Benchmarking discussion

Fengping Hu, Lincoln Bryant, Ilija Vukotic
Enrico Fermi Institute
University of Chicago

IRIS-HEP AGC workshop 2023
05.05.2023

Target

- Process 200TB in 20 minutes
  - Assume read 10% ⟶ 20TB/20*60s = 133Gbps
  - Assume each event is 2KB and 10% is streamed ⟶ 133*10^6/8*0.2 = 83.5Mevents/s
  - Assume 25 kevents/core/second => needs 3340 cores
  - 133Gbps/35 nodes ⟶ 3.8Gbps per node
  - Each core 133 Gpbs/3340 = 40Mbps

Sanity Check

	Target	**UChicago AF**
Disk	200 TB	>1 PB
Network	133 Gbps	200 Gbps WAN >200 Gbps LAN
CPU	3340 CPU Cores	4560 CPU Cores (K8S)

- The target is well-scoped for the UChicago AF

Throughput test setup

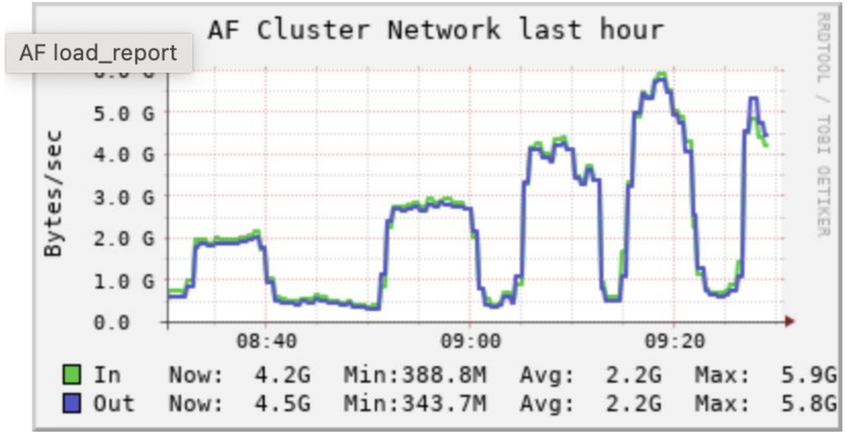
- Coffea-casa Dask workers read data from ~800 files across 15 fast compute nodes with 25Gbps links
- Each Dask worker requests a few selected columns via:

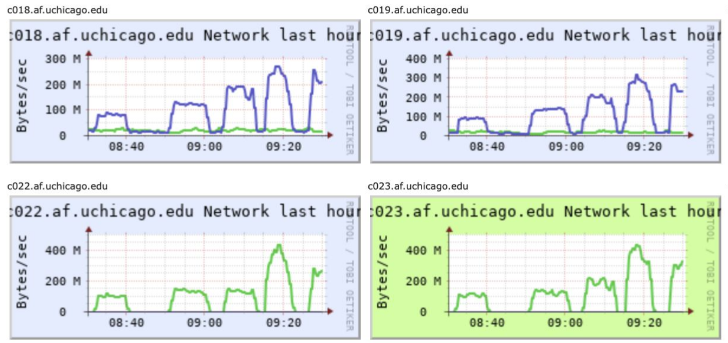
```
uproot.open(filename)["Events"].arrays(arrays_to_read)
```

- Compared three file access methods:
  - Locally, via CephFS shared filesystem:
    - 19 hyperconverged nodes with spinning disks and 10Gbps links
  - Remotely, via XRootD at UNL:
    - https://xrootd-local.unl.edu:1094//store/user/AGC/nanoAOD/
  - Remotely, via XRootD with a local XCache at UChicago

Local filesystem throughput scaling with number of workers

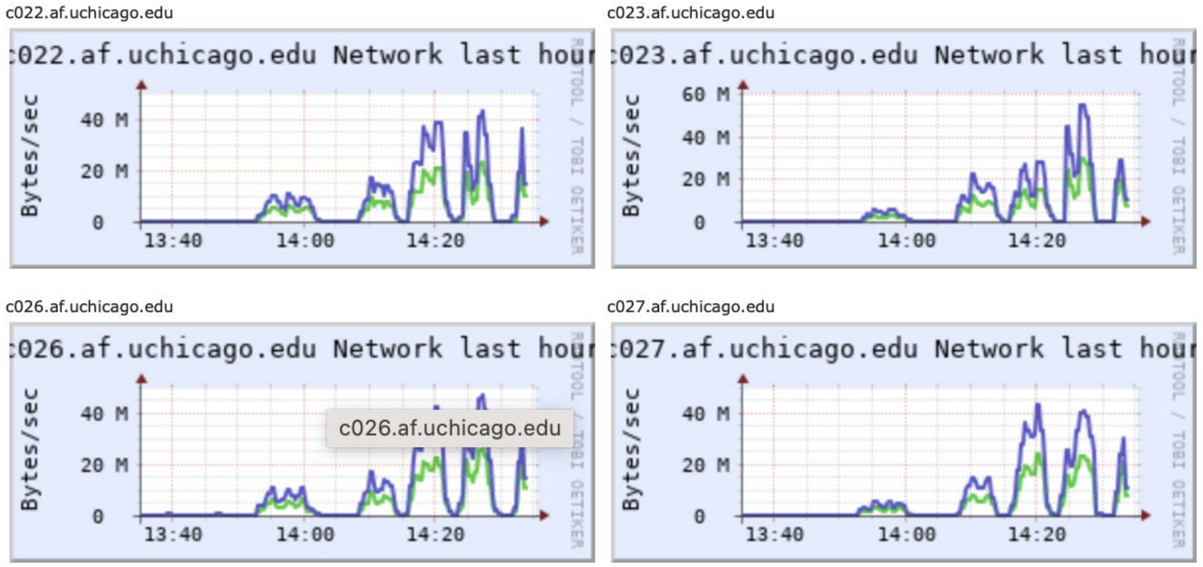
- From left to right the number or workers: 50(~1GB/s), 100(~2GB/s), 200(~3GB/s), 400(5GB/s), 800(5GB/s)
- Scales almost linearly in the beginning
- 10–20MBps per worker





Remote files throughput scaling with number of workers

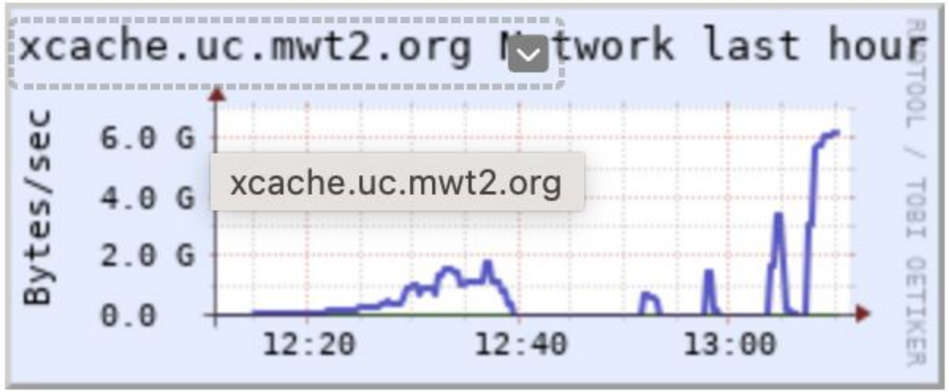
- Scales linearly from 50, 100 and 200 workers, bottlenecked somewhere from 200 workers to 400 workers
- Total is 40MBps/node * 15 nodes = 600MBps
- Per workers is 600MBps / 200 workers = 3MBps/worker

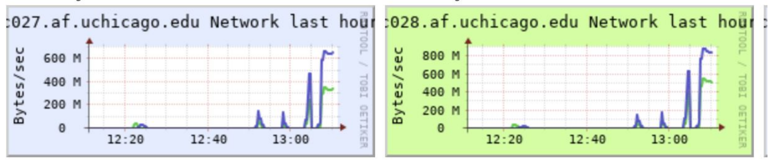


Remote files throughput with xcache

- xcache.af.uchicago.edu:1094
- Xcache at UChicago AF: 2*25Gbps links

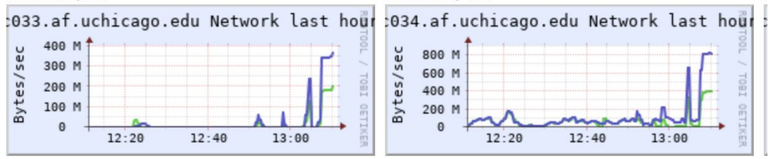
xcache.uc.mwt2.org





c033.af.uchicago.edu

c034.af.uchicago.edu

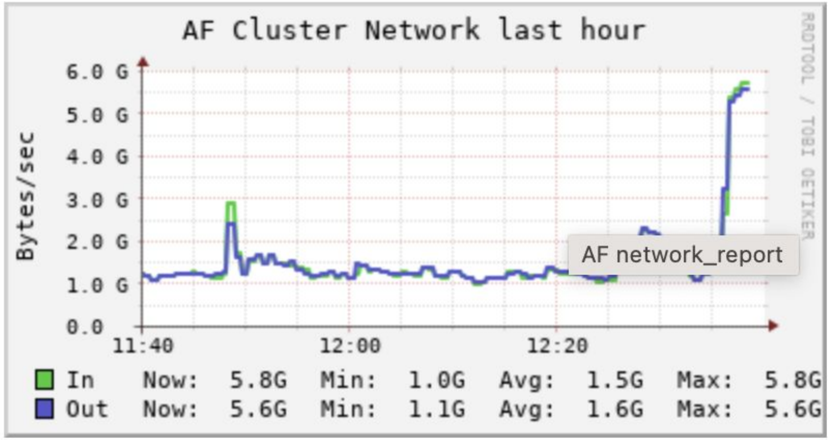


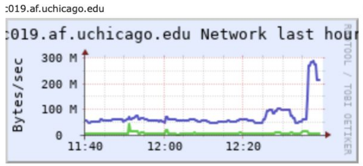
Misc data points

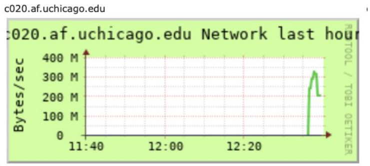
- Cat file to /dev/null – 113MBps
- Cat file to /dev/null after os cache – 3300MBps
- uproot.iterate(f”{fname}:Events”, expressions = arrays_to_read)
  - Read a few columns –50MBps(filesize/time)
  - Read a few columns with os cache – 823MBps
  - Read all the columns – 35MBps
  - Read all the columns with os cache – 113MBps

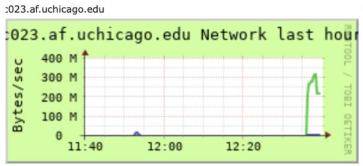
Running the benchmark as htcondor job directly vs in dask

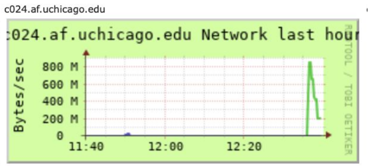
- Dask worker are also Docker universe HTCondor jobs, we can control the number of workers via the scaling parameters
- With direct HTCondor jobs, we run one task as one job, so there will be 787 jobs running concurrently ,this will be similar to the 800 Dask worker case
- The data throughput graph shows the similar results indeed











Meeting the Challenge

- What needs to be improved to hit our target of 200TB in 20min?
    - Identify and resolve bottlenecks in the Dask pipeline

ServiceX data Access

Tested current production version (1.1.4).

Two instanced deployed on UC AF (xAOD and Uproot).

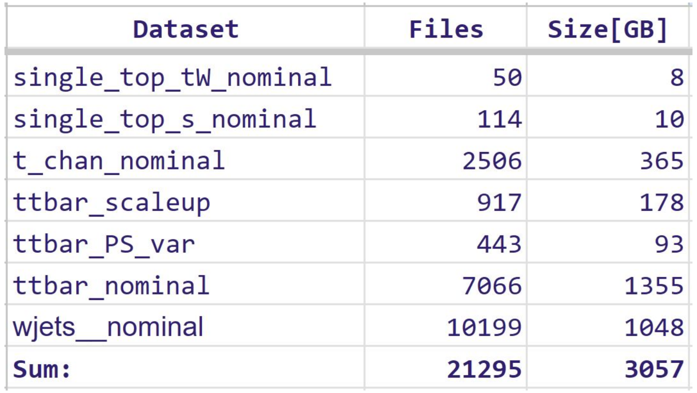
Allocated 1k cores for the tests.

Each dataset run concurrently (using ServicexDataBinder).

Datasets:

- Uproot – 6 datasets – 3TB in 21k files
- xAOD – 9 datasets – 136TB in 117k files

ServiceX – Uproot



Dataset	Files	Size[GB]
single_top_tW_nominal	50	8
single_top_s_nominal	114	10
t_chan_nominal	2506	365
ttbar_scaleup	917	178
ttbar_PS_var	443	93
ttbar_nominal	7066	1355
wjets__nominal	10199	1048
Sum:	**21295**	**3057**

Reading a single variable.

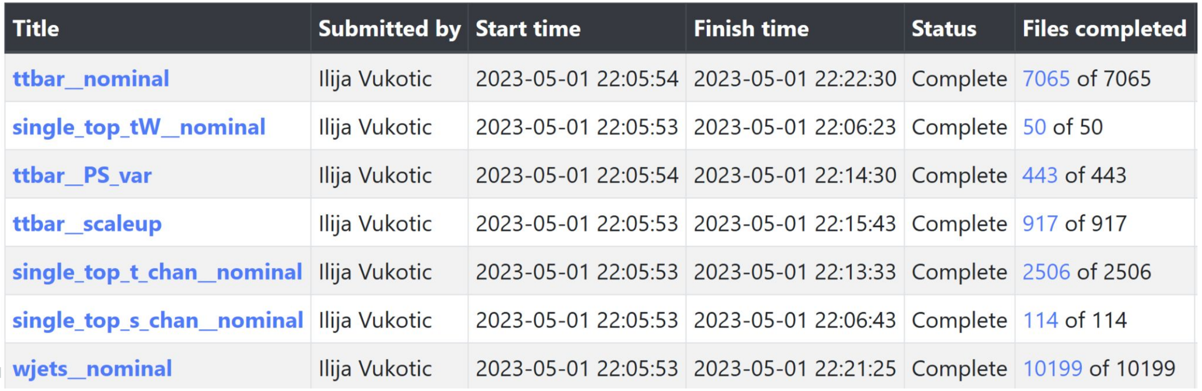
When reading remotely, it takes hours.

All requests stay at 10 transformers – default minimal scale.

Reading fully cached data: 16 minutes.

CPU utilization never goes above 15%, so horizontal autoscalers never trigger (default is 30%).

Had to manually lower HOA to 10%, then it scales up but not very fast.



Title	Submitted by	Start time	Finish time	Status	Files completed
ttbar__nominal	Ilija Vukotic	2023-05-01 22:05:54	2023-05-01 22:22:30	Complete	7065 of 7065
single_top_tW__nominal	Ilija Vukotic	2023-05-01 22:05:53	2023-05-01 22:06:23	Complete	50 of 50
ttbar__PS_var	Ilija Vukotic	2023-05-01 22:05:54	2023-05-01 22:14:30	Complete	443 of 443
ttbar__scaleup	Ilija Vukotic	2023-05-01 22:05:53	2023-05-01 22:15:43	Complete	917 of 917
single_top_t_chan__nominal	Ilija Vukotic	2023-05-01 22:05:53	2023-05-01 22:13:33	Complete	2506 of 2506
single_top_s_chan__nominal	Ilija Vukotic	2023-05-01 22:05:53	2023-05-01 22:06:43	Complete	114 of 114
wjets__nominal	Ilija Vukotic	2023-05-01 22:05:53	2023-05-01 22:21:25	Complete	10199 of 10199

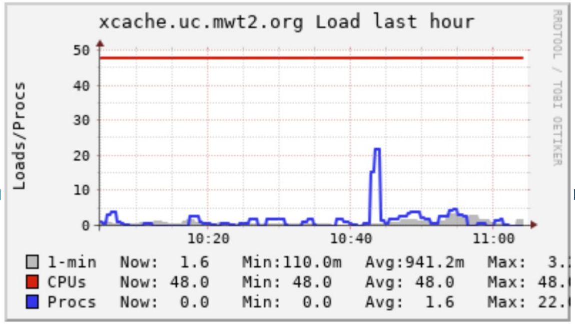
ServiceX Uproot

Never scales up to use all cores.

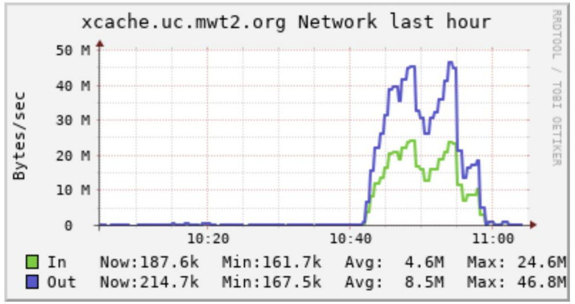
XCache and S3 endpoints basically idle.

Not all requests start right away – some more optimization possible in file path finding.

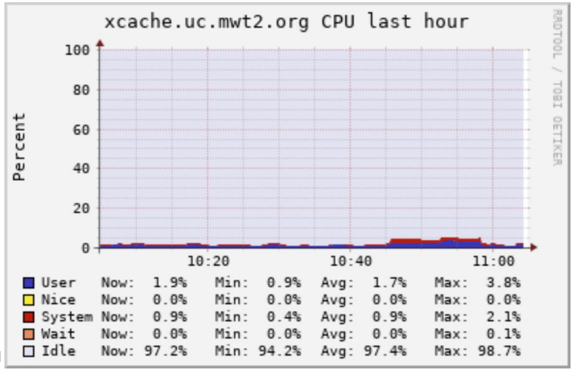
Factor 4–5 speedup probably possible.



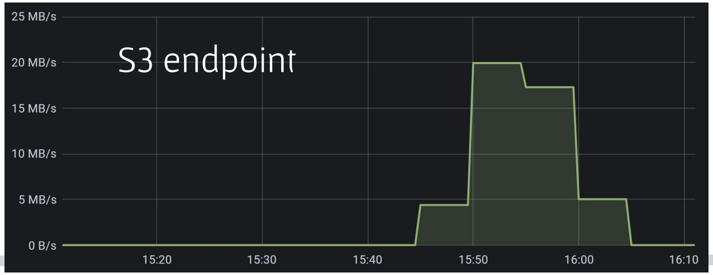
```
▢ 1-min  Now:   1.6   Min:110.0m   Avg:941.2m   Max:  3.
🟥 CPUs   Now:  48.0   Min:  48.0   Avg:  48.0   Max: 48.
🟦 Procs  Now:   0.0   Min:   0.0   Avg:   1.6   Max: 22.
```



```
🟩 In    Now:187.6k   Min:161.7k   Avg:  4.6M   Max: 24.6M
🟦 Out   Now:214.7k   Min:167.5k   Avg:  8.5M   Max: 46.8M
```



```
🟦 User    Now:  1.9%   Min:  0.9%   Avg:  1.7%   Max:  3.8%
🟨 Nice    Now:  0.0%   Min:  0.0%   Avg:  0.0%   Max:  0.0%
🟥 System  Now:  0.9%   Min:  0.4%   Avg:  0.9%   Max:  2.1%
🟧 Wait    Now:  0.0%   Min:  0.0%   Avg:  0.0%   Max:  0.1%
▢ Idle    Now: 97.2%   Min: 94.2%   Avg: 97.4%   Max: 98.7%
```



ServiceX xAOD – single dataset

Reading a single dataset (30885 files ~30TB).

Single Jet collection, simple cut selection.

Selection fully cached in xcache.

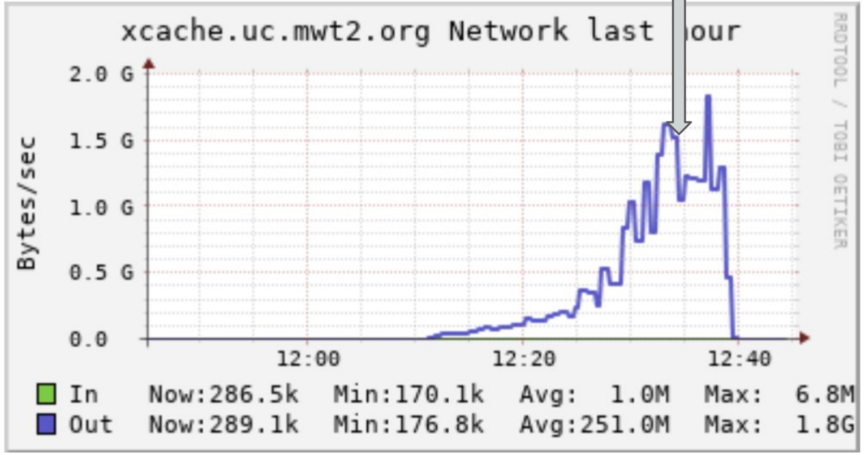
Transformer CPU utilization at 70-80%.

Transformers scaled up to the limit of 750 cores.

We can do much better with startup – write a custom HPA.

Once that's done, the next limiting factor will be number of CPUs available (our xCache can support up to ~3500 transformers like this).

Plato at 750 transformers reached.



🟩 In	Now:286.5k	Min:170.1k	Avg: 1.0M	Max: 6.8M
🟦 Out	Now:289.1k	Min:176.8k	Avg:251.0M	Max: 1.8G

ServiceX xAOD

Period	Files	Size[GB]
K	9312	11704
M	18747	24008
F	5416	5800
I	801	1000
C	9567	12544
L	30885	31006
D	16966	23311
Q	11164	12153
O	14079	17228
Sum:	**116937**	**138755**

Reading a single Jet collection, simple cut selection.

When reading remotely, it takes whole morning, several retries.

When cached CPU utilization at 70-80%.

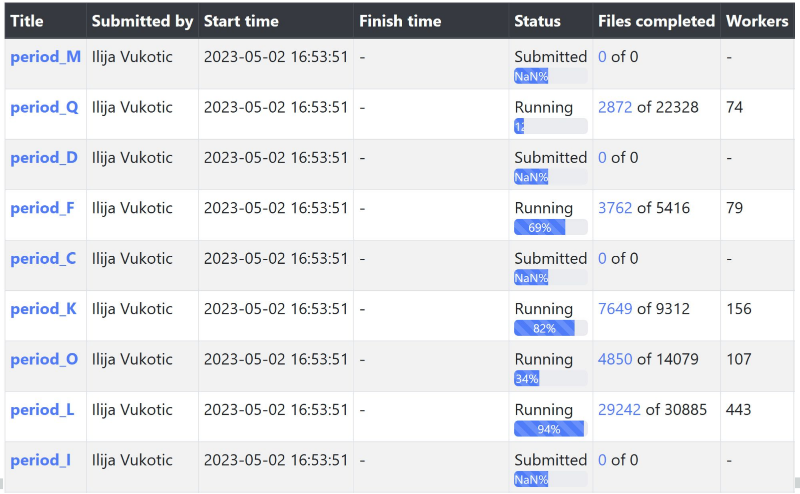
At this scale path lookups take considerable time (some of the lookups expired from the cache – TTL: 1 day).

Processing fully cached data: 1.5 hours.

Took ~40 min to get all the datasets looked up and started.

Scaled up to ~900 cores.

Sometimes transformers would suddenly scale down. It appears not all the input data was in the XCache.



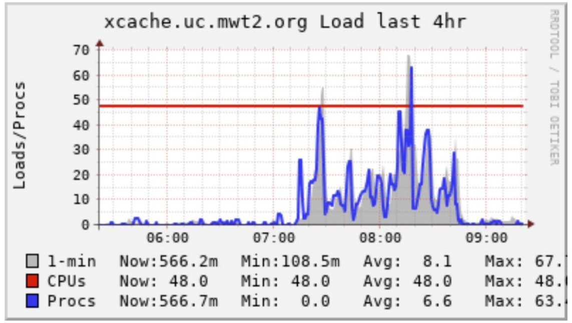
Title	Submitted by	Start time	Finish time	Status	Files completed	Workers
period_M	Ilija Vukotic	2023-05-02 16:53:51	-	Submitted NaN%	0 of 0	-
period_Q	Ilija Vukotic	2023-05-02 16:53:51	-	Running 12	2872 of 22328	74
period_D	Ilija Vukotic	2023-05-02 16:53:51	-	Submitted NaN%	0 of 0	-
period_F	Ilija Vukotic	2023-05-02 16:53:51	-	Running 69%	3762 of 5416	79
period_C	Ilija Vukotic	2023-05-02 16:53:51	-	Submitted NaN%	0 of 0	-
period_K	Ilija Vukotic	2023-05-02 16:53:51	-	Running 82%	7649 of 9312	156
period_O	Ilija Vukotic	2023-05-02 16:53:51	-	Running 34%	4850 of 14079	107
period_L	Ilija Vukotic	2023-05-02 16:53:51	-	Running 94%	29242 of 30885	443
period_I	Ilija Vukotic	2023-05-02 16:53:51	-	Submitted NaN%	0 of 0	-

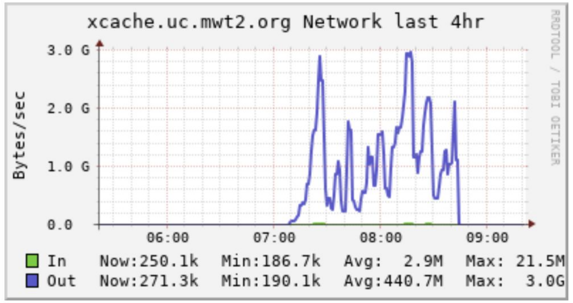
ServiceX xAOD

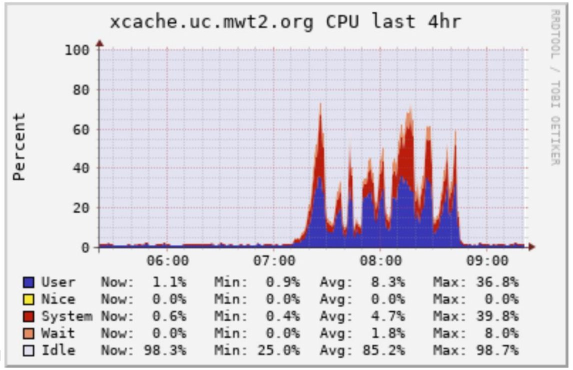
Only a little bit of space to optimize transformers.

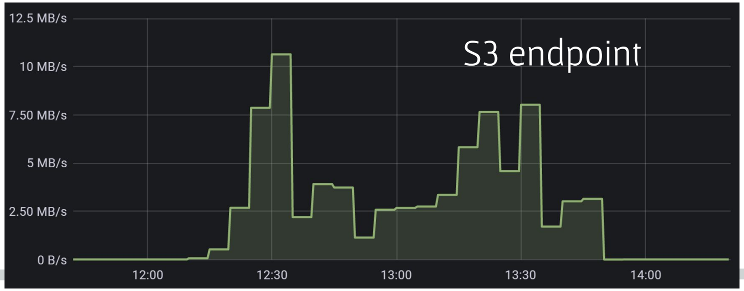
Requests take time to ramp up.

x 3 improvement possible.


xcache.uc.mwt2.org Load last 4hr
Loads/Procs
06:00
07:00
08:00
09:00
1-min Now:566.2m Min:108.5m Avg: 8.1 Max: 67.
CPUs Now: 48.0 Min: 48.0 Avg: 48.0 Max: 48.
Procs Now:566.7m Min: 0.0 Avg: 6.6 Max: 63.
RRDTOOL / TOBI OETIKER


xcache.uc.mwt2.org Network last 4hr
Bytes/sec
3.0 G
2.0 G
1.0 G
06:00
07:00
08:00
09:00
In Now:250.1k Min:186.7k Avg: 2.9M Max: 21.5M
Out Now:271.3k Min:190.1k Avg:440.7M Max: 3.0G
RRDTOOL / TOBI OETIKER


xcache.uc.mwt2.org CPU last 4hr
Percent
06:00
07:00
08:00
09:00
User Now: 1.1% Min: 0.9% Avg: 8.3% Max: 36.8%
Nice Now: 0.0% Min: 0.0% Avg: 0.0% Max: 0.0%
System Now: 0.6% Min: 0.4% Avg: 4.7% Max: 39.8%
Wait Now: 0.0% Min: 0.0% Avg: 1.8% Max: 8.0%
Idle Now: 98.3% Min: 25.0% Avg: 85.2% Max: 98.7%
RRDTOOL / TOBI OETIKER


S3 endpoint
12.5 MB/s
10 MB/s
7.50 MB/s
5 MB/s
2.50 MB/s
0 B/s
12:00
12:30
13:00
13:30
14:00