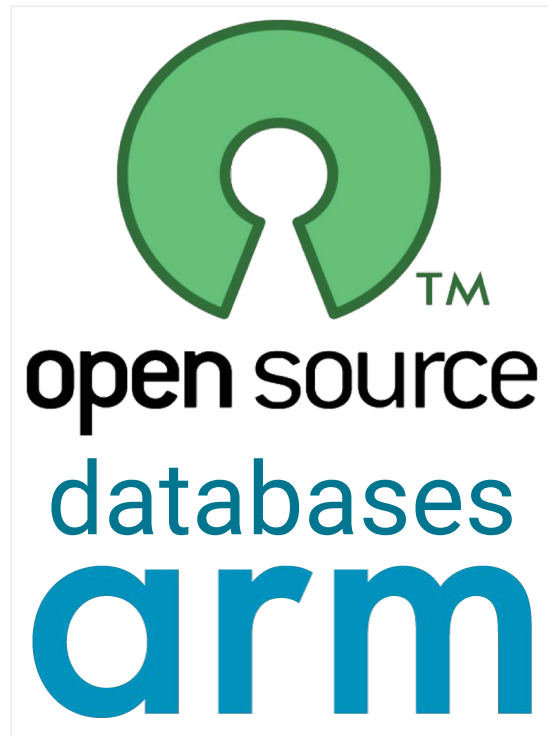# open-source databases and arm

Krunal Bauskar
(driving db on arm initiative)

quick word about me

- working in mysql/db space for more decade now.
- past worked with oracle/innodb, percona, yahoo! labs, teradata, etc..
- currently working @ huawei on #dbonarm initiative.
- trying to make open source databases and their ecosystem optimal for arm.
- blog/tweet/follow
  - https://mysqlonarm.github.io/
  - #mysqlonarm

agenda

- why arm?
- why databases on arm?
- state of open-source databases on arm
- challenges porting/running databases on arm
- future work

agenda

- **why arm?**
- why databases on arm?
- state of open-source databases on arm
- challenges porting/running databases on arm
- future work

why arm? - expanding market share

dominate mobile processor market with more than 90% share

growing market share in network equipments (estimated to touch 65% (from 32%) in next few years)

market leader in IoT devices with 90% share.

cars & automobiles (self-driving cars) (75% market share and growing)

Fast catching up in data-center space with estimated to capture 25% of the cloud-share in next few years.

why arm? - advantage over competitors

Lower cost of ownership
(lesser power consumption better performance per watt)

Massive parallelism (more cores) better throughput

open licensing model (allow multiple organizations to innovate)

leading organizations launching their variant of arm chip (huawei, amazon, apple, microsoft, oracle (ampere), google)

growing software ecosystem with regular releases for all leading softwares.

why arm? - next gen chip

next generation computing with SoC (combines cpu, graphics, memory, power module).

next generation computing with ARMv9 improved security and AI-enabled. (arm CCA).

next generation computing with cloud enable product line [neoverse] (96/128 cores per socket).

Next generation integrated ecosystem from input devices to processing devices (mobile, sensor, network, server).

Next generation developers/users are already widely using arm in form of educational kits/android/iOS/etc...

why arm? - challenges

exploiting software to use more cores in turn more numa nodes.

Industry still catching up on next gen features like scalable vector extension (neon).

tuning software for different arm architectures is challenging.

availability of arm based machines just started to catch up.

arm is still being looked upon as mobile class processor.

why arm? - why should you consider it?

agenda

- why arm?
- why databases on arm?
- state of open-source databases on arm
- challenges porting/running databases on arm
- future work

why databases on arm?

databases are compute intensive and can take advantage of more compute power for more throughput.
(improvement in range of 50-100%)

With arm offering now available on cloud, price advantage could be exploited (standalone, db-ha setup).
(estimated saving of 30%-50%)

improving software/library support. all os provider now has port (with tuned optimization) for arm including majority of the 3rd party libraries.

porting is simpler. tips and tricks to tune software on arm is now available in public forum with real-life examples.

full stack db ecosystem available on arm with majority of the open source db provider releasing regular packages on arm.

agenda

- why arm?
- why databases on arm?
- state of open-source databases on arm
- challenges porting/running databases on arm
- future work

state of open-source databases on arm

state of open-source databases on arm

feature-set
are all the features supported on new platform

performance
do we get on-par/better performance.

community support
do we continue to have community support?

ecosystem
are all available tools present in new environment

state of open-source databases on arm

mysql on arm

mysql on arm

official/regular releases from mysql/oracle (starting 8.x)

ha-features like binlog, group replication supported.

already scales better on arm. efforts continue to optimize it further (especially for more numa and new features).

mysql releases surrounding ecosystem tools like shell, router, connectors on arm helping support ecosystem completeness.

strong community support with active developers, users. blog, article, bug-fixes, reviews. (#mysqlonarm channel on community slack)

offered on on centos, rhel, oracle-linux. (distro coverage needs improvement and docker images are missing)

mysql on arm

complete ecosystem available on arm.

full stack application based on mysql ecosystem can be now enabled on arm.
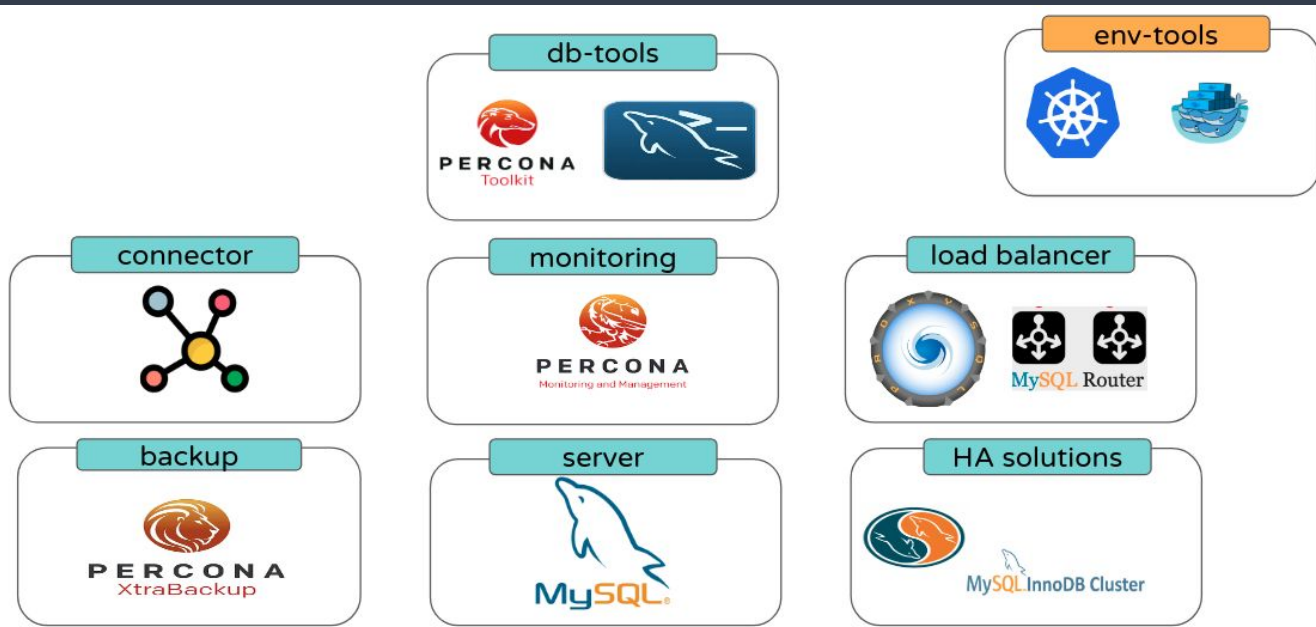
mysql on arm

mysql-8.0.24
sysbench: uniform

cost-performance-model.
cost normalized resource
allocation.

arm consistently beats
x86 for higher scalability
with differences touching
2x.

(for same cost more tps).

mysql on arm

mysql-8.0.24
sysbench: uniform

cost-constant-model.
resource normalized cost
allocation. (arm is 50+%
cheaper)

arm is on par with x86
beating it for higher
scalability despite of
lesser cost.

(for lesser cost
on-par/better tps).

mysql on arm

- oracle/mysql investing time and efforts integrating arm patches from community.
- also, revived 8.x interest on numa scalability like sharded lock, scalable log-sys, etc… helps arm, given its has more cores.
- recently (8.0.24) all atomics were moved to use standardize c+11 atomics that also helped fix/optimize the memory order (from default to optimal).
- support improved/newer compiler (gcc-10 has support for intrinsic atomics (lse) for arm).

state of open-source databases on arm

mariadb on arm

mariadb on arm

+ official packages from mariadb-server with all supported features.

+ Lot of arm specific improvement there-by helping scale mariadb on arm better on arm (recently touched a qps of 1.6 millions with 4 numa arm server).

+ active efforts being made to enable complete ecosystem on arm through community contribution.

+ strong and growing community with lot of user interested in running mariadb on arm. (already user has tried and reported issues (that are fixed)).

+ improved distro support centos, rhel, ubuntu, debian. arm evaluation is part of the mainline ci/cd for each commit.

mariadb on arm

complete ecosystem using open source software could be enabled on arm.

tools in categories are expanding.
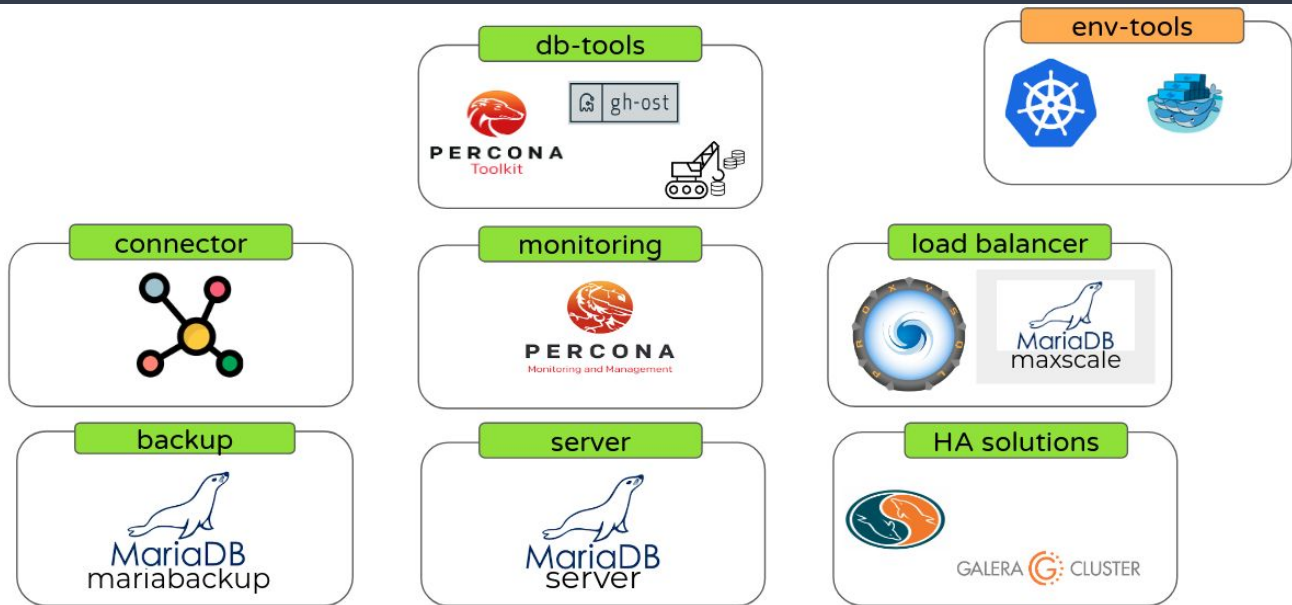
mariadb on arm
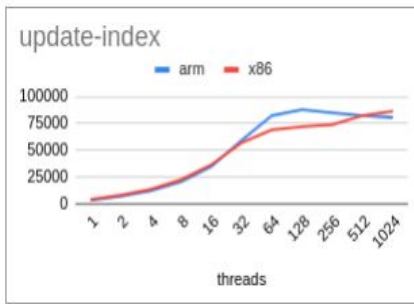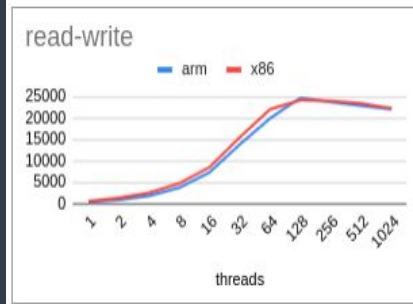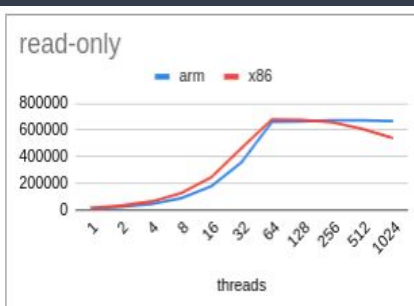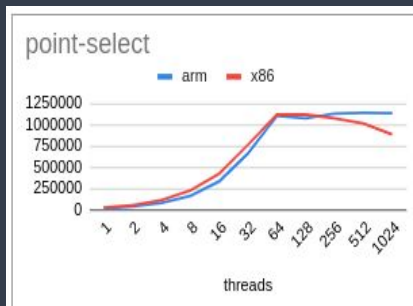

mariadb-10.6 (wip)
sysbench: uniform

cost-performance-model.
cost normalized resource
allocation.

arm consistently beats
x86 for higher scalability
with differences touching
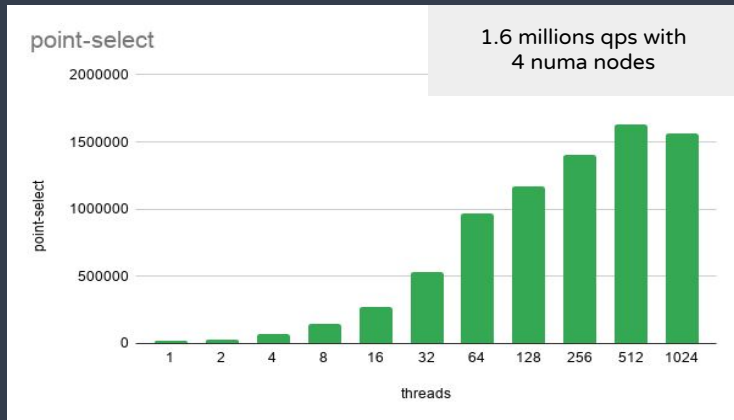2x.

(for same cost more tps).

mariadb on arm
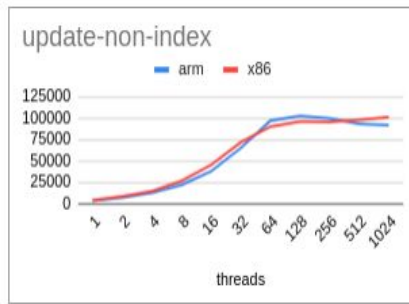
mariadb-10.6 (wip)
sysbench: uniform

cost-constant-model.
resource normalized cost
allocation. (arm is 50+%
cheaper)

arm is on par with x86
beating it for higher
scalability despite of
lesser cost.

(for lesser cost
on-par/better tps).

mariadb on arm

point-select

1.6 millions qps with 4 numa nodes

read-only

1.1 millions qps with 4 numa nodes

in mariadb-10.6, efforts made to help optimize mariadb-server for more numa nodes, resolving some critical numa bottlenecks helping it scale to touch 1.6 m qps (point-select) and 1.1 m qps (read-only) with 4 numa nodes.
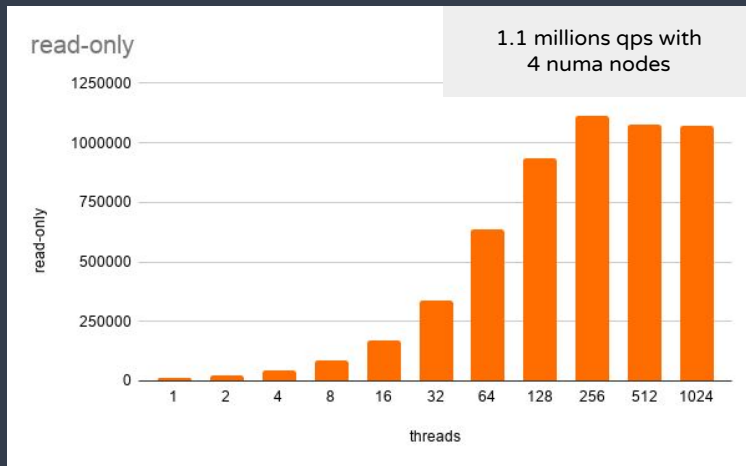
mariadb on arm

- mariadb has accepted lot of community contributions around arm helping it scale better (on arm).
- recent efforts are being made with 10.6 to further improve server performance especially for more numa nodes (numa scalability introduces different set of challenges).
- all new improvements/features undergo arm evaluation to ensure it doesn't regress on arm.
- ecosystem, distro support continue to improve for arm.
- mariadb cluster on arm is also being evaluated by community.

state of open-source databases on arm

percona on arm

percona on arm

community evaluated. works on arm. no official packages yet but since upstream compatible, core features works on arm.

scale better on arm. again, upstream inherited.

complete ecosystem like backup, monitoring, toolkit, etc… all of it works on arm (community evaluated).

enjoy pretty good community support. one of the earliest db to get tried on arm.

flexibility, openness, community support, ecosystem tools will help support percona on arm. looking forward for official packages from percona repo.

percona on arm

full stack ecosystem using majority of the percona products (community evaluated).
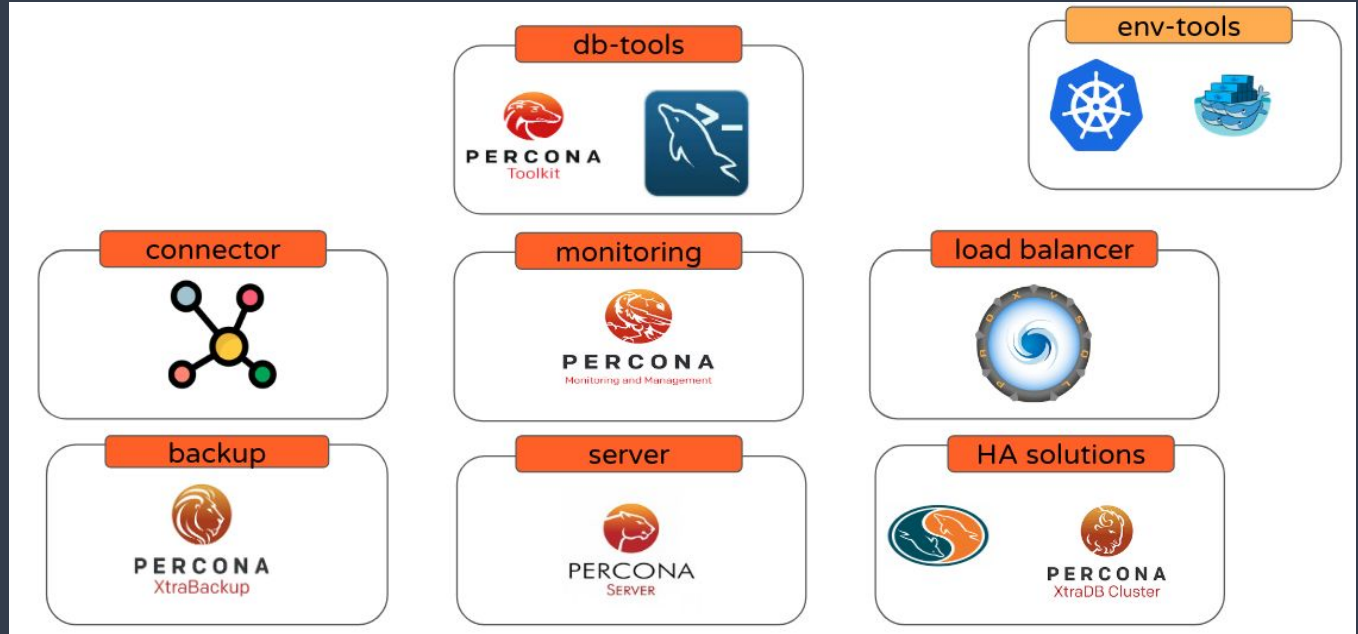
percona on arm

point-select

read-only

read-write
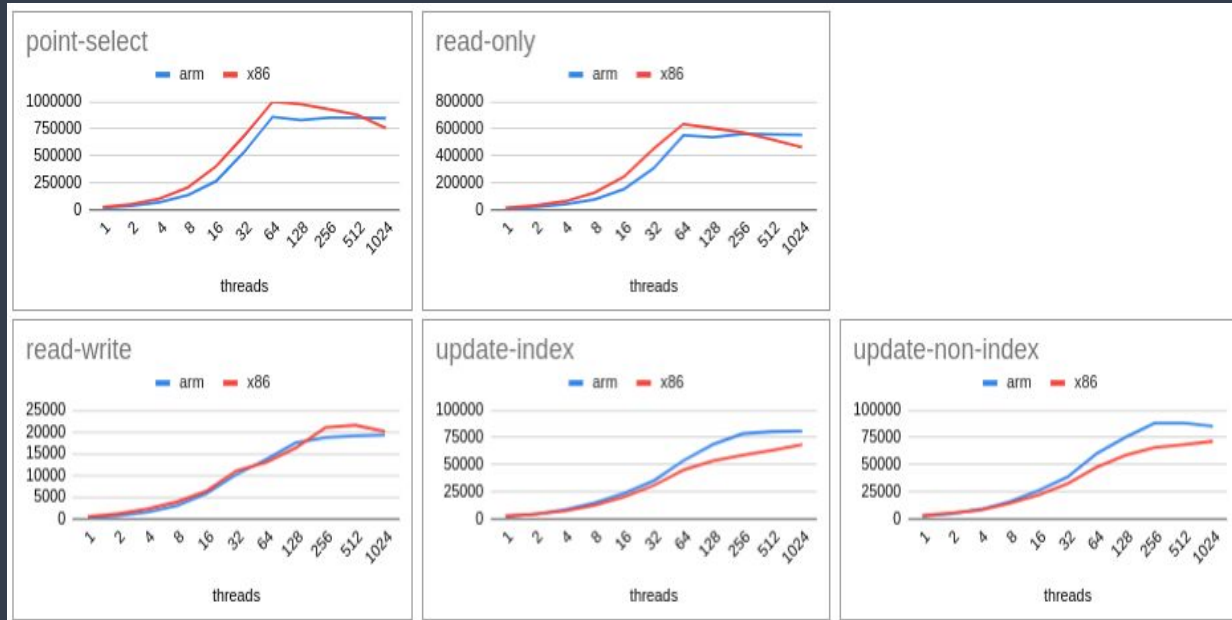
update-index

update-non-index

percona-server-8.0.22
sysbench: uniform

cost-performance-model.
cost normalized resource
allocation.

arm consistently beats
x86 for higher scalability
with differences touching
2x.

(for same cost more tps).

percona on arm

percona-server-8.0.22
sysbench: uniform

cost-constant-model.
resource normalized cost
allocation. (arm is 50+%
cheaper)

arm is on par with x86
beating it for higher
scalability despite of
lesser cost.

(for lesser cost
on-par/better tps).

percona on arm

- percona-server add-on features like audit-plugin, encryption, thread pool, improved monitoring/counters, full text improvements, etc... needs more porting/testing on arm.
- support for other ecosystem components will further aid with easier adoption of db on arm.
- numa specific improvements could also further explored to help further improve performance of percona server on arm (over and above upstream).

state of open-source databases on arm

postgresql on arm

pgsql on arm

official packages available directly from pgsql community.

community continue to contribute patches. still long way to go before we can say it is optimized for arm.

regular releases of ecosystem tools for arm. ecosystem is actively expanding with participation from different organization (for respective tools).

active community with interest for supporting arm. gaining traction especially post apple enable arm.

offered packages on centos, ubuntu/debian. (should able to build it on other distros too).

pgsql on arm

pgsql has mix of different ecosystem tools from lot of different vendors. fortunately most of the popular tools are already supported on arm allowing user to run complete full stack of pgsql on arm.
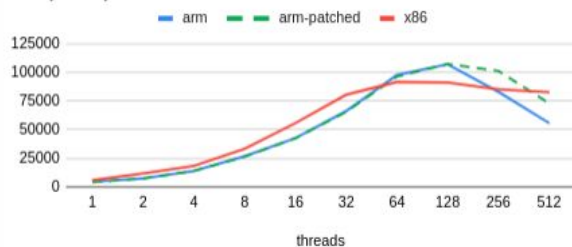(listed some representative tools. quite likely tool you are using is already ported to arm).

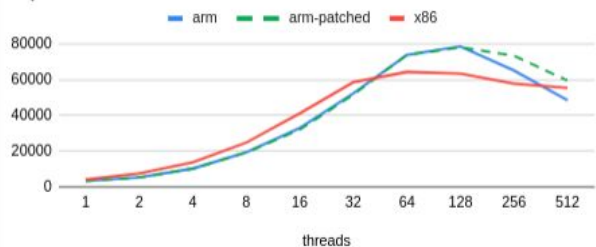# pgsql on arm
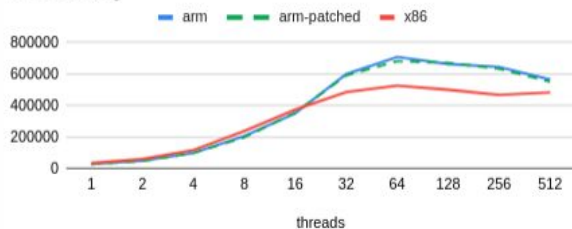
pgsql-13.2
pgbench: ro, rw, tpcb

cost-performance-model.
cost normalized resource
allocation.

arm continue to scale
better. observed drop for
higher scalability can be
taken care by patched
version (switching to use
cas vs tas).

(for same cost more tps).

pgsql on arm

pgsql-13.2
sysbench: uniform

cost-performance-model.
cost normalized resource
allocation.

mix-results with arm beating
x86 in some workloads and
lagging in some cases. again
the patched version
continue to help read-write
workload.

pgsql on arm

- pgsql has accepted some patches around arm enabling it to easily release packages for arm back in 2020.
- on optimization front, still there are things to look into including community contribution. some of the active performance patches are stuck in process.
- ecosystem has pretty good response and rising popularity to run pgsql on cloud is helping increase adoption of pgsql on arm (aws, hc).

state of open-source databases on arm

clickhouse on arm

clickhouse on arm

+ works on arm (official packages not available but supported as per documentation. recommended to build from source).

+ scale on-par with x86 (for 50% lesser cost).

+ ecosystem tools (especially drivers jdbc/odbc) are available on arm. monitoring tools, external engines works.

+ growing community support including ecosystem tools. (scope to optimize things further).

+ olap is about huge data and clickhouse can very well harvest parallel cores.
arm is best fit given the cost advantage and massive parallelism it has to offer.
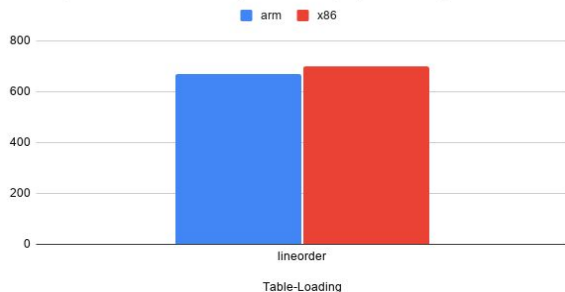
clickhouse on arm

majority of the ecosystem tools (offering the core functionality like backup, ha, load-balancing, monitoring, etc…) are already supported on arm. some trailing tools especially around loading data could be compiled for arm.
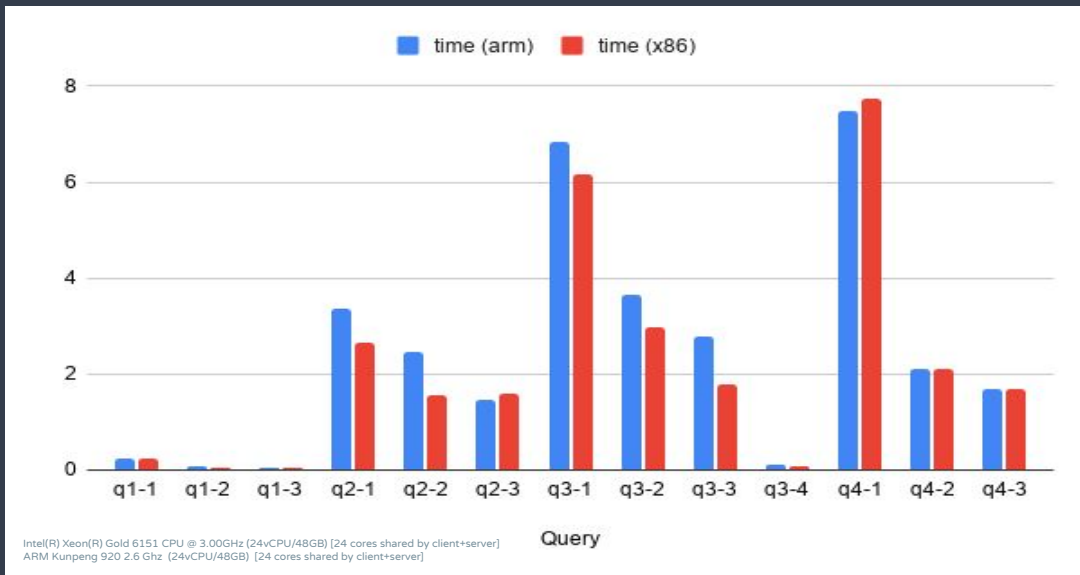
clickhouse on arm

loading time for lineorder (-s 150 roughly 100 GB)
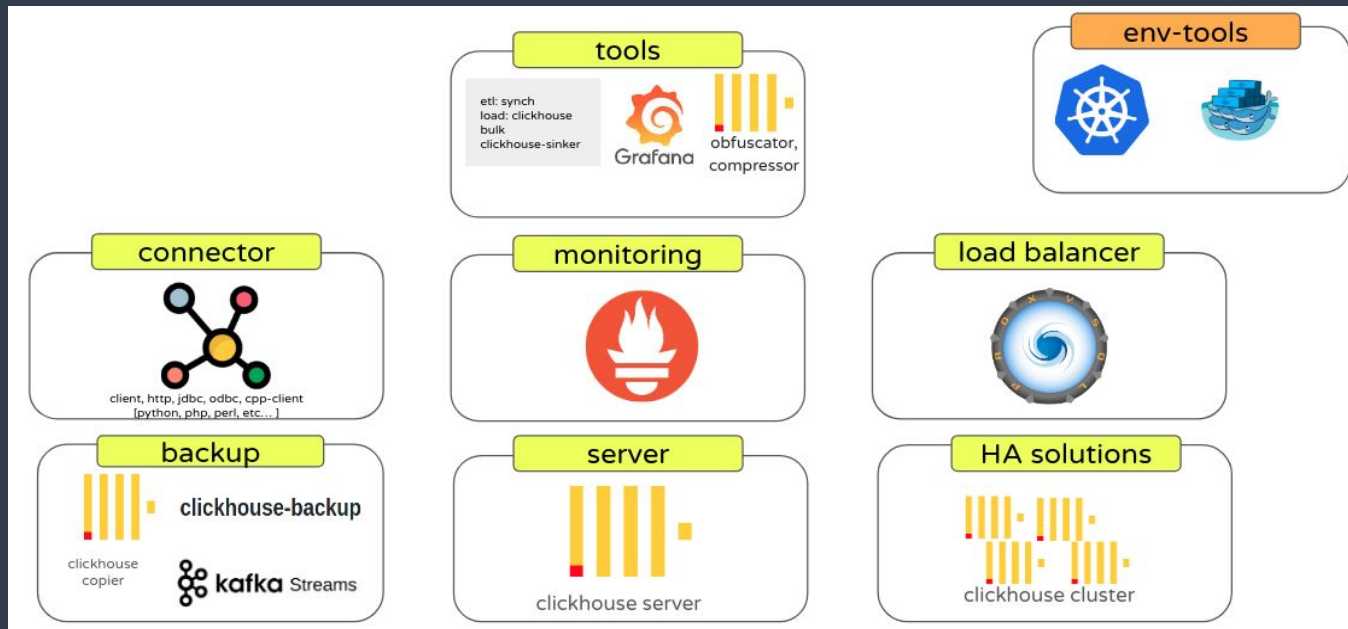
Loading time is slightly better with arm

query has mixed responses with comparable performance (for 35-40+% lesser cost).

Intel(R) Xeon(R) Gold 6151 CPU @ 3.00GHz (24vCPU/48GB) [24 cores shared by client+server]
ARM Kunpeng 920 2.6 Ghz (24vCPU/48GB) [24 cores shared by client+server]

pgsql on arm

ClickHouse implements parallel data processing and uses all the hardware resources available. When choosing a processor, take into account that ClickHouse works more efficiently at configurations with a large number of cores but a lower clock rate than at configurations with fewer cores and a higher clock rate. For example, 16 cores with 2600 MHz is preferable to 8 cores with 3600 MHz.

- arm is best suitable hardware for clickhouse, given it needs more but lesser powerful cores.
- clickhouse use vector based query execution. arm offers pretty good smid/neon support that is further set to improve with each release. it could further help accelerate clickhouse.

state of open-source databases on arm

mongodb on arm

mongodb on arm

mongodb releases packages for arm. this also means it is officially supported and all in-built components works.

mongodb scale well on arm with on-par performance in some cases. (cost parity helps arm score).

since majority of the ecosystem tools are offered by mongodb most of them are available on arm through official repo.

community is still catching up. usage is current limited to pi4 kind of setup.

promising improvement is expected. also, helps support heterogeneous db setup.

mongodb on arm

mongodb officially support arm so most of the inherent tools which are default and popular choice too are already available on arm. community addition further help improve the coverage. user can enable full stack mongodb setup on arm.
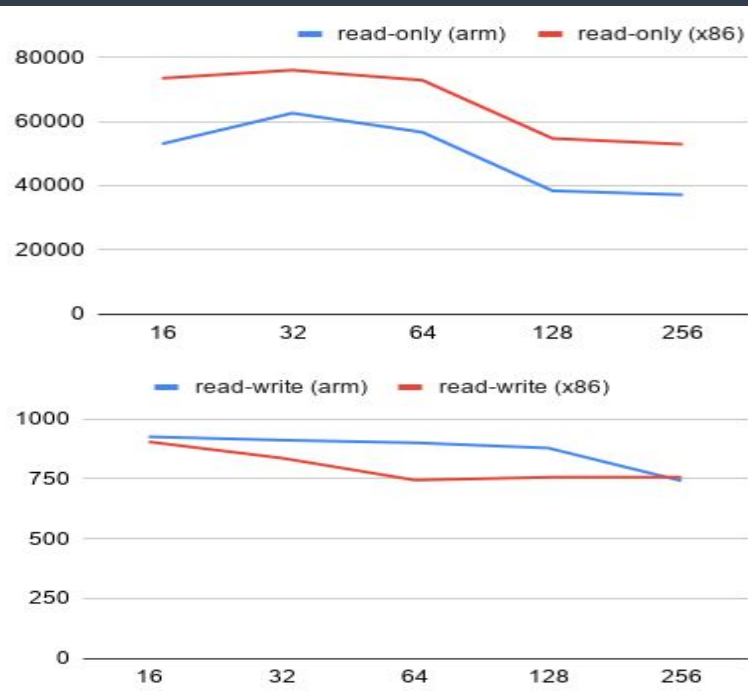
mongodb on arm

load/insert tps — 30m: load, 3m: read, 300: read/write

mixed-results (with x86 scoring in read-only and insert/load case and arm scoring in read-write case).
(note: arm comes with 35% lesser cost)

read-only (arm) — read-only (x86)

read-write (arm) — read-write (x86)

mongodb on arm

- mongodb usage is more prone toward sharding for storing large amount of web-scale data. given the scalability demand, arm cost and performance effectiveness make it an optimal choice.
- mongodb has support for arm including some optimization but there is still a scope for improvement for further optimization (based on performance result).
- missing ecosystem components could be eventually added especially dependent on npm/nodejs kind of framework.
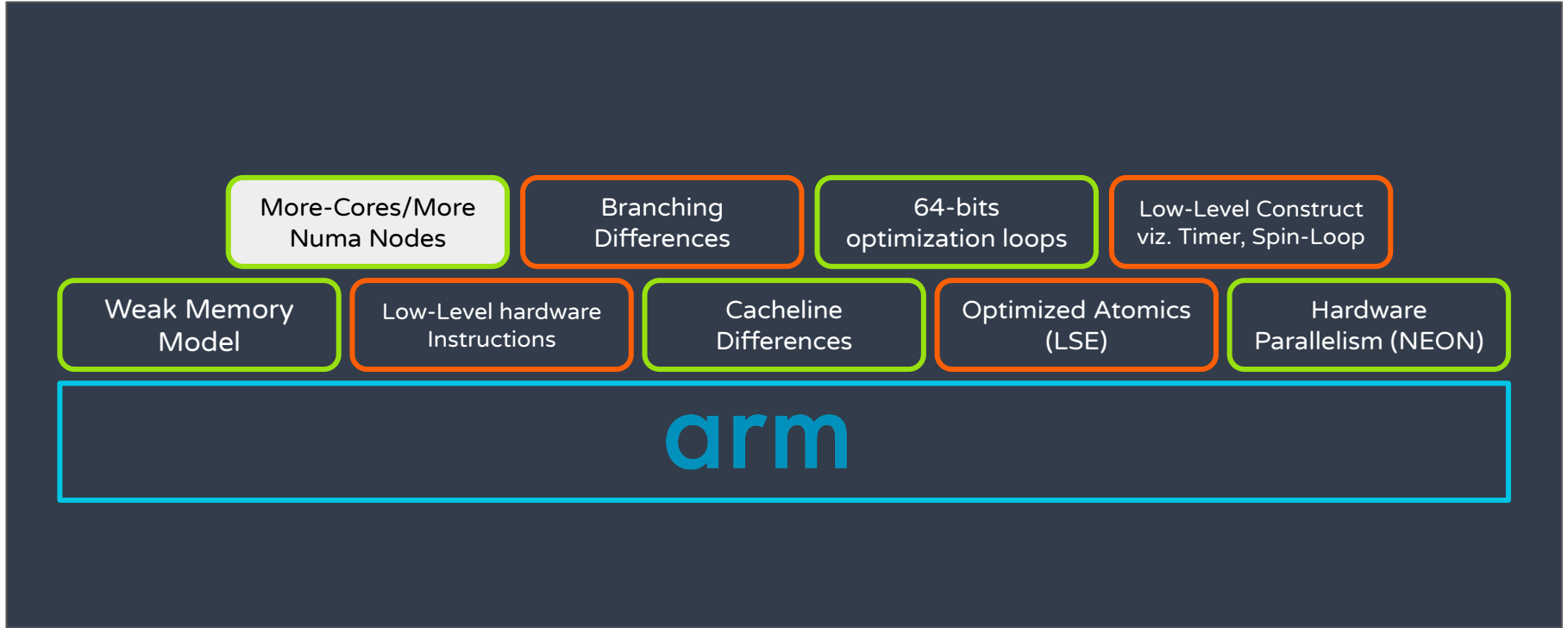
agenda

- why arm?
- why databases on arm?
- state of open-source databases on arm
- challenges porting/running databases on arm
- future work

challenges porting/running databases on arm
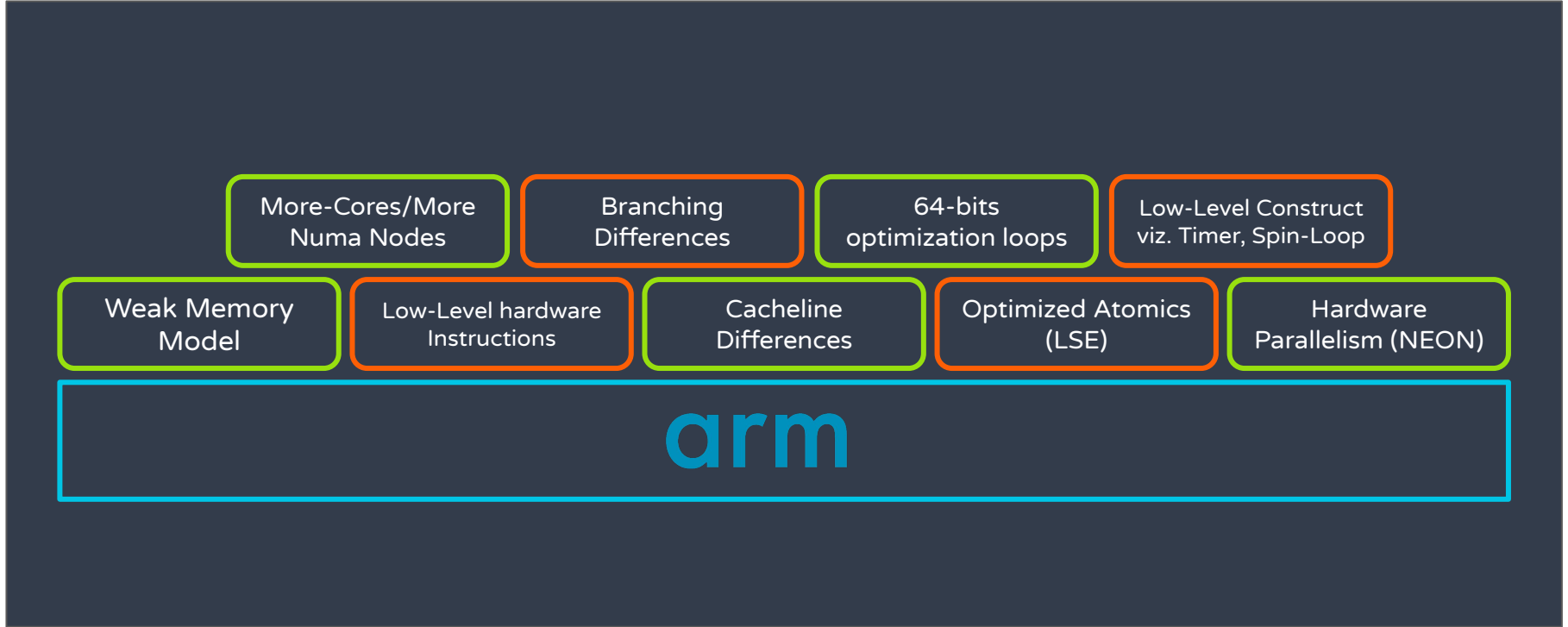
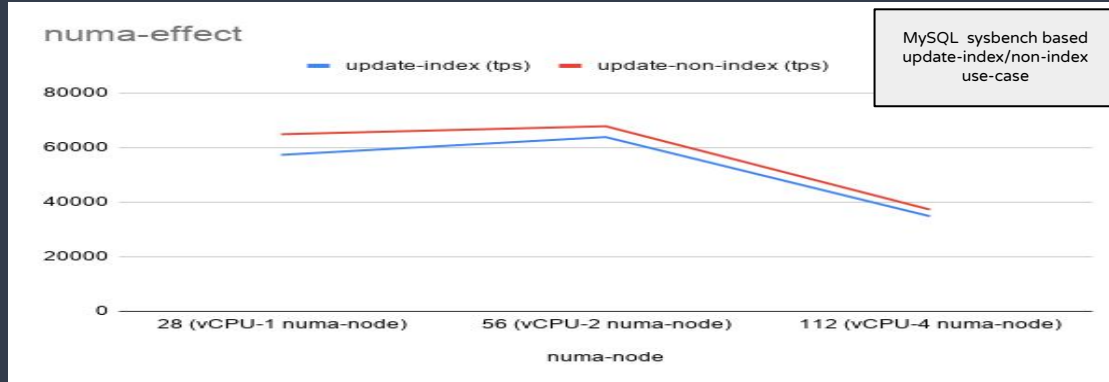# challenges porting/running databases on arm
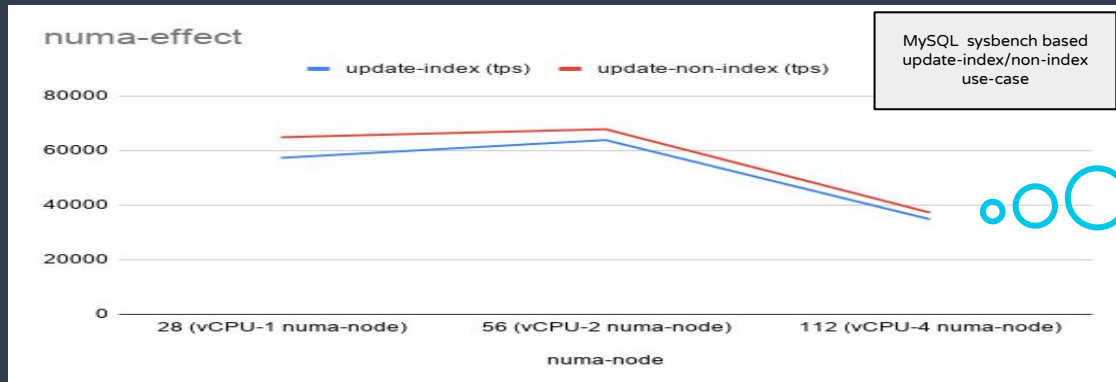
challenges porting/running databases on arm

challenges porting/running databases on arm

Scalability bottleneck

NUMA bottleneck

challenges porting/running databases on arm

agenda

- why arm?
- why databases on arm?
- state of open-source databases on arm
- challenges porting/running databases on arm
- future work

future work

- numa bottleneck and other server optimizations.
- looking beyond server. optimizing ecosystem tools.
- increasing ecosystem coverage (getting more tools added to ecosystem).
- improving on distro coverage.
- scenario based use-cases (especially around full stack usage).
- making poster complete (more database on arm).

connect with us/community

- willing to contribute or planning to port your tool to arm. please connect.
- mail:
  - krunalbauskar@gmail.com
  - mysqlonarm@gmail.com
- blog:
  - https://mysqlonarm.github.io/
- slack/community/forum channel:
  - #mysqlonarm
  - #mariadbonarm
  - pgsql-hackers@postgresql.org
- tweet: #mysqlonarm

Thanks Organizer and Sponsors for making this possible and providing the opportunity to share the ideas.