

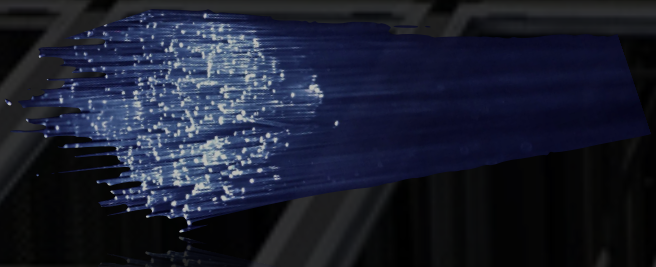
7th March 2022

CERN

CEBN

EOS monitoring of finished transfers

io stat improvements



Dr. Jaroslav Guenther
on behalf of EOS PDS team
(CERN IT-ST-PDS)
(CEBN IT-ST-PDS)

Transfer monitoring

Storage node reports:

- > reports sent to EOS MGM for all finished transfers
- > transfer metrics and metadata
(activity type, application, domain etc.)



Report volume (2020/21):

- > ~2000M transfers a year / instance
(ATLAS, CMS, PUBLIC)
- > ~30% writes (60Hz)
- > ~0.5 TB/year/instance

Report monitoring strategies

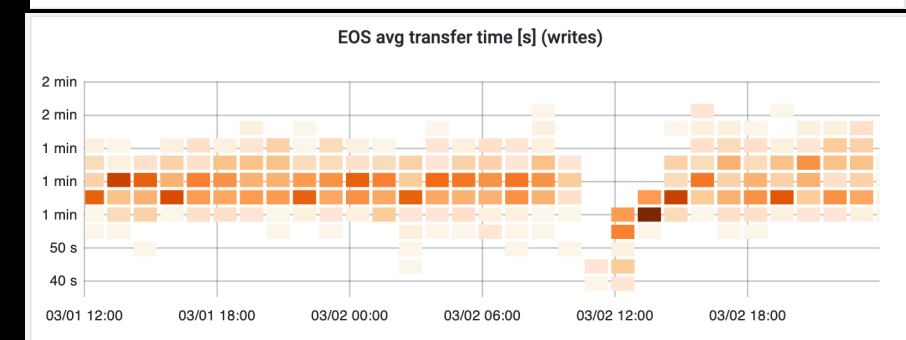
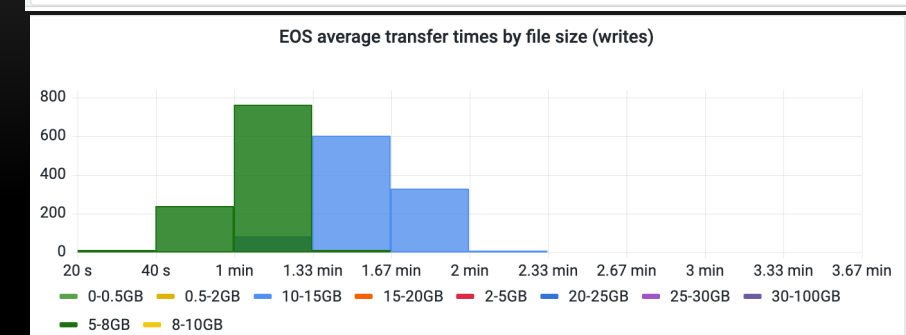
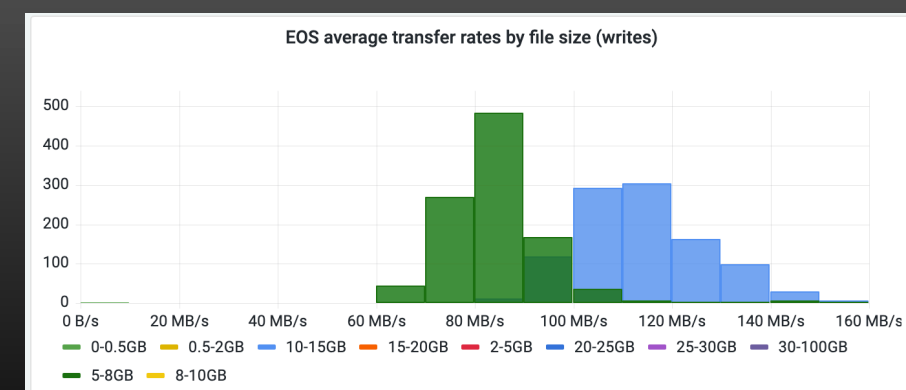


1) Analyse offline:

- > e.g. ship to CERNBox + ROOT + Jupiter Notebook
- > automatic plotting possible to implement

2) Ship reports to monitoring service

- > e.g. fluent-bit → Elasticsearch → Grafana
- > short retention period (1 month now)
- > work on post-processing
- > dynamic (but rather indicative than precise)



3) Extract strategic data directly from MGM

- > only important use-cases
- > less data to ship
- > less work on post-processing
- > eos io stat
- ship to online monitoring platform

io	application	1min	5min	1h	24h	sum
out	eoscp	0	0	0	11.93 M	1.02 G
out	eos/gridftp	707.83 K	3.73 M	3.31 G	112.80 G	2.39 T
out	cmst0	5.48 G	29.46 G	301.89 G	6.62 T	14.67 T
out	fuse::lxfplus	415.21 M	45.15 G	690.94 G	20.48 T	78.82 T
in	eoscp	79.23 M	1.00 G	19.07 G	423.17 G	971.37 G
in	eos/gridftp	484.87 M	1.31 G	7.75 G	872.41 G	1.65 T
in	cmst0	11.57 G	76.17 G	321.83 G	8.30 T	19.87 T
in	fuse::lxfplus	122.41 M	668.94 M	42.97 G	1.76 T	2.57 T

eos io stat options:

- > -a : user/group
- > -x : application (tagged traffic)
- > -d : domain
- > -1 : global activity

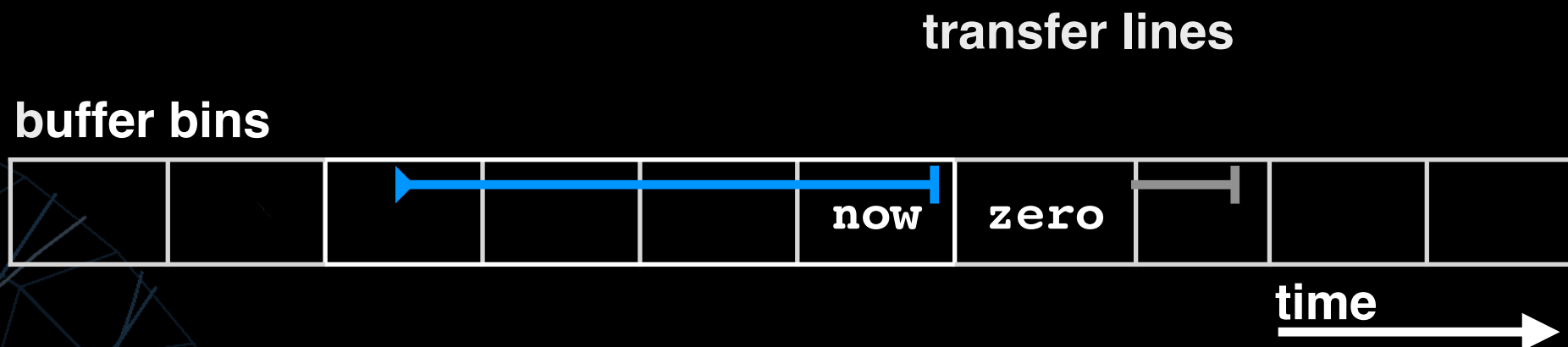
Sums all bytes of transfers:

- > finished in last : 1min 5min 1h 24h
- > finished since EOS instance restart till now : sum

io	application	1min	5min	1h	24h	sum
out	eoscp	0	0	0	11.93 M	1.02 G
out	eos/gridftp	707.83 K	3.73 M	3.31 G	112.80 G	2.39 T
out	cmst0	5.48 G	29.46 G	301.89 G	6.62 T	14.67 T
out	fuse::lxplus	415.21 M	45.15 G	690.94 G	20.48 T	78.82 T
in	eoscp	79.23 M	1.00 G	19.07 G	423.17 G	971.37 G
in	eos/gridftp	484.87 M	1.31 G	7.75 G	872.41 G	1.65 T
in	cmst0	11.57 G	76.17 G	321.83 G	8.30 T	19.87 T
in	fuse::lxplus	122.41 M	668.94 M	42.97 G	1.76 T	2.57 T

Implementation:

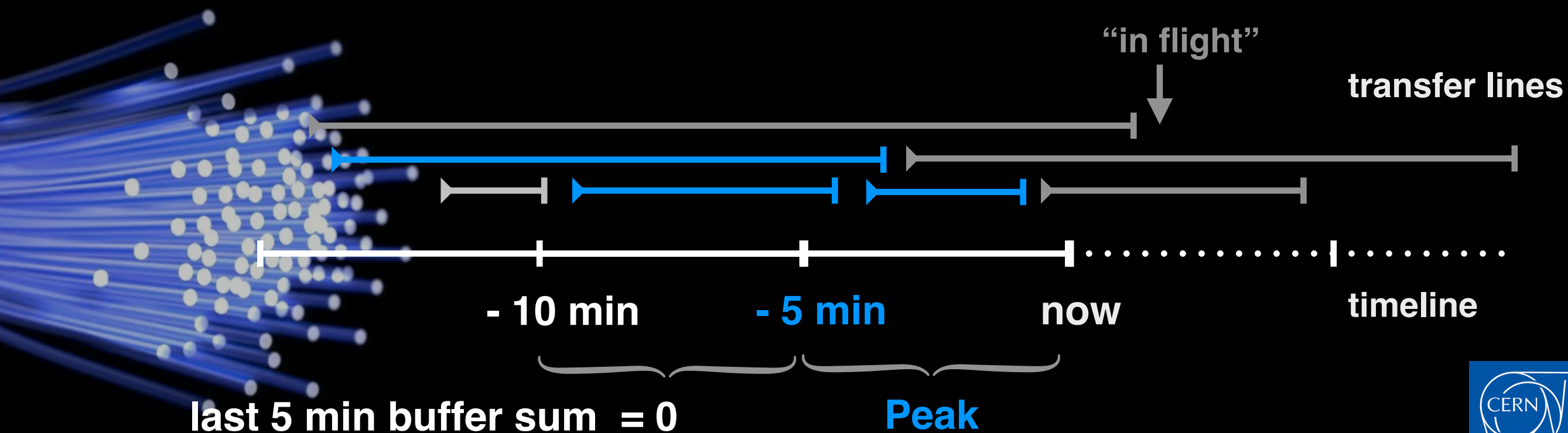
- > 4 circular buffers for the 4 intervals
- > each buffer has 60 bins (widths 1s 5s 1min 1440min)
- > current timestamp → bin now
- > **transfer duration** → buffer bins to fill relative to now
- > transfer volume distributed to bins accordingly
- > bin zero is zeroed each 512 ms



IO stat challenges

Issues:

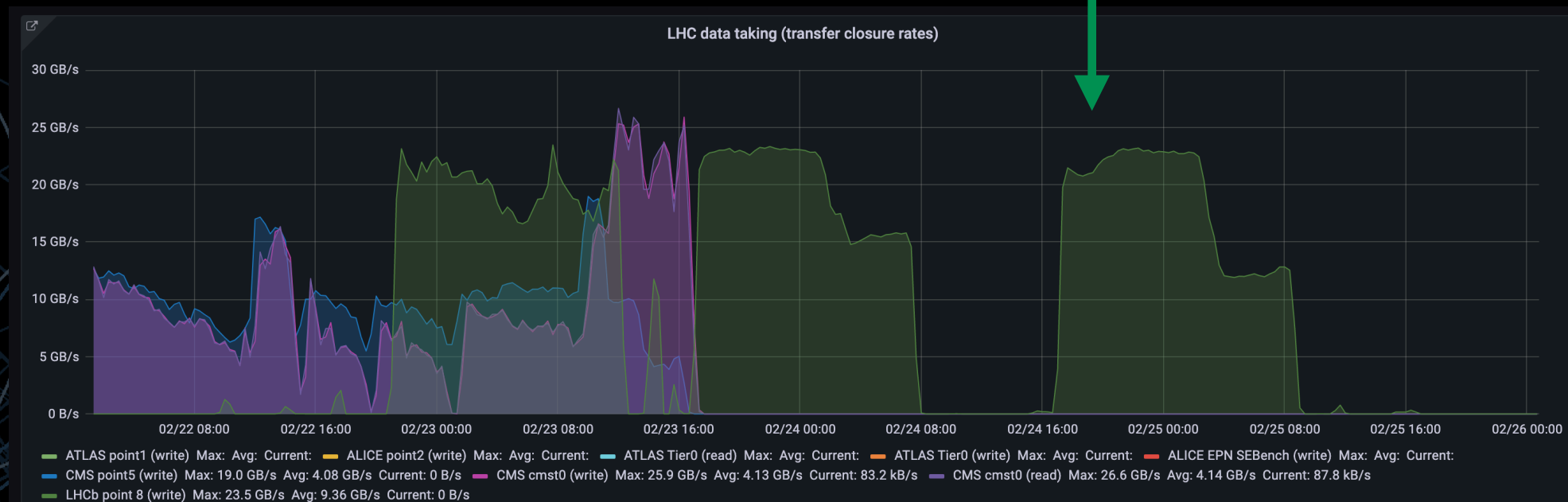
- > bin zero → buffer sum **underestimated** (most cases 1.6% only)
- > if XFR duration > buffer interval
 - all data contained in bins (circulating until data in)
- > transfer reports arriving late → **ignored**
(XFR report arrived now, but transfer finished 5min ago, will be recorded only in 1h and 24h buffers)
- > XFR “in flight” → **not visible**



IO stat monitoring

What we have:

- > missing all transfers “in flight”
- > “peaks and valleys” in io stat metric timelines
- > intervals → not reliable, but **reasonable** to use **IF**:
 - * no late reports
 - * XFR length \ll metric interval
- > total sum **correct** → all transferred data volume
- derivative of sum **timeline** → **closure rate * XFR volume**
(XFR closure rate not monitored directly yet)



IO stat next steps

What we wish to have:

- > data rate to compare with network rates at any moment in time
(constant rate over xrf duration assumed)
- > finished transfer count/rate
- > transfer size



New design:

- > similarly distribute data transfers into “timeline”
- > long circular buffer (24h in 86 400 bins)
- > cut transfer if lasting > buffer interval (only XFR > 24h)
- > zero bins only when needed
- > report can come anytime

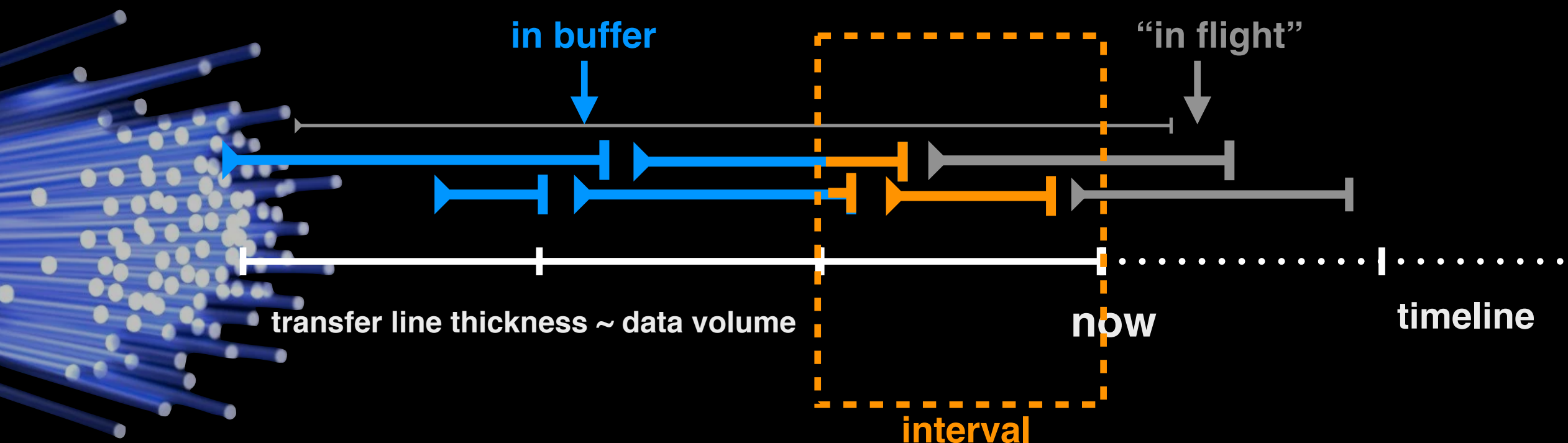


New IO stat metrics

Interval sums:

- > get bytes transferred only in given interval
- > the more interval shifter to the past
the less transfers “in flight”

What control delay is OK ?



define acceptable error
e.g. < 5%

minimal delay = time to get 95%
of data transferred
(on PDS instances = 15min)
(sampling transfer distribution)

control delay
30 min



Monitoring with control delay

Monitor data rate 'without' XFR in flight:

> calculate minimal delay every 5 min

> timelines of:

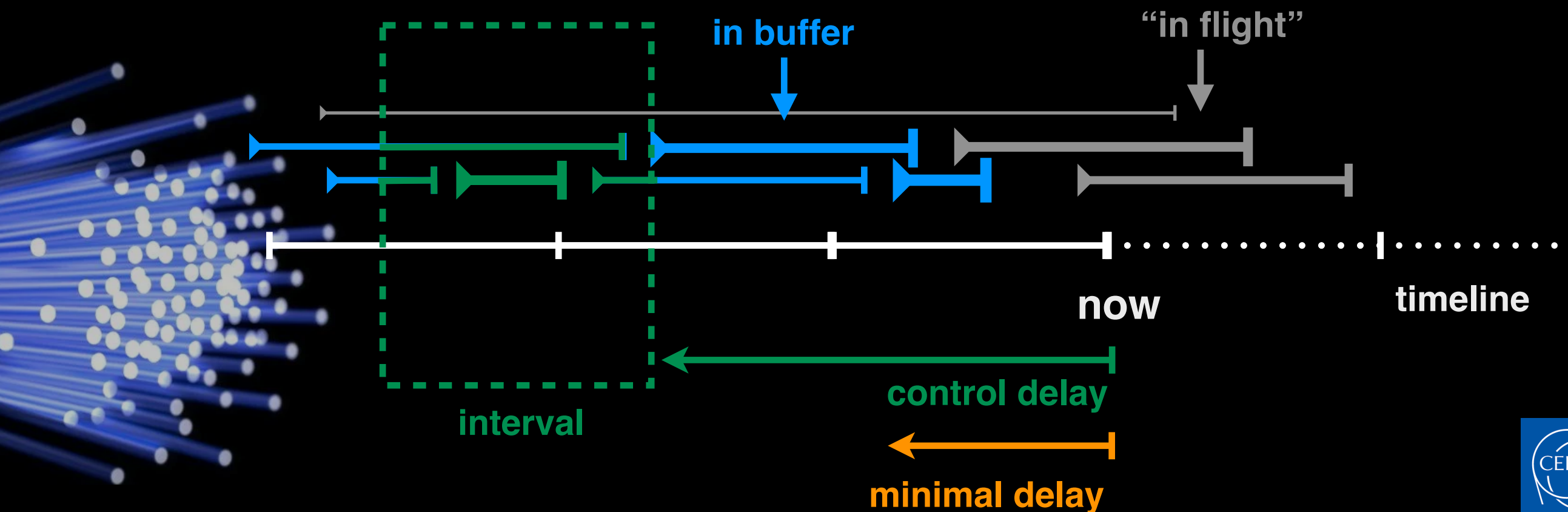
→ avg data rate in **interval**

→ **minimal delay** (95%, 99%, 100%)

> If minimal delay > control delay

→ invalidates data rate plot

→ control delay needs to be increased !



transfer line thickness ~ data volume

New IO stat example (hands on)

```
[root@jaro-dev2 build-with-ninja]# for i in {1..5}; do echo $i" minute"; date;
> eos cp test5G.file /eos/jaro/newdir/test5G.file; sleep 60; done;
1 minute
Mon Mar  7 03:45:07 CET 2022
[eoscp] test5G.file          Total 4768.37 MB  |=====| 100.00 % [480.7 MB/s]
[eos-cp] copied 1/1 files and 5.00 GB in 10.44 seconds with 478.86 MB/s
2 minute
Mon Mar  7 03:46:18 CET 2022
[eoscp] test5G.file          Total 4768.37 MB  |=====| 100.00 % [361.7 MB/s]
[eos-cp] copied 1/1 files and 5.00 GB in 13.86 seconds with 360.65 MB/s
3 minute
Mon Mar  7 03:47:32 CET 2022
[eoscp] test5G.file          Total 4768.37 MB  |=====| 100.00 % [720.0 MB/s]
[eos-cp] copied 1/1 files and 5.00 GB in 7.00 seconds with 714.05 MB/s
4 minute
Mon Mar  7 03:48:43 CET 2022
[eoscp] test5G.file          Total 4768.37 MB  |=====| 100.00 % [406.6 MB/s]
[eos-cp] copied 1/1 files and 5.00 GB in 12.33 seconds with 405.40 MB/s
5 minute
Mon Mar  7 03:49:59 CET 2022
[eoscp] test5G.file          Total 4768.37 MB  |=====| 100.00 % [688.2 MB/s]
[eos-cp] copied 1/1 files and 5.00 GB in 7.30 seconds with 684.62 MB/s
```

> 04:10:56 we go 20m 50s (1250 sec) to the past and fetch 7 seconds
[03:50:06 - 03:49:59]

```
[root@jaro-dev2 ~]# date; eos io stat -x --sa 1250 --si 7
Mon Mar  7 04:10:56 CET 2022
```

io	application	data in interval	avg rate [B/s]
in	eoscp	5000000000.	714285696.



New IO stat example (hands on)

```
[root@jaro-dev2 build-with-ninja]# for i in {1..5}; do echo $i" minute"; date;
> eos cp test5G.file /eos/jaro/newdir/test5G.file; sleep 60; done;
1 minute
Mon Mar 7 03:45:07 CET 2022
[leosp] test5G.file Total 4768.37 MB |=====| 100.00 % [480.7 MB/s]
[eos-cp] copied 1/1 files and 5.00 GB in 10.44 seconds with 478.86 MB/s
```

```
[root@jaro-dev2 ~]# date; eos io stat -x
Mon Mar 7 03:45:35 CET 2022
```

➤ Sum of bytes transferred in last 1m/5m/1h/24h and total sum:

io	application	1min	5min	1h	24h	sum
in	eoscp	5.00 G	5.00 G	5.00 G	5.00 G	5.00 G

➤ Transfer (tf) sample info every 5 min: tf time for 90/95/99/100% of data, max tf time (last 24h), average tf size, tf count.

io	application	90% [s]	95% [s]	99% [s]	100% [s]	max [s]	avg tf size	tf #	sample time
in	eoscp	0	0	0	0	10	0	0	Mon Mar 7 03:45:18 2022

**started collecting transfers
for sampling
after first report arrived**

```
[root@jaro-dev2 ~]# date; eos io stat -x;
Mon Mar 7 03:50:33 CET 2022
```

➤ Sum of bytes transferred in last 1m/5m/1h/24h and total sum:

io	application	1min	5min	1h	24h	sum
in	eoscp	5.00 G	20.00 G	25.00 G	25.00 G	25.00 G

➤ Transfer (tf) sample info every 5 min: tf time for 90/95/99/100% of data, max tf time (last 24h), average tf size, tf count.

io	application	90% [s]	95% [s]	99% [s]	100% [s]	max [s]	avg tf size	tf #	sample time
in	eoscp	11	12	14	14	14	5.00 G	5	Mon Mar 7 03:50:19 2022

**5 min distribution integrated
and sampling starts again**



Thank you



New IO stat example (hands on)

```
[root@iarc-dev2 ~]# date: eos io stat -x;
```

```
Mon Mar 7 04:04:13 CET 2022
```

```
└> Sum of bytes transferred in last 1m/5m/1h/24h and total sum:
```

io	application	1min	5min	1h	24h	sum
in	eoscp	0	0	25.00 G	25.00 G	25.00 G

```
└> Transfer (tf) sample info every 5 min: tf time for 90/95/99/100% of data, max tf time (last 24h), average tf size, tf count.
```

io	application	90% [s]	95% [s]	99% [s]	100% [s]	max [s]	avg tf size	tf #	sample time
in	eoscp	0	0	0	0	14	0	0	Mon Mar 7 04:00:21 2022

