

MariaDB on ARM



Krunal Bauskar
(Driving [#mysqlonarm](#) initiative)

About me....



- Working on MySQL for more than decade now.
- While @ Percona worked as Percona XtraDB Cluster (PXC) product lead managing all aspects of the product. Scaled PXC by improving performance and made it more stable.
- Have worked as InnoDB developer @ MySQL/Oracle developing undo-log-truncate, atomic-truncate, switching from myisam to innodb temp-tables, temp-tablespace, etc...
- Also in past, researched Big-Data problems @ Yahoo! Labs, worked on making fastest MySQL engine (for startup now part of Teradata), lot of such projects.
- Currently working @ Huawei as part of Open Source DB group responsible for flavors of MySQL ecosystem (MySQL, MariaDB and Percona).

Driving #mysqlonarm initiative (covers MySQL, MariaDB, Percona)

Agenda



- ARM ecosystem
- MariaDB and ARM
- How MariaDB flares on ARM?
- State of Ecosystem

Agenda

- ARM ecosystem



Evolving ARM ecosystem



What comes to your mind when you think of an ARM processor....?

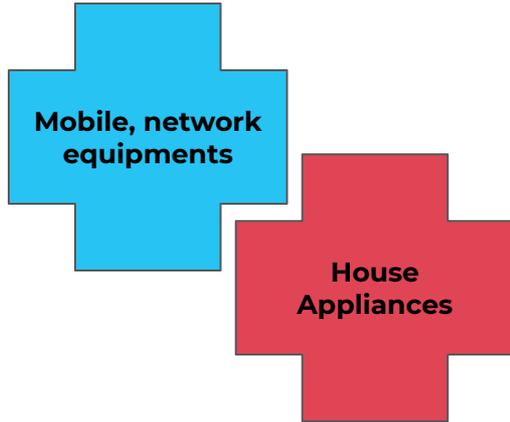
Evolving ARM ecosystem

What comes to your mind when you think of an ARM processor....?



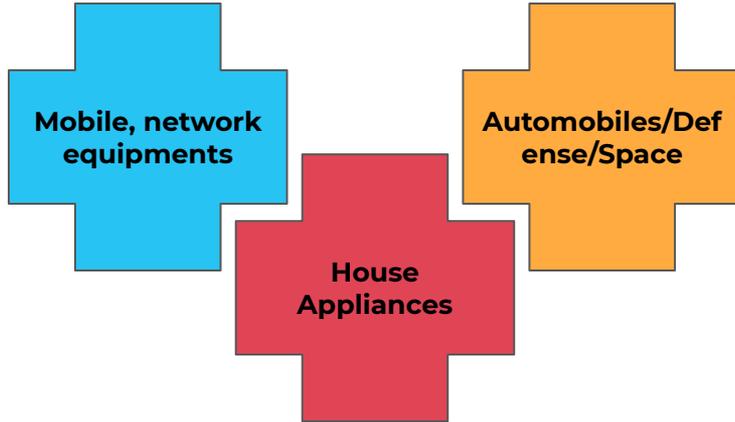
Evolving ARM ecosystem

What comes to your mind when you think of an ARM processor....?



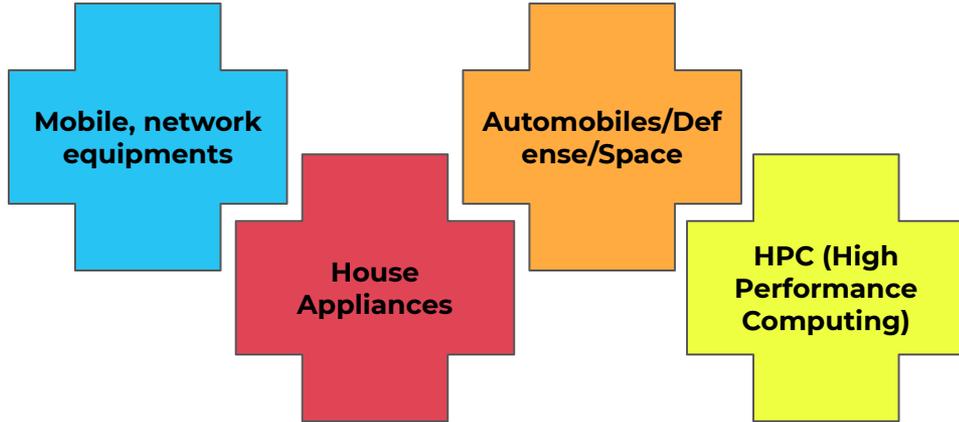
Evolving ARM ecosystem

What comes to your mind when you think of an ARM processor....?



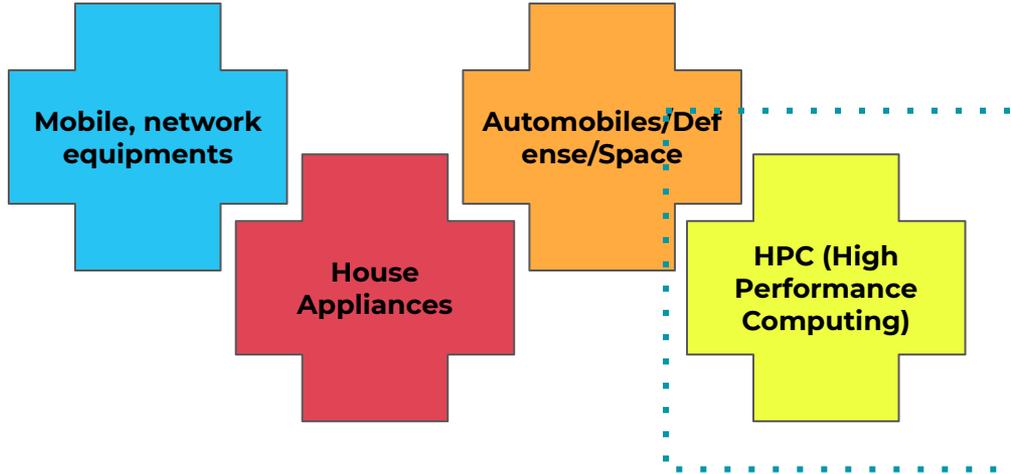
Evolving ARM ecosystem

What comes to your mind when you think of an ARM processor....?



Evolving ARM ecosystem

What comes to your mind when you think of an ARM processor....?



Growing popularity of ARM processor in running high performance application like databases, application server, etc.. due to optimal **performance per watt**.

Multiple options has mushroomed in last couple of years especially enabling cloud:

- **HiSilicon Kunpeng 920**
- **Amazon Graviton 2**
- **Ampere Altra**

Evolving ARM ecosystem



- In past few years, all major OS providers started providing respective releases on ARM.
- Multiple projects (big-data, application server, load balancer, databases, etc....) are now supported on ARM.
- ARM cloud instances are estimated to provide direct saving of upto 30% keeping the performance on-par.
- Strong presence of ARM on data-generating devices could further fuel use of ARM based backend server (HPC).

ARM is catching faster than expected and slated to become most booted instance for HPC by 2030* (could be earlier).

Agenda

- ARM ecosystem
- MariaDB and ARM



MariaDB and ARM



- MariaDB already supports ARM and releases packages for Ubuntu, CentOS on ARM (with work-in-progress to add support for more distros in near future).
- MariaDB also have optimized MDB-server on ARM and efforts still continues (optimization is never-ending).
- Infact, MariaDB was first to add support for ARM (among MySQL variants).
- Full Support/GA so all bugs should be addressed (like x86) including performance issues.

MariaDB and ARM



Challenges
adding
support for
ARM

MariaDB and ARM



Challenges
adding
support for
ARM

Weak memory order so for
optimal performance
proper use of
memory-ordering.

MariaDB and ARM



Challenges
adding
support for
ARM

Weak memory order so for optimal performance proper use of memory-ordering.

Tuning code for ARM architecture and CPU frequency (mutex-wait, spin-loop).

MariaDB and ARM



Challenges
adding
support for
ARM

Weak memory order so for optimal performance proper use of memory-ordering.

Tuning code for ARM architecture and CPU frequency (mutex-wait, spin-loop).

More numa nodes (more but less powerful cores) so software need to be numa compliant and optimized.

MariaDB and ARM



Challenges
adding
support for
ARM

Inherent ARM acceleration through use of NEON/SIMD/LSE instructions.

Weak memory order so for optimal performance proper use of memory-ordering.

Tuning code for ARM architecture and CPU frequency (mutex-wait, spin-loop).

More numa nodes (more but less powerful cores) so software need to be numa compliant and optimized.

MariaDB and ARM



Challenges adding support for ARM

Tuning cache-line difference, alignment hints, pipeline differences, etc...

Inherent ARM acceleration through use of NEON/SIMD/LSE instructions.

Weak memory order so for optimal performance proper use of memory-ordering.

Tuning code for ARM architecture and CPU frequency (mutex-wait, spin-loop).

More numa nodes (more but less powerful cores) so software need to be numa compliant and optimized.

MariaDB and ARM



Switching to use HW optimized support (ACLE)

Tuning cache-line difference, alignment hints, pipeline differences, etc...

Inherent ARM acceleration through use of NEON/SIMD/LSE instructions.

Challenges adding support for ARM

Weak memory order so for optimal performance proper use of memory-ordering.

Tuning code for ARM architecture and CPU frequency (mutex-wait, spin-loop).

More numa nodes (more but less powerful cores) so software need to be numa compliant and optimized.

MariaDB and ARM



Memory-Order

- rw-lock memory barriers ([MDEV-14529](#))
- rollback-segment memory-barrier ([MDEV-22794](#))

NUMA/Scalability

- spin-lock optimization ([MDEV-14374](#))
- Timer-counter-optimization ([MDEV-23249](#))
- cacheline sharing ([MDEV-14482](#))
- Connection scalability ([MDEV-14505](#))
- Mutex optimization ([MDEV-15059](#))
- undo-log-record-optimization ([MDEV-15090](#))
- Trx-sys-optimization ([MDEV-15158](#), [MDEV-15132](#), [MDEV-15104](#))

NEON/SIMD/LSE

- Adopting atomic for trx-sys-rw-trx-set ([MDEV-14638](#))
- Read-view optimization ([MDEV-14756](#))

ACLE

- crc32c (for page-checksum [MDEV-14129](#))
- crc32 (for table and binlog checksum [MDEV-22641](#))

Agenda



- ARM ecosystem
- MariaDB and ARM
- How MariaDB flares on ARM?

Cost Performance Model



- Understanding Cost-Performance Model

	Huawei Cloud	AWS	packet.com
ARM	2745 USD (yearly) 16 vCPU/64GB (2.6 Ghz Kunpeng 920 based instance)	3384 USD (yearly) 16 vCPU/64GB (2.5 Ghz Graviton 2 based instance)	8760* USD (yearly) 32 vCPU/128GB (3.3 Ghz Ampere eMAG with storage)
x86	4687 USD (yearly) 16 vCPU/64GB (3.0/3.4 Ghz Intel Xeon 6266C Cascade Lake based instance)	4239 USD (yearly) 16 vCPU/64GB (2.5/3.1 Ghz Intel Xeon 8175M Skylake)	17520 USD (yearly) 32 vCPU/256GB (2.5 Ghz AMD EPYC 7502P with storage pricing)
Saving	41%	20%	~ 50% (with storage)

Cost Performance Model



Can we leverage the said cost saving and still get improved or on-par throughput?

Challenges with existing model:

- CPU frequency differences. (2.5, 2.6. 3.1, Turbo mode).
- CPU generation differences (supporting advanced instruction set).
- x86 physical cores -vs- HT cores (vCPU as per cloud terminology)
- NUMA arrangements.
- Memory differences.
- Scheduler (especially for multi-numa) issues. (improved kernel support).

Cost Performance Model



Keeping cost constant let's analyze if we can get more throughput (there by more tps/per USD) from ARM based instances

More formally:

Say given X USD to spend, we can get

- M ARM resources **-- or --**
- N x86 resources ($M > N$)

Can we get increased throughputs using the said M resources.

“Given the on-par cost of both the resources can we exploit the ARM variant to its fullest benefit and get better performance when compared to x86”

Cost Performance Model



- Comparable VM (with HC)
 - x86 => 12 vCPU/48 GB 3432 USD (yearly)
 - ARM => 24 vCPU/48GB 3578 USD (yearly)

So for the approximately same cost we get 2x more cores (36 Ghz (x86) -- vs -- 62.4 Ghz (ARM)).

- Difference is even wider with bare-metal

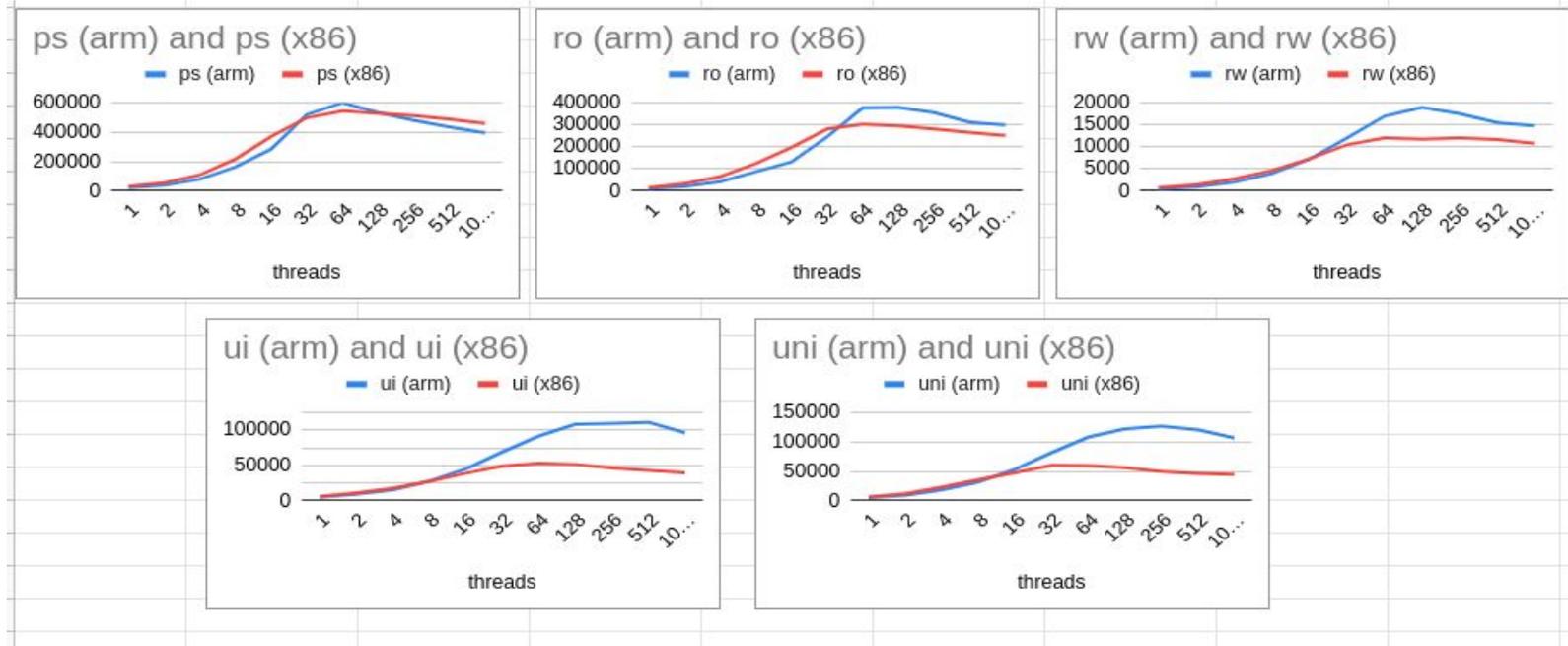
How MariaDB flares on ARM ?



- Now let's see if all those things really help. All these enhancements have been done for past 3+ years so we don't have baseline ARM to compare. In lack of good baseline let's use x86 as baseline.
- **Setup:**
 - **MDB-Server-Configuration:**
 - 100 tables * 3 millions (69 GB data)
 - 80 GB of buffer pool (CPU-bound)/35 GB of buffer pool (IO-bound)
 - 20 GB of redo-log
 - MDB-10.4.13/10.5.5 (both GAed)
 - **Test-Scenarios**
 - sysbench: point-select, read-only, read-write, update-index, update-non-index
 - **Machine Configuration (bare-metal)**
 - **x86_64:** Intel(R) Xeon(R) Gold 6151 CPU @ 3.00GHz (*HT enabled*) [28 ht-cores: 22 server + 6 client]
 - **ARM:** Kunpeng 920 (2.6 Ghz) [64 cores: 58 server + 6 client]
 - **Storage**
 - 1.6 TB NVMe SSD (random-write-iops: 70K (16K block), random-read-iops: 180K (16K block))

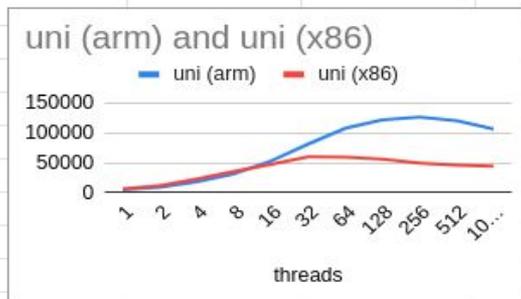
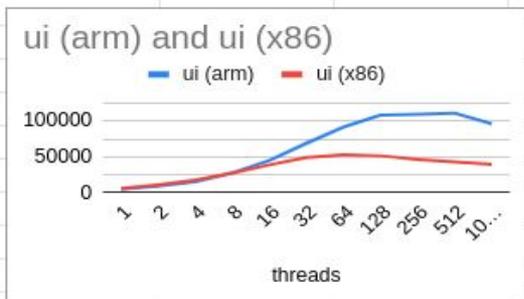
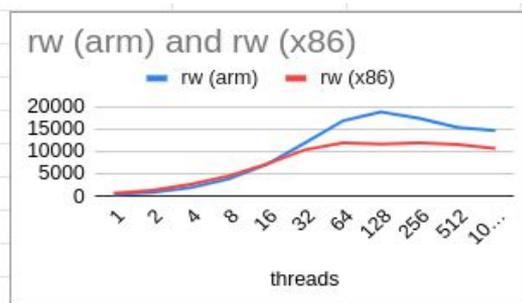
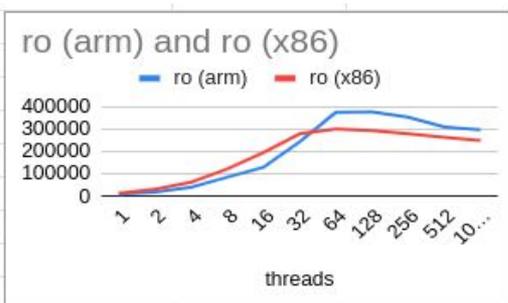
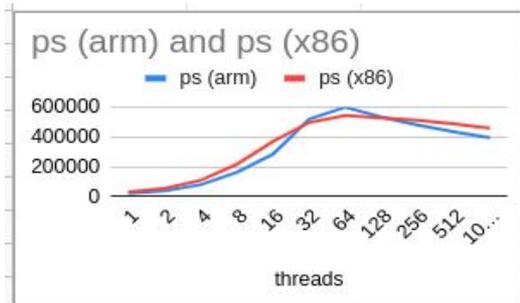
How MariaDB flares on ARM ?

- MDB-10.4.13 (CPU-Bound)



How MariaDB flares on ARM ?

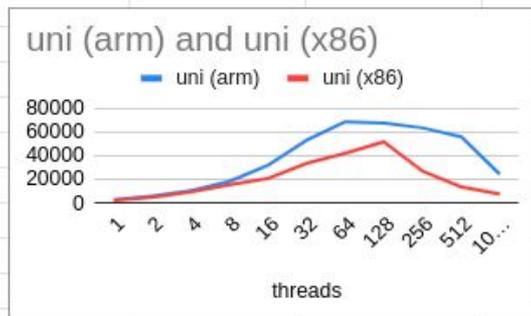
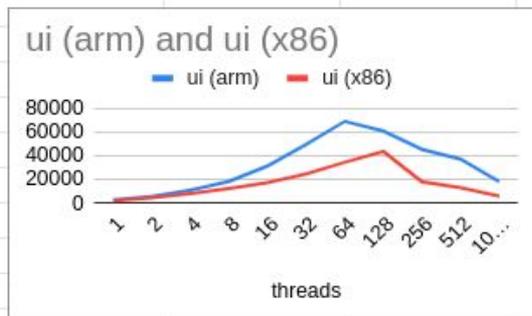
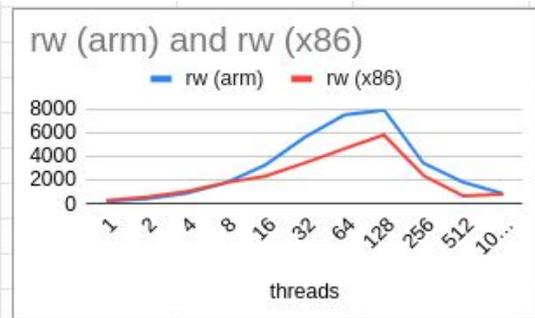
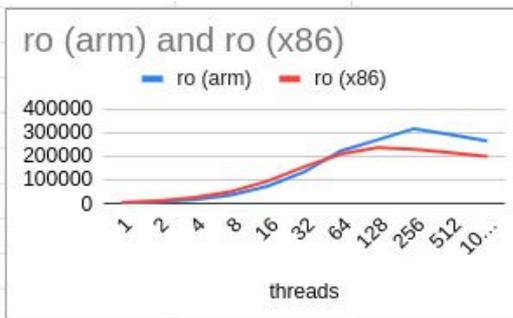
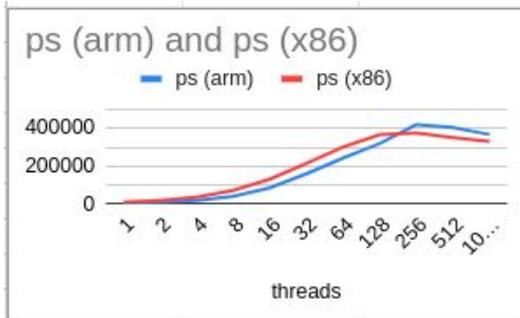
- MDB-10.4.13 (CPU-Bound)



ARM continue to beat x86 with increasing scalability for all kind of workload. Especially the difference is wider with read-write workload.

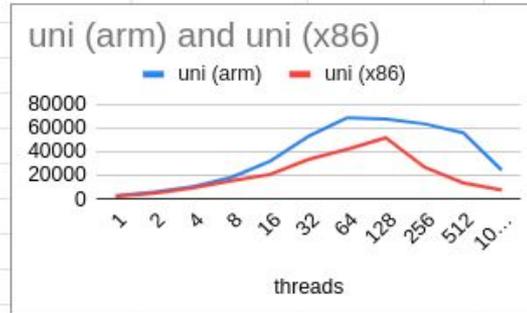
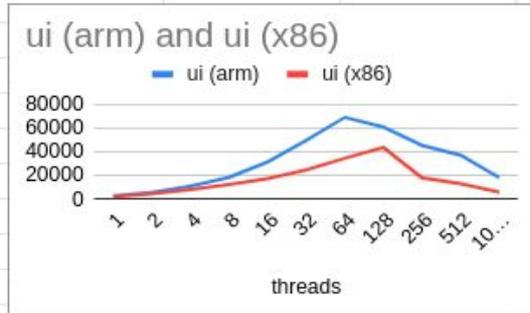
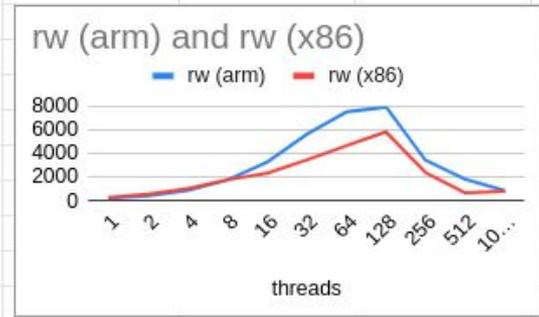
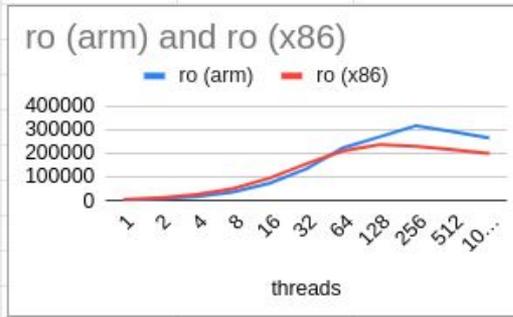
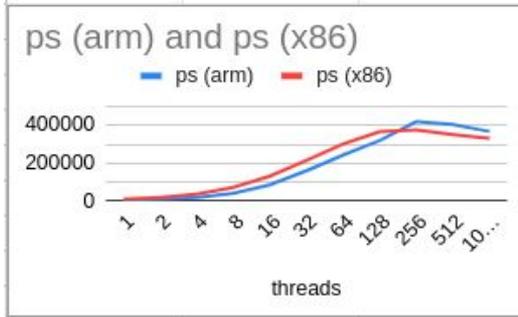
How MariaDB flares on ARM ?

- MDB-10.4.13 (IO-Bound)



How MariaDB flares on ARM ?

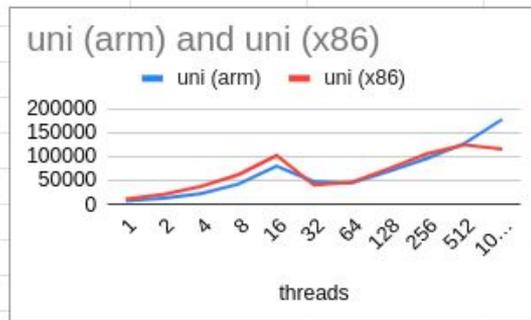
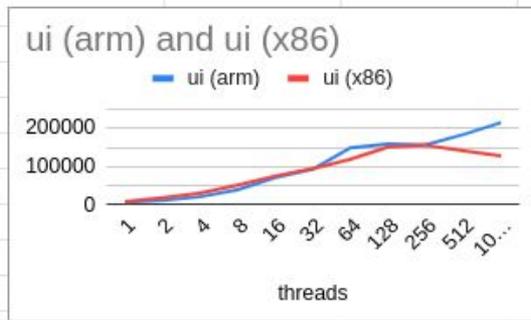
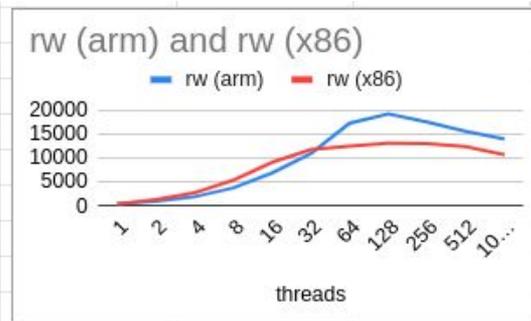
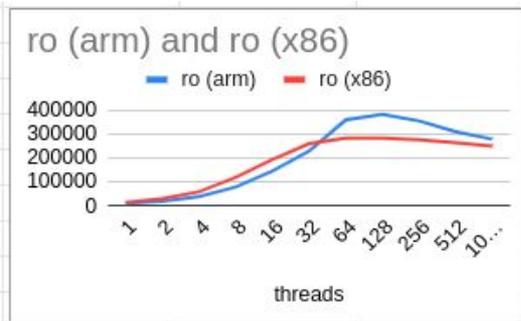
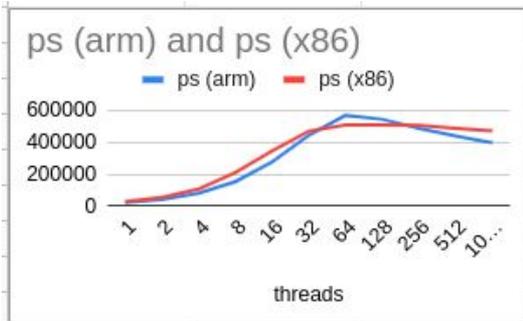
- MDB-10.4.13 (IO-Bound)



Even for IO bound workload ARM continue to beat x86. Infact, for read-write workload, ARM continue to scale better on lower-scalability too (flushing effect?)

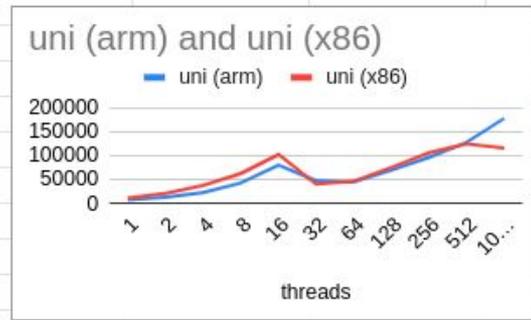
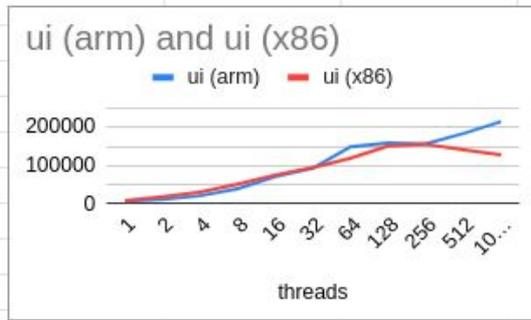
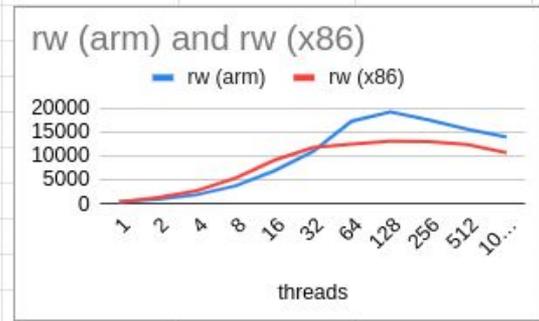
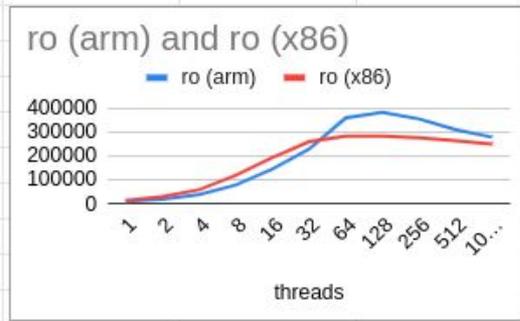
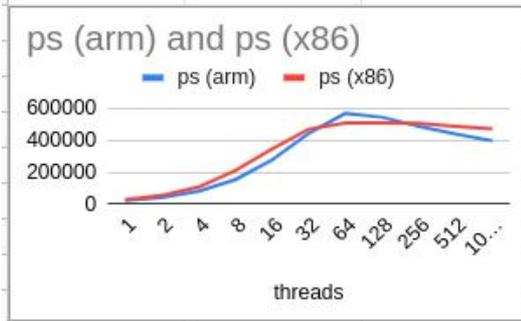
How MariaDB flares on ARM ?

- MDB-10.5.5 (CPU-Bound)



How MariaDB flares on ARM ?

- MDB-10.5.5 (CPU-Bound)



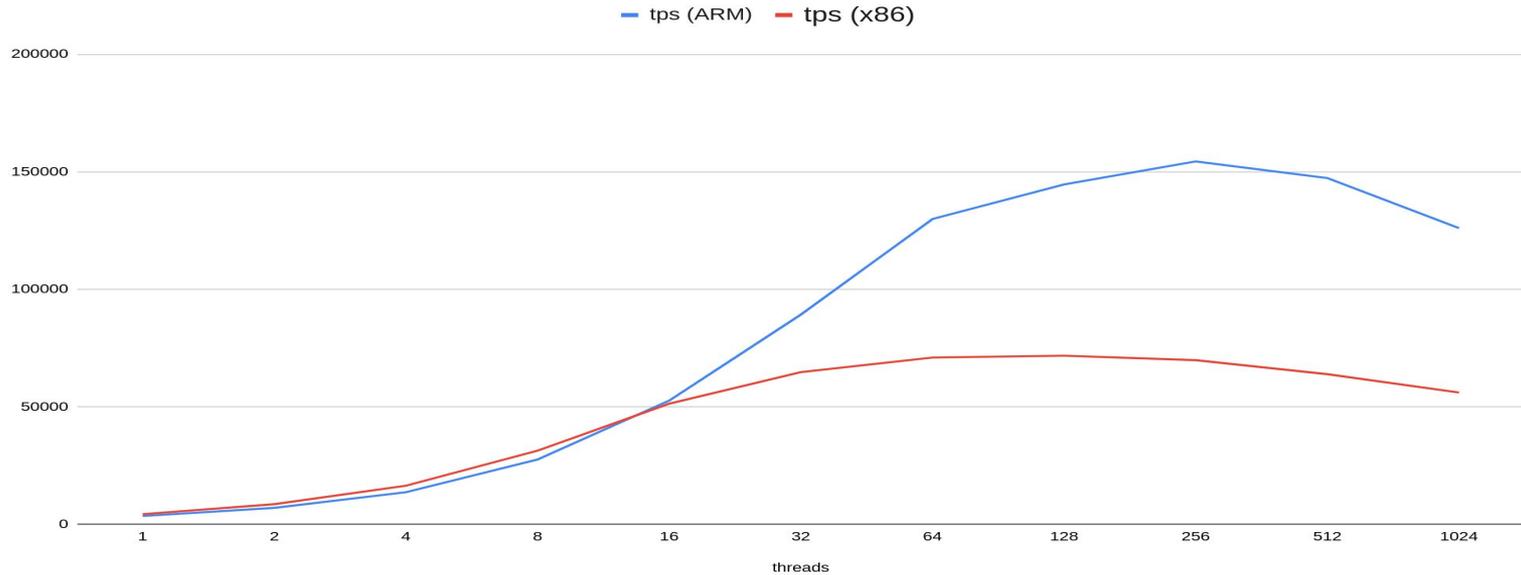
10.5.5 is known to have some flushing issue so it better to benchmark with it once all these issues are fixed. But as reference we can see ARM continue to beat x86 for higher scalability.

How MariaDB flares on ARM ?



- TPCC-benchmark (CPU bound workload)

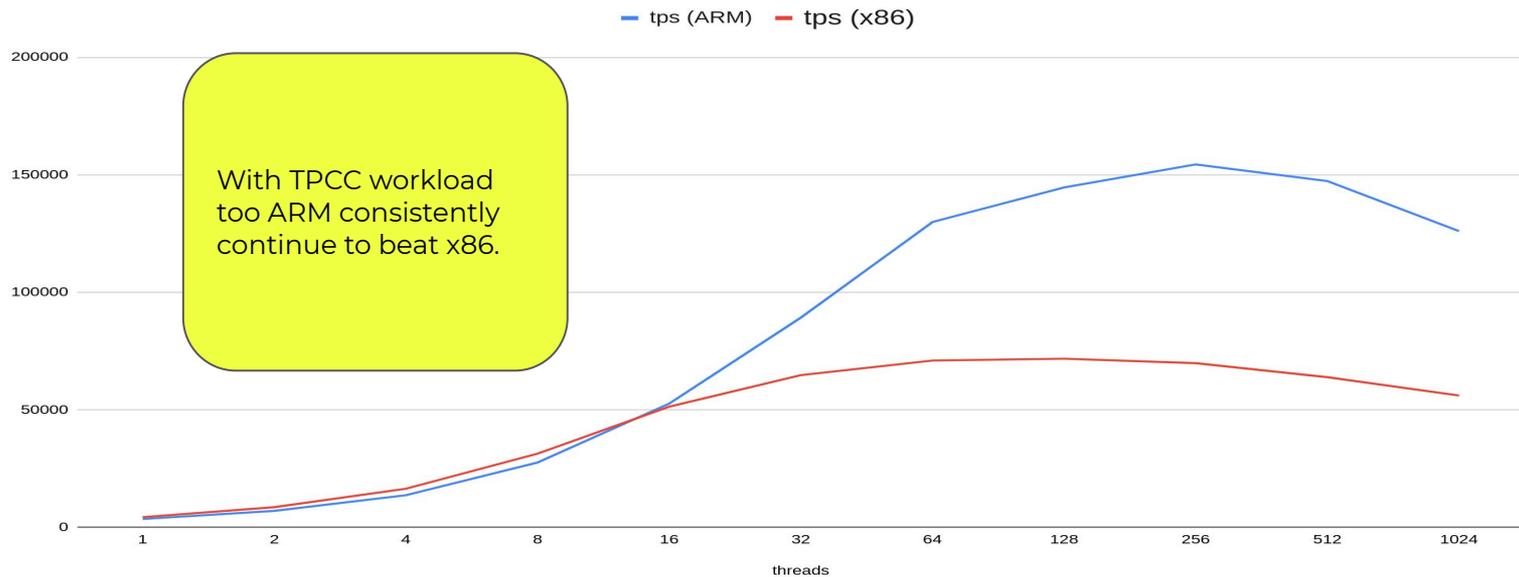
TPCC-benchmark (10 tables * 100 warehouse (scale)) - MDB-10.5.5



How MariaDB flares on ARM ?

- TPCC-benchmark (CPU bound workload)

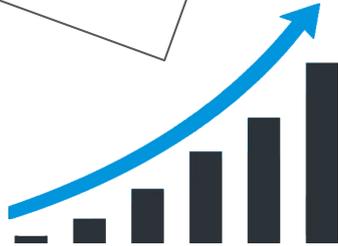
TPCC-benchmark (10 tables * 100 warehouse (scale)) - MDB-10.5.5



How MariaDB flares on ARM ?

- ARM continue to scale (thanks to more cores) for the same cost thereby generating higher tps/USD leading the effective cost saving.

Buy bigger machine at an extra cost



Switch to use ARM keeping cost same

Cost of machine (monthly)					800 USD	
1024-threads	upd (ARM) - TPS	tps/usd	upd (x86) - TPS	tps/usd	Projected saving (%) (with ARM)	
10.4.13 (CPU)	96137	120	39596	49	145	
10.4.13 (IO)	18432	23	6398	8	188	

Agenda



- ARM ecosystem
- MariaDB and ARM
- How MariaDB flares on ARM?
- State of Ecosystem

State of ecosystem (on ARM)



State of ecosystem (on ARM)



- Why aim for completion?
 - While the missing parts can still run on x86 (and interact with server using MySQL/MDB protocol) I think it would be good for end-user to eventually migrate and run everything on ARM especially from cost saving perspective.
- Supporting tools/components of ecosystem
 - There are supporting tools and components of ecosystem like Kubernetes, Vitess, Migration tools, etc... Support of them on ARM should be considered too from completeness perspective.
- Community support will help drive ecosystem completion.
 - PMM was recently compiled and run on ARM (by Percona Engineer).
 - There is JIRA ticket for producing MaxScale packages for ARM ([MXS-3065](#))
 - Connectors on ARM.

Development continues...



- MariaDB continue to consider/develop ARM compatibility/effects while developing server features.
- DB-ARM Community is growing and we continue to contribute back
 - Performance Enhancements
 - Bugs
 - Wider ARM64 compatible hardware adaptability
- If you have issues using MDB on ARM or wish to contribute please join the community channel [#mariadbbonarm](#)
<https://mariadb.zulipchat.com/#narrow/stream/118759-general/topic/mariadbbonarm>

Let's make #MDB great again (on ARM this time).

Thanks for attending

Follow/Connect

Twitter: [#mysqlonarm](#)

Blog: <https://mysqlonarm.github.io/>

Github: <https://github.com/mysqlonarm>

