THOUGHTS FROM TACC ON NEXT GENERATION ACADEMIC SUPERCOMPUTING SYSTEMS

Dan Stanzione
Executive Director, TACC
Associate Vice President for Research, UT-Austin

OU Supercomputing Symposium
September 2022

TACC - 2022

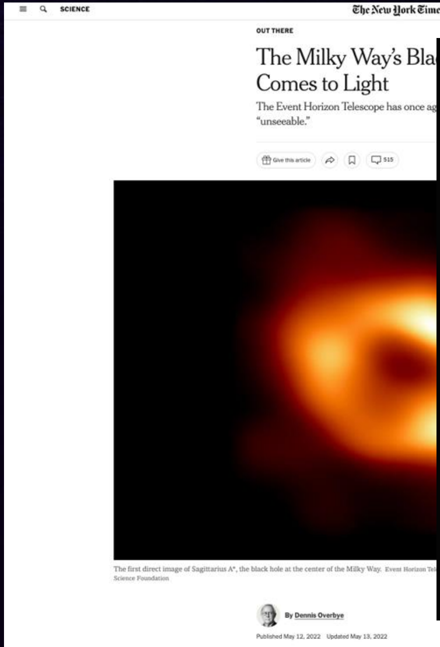
LEADERSHIP-CLASS COMPUTING FACILITY

TACC
TEXAS ADVANCED COMPUTING CENTER

A QUICK TACC REMINDER

- We operate the Frontera, Stampede-2, Jetstream, and Chameleon systems for the National Science Foundation
- Longhorn and Lonestar-6 for our Texas academic and industry users.
- Altogether, ~20k servers, >1M CPU cores, 1k GPUs
- About seven billion core hours over several million jobs per year.

ALL FOR UNCLASSIFIED, OPEN SCIENCE (2022)



SCIENCE The New York Times

OUT THERE

The Milky Way's Bla
Comes to Light

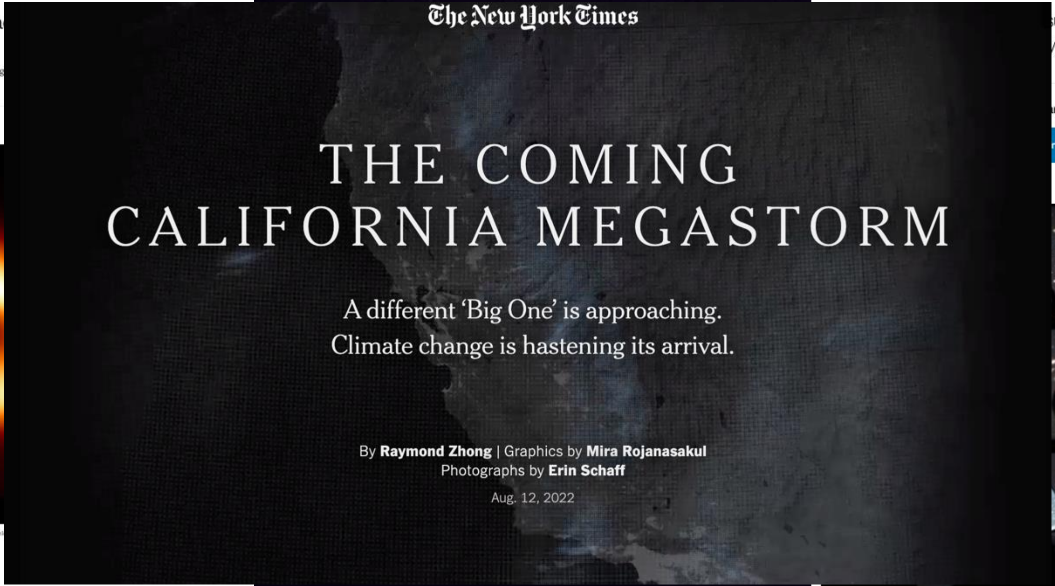
The Event Horizon Telescope has once ag
"unseeable."

Give this article 515

The first direct image of Sagittarius A*, the black hole at the center of the Milky Way. Event Horizon Tel
Science Foundation

By Dennis Overbye

Published May 12, 2022 Updated May 13, 2022



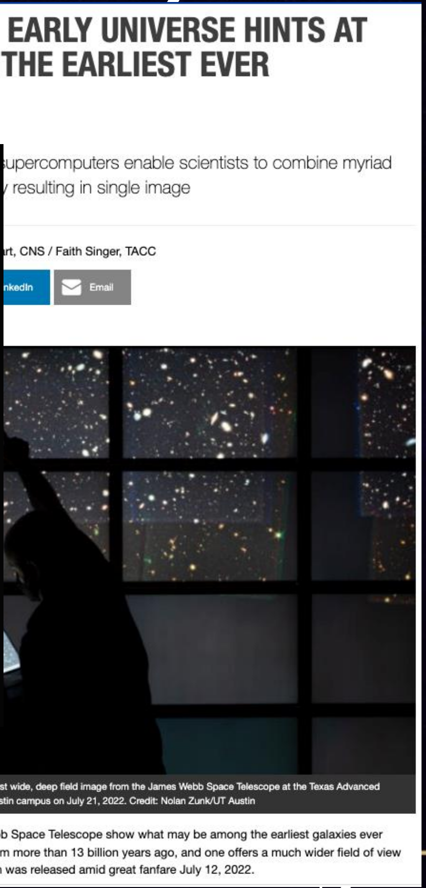
The New York Times

THE COMING CALIFORNIA MEGASTORM

A different 'Big One' is approaching.
Climate change is hastening its arrival.

By Raymond Zhong | Graphics by Mira Rojanasakul
Photographs by Erin Schaff

Aug. 12, 2022



WIDEST VIEW OF EARLY UNIVERSE HINTS AT GALAXY AMONG THE EARLIEST EVER DETECTED

supercomputers enable scientists to combine myriad
resulting in single image

rt, CNS / Faith Singer, TACC

nkedIn Email

Members of the CEERS collaboration explore the first wide, deep field image from the James Webb Space Telescope at the Texas Advanced Computing Center's Visualization Lab on the UT Austin campus on July 21, 2022. Credit: Nolan Zunk/UT Austin

Two new images from NASA's James Webb Space Telescope show what may be among the earliest galaxies ever observed. Both images include objects from more than 13 billion years ago, and one offers a much wider field of view than Webb's First Deep Field image, which was released amid great fanfare July 12, 2022.

NEW IN 2022

- Lonestar-6 Commissioned, our newest "mid" scale system (60,000 cores of AMD Milan, 240 NVIDIA A100 GPUs).
- New-ish stuff:
  - BeeGFS instead of Lustre
  - Rocky instead of CentOS for Base operating system
  - 70KW/rack – oil immersion cooling
  - Some experiments in disaggregation.
  - Virtualized nodes in "small" queue

HEADING FOR THE NEXT SYSTEM

- In 2025 we will replace Frontera.
- Which means we will order in 2024
- Which means we have to get the order approved by the Government starting in 2023.
- 2023 starts in 23 days.
- So… what do we need in the next system???
    - (That's not too crazy or risky…).

KEY QUESTIONS

- Do we need more cycles?
- Do we need bigger machines?
- What kinds of processors do we need?
- What about I/O?
- Disaggregation?
- Software?

DO WE NEED MORE CYCLES

- **Yes --** This is simple, just look at demand. . .

- Not only do we have 3x demand on the new systems (while doing almost nothing to attract more users), we have 3x demand on our *6 year old* hardware as well.
  - AI usage is growing fast.
  - Edge usage/instrument usage is growing fast.
- This part is a no-brainer.

DO WE NEED BIGGER MACHINES?

- **Yes!** Let’s take a look at the load on Frontera.

A FEW METRICS (FRONTERA) FOR YEAR 3

- In the last 12 months:
  - Uptime of 99.2%
  - Average Utilization of 95.4%
  - ~72M SUs delivered
  - 1.13M jobs delivered
  - Zero security incidents.
- Happy to compare uptime, utilization numbers with any modern supercomputer.
  - On the bright side, we are always full. On the downside, no way to squeeze anything else in.

A LITTLE MORE ON USAGE

- **>2,000 jobs were >25,000 cores** – about a **quarter** of all cycles on large jobs.
- >100 jobs at half or full system scale (Consider if all jobs were full scale, and averages 24 hours, we'd only run 365 jobs a year, as opposed to 1.1M jobs).
- Flex jobs, used for backfill, represent 20% of the jobs run (>200k), but represent less than 0.5% of SUs delivered (285k out of 70M).
- Small jobs represent ~30% of jobs, but less than 2% of cycles delivered.
    - So **97% of time goes to jobs >2 nodes**.
    - Average jobs size about **6x that of Stampede2** – this machine *is* used differently.
- We tune the scheduling policy multiple times a year… essentially adjusting to demand.

BY QUEUE

Queue	Job Count (2020)	SUs Charged (2020)	Job Count (2021)	SUs Charged (2021)	Job Count (2022)	Sus Charged (2022)
Normal	556,048	38,577,043	906,114	44,157,946	308,476	50,390,674
Development	47,119	183,901	124,526	621,317	153,635	745,604
Flex	457,392	413,471	609,180	271,791	209,706	285,247
Large	2,106	7,989,616	1,769	14,133,257	1,142	13,018,134
RTX	25,872	591,186	82,392	1,623,327	80,477	1,060,014
RTX_DEV	1,676	3,998	10,944	25,578	13,221	20,921
NVDIMM	905	7,954	9,876	90,779	6,920	115,784
Small	--	--	111,380	407,043	316,953	1,253,289
Debug*	--	--	3,793	3,236,969	2,133	2,496,290
Others	--	--	27,696	78,189	40,513	158,224
TOTAL	1,091,118	48,827,566	1,887,670	64,646,197	1,133,176	69,544,181

Longhorn had another 3.7M SU charged.
*Texascale jobs are largely in Debug Queue

WHAT KINDS OF PROCESSORS SHOULD GO IN THEM?

- OK, here is where it gets interesting.
  - Undeniably true: Much more software runs on CPUs. The programming model is widely known as “programming”.
  - Also true: GPUs are making lots of gains in software/applications, and the fraction of things using them is higher than it was a few years ago.
    - And, they have other advantages. Despite being expensive.

GPU ADVANTAGES

- **Watts/Peak FLOP**
- **Bytes/Peak FLOP**
- Cost/Peak FLOP???
- *What happens when we take away the word "peak"?*

GPU ADVANTAGE – NAÏVE FIRST CUT

	TFlops	Watts	**Gflops/Watt**	BW	**Flops/Byte**
Intel ICX (Dual-Socket)	5.9	540	**10.93**	300	**20**
AMD Milan (Dual-Socket)	5.1	560	**9.11**	300	**17**
AMD MI250x	47.9	560	**85.54**	3277	**15**
NVIDIA A100	9.7	400	**24.25**	1600	**6**
NVIDIA A100 (Tensor)	19.5	400	**48.75**	1600	**12**

In terms of FLOPS/Watt, GPUs clearly win right now!

Even at this level, the GPU cost/TF advantage isn't that clear cut (Assume a node with two A100 cards cost 3x a node with no GPUs).

BUT. . .

- Do all my codes get the same percentage of peak on both architectures?
  - V100 → A100 doubled Flops, performance of NAMD went up 1.5 – efficiency fell.
- How much of those difference is due to compilers/stacks
  - A recent benchmark showed a 19x perf difference between OpenACC and OpenMP offload on the same GPU. And it became 6x when switching compilers.
  - With those kind of differences, who cares what the hardware is?
- *Software* *Matters*
  - And in the academic space, $$$ to rewrite codes are scarce. . . Can we move our users over?
- Today, less than 10% of our workload would work on ORNL Frontier.
  - DOE Software Investment is >$1B for exascale codes – but those largely aren't our codes.

SO, HOW DO WE DECIDE THIS?

- Come up with a suite of applications that broadly represent our workload.
- Evaluate where they are, and evaluate what it takes to change them.
- Pick based on science throughput/$$ based on that workload.

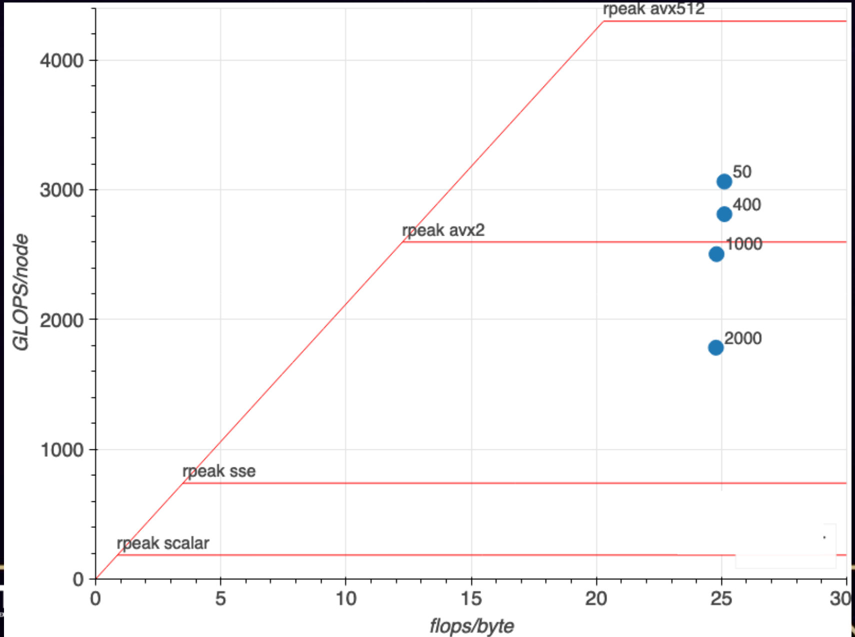
- Turns out none of those things are particularly easy. . .

- We have picked 20 science challenges in our “Characteristic Science Application” program.

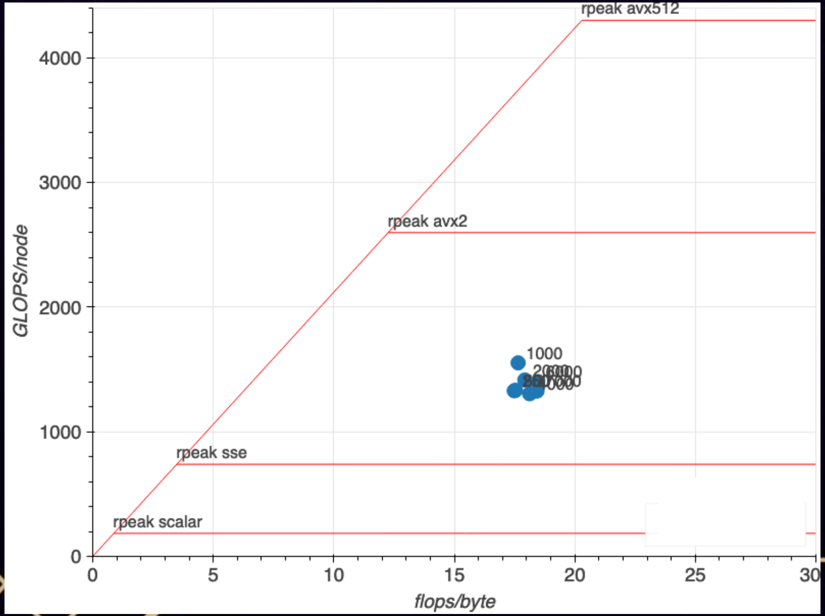
CSA EVALUATION

- Lots of evidence to make us think there is room to improve things.
- Here are some typical roofline plots from the initial eval (code names withheld to protect the guilty).

Good, but poor strong scaling



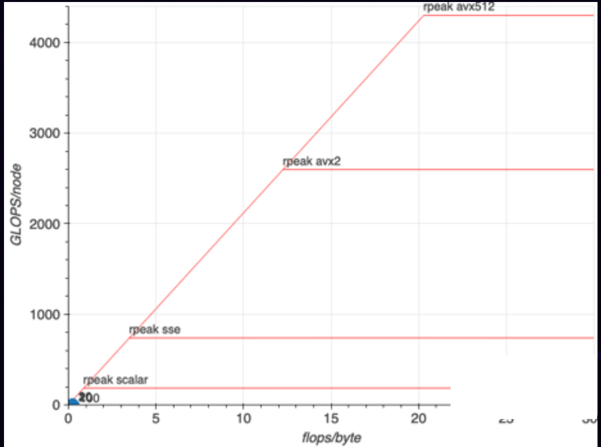
Awesome Seismic Code



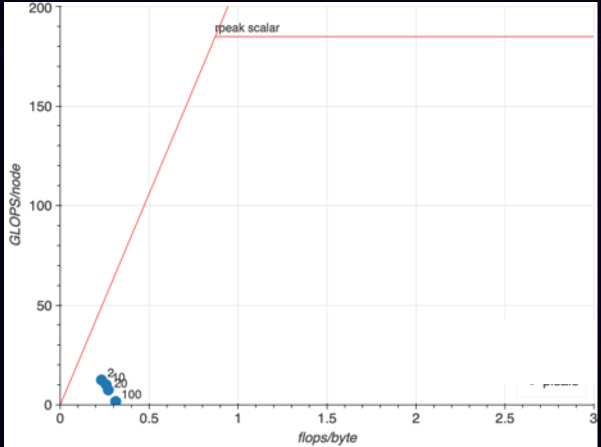
CSA EVALUATION

- Lots of evidence to make us think there is room to improve things.
- Here are some typical roofline plots from the initial eval (code names withheld to protect the guilty).

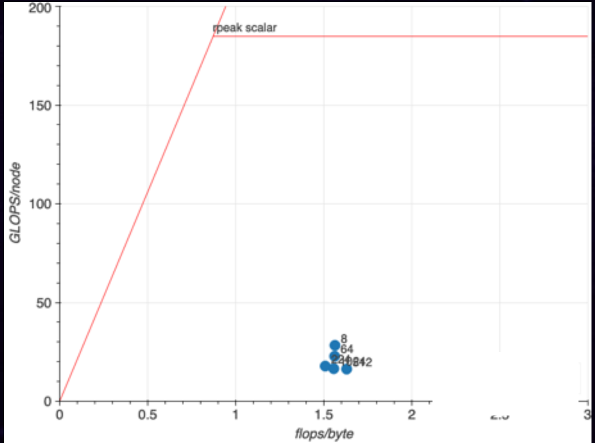
C++ Template Libraries – and a cluster near the Origin



Zoomed in – it doesn't get better.



Community codes can be terrible too.



WHAT HAVE THE CSA PROJECTS TOLD US SO FAR?

- Up to 50% are now fairly performant on NVIDIA GPUs.
  - We really need to see this reach 80%
- Starting to see success with HIP (the AMD GPU model)
- Still Fortran in the mix! Also, Python.
- Maybe 1 ready for Kokkos/Raja or OneAPI – we can't really follow the DOE path.
- These teams *can* and *do* update their codes to run at scale – so we can evolve the programming model some (not so much other teams).
- The AI focused ones can't yet get to large scale.
- There is *still room* for plenty of traditional optimization.

AND THE OTHER STUFF

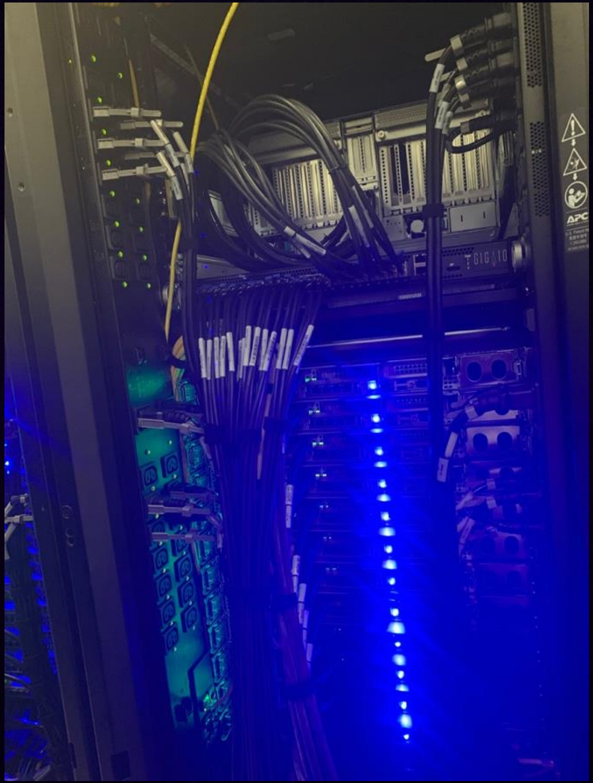
- BTW – in the timeline we are on, Quantum, AI-specific processors, are *not* serious contenders for mainline computing capacity, though they have their niche.
- And, btw, just picking the chips is the tip of the iceberg:
    - I/O and storage hierarchy, interconnect, node architecture are just as complicated decisions.
- Let’s take a dive into how we look at just one more technology, disaggregation.

DISAGGREGATION

- Or Composability, depending on whether you are an optimist or not.
- In general, taking things inside the server, and separating them, then dynamically bringing them back together – maybe memory, storage, or accelerators.
- This was the vision of Gen-Z, CAPI, etc, and is the vision for CXL, NVLINK, . . .
- We can do a pretty good job today over external PCI fabrics, or just over other low-latency fabrics.

WHERE ARE WE ON DISAGGREGATION?

- Testbeds:
  - Giga-IO: Lonestar-6
  - Liqid - Chameleon (UT/Argonne)
    - Faster (TX A&M)
  - Fungible? (coming soon)
- We also have a pretty big Rockport testbed
- And a DPU testbed
  - Those 2 may play a role in future versions of composability



SO, BACK TO OUR NEW EVALS:

HOW DO WE DECIDE IF WE PUT SOMETHING IN PRODUCTION?

- A few steps we do in any evaluation:
  - What's the hypothesis of making things better?
  - Does it *actually* work?
  - Does it perform?
  - Can we make it usable for *normal* people on our systems?
  - Does the economic model make sense?

COULD DISAGGREGATION MAKE THINGS BETTER?

- YES
- Picking system configs is among our hardest and most important tasks
  - We have all kinds of workloads
  - We have limited ability to push software changes to our users.
- Right now, we tend to put a massive amount of hardware in one homogeneous partition (Frontera -- 8,400 CPU compute nodes) and much smaller amounts in specialized subsystems (also Frontera – 16 large mem nodes, 90 quad-GPU nodes).
- Often, load conditions are such that some subsystem has idle capacity while others have wait times – this is obviously not the *best* possible thing.
  - *Caveat* -- Deep Learning Workloads are still essentially immature and all over the map.   Some are limited by the shared GPU address space. It is possible (likely) this is an artifact of the tools and not the algorithm… actually, let's take a 2 slide detour on that, because it's a relevant lesson. . .

MY TAKE ON USE CASES

- OK, there are lots of things we can disaggregate (and each will have its own value proposition):
  - Accelerators – typically GPUs, but really any PCI compute device (FPGA, IPU, Vector Engine, AI Accelerator, etc.).
  - Storage – pool remote storage devices into locally appearing block devices or filesystems (aka Wrangler).
  - Memory – Use either PCI-attached memory devices (really, CXL) or memory from a remote node.
- Keep in mind current PCI implementations are a waystation on the way to CXL (etc.) kinds of future fabrics
- Let’s dive into these separately. . .

MY TAKE ON USE CASES

- Storage – Dynamically composing storage is great; but do we need PCI/CXL level latency?
  - If not, we could do this over our conventional fabrics (NVMeOF).
  - Better software layers are needed, but roughly the entire storage industry is working on this.
  - Lessons of the past – BW/IOPS are the driver, not latency, so we probably don't need a PCI fabric for this right now.
- Remote memory
  - Here the opposite is true – we can see huge differences going from L2 to L3 cache in application performance. Latency is what matters most when finding memory in a NUMA system!!
  - CXL may bring this down some.
  - I am somewhat skeptical we will ever see great performance here.
  - *But*, in a small fraction of our systems, we have ridiculously inefficient largemem nodes, because sometimes, you just need the answer.
  - So this is probably more of a niche use case, but I'd want to have it on, say, 5% of my nodes.

MY TAKE ON USE CASES

- Accelerators
    - If there is to be a “killer app” for composability, it’s probably accelerators.
    - As previously noted, for better or worse, the current state of DL software is “fit in the address space of the GPUs on one node”.
    - >4 GPU nodes carry a premium price.
- From here, let’s look at accelerator (really, GPU) use case only.

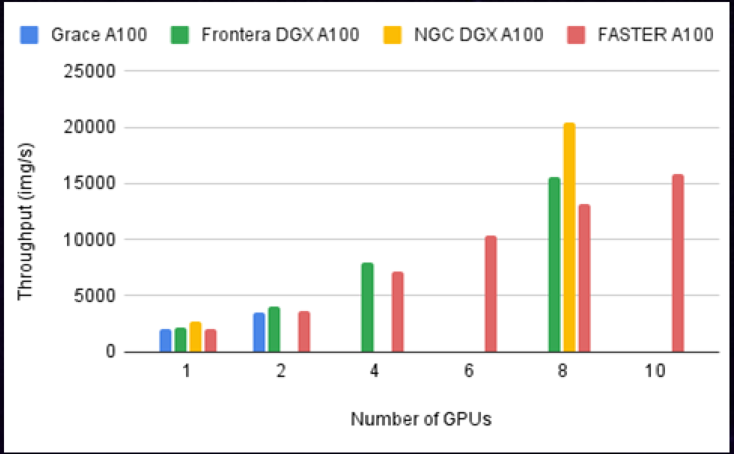
DOES IT ACTUALLY WORK?

- YES
- With both of our PCI fabric evals, and with our look at Rockport, we can verify that things function like they are supposed to – we can compose nodes, even without rebooting (!!!!), and make things appear as single large nodes.

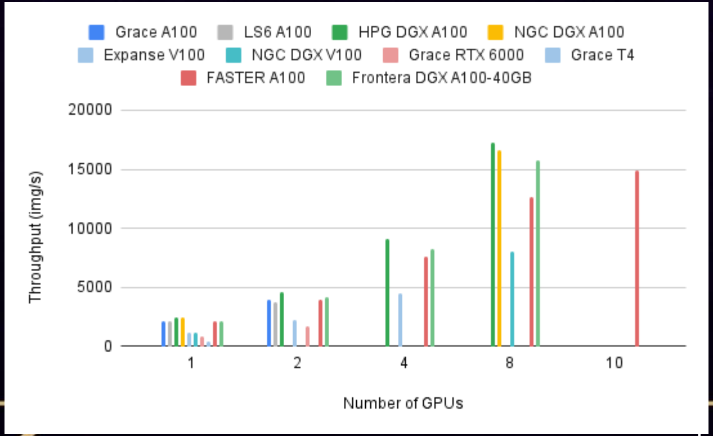
DOES IT PERFORM?

- YES, pretty well.
- Again, at least for accelerators.
- Comparisons with LIQID, all A100 GPUs, on traditional machines, DGX with 8 GPUs.
- ~80% of tuned DGX performance at 8 GPUs, scaling out to 10 GPUs.

RESNET over TensorFlow



RESNET over PyTorch



CAN WE MAKE IT USABLE?

- YES
- While algorithms for scheduling are still fun, we can provide basic Slurm integration, do the orchestration for users (transparency!), and run jobs. So, cool.
- Still messing with Kubernetes a bit, but we also have used OpenStack successfully, no reason to believe K8s won't work too (hey, it's probably the *primary* use case).
- We can also manage the physical/install rack layout stuff, so no usability barriers to introduce this.

DO THE ECONOMICS WORK?

- And here is where it gets interesting – it works, and is worth something – how much???
  - i.e. can it be sold profitably at good value for enough use cases?
  - Getting this wrong has sunk many a promising technology/company.
  - Still some work to do here.
- True Facts:
  - GPUs are expensive
  - CPUs in GPU nodes can be underutilized resources.
  - Different codes need different size nodes, and are not particularly malleable.

DO THE ECONOMICS WORK?

- More true facts:
  - GPUs are expensive. You have to buy them either way. (Though high counts per node cost non-linearly more – see SMPs).
  - CPUs in GPU nodes are underutilized, but are a tiny fraction of overall node performance.
  - Software/workloads can slowly change over time.
  - This *can* be a *third* fabric you are incorporating into a system. Which most users will never notice, but inevitably will have to be debugged at some point.
  - HPC people never pay list price (HPC=Half Price Computing)

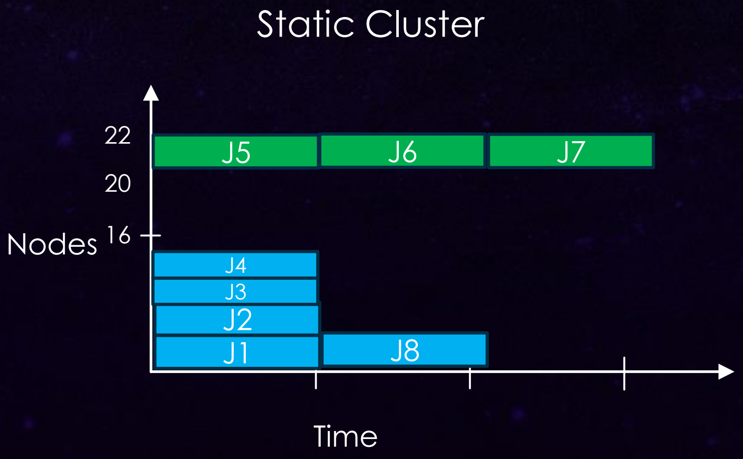
A LITTLE ANALYSIS ON THIS

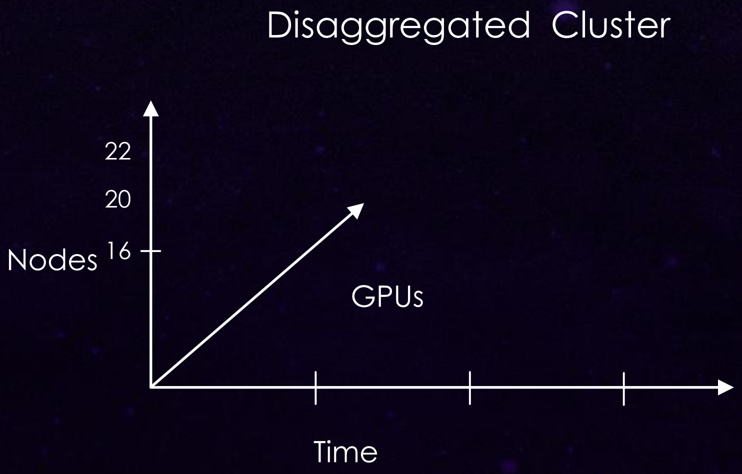
- (Frankly, it needs a lot of analysis, but my queuing theory, random variables, and programming are all a tad rusty).
- Can we make the case for some upper/lower bounds on value?
  - Sure we can.
- Let’s look at just one sample scenario, because by the time I hit this slide, I’m probably over time. . .

EXAMPLE

- Assume I have two clusters, with and without composability:
  - Cluster A, 21 nodes, 8 GPUs, 16 CPU-only, 4 1-GPU nodes, one 4 GPU node.
  - Cluster B, 21 nodes, 8 GPUs, fully composable.
- Also assume a few different workloads:
  - All CPU jobs, where we choose to reserve the GPU nodes.
  - A mix of CPU jobs and quad-GPU jobs.
  - (All fixed length of 1 hour for easier graphics).
  - Note this is a 3D problem for composable/disaggregated!
- We can run some metrics and compare things like turn around time, average utilization, time to complete the whole workload, etc.

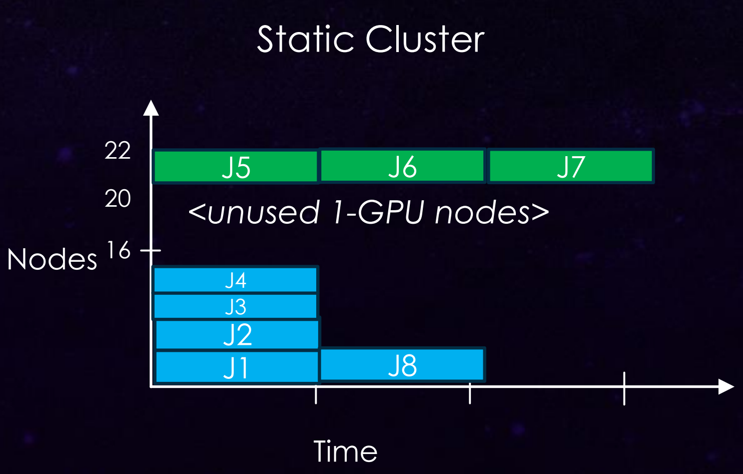
EXAMPLE – WORKLOAD 2

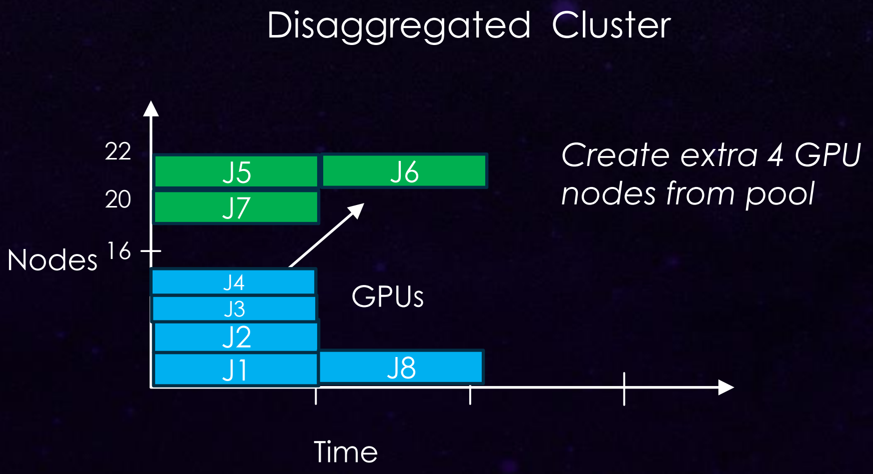




A mix of CPU jobs, 1-6 nodes, and quad-GPU jobs, all 1 hour long.

EXAMPLE – WORKLOAD 2



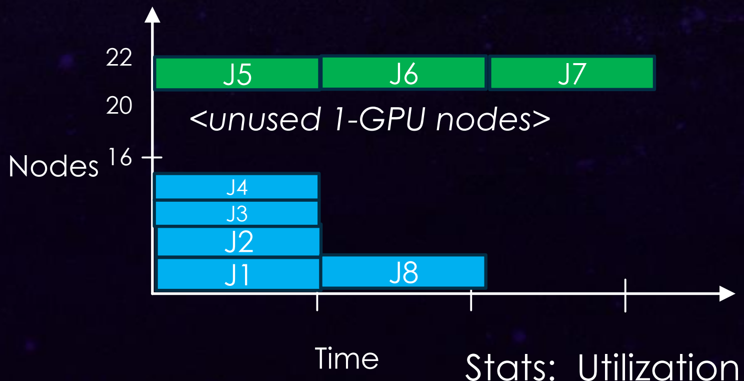


Create extra 4 GPU nodes from pool

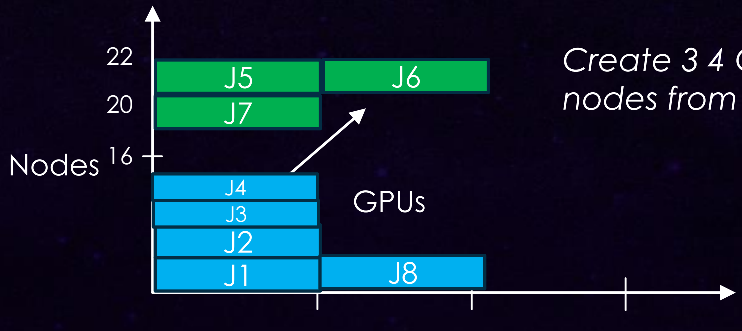
Stats: Utilization +18%
Turn-around Time +28%
Time to complete +33%

EXAMPLE – WORKLOAD 2

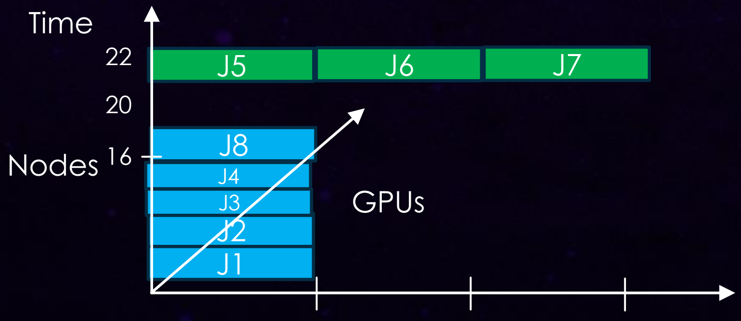
Static Cluster



Disaggregated Cluster



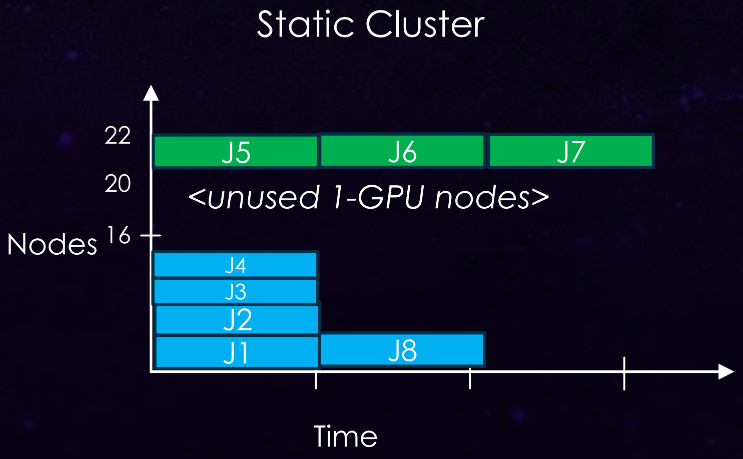
Create 3 4 GPU nodes from pool

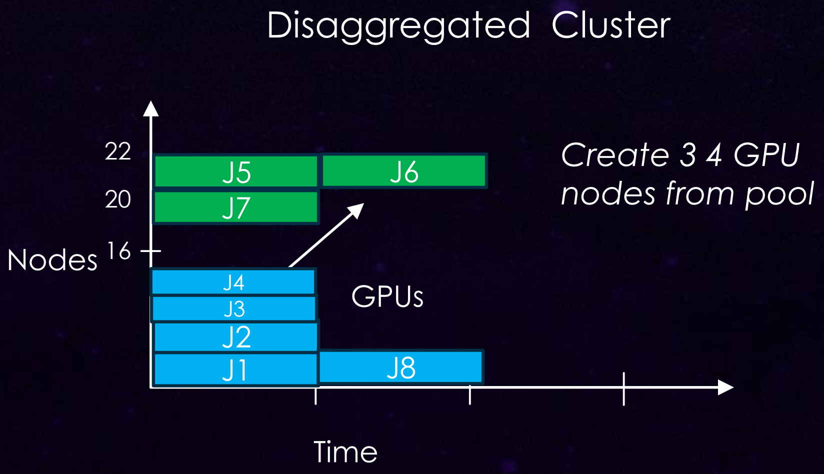


Stats: Utilization +18%
Turn-around Time +28%
Time to complete +33%

But Consider – If J8 arrived before J7, the advantage would be near-zero, other than a small improvement in turn-around-time!!

EXAMPLE – WORKLOAD 2





Create 3 4 GPU nodes from pool

Cost analysis: If we assume – (1) CPU node costs are fixed, (2) GPU nodes are fixed, (3) but a node that can hold 4 GPUs cost 50% more than a "normal" node (4) my standard discounts apply.

Then, if 18% utilization improvement were *average*, you would be justified putting **14%** of your total system cost into the Disaggregation hardware.

*repeating this with DGX 8 way nodes and slightly larger workloads, this jumps **to 18%**

SO THAT MEANS. . .

- OK, so it seems fair to say, if you do some more experiments, that unless you can fill every node type all the time, you will see utilization improvements. Sometimes small, sometimes large, but maybe 15% for a largely heterogeneous system.
- There are many confounding factors:
  - What if you can run a bigger job (e.g. 10 GPU) than you could before – what is that worth?
  - What if we could *replace* one of the fabrics with the PCI/CXL fabric – e.g. not have infiniband in every node?
    - Tough in our “little oversubscription” environment, but the IB/OPA network typically is 15% of our system cost (HCA, ports, cables).

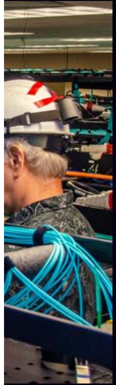
CONCLUSIONS(?) ON DISAGGREGATION

- Disaggregation opens up some exciting possibilities.
- But adds complexity and cost.
- It definitely works. And will work better once CXL arrives.
- It is pretty performant, and definitely adds value
- How much of your system budget should go towards it?
  - My current leaning (personal opinion!) is it's worth it in some fraction of the system to add new capabilities that would otherwise see little utilization.
  - What are the new capabilities worth to you?
  - What happens if (when) DL software changes?

SO… WHAT CAN I SAY ABOUT THE SYSTEM

- We believe a CPU-driven system (with enough memory bandwidth improvements) could get us to our goal within budget.
  - More nodes than Frontera – but not that many more.
  - Probably less than 20k sockets, whether CPU/GPU/other.
- Yet there is a pretty strong chance a mostly accelerated system will be better value.
  - But not all GPUs are created equal.
- Our "base" design is somewhat "vanilla" (except for filesystems).
  - But disaggregation/composable *could* still play a role.
  - AI chips/other accelerators *could* still play a role.
    - But have to show value in science throughput/$ across a *wide* swath of production applications.

TH



FRONTERA
TACC
NSF
TEXAS
NSF