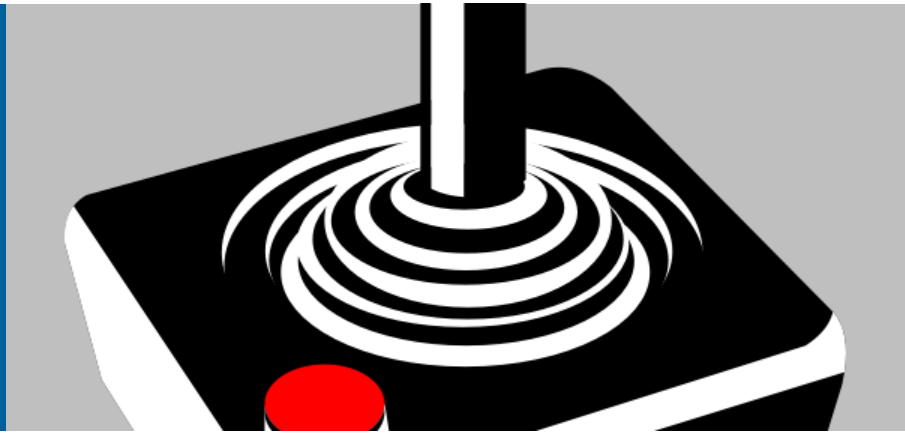
Argonne NATIONAL LABORATORY 75 1946–2021

CAN WE GAMIFY I/O PERFORMANCE?

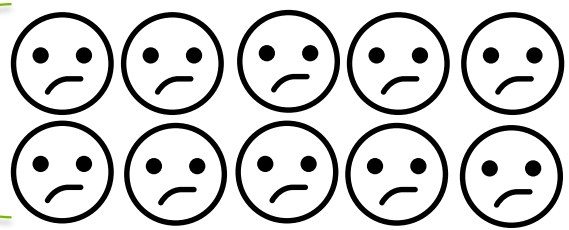


PHIL CARNS
Mathematics and Computer Science Division
Argonne National Laboratory

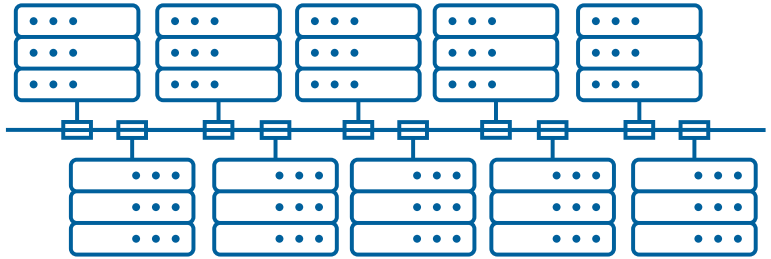
Monday August 16

HUMAN EXPERTISE ISN'T VERY SCALABLE

Scientists, wishing they could process data more quickly



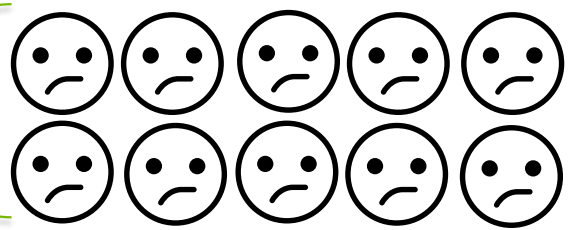
. . .



Facility experts, helping scientists optimize their codes

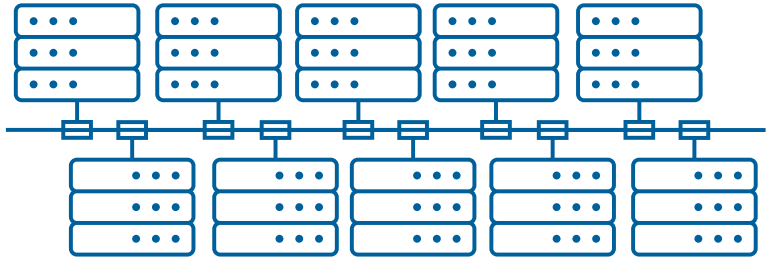
HUMAN EXPERTISE ISN'T VERY SCALABLE

Scientists, wishing they could process data more quickly



. . .

This group naturally scales as more users, applications, and scientific domains embrace the use of HPC.

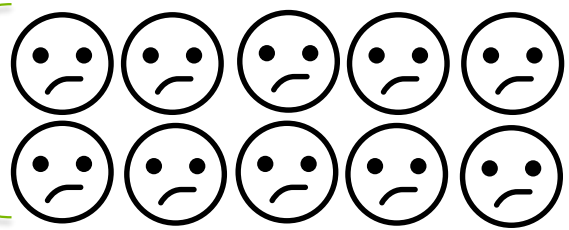


Facility experts, helping scientists optimize their codes

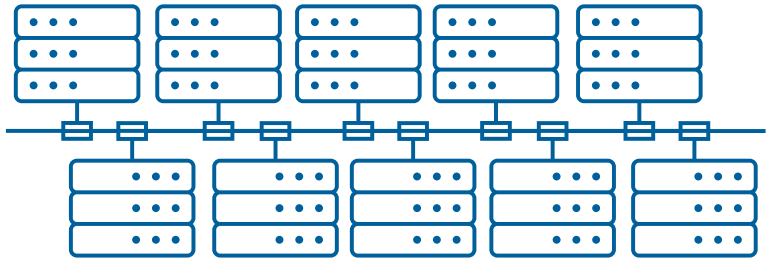
This group does not: I/O experts are difficult to cultivate, hire, and train.

HUMAN EXPERTISE ISN'T VERY SCALABLE

Scientists, wishing they could process data more quickly



. . .



Facility experts, helping scientists optimize their codes

How can we better leverage the more scalable resource here?

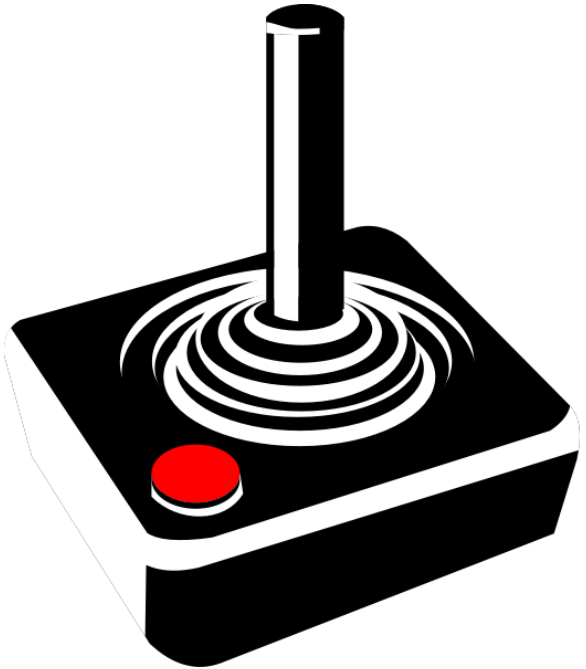
There are some problems:

- Scientists have a lot of demands on their time.
- The I/O tuning process and its payoff are murky.
- Training, outreach, and documentation are only getting us so far.

COULD WE GAMIFY I/O PERFORMANCE?

Engage and empower the users to help themselves more

- Who are your competitors?
- How do you score?
- What are the ground rules?
- What's the reward?



PROBLEM 1: WHO ARE YOUR COMPETITORS?

Let’s look at how the Olympics did it with 11,656 athletes

Ok! Line up 11,656 people and see who can swim 100M the fastest!

PROBLEM 1: WHO ARE YOUR COMPETITORS?

Let's look at how the Olympics did it with 11,656 athletes

Ok! Line up 11,656 people and see who can swim 100M the fastest!

Oh, wait, maybe who can do the coolest flips?

Climb weird stuff without falling?

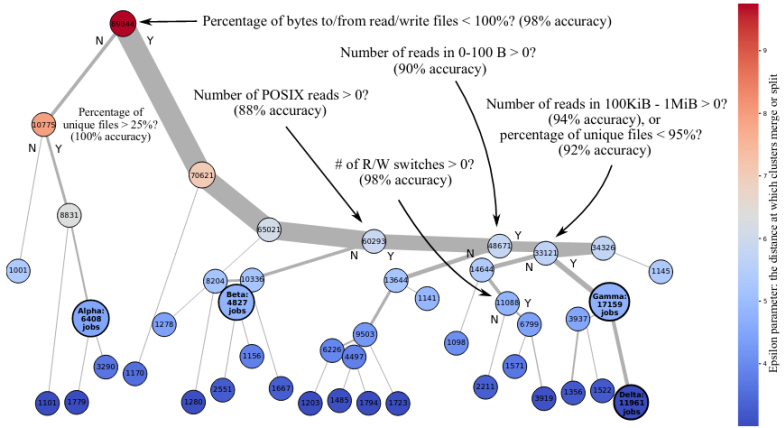
Throw all competitors to the ground?

HPC users are similarly diverse. It is not as simple as "who gets the most write bandwidth?".

Users operate at varying scales with diverse data, access patterns, and science objectives. There is no single universal metric or goal for all users.

PROBLEM 1: WHO ARE YOUR COMPETITORS?

ML? Clustering? Historical context? Yourself?



Isakov et al. “HPC I/O throughput bottleneck analysis with explainable local models” in SC 2020.

Finding relevant HPC I/O competitors is hard.

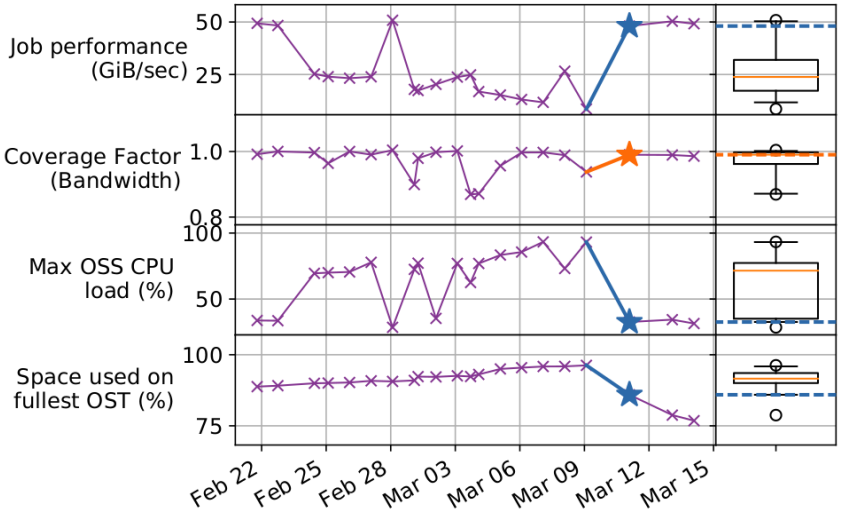
Cluster similar workloads to compare against?

Contemporaries or history?

What scale? What resources?

PROBLEM 2: HOW DO YOU SCORE?

What's the metric, and what's a reasonable goal?



Glenn K Lockwood et al. "UMAMI: a recipe for generating meaningful metrics through holistic I/O performance analysis" in PDSW 2017.

Facility documentation quotes the "stunt mode" performance for a system. This is not helpful to end users. They don't have ideal workloads, brand new empty file systems, or dedicated system time.

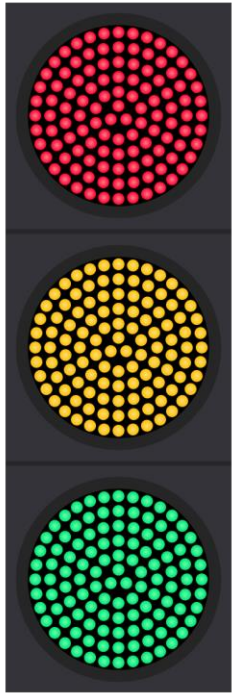
Is bandwidth even the right metric? Maybe latency? Maybe app convergence time?

We need to compare useful metrics against relevant competitors.

Oops: often the system is slow for reasons beyond any user's control. (Sorry Olympic swimmer: there seems to be a shark in the pool today and we don't know why…)

PROBLEM 3: WHAT ARE THE GROUND RULES?

What can you do to improve your score?



There are thousands of publications that report I/O tuning strategies. Pause your science work for a few years and read them? Perhaps a dozen of them will be relevant to you. Good luck!

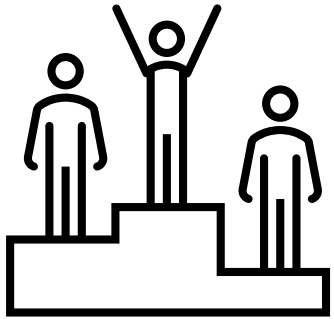
What if we could identify concise, salient features that are likely to give users the best “bang for the buck”?

- “The workload for this file would perform better on facility filesystem /mnt/fast-stuff”
- “This file is not striped well; set hint “abracadabra””
- “Writes are interleaved and unaligned; try a collective write.”

There are publications and studies about these kinds of things, but we as a community haven’t distilled it and transferred it well to production.

PROBLEM 4: WHAT'S THE REWARD?

Why would a user bother working on this?



Users can't easily tell how much there is to gain, how much work it will take to get there, or what the payoff will be.

Can we automatically quantify some of this? "Simulation A may have run 5 minutes faster and use 500 fewer node hours if…"

Bragging rights? Imagine this proposal text: "We deserve a large allocation in part because we will make efficient use of the time: we are the #1 ranked large scale sample analysis application at FOO in terms of I/O efficiency"?

There is danger here: if you *ask* users to game the system, then … they will game the system. This will not always yield a truly productive outcome. We've gone through this with job scheduling policies for decades.

… AND THE SOLUTION IS …

(LET ME KNOW AT THE END OF THE WEEK, PLEASE)

THANK YOU FOR YOUR TIME!