

Taking the Seatbelts Out of the Supercomputer

Glenn K. Lockwood, Ph.D.

Principal Technical Strategist

VAST Data

Cutting-edge engineering designed with humans in mind

Many of us flew here on a marvel of modern technology like this



source: readyfortakeoffbook.com

But humans force frivolities on technology

Putting 120 holes in a pressure vessel is technically not good



source: readyfortakeoffbook.com


source: aircharter.net

What happens when “the next user never sleeps?”

What frivolities exist in supercomputers? And what can be removed?

Compute Nodes	Parallel File System	Login Nodes	Management Nodes
Batch Scheduler	UNIX Accounts	SSH / Terminal	Job Accounting
Data Services	Observability	User Software Environment	User Documentation

A lot of junk is only around for human convenience

Human interfaces and software curated for humans can be thrown away

Compute Nodes	Parallel File System	Login Nodes	Management Nodes
Batch Scheduler	UNIX Accounts	SSH / Terminal	Job Accounting
Data Services	Observability	User Software Environment	User Documentation

Other components become critically important

Agents actually read documentation and write down what they're doing (imagine that)

Compute Nodes	Parallel File System	Login Nodes	Management Nodes
Batch Scheduler	UNIX Accounts	SSH / Terminal	Job Accounting
Data Services	Observability	User Software Environment	User Documentation

Some components are essential, but not in their current form

Attractive nuisances - convenient for humans, but problematic otherwise

Compute Nodes	Parallel File System	Login Nodes	Management Nodes
Batch Scheduler	UNIX Accounts	SSH / Terminal	Job Accounting
Data Services	Observability	User Software Environment	User Documentation

UID=0 is all you need

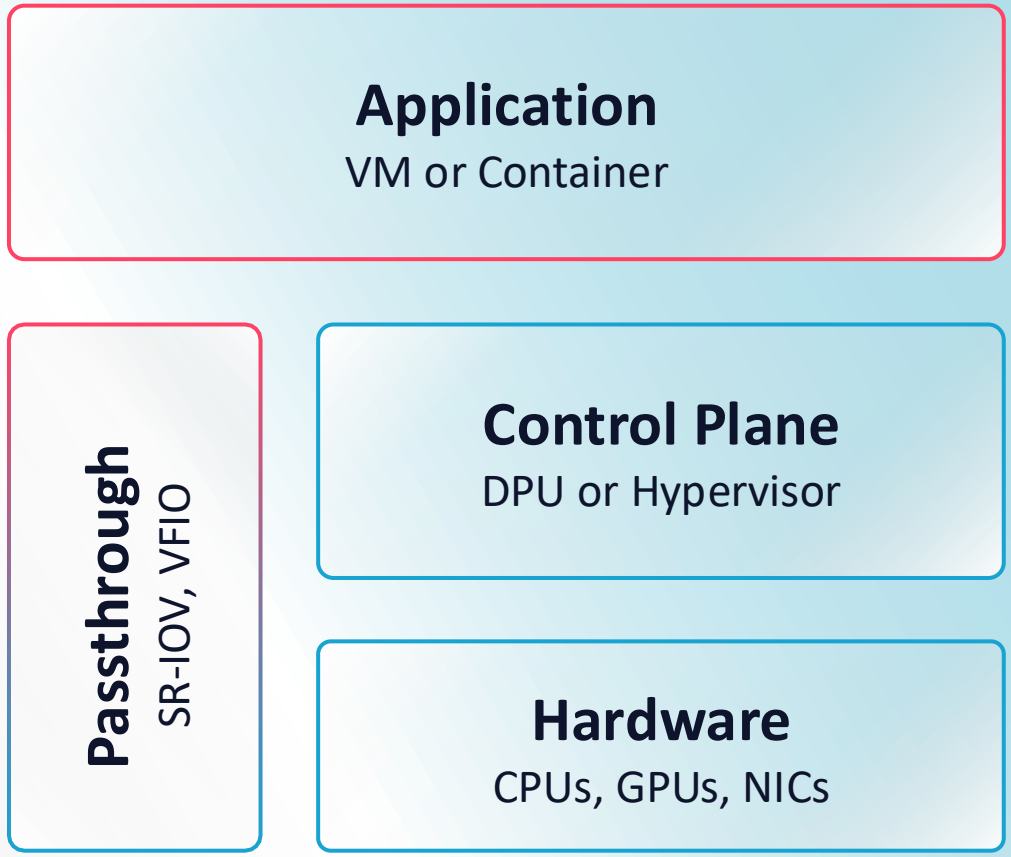
Otherwise, who are you trusting?



UID=0 is all you need

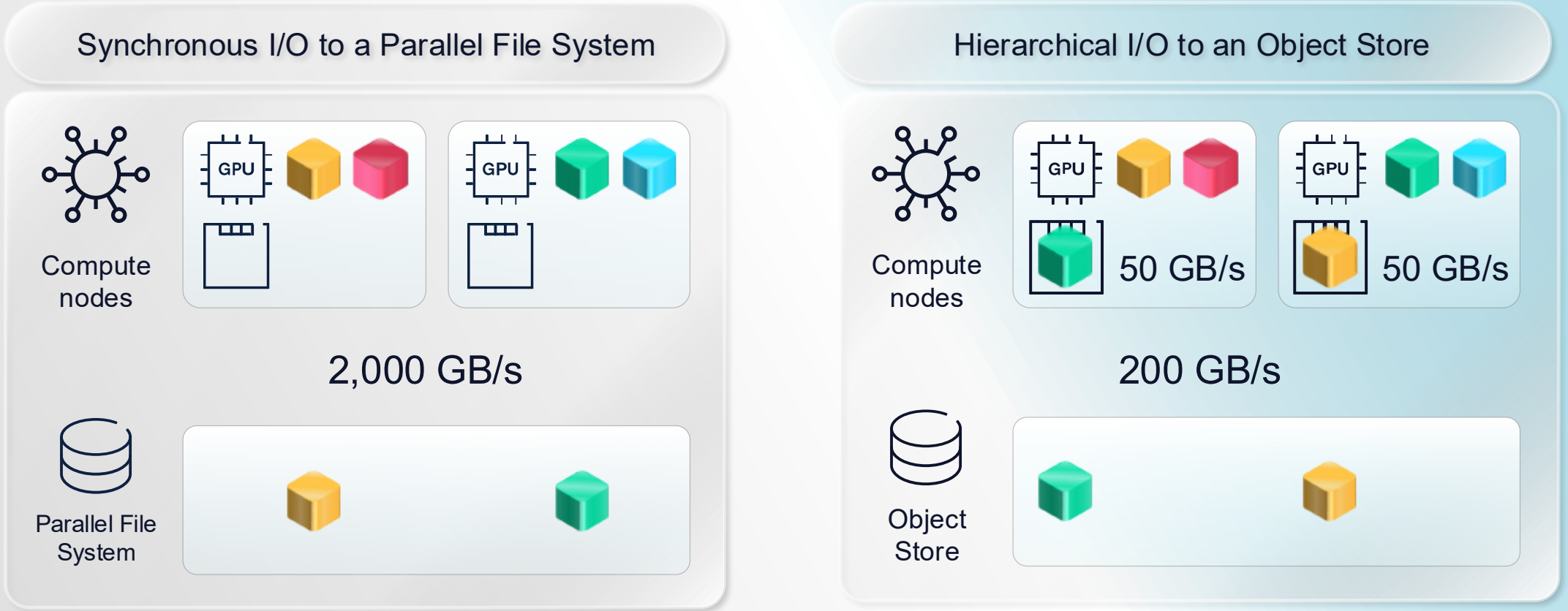
Otherwise, who are you trusting?

- Separate trust domains of **users** and **infrastructure**
- Actions, not actors, carry credentials
- Human users can:
  - **delegate** authority
  - **revoke** authority
- Admins can **revoke** authority

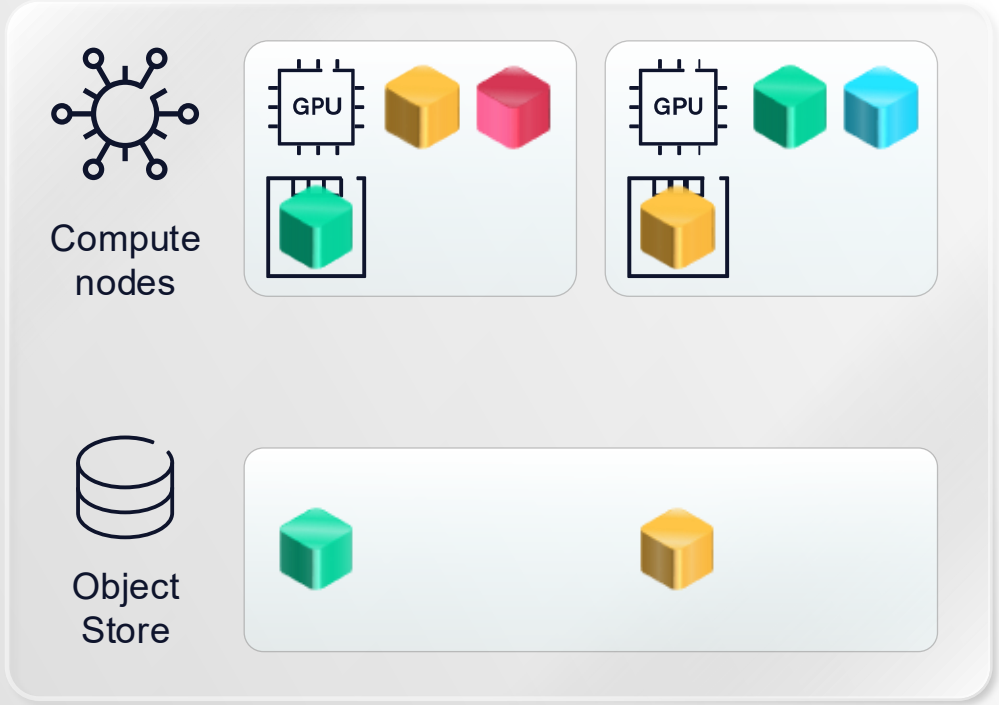


But what about file access?

File systems are easy for humans, but bad for machines: scalability, security, stability, …



Hierarchical file/objects: secure, resilient, cost-effective



No implicit trust of UIDs

No evictions crashing jobs

8× less power per PB*

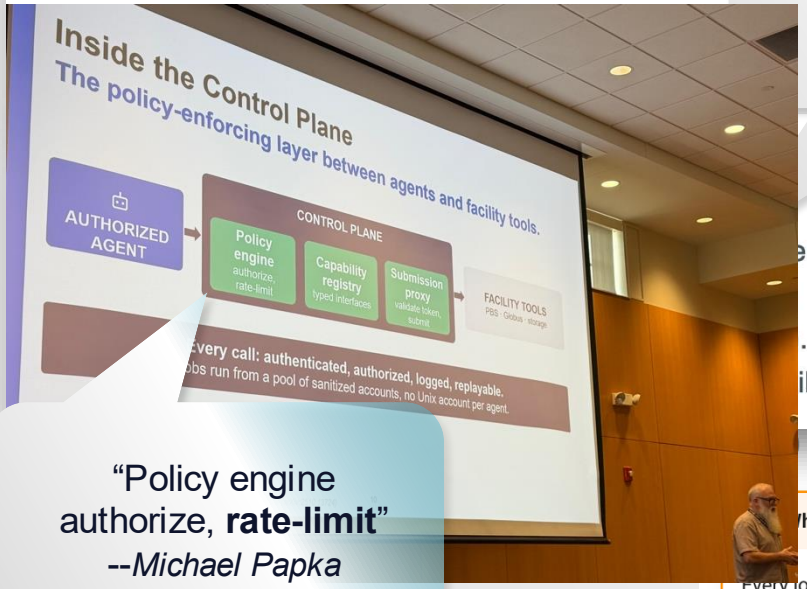
Incompatible with batch

VAST

* 40 PB VAST cluster designed for 200 GB/s write bandwidth vs. 40 PB VAST cluster designed for 2,000 GB/s write bandwidth

Scale is incompatible with batch

Humans are slow, and batch was designed for humans.



"It can accept **1,000 job submissions per second** and fully execute 500 simple jobs per second"
--Slurm documentation

"Users submitting **large numbers (> 100s) of jobs using loops will be contacted** and asked to adjust their workflows."
--UArizona HPC Documentation

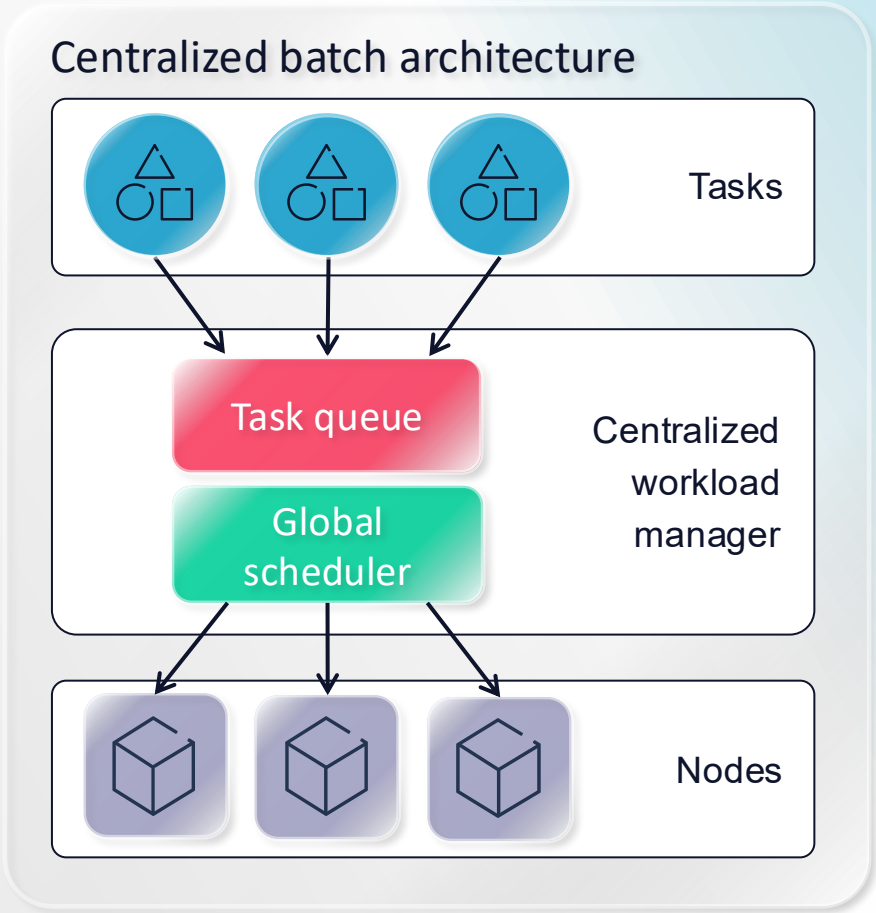
"If you plan to submit many small jobs in an array that require the allocation of **more than 10 jobs per minute, please use GNU parallel**"
--ULHPC Technical Documentation

"... whenever you are tempted to write a **shell loop around sbatch ... do not do it** - instead, use Slurm's job array feature."
--HPC Wiki

"Policy engine authorize, **rate-limit**"
--Michael Papka

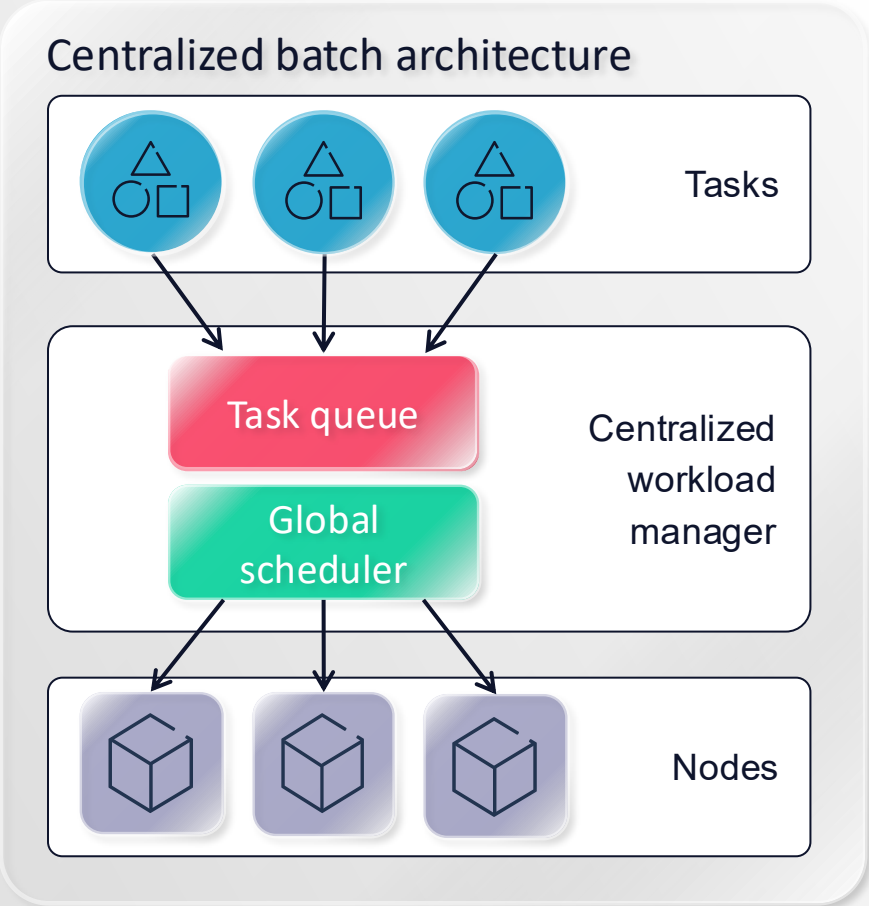
Centralized job orchestration is fundamentally not scalable

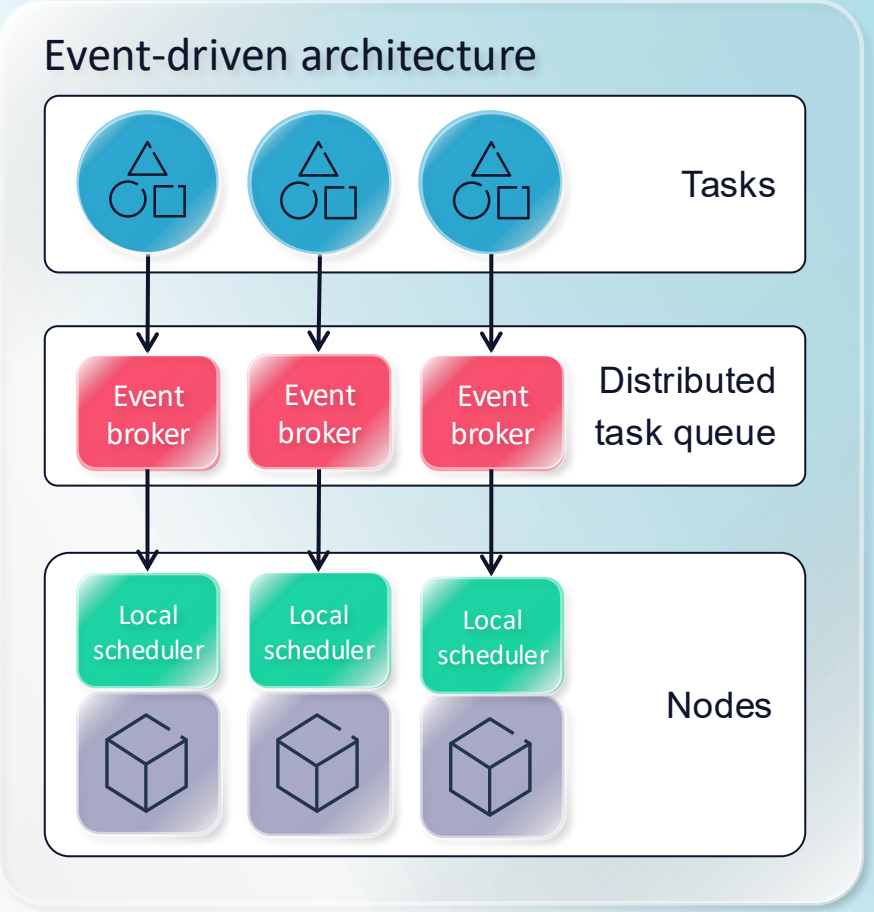
Scheduling is NP-hard and blocks job ingestion and dispatch



Event-based orchestration is fundamentally scalable

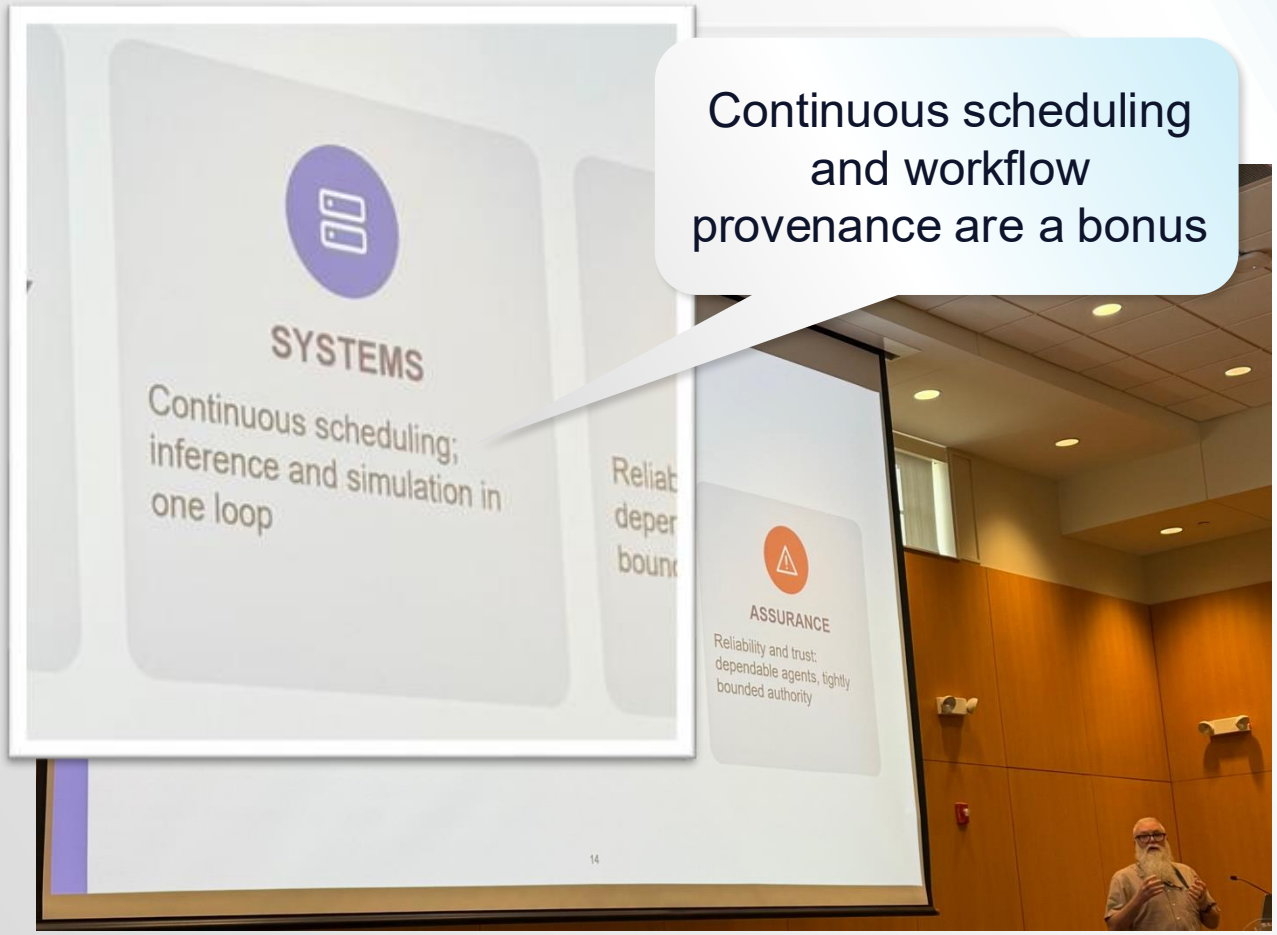
Nodes pull work from a durable queue, separating job ingest from job dispatch



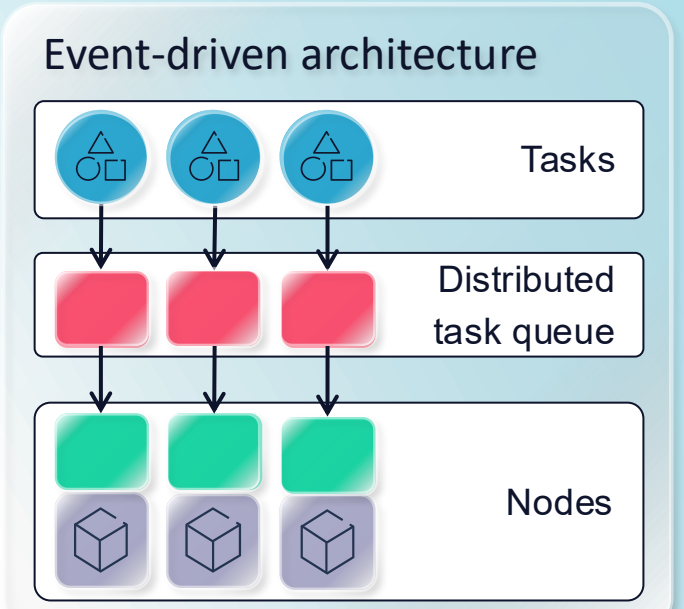


Event-based orchestration scales with agents

Added benefit of being triggerable by any event - human, agent, or environmental



Continuous scheduling and workflow provenance are a bonus



>>> 1,000 per server*
jobs per second

VAST

* Preliminary testing on early release of VAST DataEngine v5.5 resulted in > 4,000 jobs/server/sec. The real capability on production systems is higher, but VAST's diligent architects did not want me to advertise a hard number that was measured on old hardware and pre-production code.

Good news: these systems already exist for commercial AI

Though largely untested in scientific computing environments

Authentication & Authorization	SPIFFE/SPIRE and OPA VAST Security Token Service
Parallel Storage System	Scalable object stores (like VAST) AI frameworks w/ async checkpoint/restart/data loading
Job Orchestrator	Kafka, Knative VAST DataEngine

Will agents replace humans on supercomputers?

No

But agents force us to critically examine the garbage we carry forward.

A supercomputer designed for agents can be made human-compatible,
but the reverse is not true.