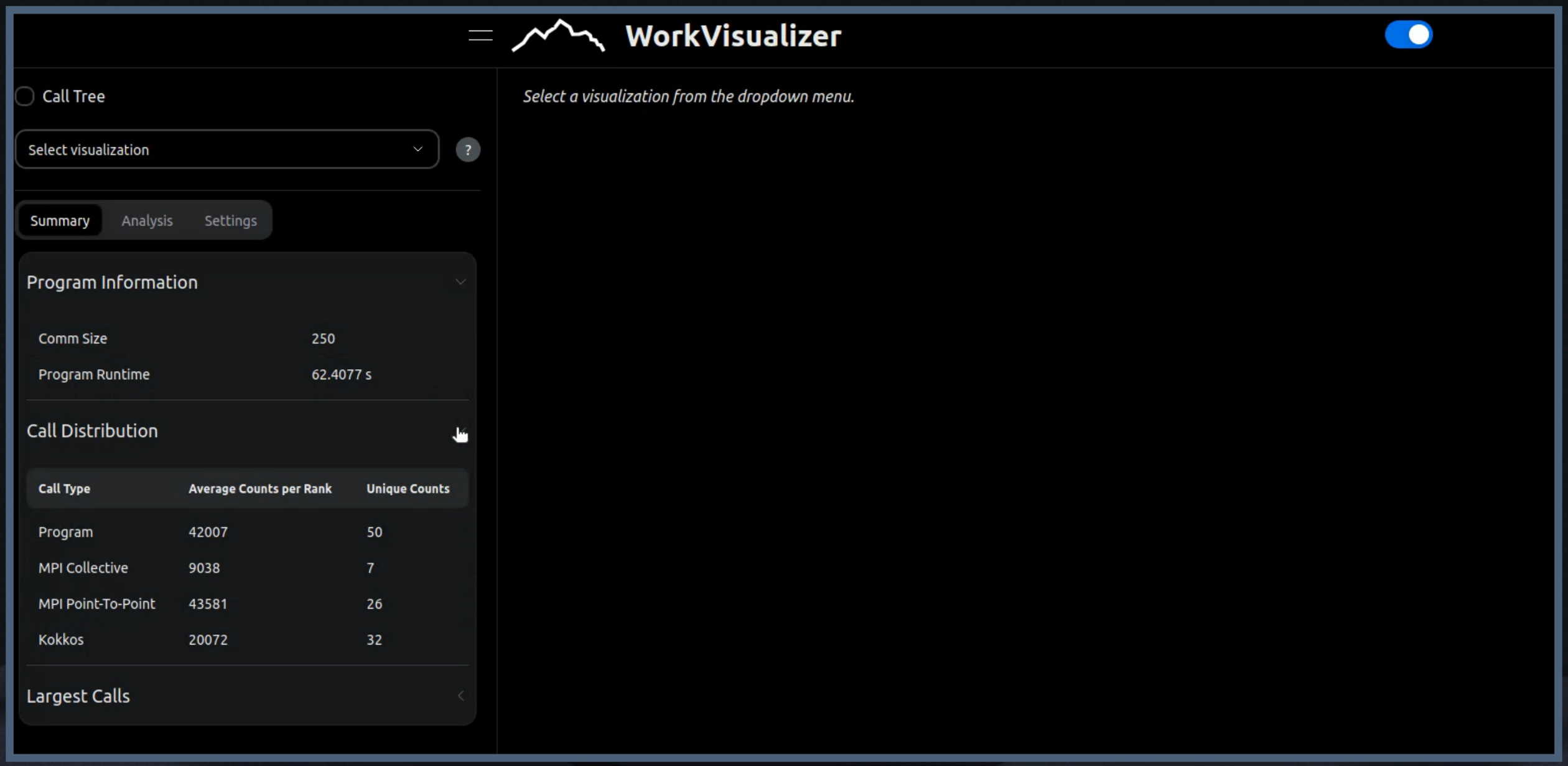
# Context: What Makes HPC Simulations Hard to Run Efficiently?

- Clusters: hundreds–thousands of nodes working together
- MPI + parallel programming: Nodes exchange data constantly → communication patterns matter
- Imbalance is common: If one rank slows down, *everyone* waits
- Heterogeneous hardware: Mix of CPUs / GPUs → different speeds, different bottlenecks
- Data volume is massive: Traces generate millions of events across thousands of ranks

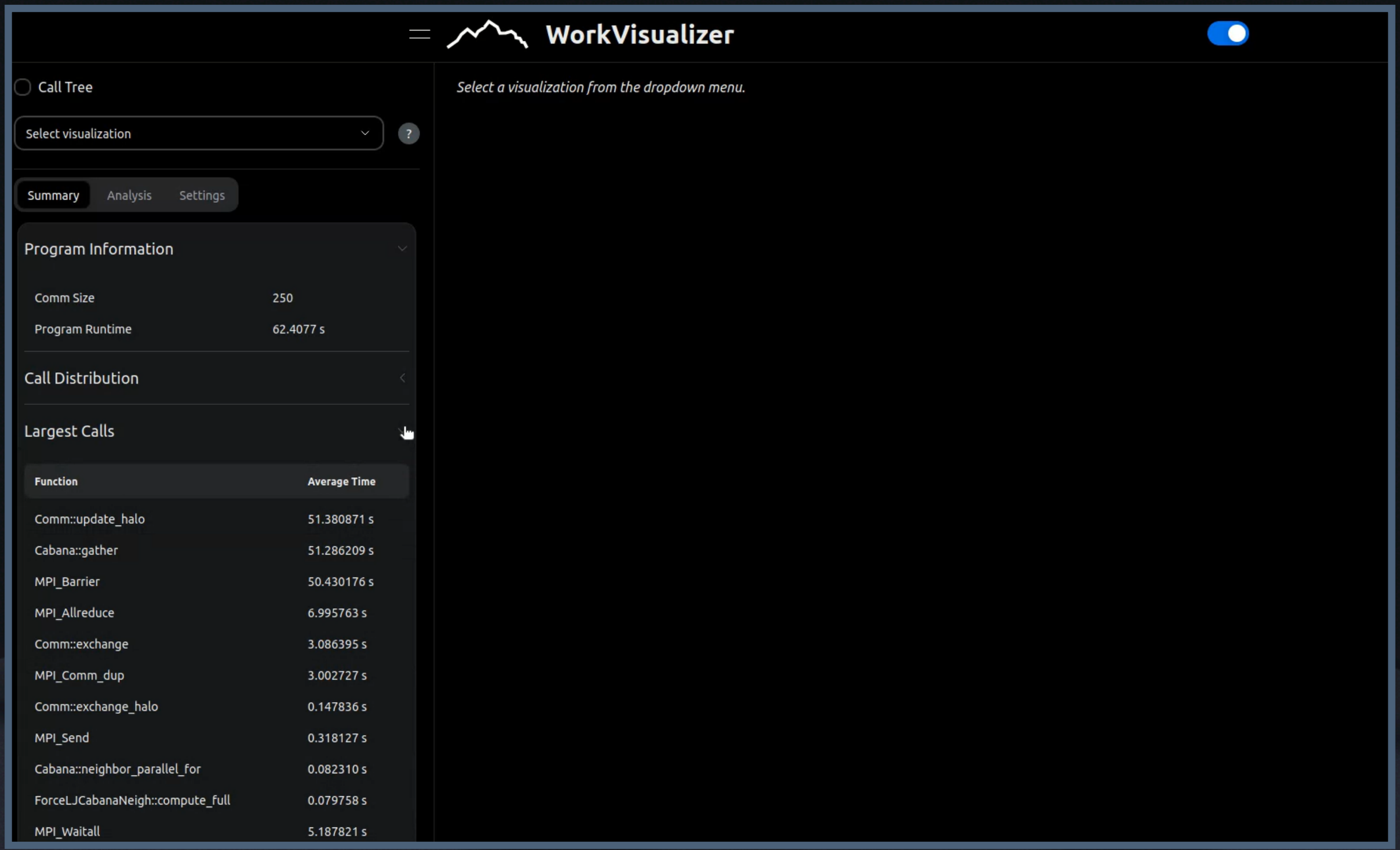
# Phase I objectives (MVP, feasibility)

- **Collection Pipeline:** low-friction data capture with zero source-code changes
- **Analysis Pipeline:** turn large datasets into actionable insights
- **Visualization Layer:** fast, interactive, high-level insights

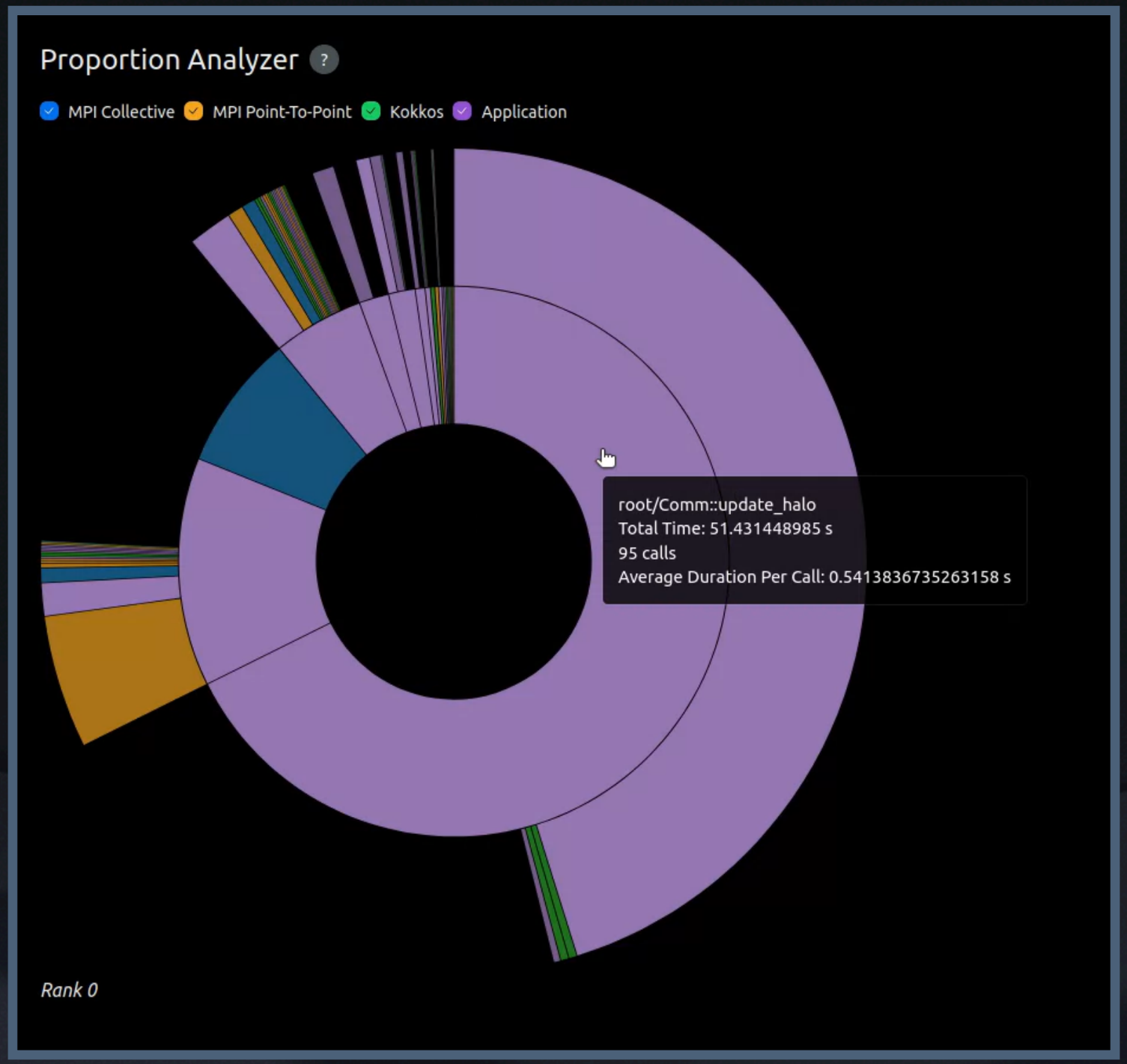
# WorkVisualizer Phase I In Practice



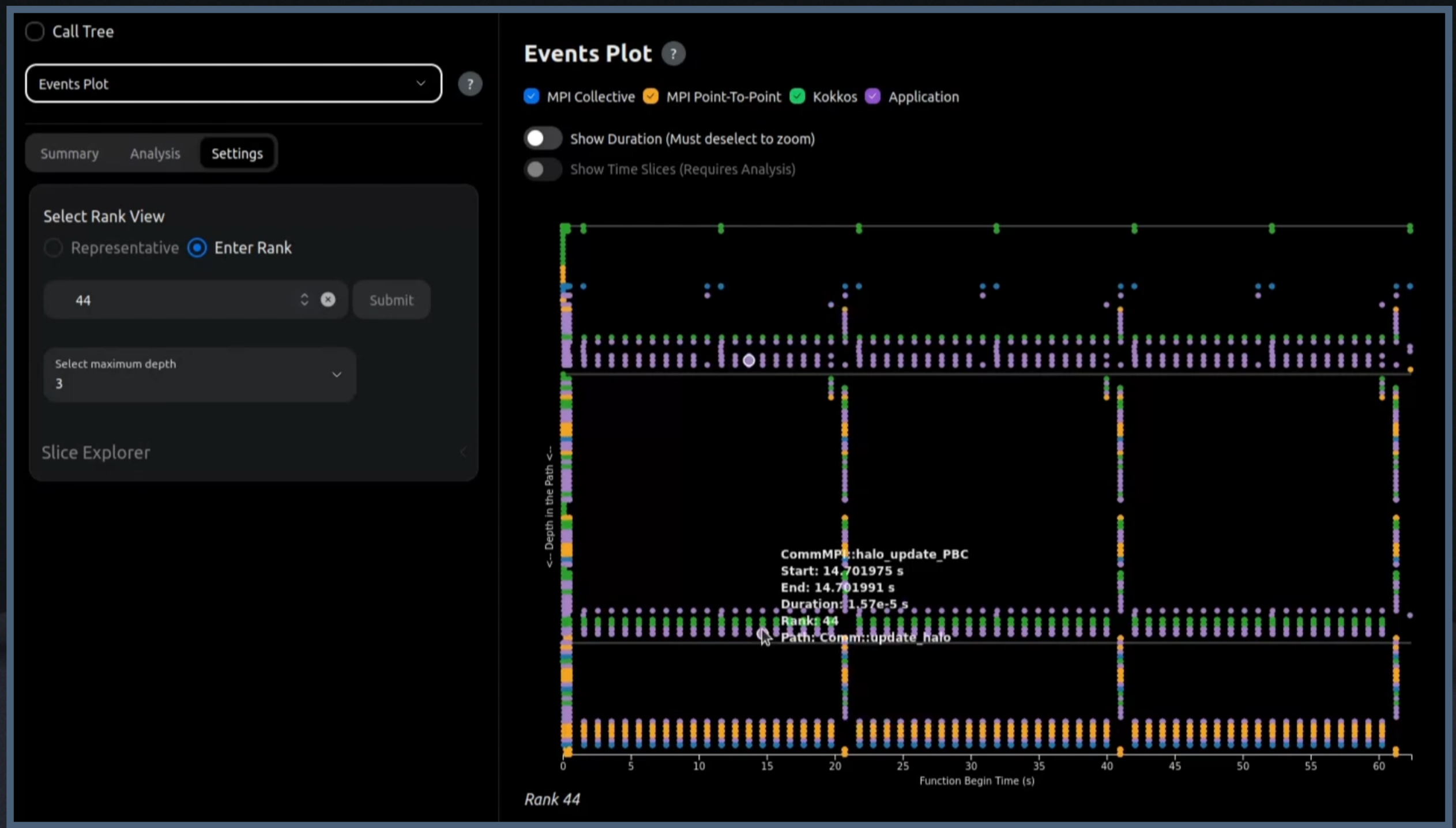
# WorkVisualizer Phase I In Practice



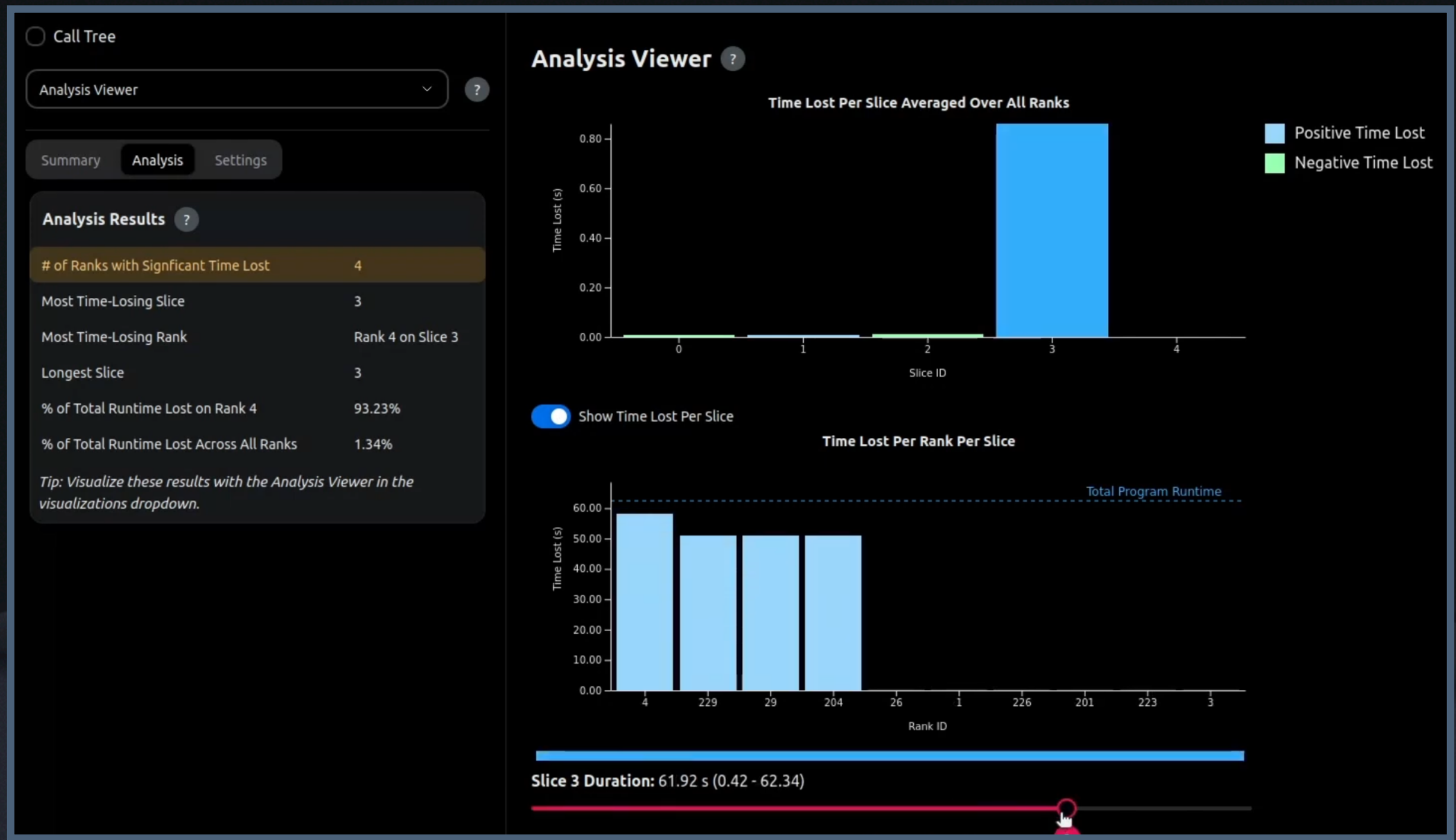
# WorkVisualizer Phase I In Practice



# WorkVisualizer Phase I In Practice



# WorkVisualizer Phase I In Practice



# Tech stack

- Backend: Python + FastAPI


- Frontend: Next.js + D3

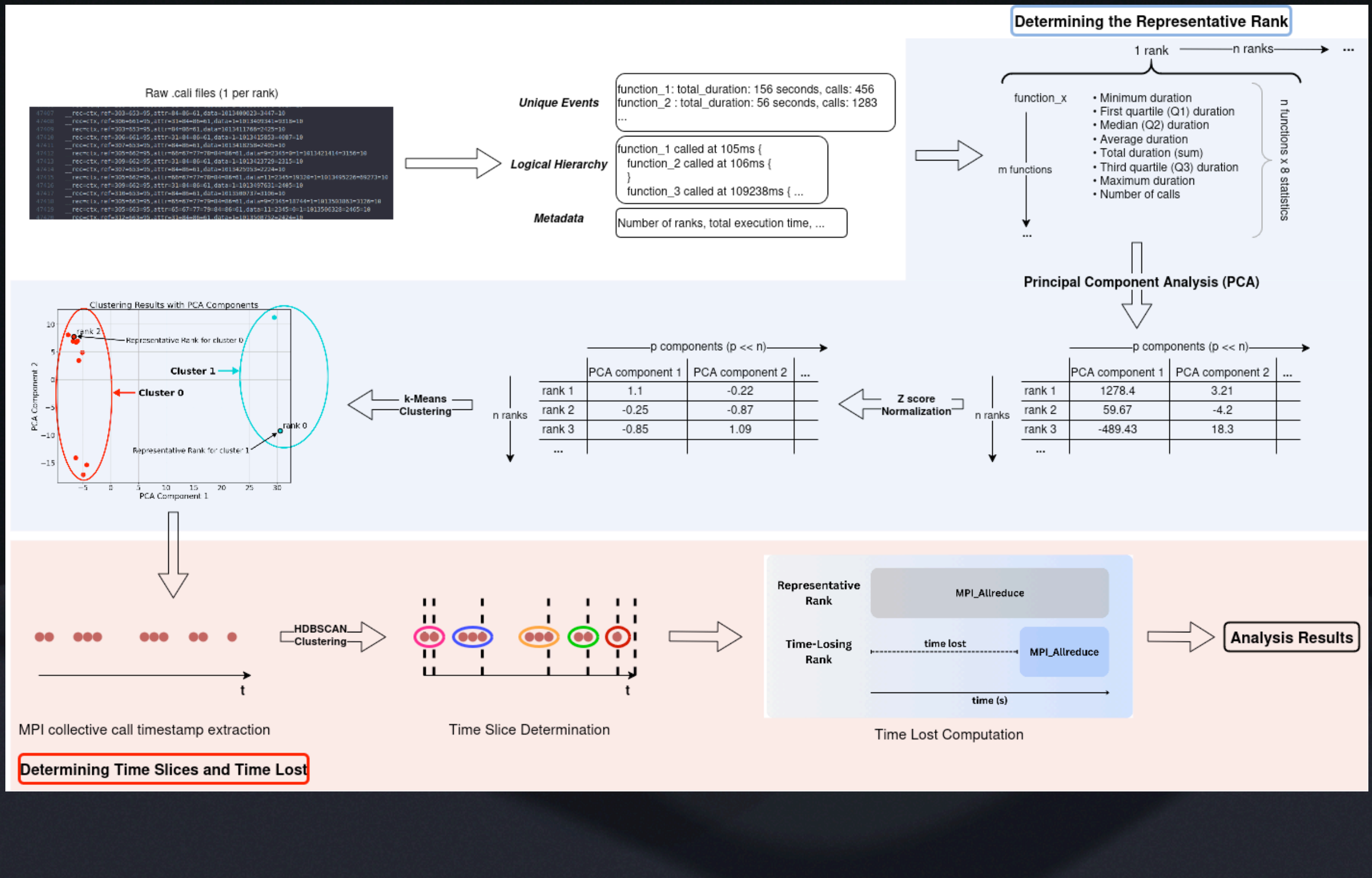

# Data Collection Pipeline

- Built on **Caliper**, a DOE-funded performance analysis toolkit
  - Seamless integration with **Kokkos** for performance-portable MPI applications
- Zero source-code changes required: Users just load Caliper + our custom `caliper.config`
- Automatic capture of:
  - Kernel execution (start/end times, durations)
  - MPI communications (source, destination, size)
- Each run produces **`data.<rank>.cali`** files: one per MPI rank

# Challenges: high-dimensional data & visualization

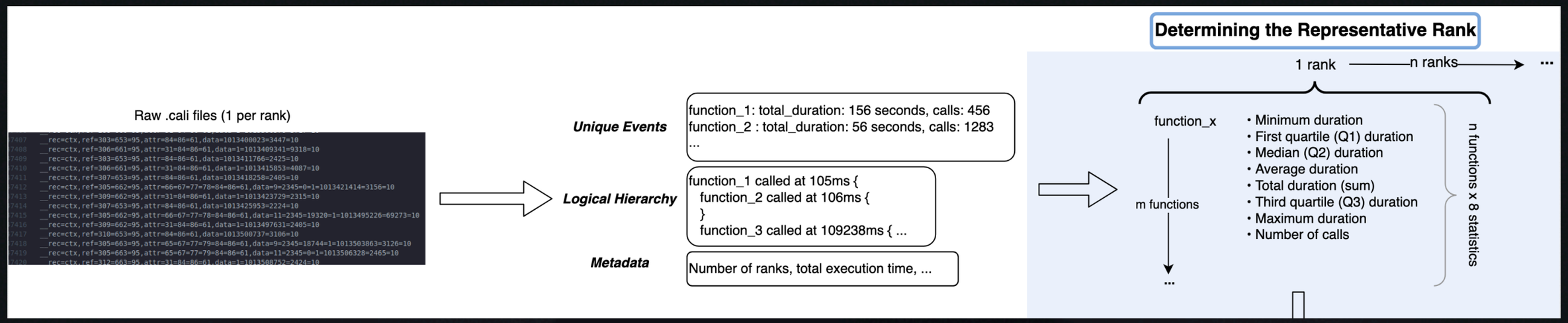
- MPI ranks, functions, call tree, time, communication
- Multiple well-chosen visualizations can capture a lot of information
- But: we need analysis

# Analysis Pipeline



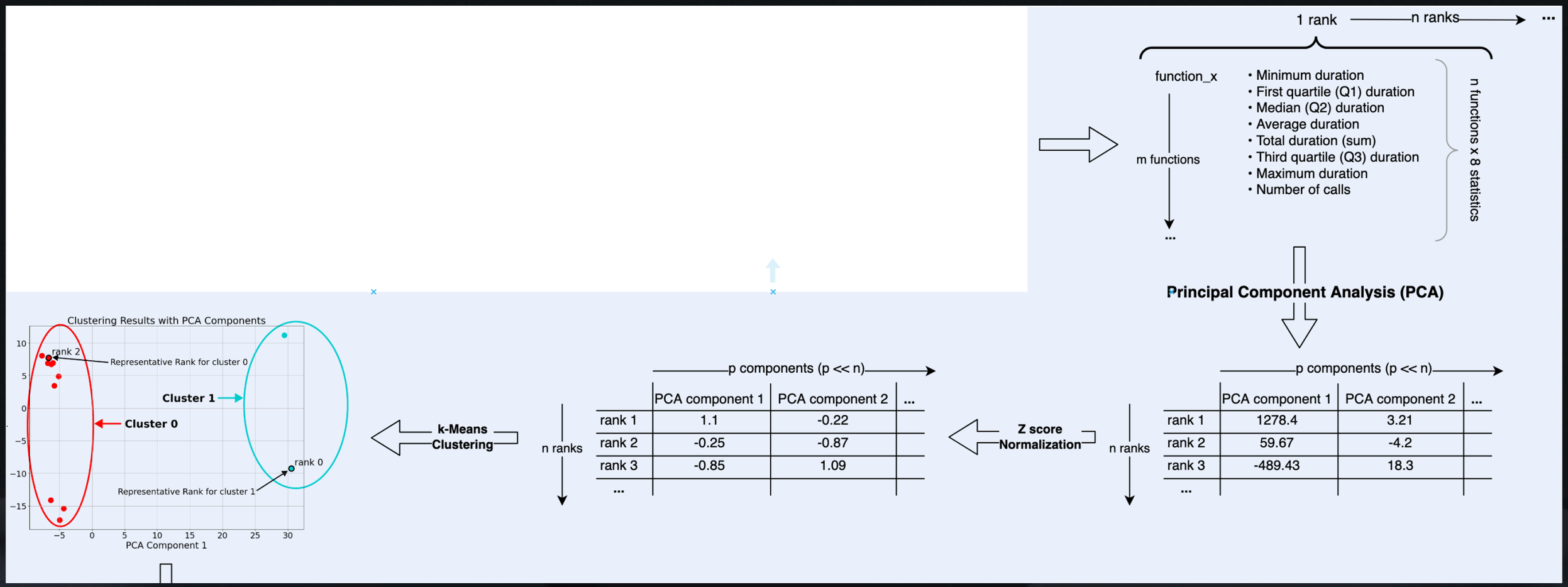
# Analysis Pipeline

## Preprocessing



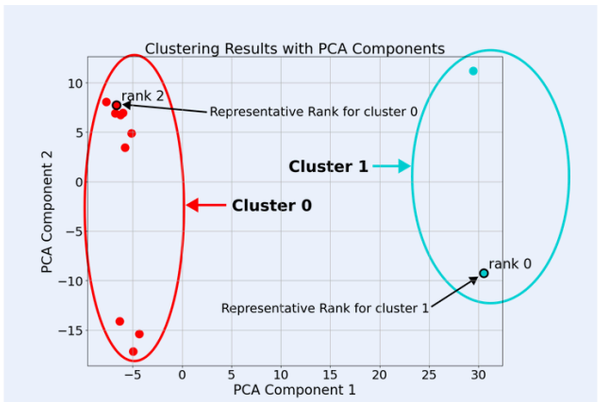
# Analysis Pipeline

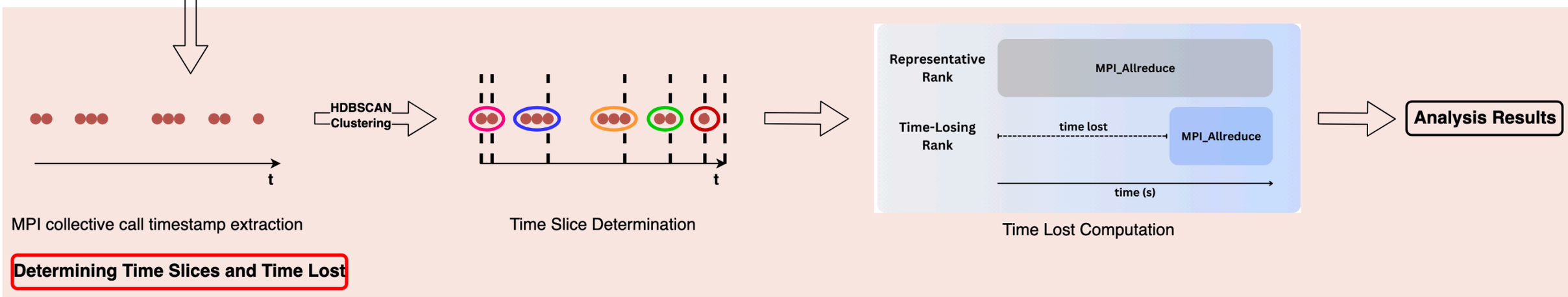
## Determining the Representative Rank



# Analysis Pipeline

## Determining the Representative Rank




MPI collective call timestamp extraction

Time Slice Determination

Time Lost Computation

**Determining Time Slices and Time Lost**

# Phase I Takeaways

- Delivered full end-to-end MVP (open-source: https://github.com/NexGenAnalytics/WorkVisualizer)
- Proven low-overhead data collection with Caliper + Kokkos
- Novel analysis pipeline: rank clustering + time slicing + “time lost” metric
- Visualizations are fast, interactive, and actionable
- Validated on real HPC apps (e.g., CabanaMD) → successfully detected injected performance issues

# Phase II: What's Next

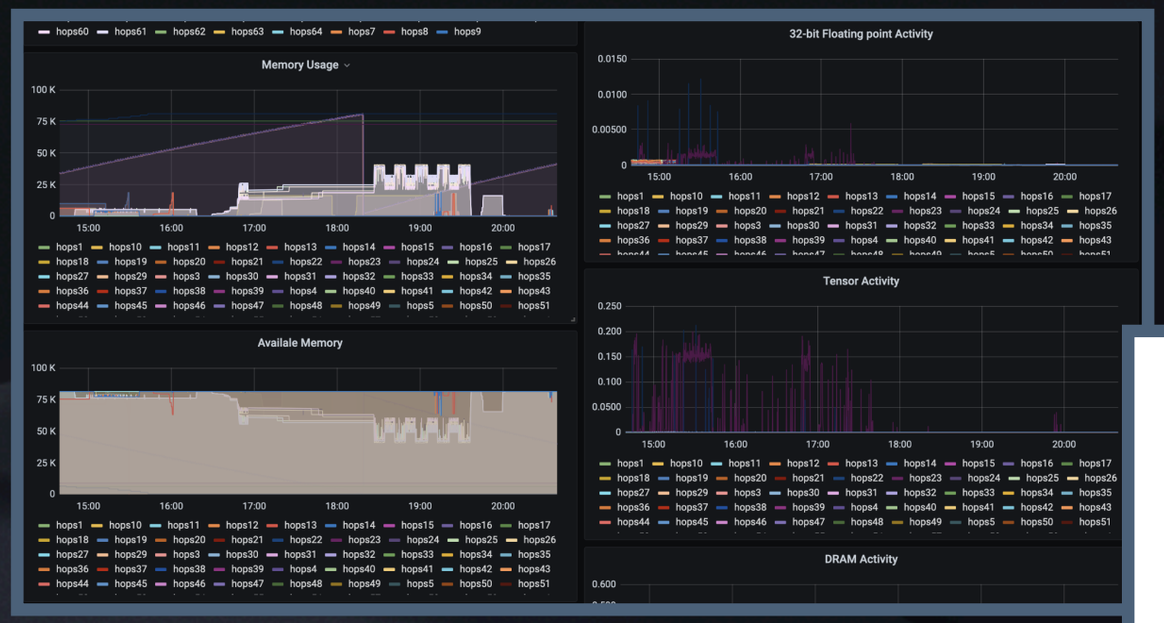
- Focus on scaling, speed, and usability
  - **Accelerate time-to-results** (faster ingestion, real-time interaction)
- Expand analysis capabilities & GPU support
  - Smarter detection of bottlenecks (communication, imbalance, transfers)
- Unified views: select a function → highlighted everywhere
- Move toward deployment + in-situ workflows
- Some work will remain open-source; advanced features become commercial

# Phase II: Research Directions

- AI/ML methods for pattern & anomaly detection
- Generalized, improved time-slicing
- Real-time / *in-situ* WorkVisualizer
- OVIS/LDMS partnership

# OVIS/LDMS Collaboration

- Framework for collecting, aggregating, and transporting system metrics in HPC environments
- Partnership: Predicting active memory usage, automating scheduling, visualizing the results



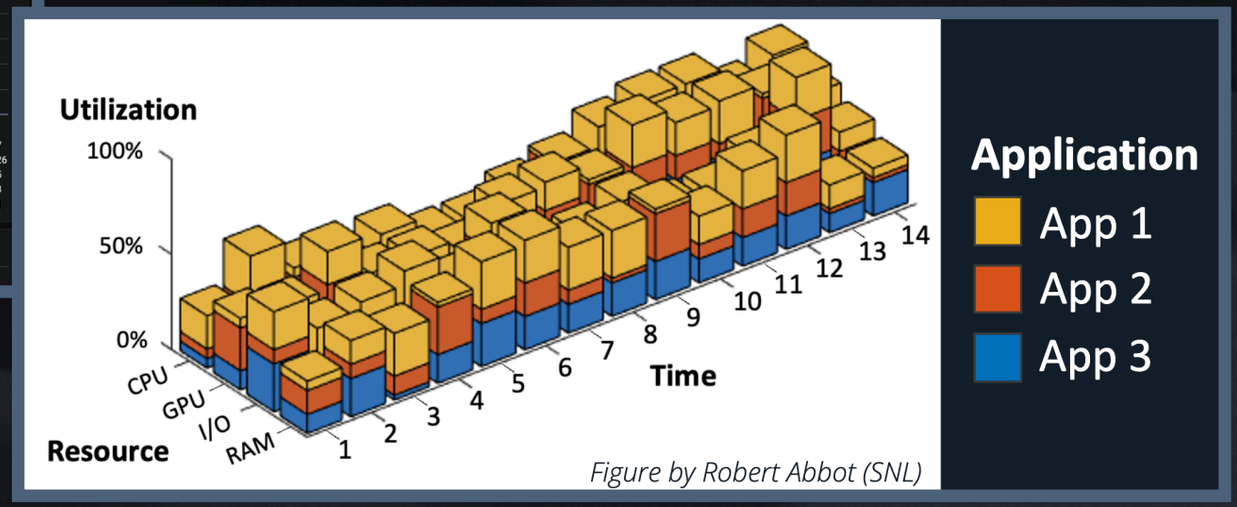




*Figure by Robert Abbot (SNL)*

# Summary & Q&A

- Phase I proved feasibility with a full open-source prototype
- Phase II focuses on speed, scalability, GPUs, and deeper analysis
- Aim: make HPC performance insights accessible & actionable
- Looking forward to engaging with fusion applications!