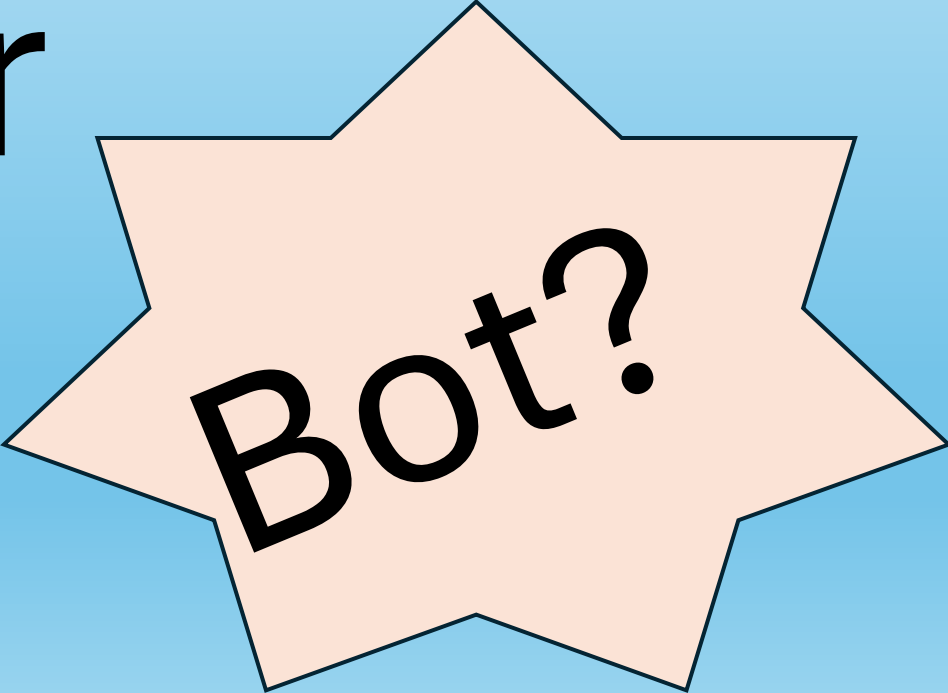




Todd

or



Bot?



**Chat GPT**

**.vs. Todd**



# Rules of the Game

- We pose an HTC-centric question
- With TWO answers:
  - One from from Todd
  - One from Chat-GPT

Your job is to pick which is most correct.

Always a third choice – when asked

**yell out an even better answer**

After each question YOU vote

# Warmup Question

Why is the data management system called "Pelican"?

# Answer "A"

A pelican is a bird known for carrying and transporting things in its large bill pouch, which loosely matches Pelican's job of moving and making data available across distributed systems.

# Answer "B"

CHTC has a long history of picking project and technology names that are bird-related. Think Condor, Flocking, Chirp, Nest, Rooster, Kingfisher, BirdBath, HawkEye. Pelican was chosen because it has a large beak for stuffing in a lot of data before flying it someplace else

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

# Question 1

What is one thing that a user new to High Throughput Computing should know?

Answer "A"

HTC rewards thinking about throughput (how much work gets done over time) rather than latency (how quickly one job finishes)

# Answer "B"

You probably do not need HPC. If one work-step can fit on a single node, forget HPC, and go with HTC. This way you will focus on science instead of spending your career tweaking MPI code



Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

## Question 2

What's the best way to schedule very long running jobs?

# Answer "A"

If a job runs for weeks, treat it as a workflow of recoverable stages rather than a single process that must survive for weeks uninterrupted.

# Answer "B"

Long-running jobs will ideally be self-checkpointing, and HTCondor has support for user-level checkpoints.

Encourage use of tools that support checkpointing out-of-the-box, such as PyTorch Lightning instead of regular PyTorch. Failing that, set aside a subset of pool capacity for long running jobs and backfill that capacity with short or preemptable jobs

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

# Question 3

Another new Linux critical security CVE has just been announced. What should pool admins do?

# Answer "A"

If you're asking specifically about an HTCondor/Pelican-style pool, I can walk through a practical response plan (drain commands, maintenance scheduling, and prioritization of submit vs. execute nodes).

# Answer "B"

Get used to it.

This is likely to be the new normal for a year or so, until AI tools have successfully found most of the bugs that have been lurking for years undetected. Make a plan in advance for what to do when a zero-day exploit is announced, run fire drills on the plan, decide in advance who can make decisions to reboot everything or shut it all down at a moment's notice

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

## Question 4

What's the best way to change contents of Pelican objects?

# Answer "A"

Pelican assumes objects are immutable, because caches may already hold copies of the object.

If you change the bytes behind an existing object name, caches can serve stale data or even a mixture of old and new content.

# Answer "B"

You cannot change the contents; they are immutable so they can be safely cached. Suggest you use an object name that contains a SHA checksum of the contents, or some other way of indicating a version in the object name

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

# Question 5

How to debug overloaded AP?

# Answer "A"

Check whether the AP is CPU, memory, I/O, or HTCondor-schedd-bound using uptime, top/htop, free, iostat, and HTCondor logs like SchedLog and CollectorLog to see if you're hitting resource limits or experiencing daemon latency.

# Answer "B"

Run “htcondor ap status” to confirm it is really the AP that is overloaded. If yes, likely culprits are (a) many thousands of very short (second-long) jobs, (b) not using a locally attached SSD to store the AP database, or (c) overloaded shared filesystem.

Use “condor\_sos” to inspect the queue when schedd is unresponsive. Worst case increase logging with `SCHEDD_DEBUG=D_FULLDEBUG`, email htcondor-users for help with output from tool “condor top”

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

# Question 6

How to show stakeholders value of pool?

# Answer "A"

Show value by translating pool activity into outcomes stakeholders care about: total jobs completed, CPU/core-hours delivered, throughput over time, queue wait times, and “science enabled” metrics (papers, experiments, or deadlines met), ideally broken down by research group or project.

# Answer "B"

Besides just collecting utilization numbers (X jobs, Y users, Z CPU hours), collect testimonials for users on how it impacts their science. If your pool is contributing capacity to the OSPool, look at your institution's CE Dashboard. Consider adding opportunistic capacity to demonstrate additional value

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

# Question 7

If you could change one thing about Pelican by HTC 27, what would it be?

# Answer "A"

Make object versioning and “intentional mutability” first-class, not a convention. Right now, Pelican works best when you treat objects as immutable and encode versioning in names—but that pushes discipline onto users and workflows.

# Answer "B"

Pelican should have safeguards to prevent overloading – either queue requests, load balance across caches, return ‘try later, I am overloaded’, or all of the above

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0

## Question 8 (Final Question!)

If you could change one thing about HTCCondor by HTC 27, what would it be?

# Answer "A"

Make observability and debugging a first-class, unified experience across the whole system (submit, schedd, collector, execute nodes), instead of a set of powerful but fragmented logs and commands.

# Answer "B"

A web portal for users to place and manage jobs. Open OnDemand, or a custom one... let's give newbies an option other than the terminal. But keep the CLI of course, as not everyone is a beginner

Ummm, Actually...

# Scorecard

Todd: 0

Bot: 0

Audience: 0