WfBench: Automated Generation of Scientific Workflow Benchmarks

Tainã Coleman[1]
Henri Casanova[2]
Ketan Maheshwari[3]
Loïc Pottier[1]
Sean R. Wilkinson[3]
Justin Wozniak[4]
Frédéric Suter[3]
Mallikarjun Shankar[3]
Rafael Ferreira da Silva[1,3]

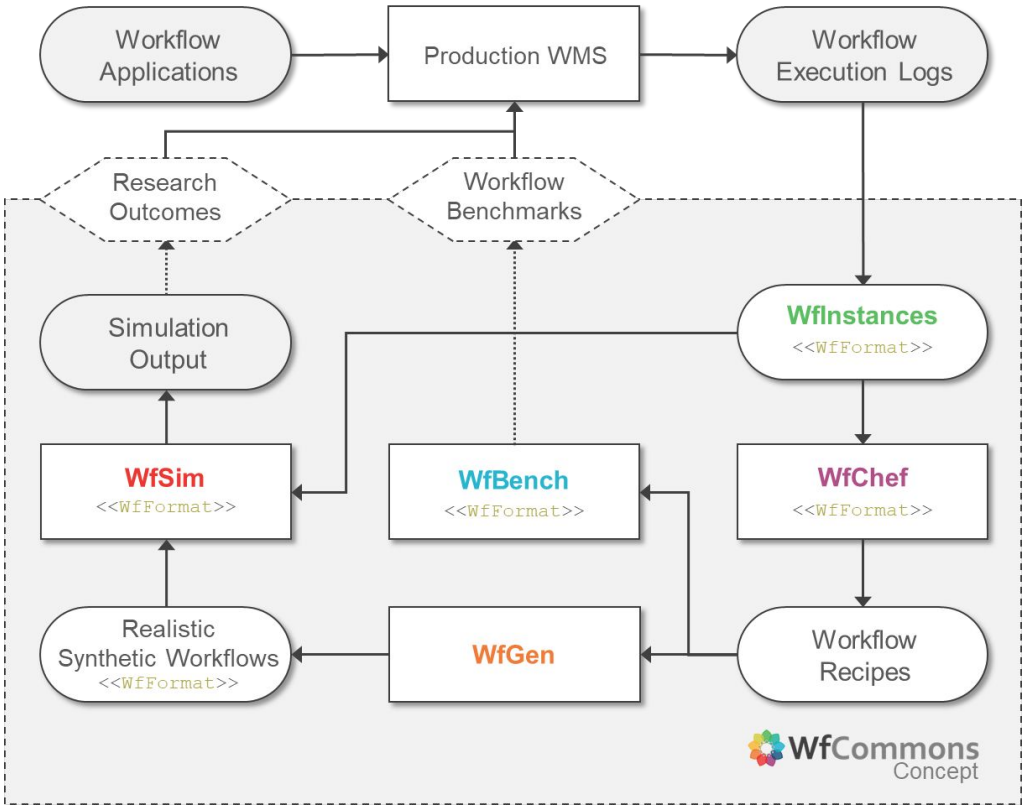
WfCommons

[1] USC Information Sciences Institute, [2] University of Hawai'i at Mãnoa, [3] Oak Ridge National Laboratory, [4] Argonne National Laboratory

WfCommons

https://wfcommons.org

WfCommons is a framework that provides a collection of tools for analyzing **workflow execution traces**, producing realistic **synthetic workflow traces**, and **benchmarking** / **simulating workflow executions**.



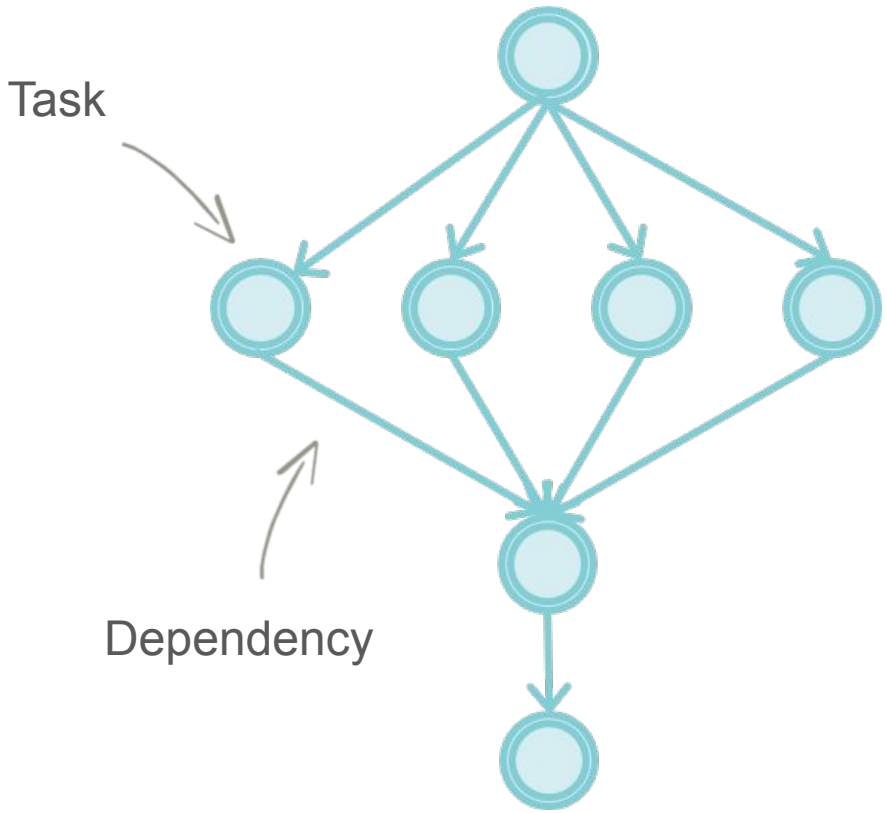
Scientific Workflows

directed acyclic graph

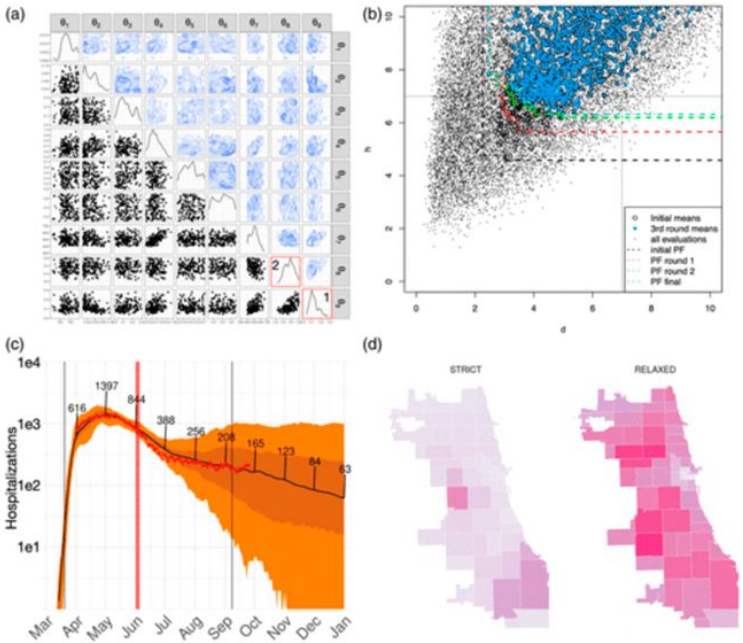
DAG

A task represents a computation with inputs and outputs (often a **program** or script)

Dependencies typically represent **data flow** but also **conditions, exceptions, user triggered action**, etc.



Example of Workflows-Enabled Research



A population data-driven workflow for COVID-19 modeling and learning

Jonathan Ozik, Justin M Wozniak, Nicholson Collier, Charles M Macal, Mickaël Binois

First Published September 10, 2021 | Research Article | Check for updates
https://doi.org/10.1177/10943420211035164

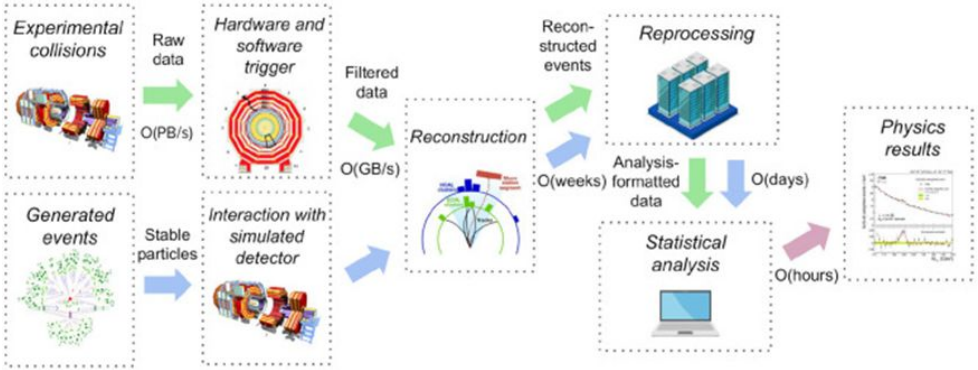
Figure 2. (a) Joint posterior distributions of CityCOVID input parameters (Table 1) from sequential ABC, (b) successive Pareto fronts of errors in deaths (x-axis) and hospitalizations (y-axis) from HVR workflow, (c) COVID-19 attributed hospitalization outputs from CityCOVID (red dots: empirical Chicago data, dark line: median simulation output, dark band: 50% simulation intervals, and light band: 95% simulation intervals), (d) CityCOVID zip code level snapshot of weekly infection outputs at 47 days after June 3, 2020, initial easing of restrictions in Chicago for two scenarios (strict: population wide adherence to protective behaviors, that is, θ_9 is maintained as reopening occurs, relaxed: gradual increase of θ_9 to a value corresponding to 80% viral transmission reduction).

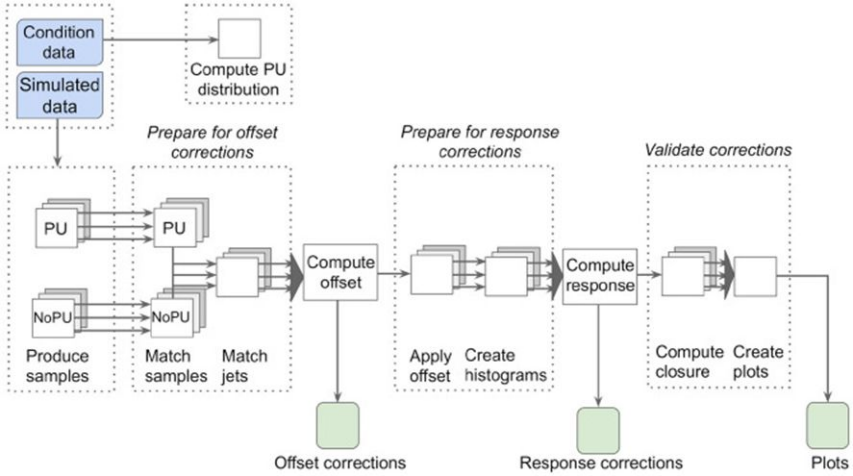
Example of Workflows-Enabled Research

Front Big Data. 2021; 4: 661501.
Published online 2021 May 7. doi: 10.3389/fdata.2021.661501
PMCID: PMC8138321
PMID: 34027400

Scalable Declarative HEP Analysis Workflows for Containerised Compute Clouds

Tibor Šimko,[1,*] Lukas Alexander Heinrich,[1] Clemens Lange,[1,*] Adelina Eleonora Lintuluoto,[1,2] Danika Marina MacDonell,[3,*] Audrius Mečionis,[1] Diego Rodríguez Rodríguez,[1] Parth Shandilya,[1,4] and Marco Vidal García[1]





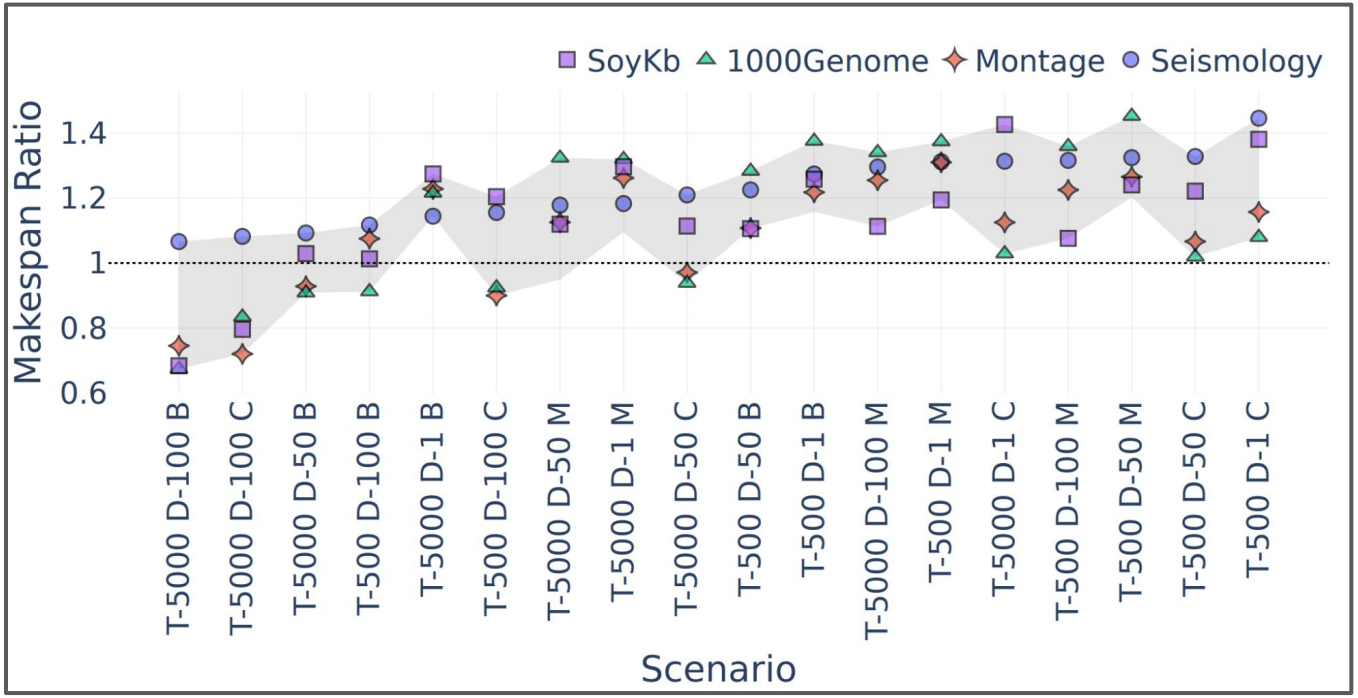
Motivation

Workflows are becoming **more complex** and require more sophisticated management capabilities

Workflows can now have **millions of tasks** and analyze **terabyte-scale datasets** that can take milliseconds to hours on distributed heterogeneous systems.

Large variety of **WMS** were created to accommodate these feature demands and to meet the specific needs of a domain

The **few** existing benchmarks for workflows detect **some but not all** of the important features of production workflows



- Same amount of compute work
- #tasks: 500, 5000
- Data: 1, 50, 100GB
- CPU-bound, memory-bound, balanced
- Ratio: Skylake/Cascadelake

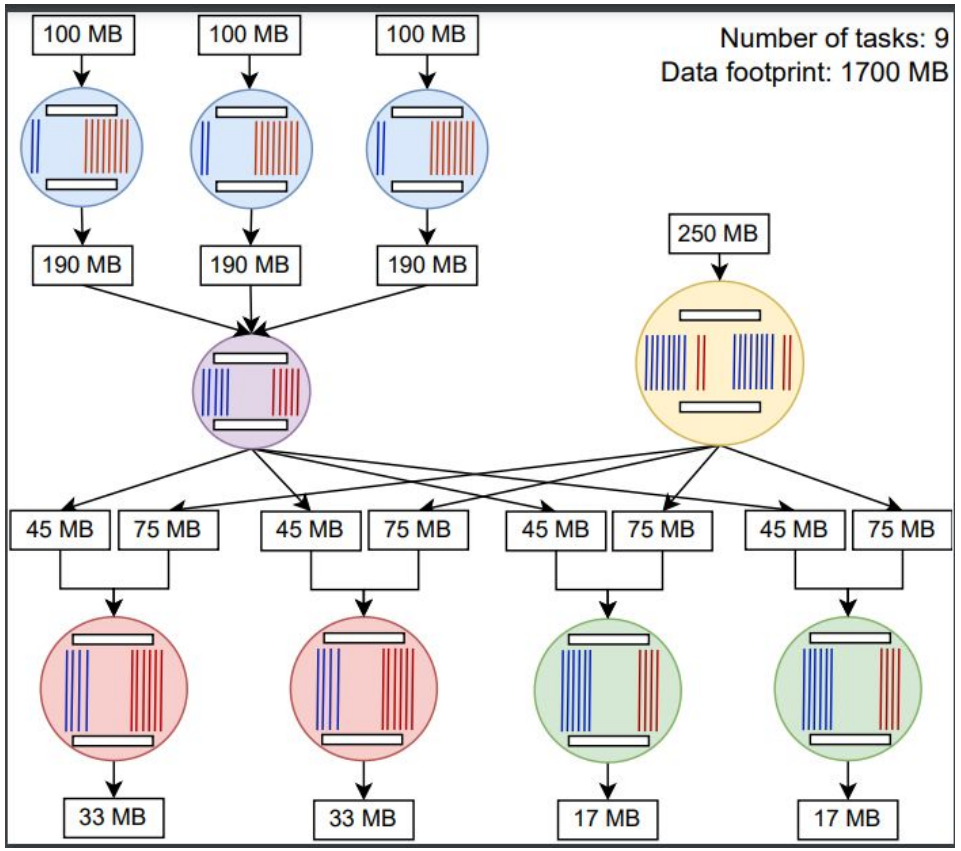
Results differ significantly across configurations

Very difficult to explain or predict, workflow (relative) makespans based on platform and configuration

WfBench

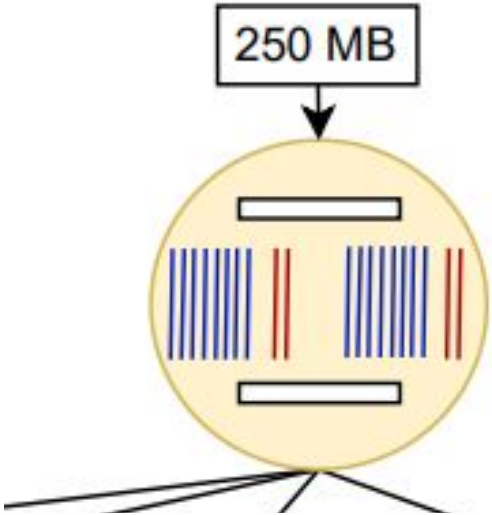
Representative **task** benchmarks

Representative **workflow** benchmarks with multiple tasks

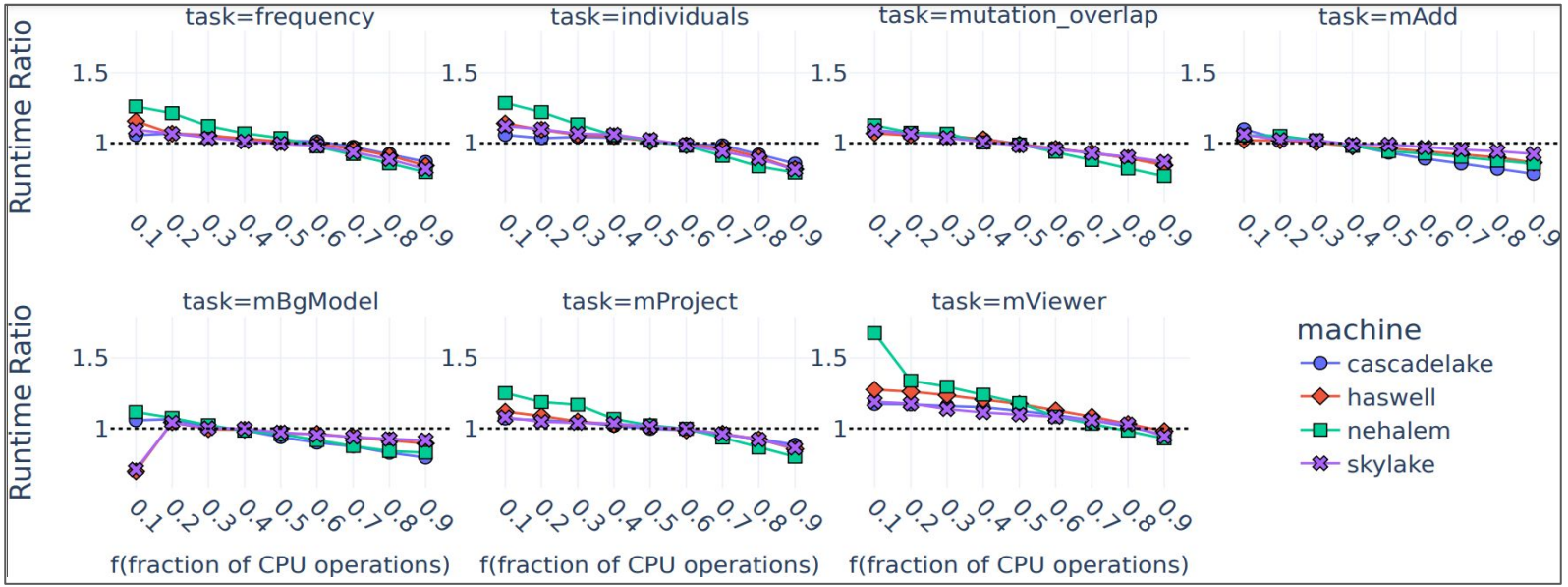


Representative workflow task benchmark

- Read input
- Compute
  - Inputs:
    - cpuwork
    - memwork
    - n cores
    - non-mem computation (f)
  - CPU-intensive: Calculates π up to cpuwork
  - Mem-intensive: Access random positions in array adding one unit to it up until memwork
- Write output



Validation

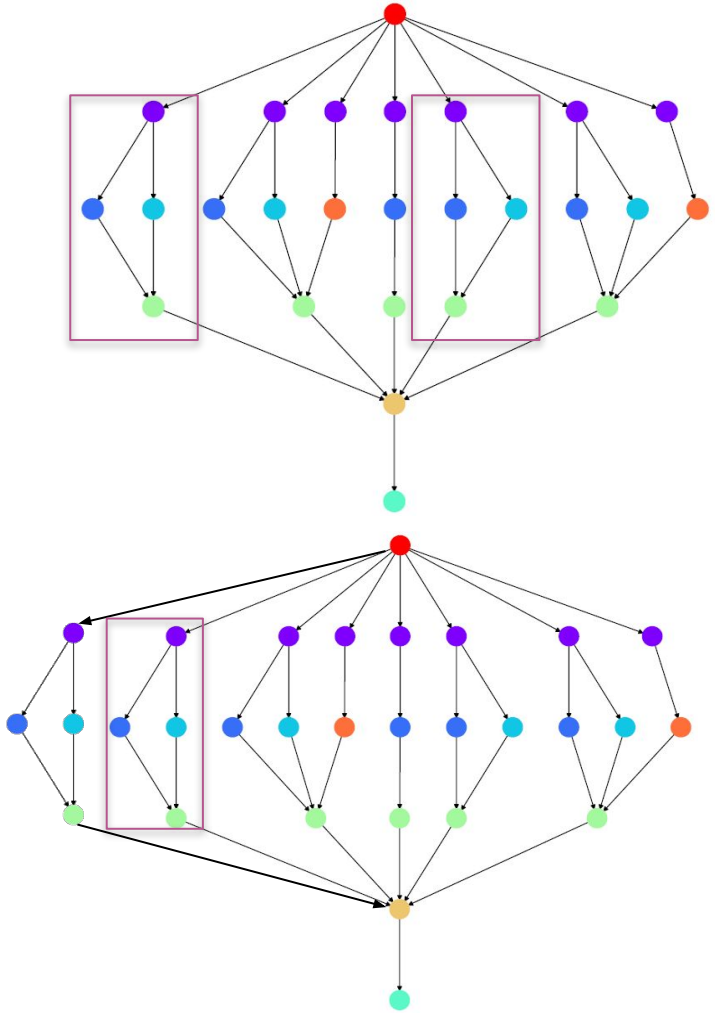


Is it possible to configure our workflow task benchmark so that its performance behavior is similar to that of each of these real workflow tasks?

Representative Workflow Benchmark

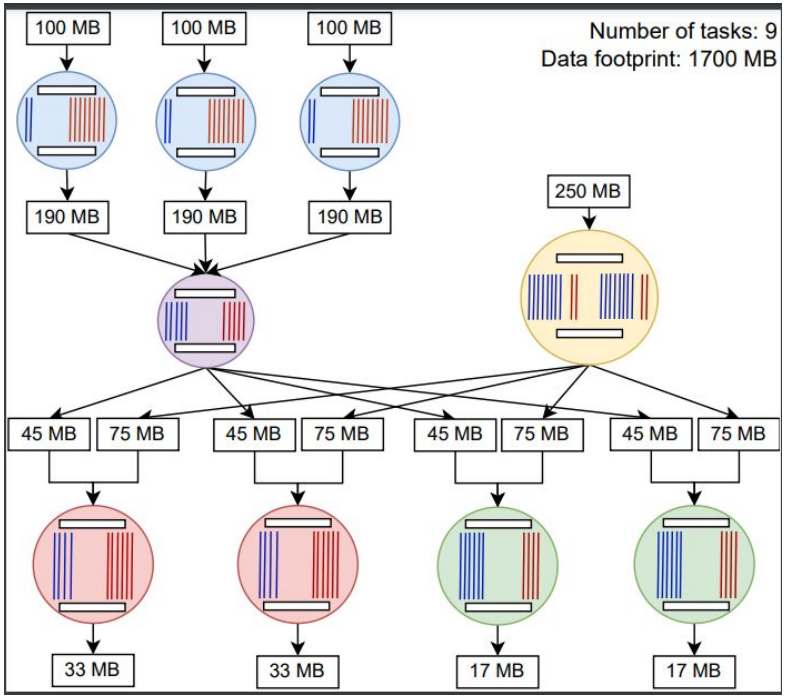
WfChef

- Automatic generator of realistic synthetic workflow instances
- Inputs
  - Set of real world workflow instances
  - Desired instance size (number of tasks)
- Analyzes the instances
- Records common patterns
- Creates a recipe
- Replicate patterns



Representative Workflow Benchmark

- Input:
  - Desired # tasks
  - WfChef workflow recipe
- Generation:
  - Uses the recipe to generate task graph
  - For each task user can specify:
    - (n, cpuwork, memwork, f)
    - Data volume/task or total data footprint
- Output:
  - JSON object that fully describes workflow
- JSON + Tasks benchmarks = Workflow Benchmark



Experimental Evaluation - Set up

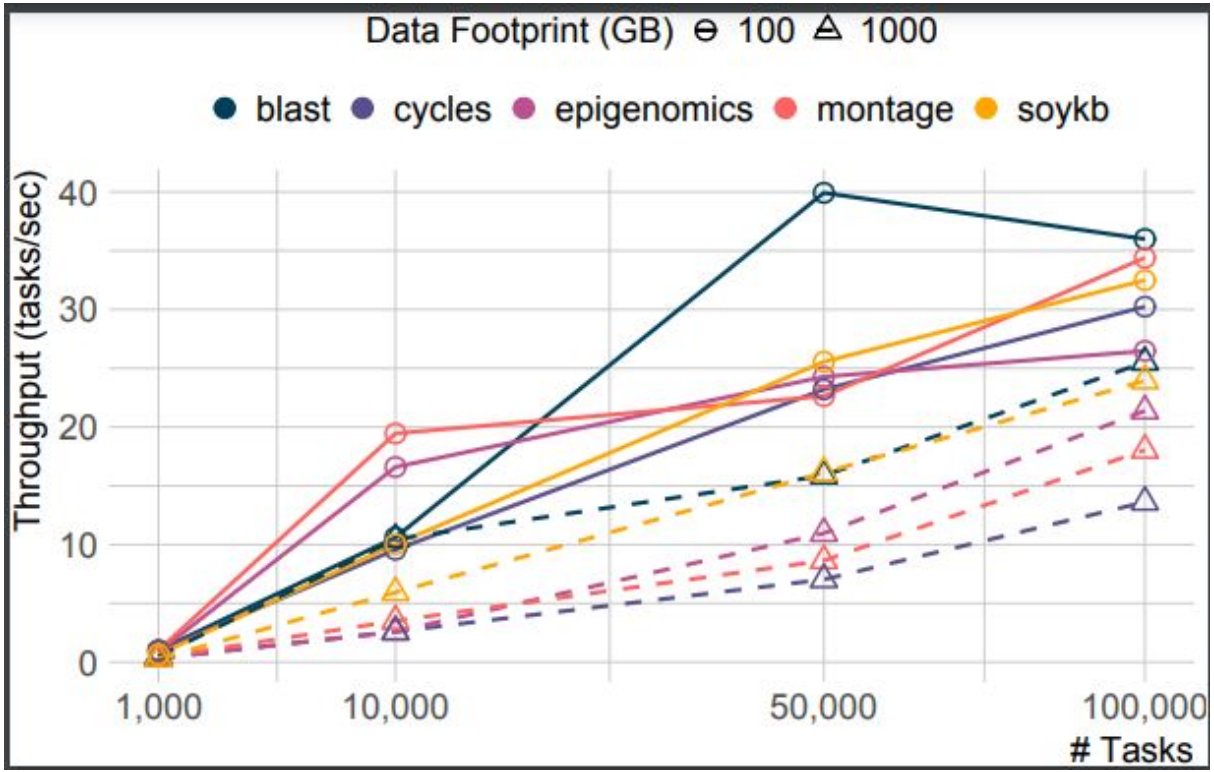
- 40 instances:
  - 1k, 10k, 50k and 100k tasks
  - Total data footprint: 100GB, 1TB
- Run on ORNL's Summit
  - Swift/T workflow system
  - 40 CPU cores per compute node
  - Total # nodes = (0.1 × #tasks)/40
  - All tasks:
    - cpuwork = 500
    - memwork = 0

Experimental Evaluation

Workflow throughput (#tasks/sec)

↑ data footprint ↓ throughput

↑ #tasks ↑ throughput



Experimental Evaluation

Blast vs Epigenomics

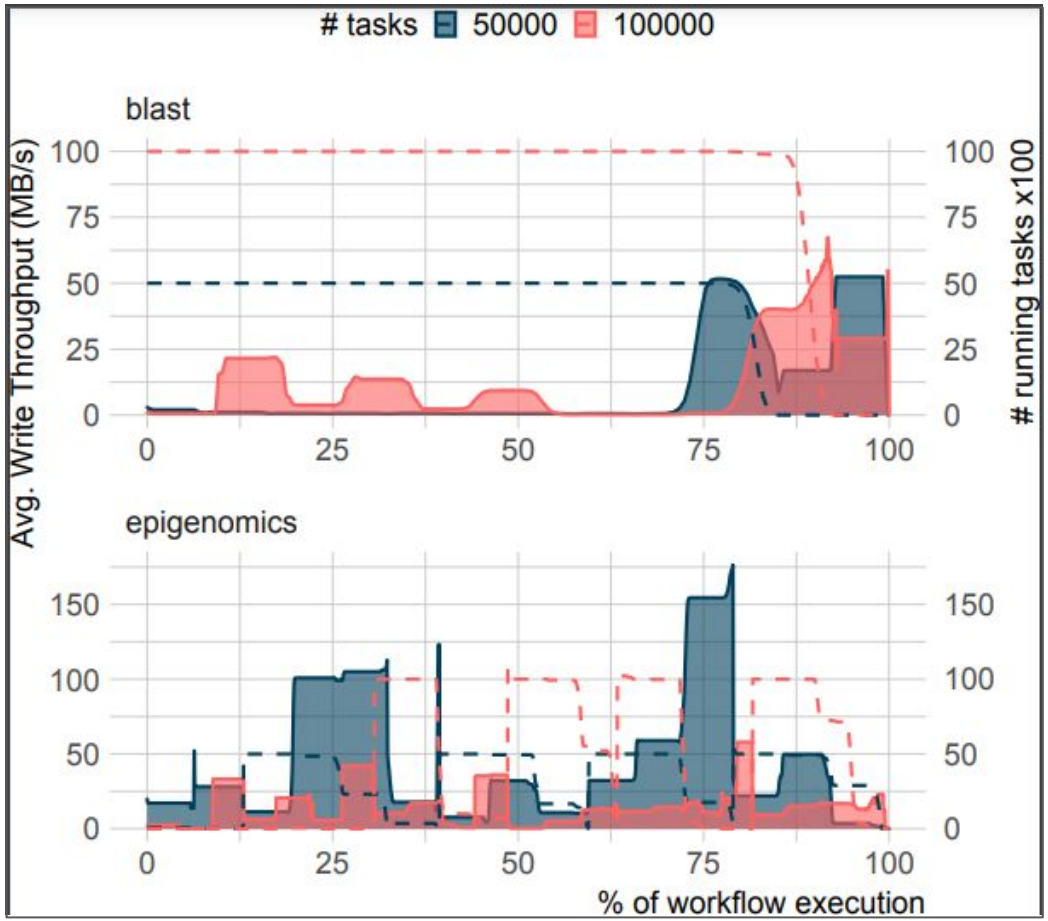
Average write throughput (MB/s) - overlapping area chart

Dashed lines are the #concurrent running tasks

↑ #tasks ↓throughput

Causes:

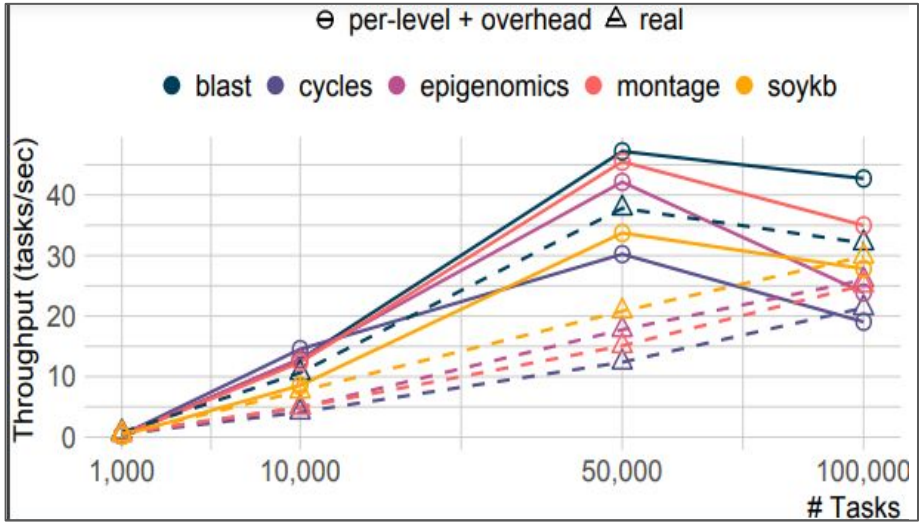
- Small files
- 2x number of files



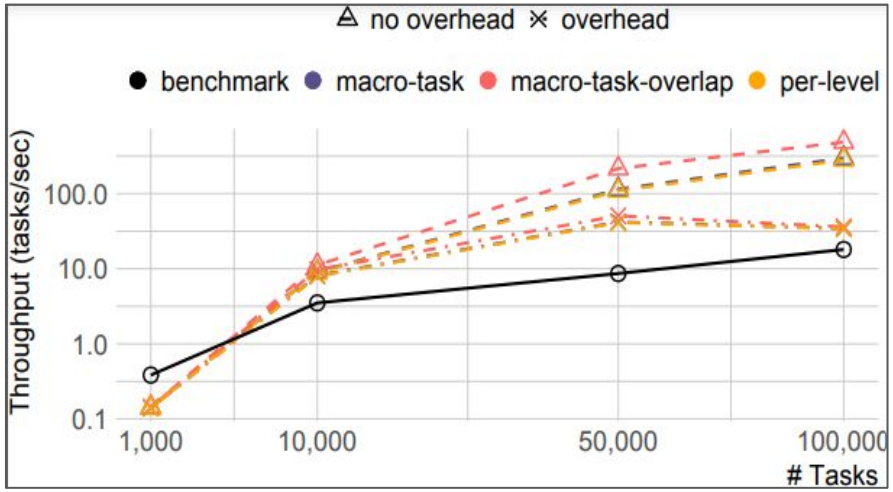
Benchmark Usefulness

Models

Single macro-task-no-overlap
Single macro-task-overlap
Per-level



Montage 1TB data footprint



Next Steps

Support workflows that tasks are MPI-based parallel program

Support GPU workflow task benchmark

Extend WfBench to benchmarks that perform in-situ executions

THANK YOU

Tainã Coleman
tgcolema@usc.edu
https://tainacoleman.com/
https://wfcommons.org/