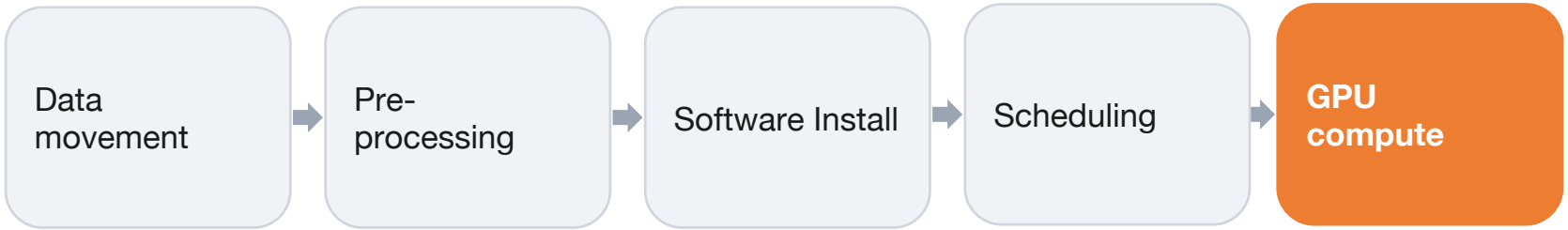
From AlphaFold3 to AI Facilitation

A translational computer science journey

CHTC HTCondor Software Suite PATh

It's not just about GPUs

The accelerator is one box in a much longer workflow.



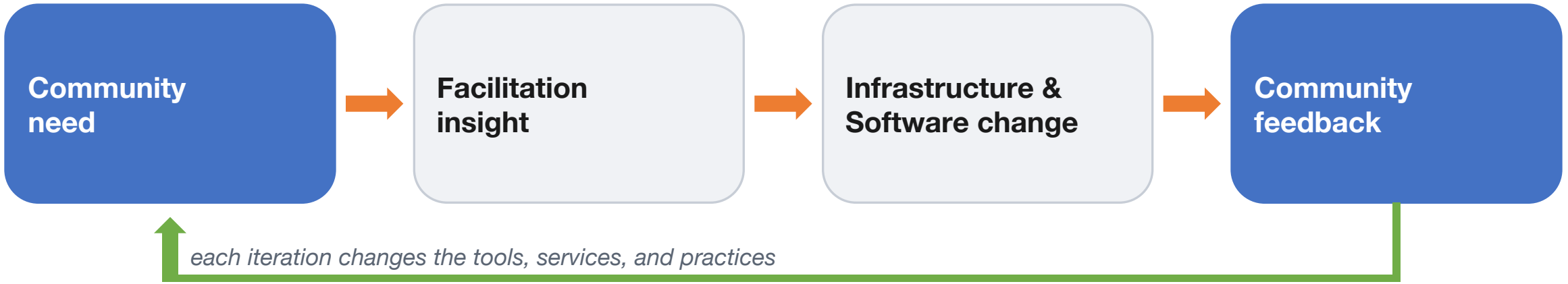
Adding GPUs is the "easy" part. The hard part is everything around them — and making it usable, schedulable, and reproducible.*

*Your milage may vary

CHTC HTCondor Software Suite PATh

Translational computer science

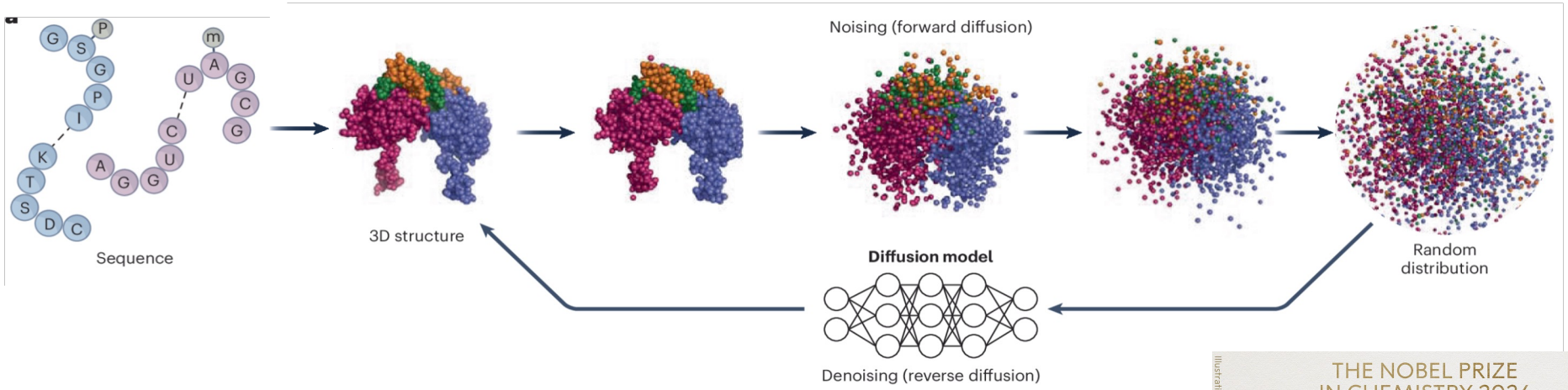
Turning researcher needs into usable infrastructure — then letting the community reshape it.



CHTC HTCondor Software Suite PATh

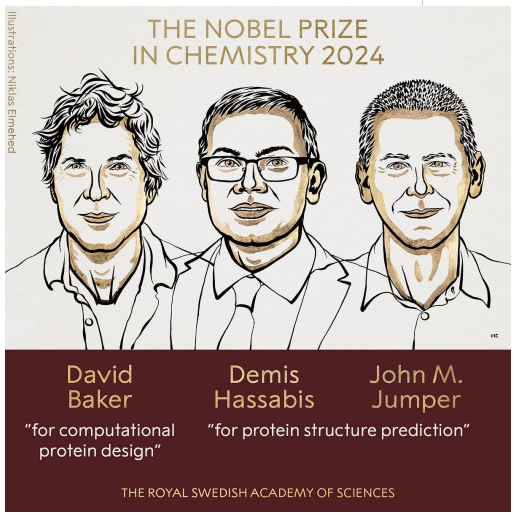
What is AlphaFold3?

AlphaFold3 is a **generative ML model that predicts the structures of biomolecules.**



AlphaFold3 is able to predict the structures of biomolecules, such as:

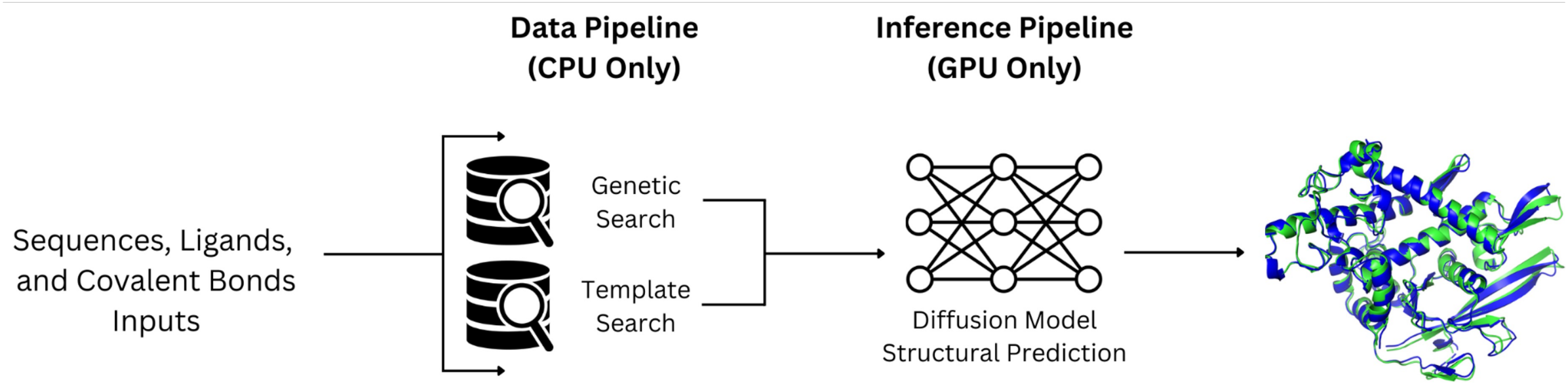
- Proteins
- DNA
- RNA
- Small molecules (ligands, drugs)
- Ions and modified residues
- Multi-molecule complexes



CHTC HTCondor Software Suite PATh

How AlphaFold3 works

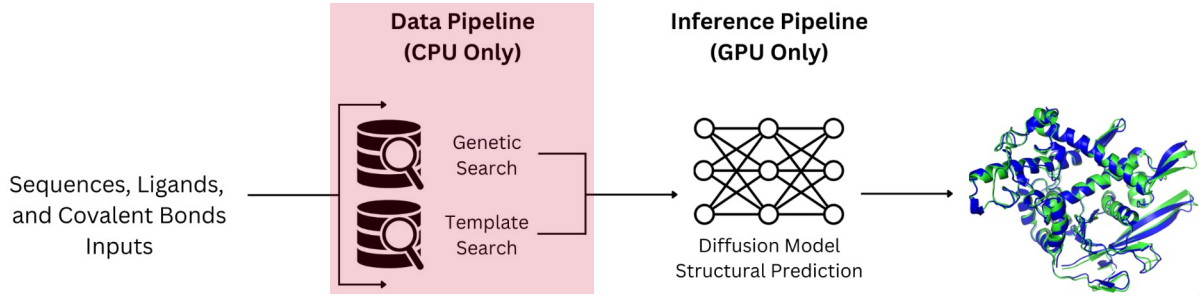
Sequence and database search → alignment generation → templates → GPU inference.



CHTC HTCondor Software Suite PATh

How AlphaFold3 works

Sequence and database search → alignment generation → templates → GPU inference.



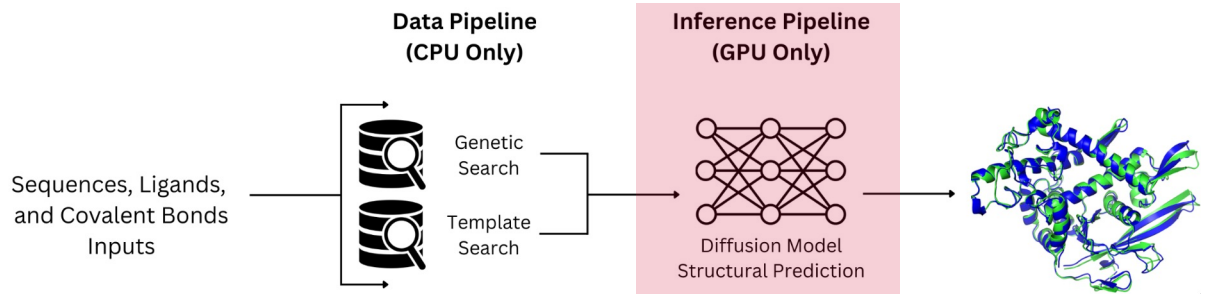
Data Pipeline

- Prepares alignments and inputs
- Depends on AF3 Databases (~750 GB)
  - Requires high disk usage
- Uses only a few (<8) CPUs
- Longer run times (>1hr)†

*Run times depend on the input sequence(s) size and depth of alignments

How AlphaFold3 works

Sequence and database search → alignment generation → templates → GPU inference.



Inference Pipeline

- Preforms the structural prediction via model inference
- Requires GPUs
- No Databases requirement
- Shorter run time (<1hr) *

*Run times depend on the input sequence(s) size and depth of alignments

AlphaFold3 arrives at CHTC / OSPool

The first question was simple:
Can researchers run AlphaFold3 here?

The deeper question became:
Could we support it?

The implementation we inherited

Sameer D'Costa
GLBRC

A repo from a fellow UW–Madison researcher. It worked — by shipping the entire database to every job.

AlphaFold3 Databases

Researcher repo
(UW–Madison/GLBRC)
~750 GB databases

databases transferred with every job

Batched job

many alignments bundled into one job resulting in longer runtimes

+ very large per-job disk allocation

Runs on CHTC only

local cluster with the disk to hold 750 GB
— not realistic on the OSPool

Room for Improvements

- **Limited to CHTC**
  Not OSPool-Friendly
- **Large per-job disk allocations**
- **Transfers cumbersome & failure prone**

CHTC HTCondor Software Suite PATh

The call to adventure: a researcher's large complex

We have a very large **multi-chained protein structure** I want to predict using AlphaFold3. Each chain is **~998 aa** and we want to predict the **11-, 12-, and 24-mer** of the structure.

Raison Dsouza – Morgridge

The call to adventure: a researcher's large complex

OOOF… Those are **pretty big complexes**. Let's see what we can do!

Raison Dsouza – Morgridge

The limits of our scope of support become visible

These structures pushed past our scope of knowledge — and exposed where support ran out.

GPU memory

Large complexes exceeded single-GPU memory (>140 GB of vRAM)

Runtime

Jobs ran long and unpredictably

Job shape

CPU / GPU / disk needs were unclear up front

Support confidence

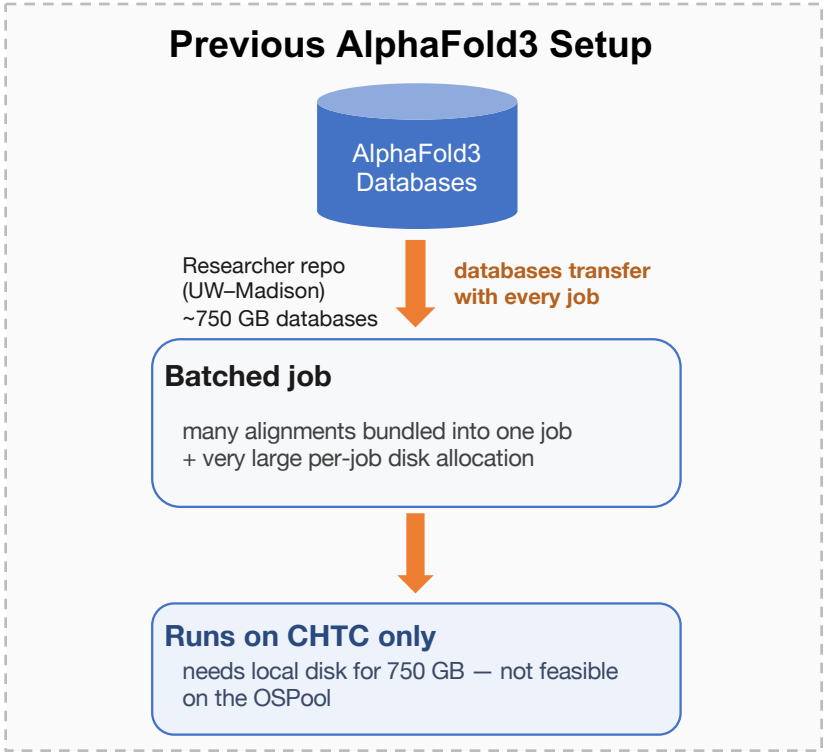
Hard to tell a user what would actually work

CHTC HTCondor Software Suite PATh

Inheriting the AlphaFold3 repo

The repo proved AlphaFold3 could run. Running and supporting are different things.

- **Unclear job shapes**
- **Large database handling**
- **Runtime variability**
- **Memory pressure**
- **GPU scheduling constraints**
- **Documentation gaps**
- **Repeated expensive preprocessing**
- **Ownership of the material**



CHTC HTCondor Software Suite PATh

Supporting AlphaFold3 at CHTC / OSPool

The first question was simple:
Can researchers run AlphaFold3 here?

The deeper question became:
How do we better support it?

CHTC HTCondor Software Suite PATh

Trial 1: understanding job shapes

To guide users, we first had to characterize how an AlphaFold3 job actually behaves.

CPU

core counts that actually help

GPU

device + memory required

Memory

peak RAM under load

Disk

scratch + database footprint

Runtime

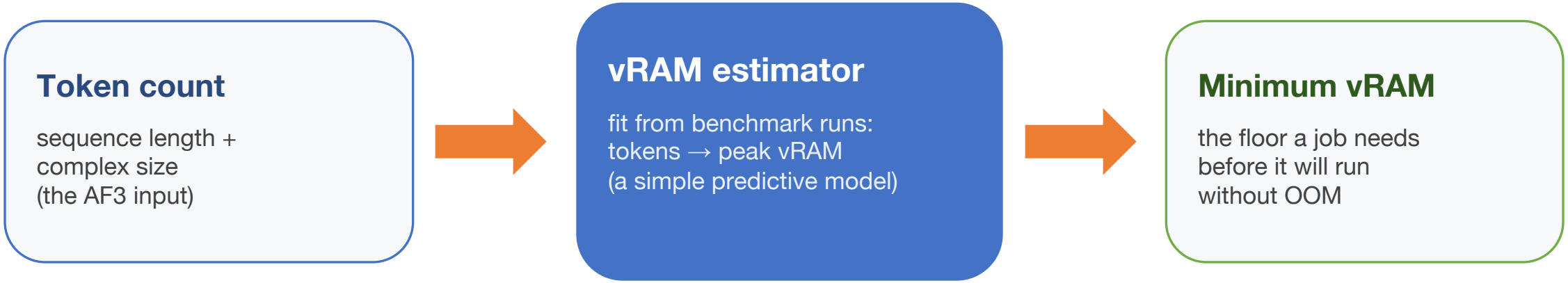
hours, highly variable

A job's "shape" — not the GPU alone — determines where it can run and whether we can support it.

CHTC HTCondor Software Suite PATh

Predicting vRAM before the job runs

AlphaFold3's inference vRAM scales with token count — so we modeled it as a minimum-vRAM estimator.



Right-size the GPU

match each job to a GPU with enough memory

Fewer OOM failures

catch jobs that won't fit before they launch

Predictable scheduling

turn vRAM from a guess into a known requirement

Job shape became a number we could compute — the foundation for routing AF3 to the right GPU.

CHTC HTCondor Software Suite PATh

The call to adventure: a researcher's large complex

We ran the numbers. Looks like your 11-, 12-, and 24-mer complexes will require **~190, 210, and 425 GB of vRAM** each. We don't have GPUs of that capacity, but **we can give unified virtual memory at try!**

Raison Dsouza – Morgridge

CHTC HTCondor Software Suite PATh

The ordeal: pushing past limits with unified memory

When a structure needs more memory than a single GPU has, the job simply cannot run — until the memory model changes.

BEFORE

Structure needs more memory than a single GPU holds — the job fails.

unified memory

AFTER — UNIFIED MEMORY

GPU memory + system memory are pooled. Structures that overflowed one GPU now fit — trading some speed for reach.

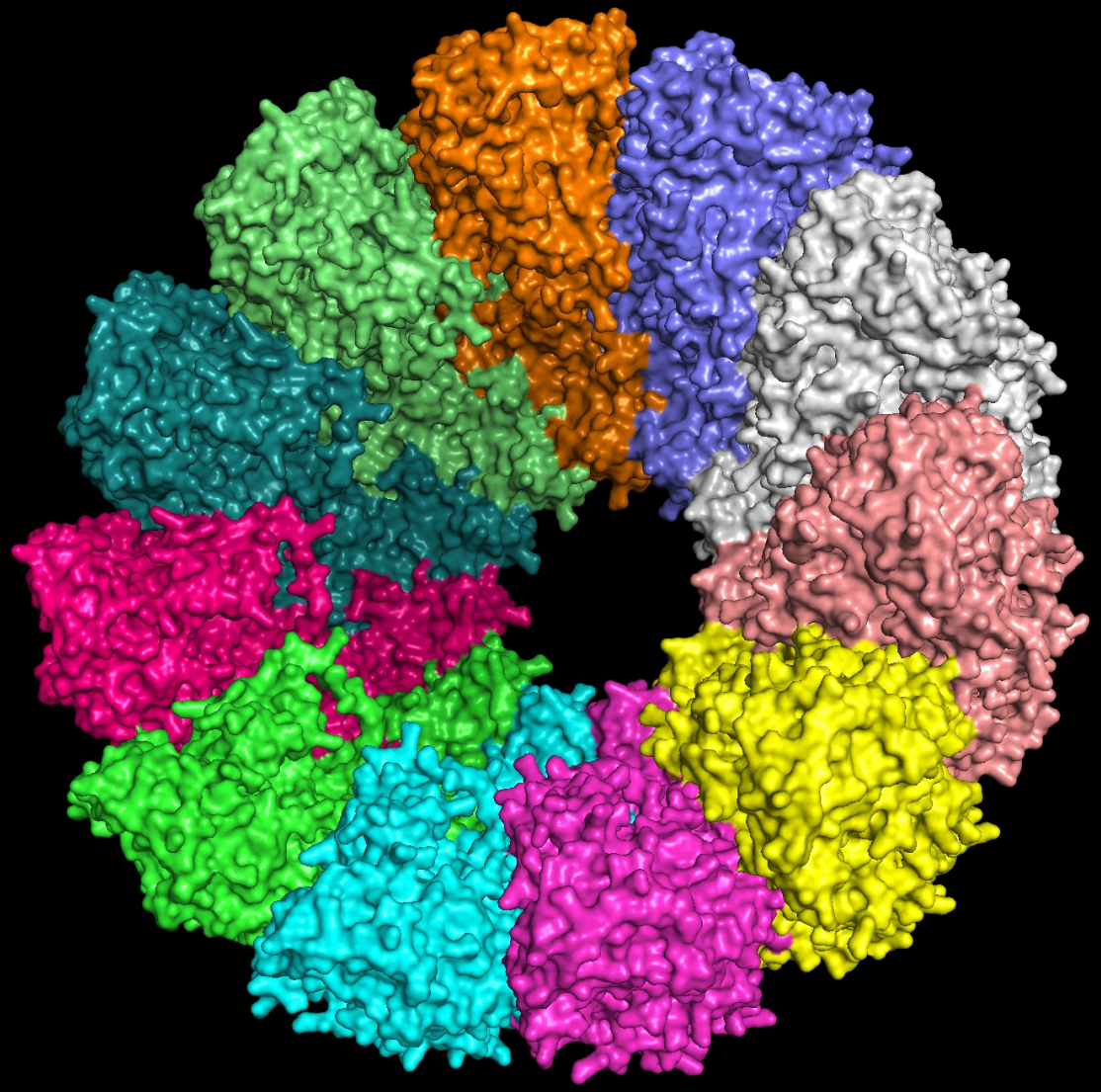
The catch in a multi-tenant setting

Unified memory needs a contiguous block — hard to get while neighbors are running. So, we drain and prioritize the node until the UVM job matches; remaining partitioned resources then backfill with other jobs.

How a UVM job lands:

CHTC HTCondor Software Suite PATh

Victory!



The victory: Raison's structures become supportable

The win was not one solved case. It was redrawing the line between impossible, unsupported, and supportable with care.

Impossible

A hard physical limit — genuinely cannot run.

Raison's jobs started here.

Supportable with care

Can run with direct RCF intervention and assistance

Raison's 11-, 12-, and 24-mer now live here.

Supported as Self-Service

Can run as a self-service job without RCF/Admin intervention

Ultra-large structures moved across the boundary →

Facilitation moved the boundary — cases weren't possible we can now support.

CHTC HTCondor Software Suite PATh

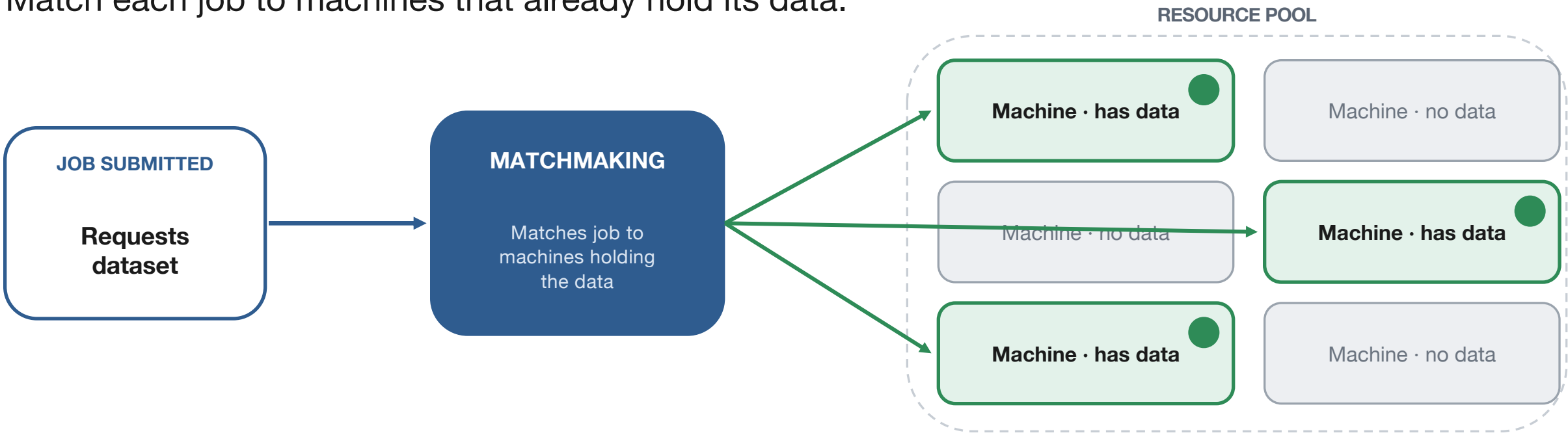
Trial 2: understanding the data problem

AlphaFold3 depends on large reference databases — and moving them again for every job does not scale. Significant barrier to running AlphaFold3 on the OSPool.

Facilitation insight: the dataset itself should be treated as a property of infrastructure.

Data-as-a-Resource

Treat data as a schedulable resource of the execution point.
Match each job to machines that already hold its data.

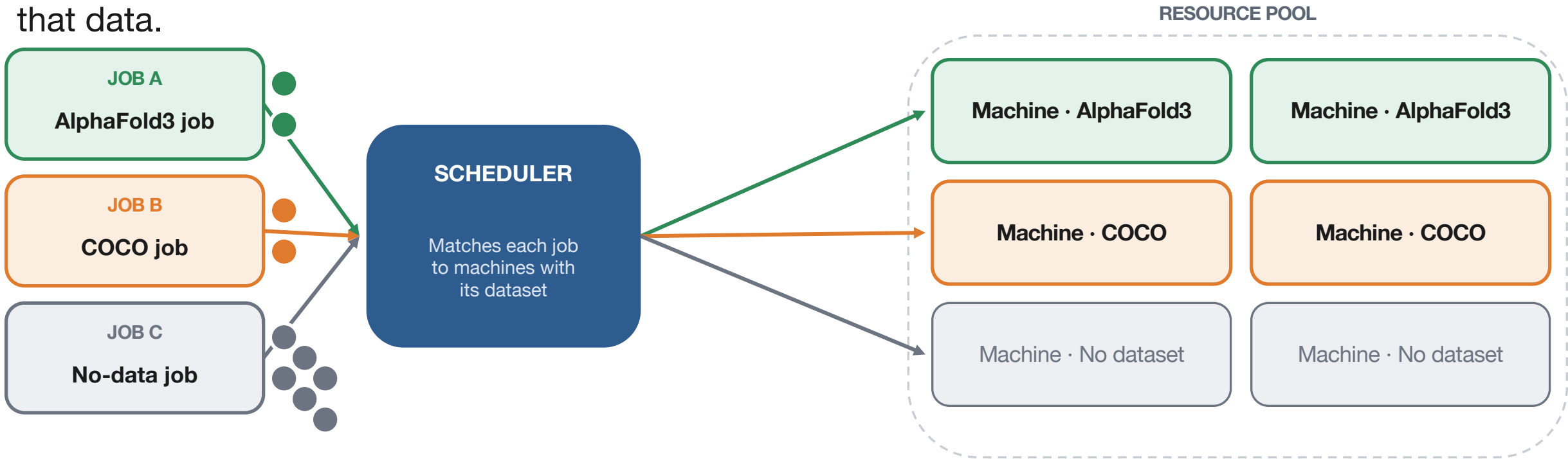


Data locality becomes a matchmaking property — HTCondor sends work to where the data already lives.

Data-as-a-Resource

Generalizing beyond AlphaFold

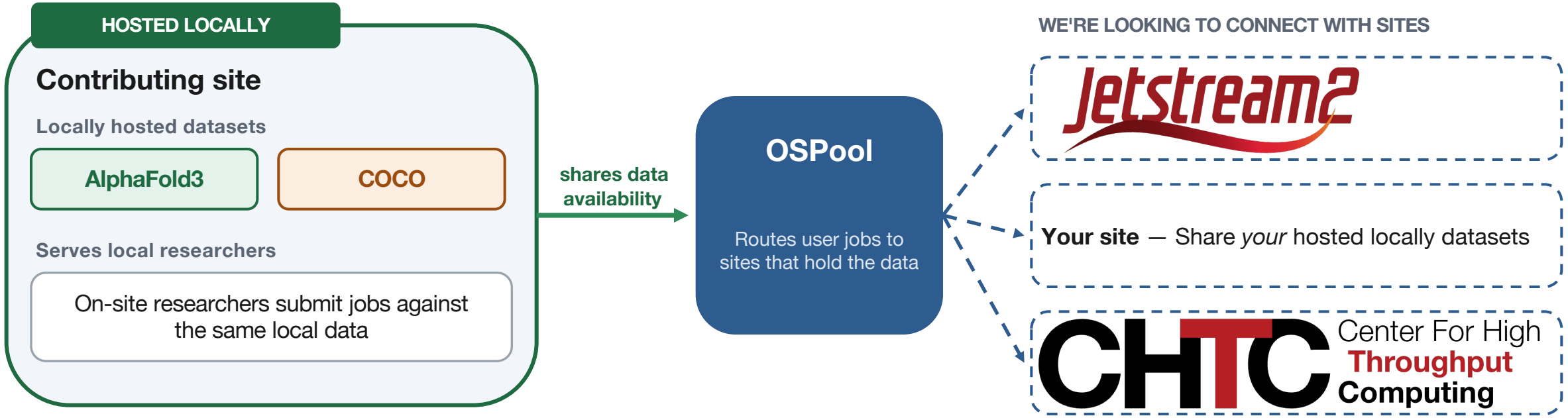
Each job declares the dataset it needs. The scheduler matches it to machines that already hold that data.



One pool may have many datasets: AlphaFold3 jobs land on AF3 machines, COCO jobs on COCO machines, and data-agnostic jobs run anywhere.

Sharing datasets with OSPool users

Proven at one site today — we want to partner with *you* to share *your* locally hosted datasets with OSPool jobs.



Interested in contributing? If your site already hosts reference datasets, sharing them with OSPool turns local data into shared capacity. Reach out to a Research Computing Facilitator!

CHTC HTCondor Software Suite PATh

Benchmarking the support envelope

Benchmarking turned scattered troubleshooting into a map of what we could actually support.

Easy to support

- Standard single-protein predictions
- Modest sequence sizes
- Fits comfortably in GPU memory
- Predictable runtime

Supportable with guidance

- Larger complexes
- Deeper alignments
- Needs job-shape tuning
- Some hand-holding on resources

Edge of support

- Very large multimers
- GPU-memory limited
- Long / variable runtime
- Outcome not guaranteed

CHTC HTCondor Software Suite PATh

New allies: the community shows up

75+

attendees at the one of the largest CHTC training to date — with people standing along the walls and sitting on the floor.

Demand for these workflows was real — and bigger than we expected.

CHTC HTCondor Software Suite PATh

The community changes the question

Users didn't want one-off predictions. They wanted to ask bigger questions.

“Can we screen many molecule–molecule interactions?”

High-throughput screens
Hundreds to thousands of predictions, not one

Molecule–molecule interaction studies
Systematic exploration of binding and complexes

Reusable workflows
Pipelines they can run again and share

The question shifted from “can it run?” to “how can we do this at scale?” — and that reshaped the facilitation model.

CHTC HTCondor Software Suite PATh

The adventure continues: high throughput protein-protein interaction screenings

We want to run **180K pairwise protein-protein interaction** screenings using AlphaFold3.

Kavi Mehta – UW Madison

CHTC HTCondor Software Suite PATh

The adventure continues: high throughput protein-protein interaction screenings

Now that's some real high throughput AlphaFold! Let's give it a try!

Kavi Mehta – UW Madison

CHTC HTCondor Software Suite PATh

Trial 3: understanding repeated preprocessing

The same expensive alignments get regenerated over and over — especially as users scale up.

One-off prediction

Generate alignment → run inference once.
The cost is paid a single time.

compute ✓ once

then reused, not repeated

users scale up

Scaling to many predictions

The same alignments are regenerated for every related run — the expensive CPU/database step repeats needlessly.

recompute

paid again every job

If preprocessing is reusable, it should be computed once — not regenerated by every job.

CHTC HTCondor Software Suite PATh

The AlphaFold3 cached alignment library

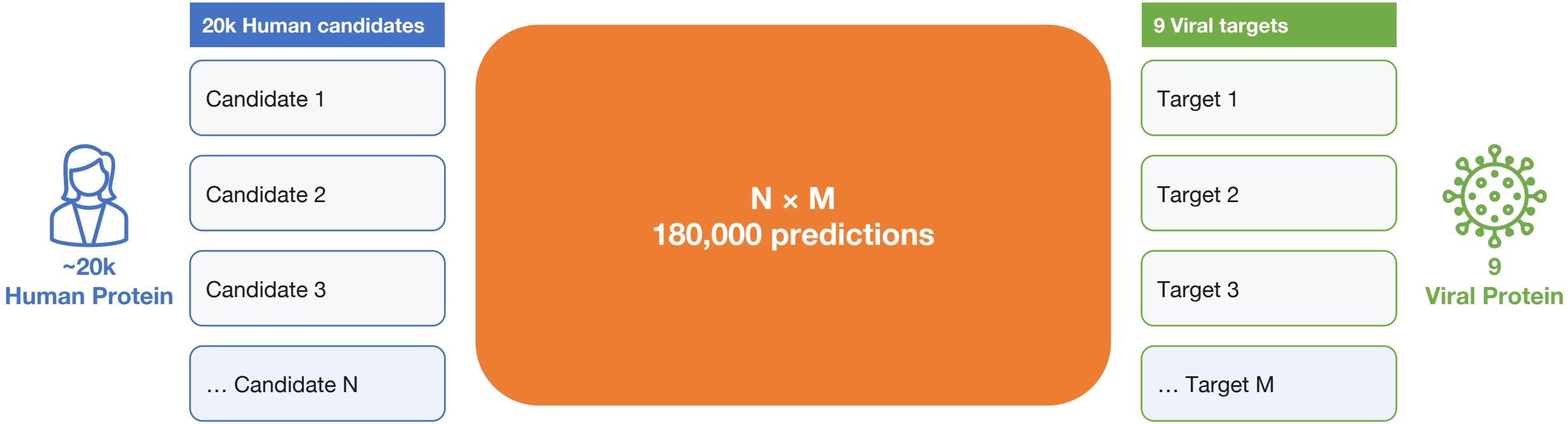
Compute each alignment once, store it, and let every later prediction reuse it.

The intermediate product becomes a reusable resource — not disposable work.

Use case: many-to-many screening

Mehta Lab

Screening every candidate against every target means N × M predictions — the combinatorics, not the model, become the challenge.



↻ The cached alignment library computes each sequence's alignment once — and reuses it across every pairing.

CHTC HTCondor Software Suite PATh

The library collapses the compute

Mehta Lab

Same per-job profile — ~4 cores, ~60 minutes each. Only the job count changes.

WITHOUT THE LIBRARY

180,000

alignment jobs (N × M = 20,000 × 9)

≈ 720,000 core-hours
4 cores × 60 min × 180,000 jobs

WITH THE LIBRARY

20,009

alignment jobs (N + M = 20,000 + 9)

≈ 80,000 core-hours alignments
≈ 15,000 core-hours post-alignment processing
each alignment computed once, then reused

≈ 160,000 fewer alignment jobs · ~87% less alignment compute and wall time

CHTC HTCondor Software Suite PATh

Making the previously impossible routine

When the combinatorics stop being the limit, researchers can ask questions that were simply off the table before.

ENABLING RESEARCH

Hannan Shabaan — Molecular mechanics of cnidarian–algae endosymbiosis

1,000,000,000 → **65,000**

possible protein–protein pairings
≈ 25,000 anemone × ≈ 40,000 symbiont proteins

alignments with the library
(N + M, computed once and reused)

A screening that no cluster could support becomes a screen a single graduate student can run.

CHTC HTCondor Software Suite PATh

The AlphaFold3 cached alignment library

Compute each alignment once, store it, and reuse it across every prediction that needs it.

AS OF JUNE 8, 2026

410K alignments cached

342K
unique protein sequences

10K
source organisms

29K
community-contributed records

N + M
alignments instead of N × M

~89%
less alignment compute & wall time

once
computed per sequence, then reused

CHTC HTCondor Software Suite PATh

Current status and early impacts

Where our AlphaFold3 support stands today.

526,776

Jobs ran on CHTC and OSPool by 70 users

75+

attendees at the flagship training

Live

Pre-staged dataset + alignment caching in use

Next

extending the access on the OSPool

Early, but real — and already reusable beyond the first researcher.

CHTC HTCondor Software Suite PATh

Supporting AlphaFold3 required facilitation patterns that translate

The work evolved from helping individual jobs run into building service patterns the whole community can reuse.

Workflow-aware facilitation

Support the whole pipeline shape, not just the GPU step.

Data as a resource

Treat large reference datasets as shared infrastructure, not per-job baggage.

Caching & Reusing Artifacts

Compute expensive intermediates once; reuse them across jobs and users.

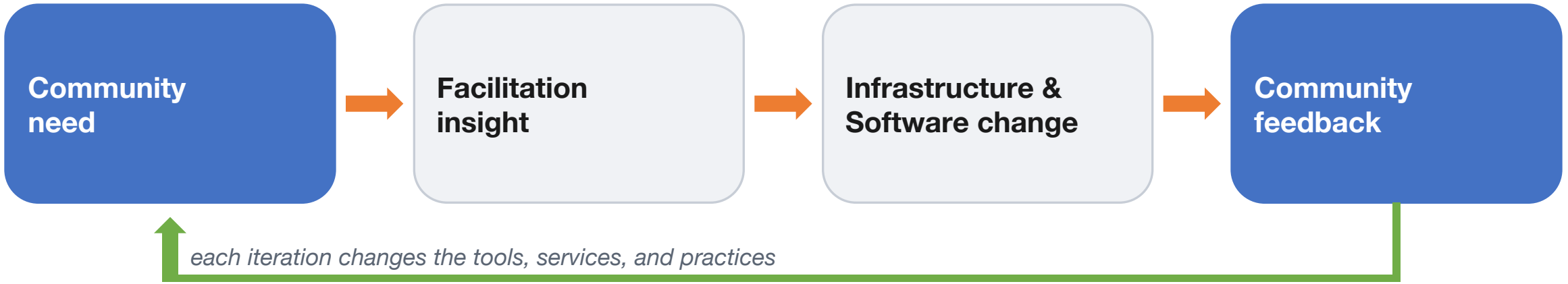
Meaningful docs & training

Turn one-off troubleshooting into durable guidance and examples.

CHTC HTCondor Software Suite PATh

Translational computer science

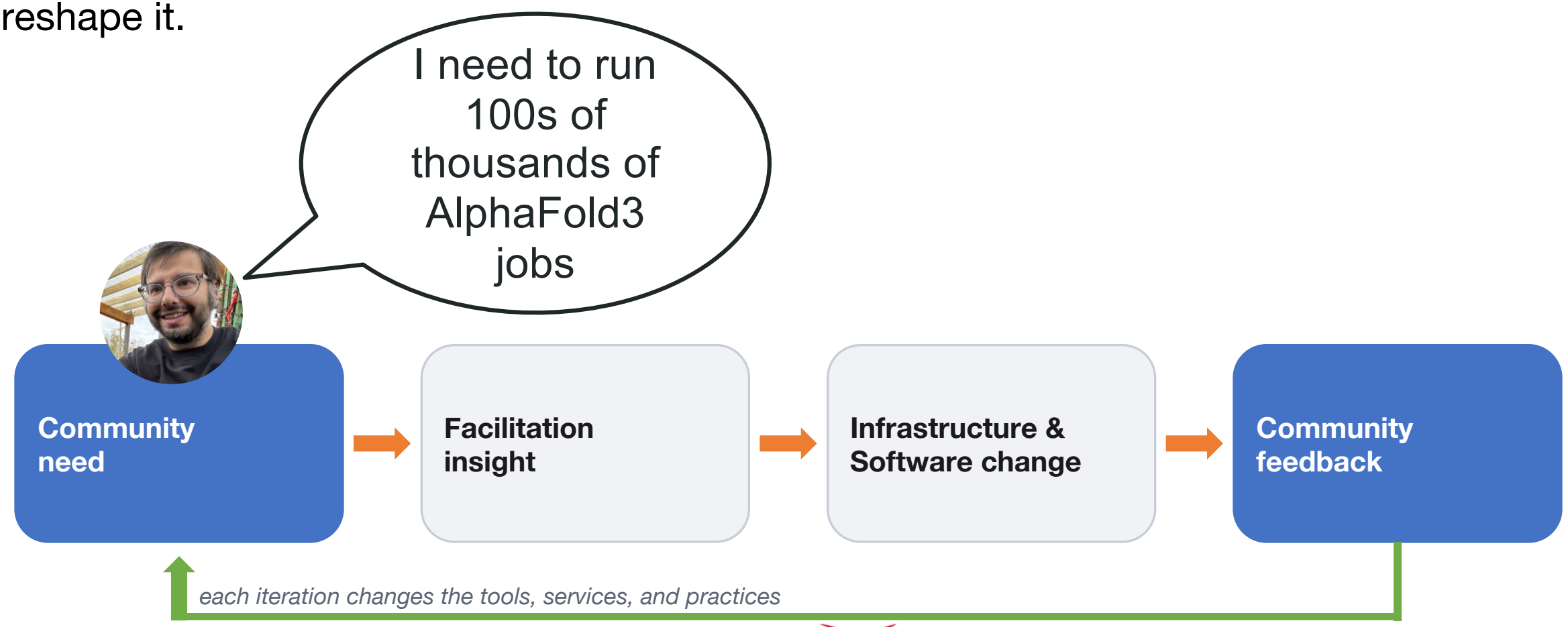
Turning researcher needs into usable infrastructure — then letting the community reshape it.



CHTC HTCondor Software Suite PATh

Translational computer science

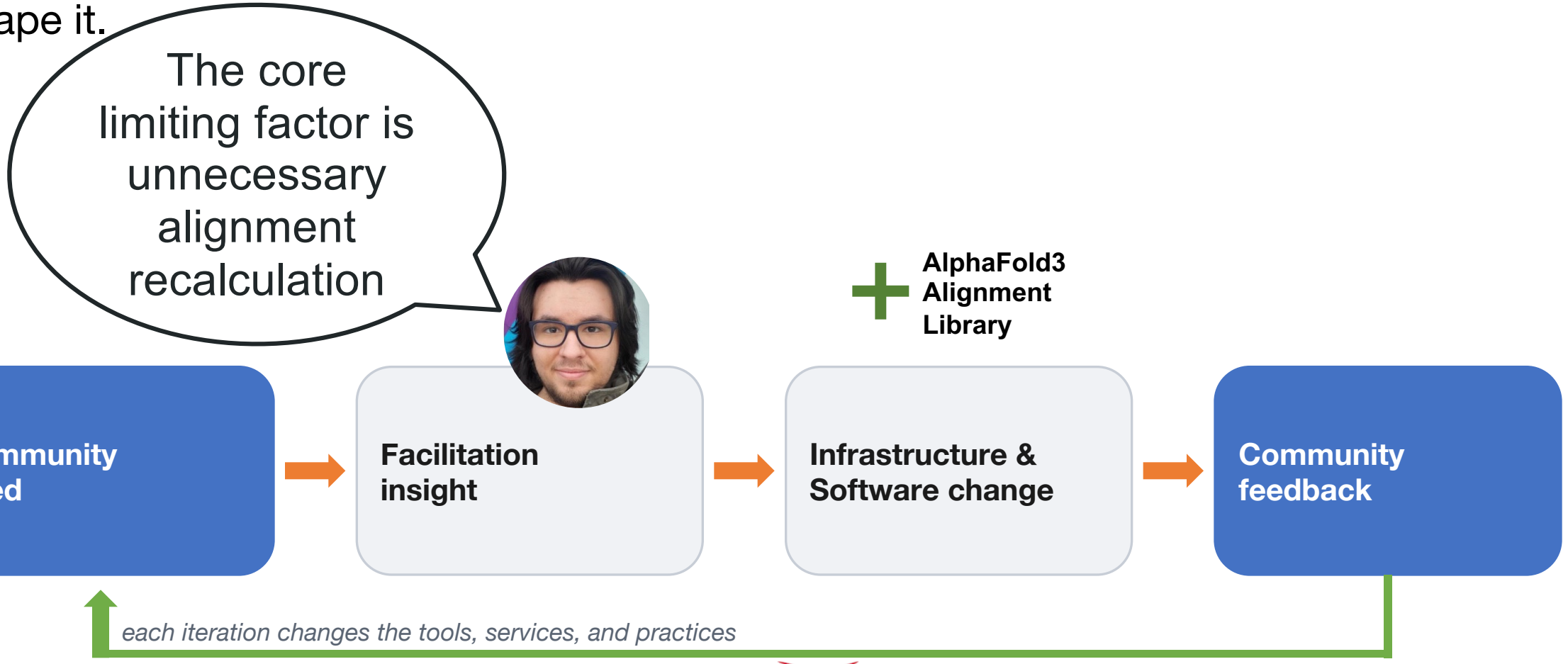
Turning researcher needs into usable infrastructure — then letting the community reshape it.



CHTC HTCondor Software Suite PATh

Translational computer science

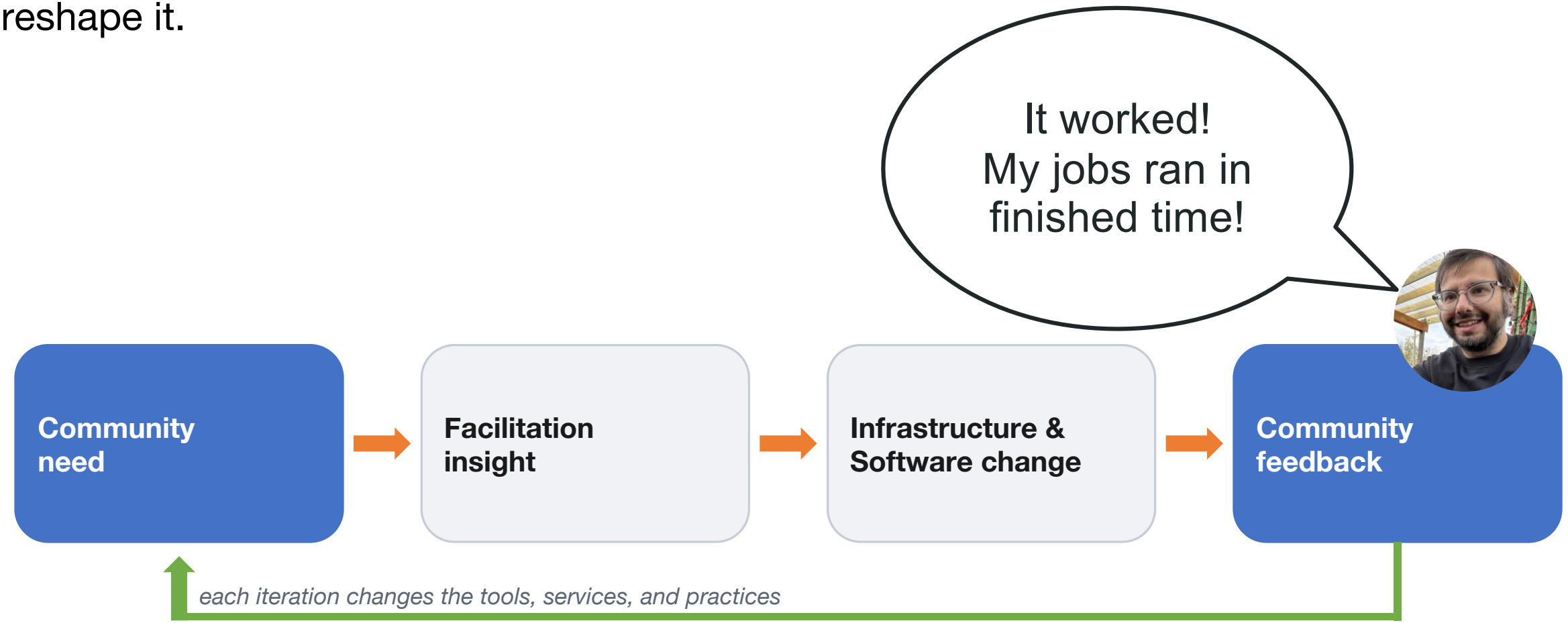
Turning researcher needs into usable infrastructure — then letting the community reshape it.



CHTC HTCondor Software Suite PATh

Translational computer science

Turning researcher needs into usable infrastructure — then letting the community reshape it.



CHTC HTCondor Software Suite PATh

From AlphaFold3 to AI facilitation

Supporting the next generation of AI-enabled research will require translational computer science: continuous feedback between researchers, facilitation, software, data, and infrastructure professionals.

community need → *facilitation insight* → *infrastructure change* → *community feedback*

CHTC HTCondor Software Suite PATh

Acknowledgements

With thanks to the CHTC and PATh teams, the OSG Consortium, our collaborating researchers, and the AlphaFold3 user community — whose questions and feedback shaped this work at every step.

This material is based upon work supported by the National Science Foundation under Grant No. 2030508. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

CHTC HTCondor Software Suite PATh