SWARM
DISTRIBUTED RESILIENCE SYSTEM

U.S. DEPARTMENT OF ENERGY
DE-SC0024387

Reimagining Workflow Management

Ewa Deelman

University of Southern California

PegasusAI

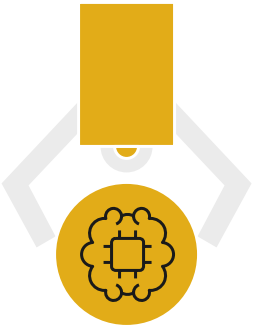
NSF award #2513101

NSF

Progression of Automation

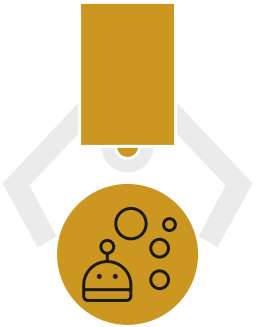
Pegasus

Computation automation



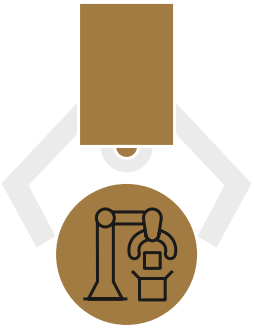
Pegasus AI

Infusing AI techniques



Agentic Workflows

Based on swarm intelligence



Self-driving Labs

Automation of experimental workflows

Pegasus Workflow Management System est. 2001

Automates the execution of scientific workflows across national CI

Heterogeneous Environments

Pegasus can execute workflows in a variety of distributed computing environments such as HPC clusters, Amazon EC2, Google Cloud, Open Science Grid or ACCESS

Data Management

Pegasus handles data transfers, input data selection and output registration by adding them as auxiliary jobs to the workflow

Provenance Tracking

Pegasus allows users to trace the history of a workflow and its outputs, including information about data sources and software used

Error Recovery

Pegasus handles errors by retrying tasks, workflow-level checkpointing, re-mapping and alternative data sources for data staging

USC Viterbi School of Engineering Information Sciences Institute

Collaboration with HTCondor

This work is funded by NSF, award #2513101

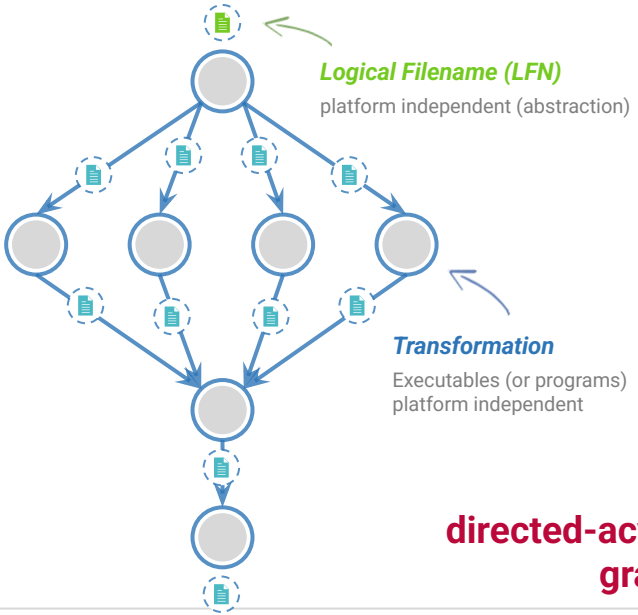
Abstraction: Resource-Independent Specification

Input Workflow Specification

YAML formatted

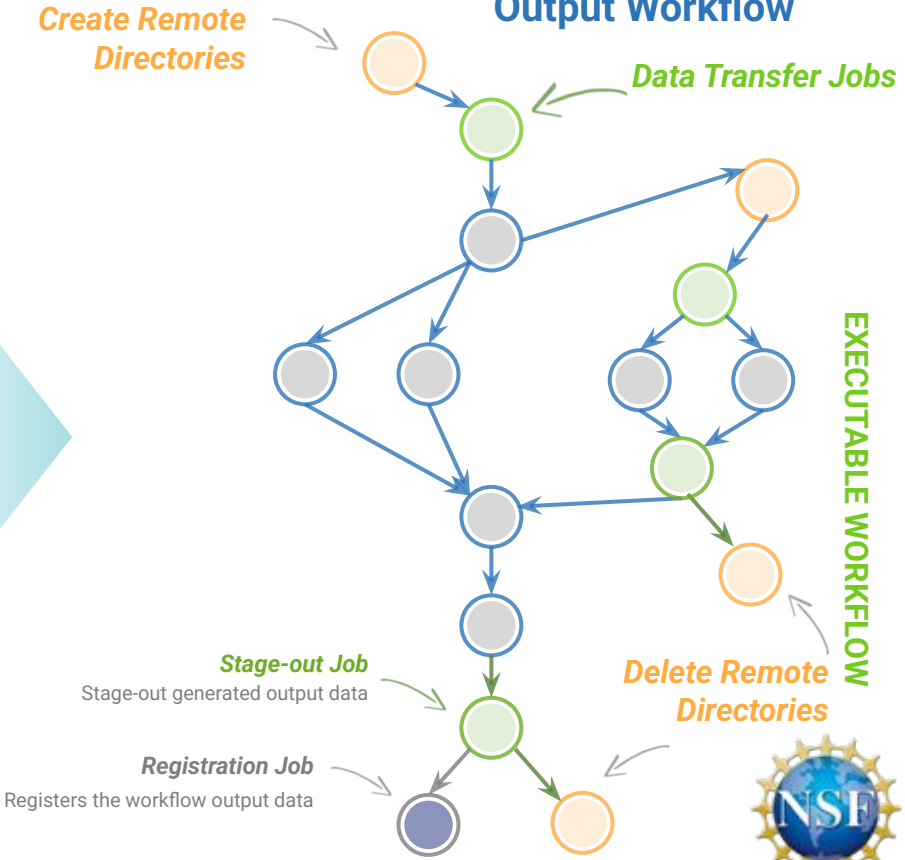
Portable Description

Users do not worry about low level execution details



directed-acyclic graphs

Output Workflow



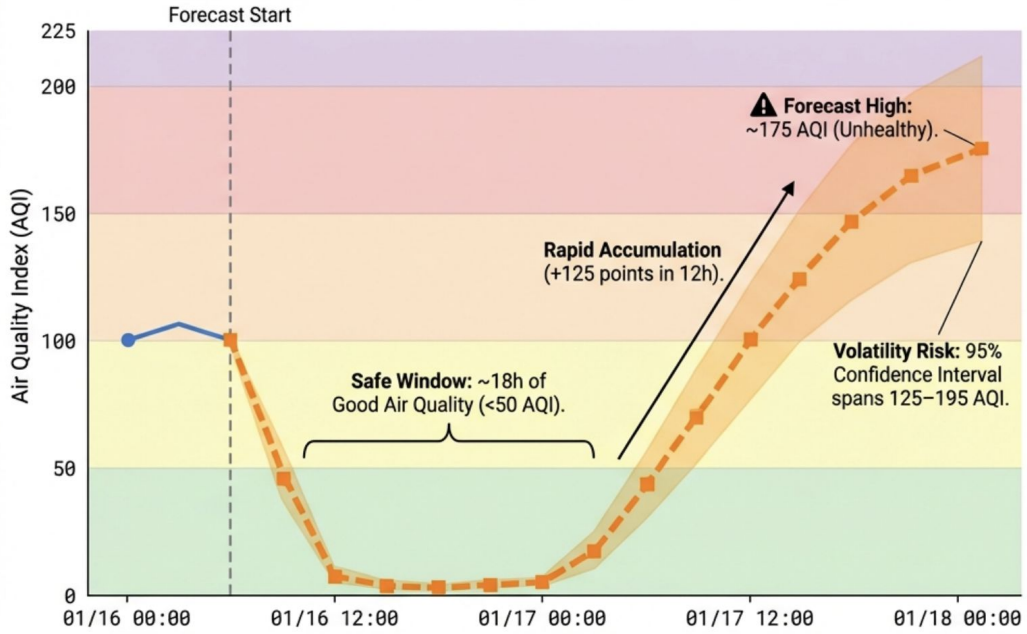
USC Viterbi School of Engineering Information Sciences Institute

Simple Data-Driven Workflows

Forecast Alert: Severe Air Quality Degradation Predicted for Jan 17-18

Location: Illmitz am Neusiedler See | Model: LSTM (48h Horizon) | Generation Date: 2026-01-19

Status: UNHEALTHY (>150 AQI)



EXECUTIVE SUMMARY

The LSTM model predicts extreme volatility over the next 48 hours. While conditions will remain favorable for the majority of Jan 16th (dipping to near-zero AQI), a rapid accumulation of pollutants is forecast to begin midday Jan 17th, effectively ending the safe window.

CRITICAL TIMING (UTC)

Jan 16 12:00 — Jan 17 10:00: Ventilation Opportunity (AQI < 50)

Jan 17 12:00: Degradation Onset (AQI crosses 50)

Jan 17 20:00: Unhealthy Threshold (AQI crosses 100)

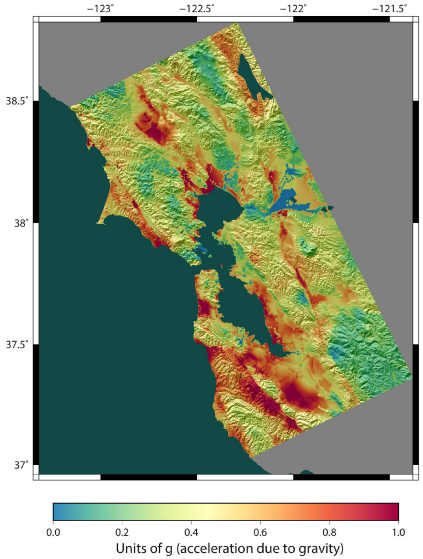
Jan 18 00:00: Peak Impact (AQI ~175)

TECHNICAL CONFIDENCE

Model confidence is high during the clear window but degrades during the spike. The widening 95% Confidence Interval suggests the peak could range from 'Unhealthy for Sensitive Groups' (125) to bordering on 'Hazardous' (195).

USC Viterbi School of Engineering Information Sciences Institute

Pegasus for Seismic Hazard Applications



45
Days
Duration of 24/7 operation

55K
Frontier
Node-hours

125K
Frontera
Node-hours

*High-water marks were 4130 nodes on Frontier (44%) and 1029 nodes on Frontera (12%)**

1 PB
Data
Managed by Pegasus

9M
Files
Staged to USC CARC

945
Workflows
27,720 jobs

Useful information for:

- ✔ Building engineers
- ✔ Disaster planners
- ✔ Insurance agencies

Two new Northern California hazard models were produced.

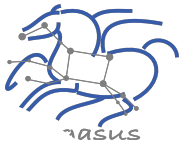
USC Viterbi School of Engineering Information Sciences Institute

Slide Courtesy of Scott Callaghan, USC

* *Neither used a reservation or priority bump.*

SCEC AN NSF+USGS CENTER

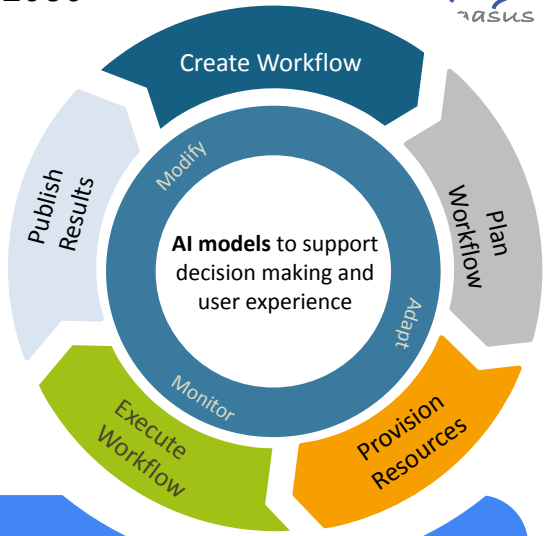
2025 - 2030



Manual Workflows

Automated Workflows

AI-Augmented Workflows



Human-orchestrated decisions

Static scripts, manual scheduling

- WMS, static plans, DAGs
- Predefined execution plans

- Workflow composition (chatbots, LLMs)
- Resource need and performance prediction
- Anomaly detection
- Dynamic workflow execution adaptation
- Learn about the workloads and systems
- Predict/design systems to serve the workflows

USC Viterbi School of Engineering Information Sciences Institute

PegasusAI Team

PegasusAI

USC Viterbi School of Engineering Information Sciences Institute

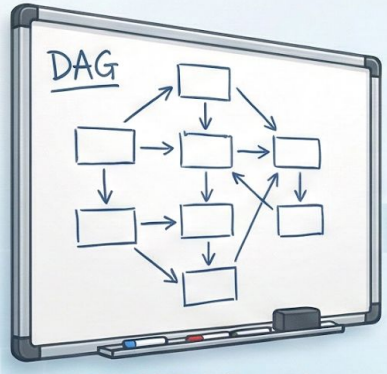
THE UNIVERSITY OF TENNESSEE KNOXVILLE

UMassAmherst

THE UNIVERSITY of NORTH CAROLINA at CHAPEL HILL

NSF

Designing Pegasus WMS Workflows with Claude: An Iterative Process



Initial Workflow Idea on Whiteboard

Executables
(scripts, binaries)

Input Data
(files, datasets)

Requirements & Data

Feedback & Refinement Loop

1. Implement Workflow Generator
Pegasus API + Claude
(AI Pair Programmer)

2. Generate Workflow & Check Consistency
(DAX/YAML)

Script

3. Test Small & Validate
Functional Tests
(Local/Small Scale)

PASS

4. Execute Production Workflow
DAG
(Target Infrastructure)

ACTIVE

PegasusAI

NSF

PegasusAI

Next major version of Pegasus will have AI assistant

NSF

========== Pegasus AI Analysis ==========

The workflow failed due to a missing input file. The job `stage_in_local_local_0_0` encountered an error:
"Expected local file does not exist: /path/to/Alices_Adventures.txt"

Root Cause:
- The required input file is missing from the specified path.
- This prevents the transfer process from completing, causing the workflow to fail.

Next Steps:
1. Verify the file exists at the specified path.
2. Ensure the file path in the workflow configuration matches the actual location.
3. Resubmit the workflow after resolving the file issue.

The remaining unsubmitted jobs (7 total) likely depend on this staged file, so fixing this error will enable further execution.

SWARM team


Ewa Deelman, Ph.D.
USC


Prasanna Balaprakash, Ph.D.
ORNL


Anirban Mandal, Ph.D.
RENCI


Krishnan Raghavan, Ph.D.
ANL


Franck Cappello, Ph.D.
ANL


Jean Luca Bez, Ph.D.
LBNL


Fred Sutter, Ph.D.
ORNL


Zizhong Chen, Ph.D.
UCR


Chandreyee Bhowmick, Ph.D.
ORNL


Hongwei Jin, Ph.D.
ANL


Komal Thareja
RENCI


Erik Scott
RENCI


Shixun Wu
UCR


Sheng Di, Ph.D.
UCR


Suman Raj
USC


Prachi Jadhav
ORNL

USC Viterbi School of Engineering

Argonne NATIONAL LABORATORY

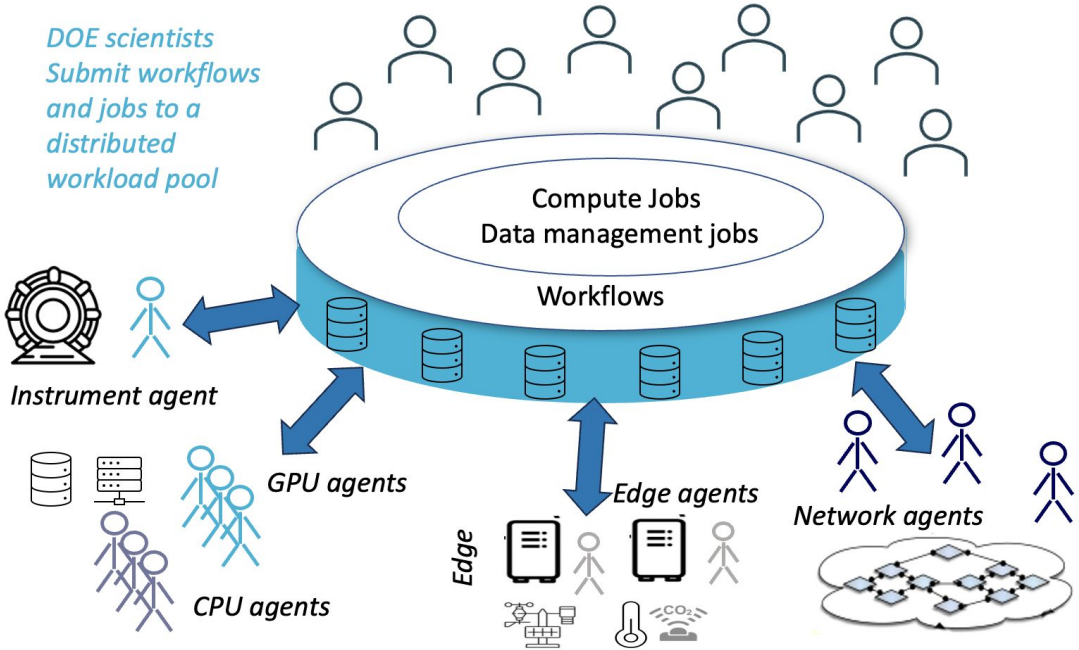
BERKELEY LAB

OAK RIDGE National Laboratory

renci

SWARM: Scientific Workflow Applications on Resilient Metasystem

SWARM aims to improve resilience by employing multi-agent approach



Swarm Intelligence agents select workload to execute and autonomously adapt

Funded by DOE: DE-SC0024387

2023- 2028

SWARM
DISTRIBUTED RESILIENCE SYSTEM

Swarm Intelligence Features

Nature-Inspired Elegance

Our system draws inspiration from natural swarm behavior, offering decentralized and resilient solutions for workflow management.

Self-Monitoring and Self-Healing

The multi-agent approach enables the system to continuously monitor itself and automatically adapt to changing conditions.

Decentralized Coordination

Agents interact locally and with the environment to achieve the desired collective behavior without centralized control.

Challenges

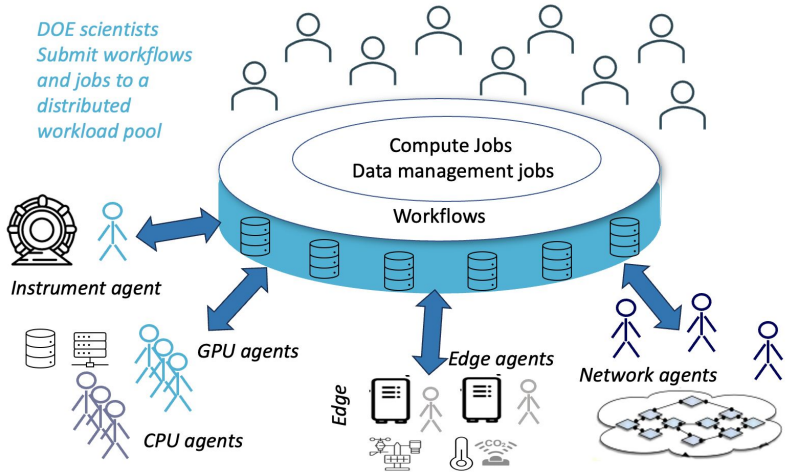
- Slow Consensus
- Limited Perception
- Network Challenges
- Suboptimal Cooperation

SWARM takes novel approach to workflow management

- Infusing SI with knowledge of traditional task scheduling and resource management algorithms
- Taking advantage of the foundational and algorithmic advancements in ML to design more powerful and capable agents
- Integrating conventional methods to ensure resilience, along with SI-based control and autonomy with innovative SI techniques to promote self-monitoring, self-diagnosis, and self-healing
- Leveraging resource capabilities (HPC systems) to support more complex SI algorithms

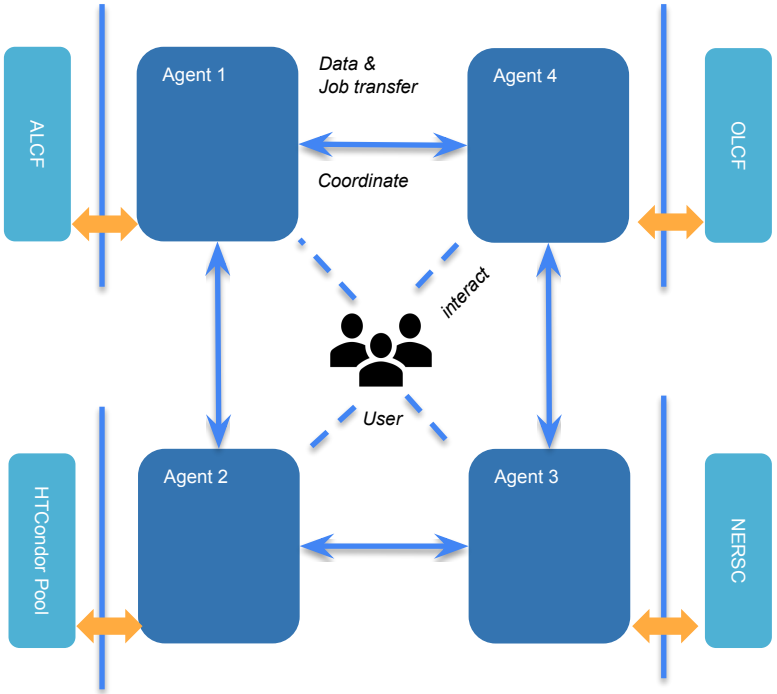
Problem Formulation

- Agents are observing a common job pool
- Relies on resilient distributed store/DB
- Jobs are submitted to that entity



Swarm Intelligence agents select workload to execute and autonomously adapt

- There is no shared state
- Users interact with ”local” or “remote” resource
- Jobs flow between agents



SWARM
DISTRIBUTED RESILIENCE SYSTEM

U.S. DEPARTMENT OF ENERGY

Problem Formulation

Shared state

- Easier to assess the overall system
- Agents need to be able to progress in light of connectivity issues
- Potentially increased delays from job submission to execution

Local state

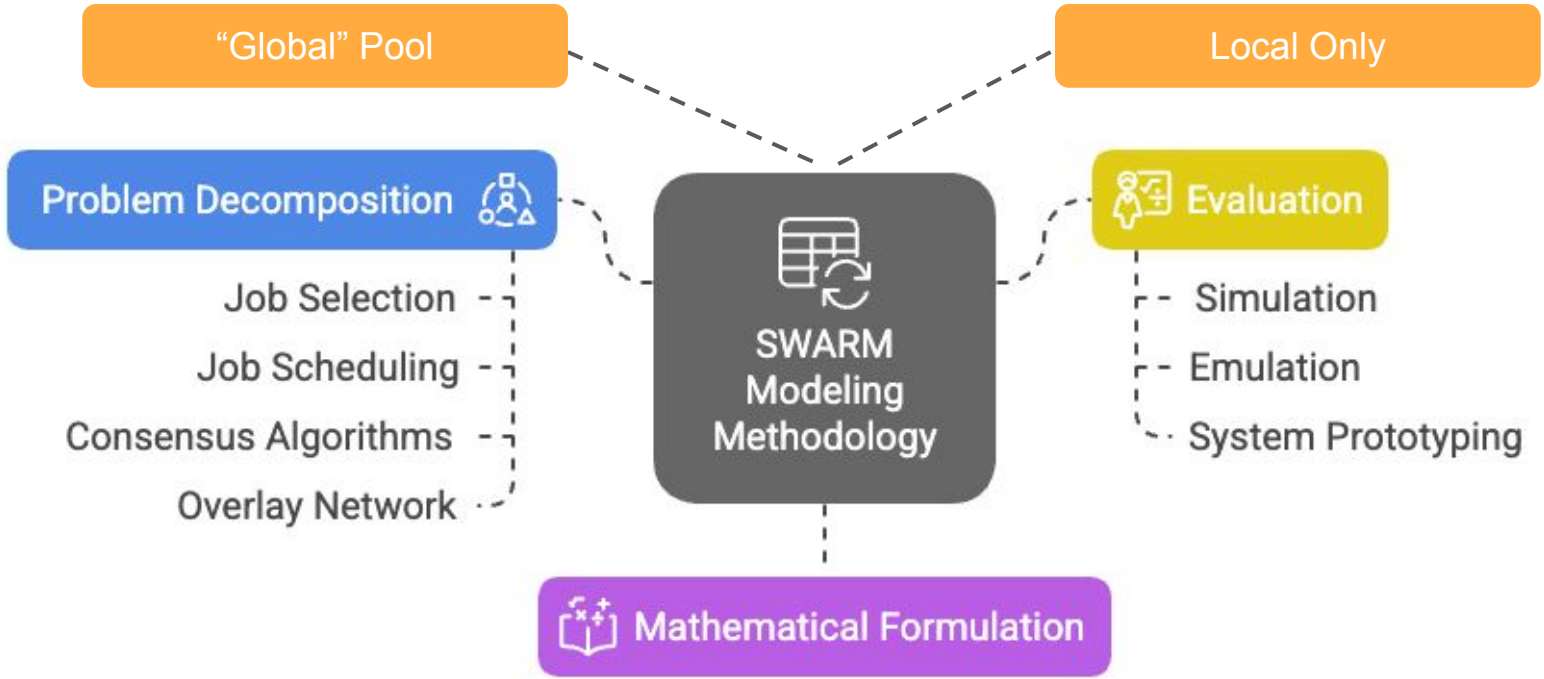
- Harder to assess whether the overall workload is making progress, additional aggregation is needed
- Users need to decide where to submit job,
- Resilience may be more complicated, no common view of workload

Monitoring and Perception: jobs and system status, detect anomalies, predict performance
Reasoning: Simple heuristics to LLMs– depends on underlying platform (where agent executes)
Communication: Which agents to talk with to make decisions: broadcast vs selected agent groups
Consensus: How agents agree on scheduling decisions
Actions: What actions they need to take: (re) submit jobs, kill malfunctioning jobs, repair resources
Learning: What/how should agents learn over time from successes and failures

How do we achieve scalability and resilience?

USC Viterbi School of Engineering · renci · Argonne NATIONAL LABORATORY · BERKELEY LAB · OAK RIDGE National Laboratory · UC RIVERSIDE

SWARM Methodology



SWARM Consensus Algorithm for Job Selection

Anirban Mandal
Komal Thareja
RENCI

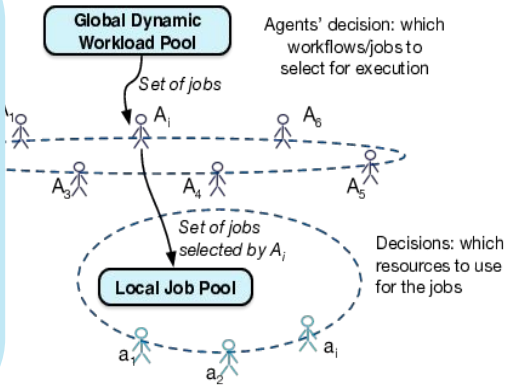
Our Approach: Multi-Agent Systems (MAS) for Resilient Job Selection

- Globally distributed agents perform job selection
- ***Novel***
  consi
- ***Gree***
  Tolera
  resilie
- Conse
  -
  -
  -
- All agents communicate with all other agents

Assumptions:

1. Each agent knows the capabilities and workload of other agents and can compute their job selection
2. Each agents communicates with each other

Relaxing the assumption:
Agents can learn each other's capabilities over time, and potentially anticipate their selections

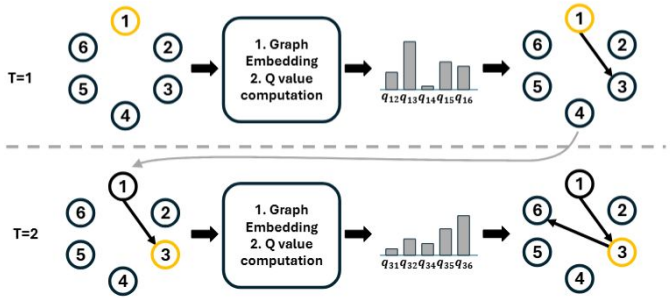


- **Improved scheduling latency by 63.5 %**
- **Improved idle time by 63.8 % compared to PBFT**

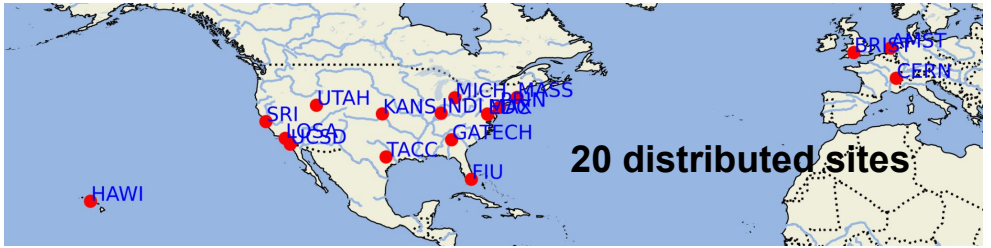
SWARM Overlay Network

Franck Cappello
Shixun Wu
ANL, UCR

- **Motivation**
  - Existing membership protocols use logical ring, not considering underlying **physical latency**.
  - Consensus on membership is upper bounded by the **diameter of the overlay topology**.
- **Challenge:** Degree-constrained diameter minimization is an NP-hard problem.
- **Our Contributions**
  - **D**iameter-**G**uided **R**ing **O**ptimization (**DGRO**), a reinforcement-learning based overlay construction that outperforms existing overlay topology.



Our Node Selection with Deep Q-Network.



Fabric Testbed: https://portal.fabric-testbed.net/

Action: Selecting the next node to connect.
Reward: Reduction in network diameter between consecutive steps, with an additional latency penalty/bonus to encourage low-latency links
Q-function: A neural network estimates the expected future reward of connecting the current node to candidate node

K-Ring constructed by DGRO outperforms Chord, Nearest Neighbour, Rapid, Perigee.

Example Simulator Prompt for LLM

```
You are an expert HPC resource manager, and your task is
to schedule jobs in a high-performance computing (HPC)
environment. Use the current system state, job queue,
scratchpad (decision history), and fairness indicators to
make well-balanced decisions.
System capacity: 256 nodes, 2048 GB memory
Current time: 0
Available Nodes: 256
Available Memory: 2048 GB
```

```
Running Jobs:
None
Completed Jobs:
None
Waiting Jobs (eligible to schedule):
None
# Scratchpad (Decision History)
(nothing yet)
```

```
Your scheduling objectives are:
You must balance all of the following:
  • Fairness: Minimize variance in user wait times. Avoid
    starving any user.
  • Makespan: Minimize total time to finish all jobs.
  • Utilization: Maximize Node & memory usage over time
    (avoid idle resources).
  • Throughput: Maximize the number of jobs completed per
    unit time.
  • Feasibility: Do not exceed 256 Nodes or 2048 GB memory
    at any time.
```

```
Trade-offs are allowed. Do not over-optimize one metric at
the expense of others.
For example:
  • Prioritizing a long-waiting job improves fairness, but
    may slightly hurt makespan.
  • Choosing short jobs improves throughput, but may
    increase wait time for large jobs.
```

```
Decide:
(1) Which job should be started now (if any)?
(2) Justify your decision in thought.
(3) Return only one of:
    • StartJob(job_id=X)
    • BackfillJob(job_id=Y)
    • Delay
    • Stop (when all jobs have been scheduled)
Output format:
Thought: <your reasoning>
Action: <your action>
```

Multi-objective scheduling comparison

Heterogeneous workload

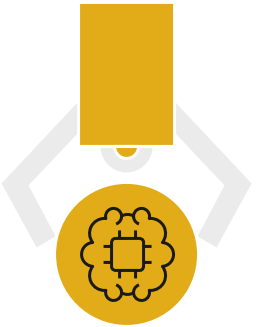
- We use OR-Tools from Google to optimize the makespan
- First Come First Serve
- Shortest Job First

LLM-based approach provides flexibility, you can change criteria on the fly

Progression of Automation

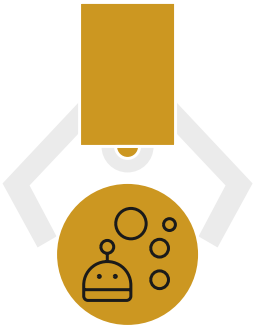
Pegasus

Computation automation



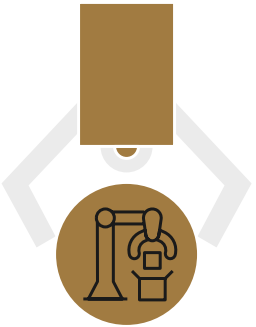
Pegasus AI

Infusing AI techniques



Agentic Workflows

Based on swarm intelligence



Self-driving Labs

Automation of experimental workflows

SWARM for Scientific Workflows at a User Facility

Instrument

Agent

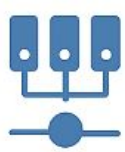
Checks instrument status, Checks data quality, triggers pre-processing

Edge Cluster

Agent

Applies denoising, checks for patterns, starts classification using ML

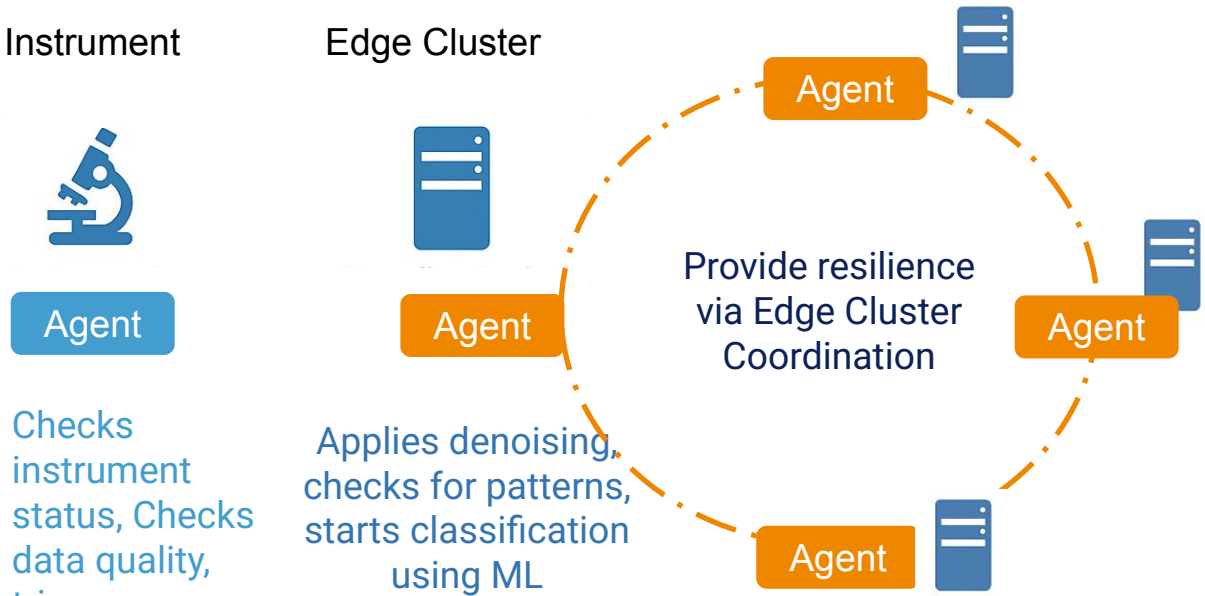
HPC Cluster



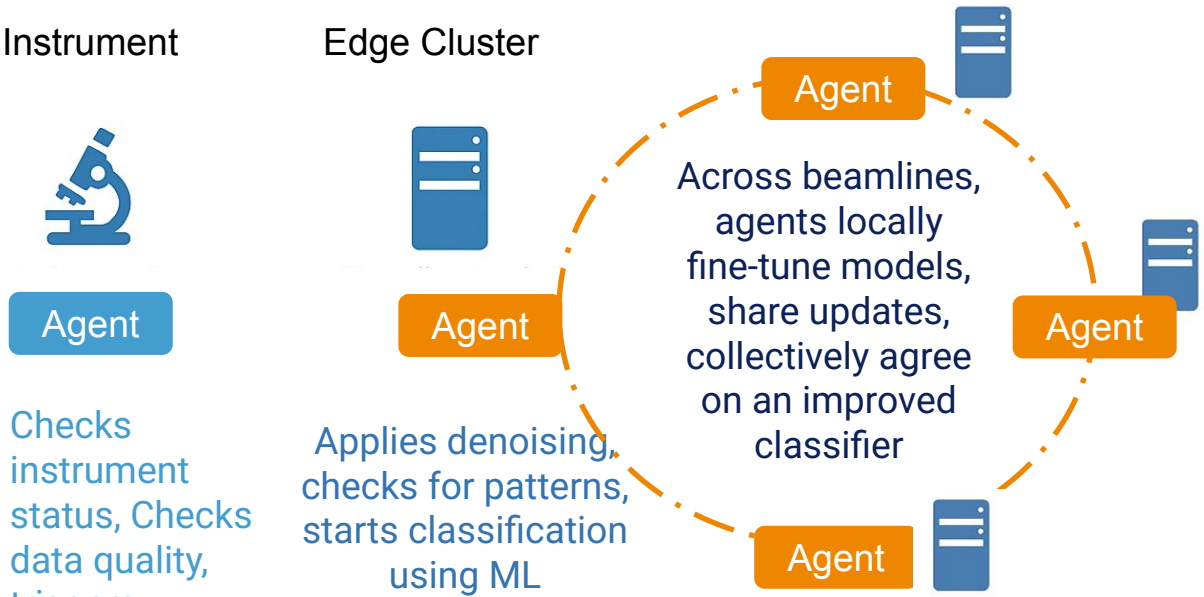
Agent

Handles full 3D reconstruction and/or simulation matching

SWARM for Scientific Workflows at a User Facility



SWARM for Scientific Workflows at a User Facility



Federated Learning:
Local model training, peer-to-peer exchange of updates, decentralized consensus

Future Work

- Integrate system components
- Explore tradeoffs in problem representation
- Test behavior (resilience, scalability, performance) under multiple failure conditions occurring in real world scenarios and real workloads
- Materialize our research into a prototype SWARM Workflow Management System for workload scheduling
- *Potentially explore application to self-driving-labs*

https://swarm-workflows.org/
https://pegasus.isi.edu