



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA

CREATE CHANGE

UQ's next gen ESS journey

AKA: What we do in the shadows, minus those hilarious vampires from New Zealand.

Jake Carroll, Chief Technology Officer, Research Computing Centre, The University of Queensland, Australia.

jake.carroll@uq.edu.au

It was a cold day in Colorado...

At SC 2019, UQ and IBM talked about doing things together around early access for the next ESS. We've moved beyond "storage" and into more strategic discussions about "capability".

Several IBM employees championed this idea with us as they saw value in working closely together.

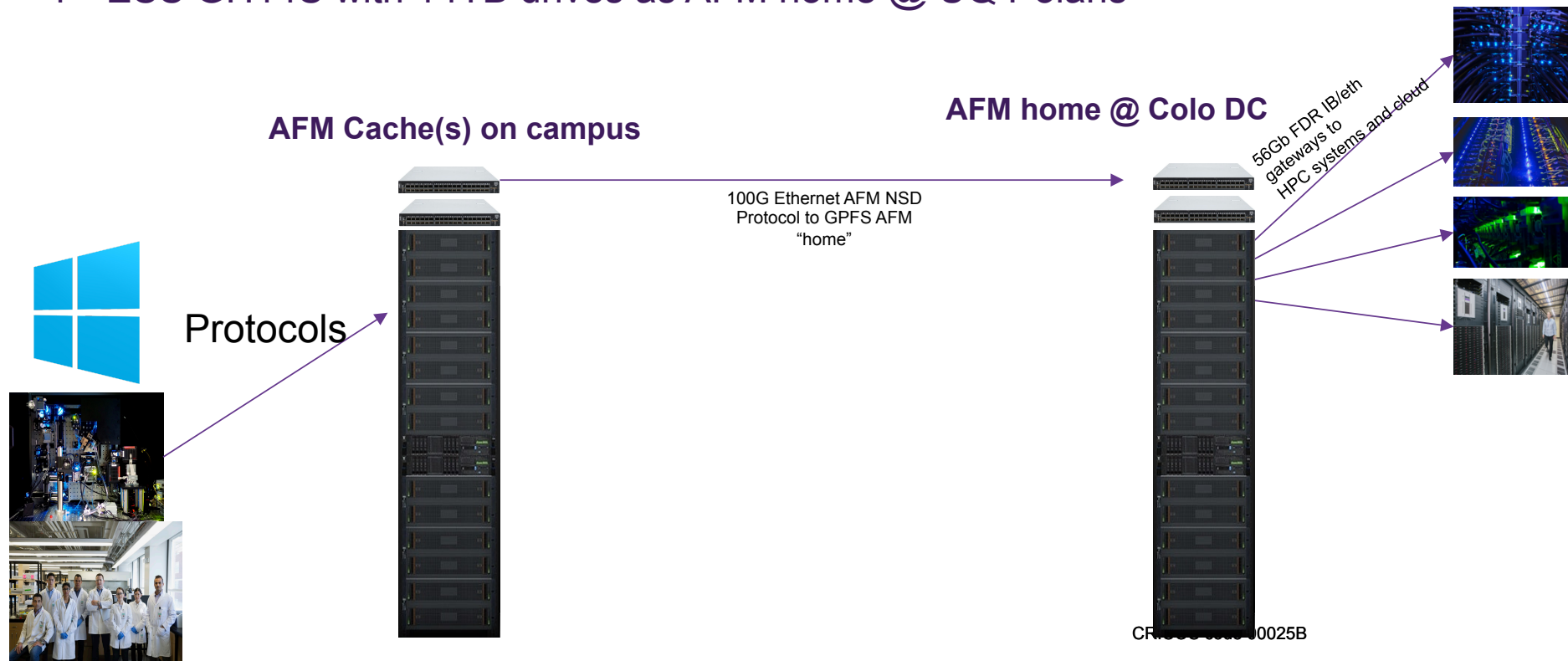
This is a brief story of how it came together and what we've found, so far...



UQ is no stranger to ESS.

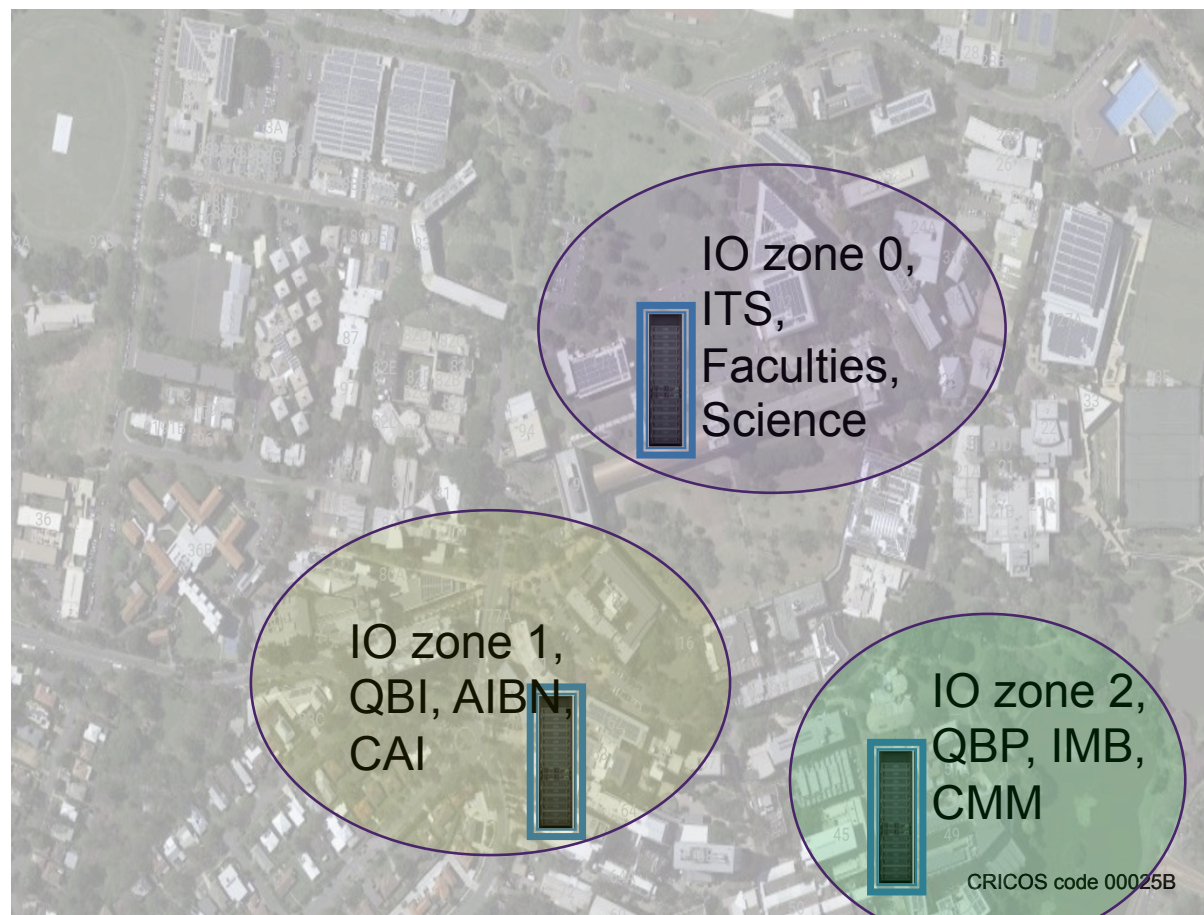
We've got a few of them now...

- 1 * ESS GH14S with 4TB drives as AFM cache + HPC Scratch @ UQ QBI, St Lucia
- 1 * ESS GH14S with 10TB drives as AFM cache @ UQ IMB
- 1 * ESS GH14S with 14TB drives as AFM home @ UQ Polaris



We developed AFM cache “zones” around our campus

Cache and IO zone distribution around campus – putting caches near our instruments and IO intensive locations, using AFM, protocols and an SMB, to “knit” the fabric together...



David Abramson's MeDiCI data fabric vision was realised.

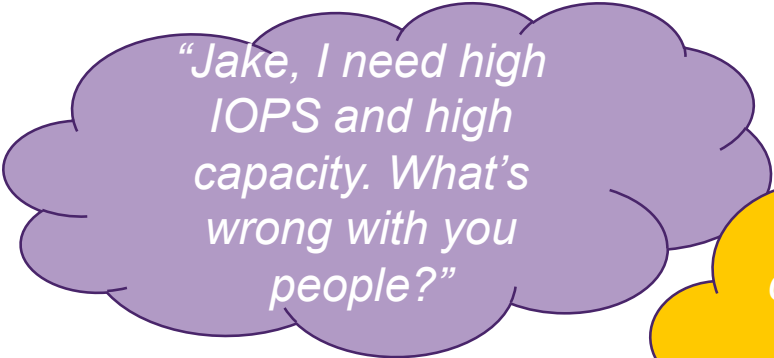


Me

David

Check. What comes next?

- We started thinking about where we head next. Importantly, we didn't do that by having a “go faster” discussion. We did it by looking at what our workloads were doing and where our strategic growth drivers were.
- It is incredibly tempting to “go buy the faster one” when it exists, but you lose sight of what you're trying to achieve when you do that.
- What **problems** did we need to solve, next?



“Jake, I need high IOPS and high capacity. What's wrong with you people?”



I need big streaming IO. I don't care about your trendy ML/AI workloads”



“I need more collection storage in my AFM filesets so I can spring it up into scratch later...”



“I want the device to acknowledge my IO faster but I also want capacity”

Market analysis time.

- We started getting inundated by all flash discussions. Vendor land kept pumping up the solid-wall-of-flash-for-your-next-scratch array premise to us. Seemed really strange to us, because despite all the economics that they claimed, we just couldn't make good on it, \$/PB-wise. Not even close.
- Trends from the market:
 - *"You want an all flash device"*
 - *"You can do some great stuff with dedupe and compression with our flash product X"*
 - *"Nah, you don't need parallel filesystems anymore bro. That just makes stuff more complicated"*
 - *"Tiering? Nah. You don't need that anymore. You can run it all in the one place with this much IO"*
 - *"We've got this cool <insert something about composable infrastructure something or other switch> and it is faster than all the others"*
- But what do we see?
 - Good luck getting all those apparently attainable IOPS out of your "one way in, one way out" filesystems and bricks.
 - Their economics and "erosion of flash" discussions are fantasy and theatre, at the moment.
 - The technologies behind the interconnects are vastly outstripping the bus. Exemplar, the SAS-Gen3/12Gbit/sec limits of current host bus adapters.
 - [Some] companies are getting much (much) smarter about how much they can squish and squeeze out of ye-olde NL-SAS spindles.

What we're gravitating towards.

- I *think* we're beginning to approach some of the problems the pre-exascale people were, about three or four years ago.
- We are nowhere near their scale, but some of the problems characterise the same way.
 - The "IO acknowledgement fast/checkpoint it all NOW" problem. We're seeing that with big ML solvers.
 - The "I need to throw 70,000 4k files into the air and train them then acknowledge them all back without IO wait" problem. Again, label and training runs in ML/AI solvers push this stuff incredibly hard – but we're finding this also is the case with some of our Oxford NanoPore/ PacBio longread sequencers codes like the GPU accelerated code, GUPPY.
- Where the IBM ESS GH14S did well – it has some SSD inside each of the JBOD canisters for the GPFS logtip device for write acknowledgement.
- Where it didn't do so well? We could drown that pretty convincingly if there were a couple of mixed aggressive codes.
- In our estimation it actually slowed the maximum aggregate IO of the ESS down as perceived by the host due to IO wait state being fed back up the chain to the host/OS layer of the GSS-IO's and NSD clients driving the workload...
- So – we needed something that was better at this. Was the GH14S bad at it? Gosh no. It just wasn't **epic** at it...and we needed **epic...**.

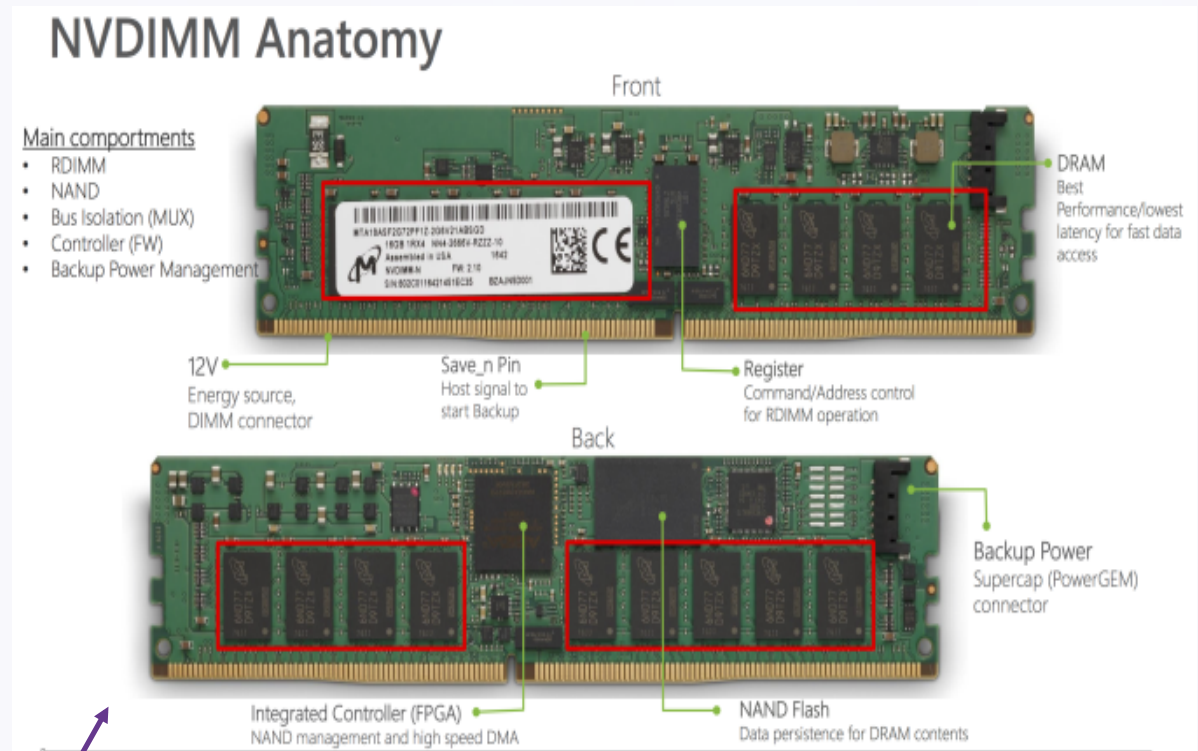
What happened next? UQ+IBM EAP* for the IBM ESS5000.

UQ was the first entity in the world to have an ESS5000 running outside of development/IBM engineering. One of three globally.

* Early Access Program

Hello IBM ESS 5000.

- What is different?
 - Power9 vs Power8, for starters.
 - Super dense SAS links. More dense than anything I've seen before. 4 * optical lines out of each SAS HBA come out of the back of each port.
 - NVDIMM logtip device inside each Power9 GSS-IO node [2 * 64GB NVDIMM arrays].
 - A whole bucket load more PCI-E Gen4 bandwidth.
 - GNR looks and feels pretty different now.



GNR logtip lives on these. IO acknowledgement through the cache chain “here” first in conjunction with GPFS PagePool.

There is no unboxing video. *Sorry.*

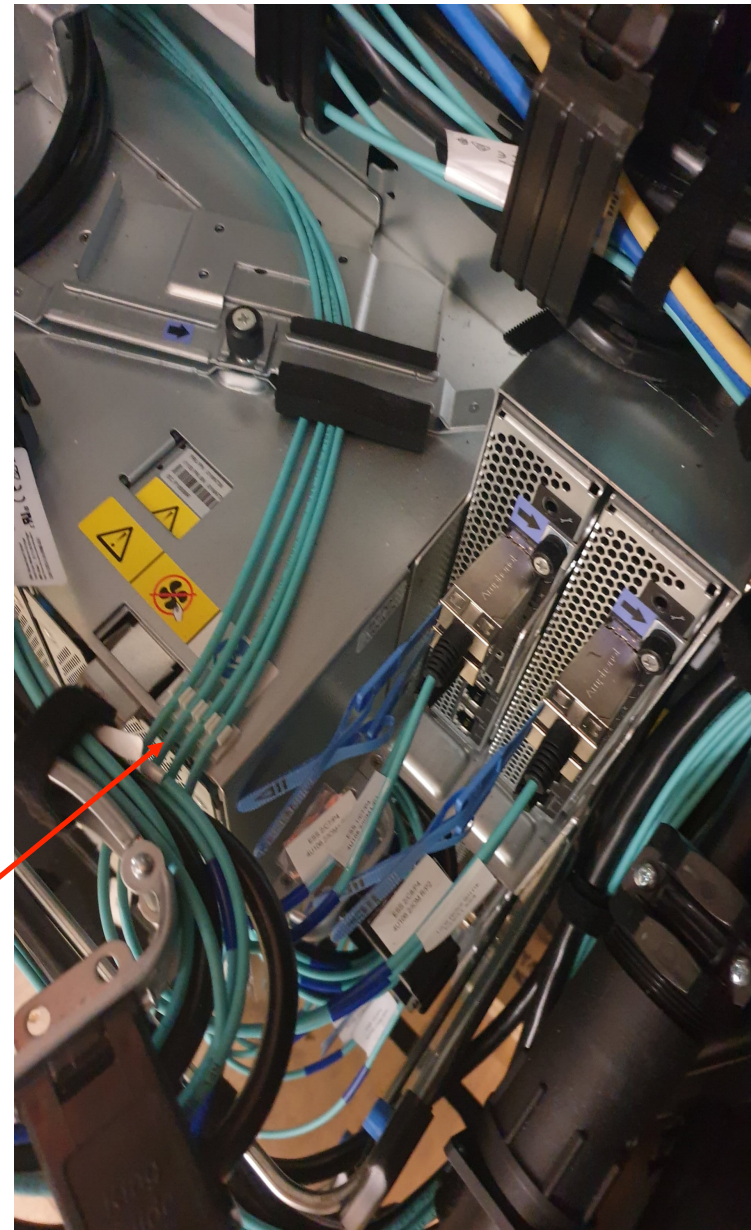


- Rack was bolted to pallets in a way that prevented it from being removed.
 - Shane (our local IBM guy) and a bunch of people in HiViz needed to essentially take the rack apart to remove it from the pallet it was bolted to.
 - This was fed back to IBM's *experience* team. They have rectified the way shipping occurs.

Those dense SAS channels.

- So dense, that we actually had out of box delivery issues on the engineering EAP units because of the way the optical interconnects got plumbed in.
- Crushed Cable one one of the SAS optical channels spotted by Michael Mallon the morning of unboxing.
- IBM engineering fixing this for production units that you would potentially obtain if you procure one of these.

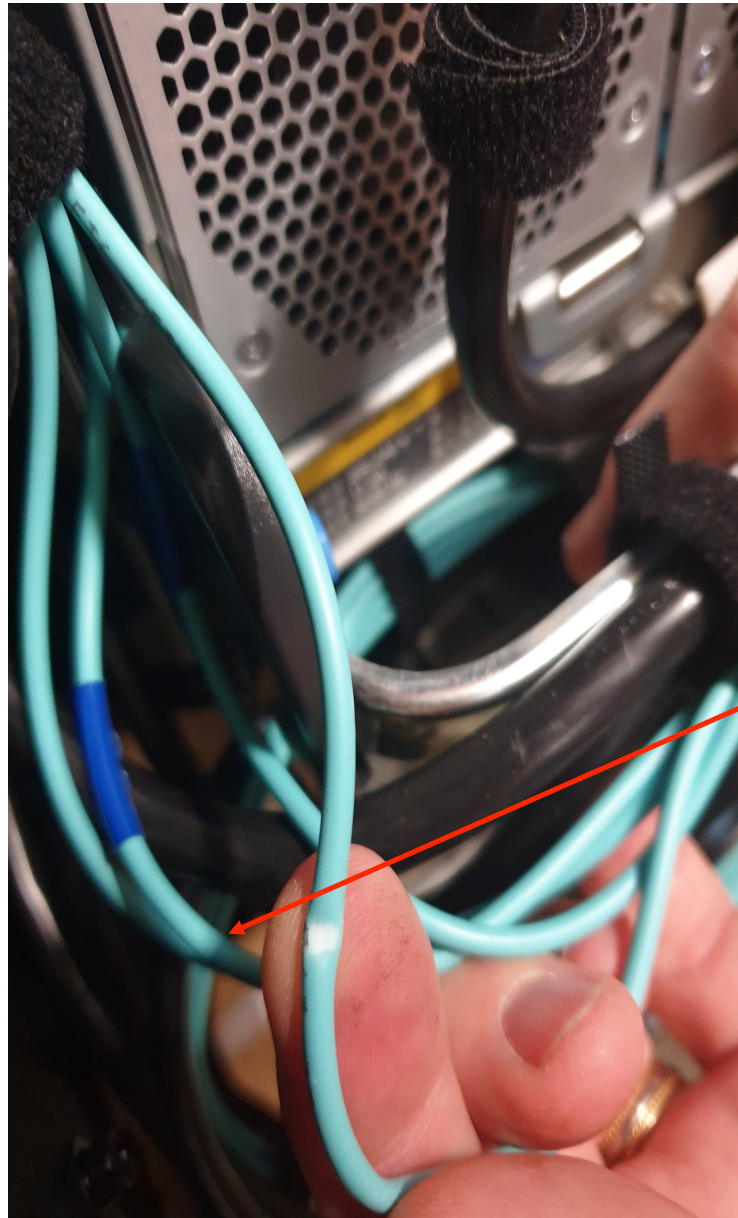
Stuff gets tight in here, 4 * wide optical SAS bus



Mmm...crimpy.



Do not get yourself
into a crimp-off with
these guys. You won't
win.



*"Oh, that isn't
good", said
Michael.*

CPU planar on Power 9 needs replacement – thermals.

We got early access/beta warning about some thermal planar issues on some of our CPU boards.

I don't know a lot about it – but I know IBM have people replacing some planar components for us this Friday to bring it up to “production” specification and tolerances.



What does an ESS 5000 look like?

- Roughly, this.



This particular unit shipped with 2 * large 106 (?) drive tray enclosures plus Power9 GSS-IO's and EMS nodes. *Very shiny.*



That's the hardware. What about software?

- Spectrum Scale “6.0.1” so Andrew tells me. I probably should check the yum outputs...typical confusing fashion of package names:

```
[root@carlo-ems2 ~]# rpm -qa | grep -i gpfs
gpfs.base-5.0.5-1.1.ppc64le
gpfs.java-5.0.5-1.ppc64le
gpfs.gpl-5.0.5-1.1.noarch
gpfs.msg.en_US-5.0.5-1.200629.104017.noarch
gpfs.ess.firmware-6.0.0-5.ppc64le
gpfs.gskit-8.0.55-12.ppc64le
gpfs.gss.pmsensors-5.0.5-1.el8.ppc64le
gpfs.adv-5.0.5-1.1.ppc64le
gpfs.compression-5.0.5-1.1.ppc64le
gpfs.gss.pmcollector-5.0.5-1.el8.ppc64le
gpfs.crypto-5.0.5-1.1.ppc64le
gpfs.gnr-5.0.5-1.1.ppc64le
gpfs.gnr.support-ess3000-1.0.0-1.noarch
gpfs.gnr.support-essbase-1.0.0-1.noarch
gpfs.gnr.support-ess5000-1.0.0-0.noarch
gpfs.ess.tools-6.0.1.0-0.el7.noarch
gpfs.gui-5.0.5-1.noarch
gpfs.docs-5.0.5-1.1.noarch
gpfs.license.dmd-5.0.5-1.1.ppc64le
```

Why the ESS 5000 is useful to us.

- It took us a little bit of effort and back and forth to arrive here, but we now achieve better perf with two trays of NL-SAS disk linked up to GSS-IO's on Power9 populated with NVDIMM on most of our workloads, compared to the GH14S with trays of dedicated meta-data on SSD. *Interesting, eh?*
- TL;DR – I'm satisfying my noisy users and their "*more IOPS. More this. More everything*" but I'm not laying out huge piles of money (that I don't have) for flash for my meta data or my primary filesystems using hot/warm tiering to do it. It isn't what I expected.
- Do I like IOPS, lower lowait state and more bandwidth without having to gold-plate with flash? *Yeh.*
- As a colleague pointed out, I'm getting roughly the same, if not better performance than my GH14S fleet but I'm doing it with less physical space in the rack and less expensive media technologies.
- Maybe it all goes without saying and this was an inevitability – but I didn't expect the Power9 + NVDIMM combination to be able to do this in confluence with NL-SAS spindles.

Workloads that exemplify ESS 5000 logtip IO superiority.

Application and Workload	IBM GH14S /scratch, 16MB block	IBM ESS5000, TestESS5k16M, 16MB block
[CryoEM] Relion 3.1.0-mvapich2-cuda10.2; nCOV-Refine3D	13:22:19	10:18:36
[NGS] Guppy-cuda-10.2; Wheat-Genome-Basecall	0:48:22	0:36:11
[MRI] Python Unet PyTorch-cuda10.2; Human Brain Auto-segmentation	26:41:12	21:09:29
[Molecular Dynamics] Desmond Schrödinger 2020-02-cuda10.2, Toxin-finder	3:11:19:46	3:11:21:16

Because people seem to care way too much (I don't)...io500

IO500 ISC Edition:
UQ GH14S, Ten

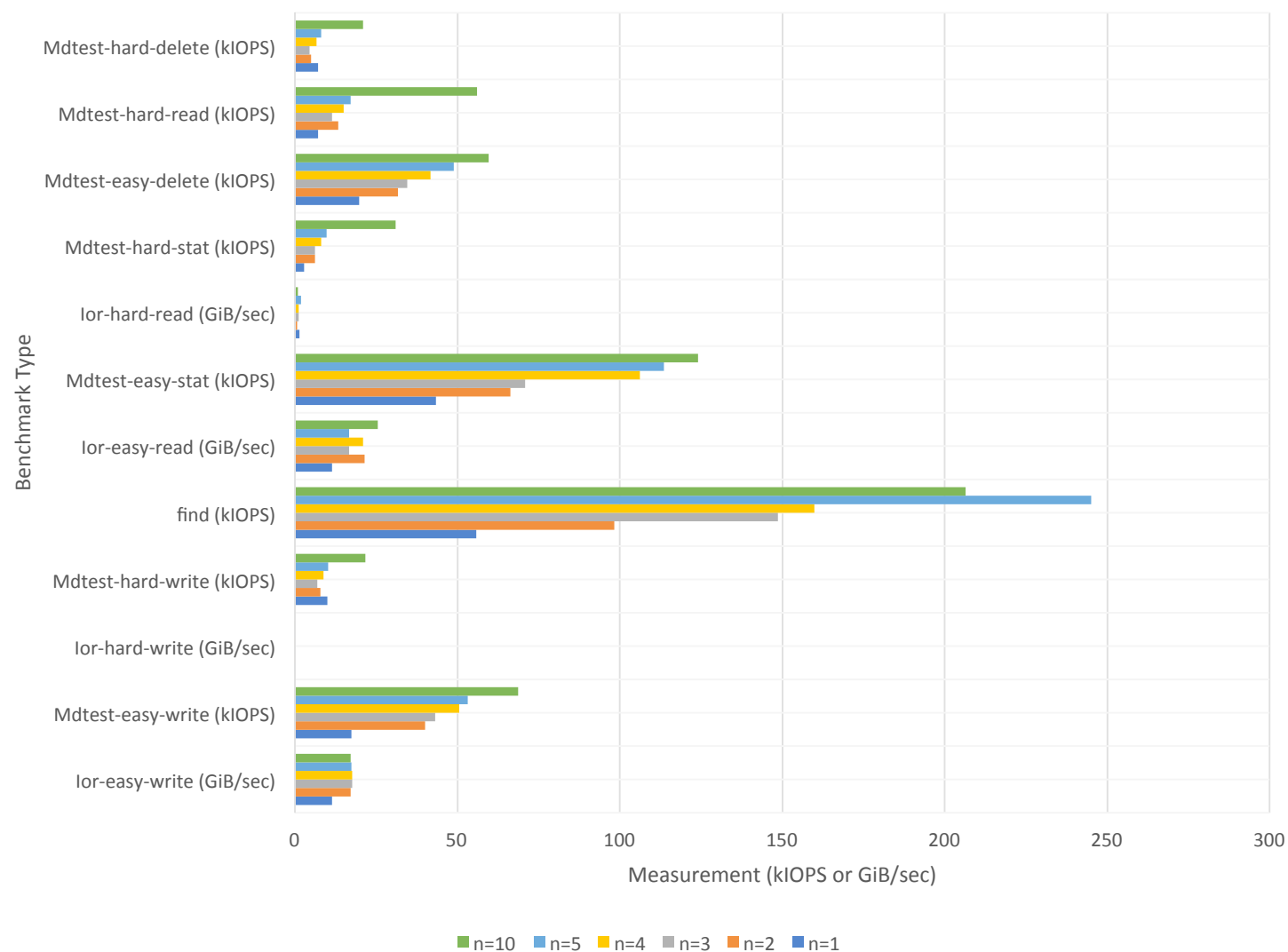
Node	lor- easy- write (GiB/ sec)	Mdtest- easy- write (kIOPS)	lor- hard- write (GiB/ sec)	Mdtest- hard- write (kIOPS)	find (kIOPS)	lor- easy- read (GiB/ sec)	Mdtest- easy- stat (kIOPS)	lor- hard- read (GiB/ sec)	Mdtest- hard- stat (kIOPS)	Mdtest- easy- delete (kIOPS)	Mdtest- hard- read (kIOPS)	Mdtest- hard- delete (kIOPS)	Aggreg ate BW Score (GiB/ sec)	Aggreg ate IOPS Score (kIOPS)	IO500 Total
10	17.2002	68.6886	0.25701	21.7139	206.444	25.4068	124.014	0.96940	30.8919	59.6602	56.1483	21.0456	3.23026	54.9529	13.3233
	4	9	8	5	4	6	4	2	6	5	8	21.0456	5	5	9

IO500 ISC Edition:
UQ ESS5000, Ten

Node	lor- easy- write (GiB/ sec)	Mdtest- easy- write (kIOPS)	lor- hard- write (GiB/ sec)	Mdtest- hard- write (kIOPS)	find (kIOPS)	lor- easy- read (GiB/ sec)	Mdtest- easy- stat (kIOPS)	lor- hard- read (GiB/ sec)	Mdtest- hard- stat (kIOPS)	Mdtest- easy- delete (kIOPS)	Mdtest- hard- read (kIOPS)	Mdtest- hard- delete (kIOPS)	Aggreg ate BW Score (GiB/ sec)	Aggreg ate IOPS Score (kIOPS)	IO500 Total
10	18.8000	58.4923	0.25113	17.8102	262.021	21.4057	77.8751	2.27599	43.9069	37.9825	43.2487	10.0027	3.89439	44.4850	13.1621
	7	4	1	2	4	4	9	3	6	6	4	5	8	8	7

We think this is NVDIMM logtip in action...(so do IBM engineering)

Effect of IB NSD n client count on IO Parallelism of ESS5000, 16MB block size,
io500-isc-2020



Where do we head next?

- Probably pair it with the ESS 3000 we have as part of the work we're doing here and start to mix and meld for other reasons. We believe that many of our meta-data intensive operations on /scratch will still benefit immensely from a flash layer.
- Probably do some basic work around hot/warm block migration via policy into flash, where appropriate, similar to Sean @ UoM's ideas and work so far.
- Start looking at next-gen bandwidth blocker fixing – SAS-Gen4 options. We know SAS-Gen3 is a critical bottleneck as things currently stand.
- GNR improvements.
- Features...

Take away from the EAP

- It has been incredibly valuable for us to be on the inside track of the EAP program. We got to actually help in the design, experience and the process of the product that people will consume in GA form.
- It is time consuming. If you don't have the resources to dedicate to it, this isn't for you. This isn't for everyone – but I would absolutely advocate for the approach in order to design and build better products together. In the “biz” world, we call this value co-creation. Some people just call it doing stuff together so that you get a valuable outcome, together that everyone “wins” from.
- COVID-19 did bring its challenges – but given we've been working remotely with the US teams the whole time and it wouldn't have been any different even if COVID-19 had never existed – the remote collaboration aspects of it worked for us.
- Deep engagement from all angles of the IBM sphere, which we've found really interesting. Everyone from product experience groups, through to the Power team, product offering, deep engineering and hardware dev involved. We got to hang out with some really cool and interesting people along the way – and we're really thankful for that.



THE UNIVERSITY
OF QUEENSLAND
AUSTRALIA

CREATE CHANGE

Thank you

Jake Carroll | Chief Technology Officer
Research Computing Centre
jake.carroll@uq.edu.au
07 3346 6407



facebook.com/uniofqld



[Instagram.com/uniofqld](https://instagram.com/uniofqld)



twitter.com/RCCUQ



facebook.com/rccuq

<https://rcc.uq.edu.au>

A lot of people to thank:

At UQ:

- David Abramson
- Rob Moffatt
- Irek Porebski
- Michael Mallon
- Stephen Bird
- Sarah Walters
- Doug Stetner
- Matthew Bryant
- Leslie Elliot
- Owen Powell
- Paul Schakel

At IBM:

- Andrew Beattie
- Shane Handcock
- An Chen
- Chris Maestas
- Jodi Everdon
- Luis Bolinches
- Doug Petteway
- Puneet Chaudhary
- Kedar Karmarkar
- Suad Musovich