HS23 from CentOS7 to AlmaLinux9

Evgeny Stambulchik

Weizmann Institute of Science, Rehovot 7610001, Israel

HEPiX Benchmarking WG
June 19, 2024

Introduction

- In preparation to transition from CentOS7 to AlmaLinux9, benchmarks on a new hardware were run & compared between the OS'es
- Initial results were both surprising and confusing (the details are in the GGUS ticket 166741)

Hardware

Two identical systems:

- Motherboard: ASUS ESC4000-E11
- CPU: 2 $\times$ Xeon Gold 6530 @ 2.10 GHz (64 HW cores in total)
- Hyperthreading: enabled
- RAM: 512 GB
- Disk: 512 GB SATA and/or 2 TB NVMe[1]
- GPU: 4 $\times$ NVidia A6000 (not used for the benchmarking)

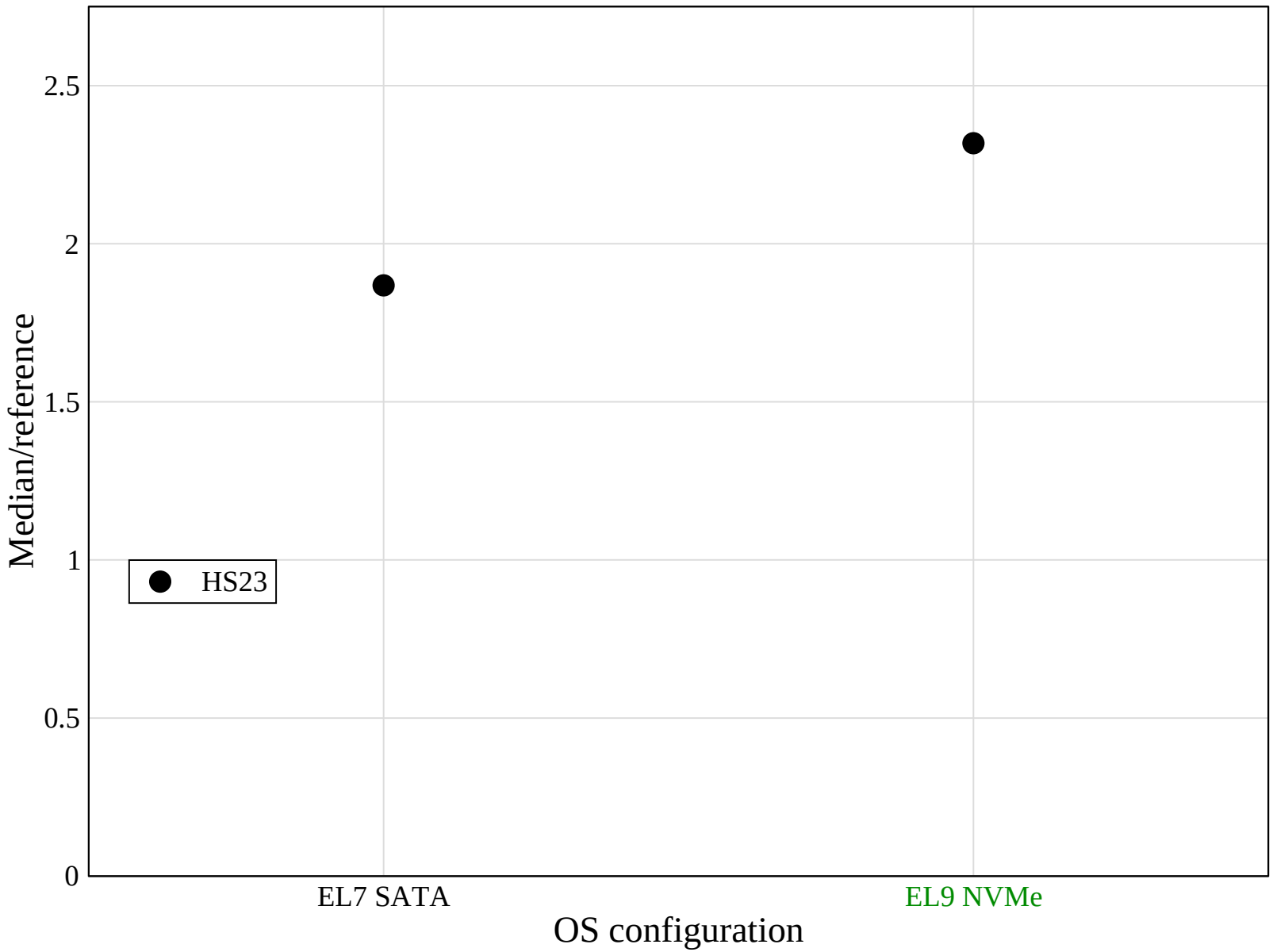
[1] Not supported by CentOS7

System setup – notable differences

	CentOS7	AlmaLinux9
Kernel	3.10	5.14 PREEMPT_DYNAMIC [2]
Local disk	SATA SSD	NVMe *or* SATA SSD
File system	Primary/Ext4	LVM/XFS

[2] `dmesg | grep Preempt`
`Dynamic Preempt: voluntary`

First results



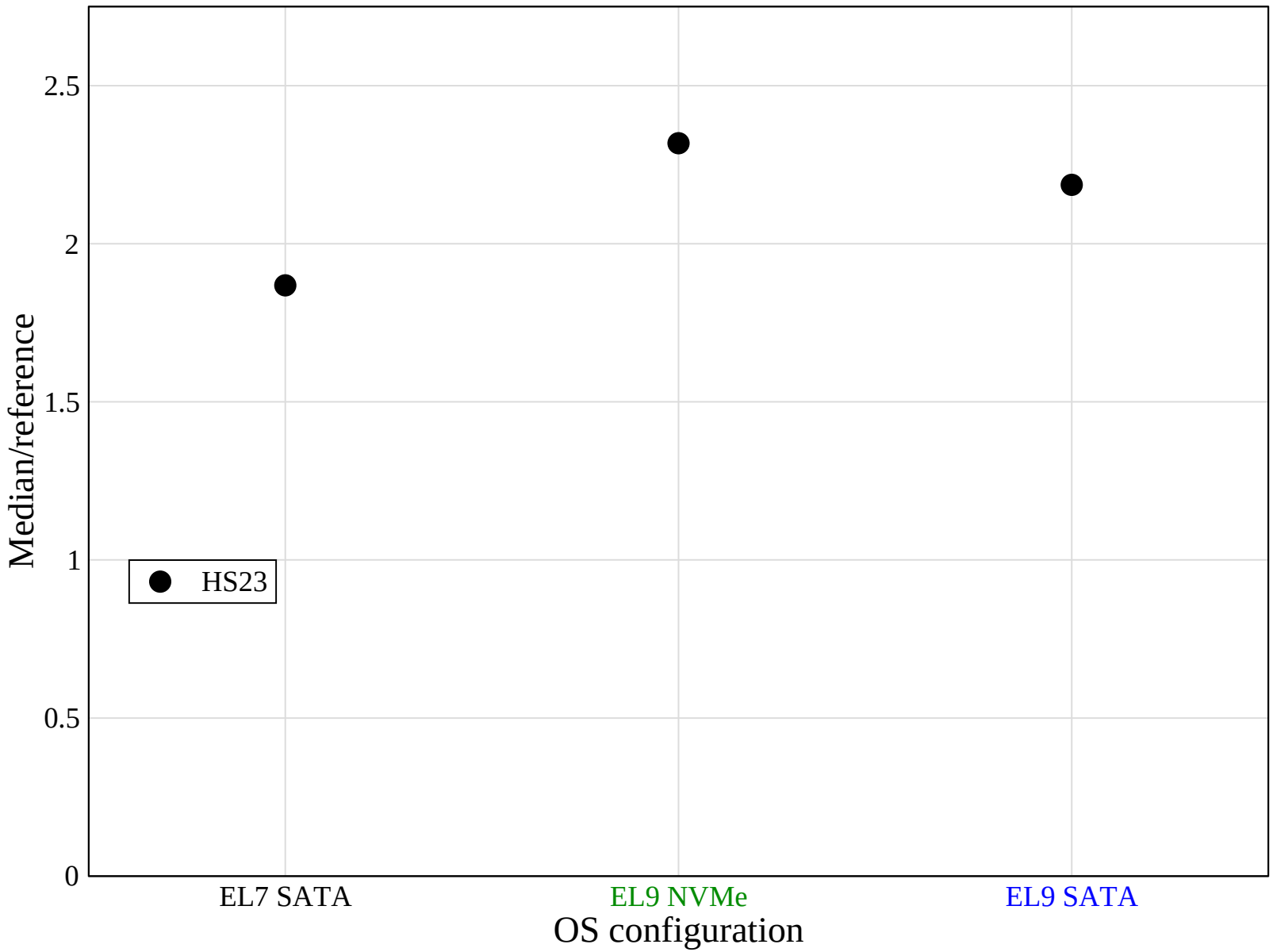
First results

- ☛ A 25% increase; could it be due to the disk (SATA vs NVMe)?

First results

- A 25% increase; could it be due to the disk (SATA vs NVMe)?
- To find out, AlmaLinux9 was installed on the same type of SATA disk.

First results



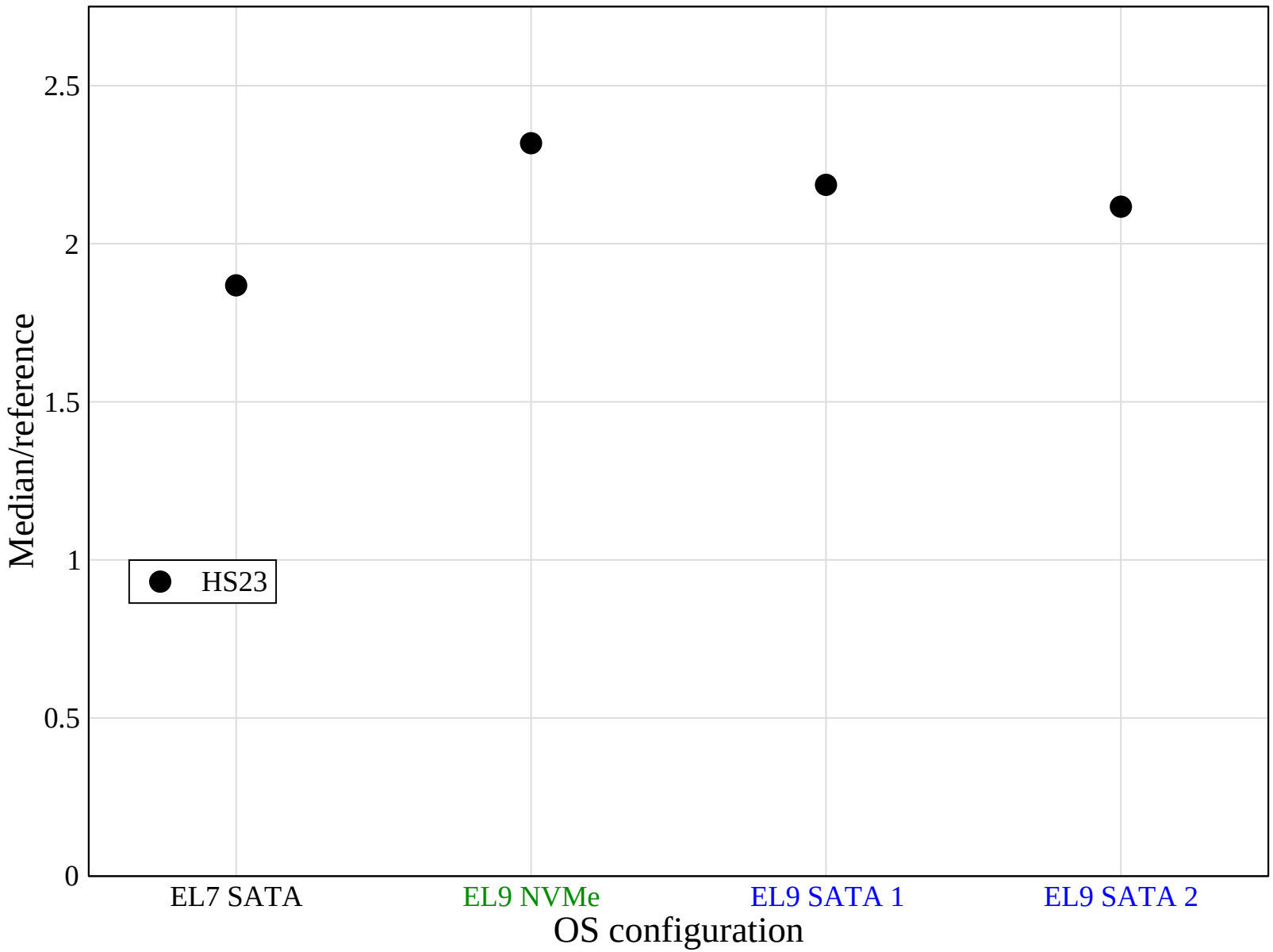
First results

- Some of the difference can definitely be attributed to the storage; but not all

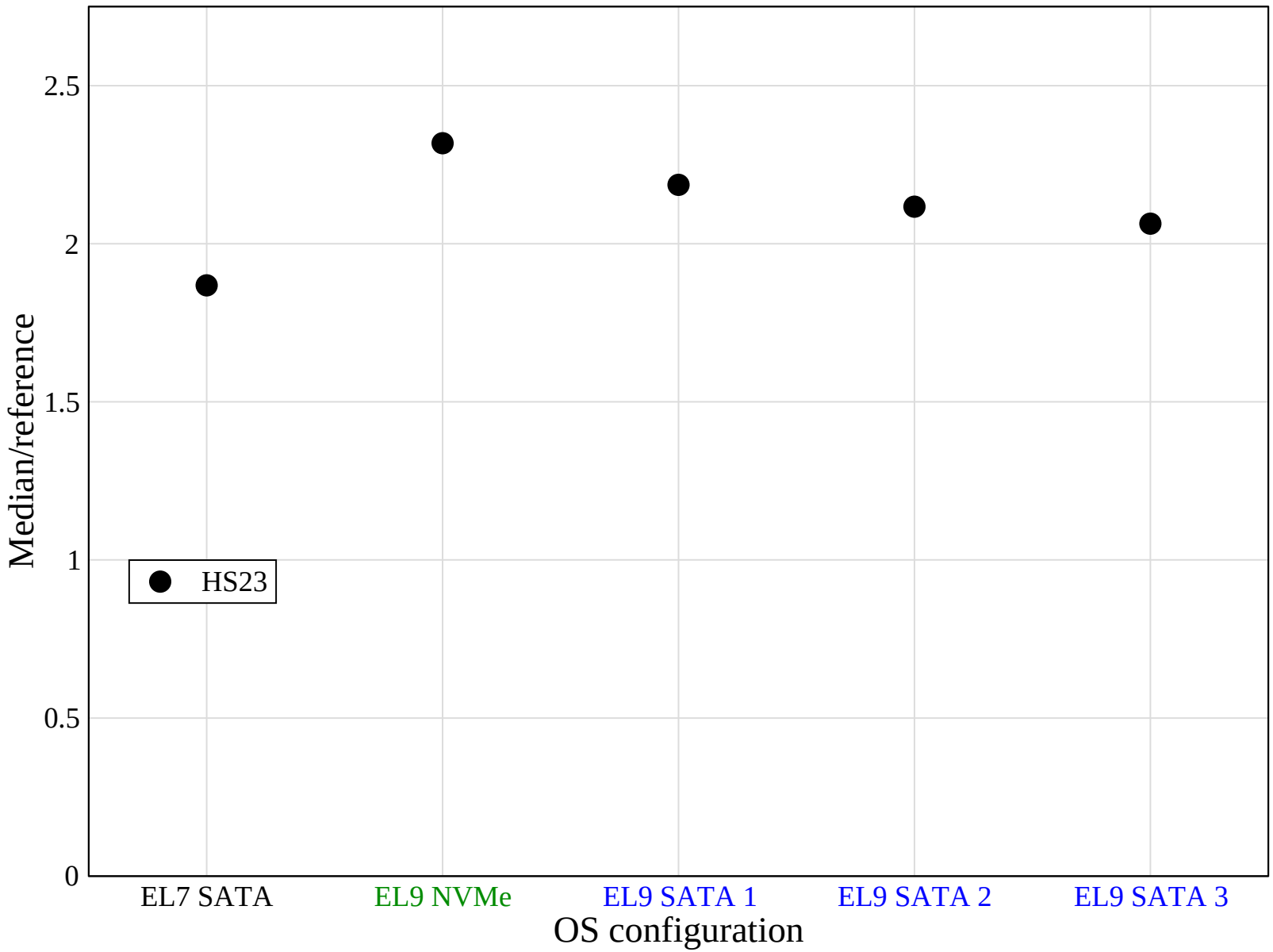
First results

- Some of the difference can definitely be attributed to the storage; but not all
- Furthermore, a strange phnomenon was observed...

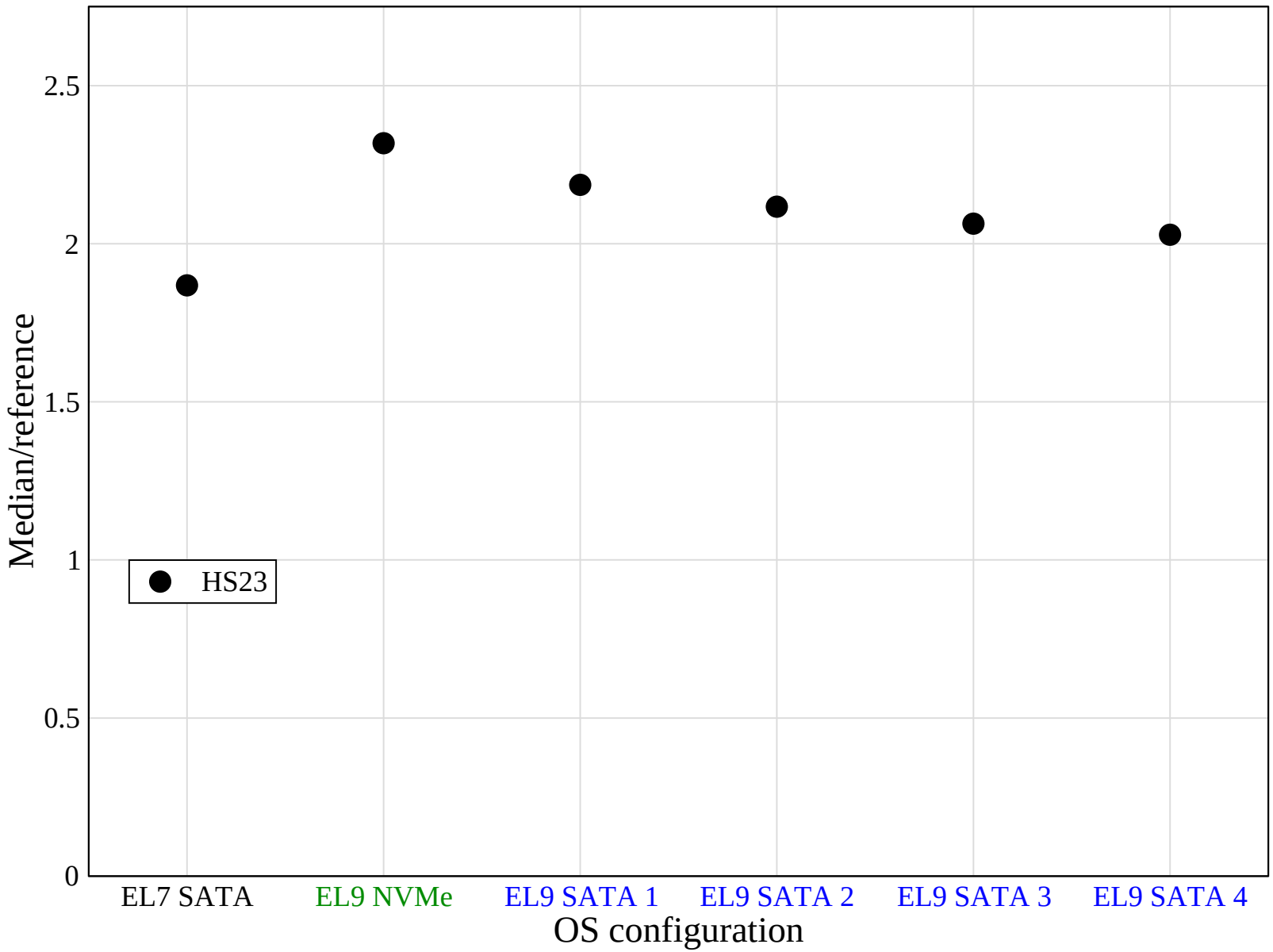
First results



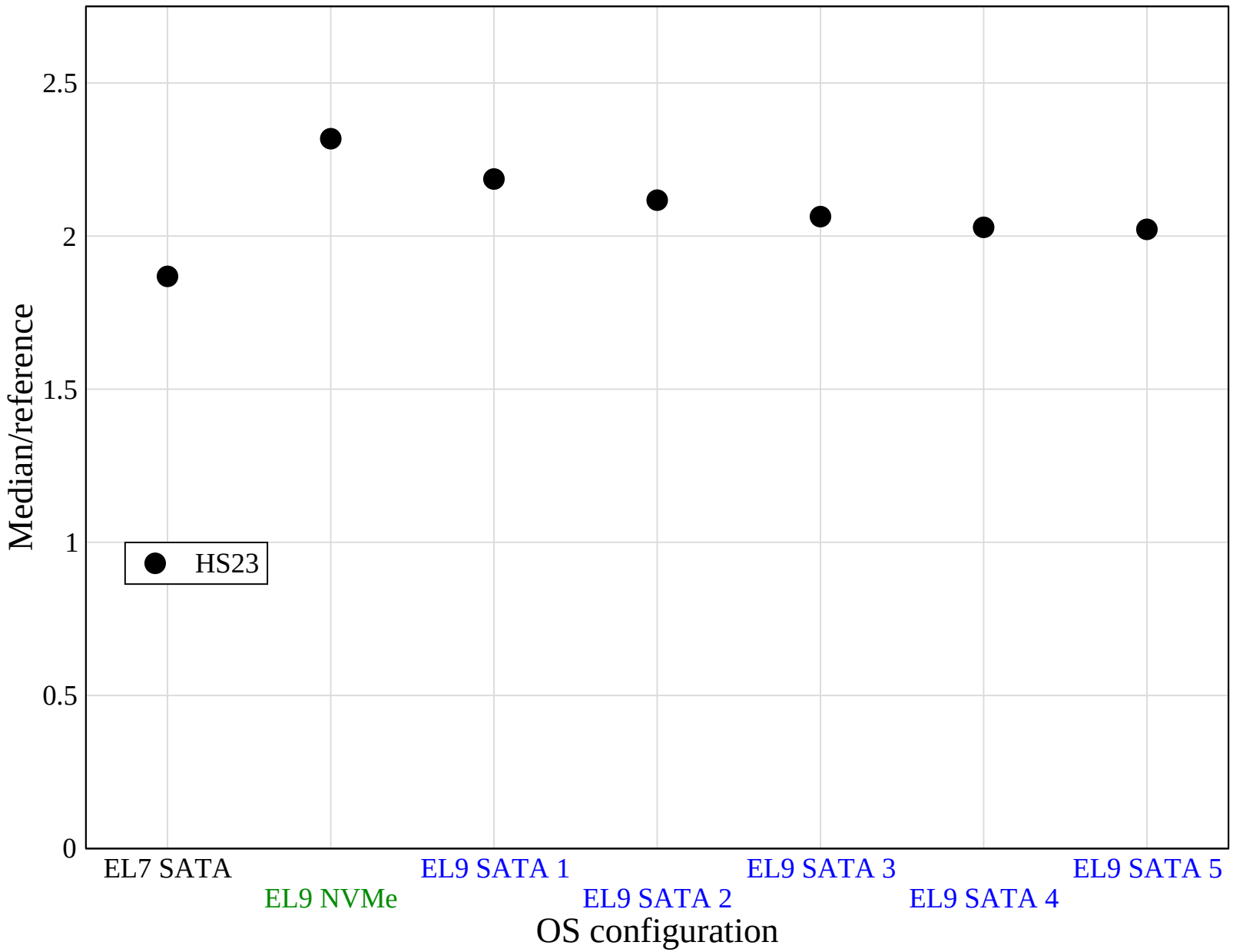
First results



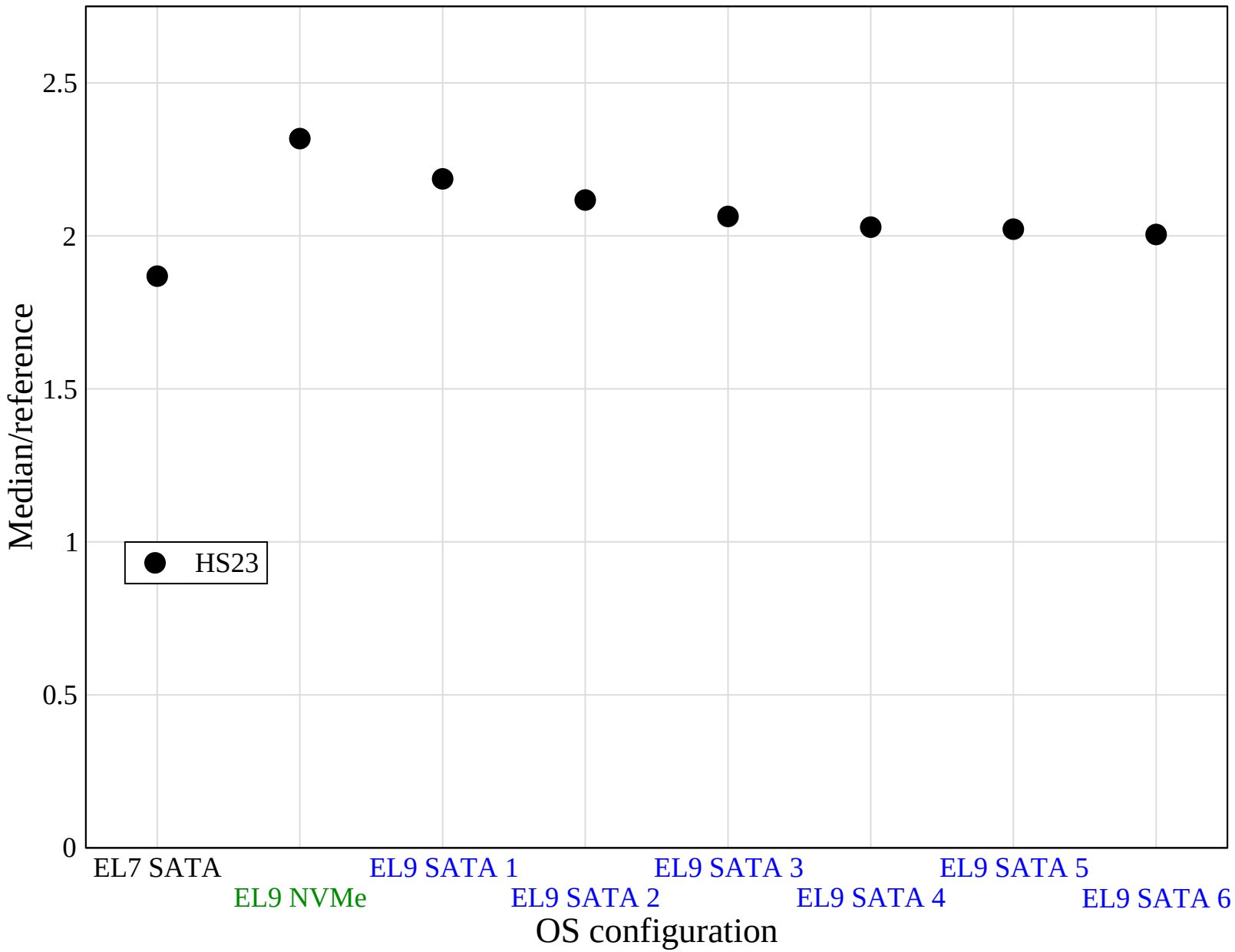
First results



First results



First results



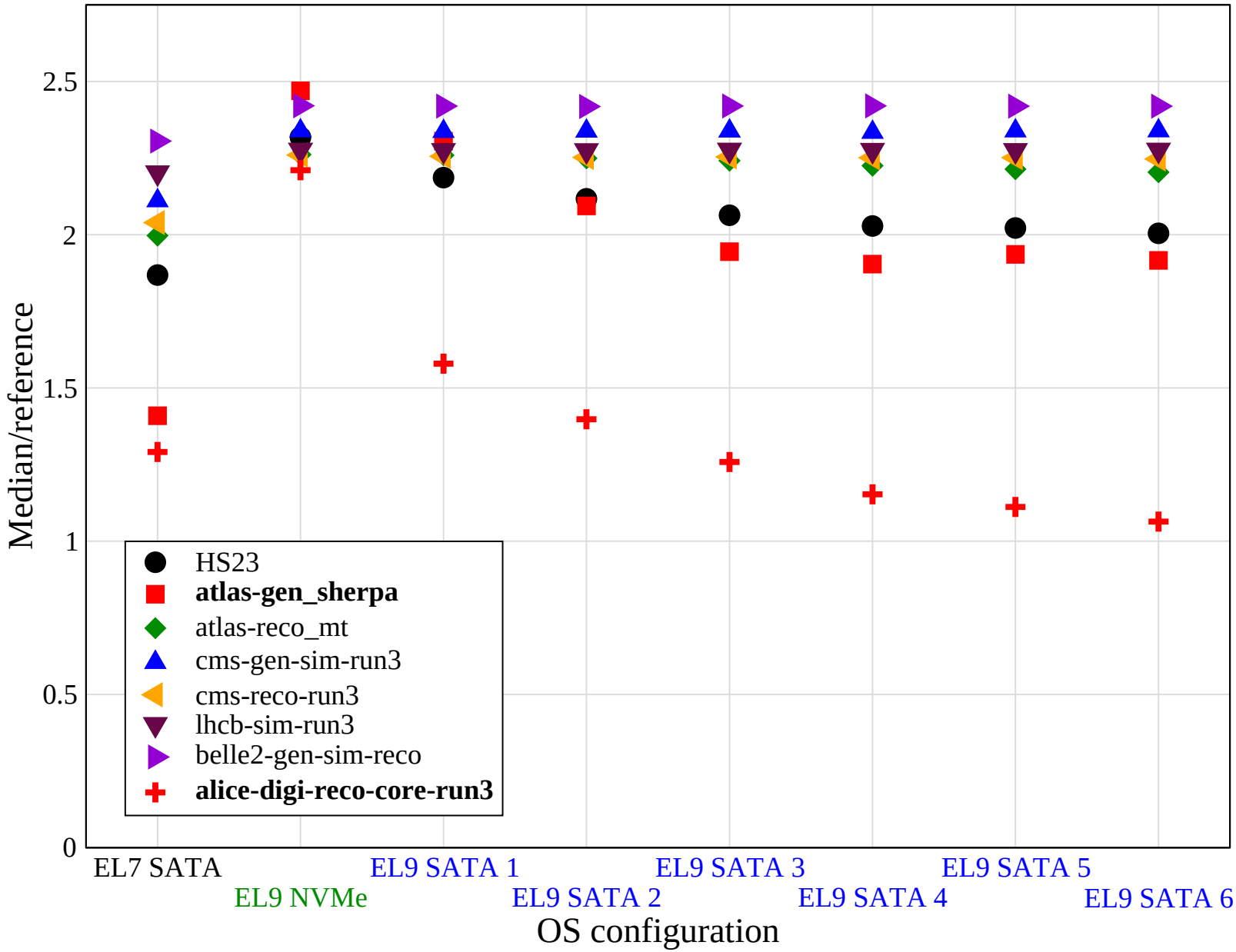
First results

- ☛ An obvious degradation of the results with time

First results

- An obvious degradation of the results with time
- Let's have a look at the separate benchmark workloads:

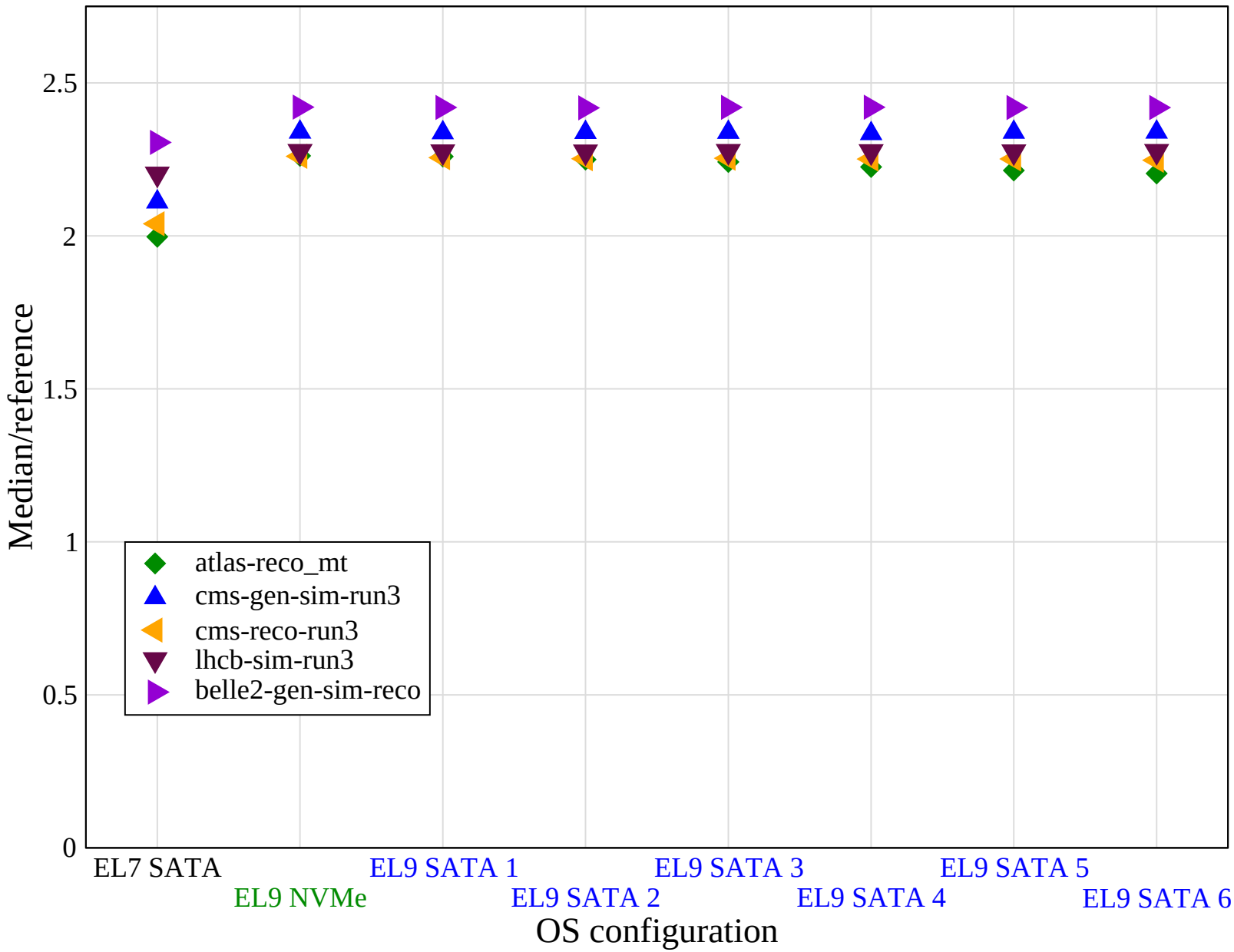
First results



First results

- ☛ Most workload scores remain constant; furthermore, no difference between EL9 SATA and NVMe

First results



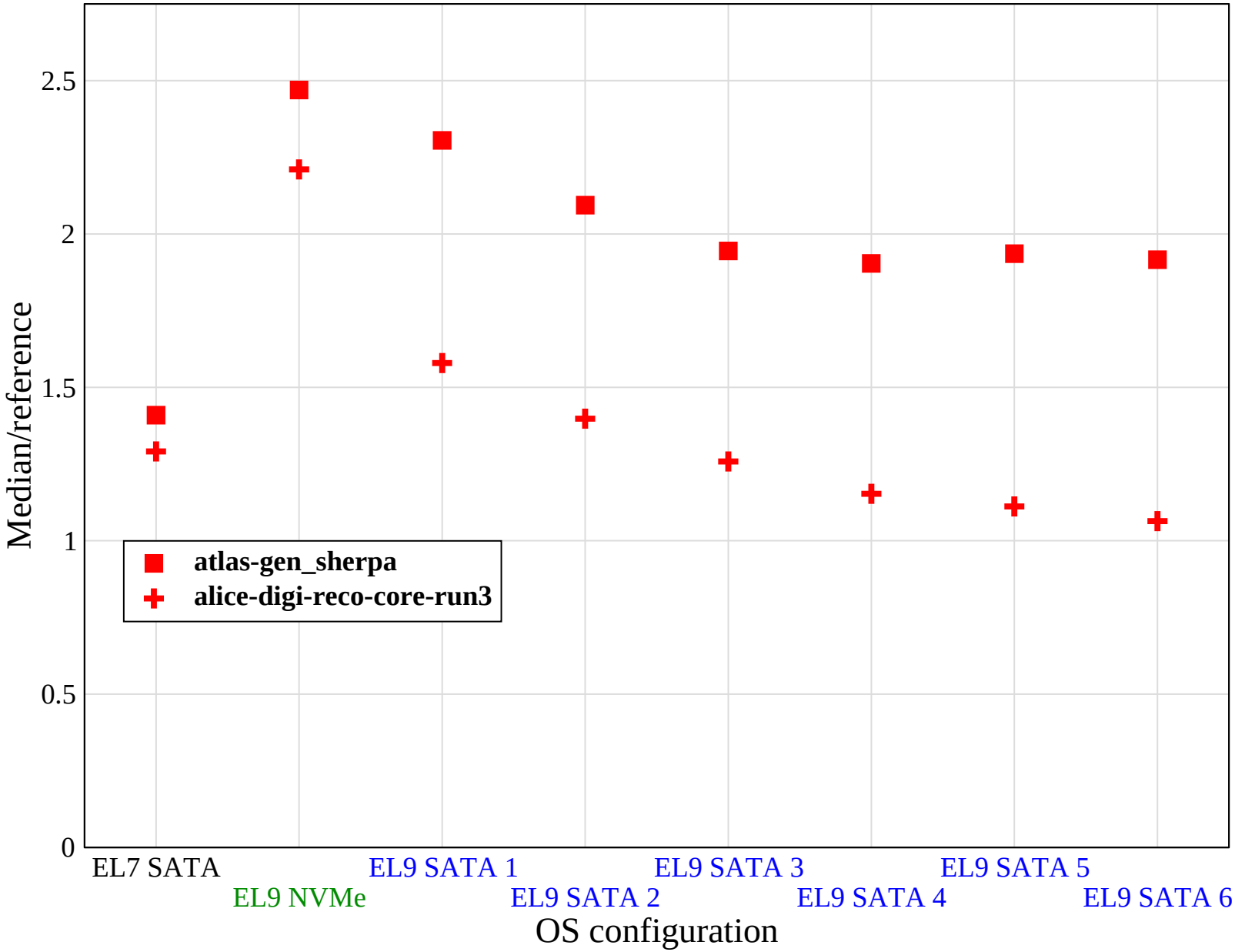
First results

- Most workload scores remain constant; furthermore, no difference between EL9 SATA and NVMe
- But two – `atlas-gen_sherpa` and `alice-digi-reco-core-run3` – show the degradation clearly

First results

- Most workload scores remain constant; furthermore, no difference between EL9 SATA and NVMe
- But two – `atlas-gen_sherpa` and `alice-digi-reco-core-run3` – show the degradation clearly
- These two workloads also show the most striking difference between EL7 and EL9

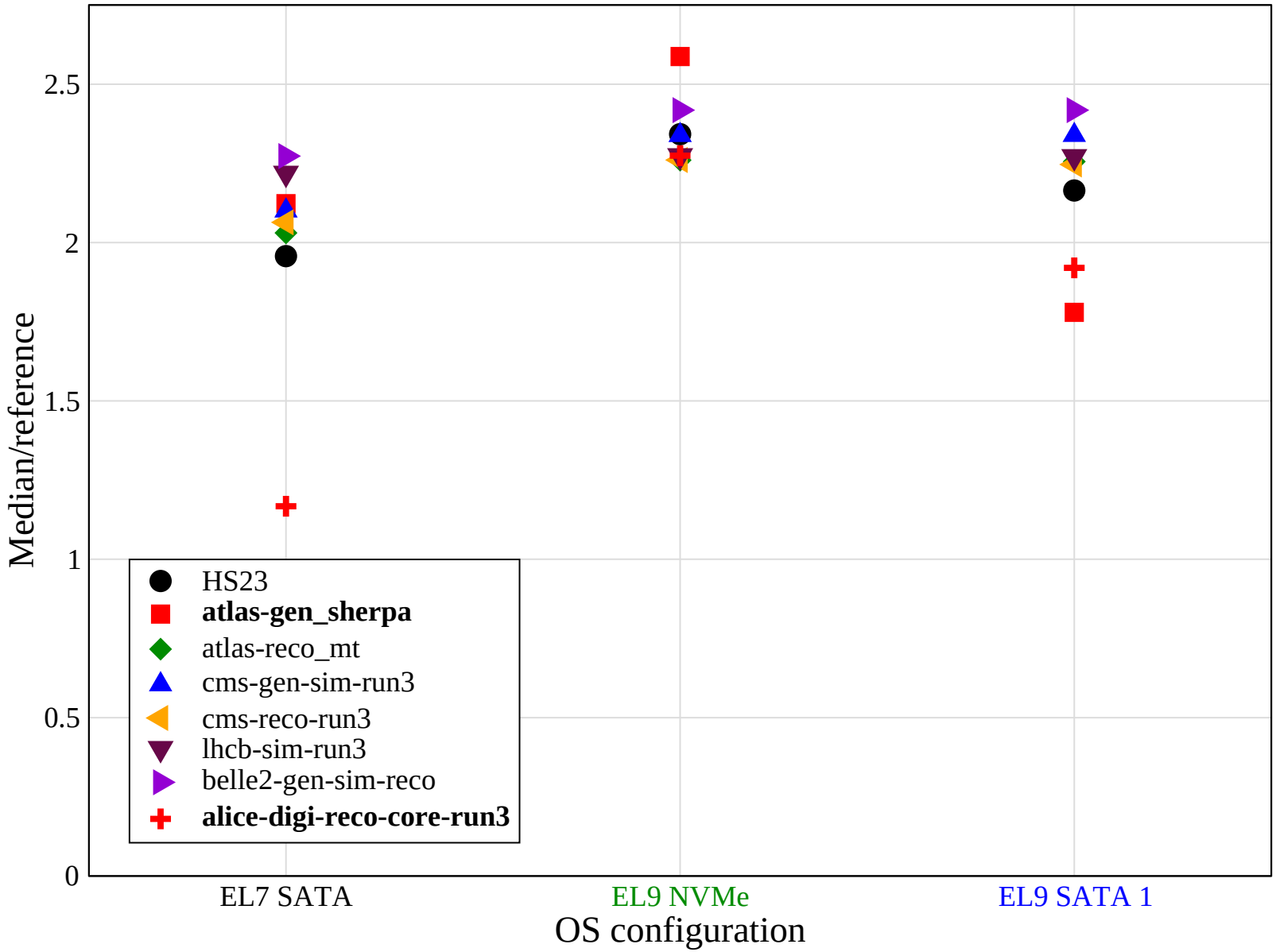
First results



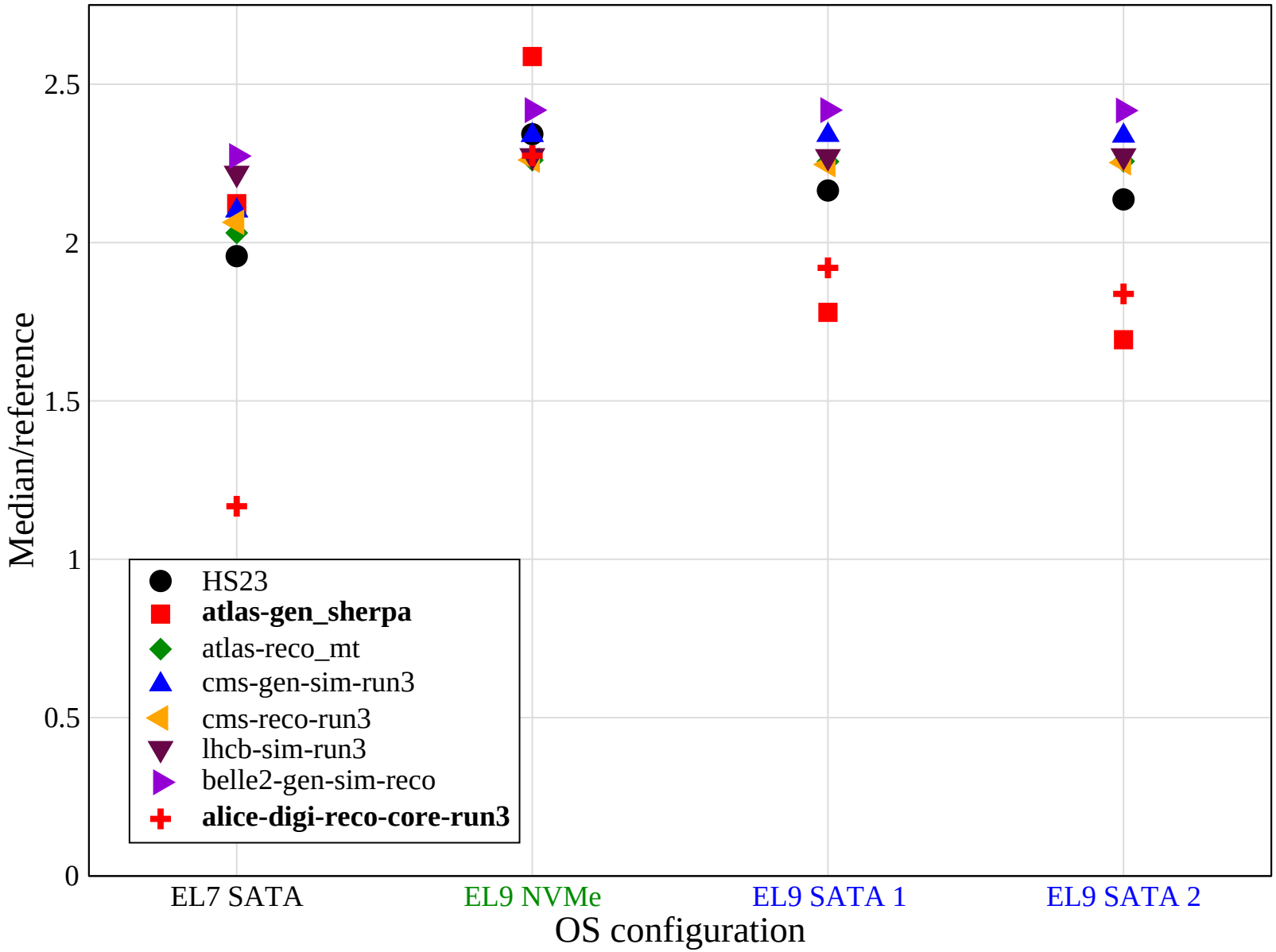
New workload configuration

- ☛ It was suggested to use an improved configuration (`-b hepscore-new-wl`), using hep-score v2.0rc8

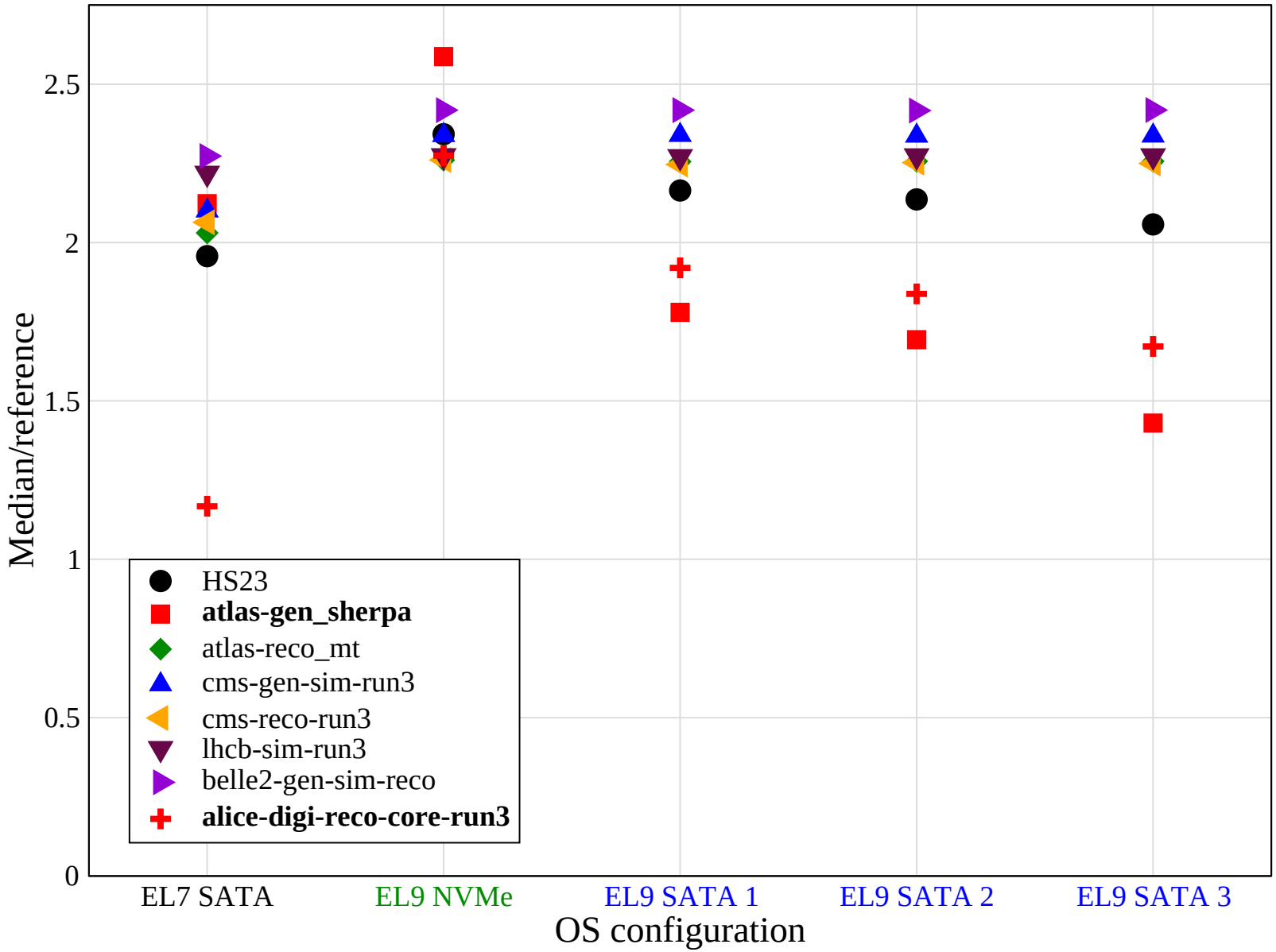
New workload configuration



New workload configuration



New workload configuration



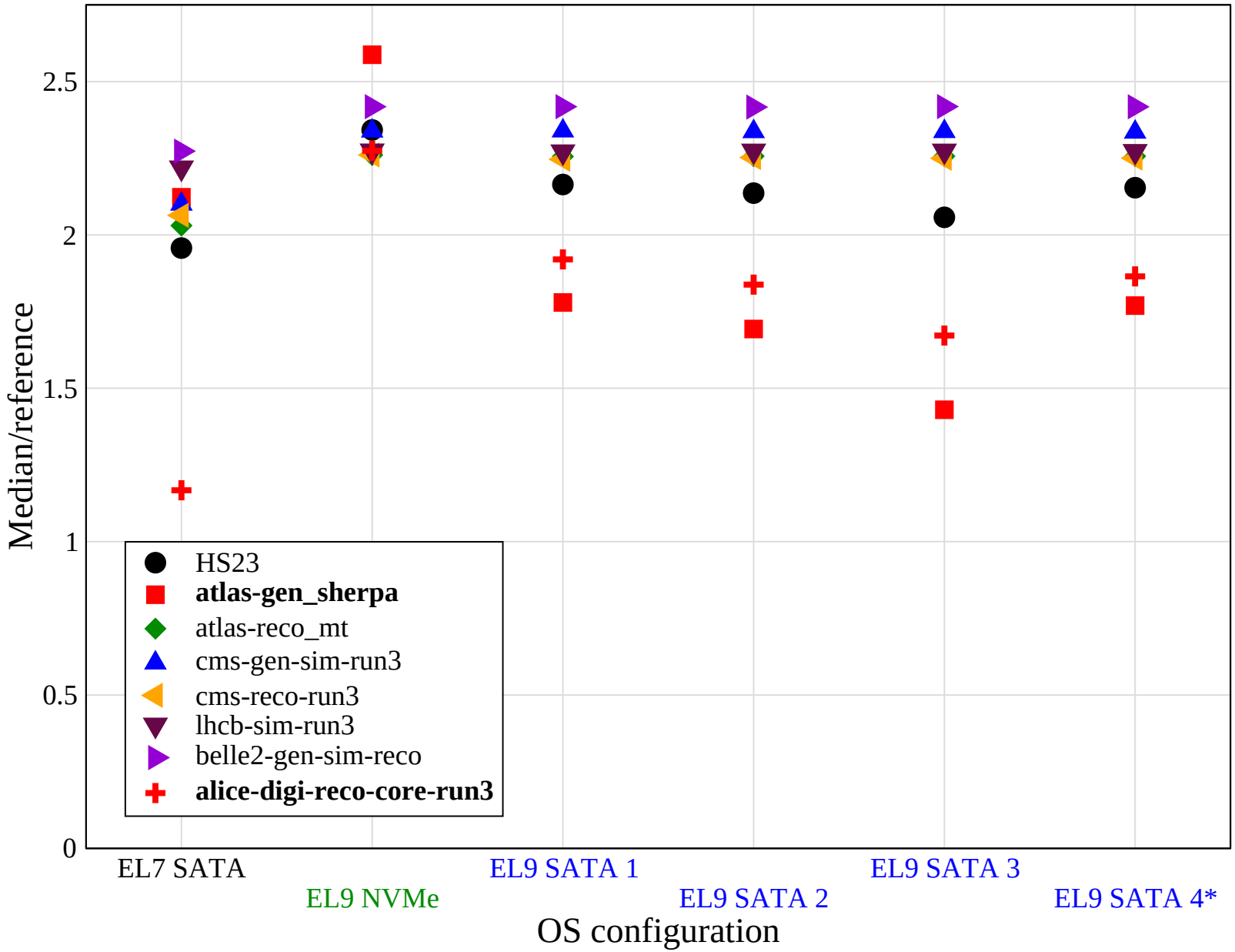
New workload configuration

- It was suggested to use an improved configuration (`-b hepscore-new-wl`), using hep-score v2.0rc8
- The same qualitative picture with `atlas-gen_sherpa` and `alice-digi-reco-core-run3` degrading from run to run

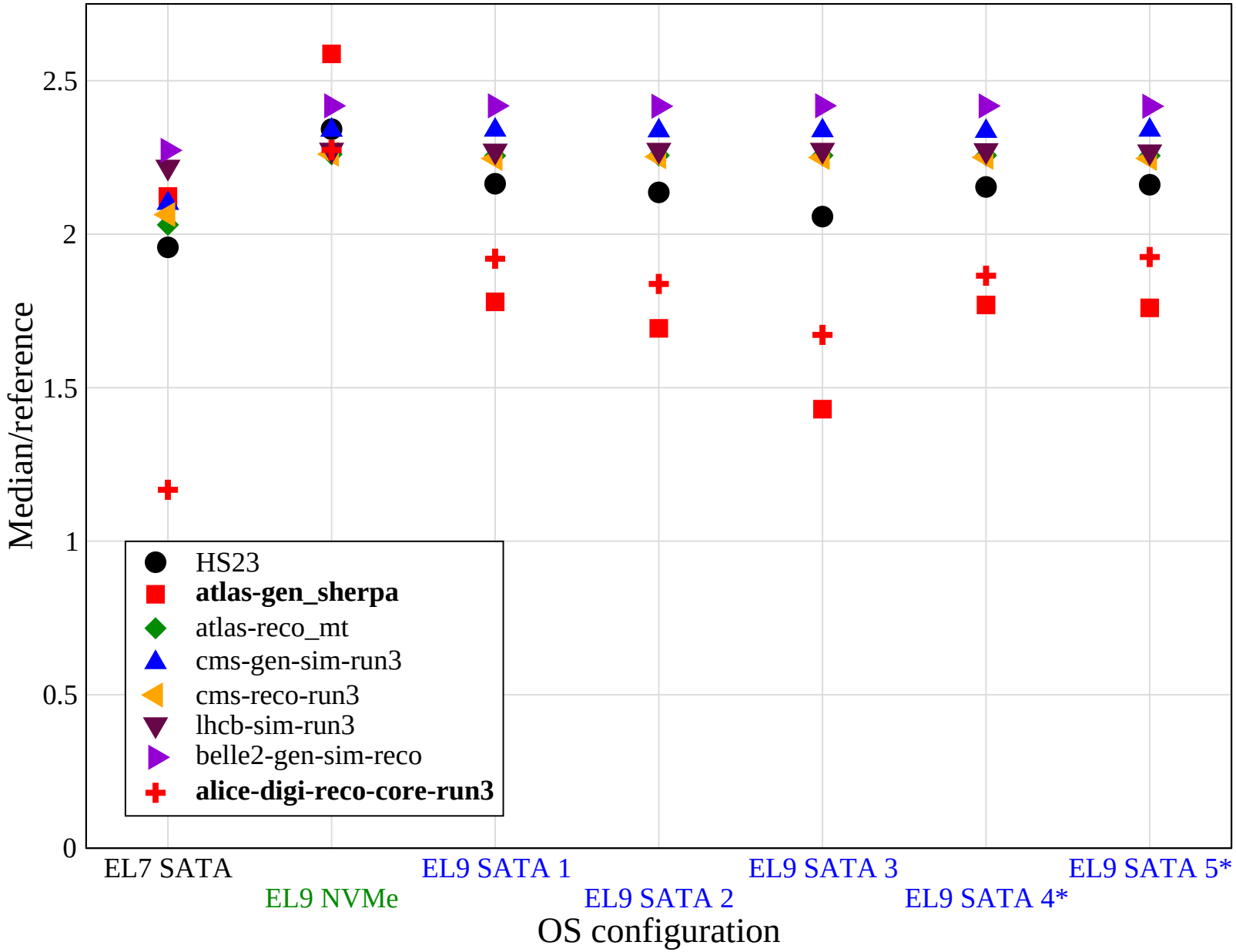
New workload configuration

- It was suggested to use an improved configuration (`-b hepscore-new-wl`), using hep-score v2.0rc8
- The same qualitative picture with `atlas-gen_sherpa` and `alice-digi-reco-core-run3` degrading from run to run
- `fstrim` to the rescue!

New workload configuration



New workload configuration



New workload configuration

- It was suggested to use an improved configuration (`-b hepscore-new-wl`), using hep-score v2.0rc8
- The same qualitative picture with `atlas-gen_sherpa` and `alice-digi-reco-core-run3` degrading from run to run
- `fstrim` to the rescue!
- `fstrim` makes no visible changes in the case of EL9 NVMe and very minor ones ($\sim$1%) for EL7 SATA

New workload configuration

- It was suggested to use an improved configuration (`-b hepscore-new-wl`), using hep-score v2.0rc8
- The same qualitative picture with `atlas-gen_sherpa` and `alice-digi-reco-core-run3` degrading from run to run
- `fstrim` to the rescue!
- `fstrim` makes no visible changes in the case of EL9 NVMe and very minor ones ($\sim$1%) for EL7 SATA
- The `fstrim` service/timer is enabled by default in neither of these OS'es. Why? Debian-based distros have it running once a week by default. It would suffice for EL7 SATA – but not for EL9 SATA!

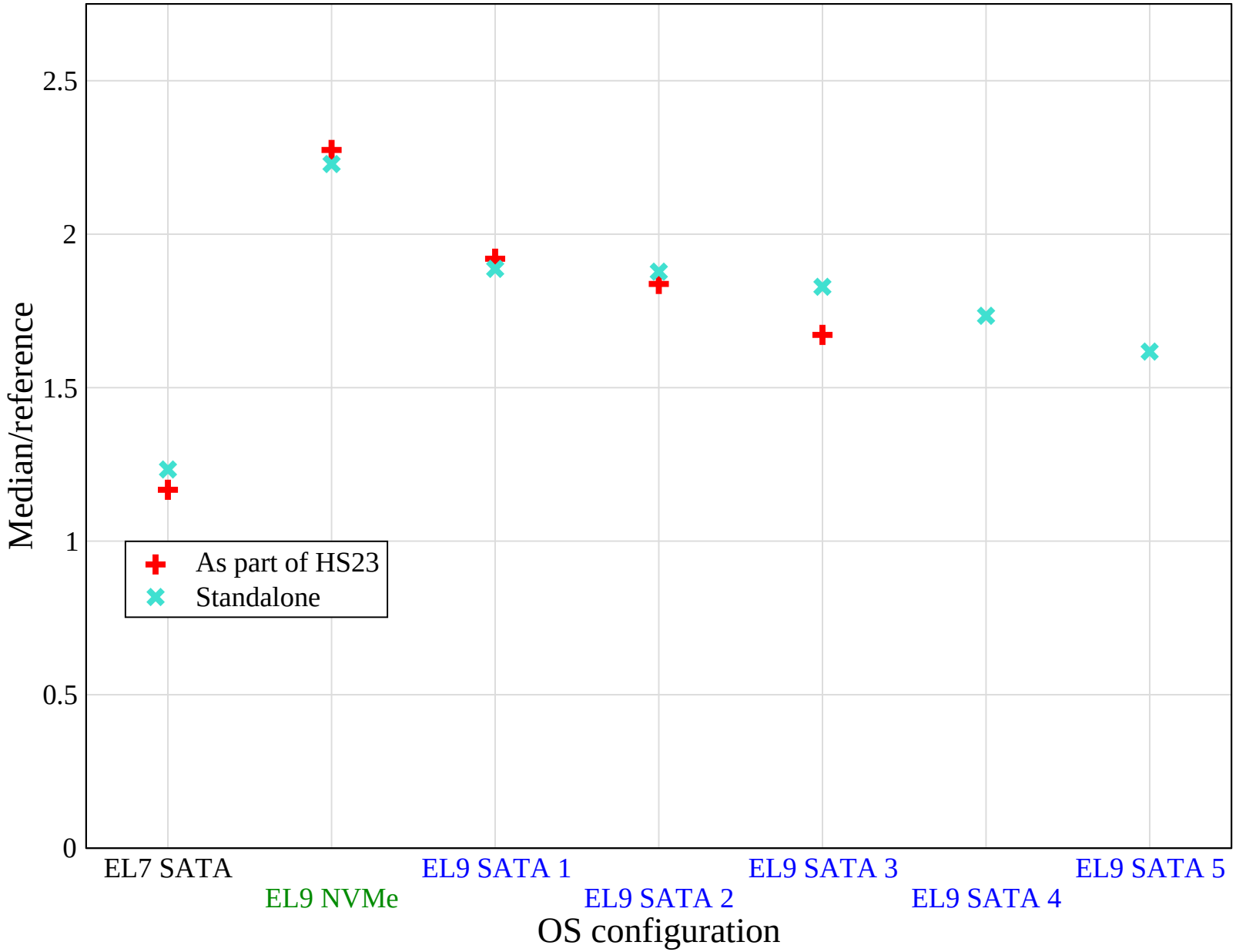
New workload configuration

- It was suggested to use an improved configuration (`-b hepscore-new-wl`), using hep-score v2.0rc8
- The same qualitative picture with `atlas-gen_sherpa` and `alice-digi-reco-core-run3` degrading from run to run
- `fstrim` to the rescue!
- `fstrim` makes no visible changes in the case of EL9 NVMe and very minor ones ($\sim$1%) for EL7 SATA
- The `fstrim` service/timer is enabled by default in neither of these OS'es. Why? Debian-based distros have it running once a week by default. It would suffice for EL7 SATA – but not for EL9 SATA!
- Under EL9, `fstrim` only *queues* the TRIM operations

Focus on the ALICE workload

- ☛ The ALICE workload was run as a single workload

Focus on the ALICE workload



Focus on the ALICE workload

- The ALICE workload was run as a single workload
- The degradation rate is twice slower $\Rightarrow$ it is a *cumulative* effect of I/O of all workloads

Focus on the ALICE workload

- The ALICE workload was run as a single workload
- The degradation rate is twice slower $\Rightarrow$ it is a *cumulative* effect of I/O of all workloads
- The arithmetics of `fstrim` is confusing (even with its "queued" mode of operation in mind):

Focus on the ALICE workload

- The ALICE workload was run as a single workload
- The degradation rate is twice slower $\Rightarrow$ it is a *cumulative* effect of I/O of all workloads
- The arithmetics of `fstrim` is confusing (even with its "queued" mode of operation in mind):
  - According to `iostat`, each (triple) run of `alice-digi-reco-core-run3` writes $\sim$160 GB of data to `$TMPDIR`

Focus on the ALICE workload

- The ALICE workload was run as a single workload
- The degradation rate is twice slower $\Rightarrow$ it is a *cumulative* effect of I/O of all workloads
- The arithmetics of `fstrim` is confusing (even with its "queued" mode of operation in mind):
  - According to `iostat`, each (triple) run of `alice-digi-reco-core-run3` writes ~160 GB of data to `$TMPDIR`
  - But `fstrim -v` shows the same ~390 GB (nearly the size of the file system) before and after one or five consecutive runs!

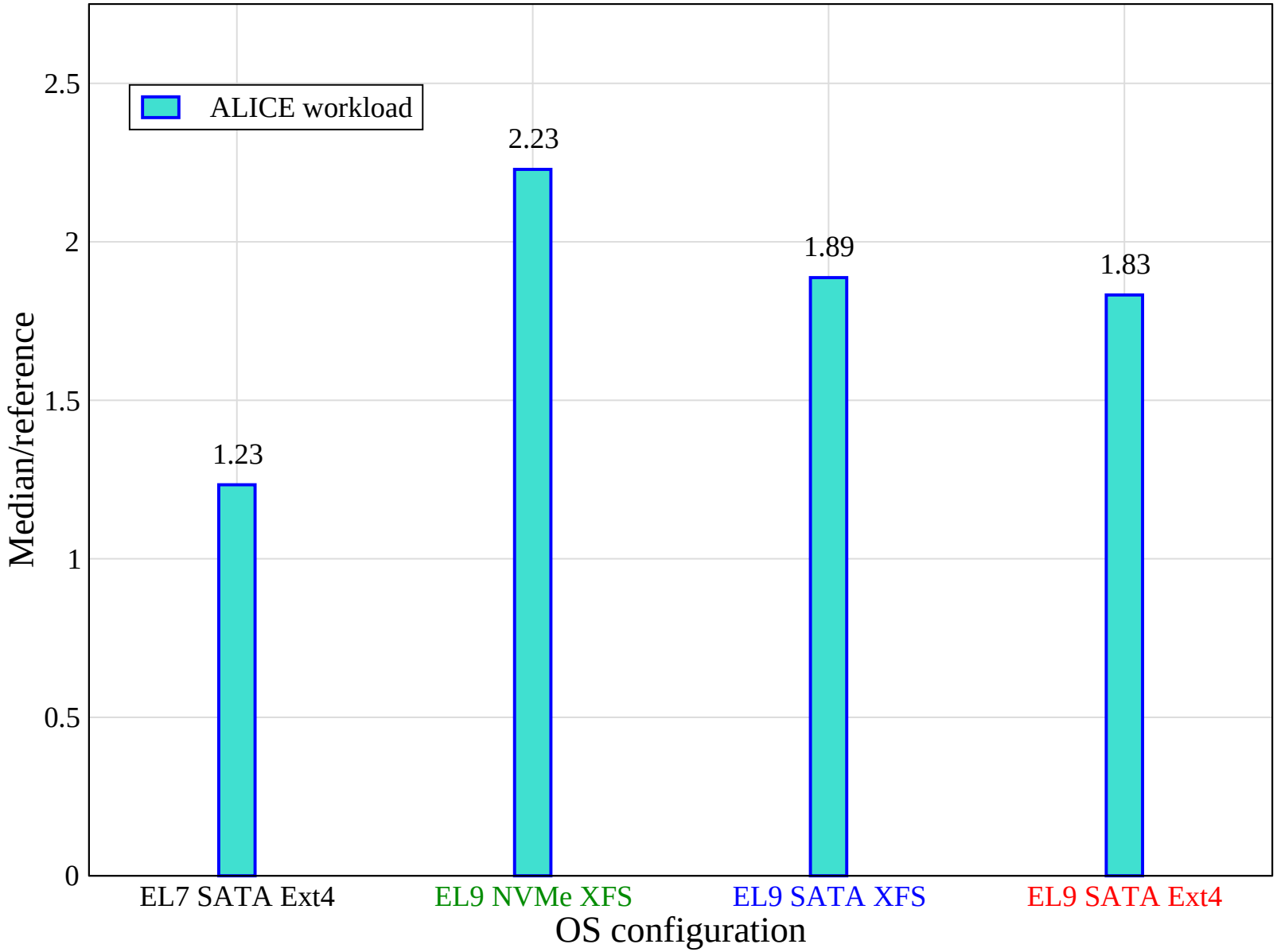
Focus on the ALICE workload

- The ALICE workload was run as a single workload
- The degradation rate is twice slower $\Rightarrow$ it is a *cumulative* effect of I/O of all workloads
- The arithmetics of `fstrim` is confusing (even with its "queued" mode of operation in mind):
  - According to `iostat`, each (triple) run of `alice-digi-reco-core-run3` writes $\sim$160 GB of data to `$TMPDIR`
  - But `fstrim -v` shows the same $\sim$390 GB (nearly the size of the file system) before and after one or five consecutive runs!
  - And yet it helps...

Focus on the ALICE workload

- The ALICE workload was also run directly (i.e., without the `hep-score` wrapper) with the `prmon` option to gather detailed stats
- The results have not been analyzed yet (but the data files are available in the GGUS ticket)

(Un)importance of file system



Conclusions

- Most of the HS23 workloads show a stable ∼10% improvement in EL9 vs EL7, no matter which disk/file system is used. It is surprising given that the benchmarks run in an EL7 containerized environment.
- Two workloads – `atlas-gen_sherpa` and `alice-digi-reco-core-run3` – are highly sensitive to the storage type used.
- It has implications both for the benchmark calibration and performance of the real workloads.
- Running `fstrim` periodically is crucial in the case of the EL9 SATA setup.
- These findings and their generality need to be further investigated.

Thank you
for your attention!

The help of Alexey Konvisher with the hardware setup is highly appreciated.

Extra material

Script used for running HS23

```
#!/bin/sh

HEPSCORE=${HOME}/.local/bin/hep-score
HEPRESDIR=${HOME}/hs23/results
SING_HOME=/cvmfs/atlas.cern.ch/repo/containers/sw/singularity/x86_64-el7/3.8.6

# Respect definition from the batch system
if test -z $TMPDIR
then
    WORKDIR=`mktemp -d`
else
    WORKDIR=$TMPDIR
fi

# A Lustre volume with prefetched singularity images
CACHEDIR=${HOME}/storage/singularity

mkdir -p ${WORKDIR}
mkdir -p ${WORKDIR}/singularity
mkdir -p ${CACHEDIR}
mkdir -p ${HEPRESDIR}

export PATH=${SING_HOME}/bin:$PATH
export SINGULARITY_CACHEDIR=${CACHEDIR}
export SINGULARITY_TMPDIR=${WORKDIR}/singularity

outfile=${HEPRESDIR}/${HOSTNAME}.ref.txt
logfile=${HEPRESDIR}/${HOSTNAME}.log

${HEPSCORE} -b hepscore-new-wl -v -o ${outfile} ${WORKDIR} > ${logfile} 2>&1
```

Script used for running ALICE workload directly

```
#!/bin/sh

HEPSCORE=${HOME}/.local/bin/hep-score
HEPRESDIR=${HOME}/hs23/results
SING_HOME=/cvmfs/atlas.cern.ch/repo/containers/sw/singularity/x86_64-el7/3.8.6
SING_IMAGE=oras://gitlab-registry.cern.ch/hep-benchmarks/hep-workloads-sif/alice-digi-reco-co

# Respect definition from the batch system
if test -z $TMPDIR
then
    WORKDIR=`mktemp -d`
else
    WORKDIR=$TMPDIR
fi

# A Lustre volume with prefetched singularity images
CACHEDIR=${HOME}/storage/singularity

mkdir -p ${WORKDIR}
mkdir -p ${WORKDIR}/tmp
mkdir -p ${WORKDIR}/results
mkdir -p ${WORKDIR}/singularity
mkdir -p ${CACHEDIR}
mkdir -p ${HEPRESDIR}

export PATH=${SING_HOME}/bin:$PATH
export SINGULARITY_CACHEDIR=${CACHEDIR}
export SINGULARITY_TMPDIR=${WORKDIR}/singularity

singularity run -i -c -e \
 -B ${WORKDIR}/results:/results \
 -B ${WORKDIR}/tmp:/tmp \
 -B ${WORKDIR}/tmp:/var/tmp \
  ${SING_IMAGE} -W --threads 4 --events 3 --prmon
```