

Tape Carousel Experience

NeIC NT1 Manager
Mattias Wadenstein
<maswan@ndgf.org>

2018-11-06
NDGF All Hands
Umeå

Overview

- Carousel theory
- How did it go
- Future plans



Tape carousel

- Tape is much cheaper than disk per PB
 - Factor 3-4, depending on how many tape drives you need etc
- LHC experiments (especially ATLAS and CMS) will need lots more storage space for HL-LHC
- If you move the bulk of storage to tape and just keep a sliding window of data on disk for analysis you can maybe save lots of \$£€
 - But in order to science properly, you don't want to have to wait too long for a particular dataset to come online
 - So the sliding window needs to slide reasonably fast

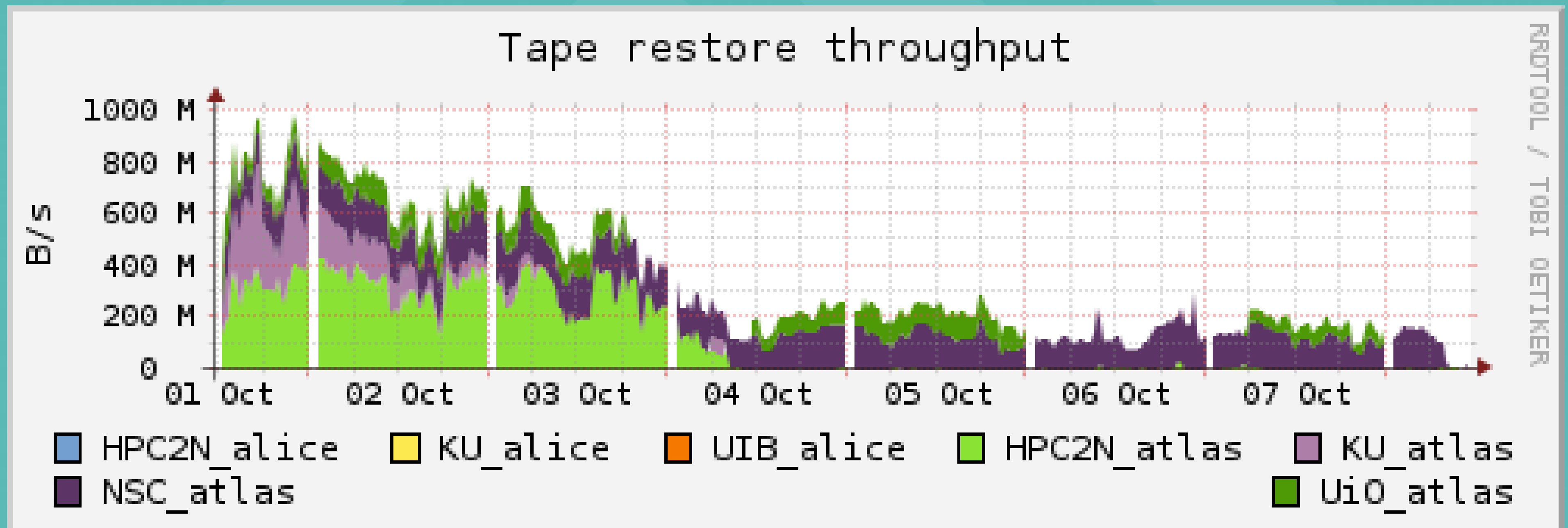


Tape carousel test

- ATLAS decided to test the capability of tier-1 tape systems by recalling a reasonably large set of datasets from tape
- In a carousel environment this would be the weekly/biweekly recall set to keep the sliding window sliding



Results of a ~100TB restore



dCache issues to deal with

- Tape read pool oversubscription
 - Silly to split the reads up in tape read pool sized chunks, loss of efficiency
 - Also sometimes the sweeper got the files before p2p, leading to rerestores
 - Discussion has been lifted with dCache.org team
- Internal p2p transfers clogged up by a lack of internal bandwidth
 - File transfers to: IJS, UiO, KU



What do we need from sites?

- Been trying to get this answer from ATLAS
- My opinion: If all our tape libraries behaved as HPC2N and KU we'd meet any reasonable demand on a small T1



Tape write performance

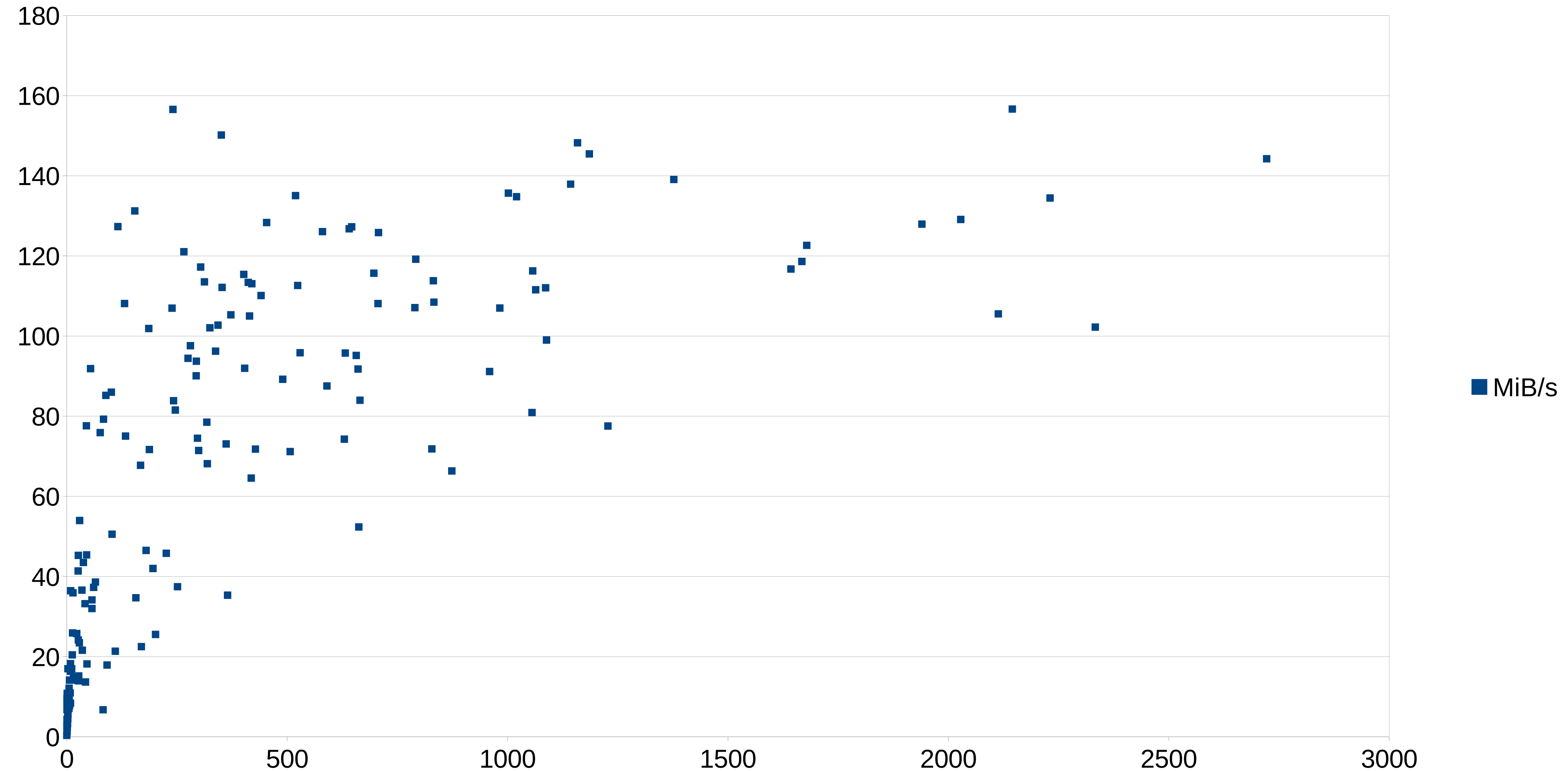
- Spinning disks are starting to get problematic to keep up with tape drives
- SSDs are unfortunately expensive if you need large space
- We'd like a large space for handling a few hours (pref a few days) of incoming data before telling clients "stop!"
- Could we use free space on atlas_disk as overflow space? Other ideas?



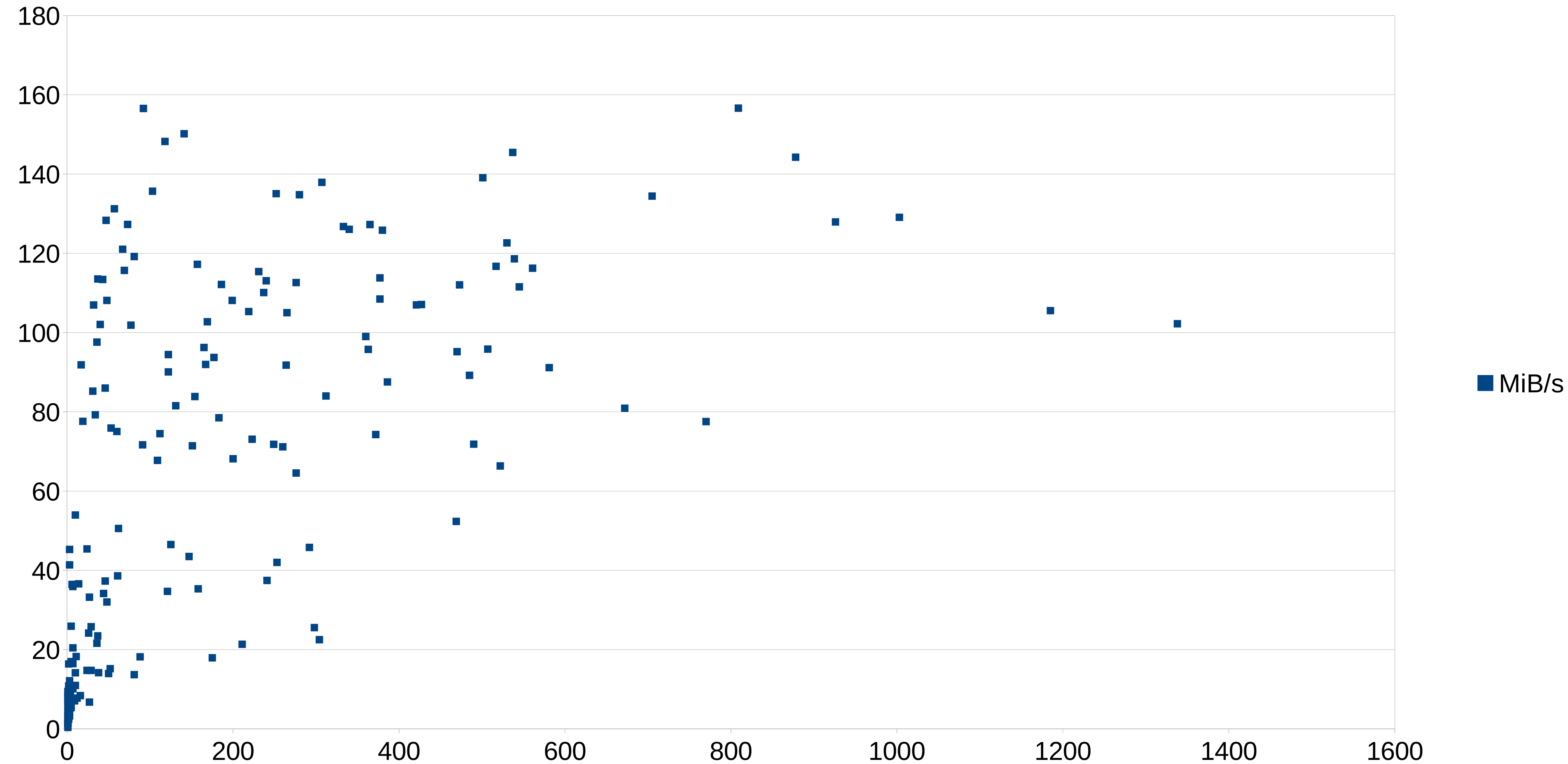


Questions?

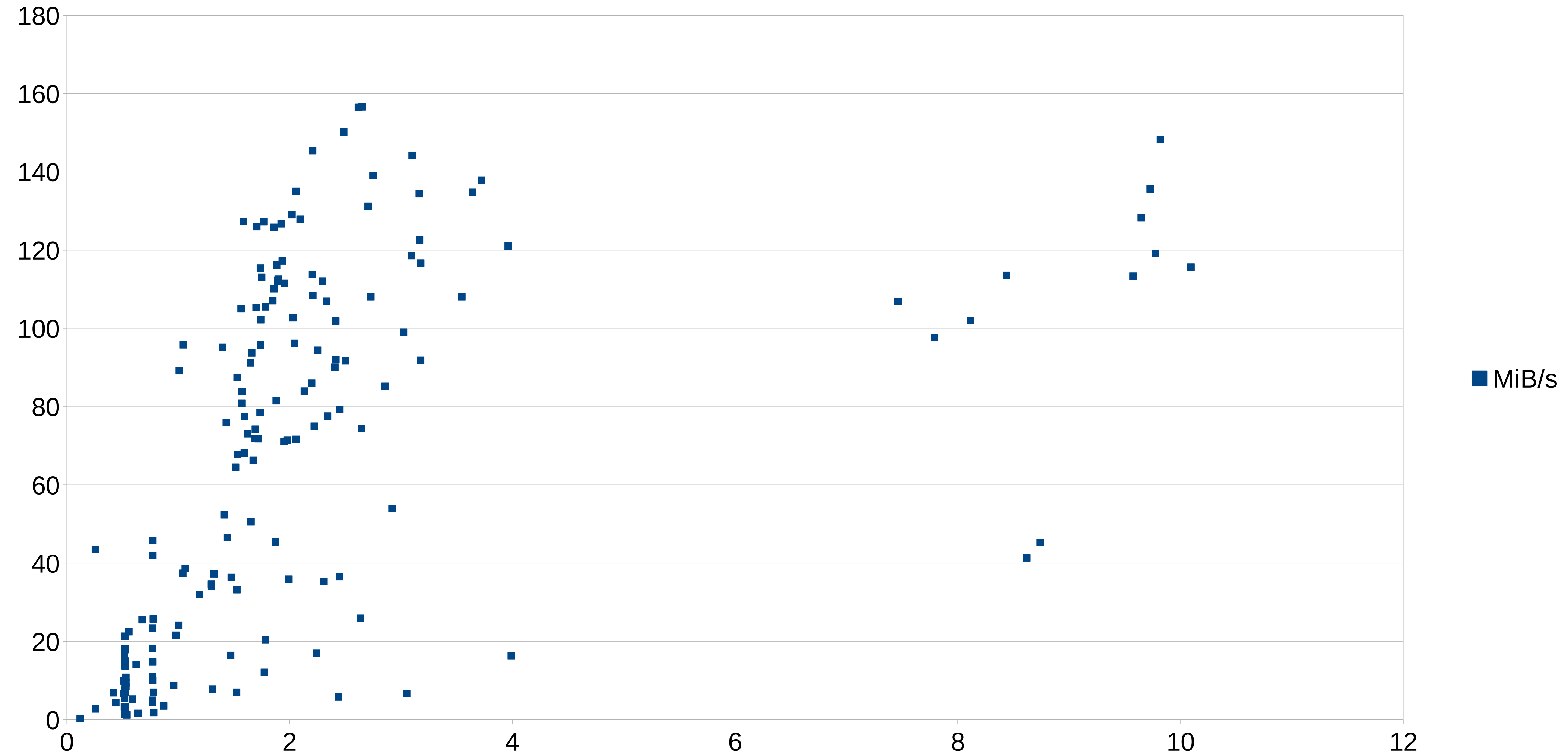
HPC2N GiB



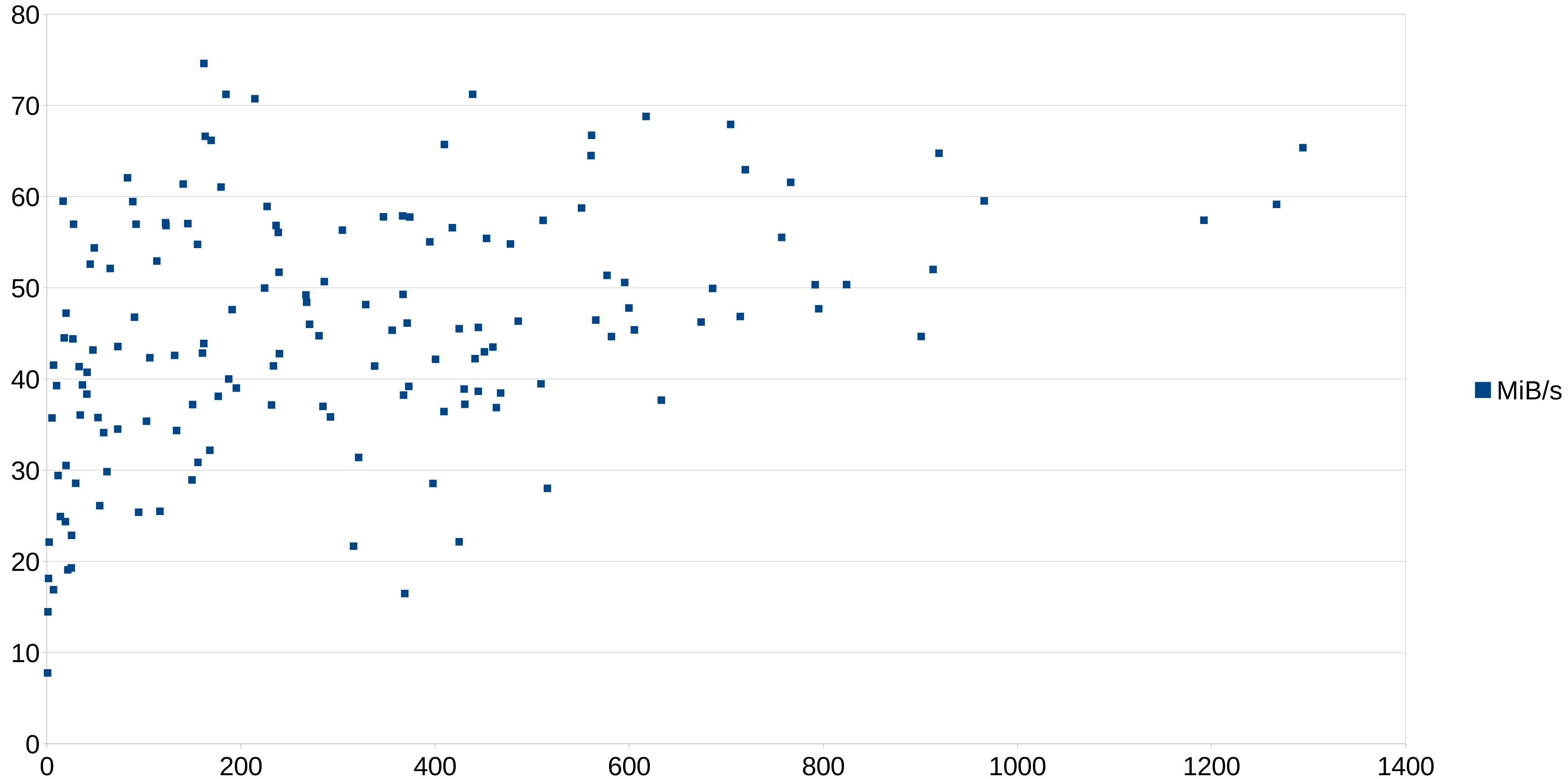
HPC2N nofiles



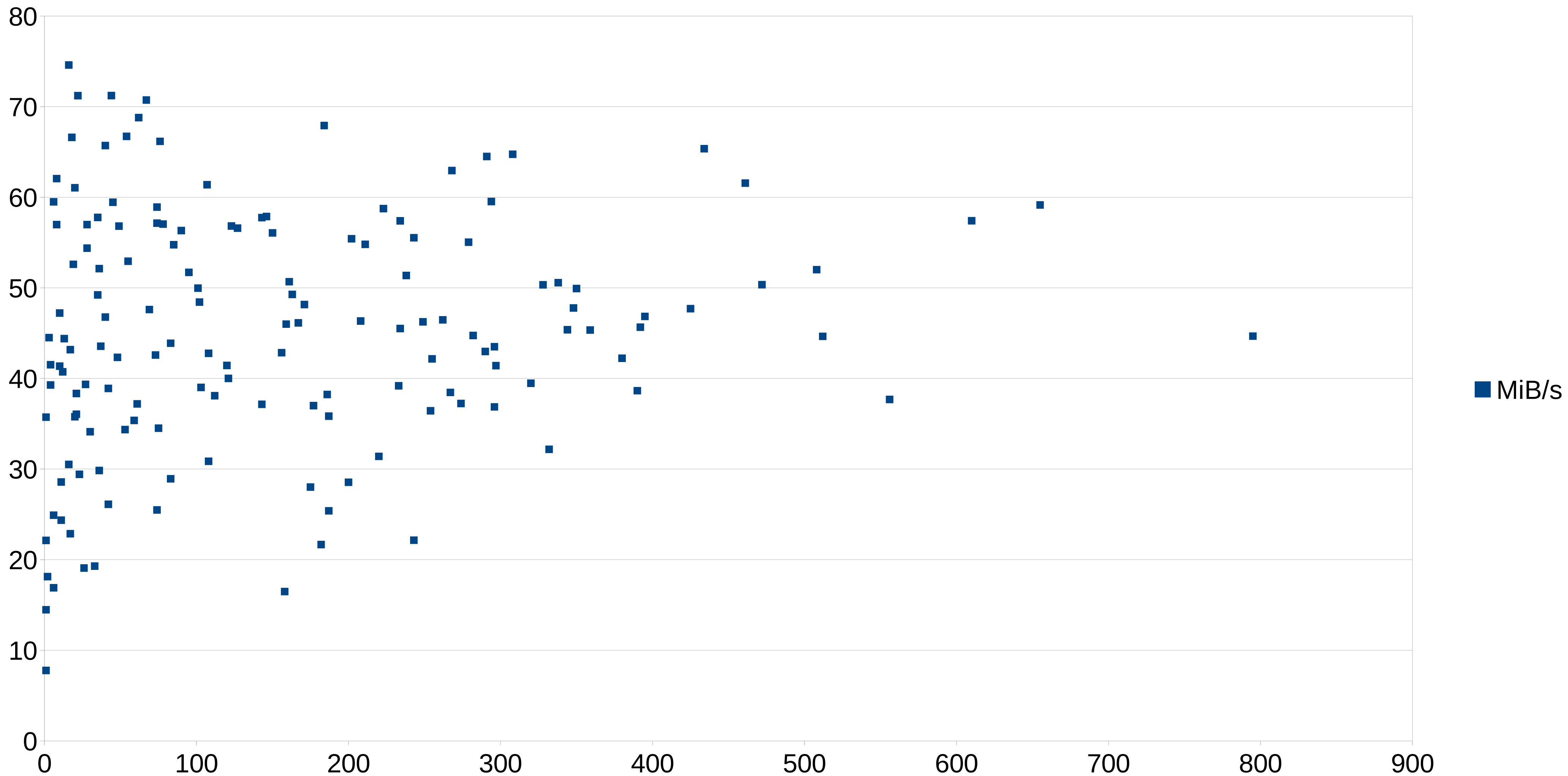
HPC2N avg filesize



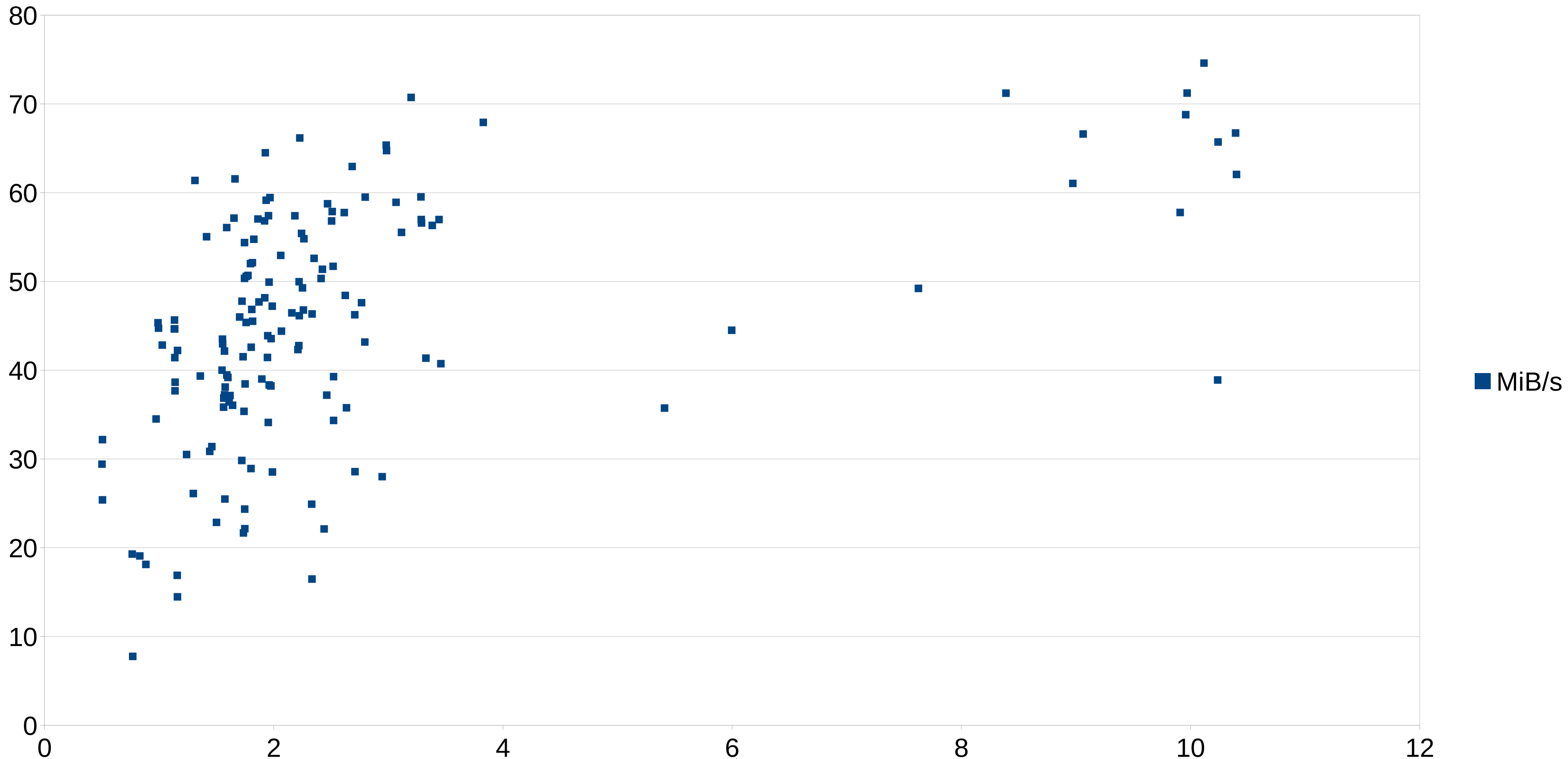
NSC GiB



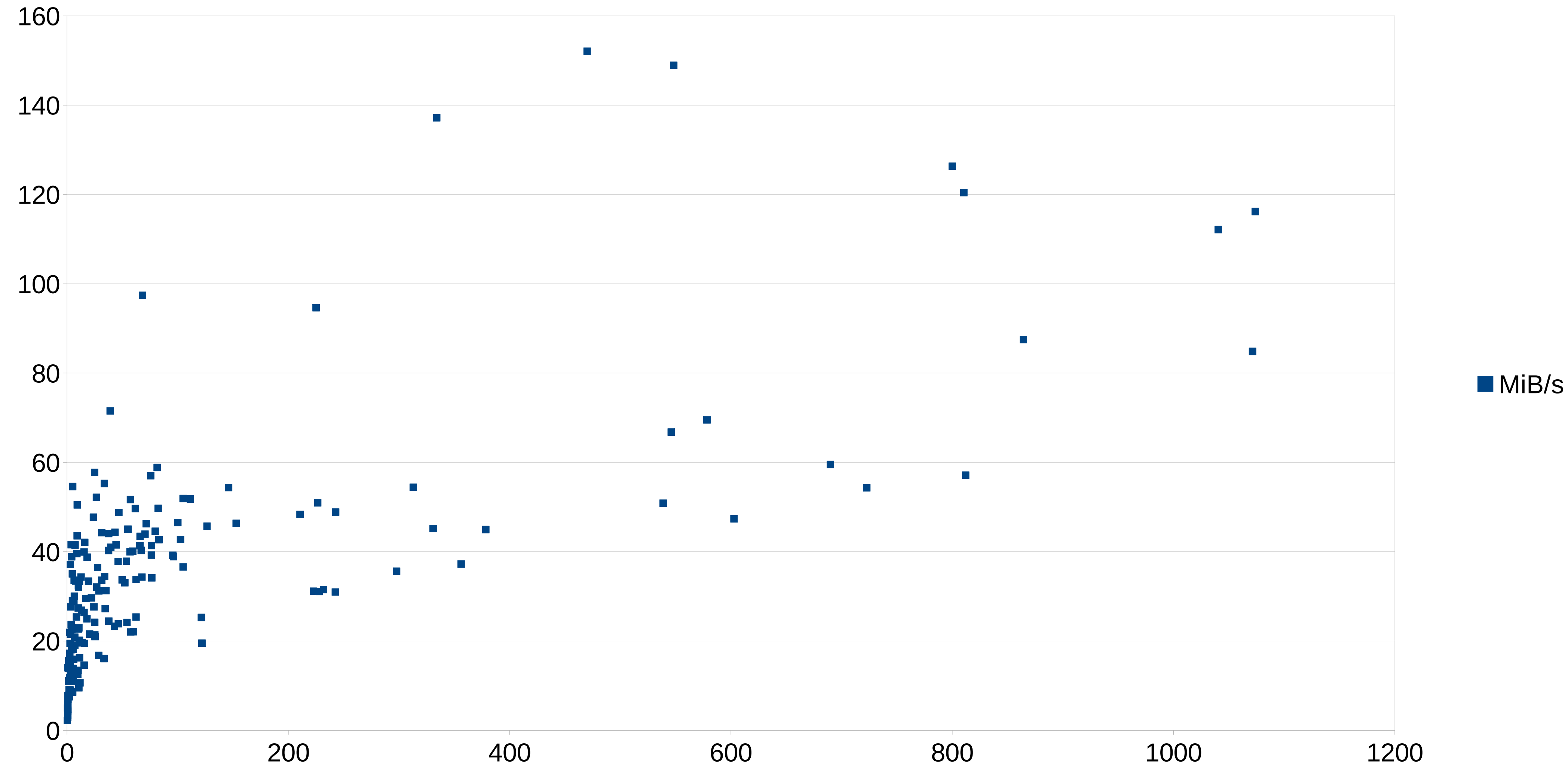
NSC nofiles



NSC avg filesize



KU GiB



KU nofiles

