Apache Arrow community meeting notes

The Apache Arrow community meeting is held biweekly on Wednesday at 9:00am Pacific / 12:00 noon Eastern Time, which is 17:00 UTC during US Standard Time and 16:00 UTC during US Daylight Time. To add these meetings to your calendar, use one of these links:

- Google Calendar
- iCal

This Google Doc is globally readable, but edit access is limited for safety. Use the **Request edit access** button to ask for edit access. Add items under the **Discussion** heading to propose items for discussion in the next meeting.

For meetings prior to 2023-01-18, meeting notes were sent directly to the dev@arrow.apache.org mailing list.

2025-11-05

Attendees

- Scott Routledge
- Bryce Mecum
- Martin Prammer
- Lucas Paes
- Alina Li
- Rok Mihevc
- Michael Chavinda
- Raúl Cumplido
- Rossi Sun
- Pawel Wojciechowski
- Zhao Elton

Discussion

- Timestamp with offset proposal (ML thread, draft format PR)
    - Question about whether format proposal should include encodings (yes) and whether initial implementation needs to be complete (no)
- Followup on https://github.com/apache/arrow/pull/43661/files Parquet Reader applying casts before slicing (Scott)
    - Looking for feedback on current review comments
- Flight SQL ODBC move from MinGW to MSVC CI https://github.com/apache/arrow/issues/48019 (Alina)

    - Currently looking into it;
    - Question about whether it's okay to enable flight+flightsql in the MSVC CI: Yes. There doesn't seem to be a reason they're disabled.

2025-10-22

Attendees

- Raúl Cumplido
- Ian Cook
- Antoine Pitrou
- Martin Prammer
- Rok Mihevc
- Bryce Mecum
- Rossi Sun
- Jade Wang

Discussion

- Arrow 22.0.0 release status
    - First release candidate had an issue with Parquet Variant
    - This is fixed in the new release candidate
    - Discussion about timing of Python 3.14 wheels
        - The previous release of Arrow didn't have 3.14 wheels. 22.0.0 will. Users complained about PyArrow not having a release that supported 3.14
            - https://github.com/apache/arrow/issues/47438
            - https://github.com/apache/arrow/issues/47700
        - The 22.0.0 release will include freethreaded 3.14 wheels for all OSs
- RunsOn new runners
    - We used to have access to CUDA runners but that was recently lost
    - There is an ongoing discussion about using RunsOn (https://github.com/apache/arrow-nanoarrow/issues/814)
    - This will allow us to use our AWS credits
    - Rok and Jacob are working on this
    - It's unclear whether we can run RunsOn in ASF infra, at least without a long delay before an audit is completed and this is approved
        - Because the tool requires repo admin permission and it's not open source
    - Possible other solution is to use a separate GitHub organization outside the ASF to host the non-OSS components of RunsOn
        - This seems like the most direct and fast path to using RunsOn
        - Main downside of this is that integration with GitHub will not be as seamless

- Reproducible builds
    - This is a step we are taking to help safely automate releases
    - Raúl is seeing a small binary difference in the builds depending on the build path; unclear what's causing it
    - Raúl has tried to solve this e.g. using the instructions at https://reproducible-builds.org/docs/build-path/
    - Help available through the list at https://alioth-lists.debian.net/cgi-bin/mailman/listinfo/reproducible-builds
    - Raúl will post on Zulip about this soon and ask for help solving this
- Type annotations almost ready for review (https://github.com/apache/arrow/pull/47609)
    - Rok has managed to eliminate most errors
    - Please take a look if you're interested
- ADBC foundry status
    - David Li gave a talk at Arrow Summit that discussed this
    - Slides are at https://github.com/lidavidm/arrowsummit2025/blob/main/Where%20We%E2%80%99re%20Going%2C%20We%20Don%E2%80%99t%20Need%20Rows.pdf
- Conbench homepage very slow and sometimes errors out (error "503 Service Temporarily Unavailable"); potential fix here: https://github.com/conbench/conbench/pull/1614
    - Errors are not deterministic
    - Short-term priority is to get this PR merged
        - Bryce and Jacob can help get this done
- Ian talk about ADBC from CMU seminar
    - https://www.youtube.com/watch?v=TjlmNGNx77E

2025-10-08

Attendees

- Martin Prammer
- Ian Cook
- Rok Mihevc
- Scott Routledge
- Raúl Cumplido
- Felipe Carvalho
- Jacob Wujciak-Jens
- James Duong
- Alina Li

Discussion

- (Scott) Followup on https://github.com/apache/arrow/pull/43661 to discuss C++ benchmark added
    - Reviews needed (Antoine, Xuwei, Rossi, Ben, …)
- (Raúl) Arrow 22.0.0 status
    - Code freeze was yesterday
    - There are several open issues with the 22.0.0 milestone
    - Biggest open issue is the Python 3.14 wheels (issues with protobuf on musl causing segfault)
        - https://github.com/apache/arrow/issues/47438
        - Discussions ongoing on Zulip
    - Other minor issues (e.g. Windows runner disk is filling up)
- (Alina) ODBC Build & Testing discussion
    - https://github.com/apache/arrow/pull/47689
    - Alina will remove the Ruby builds
    - Flight SQL is not currently enabled on macOS builds; Alina will open separate PR to try to enable in future
    - Tests currently use (a) mock SQLite server, and (b) live remote server (Dremio Docker container). (a) is artificially simple so we want to have (b) but it's unclear how to add this
- Discussion with Martin about the F3 paper
    - https://db.cs.cmu.edu/papers/2025/zeng-sigmod2025.pdf

2025-09-24

Attendees

- Ian Cook
- Scott Routledge
- Bryce Mecum
- Alina Li
- Michael Chavinda
- Raúl Cumplido
- Rossi Sun
- Jacob Wujciak-Jens

Discussion

- Please take a look at type annotations for PyArrow draft proposal: https://github.com/apache/arrow/pull/47609. General PyArrow annotations discussion here: https://github.com/apache/arrow/discussions/45919.

    - This is the culmination of a long discussion
    - There is strong community interest in this
    - This will be a topic for discussion at the Arrow Summit in Paris next week
- (Scott) Update on https://github.com/apache/arrow/issues/43660 from last meeting with a more concrete use case, was wondering if it fell under the scope of the benchmarks
    - This issue arose when creating a PyArrow Dataset using LargeString type for the string columns, on top of Parquet files that used String for the string columns
- C data interface still the best first integration?
- (Alina) GLOG availability on Windows
    - Found this old commit: https://github.com/apache/arrow/commit/7104d64ff2cd6c20e29d3cf4ec5c58bc10798f66#diff-1bba462ab050e89360fd88110a689e85ee037749cea091a1848ab574381d3795R310
    - glog is deprecated https://github.com/apache/arrow/issues/47465
- PyCapsule Interface (stable?)
    - Raúl will start a discussion on the mailing list about removing the experimental flag on this
- Arrow v22.0.0
    - Blockers
    - Plan to add wheels with support for Python 3.14
    - Planned feature freeze on October 6 (shortly after the Arrow Summit in Paris)
-

2025-09-10

Attendees

- Nic Crane
- Raúl Cumplido
- Bryce Mecum
- Ian Cook
- Scott Routledge
- Rok Mihevc
- Rossi Sun

Discussion

- (Scott) Discuss Overheads while casting String to Large String during Parquet read with small batch size:
    - https://github.com/apache/arrow/issues/43660
    - Possibly solved by https://github.com/apache/arrow/pull/46532?
        - If this didn't solve the issue: get the PR up to date and @-mention the Parquet reviewers
- (Scott) Quick question about current status of Parquet variant support

- (Raul) Arrow v22
    - Planned feature freeze on October 6 (shortly after the Arrow Summit in Paris)
    - Raúl plans to serve as release manager
    - Please tag any blockers and plan accordingly
- (Ian) Follow up about incorrect result issue in Arrow v21
    - https://github.com/apache/arrow/issues/47234
    - Underlying cause was an MSVC bug
    - Issue is fixed now; new source build will not have bug
        - Using the latest conda-forge builds should also fix the issue
    - No patch release or downstream package updates are currently planned
    - We will prominently mention this in communications about the v22 release

2025-08-27

Attendees

- Raúl Cumplido
- Bryce Mecum
- Scott Routledge
- Victor Tsang
- Alina Li
- Ruoxi Sun
- Ian Cook & Matt Topol
- Justin G
- Rob Scales
- Martin Prammer
- Nic Crane
- Jade Wang
- Rok Mihevc

Discussion

- (Bryce) Discuss this improvement to visual identity work?
    - https://github.com/apache/arrow/issues/38484
    - If you feel inspired, please chime in on the issue
- (Scott) Discuss Adding and exec_context as an option for Scanners
    - https://github.com/apache/arrow/issues/43694
- (Matt) PR for Parquet-style variant canonical extension type coming soon
    - See https://github.com/apache/arrow/issues/46908
- (Alina) Demo for Flight SQL ODBC driver installer
- (Alina) Q: difference between ruby.yml and cpp.yml
- (Ian) Update about plans to shift driver development to a contrib GitHub org

    - Proposed name “ADBC Driver Foundry”
    - Currently discussing with main stakeholders; will share more details soon
- (Jade) follow up question about the ADBC foundry plan

2025-08-13

Attendees

- Martin Prammer
- Ian Cook
- Nic Crane
- Victor Tsang
- Alexander Taepper
- Rok Mihevc
- Alina Li
- James Duong
- Ruoxi Sun
- Bryce Mecum
- Rob Scales

Discussion

- PyArrow 21.0.0 incorrect result bug
    - https://github.com/apache/arrow/issues/47234
    - Still no definitive diagnosis or solution
- Arrow Flight SQL ODBC demo with PowerBI
- Discussion about build/linking issue on Intel Mac
    - Similar to https://github.com/apache/arrow/issues/46077
- Discussion about PR to improve signature matching constraint for parameterized types
    - https://github.com/apache/arrow/pull/47297

2025-07-30

Attendees

- Rok Mihevc
- Bryce Mecum
- Ian Cook

- Alina Li
- Rossi Sun
- Neal Richardson
- Rob Scales
- Jacob Wujciak-Jens
- Jade Wang

Discussion

- Removing Gemfury for PyArrow nightlies in 22.0.0 release
    - https://github.com/apache/arrow/issues/43904
    - Nightlies will continue to be available from scientific-python, see Python Development
- PyArrow 21.0.0 incorrect result bug
    - https://github.com/apache/arrow/issues/47234
    - Bryce will ping a few contributors early today
    - We should consider an email to user@ once we get a better sense of the scope of the bug (just fill_null, broader?)
    - https://github.com/apache/arrow/pull/46963 (xsimd) landed 3 weeks ago, touched the windows wheel build script (just looking for what changed in this area esp. For windows)
- Benchmarking transition
    - Infrastructure is complex, code is open source, but some infrastructure is opaque and currently controlled by Voltron Data; Jacob and Rok are investigating options for improving this
    - Options include using GHA runners (free but potentially noisier results), using AWS (now that we have credits), using CodSpeed as frontend
        - Rok is in touch with the CodSpeed founders; they are interested to collaborate; they offer the service free for open source
        - Could use their instrumentation for microbenchmarking, it just counts instructions not wall time. Though switching would break the history (if we care)
    - Anyone else who is interested: contact Jacob and Rok
    - Discussion about whether we should consider scaling back the scope of benchmarking to allow the infrastructure to be less costly and more manageable
    - We could consider stopping running benchmarks on every commit if that will help to reduce cloud costs sufficiently to make this more affordable
    - Separate but related issue: Voltron Data also hosts some self-hosted GPU runners for CUDA tests; this infrastructure is also complex and unmaintained
- Pyarrow-stubs experiment @EuroPython - https://github.com/rok/arrow/pull/42
    - Options: (1) keep pyarrow-stubs and transfer maintainership; (2) bring pyarrow-stubs into PyArrow; (3) change to using auto-generated type annotations
    - Rok and others have pursued (2), in Rok's fork at first; this will require a code donation to upstream
    - Discussion welcome

        - Looking for downstream app devs who are interested to test this
            - Rok will find other people to ping
- Apache Flight SQL ODBC basic query demo
    - Alina presented a demo using an ODBC test client on Windows
- Databricks Go ADBC driver PR, https://github.com/apache/arrow-adbc/pull/2998
- Source code/repo structure for ADBC drivers
- Python DB API discussion (https://discuss.python.org/t/extend-dbapi-with-apache-arrow-support/94397/14)

2025-07-16

Attendees

- Martin Prammer
- Rossi Sun
- Raúl Cumplido
- Nic Crane
- James Duong
- Bryce Mecum
- Dewey Dunnington
- Alenka Frim
- Rok Mihevc
- Jakob Wujciak-Jens

Discussion

- Arrow Monorepo 21.0.0 release status
    - Vote is up: https://lists.apache.org/thread/0k08fmzonyj00rtwt63fpt9c8hkkcyg9. Please verify and vote. Will close today/tomorrow.
    - Blog post PR is up: https://github.com/apache/arrow-site/pull/668. Please review and contribute where you can.
- Benchmarks
    - https://github.com/voltrondata-labs/arrowbench/pull/139
        - R benchmarks maybe not working?
    - Current state of resources and process docs a bit spread out, low clarity on where things live
    - Raúl and Jacob to start documentation on current status and we can go from there
    - Related
        - We're planning to move JS benchmarks out: https://github.com/apache/arrow-js/issues/203
- EuroPython and PyArrow type stubs sprint (Raúl, Alenka, Rok attending)
    - Sprint: https://www.ep2025.europython.eu/sprints/

- Arrow Summit
    - CfS: https://sessionize.com/arrow-summit-2025/
    - Maybe no submissions yet? Closes July 26
    - Social Media Links
        - Bsky: https://bsky.app/profile/arrow.apache.org/post/3ltebp333xk2q
        - LI: https://www.linkedin.com/feed/update/urn:li:activity:7347890035305713666

2025-07-02

Attendees

- Ian Cook
- Alenka Frim
- Antoine Pitrou
- Matt Topol
- Rok Mihevc
- Martin Prammer
- Nic Crane
- James Duong
- Raúl Cumplido
- Jacob Wujciak-Jens
-

Discussion

- Arrow monorepo 21.0.0 release status update
    - Feature freeze was on July 1
    - 7 open issues (really 6) with PRs that are close to mergeable: 21.0.0 milestone (open issues)
        - Notable issues:
            - Severe performance regression for one kernel: https://github.com/apache/arrow/issues/46777
                - PR with fix: https://github.com/apache/arrow/pull/46859
            - Problem initializing kernels: https://github.com/apache/arrow/pull/46922
                - Complicated by linking issues
            - Work to support Parquet variant
                - https://github.com/apache/arrow/issues/45937

            - We should tag a different issue that better reflects what exactly is being merged in 21.0.0, and/or consider renaming the issue(s) to avoid misunderstandings
        - Will continue with release once those are resolved
        - Discussion about whether the changelog currently lists PR titles, issue titles, or commit titles
- Arrow Summit - 3 Oct 2025 - Paris
    - Co-located with PyData Paris
    - Single track (one room)
    - Selection Committee has been selected by the Selection Committee Selection Committee
    - The event has been approved by ASF management, so it can now be announced publicly
    - The call for proposals will start now and last until ~end of July
    - Sessionize setup is underway
        - https://sessionize.com/apache-arrow-summit-2025/
            - Do not share this yet, but this will be the link that we share to solicit CFPs
- Arrow board update due soon
    - See email from Neal
    - https://docs.google.com/document/d/1fSYEmfGtmzjt1prHkdRoyxwGMmcX3onAStWArfiSBis/
    - Reminder that the focus of this is community engagement, ecosystem changes, and other higher-level changes—not the minute details of what work has been completed
    - Discussion about whether we could use the LLM-generated dashboard to help initially populate this
- ~~Arrow~~ Parquet variant canonical extension type
    - Matt has implemented this in the Arrow Go implementation, but we need another Arrow implementation before we can close the vote and merge the PR to add this to the spec docs
    - Discussion about naming
        - This would be the first canonical extension type to begin with “parquet.” instead of “arrow.”
    - Discussion about whether the shredded and non-shredded variants should be separate extension types
        - No, because it is not a binary distinction; a variant column can be partially shredded
    - Discussion about stability of the type over time
- Proposed new Parquet interval type
    - Discussion about merits of the proposed type
    - Discussion about reasons for hesitance

2025-06-18

Attendees

- Bryce Mecum
- Alenka Frim
- Martin Prammer
- Jade Wang
- Nic Crane
- Jacob Wujciak-Jens
- Ian Cook
- Matt Topol
- Rossi Sun
- Rok Mihevc

Discussion

- Databricks ADBC driver development
    - Started with C# driver development with Microsoft, targeting a release later this year
    - Now they're working with dbt on a Go driver for Databricks
        - The existing Databricks driver implementation in dbt's fork of ADBC uses an obsolete Databricks connector; the Databricks team wans to reimplement it
    - Databricks wants a good story for ADBC drivers for all languages
    - Difficult to develop drivers in apache/arrow-adbc because the Databricks developers do not yet have commit privilege
    - Possible solutions
        - Use a long-lived fork for ongoing development
        - Find an Arrow committer (or multiple) who will work closely with you
        - Work to earn committership
        - The Arrow community could establish an arrow-contrib GitHub organization outside of the apache organization in GitHub, which would give us more discretion to give write access to non-committer contributors
    - Databricks wants to offer some features that are not a part of the ADBC interface
        - We are open to discussion about extending the API to include these
- Just FYI, Arrow monorepo 21 code freeze July 1 (see ml, milestone)
    - Let's be mindful of documentation changes with language implementations moving out (broken links etc.)
- Arrow Summit - 3 Oct 2025 - Paris
    - Selecting a committee still in progress
- AWS credits
    - https://lists.apache.org/thread/q33oofy2v3zpg9s9l8o0w68rmjr3ocsv
    - They have been granted

    - We receive them monthly
    - We have not yet started spending them
    - Good places to start spending them?
        - Large runners to speed up slow tests, e.g. integration tests
        - GPU instances (but that might be a lot of work)
        - For CI runners in the above-mentioned arrow-contrib organization (since we won't be able to use ASF GitHub resources there)
        - Other ideas?
- Need for more work on Conbench and Arrow benchmarking generally
    - General discussion on whether this work is interesting to the database research community
    - Martin is happy to serve as a point of contact for discussions about this
- Repo clean-up
    - Nic started an ML thread: https://lists.apache.org/thread/xx2t85h0zfoyhrj5z3hdjm0n7zrcl51r
    - Please reply with feedback, suggestions, ideas

2025-06-04

Attendees

- Alina Li
- Martin Prammer
- Rob Scales
- Ian Cook
- Bryce Mecum
- Raúl Cumplido
- Rossi Sun
- Antoine Pitrou
- Alenka Frim
- James Duong
- Rok Mihevc
- Jacob Wujciak-Jens
- Nic Crane

Discussion

- CMU involvement in Arrow variant work
    - Martin can serve as point of contact
- Alina delivered a demo of the Arrow Flight SQL ODBC connectivity on ODBCTest
- Split Arrow Compute kernels into its own Shared library
    - https://github.com/apache/arrow/pull/46261

    - Needs review from one of the maintainers of the R bindings - Nic has pinged Jon and Neal
- Arrow v21 release
    - We are about one month away from a planned feature freeze
        - July 1 might be a reasonable date
    - We will need a release manager volunteer
    - Can we prepare the blog post sooner?
        - The delay getting the blog post finished has in the past delayed the announcement of the release
    - Concern that Conan PR delays could delay release
    - JavaScript and Swift will be moved out of the monorepo in time for v21
        - .NET is TBD
- Dev docs not building?
    - Ian will open an issue for this
        - https://github.com/apache/arrow/issues/46712
    - Raúl suspects it could be a Cython issue
- Status of AWS credits application?
    - No updates yet
- Planned Arrow summit (to be held after PyData Paris 2025)
    - Employer/affiliation information needed
- Extension type support
    - There was a recent question about using compute functions on extension types, especially in the case where the extension type is a thin wrapper around one of the simple core types: https://github.com/apache/arrow/discussions/46671
        - Input requested on the discussion so we can work out the possible paths forward and their complexity

2025-05-21

Attendees

- Rob Scales
- Rok Mihevc
- Ian Cook
- Bryce Mecum
- Andrey Shandybin
- Nic Crane
- Alina Li
- James Duong
- Neal Richardson
- Jacob Wujciak-Jens

Discussion

- Upcoming Arrow talks/events
    - Work is ongoing to plan Arrow Summit 2025, to be held after PyData Paris 2025
        - More news coming soon
        - We should add a blog post, share news on social media, and add a banner to the Arrow website
- ODBC driver PR is now merged
    - https://github.com/apache/arrow/pull/40939
    - Comments are being addressed in follow-up PRs
    - Question about why the button to approve CI workflow runs was not visible
        - This can happen because of an error in workflow YAML config
            - Jacob is looking at adding a pre-commit hook to detect this
        - If there are merge conflicts, it also will not run the CI workflows
- Mailing List Activity
    - Arrow Variant Discussion
        - https://lists.apache.org/thread/w06cxdojjcmry4m9vb0bo7owd1jsbtz5
    - Switch to C++ 20 Discussion
        - https://lists.apache.org/thread/48zlj0dn2y0f53y2k37qsr90y781wfnj
    - Vote to move Swift out of monorepo
        - https://lists.apache.org/thread/8zwy57k27dw7l2hrjx8qjnh33h9lbhb9
- Discussion about how to better link to different implementation source repos (instead of just the monorepo)
    - Ian to file issue, Nic and Alenka to take a look and work on this
- Feedback on Kapa AI bot
    - Generally positive
- New arrow DuckDB community extension being released this week
    - See past discussion: April 9 Meeting
    - https://duckdb.org/community_extensions/extensions/nanoarrow.html
    - (Currently published version of the extension doesn't have the new stuff Ian is talking about yet)
    - https://duckdb.org/community_extensions/extensions/arrow.html
    - Maybe move source code out of https://github.com/paleolimbot/duckdb-nanoarrow
        - One possibility is to create an arrow-contrib *Organization*, similar to Datafusion (https://github.com/datafusion-contrib/)
            - Look for discussions on DataFusion mailing list (or on the Arrow mailing list if this was created before DataFusion was a top-level project)
- Next meeting on June 4th - demo Arrow Flight SQL ODBC connectivity on ODBCTest

2025-05-07

Attendees

- Rok Mihevc
- Bryce Mecum
- Matt Topol
- James Duong
- Dewey Dunnington
- Ian Cook
- Neal Richardson
- Alina Li
- Rob Scales
- Jacob Wujciak-Jens
- Srikanth Manamohan Nadukudy

Discussion

- Arrow 20.0.0 release status
    - Release is almost complete but not yet announced because blog post is still in the works
        - Release blog post: https://github.com/apache/arrow-site/pull/646
        - Contributions needed!
    - Several post-release tasks are pending
- AWS credits application (Rok submitted an application as per process describe at https://aws.amazon.com/blogs/opensource/aws-promotional-credits-open-source-projects/)
    - Application is pending
    - Current Arrow spending estimate on self-hosted runners is roughly $600 but costs would grow if we need GPU instances
    - Do other cloud providers offer similar credits?
        - We're not sure but if so we can apply for it
    - What are next steps after we receive credits?
        - Create an organization account
- Arrow ODBC Driver updates
    - Alina and Rob are working with JB to create a PR on his branch
    - JB's PR provides the backend; Alina and Rob are also working on the frontend
    - There is a PR in draft currently; should be ready for review by end of May
    - Questions on vcpkg.json and lint bypass
        - The vcpkg.json changes should appear in JB's PR
        - Compliance with linting rules would require many changes in vendored code (originating from Dremio's Flight SQL ODBC repository); can we bypass the linter?

        - If it is vendored code that will continued to be maintained upstream, it's fine to keep it as is and skip the linting because that makes it easier to keep it updated.
        - If the Arrow repo will become the canonical place where this code will be maintained, then we should get the linter to pass
- Upcoming Arrow talks/events
    - Neal will be speaking about Arrow at useR!
        - https://user2025.r-project.org/

2025-04-23

Attendees

- James Duong
- Rossi Sun
- Bryce Mecum
- Rok Mihevc
- Antoine Pitrou
- Raúl Cumplido
- Sri Nadukudy
- Nic Crane
- Jacob Wujciak-Jens
- Dewey Dunnington
- Weston Pace

Discussion

- Arrow monorepo 20.0.0 RC2
    - Vote thread is open on the dev ML https://lists.apache.org/thread/8xroqpf9wd7gx7kgb60tj515tkxxs5hh
- AWS CI resources update (Rok)
    - There's a submission form we can fill in: https://aws.amazon.com/blogs/opensource/aws-promotional-credits-open-source-projects/
    - Need to identify an AWS Root account that can own the credits
    - Rok will investigate creating an org account for Arrow if we can
- Upgrading Arrow C++ to C++20 (Antoine)
    - Experimental PR: https://github.com/apache/arrow/pull/45445
    - Stuck on packaging jobs, volunteers welcome, needs a rebase
- Migrating from rapidjson to simdjson (Antoine)
    - https://github.com/apache/arrow/issues/35460

    - Incoming ODBC driver is building with RapidJSON (and will continue to be)
- Splitting JS into its own repo
    - ML discussion https://lists.apache.org/thread/qpjt8ypmw1h8j1445kmn8s7wqml0cwoy
    - Weston has a PR for a JS Flight impl, is wondering about the timing on the split
    - C# split is in progress and may happen first, see https://lists.apache.org/thread/n5jxzhy5qbdj3qjgnkgckbnc0ktrt757
- Raúl working on migrating non-core Compute kernels into separate SO
    - PR: https://github.com/apache/arrow/pull/45618
    - Raúl will continue working and ping people for reviews
- Arrow C++ Support policy issue: https://github.com/apache/arrow/issues/46002
    - Needs someone to draft language and post to ML for discussion
- PyArrow types
    - Topic discussed on PyArrow Community Call (https://docs.google.com/document/d/1ioiJdEYf5mJwQ-rOjzjPYCeHTjOhAPo5ppUHy6iBrxU/edit?tab=t.0#heading=h.jpxt32n00bns), see Notes 2025 tab
    - Related GH issue: https://github.com/apache/arrow/issues/32609

2025-04-09

Attendees

- Bryce Mecum
- Alina Li
- Ian Cook
- Dewey Dunnington
- Rossi Sun
- Andrey Shandybin
- Jacob Wujciak-Jens
- Rob Scales
- Sri Nadukudy

Discussion

- Arrow Monorepo 20.0.0 Release Status
    - RC0 created, likely need an RC1
        - GH-46075: [Release][CI] Binary verification fails when downloading assets from Github release
        - GH-46067: [CI][C++] Flatbuffers version mismatch
        - See #dev > Arrow Monorepo 20.0.0 Release Plan
- Work on the ODBC PR
    - JB working on the main PR

    - More PRs coming after that from Alina, Rob
- New Arrow community extension for DuckDB
    - https://duckdb.org/community_extensions/extensions/nanoarrow.html
    - In early release; please try it
    - Will be renamed “arrow” later
    - Ultimately might be moved into an “arrow-contrib” organization in GitHub which is technically not governed by all the ASF processes which allows us to release more flexibly and give commit access to DuckDB maintainers who are not Arrow committers
        - Similar to https://github.com/datafusion-contrib
        - A contrib org like this might help with other purposes; for example Dewey cited an example of someone who wanted to contribute an Apache Cassandra driver for ADBC
- Issue/discussion about official support policy
    - https://github.com/apache/arrow/issues/46002

2025-03-26

Attendees

- James Duong
- Sri Nadukudy
- Alina Li
- Matt Topol
- Dewey Dunnington
- Saurabh Singh
- Ian Cook
- Robert Scales
- Rossi Sun
- Jacob Wujciak-Jens
- Hao Xu

Discussion

- Reminder: Arrow Monorepo 20.0.0 feature freeze is April 1, 2025 (see mailing list)
    - Currently 232 issues in milestone, 28 open. (see monorepo 20.0.0 milestone)
    - Use priority: blocker and critical fix tags as needed
- Revisiting discussion from last meeting about possible challenge of getting enough votes for releases cut from separate non-monorepo repos
    - We are mostly bound by ASF policy, and the concept of release votes are very central to the principles of the ASF

    - For now best course of action seems to take steps to get more votes but not propose to fundamentally change the process
    - Jacob shared context on how the ASF is considering how to modernize policies to allow automatic signing and other common release practices, but doesn't expect the process to fundamentally change in the near term
        - Some work we are doing for reproducible releases will help to prepare for this
    - Jacob mentioned upcoming EU legislation intended to create safer open source releases, avoid supply-chain attacks, etc.
- Alina and Rob from Improving are working on the ODBC driver for Arrow
    - First goal is to help JB get this finished and released
- Question about interaction between Arrow and Spark
    - Dewey noticed that the Arrow serializer in the Spark Connect JAR emits non-interoperable Arrow IPC streams; is fixing in Sedona (which is downstream of Spark); is looking to upstream to Spark

2025-03-12

Attendees

- Rok Mihevc
- Bryce Mecum
- Ian Cook
- Dewey Dunnington
- Sri Nadukudy
- Rossi Sun
- Matt Topol
- Andrey Shandybin

Discussion

- Arrow monorepo 20.0.0 release plan (Bryce)
    - Plan is for April 1 feature freeze
    - Jacob to act as release manager, Bryce will assist with PMC-only tasks
    - Currently 29 open issues in the milestone
- Debrief about spinning out some implementations to new repos
    - How did it go for Go (in v18)?
        - Some early hiccups but it's been smooth recently
        - But getting enough votes for the Go-specific release has been more challenging (compared to in the past where the votes were for the whole monorepo release, of which Go was a part)
- Arrow-Go v18.2.0 release (Matt)

    - Release vote is on the mailing list
        - Almost enough votes but please vote/comment
        - Assuming enough binding votes, Matt will do the release this week
- PRs that need review (ideally before next releases):
    - Dewey: Implementation of new Parquet geo types: https://github.com/apache/arrow/pull/45459
- Status update about Amazon donating AWS instances for CI/benchmarking
    - Next step: Rok is going to email David Nalley at AWS this week

2025-02-26

Attendees

- Rok Mihevc
- Matt Topol
- Saurabh Singh
- Bryce Mecum
- Alenka Frim
- Ian Cook
- Rossi Sun

Discussion

- [GO][Arrow] PlainFixedLenByteArrayEncoder behaves different from DictFixedLenByteArray #71
    - Question about what intended behavior should be
        - General consensus seems to be that the other encoders should error the same way the DictFixedLenByteArrayEncoder does
        - Suggested next steps:
            - Look at Git history to make sure the seemingly inconsistent behavior was not added for a good reason
            - Check the behavior of other Arrow implementations
- Potentially using Amazon donated AWS instances for CI/benchmarking
    - This was discussed at FOSDEM
    - Matt needs to send an email to David Nalley (who works at AWS and is the EVP of ASF) to organize the approval/receipt
        - Rok will do this in place of Matt
        - Link to apply for credits is https://aws.amazon.com/blogs/opensource/aws-promotional-credits-open-source-projects/

        - But you need an AWS account and a valid payment method to apply for credits; we should use an ASF account not someone's personal account; so we want to reach out first
        - Mention involvement of Jarek Potiuk
        - Request will be for EC2 spot instances and S3 storage
        - We should ask Jarek how Airflow uses spot instances for CI
- Generic pyarrow presentation for docs to be used at meetups etc
    - Rok will open a PR in the apache/arrow-experiments repo to kick this off
- Acero Swiss join PR needs review: https://github.com/apache/arrow/pull/45612

2025-02-12

Attendees

- Matt Topol
- James Duong
- Bryce Mecum
- Joe Yen
- Saurabh Singh
- Dewey Dunnington
- Logan Riggs
- Rossi Sun
- Ian Cook
- Sri Nadukudy
- Rok Mihevc

Discussion

- Arrow monorepo 19.0.1 Release Update (Bryce)
    - Vote thread should be posted just after this call
- Geometry/Geography in Parquet PR(s?)
    - https://github.com/apache/arrow/pull/45459
    - GeoParquet is just metadata on top of regular Parquet, whereas this adds first-class geometry and geography logical types (based on byte array physical types)
        - This improves query efficiency (because better statistics)
        - In practice, older Parquet readers will reject Parquet files with these new geo types; in theory they could read them as byte arrays
    - This increases the need for Arrow to have first-class or canonical extension types for geometry/geography (instead of the "community extension type" that GeoArrow is)

    - Gang Wu implemented the parquet-format change
        - PR: https://github.com/apache/parquet-format/pull/240
        - Commit: https://github.com/apache/parquet-format/commit/94b9d631aef332c78b8f1482fb032743a9c3c407
        - Parquet version number will need to be incremented before the next release
    - parquet-java implementation is being developed now
    - Dewey will open issues to implement this in the other implementations
- Arrow variant type
    - Someone proposed this a while ago, but they did not champion it and it didn't go anywhere
    - What's holding up Variant in Parquet is a debate about how to handle erroneously written files (at least as of the last Parquet meeting)
    - Variant parquet-cpp PR - https://github.com/apache/arrow/pull/45375#issuecomment-2649229837
    - Discussion about adding a variant Arrow type: https://lists.apache.org/thread/lsmkmxsp1qvjzn497z582hjm0w8hmg0n

2025-01-29

Attendees

- Bryce Mecum
- Ian Cook
- Saurabh Singh
- Raúl Cumplido
- Nic Crane
- Rok Mihevc
- Sri Nadukudy
- Bruce Irschick
- Joris Van den Bossche
- Kazim Mir
- Dewey Dunnington

Discussion

- FOSDEM this weekend
    - Many Arrow community members attending

    - Committer meetup (Saturday 1 February in Brussels) https://docs.google.com/document/d/13IN60zN-uX6TUOkvTa8IP3frvi_bnNY36G7W4IO-7AU/
    - BoF Session: Future of the Arrow ecosystem BOF
- 19.0.1 release
    - Freeze planned tomorrow (30 Jan)
    - Bryce will manage with Kou helping
    - Needed because of https://github.com/apache/arrow/issues/45283
    - https://github.com/apache/arrow/milestone/68?closed=1
        - Main issue that created need for patch release: A recent patch caused an incompatibility with Parquet files written by the Rust Parquet implementation
    - Postmortem discussion
        - Timeline
            - - **November 13, 2023:** parquet-format merged [PARQUET-2261: add statistics for better estimating unencoded/uncompressed sizes and finer grained filtering by emkornfield: https://github.com/apache/parquet-format/pull/197
            - - **July 26, 2024:** arrow-rs merged pr [Add support for level histograms added in PARQUET-2261 to \`ParquetMetaData\` by etseid https://github.com/apache/arrow-rs/issues/5022
            - - **December 18th, 2024:** arrow-cpp merged [\[C++\]\[Parquet\] Implement SizeStatistics · Issue #40592 · apache/arrow](https://github.com/apache/arrow/issues/40592) for issue [\[C++\]\[Parquet\] Implement SizeStatistics · Issue #40592 · apache/arrow](https://github.com/apache/arrow/issues/40592)
            - - **January 16th, 2025:**
            - - Bug reported: [\[Python\]\[C++\]\[Parquet\] "OSError: Repetition level histogram size mismatch" when reading parquet file in pyarrow since 19.0.0 · Issue #45283 · apache/arrow](https://github.com/apache/arrow/issues/45283)
            - - PR with fix merged: [GH-45283: \[C++\]\[Parquet\] Omit level histogram when max level is 0 by wgtmac · Pull Request #45285 · apache/arrow · GitHub](https://github.com/apache/arrow/pull/45285)
        - What could we have done to catch this before the 19.0.0 release?
            - Test that would have caught it (but that we didn't run) was: writing test Parquet files live instead of pulling pre-written "golden" files from https://github.com/apache/parquet-testing
                - This seems like a problem best solved in the Parquet project
            - Low-hanging fruit actions
                - Add an example of the bad Parquet file to bad_data in parquet-testing

                - More comprehensive approaches to avoiding this problem in the future
                    - Create a matrix of implementations to test against
                    - Unclear who is available to take this on
                - Next steps?
                    - Email the Parquet dev list. Bryce to draft and pass around via Ian.
        - Any other fixes that should go into 19.0.1?
            - https://github.com/apache/arrow/issues/45304 (was mislabeled as a feature not a bug, but should go in)
- Any PRs need review?
    - https://github.com/apache/arrow-go/pull/263
- Next nanoarrow release (0.7)
    - Dewey plans to do this in the next month or so
    - Dewey cannot sign the release and would prefer not to ping a PMC member with keys on the day of the release to be able to upload the release
        - Will work with Kou for now
    - Release highlights: Zstd compression support, bug fixes
-
    -

2025-01-15

Attendees

- Rok Mihevc
- Dewey Dunnington
- Bryce Mecum
- Ian Cook
- Curt Hagenlocher
- Matt Topol
- Raúl Cumplido
- Rossi Sun
- Michael Chavinda
- Antoine Pitrou
- Saurabh K Singh
- Jacob Wujciak-Jens
- Xuwei Fu

Discussion

- 19.0.0 release update

    - RC0 vote is still open: https://lists.apache.org/thread/s94w2cdb85k11wo9q0nhosc5o7vzlx3p
    - Vote will close in ~1hour, RC0 looks like it will pass, I'll start on post-release tasks today so ADBC can restart their release
- Upcoming Arrow meetups
    - FOSDEM committer meetup (Saturday 1 February in Brussels) https://docs.google.com/document/d/13IN60zN-uX6TUOkvTa8IP3frvi_bnNY36G7W4IO-7AU/edit
    - GoodData Meetup (Wednesday 5 February in Prague) https://www.gooddata.com/resources/goodmeetup-7-how-to-get-the-fastest-analytics-with-apache-arrow/
- Question about C Device Interface
    - Currently marked as experimental, not a lot of implementations of it, no extensive cross-implementation testing
    - Will eventually supersede the C Data Interface, but unclear how we get there
    - New features like the async support is only being added to the Device interface
    - It is nearly trivial to implement support for the C Device Interface just by adding a small wrapper around the C Data Interface
    - Should make this clearer in the docs
        - https://github.com/apache/arrow/issues/44535
    - Should add the C Device Interface to the implementation status page
    - Should create an umbrella issue tracking the implementations of the C Data Interface that should be modified to support the C Device Interface.
    - The tradeoff is between encouraging adoption of the device-aware ABI over the non-device-aware ABI and ultimately reducing the total size of ABI surface area, vs. making life easier for implementers
- Continuous benchmarking of Arrow Go implementation
    - Benchmark workflows were broken/disabled when the Go implementation moved out of the monorepo
    - There is a ticket open to add them back: https://github.com/apache/arrow-go/issues/85
    - Discussion about Conbench maintenance obstacles
        - Previously all/most benchmarks were running on Voltron Data-controlled hardware, but now they are mostly running on AWS or GH runners which can create noisy neighbors problems
        - We should pursue goal to make this infrastructure more openly visible and maintainable by Arrow devs

2025-01-01

No meeting today in observance of the New Year's Day holiday.

2024-12-18

Attendees

- Bryce Mecum
- Ian Cook
- Michael Chavinda
- Dewey Dunnington
- Joris Van den Bossche
- Rossi Sun
- Rok Mihevc
- Raúl Cumplido
- Jade Wang

Discussion

- Next meeting is January 1, 2025. Keep it or cancel it?
    - Informal vote says to cancel
    - Ian will email dev list to say it's cancelled
- Arrow 19.0.0 release planning (Bryce)
    - Planned feature freeze Jan 6 2025
    - Issues tagged with 19.0.0 milestone: https://github.com/apache/arrow/issues?q=is%3Aopen+is%3Aissue+milestone%3A19.0.0
        - We tag issues and minor PRs with the milestone
        - We don't tag non-minor PRs with the milestone because the associated issue is tagged
    - Email to dev list coming later today
    - 19.0.0 Milestone: https://github.com/apache/arrow/milestone/66
- Blog post PR coming soon on Arrow for query result transfer (Ian)
    - General discussion about Arrow for query result transfer
- Interest in implementing Flight SQL in a data warehouse system
    - Ian (ianmcook@apache.org) offered to discuss in a separate meeting. Email Ian if interested to join.
- Reference implementation for library maintainers (Michael)
    - We don't have anything exactly like this
    - We do have some integration testing infrastructure: https://arrow.apache.org/docs/format/Integration.html

2024-12-04

Attendees

- Rok Mihevc
- Ian Cook
- Joe Yen
- Matt Topol
- Saurabh K Singh
- Ruoxi Sun
- Jacob Wujciak-Jens
- Raúl Cumplido
- Joris Van den Bossche

Discussion

- Are there any Arrow meetups?
    - There was one at posit::conf last summer
    - There will be a “birds of a feather” event at FOSDEM in Brussels
- Ways for a C++ learning to make contributions to Arrow C++?
    - There are some issues tagged with the “good first issue” label (but they are not necessarily good first issues for someone new to C++)
    - C++ compute kernels are maybe a good place to start https://github.com/apache/arrow/issues?q=is%3Aissue+is%3Aopen+kernel+label%3Agood-first-issue
- Conbench sustainability and maintenance
    - I (Raúl) brought this topic to a Zulip chat, here:
        - https://ursalabs.zulipchat.com/#narrow/channel/180245-dev/topic/Conbench.20sustainability.20and.20maintenance
    - From my understanding there is currently no-one working on conbench. How should we proceed with those specific issues? What about longer term sustainability of benchmarks?
        - https://github.com/apache/arrow/issues/44883
    - There are two parts: runners (which run the benchmarks) and Conbench itself (the UI and so forth)
        - Runners are all on AWS at this point
        - Moving to GHA is not straightforward because it depends on hosts with stable performance (no noisy neighbors, etc.) for accuracy
            - But running on AWS greatly complicates maintenance
    - Is Conbench currently used anywhere outside of Arrow?
        - Yes, Velox is using Conbench
            - But they are using GitHub runners, not AWS
            - They have overcome the noisy neighbors problem by using larger runners

        - There were some proof-of-concept projects to use it elsewhere including in pandas, but these never got past that phase
    - What is the level of effort to move everything out of Voltron Data-dependent resources?
        - <discussion about this>
    - Alternative tools like Conbench?
        - asv (airspeed velocity) is a similar project
        - https://codspeed.io is another
    - Crossbow is another piece that currently exists outside of the formal Arrow project (it's controlled by Voltron Data)
        - This could be moved all into Arrow somewhat more easily than Conbench, but it would ~double our usage of GHA (but still be under our cap)
- 18.1.0 release
    - Released and announced
    - Blog post PR merged
    - A few post-release tasks still pending
        - R package
        - Problem with dev version of cookbook https://github.com/apache/arrow-cookbook/issues/362
- 19.0.0 release
    - Bryce has volunteered to be release manager
    - Raúl will do 20.0.0

2024-11-20

Attendees

- Sri Nadukudy
- Ian Cook
- Marcus Hanwell
- Jacob Wujciak-Jens
- Matt Topol
- Raúl Cumplido
- Ruoxi Sun
- Dewey Dunnington
- Rishabh Maurya

Discussion

- 18.1.0 release
    - Release vote is on track to pass and Jacob plans to close the vote later today

    - Bryce plans to lead post-release tasks
    - Raúl and/or Kou will help with tasks that require PMC member permissions
        - For example uploading to PyPI
            - Side note: Jacob is looking into reproducible builds and automatic signing to automate some of this with GHA, which would enable a committer (non-PMC) to kick off release processes
                - There are discussions about this ongoing with ASF members and other ASF project teams https://cwiki.apache.org/confluence/display/SECURITY/Reproducible+Builds
    - Help appreciated as always
- Moving the Java implementation to a new apache/arrow-java repo
    - Discussion at https://lists.apache.org/thread/b99wp2f3rjhy09sx7jqvrfqjkqn9lnyy
    - There is general agreement that we should do this
    - Related to this: we have shared release architecture for all the various arrow-repos, but we have achieved this by copy-pasting code. We should take this opportunity to do this properly in a truly reusable way
    - Takeaways from recent experience doing this for arrow-go:
        - Hardest part has been redirecting users to file issues, etc. in the new repo instead of the monorepo
            - We should update the GH issues comment bot to look for "[Go]" in the issue topic or the Component: Go label and redirect user to the new repo (and same for Java)
                - Jacob will look into this
                    - https://github.com/apache/arrow/pull/44818
        - One of the nicest benefits has been the ability to use conventional commits style PR messages instead of GH- prefixes
    - We should better document that the project is not all in the monorepo
        - TODO: Update main README in monorepo to link to other arrow- repos
            - Ian will look into this https://github.com/apache/arrow/issues/44801
        - TODO: Update the website GitHub links to give people a choice to go to other arrow- repos
            - Ian will look into this https://github.com/apache/arrow-site/issues/560

2024-11-06

Attendees

- Joe Yen
- Raúl Cumplido

- Rakshika B
- Bryce Mecum
- Ian Cook
- Joris Van den Bossche
- Dewey Dunnington
- Jacob Wujciak-Jens
- Bruce Irschick

Discussion

- 18.0.0 release
    - Post-release tasks are nearly complete
    - Issue identified: binaries were generated from incorrect commit
    - Patch release is needed to solve this
        - Possibly 18.0.1, but possibly 18.1.0 because of new ChunkResolver API being included
            - This is ambiguous because the ChunkResolver PR was originally intended to be included in 18.0.0 but was pulled out at the last minute. The ChunkResolver API is not a *breaking* change, but it is a new API.
            - We could leave this at the discretion of the release manager.
        - There are also some backport candidates: https://github.com/apache/arrow/issues?q=is%3Aissue+label%3Abackport-candidate+is%3Aclosed
        - See discussion at https://lists.apache.org/thread/t8k7l2hsbgdt7cszj7hrpjdfpn91n5zb
        - Help needed. Jacob and others will help. Would be good to have some new people involved.
            - How to volunteer?
            - For new community members, volunteering for post-release tasks is a good way to get started: https://arrow.apache.org/docs/developers/release.html#post-release-tasks
- Moving impls out of monorepo (Continuing discussion)
    - Swift may be a good candidate because Swift package management prefers source code at the root
        - See apache/arrow#38872
        - But there are concerns about whether there is an active maintainer
            - This causes a risk that Swift package releases might not happen during the regular release cycle; but maybe that's not a significant problem
    - Other languages we're considering this for include Java, JavaScript, C#, and D (https://github.com/apache/arrow/pull/44536)
        - For D, the issue is different because the PR is not merged yet

    - Swift and D are both problematic at this stage because there is no maintainer of those who have commit rights
    - This raises discussions about how to handle this situation with the D implementation PR and in future such cases
        - We should define a set of guidelines and expectations for acceptance of a new implementation
            - We expect a commitment of future maintenance from the author(s)
            - We expect the author to work toward becoming a committer
            - Ideally there should be more than one person working on the new implementation
            - Ideally there should be some libraries or applications which depend on the library so it is not just an academic exercise
            - Maybe we could
        - Bryce will open a PR to add a section to the developer documentation describing these guidelines
- Next ADBC libraries and drivers release (v15)
    - David Li plans to kick this off soon
    - Notable improvements:
        - Numerous bugs fixed in Postgres driver
        - Many improvements to Rust implementation

2024-10-23

Attendees

- Ian Cook
- Raúl Cumplido
- Shoumyo Chakravorti
- Saurabh K. Singh
- Rok Mihevc
- Xuwei Fu
- Rakshika B
- Matt Topol
- Sri Nadukudy
- Weston Pace
- Ruoxi Sun
- Joris Van Den Bossche
- Bryce Mecum
- Bruce Irschick
- Yiwei Wang

Discussion

- A bugfix for BufferedReader which requires review
  - https://github.com/apache/arrow/pull/44387
  - Bug was introduced in 17.0.0 release
  - Fix will not be included in 18.0.0 unless there is a new RC (which looks unlikely)
- 18.0.0 release
  - RC0 is available for verification
  - Some minor issues, but nothing that will force an RC1 (so far)
    - There is a conda-related issue which can be fixed with a backport
  - Raúl has not closed the vote yet but plans to release soon if there are no blocking issues raised
- Status of project to split out some language implementations out of the monorepo?
  - Rationale: Simplify monorepo release process; better support versioning idioms for specific language communities
  - Recently completed for Go
  - We would like to do this for some other implementations (Java? JavaScript? C#?) but the main obstacle is commitment from a maintainer who is deeply familiar with that language implementation.
  - The implementations that are primarily bindings (e.g. Python) would be more difficult to do this for.
- Change to project description on website and GitHub
  - https://github.com/apache/arrow/issues/44474
- Recent news that QuantStack is forming an Arrow-focused team with Antoine as the first member
  - https://medium.com/@QuantStack/quantstack-steps-up-to-support-apache-arrow-with-new-dedicated-team-9ddc952f20e2
- Issue about opening a Cassandra ADBC driver
  - https://github.com/apache/arrow-adbc/issues/2245
  - Ideally, ADBC drivers live in the database's/engine's repos (where possible), but they can live in an ADBC repo if that's not possible
- Work to implement async C interface
  - https://github.com/apache/arrow/pull/43632
  - Matt will start a vote this week

2024-10-09

Attendees

- Rok Mihevc
- Antoine Pitrou
- David Coe

- Ian Cook
- Nic Crane
- Rishabh Maurya
- Dewey Dunnington
- Joris Van Den Bossche
- Bryce Mecum
- Jacob Wujciak-Jens

Discussion

- Arrow 18.0.0
    - Feature freeze has been done today.
    - There are still some issues to be finalized:
        - https://github.com/apache/arrow/milestone/64
        - One of these PRs removes NumPy as a PyArrow dependency ← this should be clearly mentioned in the release notes
    - I (Raúl) plan to execute verification and tests on the maintenance branch and open any other release blocker before creating initial RC
- nanoarrow 0.6.0
    - On mailing list
    - Verification welcome
- Discussion about ADBC PR review capacity
- Second edition of Matt's book is out: https://www.amazon.com/Memory-Analytics-Apache-Arrow-hierarchical/dp/1835461220/
    - Has new content about ADBC and more

2024-09-25

Attendees

- Xuwei Fu
- Sri Nadukudy
- Marcus Hanwell
- Raúl Cumplido
- Rok Mihevc
- Ruoxi Sun
- Jacob Wujciak
- Rishabh Maurya
- Joel Lubinitsky
- Ian Cook

- Bruce Irschick
- Matt Topol
- Bryce Mecum
- Steve Lord
- Dane Pitkin
- Dewey Dunnington
- James Duong

Discussion

- Arrow 18.0.0
    - Feature freeze next Tuesday 1st of October
    - Current open issues: https://github.com/apache/arrow/milestone/64
    - Current blockers: https://github.com/apache/arrow/issues?q=is%3Aopen+is%3Aissue+label%3A%22Priority%3A+Blocker%22
    - Process is ongoing to separate out the Go implementation into a different repo and a separate release process (this is a release blocker)
        - First version in new repo will be released as v18
        - Subsequently we will adopt a different versioning scheme for Go: if there are no breaking changes, new versions will be released as minor releases (incrementing the second number in the version)
    - Please address nightly failures
    - Any PRs need review/help to merge by Tuesday?
        - Decimal32/64 C++ implementation https://github.com/apache/arrow/pull/43957#issuecomment-2371852920
        - Async C data interface PR and upcoming vote https://github.com/apache/arrow/pull/43632#issuecomment-2372249270
- nanoarrow
    - New release starting next week
    - IPC writer has been added
    - IPC integration tests have been added (and it has already allowed us to diagnose and fix integration problems)
    - StringView support has been added

2024-09-11

Attendees

- Jacob Wujciak-Jens
- Matt Topol
- Joel Libintsky

- Ruoxi Sun
- Dewey Dunnington
- Bryce Mecum
- James Duong
- Felipe Carvalho
- Bruce Irschick
- Dane Pitkin
- Xuwei Fu
- Srikanth Nadukudy
- Michael Chavinda
- Joris Van Den Bossche
- Rishabh Maurya
- Rok Mihevc

Discussion

- 18.0.0 feature freeze is 1st of October
- Nanoarrow release before the end of the month
    - New IPC writer is coming
- Matt looking for more feedback on the ArrowAsyncDeviceStreamHandler PR: https://github.com/apache/arrow/pull/43632
- Go move out of apache/arrow into apache/arrow-go
    - Goal is to switch in time for 18.0.0
    - Announce move in 18.0.0 patchnotes of the mono repo
- Follow up on last weeks 'Ruoxi seeing unexpected performance from AVX2 intrinsics'
    - It looks like a microcode update that fixed an intel security vulnerability causes the slow down
    - https://github.com/apache/arrow/pull/43832#issuecomment-2326646353
- https://github.com/apache/arrow/pull/43995 A pr about parquet schema with map/list to arrow schema without arrow schema being stored, maybe it affect the user reading from which have nested list. We found this when testing Hudi. Maybe the go can also checks this in go after this patch is merged.
- GitHub workflow discussion (monorepo) about allowing multiple PRs for one issue: https://lists.apache.org/thread/rov5kkkct73ym2jnwfmknlp946djq6j5
- The question came up how to handle abandoned PRs that provide critical fixes
    - Taking over PRs is fine after prolonged inactivity by the author
    - Ideally preserve git history for proper attribution
- Rishabh is looking to add the Arrow Format to opensearch to improve join performance and looking for pointers (opensearch currently supports only JSON format)
    - https://github.com/opensearch-project/OpenSearch/issues/15185
    - It will be a Java implementation using Arrow and Flight.

2024-08-28

Attendees

- Ian Cook
- Matt Topol
- Raúl Cumplido
- Srikanth Manamohan Nadukudy
- Steve Lord
- Joel Libinitsku
- Ruoxi Sun
- Rok Mihevc
- Bruce Irschick
- Xuwei Fu
- Dane Pitkin
- James Duong
- Felipe Oliveira Carvalho
- Rishabh Maurya

Discussion

- Increasing awareness of new canonical extension types (UUID, JSON, 8-bit boolean) outside of the libraries where they were initially implemented
    - Other Arrow implementations – do we have issues open?
    - Other projects (DuckDB, LanceDB) – Rok, Ian increasing awareness with these
    - We could use the release notes to increase awareness
- C++ stringview <-> string casting PR https://github.com/apache/arrow/pull/43302
    - Needs review
- Problem of many Arrow string types (regular, large, dictionary, view) complicates Parquet / Arrow interop. When you read in a Parquet string, which Arrow type should it be in?
    - We are missing a notion of semantic subtyping that would facilitate this
    - Existing workarounds:
        - You can store an Arrow schema in the Parquet metadata, which Arrow implementations use to set the schema it reads in with
        - You can specify whether you want each string column to be read in as a dictionary column or not
    - We could use an API with cleaner abstractions for specifying deserialized types
        - Xuwei will open an issue to share some ideas about how to do this from Arrow Rust
    - See recent blog post from InfluxData https://www.influxdata.com/blog/faster-queries-with-stringview-part-one-influxdb/
    - Ideally we would default to deserializing to the most performant Arrow type (so probably stringview) but we should be reluctant to do this if it could cause

compatibility problems for users (which it surely would because support for stringview is sparse)
- Recent work to improve support for non-CPU device arrays / chunkedarrays in C++ and Python
    - https://github.com/apache/arrow/pull/43542
    - https://github.com/apache/arrow/pull/43853
    - https://github.com/apache/arrow/pull/43795
    - https://github.com/apache/arrow/pull/43729
- Upcoming 18.0.0 release
    - Still weeks away, but we are targeting feature freeze for Monday 7 October
- Supporting nested types in Acero joins
    - Xuwei is working on https://github.com/apache/arrow/issues/43716
    - < discussion about whether we need support for arbitrarily deep nesting and whether we need support for nested join key columns or just in other columns >
    - Related code in other implementations: https://github.com/facebookincubator/velox/blob/db8875c425e8132f553adf12e106cd2e28a811c0/velox/exec/ContainerRowSerde.cpp, https://github.com/apache/arrow-rs/blob/master/arrow-row/src/lib.rs#L147
- Ruoxi seeing unexpected performance from AVX2 intrinsics
    - < discussion of possible solutions >

2024-08-14

Attendees

- Rok Mihevc
- Ian Cook
- Ruoxi Sun
- Sri Nadukudy
- Matt Topol
- Dane Pitkin
- Joel Lubinitsky
- Xuwei Fu
- Dewey Dunnington
- Jacob Wujciak
- Joris Van Den Bossche

Discussion

- Propose to widen the offset type of the row table from 32-bit to 64-bit [Rossi]

    - https://github.com/apache/arrow/pull/43389
    - No objections from dev@ or user@
    - Antoine is reviewing
    - Any plans to address other limitations in Acero joins?
        - For example https://github.com/apache/arrow/issues/30074
        - No, no immediate plans
- Propose add 32-bit and 64-bit bit widths for Decimal type [Matt]
    - Currently Arrow only supports 128- and 256-bit Decimals
    - Adding 32- and 64-bit Decimals would help with compatible with some other systems
    - Idea: Since the concept of parameterized bit widths already exist in the Arrow spec, can we extend the spec to include this narrower bit widths simply by stating in the spec that the width parameter can take these additional bit widths?
        - As opposed to (for example) defining Decimal32 and Decimal64 as logical types (possibly canonical extension types?) with metadata that annotates physical Int32 and Int64 storage
        - The place that defines the currently allowed bitwidths: https://github.com/apache/arrow/blob/ab432b1362208696e60824b45a5599a4e91e6301/format/Schema.fbs#L235-L237
        - We would need to establish the limits on scale and precision for the 32 and 64 bit widths (following norms set by other systems)
        - Is there any benefit in broadening it further beyond 32, 64, 128, 256?
            - Probably not because this would make support very difficult for implementations to be confirmed to be compatible
- ArrowAsyncDeviceStreamHandler (async callback handler for C data interface) [Matt]
    - https://github.com/apache/arrow/pull/43632
    - Reviews requested
    - Currently the PR is only a change to the header and docs
    - Note that we already use callbacks in the C data interface ABI, but with different semantics. This PR reverses the semantics so that the consumer establishes callbacks and passes them to the producer which calls the callbacks. Other proposals had different semantics, different ways of managing backpressure and handling concurrency, etc. A lot of the discussion is about where the complexity should live (in application code or in the API itself)
        - There are tradeoffs between ease implementation complexity vs. built-in async support

2024-07-31

Attendees

- Steve Lord

- Ian Cook
- Raúl Cumplido
- Rok Mihevc
- Ruoxi (Rossi) Sun
- Felipe Oliveira Carvalho
- Xuwei Fu
- Joris Van den Bossche
- Sri Nadukudy
- Bryce Mecum
- Bruce Irshick
- Dane Pitkin
- James Duong
- Jacob Wujciak

Discussion

- Propose to widen the offset type of the row table from 32-bit to 64-bit [Rossi]
    - https://github.com/apache/arrow/pull/43389
    - Context: Acero hash join is limited to 4 GB size because of max row offset limitation
        - There is no practical workaround for this (e.g. splitting into smaller batches) because we hit the limitation when the hash table is being built
    - Changing to 64-bit will enable a range of new use cases
    - But tradeoff is larger memory consumption (fixed at 4 bytes per row) and more CPU instructions
        - However, benchmarks seem to show that additional CPU usage is not significant. So larger memory consumption is the main concern.
        - Weston (already requested as reviewer) recommended to review
        - Felipe (felipecrv on GitHub) will also try to review
    - Is it possible that we could detect if 64-bit offsets are needed and only incur the extra overhead if it's actually needed?
        - Maybe, but this will add code complexity
    - How does this change the performance?
        - E.g. does use of 64-bit offsets cause more cache misses that causes performance to be significantly worse?
            - It seems not; Conbench test results do not show any runtime increase
    - See also: polars "big index": https://docs.pola.rs/user-guide/installation/#big-index
        - but this is about the number of rows in a single dataframe
    - Recommendation: Send email to the dev@ and user@ ML lists asking if any users are dependent on current memory use patterns for whom this would create a big problem
    - In the longer term, it would be nice to have several types of joins including one that works in memory constrained environments
- https://github.com/apache/arrow/issues/43408

    - TL;DR `InputStream::Advance()` would always call `Read`, which might be low efficient, besides, it doesn't return the actual size it skip (read)
    - This is easy enough to fix, but is there a better API for this?
- https://github.com/apache/arrow/issues/43382
    - TL;DR Reading Parquet min-max stats and pruning would has bug when only one of them is truncated. This can be fixed by disable min-max in this case or also add “HasMin” “HasMax” api here
    - Recommend asking for review from Antoine
- Java 11 deprecation?
    - Make Java 17 min version
    - Context: In Arrow version 18 we will drop support for Java 8. Java 11 has recently ended long term support and we intend to propose dropping support for that in Arrow version 19
        - Apache Spark 4.0 drops support for Java version 8 and 11, making 17 the minimum version
    - Dane will start discussion on mailing list
- Felipe is working on a structured DSL for automatically generating Flight services
    - Ultimate goal would be to have the configuration of a Flight service (including Flight SQL) be fully specifiable through this DSL
    - Motivation: there are a lot of finicky details (manual type checking, packing and unpacking of protobuf types inside Any, etc) that this could abstract away
        - This also gives us a path to remove gRPC
    - Felipe is looking at implementing this in KDL
- Blue-sky question: Have we ever thought about having canonical custom metadata fields in Arrow IPC?
    - For example to tell IPC readers that the data in a record batch is sorted?
    - There are some complications; for example if the data is sorted by an *expression* of some fields in the data, we don’t have a way to represent that expression in the Arrow spec

2024-07-17

Attendees

- Xuwei Fu
- Rok Mihevc
- Steve Lord
- Raúl Cumplido
- James Duong
- Bruce Irschick
- Dewey Dunnington
- Matt Topol

- Joel Lubinitsky
- Bryce Mecum
- Ian Cook
- Kasim Mir
- Dane Pitkin
- Jacob Wujciak
- Ruoxi Sun
- Weston Pace
- Felipe Oliveira Carvalho

Discussion

- 17.0.0 release status
    - https://lists.apache.org/thread/mnzdpwzhctx6yrjl16zn8hl7pcxxt575
    - Post-release tasks that still need doing:
        - R package
        - Cookbook (Bryce?)
        - Blog post
            - https://github.com/apache/arrow-site/pull/537/
    - Post-release tasks pending action by others
        - vcpkg (waiting on merge)
        - Homebrew (waiting on merge)
        - Conda-forge (waiting on merge)
    - Publishing Emscripten wheels?
        - https://github.com/pyodide/pyodide/issues/2933#issuecomment-2232098403
- Bool8 Canonical Extension Type [Joel]
    - https://lists.apache.org/thread/nz44qllq53h6kjl3rhy0531n2n2tpfr0
- C# adding support for Large Binary / Large String but only supporting < 2GiB and offset int32 [Raul]
    - https://github.com/apache/arrow/pull/43269
    - Consensus seems that this is not (yet) a LargeList/LargeBinary/LargeString implementation; it is a compatibility layer
    - PR seems OK from a documentation perspective, but we should check what error messages get thrown if the user tries this with a string > 2 GiB or an offset larger than the max of int32
- Seattle Arrow meetup
    - Tuesday August 13
    - We have a room in the same venue where posit::conf is happening
    - Details at https://github.com/apache/arrow/issues/41881
- Proposal to remove Flight UCX
    - Discussion at https://lists.apache.org/thread/g89x2y6pvlq6gyf0d1jnxfl2onsrkyt8
    - PR to remove: https://github.com/apache/arrow/pull/43297
    - Raúl will reply to ML thread with link to PR
    - The intention is that this is superceded by disassociated transports

        - I.e. you would use a ucx: protocol for the location
- Ian to fix Zoom meeting to allow duration > 40 mins

2024-07-03

Attendees

- Bryce Mecum
- James Duong
- Xuanwo Vars
- Raul Cumplido
- Kazim Mir
- Logan Riggs
- Rok Mihevc
- Ruoxi Sun
- Xuwei Fu
- Felipe Oliveira Carvalho
- Dane Pitkin
- Matt Topol
- Ian Cook

Discussion

- Arrow 17.0.0
    - Code freeze done on Monday 1st of July
    - Currently 4 opened issues for 17.0.0: https://github.com/apache/arrow/milestone/62
        - Only 1 blocker (protobuf)
        - Others are nice-to-have if merged in the next few days
- Donation of a User-Defined Function Framework
    - Clarifying some confusion about whether this is a Rust-centric library or cross-language
        - UDFs can be defined natively (Rust, Python, Js, Java), as WASM binaries (any language), or Flight Server (provided by client, server sends RPC call by URI when UDF is executed)
        - Not necessarily coupled to DataFusion. DataFusion is an example of one engine that may import and use this lib.
    - Suggested to clarify protocol itself to support language-independent / cross-language usage. Once this is defined the existing lib may be considered the first reference implementation of the protocol.
    - How does this compare to user-defined compute kernels?

        - Compute leaves a lot of room for custom behavior, not necessarily standardized
    - There's currently a lot of (different) functionality under the umbrella of arrow-udf. Perhaps by splitting certain features out they can be more easily adopted into various arrow projects/repos.
- Parquet-cpp issues were migrated to GitHub.
    - New issues should be opened on github for the respective repo

2024-06-19

Attendees

- Ian Cook
- Raúl Cumplido
- Rok Mihevc
- Matt Topol
- Ruoxi Sun
- Joel Lubinitsky
- Alenka Frim
- Jacob Wujciak
- Joris Van den Bossche
- James Duong
- Felipe Oliveira Carvalho

Discussion

- 1st of July feature freeze for 17.0.0
    - Will send out reminder one week prior to start of freeze
    - There are a few blockers for nightly test failures
    - Please mark issues as Priority: Blocker if they should block the release
    - https://github.com/apache/arrow/labels/Priority%3A%20Blocker
    - Help needed identifying causes of nightly test failures, opening issues, and fixing them if able!
    - 17.0.0 will be the first release where we officially release the MATLAB implementation
- Managing the LinkedIn page
    - Raúl has been doing it so far
    - So far he has been posting the same stuff that's posted to Twitter (X) and posting this requested by other trusted community members or PMCs, but we could do more if anyone has ideas/requests
    - Ideas?

        - Posting about releases and linking to blog post (already doing this)
        - Including highlights of release notes directly in LinkedIn post?
        - Linking to new tutorials?
        - News about Arrow-focused meetups and events?
        - Sharing new blog posts on the Apache Arrow blog
    - If there is anyone with time available to do devrel-type work and wishes to volunteer to create/post more content to the
- Recording the Meetings
    - Should we capture recordings / transcripts of this meeting and share them publicly?
        - Consensus among attendees who spoke up is mixed
            - Some attendees are reluctant to have their voice, image, and words shared publicly
            - Some attendees expressed that attendees in timezones that make this meeting inconvenient would benefit more from the meeting if a full recording or transcript were shared
                - But perhaps a better solution is to have a separate meeting at a different time that is better for Asia timezones?
                - We could potentially consolidate the meeting notes
        - If individual attendees have accessibility needs, then everyone seems to agree that they are absolutely welcome to use personal accessibility tools to help them participate
        - The bulleted notes (what you're reading now) are OK, but likely not as rich as an AI-generated summarized transcript
            - But will we need to fix / clean up the transcript?
            - Is anyone willing to volunteer to do this regularly?
    - A related topic which has been discussed previously is whether we should attempt to reach a broader audience for this meeting
        - In practice, this meeting is more focused on the Arrow developer community (or at least more technical members of the user community)
        - But anyone is welcome to attend, and we do occasionally have people join who discuss interesting use cases from a non-developer perspective
        - We would like to open up to the broader community somewhat more; the small number of regular attendees at this meeting is notable when we consider how popular and widely used Arrow is
        - Opportunities to do this are limited by the nature of the Arrow project—it is often deliberately invisible to end users
- Discussion about whether we should be doing more outreach to other projects instead of investing more in internally-focused Arrow discussions
    - For example, there is not much communication/awareness of what is happening with PyArrow in pandas in the broader Arrow developer community
    - Maybe we could have joint meetings together with other open source project developers?
        - We could invite them to this meeting, or join their meetings

        - We could coordinate with specific members of the Arrow community to serve as “ambassadors” to other projects (like Joris with pandas)
        - Some projects (such as Spark) have used Arrow internally but been reluctant to publicly expose Arrow interfaces/methods to their users; this type of outreach could help improve this
        - A good starting point might be coming up with a list of the most notable adjacent projects
            - There are MANY projects that depend on Arrow (https://github.com/apache/arrow/network/dependents) so we need to narrow it down
            - Attempt to do this (subjectively):
                - “Tier 1” (using Arrow directly)
                    - Parquet
                    - pandas
                    - Spark
                    - Datafusion (but community is already doing a lot of cross-communication)
                    - Polars
                - “Tier 2”
                    - Iceberg
                    - DuckDB
                    - Gluten?
                    - Velox?
                    - Ibis
                    - Substrait
                - …
- Upcoming events
    - Talk with Wes McKinney tomorrow: https://lu.ma/vkd8h5nu
-

2024-06-05

Attendees

- Dewey Dunnington
- Jacob Wujciak-Jens
- Sri Nadukudy
- Rok Mihevc
- Raúl Cumplido
- Joel Lubinitsky
- Arun

- Christian Casazza
- James Duong
- Fireflies.ai Notetaker Christi
- Jeremy Taylor (XTDB)
- Logan Riggs
- Steve Lord
- Weston Pace
- Dane Pitkin
- Bruce Irschick
- Kazim Mir

Discussion

- Early planning for Seattle meetup in August
    - Potential date/time for meetup?
    - Github issue: https://github.com/apache/arrow/issues/41881
- Migration of parquet-cpp issues
    - Vote succeeded; moving parquet-cpp from Jira to Github
    - Expected to reduce friction around creation/maintenance of issues
- 1st of July feature freeze for 17.0.0
    - Will send out reminder one week prior to start of freeze
- Recommendations for Arrow + DuckDB workflow?
    - Run DuckDB on the client and querying arrow/parquet data. Does Arrow Flight help?
    - Balancing CPU/Network throughput when setting compression
    - Aggregating on the server side can significantly reduce egress in some cases
    - https://docs.google.com/document/d/1-x7tHWDzpbgmsjtTUnVXeEO4b7vMWDHTu-lzxlK9_hE/
- Use of screen recorders / note taking apps for community calls
    - Requesting consent from participants
    - Is there somewhere we can share recordings in an official capacity?
- Attention to thread on mailing list to replace memory allocator for arrow cpp
    - https://lists.apache.org/thread/dts9ggvkthczfpmd25wrz449mxod76o2

2024-05-22

Attendees

- JB Onofré
- Ian Cook
- Rok Mihevc
- Joel Libinitsky

- Matt Topol
- Ruoxi Sun
- Norman Jordan
- James Duong
- Bryce Mecum
- Christian Casazza
- Kazim Mir
- Dewey Dunnington
- Steve Lord
- Dane Pitkin
- Joris Van den Bossche

Discussion

- Potential parquet spec change https://lists.apache.org/thread/5jyhzkwyrjk9z52g0b49g31ygnz73gxo
    - There will be a Parquet community meeting next week; find the link on the mailing list - https://lists.apache.org/thread/znl7kcs00qdlwndml4yzvf6rssjw79c0
- Flight SQL ODBC Driver status?
    - Update from JB: Driver is updated to support latest version, CMake changes have been made, but didn't have time to finish. JB will resume work in the coming weeks. From a legal standpoint it seems OK but will need PR to be reviewed.
    - James can help with PR when it's ready
    - JB can discuss on Zulip
- Dropping support for Java 8
    - Two considerations raised by JB:
        - Clear communication is important for downstream dependencies
            - Parquet Java?
        - We will need a new Avro version (1.12)
- nanoarrow 0.5.0 will be available for voting on the ML later today
- Discussion on the mailing list about passing statistics through the Arrow C data interface
    - https://lists.apache.org/thread/z0jz2bnv61j7c6lbk7lympdrs49f69cx
- Discussion about async support in ADBC
    - https://github.com/apache/arrow-adbc/issues/811
- Arrow Flight RPC vs. Arrow Flight SQL
    - If you want to serve both files and connections to databases, how would you do that with Flight or Flight SQL? What about ADBC?
        - ADBC is purely on client side
        - An ADBC driver can wrap Flight SQL
        - Flight SQL is built on top of Flight RPC
        - Flight RPC is very flexible and extensible; Flight SQL is more standardized
    - Is it possible to download a Parquet file instead of converting Parquet to Arrow and sending Arrow through Flight?

        - This is the subject of the recent “extending Flight locations” / “disassociated transports” proposal:
            - https://docs.google.com/document/d/1-x7tHWDzpbgmsjtTUnVXeEO4b7vMWDHTu-lzxlK9_hE/
            - https://lists.apache.org/thread/x1pgdt7pswd7f9olfwkwmmvpvvb9q62r
        - See also: Arrow data over HTTP APIs
            - discussion at https://lists.apache.org/thread/vfz74gv1knnhjdkro47shzd1z5g5ggnf
    - Single vs multi-tenant deployments of Flight SQL?
        - Flight SQL can handle concurrent connections and authentication; either deployment strategy can be compatible

2024-05-08

Attendees

- Adam Reeve
- Ian Cook
- Matt Topol
- Jacob Wujciak
- Christian Casazza
- Rok Mihevc
- Steve Lord
- Ruoxi Sun
- Norman Jordan
- Raúl Cumplido
- James Duong
- Bryce Mecum
- Sri Nadukudy
- Hao Xu
- Dane Pitkin
- Dewey Dunnington
- Soumya Sanyal

Discussion

- Introductions
- Java 8 deprecation update
    - https://lists.apache.org/thread/65vqpmrrtpshxo53572zcv91j1lb2y8g

    - Looks like we will only drop Java 8 for now; we will keep Java 11 support for now
        - Some other projects are dropping support for 11 and jumping to 17 but we want to be more cautious because some other projects depend on Arrow and are not ready to move off of Java 11 yet
        - Main downside of keeping 11 for now is the chance that Arrow dependencies will drop support for 11 soon
        - Java 11 mainstream support has already ended; it's in extended support now
        - Decision to only drop Java 8 for now is based on the need to consider 8 and 11 separately; it still seems important to consider dropping support for 11 given security concerns about keeping support for 11. We do not intend to keep support for 11 until the end of the extended support period.
- Using glib libraries from .NET (https://lists.apache.org/thread/5jfk0fcqy90cl8w6v45ny50pwrgfpp1y)
    - Adam was interested in adding Dataset library bindings to .NET implementation
    - Someone opened a draft PR a while ago to add Acero bindings using glib
        - Is this the right approach, or should we bind directly to the C++ library?
            - Adam has some investment in the C++ binding approach because this is how ParquetSharp (which he created) works
    - Matt: .NET currently has a standalone C# implementation rather than use C++ Arrow and most Arrow implementations do one or the other, but not mix both
    - Raúl: keep in mind that the Arrow .NET maintainer community is small; need more help with releases, etc.
    - Dewey: DuckDB is another option if you're looking for an Arrow-based engine that you can use through ADBC
        - For example https://twitter.com/paleolimbot/status/1787525433111556180, https://github.com/apache/arrow/blob/main/python/pyarrow/tests/test_flight.py#L482
- Arrow Flight with Iceberg tables
    - Christian is working on a project to serve data from Iceberg tables to a Python client
        - Has considered alternative approaches including Ibis -> Trino -> Iceberg and Arrow -> Iceberg (reading Parquet files)
    - Matt: Best approach will depend heavily on what interface you want to give to clients.
        - For example there is PyIceberg which can return PyArrow tables / recordbatch readers. This is less complex than homespun solutions.
        - Dremio is another solution; it supports Iceberg and exposes a Flight SQL server
        - But if you want to limit what access clients have or what level of abstraction you want to provide, a different approach might be preferable

    - Matt: Currently for Iceberg only the Java and Python implementations are mature as far as I'm aware. There is also https://github.com/apache/iceberg-rust but not sure of status.
    - Christian: RBAC in Flight?
        - Matt: There are a couple of approaches. In Flight RPC there is a handshake method. You can call it with your header tokens or whatever, and it will return a token that can be used as a bearer token for the header. You could also integrate something like an OAuth flow because the implementations allow you to add in authentication middleware.
            - Example https://arrow.apache.org/cookbook/py/flight.html#authentication-with-user-password
            - This uses just basic auth, but you could expand upon this pattern to implement other auth flows
        - James: Are you looking for just authentication, or also authorization/access to resources?
            - Christian: mostly the former at least for now
        - If HTTP/2 is a problem or if you're trying to fit Arrow-based data transport on top of an existing REST/HTTP API, another option is the Arrow over HTTP approach being discussed currently
            - See https://github.com/apache/arrow-experiments/tree/main/http and https://lists.apache.org/thread/886cnx6ytjst3smmytz4r4ddcbv95191
- 16.1.0 release?
    - Rok is curious about getting a couple of PRs into this
    - Feature freeze has happened; Raúl created RC0 yesterday but there were a couple of issues with Java and C#; Raúl is working to create an RC1 next that will solve these; too late at this point to add new commits to 16.1.0.
    - Why are we incrementing the minor version number when we don't usually do that?
        - Originally we created 16.1.0 because there were several new feature commits in Javascript that didn't make it in time for 16.0.0 and they wanted them to be available sooner than 17.0.0.
- PyArrow has been split on conda-forge
    - pyarrow, pyarrow-core, pyarrow-all
    - See https://lists.apache.org/thread/tcjz5pkkokp3hzr413yltc26gt0xd016
- General discussion about Arrow popularity, awareness, etc.

2024-04-24

Attendees

- Xuwei Fu
- Ian Cook
- Ruoxi Sun
- James Duong
- Norman Jordan
- Dewey Dunnington
- François Michonneau
- Curt Hagenlocher
- Logan Riggs
- Matt Topol
- Bryce Mecum
- Dane Pitkin
- Sri Nadukudy
- Joel Lubinitsky
- Jacob Wujciak
- Clif Houck

Discussion

- Java 8 deprecation
    - Prior discussion https://lists.apache.org/thread/kml53f81z1oskcf00xl7wlbcjssmn91g
    - Issue https://github.com/apache/arrow/issues/38051
    - For context: Spark intends to drop support for Java 8 and Java 11 in next major release (4.0, due in June of this year, around the same time we are due to release Arrow 17); various other projects have also set similar precedents
    - Issue tracking various headaches associated with keeping support for Java 8: _____
    - Potential hurdles?
        - Newer Java versions are stricter about memory management
            - [Ian] Will this kind of thing still work? (This is needed when building a lot of real-world Arrow Java apps) `_JAVA_OPTIONS="--add-opens=java.base/java.nio=ALL-UNNAMED" mvn exec`
                - [Dane] We will investigate this
        - [Ruoxi] Will we make public announcements / warnings about this?
            - [Dane] In the past we have just used the dev@arrow mailing list; this time we should also use the ApacheArrow Twitter account; we will also make this clear in the release notes

            - [Jacob] We should also find out which are the biggest projects that depend on Arrow Java and email their mailing lists
            - [Jacob] I pinned this issue to the top of the Arrow issues on GitHub
- Reading Avro files?
    - Java, Rust, and Go implementations support this
    - C++ implementation does not support this
    - Various related tickets:
        - https://github.com/apache/arrow/issues/16991
        - https://github.com/apache/arrow/issues/22264
        - https://github.com/apache/arrow/issues/39644
        - https://github.com/apache/arrow/issues/39643
    - Anyone interested in working on a C++ implementation and bindings in Python?
    - We can update the implementation status for Go <-> Avro (Go has Read capabilities) https://arrow.apache.org/docs/status.html
- 16.1.0 release?
    - Raúl is out right now and his laptop died, so there will be a small delay
    - Code freeze should happen next week
    - If anyone has non-breaking-change features that missed 16.0.0, please get them in ASAP
- Reminder: [DISCUSS] Versioning and releases for apache/arrow components
    - https://lists.apache.org/thread/tvqns9d9z45mtpsqrngjyg083jdv8f1t
    - The 16.1.0 release highlights the relevance of this
- Parquet docs improvement PR
    - https://github.com/apache/arrow/pull/41187
    - [Xuwei] Would there be iceberg / parquet user uses ‘field id’ ? I’ve update the document for that but seems our reviewer doesn’t include any iceberg user
- Parquet C++ issues
    - Formally still on the ASF Jira (like the rest of Parquet), but since it lives in the Arrow repo (which moved issues to GitHub), this is a gray area
    - [Xuwei] I guess in arrow repo, now only Jinpeng Zhou in google would uses parquet’s issue now, other contributors . Arrow and arrow-rs is de-facto Go/C++/Rust ’s parquet maintainer lib
- Discussion about patterns for deploying ADBC drivers
    - See discussion at https://lists.apache.org/thread/c3lz4qjcn5jxolrdt696n8vv8oh7j9j0
    - [Joel] I am interested in the opposite direction: how to deploy compiled ADBC driver (e.g. to WASM) to run universally
        - E.g. you could distribute a URL to a driver; you wouldn’t need Cgo or anything
        - Possible obstacles:
            - WASM only single-threaded
            - ADBC is fundamentally a C API, so you would need to also ship a driver manager with it, which adds overhead

            - Getting zero-copy streaming to work across this interface
        - Reactions generally positive
        - Person to ask about this: https://github.com/kylebarron
- Recent work to make Arrow WASM-buildable
    - C++ https://github.com/apache/arrow/issues/23221 (merged)
    - Python https://github.com/apache/arrow/pull/37822/ (still ongoing)
    - Joe Marshall is contributing much of this work https://github.com/joemarshall
        - He is looking for sponsors: https://github.com/sponsors/joemarshall
- Passing statistics through Arrow C data interface
    - https://github.com/apache/arrow/issues/38837
    - Comments welcome

2024-04-10

Attendees

- Dewey Dunnington
- John Groves
- François Michoneau
- James Duong
- JB Onofré
- Norman
- Raúl Cumpildo
- Bryce Mecum
- Joel Lubintsky
- Joris Van den Bossche
- Matt Topol
- Bruce Irschick
- Sri Nadukudy
- Weston Pace
- Xuwei Fu
- Jacob Wujciak
- Ruoxi Sun

Discussion

- John Groves: Data frames in disaggregated shared memory
    - https://lists.apache.org/thread/owxw1mffnnfd2m4bg07blqhzdmrqnvmn
- Arrow 16.0.0 status
    - Feature freeze is in place (maint-16 branch is created), still lots of CI failures to fix before a first RC

        - Initial list of blockers: https://github.com/apache/arrow/issues?q=is%3Aopen+milestone%3A16.0.0+label%3A%22Priority%3A+Blocker%22+
        - Raúl will add the “blocker” label for the rest of the CI issues and ping folks for review
    - Weston looking for reviewers on https://github.com/apache/arrow/pull/40696 to enable Polars–PyArrow compute interop
    - PR to track release status: https://github.com/apache/arrow/pull/41118
- Expanding the Python Arrow PyCapsule protocol (wrapping C Data Interface in a standard Python API) with device support: discussion at https://github.com/apache/arrow/issues/38325
- [DISCUSS] Versioning and releases for apache/arrow components
    - https://lists.apache.org/thread/tvqns9d9z45mtpsqrngjyg083jdv8f1t
    -
-

2024-03-27

Attendees

- Ian Cook
- Soumya Sanual
- Xuwei Fu
- Ruoxi Sun
- Raúl Cumplido
- David Li
- Logan Riggs
- Arun
- Rok Mihevc
- Alenka Frim
- Bruce Irschick
- Bryce Mecum
- Dewey Dunnington

Not present:

- JB Onofré (due to conflict with Apache Iceberg meeting), sorry about that.

Discussion

- Arrow 16.0.0
    - Feature freeze 8th of April
- Add support for list view and large list view to Arrow C++ Parquet integration

    - https://github.com/apache/arrow/pull/38850
- PRs needing review?
    - https://github.com/apache/arrow-experiments/pull/27

2024-03-13

Attendees

- Ian Cook
- JB Onofré
- Xuwei Fu
- James Duong
- François Michonneau
- Dane Pitkin
- Logan Riggs
- Ruoxi Sun
- Sri Nadukudy
- Joris Van Den Bossche
- Matt Topol
- Steve Lord
- Soumya Sanyal
- Bryce Mecum
- David Li
- Joel Lubinitsky
- Dewey Dunnington
- Rok Mihevc

Discussion

- ODBC donation PR (even if it doesn't build yet, as proposed by Jacob, JB will open the PR)
    - Legal / license / IP issues are resolved
    - CMake issues (still some build issues found) are fixed now, work on PR is ongoing, still need a few more days of work on this
- Are we good about Arrow builds on ARM arch (gandiva, …) ? Do we have all needed resources (CI, …) ?
    - We have Apple Silicon CI / build through Crossbow
    - Various build systems (homebrew, vcpkg, etc.) have prebuilt arm64 packages
- Thoughts about Arrow 15.0.2 release ?
    - See ML thread at https://lists.apache.org/thread/4pr0pm6gmnpg259g4fqx2p5b4t84moto

    - Raúl has already cut an RC for 15.0.2 -> https://github.com/apache/arrow/pull/40504
    - Next major release should be in late April, with feature freeze around 1st week of April
- (Matt) Still trying to get more engagement on Arrow Communication Doc
    - Have split the Flight Location proposal into a separate doc
    - If there's still no comments / engagement before end of the week I'll try proposing a vote again?
- We need volunteers to refactor Skyhook
    - https://lists.apache.org/thread/77jkh04d06brs6l0j1vqzgx5yncp69g1
- gRPC changes that are related to Flight:
    - https://github.com/grpc/grpc-java/pull/10977
- New releases of Java Avro and ORC?
    - Xuwei will talk to Gang about ORC
    - Ian will talk to JB about Avro

2024-02-28

Attendees

- Ruoxi Sun
- JB Onofré
- David Li
- David Susanibar Arce
- Dewey Dunnington
- Matt Topol
- Ian Cook
- Jacob Wujciak-Jens
- Joel Lubinitsky
- James Duong
- Joris Van Den Bossche
- Rok Mihevc
- Bryce Mecum
- Sri Nadukudy

Discussion

- Arrow Flight ODBC donation update (from JB)
    - Legally this is all sorted, but JB is having a build problem
    - JB is actively working on this and plans to open a PR soon
- Does it make sense to contribute a second implementation of an ADBC driver (i.e. Go BigQuery)?

    - Joel: Looking through the language support matrix, there is now a C# driver for BigQuery, but it's unclear how the interop is with other languages
    - (Matt Topol) It's on my list of things to do in order to expose easily via the python wrappers we have set up
        - Joel: I might take a stab at this first
        - Matt: See which pieces of the existing drivers we could move into a common base package to make adding more drivers in the future easier
        - David Li: The main challenge is that we don't have a release process for the C# drivers right now
        - Matt: It should be easy to get Arrow data from the Go BigQuery client as record batches now
        - https://github.com/googleapis/google-cloud-go/pull/8506#issuecomment-1823434827
    - Ian: Are there any other systems that we should start lobbying to expose Arrow record batches to allow us to build ADBC drivers in the future?
        - Matt: Clickhouse (already can produce Arrow data but it's unclear whether the interface is exposed in a way that we can use easily)
        - David Li: Spark
        - James: Impala, Hive, Drill
            - Drill - https://github.com/factset/go-drill provides an interface to start from
        - Ian: Trino (but I believe the Trino devs might already be working on this)
    - Joel: It would be great if more upstream vendors could support ADBC's partitioned result sets
- More feedback and engagement needed on disassociated Arrow IPC proposal: https://docs.google.com/document/d/1zHbnyK1r6KHpMOtEdIg1EZKNzHx-MVgUMOzB87GuXyk/edit plz (from Matt)
    - Joel: It might be easier to discuss on vote on if it were broken into smaller discussions/proposals
- 15.0.1 patch release
    - In progress https://github.com/apache/arrow/pull/40211
    - RC0 has been created; there are some CI failures but they look like flakey tests; no blockers so far

2024-02-14

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- James Duong

- David Li
- Joel Lubinitsky
- Dane Pitkin
- Soumya Sanyal
- Ruoxi Sun
- Matt Topol
- Joris Van den Bossche

Discussion

- Should we schedule this meeting at a different day / time?
    - JB noted that the current time overlaps with the Iceberg meeting
    - What if we moved it one hour earlier?
        - Would work for most but not all of the attendees
- Should we add the calendar links (above) to the page on the Arrow site?
    - We could do this, but it's unclear if eliminating this little bit of friction is desirable or not
        - Ian will talk to JB and ask about this
        - Ian will attend an Iceberg meeting
- Possible 15.0.1 patch release
    - https://lists.apache.org/thread/xmxtzl335o6v4vfhs825f75gprgyskqg
    - Any other issues / PRs that should get into 15.0.1? If so, set milestone 15.0.1 and priority Blocker
    - Quick postmortem re the Java problem that is the main reason for needing a 15.0.1 patch release: was this avoidable?
        - Was caused by an undefined symbol not being caught in any tests
        - The issue was reported by community members and also noticed in the Arrow Cookbooks
- Arrow Extension Proposals
    - New protocol for using Arrow IPC to send data to/from shared memory and GPU memory
    - Matt is looking for more community engagement and comments
- FYI: DuckDB is implementing support for StringView in their Arrow integration
    - https://github.com/duckdb/duckdb/pull/10481

2024-01-31

Attendees

- Joel Lubinitsky
- Dane Pitkin
- David Li

- Nic Crane
- Raúl Cumplido
- Elliot Morrison-Reed
- Matt Topol
- Soumya Sanyal
- Sri Nadukudy
- Jacob Wujciak
- Bryce Mecum
- Ruoxi Sun
- Rok Mihevc

Discussion

- Introductionss
- Update re R package 14.0.2/15.0.0 released delay due to significant work required as the result of CRAN rejecting use of binaries
    - Author of an Arrow-R dependency opened an issue because they received a note from CRAN about removal of arrow on the 9th of February.
    - The R-Team is aware and working through some difficulties, we hope to avoid archival but will keep the community updated.

2024-01-17

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- James Duong
- Andy Grove
- Ben Harkins
- David Li
- Steve Lord
- Bryce Mecum
- Rok Mihevc
- Elliot Morrison-Reed
- Sri Nadukudy
- JB Onofre
- Dane Pitkin
- Logan Riggs
- Soumya Sanyal
- Ruoxi Sun
- Matthew Topol

- Jacob Wujciak

Discussion

- Introductions
- 15.0.0 release
    - Raúl sent voting email for release https://lists.apache.org/thread/5yy4cd632xs6kqkdq92f3rsj37h7vtry
    - Verifications successful so far; no blockers yet
    - Please contribute to the release blog post https://github.com/apache/arrow-site/pull/464
        - Help needed with C++ section
- nanoarrow release vote expected in the next week or so
    - Minimal Python bindings coming in this release
- Questions from Elliot about automotive use cases
    - https://lists.apache.org/thread/d6lnjgjpwf2x7cfdsnjqqldpkx6twfrk
    - Data is sent over CAN bus
    - Standard data format is MDF: https://www.asam.net/standards/detail/mdf/wiki/
    - Automotive-specific formats are good for storing data, but less mature for using the data, and very inefficient
-

2024-01-03

Attendees

- Ian Cook
- JB Onofre
- Joel Lubinitsky
- Rok Mihevc
- Raúl Cumplido
- Xuwei Fu
- Matthew Topol
- Ruoxi Sun
- Logan Riggs
- James Duong
- Bryce Mecum
- Sri Nadukudy
- David Li
- Jacob Wujciak
- Soumya Sanyal

Discussion

- Introductions
- Round trip type support for parquet <-> arrow Joel Lubi
    - Joel has been working to improve perf of bulk ingestion to Snowflake via ADBC
    - One approach is to use Parquet (because it is more widely supported than Arrow)
    - There are some snags w/ data type conversion between Arrow and Parquet
    - When writing Parquet files, the Parquet C++ and Go libraries have an option to preserve the Arrow schema for round-tripping (by serializing the Arrow schema, base64-encoding it, and putting it in the Parquet file as a standardized metadata field)
        - Here is the option for the Go Parquet library: https://pkg.go.dev/github.com/apache/arrow/go/v14@v14.0.2/parquet/pqarrow#WithStoreSchema
        - Python store_schema is documented here: https://arrow.apache.org/docs/python/generated/pyarrow.parquet.write_table.html#pyarrow.parquet.write_table
        - Do any other Arrow / Parquet libraries (besides Go and C++) implement this?
            - Yes, Rust does this too:
              We write a extended-key-value with key: `ARROW:schema` and a arrow ipc payload
    - Date conversion is one specific area:
        - Date32 type is days since Unix epoch stored as 32-bit integer
        - Date64 is milliseconds since Unix epoch, stored as 64-bit integer
            - This was a convention borrowed from Java
            - Values are required to be an integer multiple of 1 day in milliseconds: https://github.com/apache/arrow/blob/213cadbbc080399b372291f93aaaa05fe0e67de1/format/Schema.fbs#L245-L250
        - Parquet timestamps:
            - https://arrow.apache.org/docs/python/parquet.html#storing-timestamps
            - https://github.com/apache/parquet-format/blob/master/LogicalTypes.md#temporal-types
        - Version might also matter: https://arrow.apache.org/docs/python/parquet.html#parquet-file-writing-options
- 15.0.0 release
    - Code freeze still planned for Monday 8 January
    - Raúl has been working to solve bugs with nightly checks
    - Currently no blocking issues: https://github.com/apache/arrow/issues?q=is%3Aopen+milestone%3A15.0.0+label%3A%22Priority%3A+Blocker%22+

- Proposal for more efficient Parquet filtering in C++ implementation
    - https://lists.apache.org/thread/8c351l9wd4nmjzmot67321l2toyqdrg5
- R package 14.0.2 build errors
    - We might need to add some patches to 15.0.0 to solve this
    - CRAN submissions are closed until January 8
- Parquet issue tracking
    - Still using Jira?
        - Yes, for now
- Flight SQL / ADBC Substrait support
    - Are there any implementations of this?
        - Not yet
    - Could raise this in the Substrait biweekly meeting: https://groups.google.com/g/substrait/c/VUYBC9R4m1I/m/8RBwKSlzBAAJ
- Calendar link for this meeting?
    - Zoom provides this; Ian will find and share
    - Ian will add to Community page of website

2023-12-20

Attendees

- Ian Cook
- Raúl Cumplido
- Xuwei Fu
- David Li
- Joel Lubinitsky
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Dane Pitkin
- Antoine Pitrou
- Ruoxi Sun

Discussion

- 14.0.2 release
    - Release is complete
    - Post-release tasks are finished, except that the Homebrew PR is not yet merged
    - Blog post is almost ready to be merged https://github.com/apache/arrow-site/pull/443
    - Raúl will make announcement soon after blog post is up
- 15.0.0 release

    - Code freeze planned for January 8
    - Currently no blocking issues: https://github.com/apache/arrow/issues?q=is%3Aopen+milestone%3A15.0.0+label%3A%22Priority%3A+Blocker%22+
    - Any features/fixes we want to get into 15.0.0?
        - [Rok] Variable-shape tensor PRs:
            - https://github.com/apache/arrow/pull/38008
            - https://github.com/apache/arrow/pull/37533
        - [Ian] Azure filesystem PRs:
            - Umbrella issue at https://github.com/apache/arrow/issues/18014
- Ruoxi created a PR for a segmentation fault bug; needs review
    - Issue: https://github.com/apache/arrow/issues/32570
    - PR: https://github.com/apache/arrow/pull/39234
    - Antoine will take a look
- Proposal for more efficient filtering in Parquet C++ implementation
    - Discussion document: https://docs.google.com/document/d/1SeVcYudu6uD9rb9zRAnlLGgdauutaNZlAaS0gVzjkgM/
    - POC code: https://github.com/apache/arrow/pull/38867
    - Xuwei will post to the dev ML tomorrow

2023-12-06

Attendees

- Ian Cook
- Raúl Cumplido
- David Li
- Joel Lubinitsky
- Rok Mihevc
- Sri Nadukudy
- Dane Pitkin
- Soumya Sanyal
- Ruoxi Sun
- Matthew Topol
- Joris Van den Bossche
- Jacob Wujciak

Discussion

- 14.0.2 release
    - Raúl created an RC, but there was a commit missing
        - Raúl and Jacob are working on some fixes to prevent this in the future

    - Raúl will create a new RC tomorrow, verify everything, and email the ML
- Flight SQL ODBC Driver https://lists.apache.org/thread/t1r3pntpzoxdncgoj5f581hxyyl19bkl
    - Improving Vancouver is proposing to contribute a new ODBC driver wrapping Arrow Flight SQL
    - Question: Do we need to go through the IP clearance process for this?
        - David: This is a cumbersome process and it can take a while; it's better to avoid it if it's not necessary
        - Part of the code is from the AWS Timestream ODBC driver—that's the main concern regarding IP clearance
- Update on Flight SQL bulk ingestion support
    - https://github.com/apache/arrow/pull/38385
    - Joel integrated the requested changes and is awaiting review on the new changes
    - Reviews requested
- How to indicate that you want to work on an issue and know that no one else is working on it?
    - In the main monorepo, you can comment "take" and a bot will assign the issue to you
    - In some of the other repos, this is not set up yet
        - We will look at setting it up https://github.com/apache/arrow-adbc/issues/1347
- Conventions for transporting Arrow data over HTTP
    - Discussion at https://lists.apache.org/thread/886cnx6ytjst3smmytz4r4ddcbv95191
    - Examples at https://github.com/apache/arrow/pull/39081
- General discussion about Matt Topol's work to plug in UCX under Flight
- General discussion about how to measure memory usage of PyArrow objects

2023-11-22

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- James Duong
- Xuwei Fu
- David Li
- Joel Lubinitsky
- Bryce Mecum
- Rok Mihevc

- Dane Pitkin
- Soumya Sanyal
- Joris Van den Bossche
- Jacob Wujciak

Discussion

- Arrow 14.0.2 patch release?
    - https://github.com/apache/arrow/issues?q=label%3Abackport-candidate+
    - Any others?
        - https://github.com/apache/arrow/pull/38621 is fixed but it seems like it didn't solve the root cause of the regression; Xuwei is going to work on this soon
            - See https://github.com/apache/arrow/pull/38784 (perf regression) and https://github.com/apache/arrow/issues/38577 (error)
            - These regressions can be worked around by changing batch size, so we could go ahead and do a 14.0.2 patch release without these, but ideally we would include a fix at least for the error
    - Joris advocates for a 14.0.2 release soon
        - We could create a release branch and start to merge in backports, get CI working, etc.
    - Raúl will send an email to the dev list about this and start work on the release next week
    - Note that 15.0.0 is planned for mid-January
- Various updates about items from last week
    - MATLAB versioning questions — looks like there is a good path forward without any major changes
    - Tracking breaking changes — there was an ML discussion about this at https://lists.apache.org/thread/fly2yrmpptsnn72cw34sjghfm4nlbykn and some work is happening to improve the process
    - Bulk ingestion in Flight SQL — vote not started yet because there were some big changes made in response to review; hope to open vote next week
- Transporting Arrow data over HTTP APIs
    - See discussion at https://lists.apache.org/thread/vfz74gv1knnhjdkro47shzd1z5g5ggnf

2023-11-08

Attendees

- Ian Cook
- Raúl Cumplido

- Dewey Dunnington
- James Duong
- Xuwei Fu
- Ben Harkins
- Will Jones
- David Li
- Joel Lubinitsky
- Bryce Mecum
- Kazim Mir
- Sri Nadukudy
- Dane Pitkin
- Soumya Sanyal
- Jacob Wujciak

Discussion

- Arrow 14.0.0 release status
    - All post-release tasks have been completed, except uploading the arrow R package to CRAN
        - We are waiting for a reply from CRAN about the new build system
- PR that extends the Flight SQL spec with bulk ingestion (follow-up from previous meeting)
    - https://github.com/apache/arrow/pull/38385
        - The C++ and Go implementation are both included in this PR
    - There is a related discussion on the mailing list: https://lists.apache.org/thread/mo98rsh20047xljrbfymrks8f2ngn49z
    - If there are no further questions, David Li will start a proposal and vote on the mailing list soon
- Three PRs that might warrant a patch release:
    - Parquet decoder performance regression
        - Xuwei has a PR that reverts the change: https://github.com/apache/arrow/pull/38437
            - Regression goes away, but Xuwei not sure about root cause
            - Might be helpful to start a discussion on the Zulip chat
    - Fix for async bug in Parquet dataset
        - https://github.com/apache/arrow/pull/38466
        - This does not affect performance
        - This is a longstanding bug, but other recent changes made it more likely to be triggered
        - Should this be released in a new patch release?
    - Fix for another Parquet reader regression
        - https://github.com/apache/arrow/pull/38621
        - This might also justify a patch release?
    - Consensus: let's wait for these all to be merged, then start a discussion on the ML about this

        - We should also look at the performance benchmark report for 14.0.0 to see if these showed up: http://crossbow.voltrondata.com/release_reports/arrow-release-report-14.0.0-rc2.html
        - For the two performance regressions, we should quantify the size of the regression
- Tracking performance regressions
    - In addition to checking the performance benchmark report at release time, should we also generate a report and check in mid-cycle (between releases) so we have more time to address regressions before release time?
        - Raúl will look into this
    - Discussion about ways to automate notification of detected performance regressions
- Arrow development process
    - Note: Merge queues are now generally available in GitHub; we might consider using them
        - But the flakiness of our CI on main might make this impractical
- MATLAB implementation almost ready for first release
    - Note that the MATLAB implementation provides bindings to Arrow C++
    - ML message from Kevin Gurney about how to handle versioning: consistent with monorepo, or separate? https://lists.apache.org/thread/0xyow40h7b1bptsppb0rxd4g9r1xpmh6
- General discussion about whether to maintain existing setup with many implementations in monorepo and coordinated release process vs. splitting out some implementations into separate repos and using separate versioning schemes for releases

2023-10-25

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- James Duong
- Bryce Mecum
- Xuwei Fu
- David Li
- Steve Lord
- Joel Lubinitsky
- Kazim Mir
- Sri Nadukudy
- Soumya Sanyal

- Jacob Wujciak

Discussion

- Arrow 14.0.0 release status
    - Votes on RC2 are ongoing
    - Seems to be going well so far
    - There are some problems with the verification scripts (but they won't stop the release or force new RCs)
        - *(SS: brittle, long build time, driven by a bash script. Java test times improved recently, but still room for improvement. Undecided on how to solve for this. Could we make a plan for migration? Move away from bash, move to something more maintainable / scalable? Would be appreciated from new contributors. Reach out to Jacob and Raúl for questions on undocumented behavior. Jacob happy to review. Solid DX improvements are possible here, and aligned with goals of increasing release frequency. IC: Windows and Linux builds diverge. BM: Any issues written up already? Maybe, but also some floating context. RC: Likely want to work on a larger rewrite. Incremental builds, use layers and continue from failure. Maybe improve Archery? Can discuss. DD: Arrow is not the only project in this state. Others – ADBC / Nano – have forked off this build protocol, and would benefit from this work. JW: +1 – a separate build project/driver might make sense.)*
- Running Arrow on very outdated hardware
    - For example hardware with no support for modern SIMD extensions like AVX2
    - Currently there is no official Arrow support policy (although some subprojects/language libraries have unofficial policies)
    - Recent example from another project: https://grpc.io/docs/
    - Jacob has an email drafted about this to send to the developer ML and intends to send it sometime in the next few weeks
- PR that extends the Flight SQL spec with bulk ingestion
    - https://github.com/apache/arrow/pull/38385
    - There is a related discussion on the mailing list: https://lists.apache.org/thread/mo98rsh20047xljrbfymrks8f2ngn49z
    - In addition to the PR, there needs to be a proposal and vote on the mailing list

2023-10-11

Attendees

- David Coe
- Ian Cook

- Raúl Cumplido
- Dewey Dunnington
- James Duong
- Xuwei Fu
- Ben Harkins
- Will Jones
- David Li
- Joel Lubinitsky
- Dane Pitkin
- Matthew Topol
- Rok Mihevc
- Joris Van den Bossche
- Jacob Wujciak
- Felipe Oliveira Carvalho

Discussion

- Current release process (manual?) - David Coe
    - David C volunteered to do nuget process for ADBC
    - David L: Yes, a PMC member or committer runs all the scripts as part of the release process
- Arrow 14.0.0 release
    - Current status
    - Blockers: https://github.com/apache/arrow/labels/Priority%3A%20Blocker
    - Raúl did feature freeze; maintenance branch is created
    - There are several nightly failures, but it's better that in recent releases (thank you Kou for working on this)
    - Include PyCapsule PR? (https://github.com/apache/arrow/pull/37797)
        - Plan is to include this in the release but flag it as "experimental" for now
- Java 8 (11?) deprecation/removal
    - ML discussion https://lists.apache.org/thread/kml53f81z1oskcf00xl7wlbcjssmn91g
    - Backporting features is not feasible given the effort it takes to create/verify releases
    - Takeaways:
        - Waiting until after Spark 4.0 release seems like a good idea
        - We should look for ways to use newer Java features in optional modules that are not supported if you are building for Java 8 or 11

2023-09-27

Attendees

- David Coe
- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- James Duong
- Ben Harkins
- Will Jones
- David Li
- Joel Lubinitsky
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Weston Pace
- Dane Pitkin
- David Susanibar
- Matthew Topol
- Joris Van den Bossche

Discussion

- nuget releases for ADBC?
    - We are releasing C# packages for Arrow on nuget today, but what about releasing the ADBC C# implementation on nuget?
    - It boils down to: someone needs to do the work
        - David Coe can take this on
        - David Li can assist
- ADBC release cycle?
    - 0.7 just released, 0.8 planned
    - In practice we have been doing 6–8 week cycles, based on maintainer availability constraints; we would like to get to a more regular automatic 4–6 week cadence
    - Nuget supports “preleases” — we could potentially also use that
- Please vote for VariableShapeTensor proposal (https://lists.apache.org/thread/8s9s9smclhztkqj3sn4y6f6hp06b3163)
- Python Arrow C Data Interface - PyCapsule protocol (https://github.com/apache/arrow/issues/35531, https://github.com/apache/arrow/pull/37797)
    - Please review
    - Looking for suggestions for other libraries where it would be valuable to implement this in the near term

    - What should be the relationship between this and the dataframe consortium array protocol? https://data-apis.org/array-api/latest/
        - Arrow arrays typically have more than one buffer
    - The proposed interface does not expose the new C data device interface, but it can be extended to do this in the future
    - Could we implement conversion between the C Data Interface and the dataframe interchange protocol e.g. in nanoarrow?
- Arrow 14.0.0 release schedule
    - Reminder: code freeze planned for Oct 10
    - Currently there are 7 blockers flagged https://github.com/apache/arrow/issues?q=is%3Aissue+is%3Aopen+label%3A%22Priority%3A+Blocker%22
        - There are also about 50 nightly failures that might be blockers
            - Help needed to fix CI failures
        - Other potential issues:
            - Python 3.12
            - Java versions
    - PRs / votes that we really hope can get done in time for v14?
        - Flight app_metadata vote: https://lists.apache.org/thread/113xm107ptn1t76txockp11oc7wlh0ok
            - The PR implements this for C++ and Go
            - We need help implementing this for Java
        - Parquet modular encryption: https://github.com/apache/arrow/pull/34616
        - PRs for compiling Arrow for WASM (Kou already looking at them)
        - StringView
            - Vote passed to add to spec; added to spec https://github.com/apache/arrow/pull/37526
            - C++, Go implementation PRs are under review:
                - C++: https://github.com/apache/arrow/pull/37792
                - Go: https://github.com/apache/arrow/pull/35769
            - Rust implementation is in the works: https://github.com/apache/arrow-rs/pull/4585
        - ListView
            - Vote planned early next week
            - C++ and Go implementations in the works
                - C++: https://github.com/apache/arrow/pull/35345
                - Go: https://github.com/apache/arrow/pull/37468
- Proposed nanoarrow 0.3 release
    - Please vote: https://lists.apache.org/thread/5o22xwbgh1628ylsnhb8ospzxc7wtdr1

2023-09-13

Attendees

- Raul Cumplido Dominguez
- Dane Pitkin
- Xuwei Fu
- Sri Nadukudy
- David Li
- Ben Harkins
- Jin Shang
- Matthew Topol
- James Duong
- Rok Mihevc

Discussion

- Arrow v14
    - Code freeze Oct 10, 2023
- Upcoming Parquet filtering improvements
    - [WIP] Arrow Parquet Filtering
    - Index pruning
    - Lazy evaluation

2023-08-30

Attendees

- Matthew Topol
- Ian Cook
- Ben Harkins
- Will Jones
- Joel Lubinitsky
- Rok Mihevc
- Joris Van den Bossche
- Jacob Wujciak
- Dewey Dunnington
- Xuwei Fu
- David Li
- Jin Shang
- Dane Pitkin
- Sri Nadukudy

- Weston Pace
- David Li

Discussion

- Looking for comments on VariableShapeTensorArray extension (https://lists.apache.org/thread/qc9qho0fg5ph1dns4hjq56hp4tj7rk1k, https://github.com/apache/arrow/pull/37166)
    - Feedback requested on ML or PR
    - Discussion about whether the number of dimensions (i.e. length of the shape array) should be fixed and known in advance, or not ("ragged tensors")
        - E.g. It supports "A collection of vectors (of different lengths)" and "A collection of matrices (of different sizes)" But it does not support "A mixed collection of vectors and matrices"
        - Discussion about what other libraries do, for example:
            - Ray https://github.com/ray-project/ray/blob/42a8d1489b37243f203120899a23d919dc85bf2a/python/ray/air/util/tensor_extensions/arrow.py#L553
            - The pytorch nested_tensor object: https://pytorch.org/docs/stable/nested.html
        - On List vs FixedSizeList for dims: https://github.com/apache/arrow/pull/37166/files#r1310524404
    - Related discussion about whether nested nullable FixedSizeList is well-handled by the current Parquet implementation
        - Xuwei has a PR to fix this at https://github.com/apache/arrow/pull/35694 but it has poor performance
- PyArrow Dataset as a protocol
    - https://lists.apache.org/thread/ko0j6pk86p5rt24w6s3m40h68r6lcqrr
    - General discussion about what approach to take

[quick draft Joris]

```
def __arrow_dataset__(self) -> ArrowDatasetABC


class ArrowDatasetABC:

    def get_schema(self) -> capsule[ArrowSchema]:
        ...

    def get_stream(self, columns, filter) -> capsule[ArrowArrayStream]: #
Optionally pass schema here (opportunity to negotiate/fix schema?)
        ...
```

```
    def get_partitions(self) -> list[ArrowDatasetADBC]: # or does it need
to be separate object?
        # simple version when not supporting partitions
        return [self]
        ...
```

- Questions:
    - Way to specify number of partitions, use of threading, batchsize, ..?
    - “Tied to Arrow, but not tied to pyarrow” (by leveraging the C Data Interface)
    - Requires finishing PyCapsule proposal: https://github.com/apache/arrow/issues/34031
- Regarding using Substrait to represent expressions:
    - Currently Substrait expressions are always bound to a schema
    - Can we get unbound expressions in Substrait?
        - Yes maybe, if someone can advocate for this and drive the work
        - The important thing is for user libraries to allow users to initially construct unbound expressions, but libraries can add the schema before serializing it to Substrait format
        - For example the Python Substrait library could do this https://github.com/substrait-io/substrait-python

2023-08-16

Attendees

- Ian Cook
- Raúl Cumplido
- Alenka Frim
- Xuwei Fu
- David Li
- Steve Lord
- Rok Mihevc
- Sri Nadukudy
- Weston Pace
- Dane Pitkin
- Jin Shang
- Ruoxi Sun
- Matthew Topol

Discussion

- Release 13.0.0 status

    - Blocking regressions have been resolved, but we are still working to solve a couple of remaining issues, particularly this CMake issue that prevents us from releasing the Java JARs: https://github.com/apache/arrow/issues/37201
- Plans for pandas to add a hard dependency on PyArrow
    - Proposed in PDEP-10: https://github.com/pandas-dev/pandas/blob/2db0037b10aaa14994b307cbe64ff82b7c1dc260/web/pandas/pdeps/0010-required-pyarrow-dependency.md
    - Approved by pandas core developers: https://github.com/pandas-dev/pandas/issues/54106#issuecomment-1656186946
    - This is planned for the pandas 3.0 release
    - The main reason for this is to speed up the string dtypes in pandas by making the PyArrow string type the default
    - Relatedly, scikit-learn is considering switching to Polars because of this, because with PyArrow, pandas is too large
    - pandas uses PyArrow compute, so it would not be practical to use a minimal Arrow Python library like nanoarrow
        - pandas 2.0 example of using acero kernel: https://github.com/pandas-dev/pandas/blob/1c5c4efbad2873d137089a1fd32267f40c966850/pandas/core/arrays/arrow/array.py#L1072
    - This creates a need for more development and maintenance of the Arrow C++ compute kernels, and possibly Acero if pandas will use that
        - Relatedly: Weston will email the ML about opportunities to get more involved in Acero
- Variable-shape tensor extension type PR
    - https://github.com/apache/arrow/pull/37166
        - Is relatively trivial for now because it does not implement strides
    - Rok will start a discussion on the mailing list
    - Matt will look into implementing both the fixed-shape and variable-shape tensor extension types in Go
- Substrait–Arrow expression translation PR
    - https://github.com/apache/arrow/pull/34834
    - Python reviewers requested

2023-08-02

Attendees

- Ian Cook
- Xuwei Fu
- David Li
- Rok Mihevc

- Sri Nadukudy
- Ashish Paliwal
- Dane Pitkin
- Ruoxi Sun
- Matthew Topol

Discussion

- Release 13.0.0 status
  - RC2 is out
  - There were two perf regressions in the earlier RC; one is now fixed but the other is not (wide tables)
  - TBD: whether the wide tables regression should block the release
    - 10x slowdown
    - Ben says he should be able to get a fix for this by the end of this week
    - This could be a painful regression for some users, especially those using Arrow with financial data
    - Benchmark: https://conbench.ursa.dev/compare/runs/9cf73ac83f0a44179e6538b2c1c7babd...3d76cb5ffb8849bf8c3ea9b32d08b3b7/
    - Benchmark on RC2 (search for 'wide-dataframe'): https://conbench.ursa.dev/runs/833d1302bd98477caaaf2fa428db0512/
    - Consensus: let's wait for the fix and make a new RC that includes that
    - Dane will check with Raul and Jacob about availability to do RC3
- ADBC 1.1.0 status
  - Nearly done with draft implementations (https://lists.apache.org/thread/0csbol3w5dqpnjwh01ny2t1pwv7wptjm)
  - One question for Matt in the PR
  - Vote planned for next week
- Apache Con China presentation
  - https://github.com/apache/apachecon-acasia/blob/master/content/sessions/datalake-1053.md
- Discussion about https://github.com/apache/arrow/issues/35638
- PRs needing review
  - Bugfix for Parquet writing Boolean: https://github.com/apache/arrow/pull/36972
  - C++ implementation of Float16 for Parquet: https://github.com/apache/arrow/pull/36073
    - Java counterpart is in the works now
    - Two implementations are required for a vote by the Parquet PMC

2023-07-19

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- Xuwei Fu
- Will Jones
- David Li
- Aldrin Montana
- Dane Pitkin
- Antoine Pitrou
- Jin Shang
- Jacob Wijciak

Discussion

- Release 13.0.0 status
    - Feature freeze was last Monday
    - There were some patches since then
    - Today Raúl generated an RC
    - Verification, etc. is ongoing
    - Expect an email in the next few days about the RC
    - Raúl is checking for perf regressions with Conbench
- Investigating an apparent regression
    - https://conbench.ursa.dev/compare/benchmark-results/064b6f97b26470af8000f3e60807c69c...064b7e982f7b78b8800060323e1c7997/
    - This is unrelated to the 13.0.0 release; the regression occurred after a commit that was made after the release freeze
- Geospatial family of extension types
    - Active work ongoing at https://github.com/geoarrow/geoarrow
    - Question about why this work is happening to the side of Arrow, not directly through the Arrow canonical extension type mechanism
        - Because it is not just one extension type, it's a family of them, and it would probably be awkward to have them all be rendered at https://arrow.apache.org/docs/format/CanonicalExtensions.html, where they would crowd out the other canonical extension types
        - Because the developers are interested in increasing adoption of Arrow within the geospatial data community, and having the Arrow geospatial extensions logically separated out like this makes it easier to show this community only the representations that matter to them

        - However, it might be good to add a note about geoarrow in https://arrow.apache.org/docs/format/CanonicalExtensions.html with a link to to https://github.com/geoarrow/geoarrow

2023-07-05

Attendees

- Ian Cook
- Raúl Cumplido
- Alenka Frim
- Xuwei Fu
- David Li
- Matthew Topol
- Gang Wu
- Joris Van den Bossche
- Antoine Pitrou
- Jacob Wujciak
- Dane Pitkin
- Weston Pace
- Sri Nadukudy

Discussion

- Release 13.0 status
    - Plan is still to do code freeze on Monday July 10
    - There are still some nightly test failures and other blockers, see summary sent yesterday
    - There are currently 28 open issues marked for the 13.0.0 release
        - 4 are blockers
        - Blockers: https://github.com/apache/arrow/issues?q=is%3Aopen+is%3Aissue+milestone%3A13.0.0+label%3A%22Priority%3A+Blocker%22
    - Conbench maintainers will help us provide a way to compare the performance of the previous release with the RC
        - The idea is that we will provide a link to Conbench in the release verification email
        - This will probably be a more manual process for this release, but for the next release we hope to automate it
    - Any particular issues need attention?
        - Spark integration issue https://github.com/apache/arrow/issues/36332
            - Is tagged as a blocker currently

                - Dane will ask Davide to take a look
            - https://github.com/apache/arrow/pull/34834 could use review
                - Joris will continue reviewing
            - Azure filesystem integration PR?
                - Most likely not going to make it into the 13 release, but the PR author has been doing active work on it with help from maintainers
    - Thanks for attention to nightly failures
- WebAssembly (see mailing list message)
    - ML message: https://lists.apache.org/thread/50k6x7ksry4ml8lnxfx4omv7hvjyg3r7
    - PR to disable threading at compile time: https://github.com/apache/arrow/pull/35672
    - Would be nice to get this in by the 13.0.0 release, but this might not be realistic
    - What are the motivations for this?
        - Main one would be enabling PyArrow to work in Pyodide, which would enable integrations with pandas to run in browser
        - This is related to the discussion about pandas declaring PyArrow as a required dependency (https://github.com/pandas-dev/pandas/pull/52711)
    - What are the main challenges?
        - Selectively disabling unit tests that rely on threading
        - Enabling some threaded code to serially execute under the hood
    - Related: the Go Arrow implementation recently did work to enable WASM builds with tinygo
- (Matt) Major/Minor Versioning / breaking change labels etc.
    - Time to revisit the question of whether we should move from our current release versioning scheme (incrementing the major version number with each quarterly release) to a scheme that is more aligned to semver (only incrementing major version number when there is a breaking change, otherwise incrementing the minor version number in each quarterly release)?
    - The case for this: incrementing the major version for each quarterly release (even when there are no breaking changes) makes it harder for users to upgrade, doesn't follow the spirit of semver, complicates the release and packaging process for some of the monorepo language libraries. Being more disciplined about breaking changes will be good for the project. We already have tags, etc. that should make it straightforward to do this. Users will be less reluctant to upgrade to a new quarterly release if it is a minor release because they won't need to do a full process of checking for breaking changes
    - The case against this: We have so many different language libraries in the monorepo which all need to follow the same versioning scheme; in practice at least one of these has a breaking change in every quarterly release; trying to time the merging of PRs containing breaking changes will dramatically overcomplicate things and create a lot of extra work e.g. maintaining multiple branches and doing backports; this will risk causing the different language libraries maintained in the monorepo to have to adopt different versioning schemes; although some users are reluctant to upgrade to new major versions,

    - others are the opposite—they want to skip minor versions because they are seen as possibly not worth upgrading to
    - Maybe we can find a compromise solution, e.g. doing a major release every 6 months instead of 3 months and doing more frequent minor releases between them?
    - Is it possible / allowed to do different release versioning schemes for different libraries / components maintained in a single ASF project repo?
        - Yes, Iceberg does this
    - Educating other projects that use Arrow libraries could solve some of the same problems
        - For example discouraging libraries that depend on PyArrow from pinning the version
        - There is an understanding already within the Python ecosystem that setting an upper pin is harmful to the ecosystem
            - See https://iscinumpy.dev/post/bound-version-constraints/
    - Ian will gather the points made in this discussion into a doc, and share it (but probably not for 1–2 weeks)
- (Joris) Re-organizing our github labels (https://docs.google.com/document/d/1Q0kUaZuHCqes0OOtkS3WyNvrm6kiL_8cexUetSYFi0Y/edit)

2023-06-21

Attendees

- Ian Cook
- Raúl Cumplido
- Xuwei Fu
- Will Jones
- David Li
- Rok Mihevc
- Sri Nadukudy
- Antoine Pitrou
- David Susanibar
- Matthew Topol
- Jacob Wujciak

Discussion

- ADBC 1.1.0

    - As mentioned on mailing list (https://lists.apache.org/thread/7qd07z0sl8x9xp0p2sfd5cgdo96xgscj) there is a proposal to add several new APIs
    - PR open at https://github.com/apache/arrow-adbc/pull/765 ; currently has the C implementation
    - Main focus is richer error handling / error messages, statistics, and metadata hierarchies
    - Comments appreciated
    - Intention is to collect feedback, complete implementations/prototypes in Java and Go, then call vote
- Async Flight (finally for real?)
    - Related issues: https://github.com/apache/arrow/issues/16604 https://github.com/apache/arrow/issues/34607
    - PR from David coming later today with an initial implementation of an async client-side API for C++ and numerous related improvements
    - There will be follow-up PRs to implement the server-side pieces, etc.
- Arrow 12.0.1
    - Release is complete
    - Post-release tasks are almost complete
    - There is a glitch affecting source distributions of PyArrow (not wheels) https://github.com/apache/arrow/issues/36065
        - Expectation is that this should not be a major blocker for anyone; this has affected a small number of users on platforms for which we don't publish wheels e.g. Alpine Linux; there are workarounds available; a 12.0.2 patch release seems unnecessary
        - Does PyPI provide a way to replace the artifact?
- Arrow 13.0.0
    - Code freeze is planned for around 10 July
    - There are discussions happening with the Conbench maintainers to ensure that we can do a performance benchmark comparison as part of the release verification process, to avoid a performance regression vs. the previous release similar to what happened in 12.0.0

2023-06-07

Attendees

- Ian Cook
- Raúl Cumplido
- Will Jones
- David Li
- Bryce Mecum
- Sri Nadukudy

- Dane Pitkin
- Matthew Topol

Discussion

- Arrow v12.0.1 Release
    - Raúl began creating 12.0.1 RC0 on Monday but ran into a few issues
    - Raúl created RC1 today and is investigating some Homebrew issues
    - Raúl will create an RC2 once all known issues are resolved and will post to the mailing list to start a vote
    - See Zulip chat for updates
- Any follow up on benchmark before release?
    - Raúl has been talking with the Conbench developers about building new features to simplify this, but no work is complete yet
- JSON extension type proposal could use attention
    - https://lists.apache.org/thread/p3353oz6lk846pnoq6vk638tjqz2hm1j
    - https://github.com/apache/arrow/pull/13901

2023-05-24

Attendees

- Matt Topol
- Will Jones
- Bryce Mecum
- David Li
- Weibin Zeng
- Xuwei Fu
- Joris Van den Bossche
- Sri Nadukudy
- David Susanibar
- Ashish Paliwal
- Dane Pitkin
- Jacob Wujciak
- Weston Pace

Discussion

- ADBC 1.1.0 proposals
    - PR open for review: https://github.com/apache/arrow-adbc/pull/692
    - ML vote will come afterwards

- C Device Data ABI
    - Adding C/Arrow/Stream device arrays
        - https://github.com/apache/arrow/pull/34972
    - Will be marked as experimental until wider adoption occurs (~6mo)
        - We might want to be careful about leaving experimental tags on too long, like what has happened with other arrow components
    - Request for comments underway
    - ML vote underway: https://lists.apache.org/thread/o2hsw7o1gm3qgw5z51rmz6zqxh0p7bvk
    - If adopted and merged, will start work on helpers in the Arrow libraries
    - A prototype was created passing data between python Numba and libcuDF on GPU
- Arrow v12.0.1
    - Mandatory for Arrow R
    - Should release for other Arrow libraries to fix performance regressions
        - Weston Pace has a pending fix to be merged
        - Matt Topol has Go Arrow issues to merge (nice-to-have)

2023-05-10

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- Xuwei Fu
- Will Jones
- David Li
- Ashish Paliwal
- Dane Pitkin
- Matthew Topol
- Joris Van den Bossche

Discussion

- Arrow 12.0.0 release
    - Release is complete
    - Most post-release tasks are complete, except for vcpkg port update (which should be done soon) and conan (still pending)
    - We need the Parquet C++ issues to be tagged properly in Jira
        - These issues are: PARQUET-2201, PARQUET-2225, PARQUET-2232, PARQUET-2250

        - We need to:
            - Tag them with the appropriate fix version (cpp-12.0.0)
            - Mark the cpp-12.0.0 version as closed
            - Create a new cpp-13.0.0 version
        - Xuwei will notify Gang Wu and Micah
- Arrow 12.0.1 release?
    - See issues tagged with the 12.0.1 milestone
    - https://github.com/apache/arrow/issues?q=is%3Aopen+is%3Aissue+milestone%3A12.0.1
    - In particular, see the performance regression in https://github.com/apache/arrow/issues/35498
        - This is related to https://github.com/apache/arrow/issues/33313
- Benchmark checking as part of the release process?
    - One of the regressions (https://github.com/apache/arrow/issues/35498) was flagged by a benchmark but we didn't take any action before the release
    - Example: https://conbench.ursa.dev/benchmark-results/2b587cc1079f4e3a97f542e6f11e883e/
    - Perhaps as a part of the release verification process we should make it easier for reviewers to see a Conbench page directly comparing the performance of the current release candidate with the previous release
        - This would be useful not just for detecting performance regressions but also for seeing big areas of performance improvement that we can mention in the release notes / blog post / etc.
        - Raúl will take some next steps on this
- s390x CI migration from Travis
    - ASF has already switched off all Travis jobs (about 3 months ago) so this job has not been running
    - Raúl is still working on this

2023-04-26

Attendees

- Ian Cook
- Raúl Cumplido
- Xuwei Fu
- Will Jones
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy

- Matthew Topol

Discussion

- Arrow 12.0.0 RC 0 status
  - There were a lot of CI failures at the time of the code freeze so it took longer than usual to resolve these and generate RC0; thanks to everyone who helped
  - There is one outstanding question regarding an issue with pandas 2.0.1 https://github.com/pandas-dev/pandas/issues/52899
    - There is a fix that skips the failing test https://github.com/apache/arrow/pull/35324
    - It is unclear whether we should create a new RC that skips this test, or whether it is sufficient to release the current RC since pandas will fix the issue on their end
  - There are a couple of other minor issues that we don't think are blockers
- Follow-up discussion about Parquet C++ issue tracking
- Support for non-CPU memory in Arrow C data interface (https://github.com/apache/arrow/pull/34972)
  - We are looking for consensus before starting a vote
  - We are looking for input that addresses the questions posed and gives concrete recommendations
- Questions about usage of new tensor extension type https://arrow.apache.org/docs/dev/format/CanonicalExtensions.html#official-list
  - Can this be written to a Parquet file and read back in? If so, what Parquet logical and physical types does it use? https://arrow.apache.org/docs/cpp/parquet.html#logical-types
    - Fixed size list? Parquet doesn't have this. Possibly related issue: https://github.com/apache/arrow/issues/34510
  - Is it recommended for use with image data, or should we use byte arrays instead?
- Integration tests for C data interface
  - Status? Has not been implemented yet
  - Proposal mailing list thread: https://lists.apache.org/thread/nr05xwls713xpsxkobpln2f6wsdntrky
- Suggestion for next meeting: discuss priorities for Arrow 13.0.0 release

2023-04-12

Attendees

- Ian Cook

- Raúl Cumplido
- Xuwei Fu
- Will Jones
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Ashish Paliwal
- Dane Pitkin
- David Dali Susanibar Arce
- Matthew Topol
- Joris Van den Bossche
- Jacob Wujciak

Discussion

- 12.0.0 release
    - Code freeze is scheduled for later today, April 12
    - There are many nightly failures currently on main; Raúl and Jacob have opened several blocker issues and we might need to create more
    - Any PRs we should try to merge before code freeze?
        - https://github.com/apache/arrow/pull/34984
        -
        -
    - After code freeze, nightly jobs can be run on the release branch
    - Discussion of several current issues that might affect the release
        - C# tests not finding Python
        - Pyarrow tests slowness on Windows (https://github.com/apache/arrow/issues/35078)
        - Python wheels on Windows not uploading to Gemfury
    - Important items to include in release changelog, release blog, etc.
        - Drop support for Ubuntu 18.04 https://github.com/apache/arrow/issues/33800
        - Acero refactor (splitting Acero out from core Arrow library) https://lists.apache.org/thread/5h5g9k9lvbybzl8fnbg4fppxczm42g6r
        - Tensor extension type https://arrow.apache.org/docs/dev/format/CanonicalExtensions.html#fixed-shape-tensor
        - REE layout https://arrow.apache.org/docs/format/Columnar.html#run-end-encoded-layout
        - Plasma removal https://github.com/apache/arrow/pull/34718
            - Suggested alternatives https://lists.apache.org/thread/lk277x3b9gjol42sjg27bst2ggm5s0j2

        - Reminder about Jira to GitHub move (which happened just before the 11.0.0 release)
        - Initial Swift implementation https://github.com/apache/arrow/issues/20484
        - nanoarrow (not technically a part of this release, but worth drawing attention to)
        - Also see ASF board report https://docs.google.com/document/d/13FSDydEVXT2UUFdy4XKjVKNJW-WR8ylvG3aI6lD-dNI/
- Parquet tickets are still tracked in the ASF Jira
    - We have to maintain a lot of code in Archery, etc. to automate the tracking of Parquet C++ issues which are still in Jira, even though there are only a few Parquet issues in each release (4 for 12.0.0)
        - PARQUET-2201 Add stress test for RecordReader ReadRecords and SkipRecords. (#14879)
        - PARQUET-2225 Allow reading dense with RecordReader (#17877)
        - PARQUET-2232 Add an api to ColumnChunkMetaData to indicate if the column chunk uses a bloom filter (#33736)
        - PARQUET-2250 Expose column descriptor through RecordReader (#34318)
    - Can we move the Parquet C++ issues from the ASF Jira to GitHub?
        - Joris believes we can go ahead and do this; the Parquet Rust implementation did something similar
        - Related ML discussion: https://lists.apache.org/thread/crwwfdh6gw7mnvyh7mnrcc8jkxqhqdno
        - There are already some Parquet issues that were reported and resolved in the Arrow monorepo in this release without ever being opened as Parquet Jira issues: https://github.com/apache/arrow/issues?q=is%3Aissue+label%3A%22Component%3A+Parquet%22+is%3Aclosed
        - We should check with Micah Kornfield, Fatemah Panahi
        - There was a related Parquet mailing list discussion about this in
        - February: https://lists.apache.org/thread/jf9wos3t6xxk6xdyx2dof1jlkbpkr56p

2023-03-29

Attendees

- Ian Cook
- Will Jones
- David Li
- Rok Mihevc

- Sri Nadukudy
- Dane Pitkin

Discussion

- Rust ADBC progress update
    - https://github.com/apache/arrow-adbc/pull/478
    - How are ADBC drivers versioned, packaged, released?
        - ADBC drivers are packaged in native language, and can also be released in Conda as well as within Python wheels
        - If a Rust driver were created, it could be released along with a Python wheel if we wanted. That could be versioned and released with the Rust library, if desired.
        - ADBC libraries and drivers are coupled in version for convenience of release process.
- Follow up regarding Plasma removal
    - Kou removed Plasma in https://github.com/apache/arrow/pull/34718
    - See conversation about this at https://lists.apache.org/thread/lk277x3b9gjol42sjg27bst2ggm5s0j2
        - Further discussion about Plasma alternatives welcome in that thread
        - Weston left some good comments on the example Will linked to: https://github.com/wjones127/arrow-ipc-bench/pull/1
            - Will will reply to the thread with updates based on that
                - https://lists.apache.org/thread/yp8bqoggfll29yjrzp5w68djphrq2zzz
- 12.0.0 release
    - Kou confirmed that he can help Raúl with the release

2023-03-15

Attendees

- Ian Cook
- Raúl Cumplido
- Alenka Frim
- Will Jones
- David Li
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Weston Pace
- Dane Pitkin

- Matthew Topol
- Joris Van den Bossche

Discussion

- Arrow 12.0.0 Release
    - Planned date for code freeze April 11th.
    - Plan is for Raúl and Kou to collaborate to manage the release similar to the 11.0.0 release
        - [Ian] Is this process reasonable and sustainable in terms of the workload?
        - [Raul] Yes, for the release process itself; but there are some post-release tasks that have been delayed, e.g. conan and vcpkg
    - What big projects / critical bugs do we need to work on
        - Ordered execution https://github.com/apache/arrow/issues/32991
        - Incorrect hash join results: https://github.com/apache/arrow/issues/34474
        - Scanner updates (maybe)
        - Performance regressions from row default Parquet group size change https://github.com/apache/arrow/issues/34374
        - Slow Parquet reads: https://github.com/apache/arrow/issues/34319
        - Other critical bugs?
            - https://twitter.com/MurrayData/status/1635679008426799111?s=20
    - What other things would we like to see get into the release?
        - Tensor implementation https://github.com/apache/arrow/pull/8510
            - Look into slow FixedSizeList conversion to Parquet https://github.com/apache/arrow/issues/34510 (not sure if we can solve it though)
        - PyArrow bindings to Acero (+Removing custom Cython ExecPlan usage) https://github.com/apache/arrow/pull/34401
- Please tag any issues that should block the release with: label:"Priority: Blocker" and milestone:"12.0.0"
    - Are the tagging conventions documented anywhere yet?
        - We should document how to tag release blockers in GHA
        - It's done; see https://arrow.apache.org/docs/dev/developers/reviewing.html#labelling and https://github.com/apache/arrow/blob/main/docs/source/developers/reviewing.rst#labelling
- Removing Plasma
    - Plasma has been deprecated as of the 10.0.0 release, and its removal is scheduled to happen in the 12.0.0 release
        - https://lists.apache.org/thread/nw232k2lzmg9kcl8ts475m9ybl34j81p
    - There is ongoing investigation into which alternatives we should recommend to replace Plasma in different use cases

    - Should we consider deferring its removal to 13.0.0? Would we have a better idea of which alternatives to recommend by that time? What additional maintenance burden might we incur by keeping it in the codebase for another 3 months?
        - There was some inconclusive conversation on these points
        - For some uses of Plasma, it could be replaced with Flight. For others not
    - The maintainers intend to proceed as per this plan
    - Will Jones will send an email to the dev@ and user@ lists to provide a reminder and seek more input on this
        - https://lists.apache.org/thread/1mrx0qg8dflshc4k0fv7g5qm775yr282
    - In PyArrow, it would have been best to change the deprecation warning to a future warning
        - If we end up we deferring the removal to after 12.0.0, we should do this

2023-03-01

Attendees

- Ian Cook
- Raúl Cumplido
- Dewey Dunnington
- Ian Joiner
- Will Jones
- David Li
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Weston Pace
- Dane Pitkin

Discussion

- Fixed Shape Tensor canonical ExtensionType proposal
    - It seems like we have converged to the final state at this point, but we are waiting for a few conversations to conclude
    - Alenka called a vote at https://lists.apache.org/thread/3cj0cr44hg3t2rn0kxly8td82yfob1nd but this sparked some additional feedback so she plans to give it a few more days then open a new vote next week
- PR automation Workflow
    - Proposal discussed on mailing list has been implemented https://lists.apache.org/thread/1rhsd8ovy4bfr8hcdohn0vh65frw0ggk
    - There are a few hiccups that Raúl is working out
    - Feedback welcome

- Self-hosted arm64 runners
  https://lists.apache.org/thread/mskpqwpdq65t1wpj4f5klfq9217ljodw
    - Raúl has been working with ASF Infra and has set up a GitHub integration to add self-hosted runners at the organization level, which allows us to use them from multiple arrow repos in the apache organization on GitHub
    - This will allow us to retire some Travis CI jobs
        - But Travis CI will continue to be used for some Crossbow jobs, e.g. for s390x (big-endian)
        - Dewey has used `DOCKER_DEFAULT_PLATFORM=linux/s390x docker run alpine:latest` to test nanoarrow on s390x
- Initial nanoarrow release candidate!
  https://lists.apache.org/thread/slomdw52n9j7jq8zwl5v8cb4v8yfk9sj
    - We are looking for people to verify the RC
- Default Parquet row group size change
    - https://github.com/apache/arrow/pull/34281
    - This is specific to the Arrow C++ implementation and its bindings
    - Before this change, the default row group size was 64 million rows
        - This was based on a misunderstanding and is much too large
    - Weston has changed the default to 1 million rows
        - There was some discussion about whether this should be something smaller e.g. 100K rows, but overall there were no objections
    - This caused a performance regression to write performance, which Weston is investigating
        - https://github.com/apache/arrow/issues/34374
    - Is it possible to set the row group size based on bytes instead of rows?
        - There was a recent change that should enable this: https://github.com/apache/arrow/pull/33897

2023-02-15

Attendees

- Anja Boskovic
- Ian Cook
- Dewey Dunnington
- Ian Joiner
- Will Jones
- David Li
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Dane Pitkin
- Matthew Topol
- Jacob Wujciak

Discussion

- Fixed Shape Tensor canonical ExtensionType proposal
    - mailing list discussion about it
    - PR specifying and documenting it
    - PR implementing it in C++
    - PR implementing it in Python
    - Alenka, Joris, Rok are preparing a proposal for canonical tensor extensiontype
    - This is useful for applications including deep learning
    - See links above for format change, ML discussion, C++ implementation, Python implementation
    - Alenka plans to call a vote this week or next week
    - Please give input as soon as possible
    - Members of Hugging Face, Ray, and PyTorch community have given input and some of it was incorporated (strides, transposition, …)
    - It would be good to have input from some other companies and project communities including Lance, NumPy, Posit, MATLAB, DLPack, CUDA/RAPIDS, Arrow Rust, Xarray, Julia, Fortran, TensorFlow, LinkedIn, Python Array API standard project
    - One item of possible disagreement is over row-major vs. column-major
    - Plan is to have zero-copy compatibility with PyTorch
- Flight RPC/Flight SQL/ADBC proposals
    - Several changes are proposed
    - Input welcome

- Hoping to get the nanoarrow 0.1 release to the mailing list early next week (may need some help with signing)
    - Most of the heavy lifting is done, but Dewey might need some help with signing keys
    - There are no binary packages being distributed, just a source tarball
- Improving the web of trust for signing
    - Can we look for opportunities to improve this to enable more committers to help with releases? Maybe a remote signing party
    - We need to check what is allowed under policies; see https://infra.apache.org/release-signing.html and https://infra.apache.org/release-publishing.html

2023-02-01

Attendees

- Ian Cook
- Nic Crane
- Raúl Cumplido
- Dewey Dunnington
- Will Jones
- David Li
- Bryce Mecum
- Rok Mihevc
- Sri Nadukudy
- Dane Pitkin
- Soumya Sanyal
- Matthew Topol
- Jacob Wujciak

Discussion

- Mailing list label/tag guidance for new contributors (Bryce Mecum)
    - Should we use tags like “[DISCUSS]” and “[RFC]” in addition to the language tags in the subject line of emails?
        - There is currently no documentation of what practices we should use to tag/label emails to the mailing lists, even for commonly used tags
        - Other common mailing list conventions (like saying whether your vote is binding or non-binding) are also not formally documented anywhere
        - For some users, it is not immediately obvious that they should label their emails with the language implementation
        - The consensus seems to be that it is worth documenting this on https://arrow.apache.org/community/

        - Bryce will open a PR
- Should Rust ADBC libraries be in apache/arrow-adbc? (Will Jones)
    - Should the Rust ADBC libraries be released per the Rust library release schedule or the ADBC library release schedule?
    - Considerations include: whether it will be used within the Rust ecosystem (or as a standalone tool that uses Rust); which component it should have tighter integration testing with; what is most convenient for development
- Known alternatives to Plasma (https://arrow.apache.org/docs/python/plasma.html) that we can point users to? (Will Jones)
    - For context: Plasma was added to Arrow C++ by Ray developers, but has no active maintainers any longer and is deprecated and planned for removal in 12.0.0 (https://lists.apache.org/thread/nw232k2lzmg9kcl8ts475m9ybl34j81p)
    - Plasma continues to exist as an internal utility in Ray (https://discuss.ray.io/t/plasma-store-apis/5421/6)
    - Weston Pace has been considering how we might solve some of the problems that Plasma solves, but by building on existing Arrow interfaces instead of taking a general-purpose approach like Plasma
- Release 11.0.0 status (Raúl Cumplido)
    - Arrow 11.0.0 has been released
    - There are some post-release tasks still in progress, including downstream packaging and distribution tasks
    - Raúl will merge the blog post PR and make an announcement on the mailing list soon
- PR workflow automation (Raúl Cumplido)
    - Raúl has proposed to implement some automation to improve the PRs and issues workflows; feedback is welcome in the mailing list thread
    - See mailing list thread (https://lists.apache.org/thread/1rhsd8ovy4bfr8hcdohn0vh65frw0ggk)
- Canonical TensorArray extension type (https://github.com/apache/arrow/pull/33925) (Rok Mihevc)
    - This would be the first canonical extension type since we adopted the framework for that (https://lists.apache.org/thread/qxc1g7h9ow79qt6r7sqtgbj8mdbdgnhb)
    - Looking for input from users/developers who are familiar with working with tensor/multidimensional array data
- nanoarrow release process (Dewey Dunnington)
    - Dewey is hoping to do a 0.1 release candidate in the next couple of weeks
- Jira to GitHub migration (Ian Cook)
    - There was a discussion in the previous biweekly meeting about how with GitHub Issues we cannot associate bug issues with two milestones—one representing the next (possible/actual) maintenance release and one representing the next major release—like we used to with Jira; the newly proposed “backport candidate” provides a solution to this (https://lists.apache.org/thread/38xsz3ycr6jghv6h0d4bsb2y0z093lkf)

    - The migration dry-run repos discussed in the previous meeting have been deleted
    - Some users have reported that Jira offered richer options for filtering issues than GitHub does
- Can we better promote this and other Arrow community meetings? (Ian Cook)
    - Information about this meeting and the Arrow R developers meeting is shared in biweekly emails Arrow dev mailing list
    - The Arrow Rust community used to have a sync meeting but stopped having regular dedicated meetings in 2021
    - Do any other Arrow language sub-communities hold regular meetings?
    - We could better promote these biweekly meetings, not just on the mailing lists
    - Ian will open a PR to add information about these meetings to the Arrow Community page (https://arrow.apache.org/community/)

2023-01-18

Attendees

- Anja Boskovic
- Rusty Conover
- Ian Cook
- Raúl Cumplido
- Ian Joiner
- Will Jones
- Sean Kelly
- David Li
- Bryce Mecum
- Rok Mihevc
- Matthew Topol
- Jacob Wujciak

Discussion

- 11.0.0 release
    - Code freeze was on Monday
    - Release candidate was created today
    - Packaging and validation is underway
    - We encountered a few small problems but so far we have not needed to create a new release candidate
    - Help verifying the release is welcome once the RC is shared
        - We test on a large matrix of environments in our CI, but verification in real-world environments is helpful

- Jira to GitHub post-migration tasks (if any)
    - The migration from Jira to GitHub issues is now complete
        - Thank you to Rok, Todd, and all others who contributed to this
    - Anyone notice any problems?
        - There was a problem with selecting components, but Jacob fixed this
        - Some issues were removed from the 11.0.0 milestone but were not assigned a new milestone. As a general rule, should we assign a new milestone corresponding to the next major release when an issue is bumped out of release scope?
            - Probably not, because in the past, many of the issues that were assigned Fixed In versions in Jira were very old and had just been repeatedly bumped from one major release to the next
            - It might be more meaningful to track coarser-grained initiatives/projects for roadmap planning, rather than fine-grained issues
                - We could use the Projects feature in GitHub for this
        - Note that each GitHub Issue can only have one milestone associated with it, so we cannot assign bugfixes to a maintenance release and to the next major release like we often did in Jira
        - Note that GitHub Issues can have multiple assignees, unlike in Jira
        - We need to have a broader discussion about our maintenance policy and versioning conventions
            - See the notes from our discussion about that in the December 7 sync call: https://lists.apache.org/thread/gbywpzbvpfydq24m1c0w6jgybnsrf9xm
    - Keeping “GH-” as the issue prefix for PRs?
        - Changing this would break the auto-linking that GitHub does between issues and PRs, so we should probably keep it for now
        - Other repos like CPython also use the “GH-” prefix
    - Appropriate use of “wip” tag in GitHub Issues (if any)?
        - This is one of GitHub's built-in issue tags
        - Usually, assigning the issue to yourself signals that you are working on it, so using the "wip" tag does not seem necessary
        - Jacob and Raúl are looking at how to improve the developer experience, for example with a bot that removes the assignee for stale issues
    - Do we need to disable the Jira bots?
        - Jacob will do this
    - Can we delete the migration dry-run repos?
        - Rok will do this and ask Todd to do it
    - If you were previously subscribed to receive notifications on Jira issues, you will not automatically receive notifications from GitHub
        - Use the script provided by Rok to re-subscribe to notifications https://github.com/rok/arrow-migration/blob/main/transfer_arrow_subscript

ions.py

- Question about how best for contributors to communicate ideas before they have a PR ready
    - Open an issue
        - Appropriate for bugfixes, minor enhancements, etc.
    - Send an email to the dev mailing list
        - Appropriate for larger-scale feature proposals, spec changes, etc.