

# STAMPEDE: A Framework for Monitoring and Troubleshooting of Large-Scale Applications on National Cyberinfrastructure

Fabio Silva<sup>3</sup>, Christopher Brooks<sup>5</sup>, Ewa Deelman<sup>3</sup>, Monte Goode<sup>1</sup>, Dan Gunter<sup>1</sup>, Gideon Juve<sup>2</sup>, Gaurang Mehta<sup>3</sup>, Priscilla Moraes<sup>4</sup>, Taghrid Samak<sup>1</sup>, Martin Swamy<sup>4</sup>, Prasanth Thomas<sup>3</sup> and Karan Vahi<sup>3</sup>  
<sup>1</sup>Lawrence Berkeley National Laboratory, <sup>2</sup>University of Southern California, <sup>3</sup>University of Southern California Information Sciences Institute, <sup>4</sup>University of Delaware, Newark, <sup>5</sup>University of San Francisco

Scalable collection and analysis of performance and troubleshooting data from distributed workflows

Online detection and analysis of workflow failures and anomalies

Access via streaming subscription-based services

## Large-Scale Workflows

- Composed of thousands to millions of coordinated tasks
- Executed in complex distributed environments
- Difficult to track failures, search through thousands of files

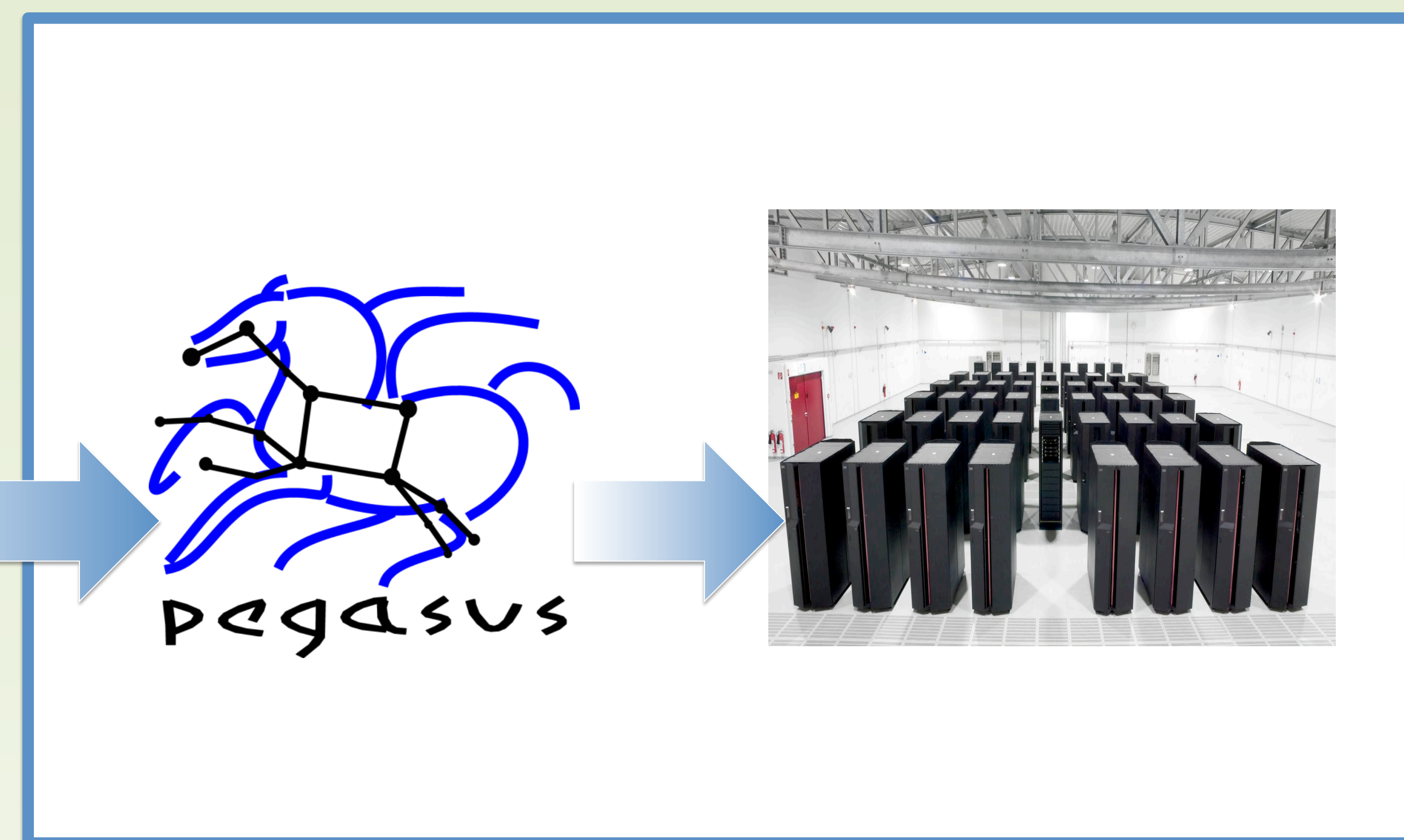
## Failure Analysis Algorithms

- Machine learning algorithms *predict workflow failures* based on behavior patterns
- Hard failures can be easily determined using database queries
- Soft failures are often stochastic, should be detected early for quick error recovery

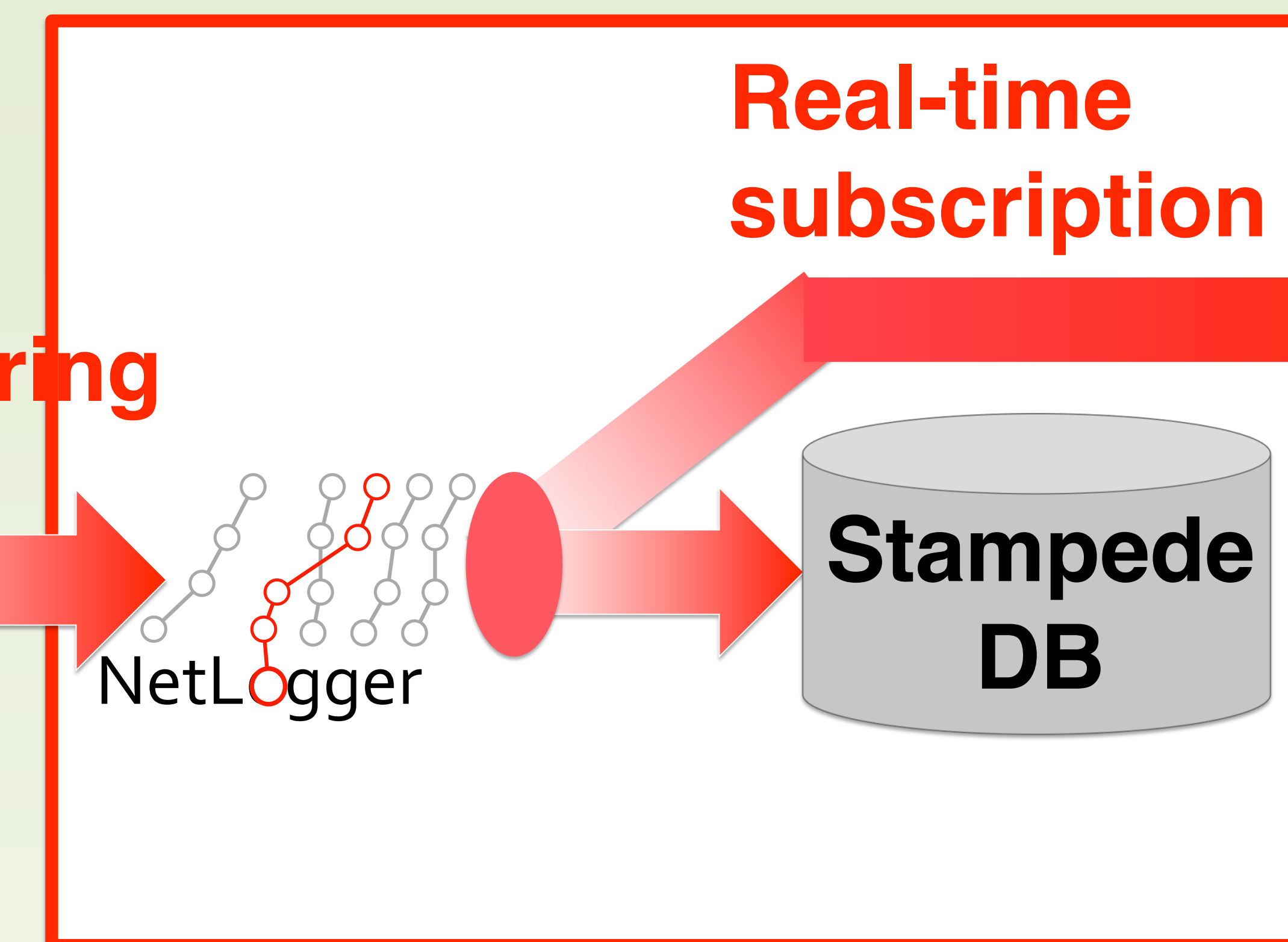
## End-user Tools

- Stampede Python Analysis API*: Simple, uniform access to back-end database
- Stampede-analyzer*: Quickly debug a workflow after execution is completed
- Stampede-statistics*: Generates statistics about a running or finished workflow
  - Number of tasks/jobs/sub workflows ran/succeeded/failed/retried, ...
  - Job execution site, scheduler queuing time, execution delay, ...
- Stampede-plots*: Interactive graphs and charts for workflow visualization

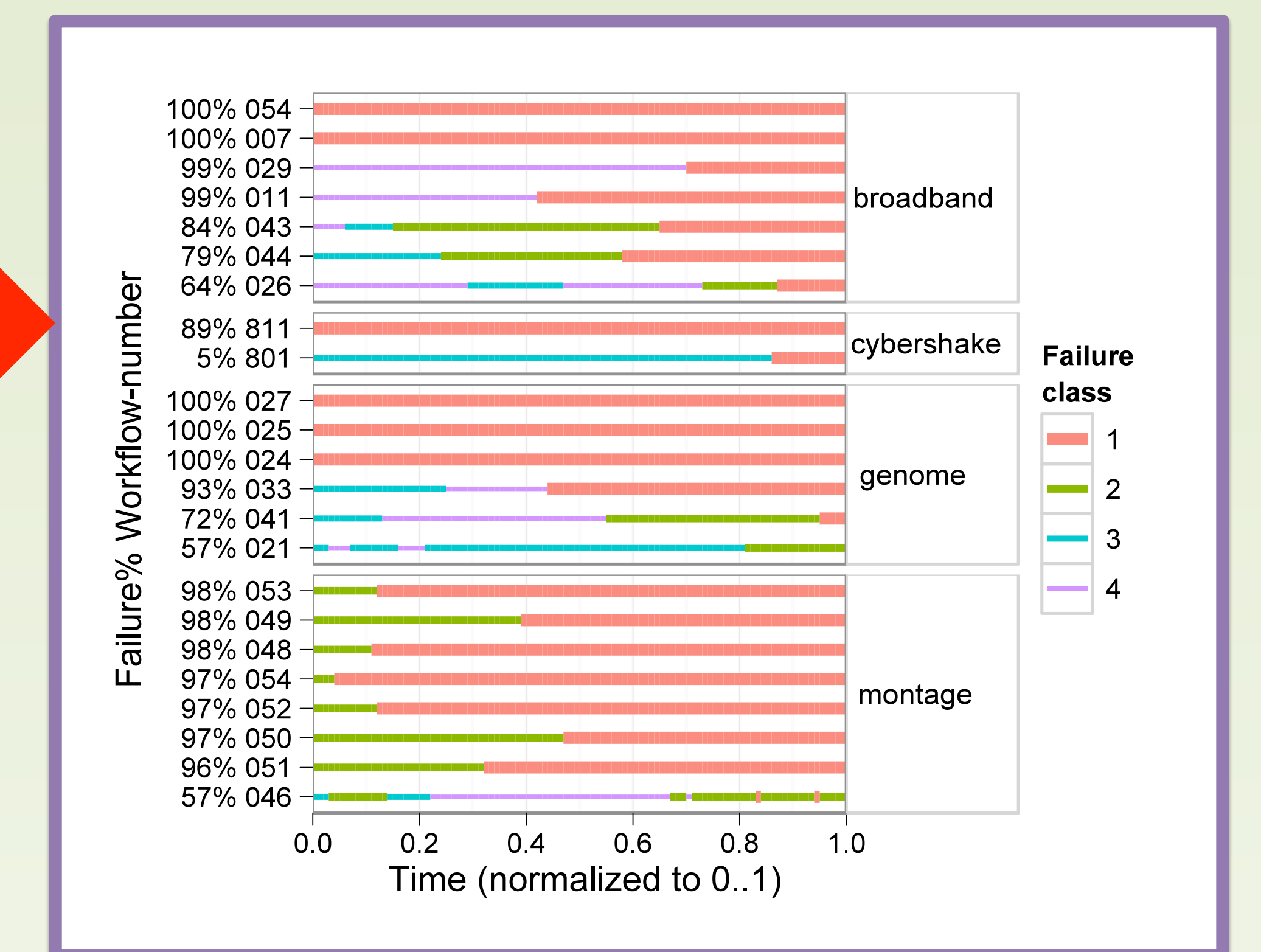
## Plan and Execute



## Transform and Archive



## Real-time Analysis



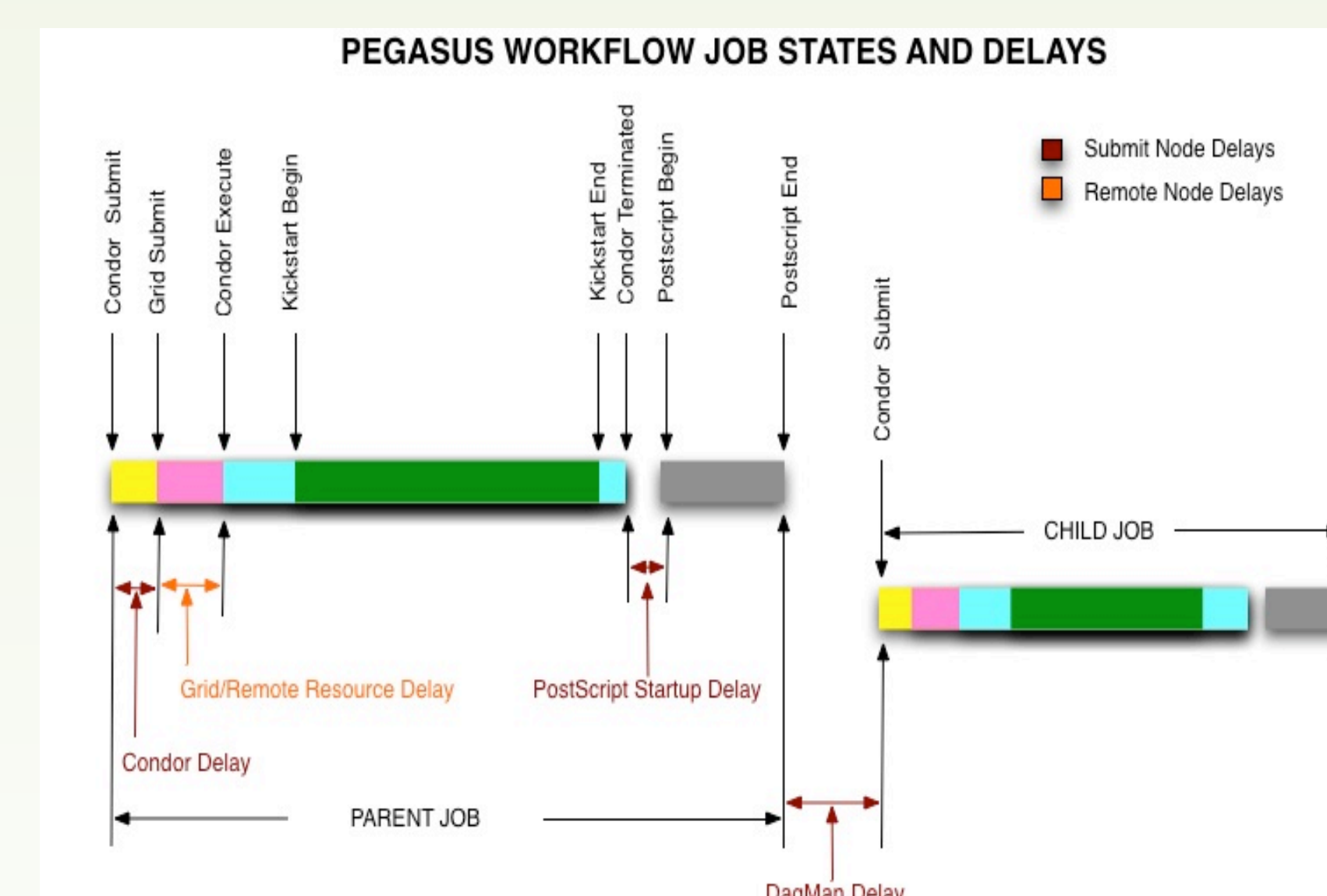
Scientific Workflow

## Log Collection Tools

- monitord* parses data in real-time
- Collects workflow execution logs

## NetLogger Tools

- nl-loader* stores data in a database, such as SQLite, MySQL, etc.
- Data goes to a broker, where it can then be sent to many subscribers



## STAMPEDE database schema

- Represents both the abstract workflow plan and running workflow, including the associations between the two
- Handles parent and child workflows
- Stores data at high granularity

## PeriScope

- Provides end-to-end system performance view to users
- Presents data from processes, hosts, and network elements using a scalable analysis and presentation framework