

git_comments:

1. **** garrow_uint8_array_sum:** * @array: A #GArrowUInt8Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
2. **** garrow_double_array_sum:** * @array: A #GArrowDoubleArray. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
3. **** garrow_int16_array_sum:** * @array: A #GArrowInt16Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
4. **** garrow_int8_array_sum:** * @array: A #GArrowInt8Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
5. **** garrow_float_array_sum:** * @array: A #GArrowFloatArray. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
6. **** garrow_uint16_array_sum:** * @array: A #GArrowUInt16Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
7. **** garrow_uint32_array_sum:** * @array: A #GArrowUInt32Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
8. **** garrow_uint64_array_sum:** * @array: A #GArrowUInt64Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
9. **** garrow_int64_array_sum:** * @array: A #GArrowInt64Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
10. **** garrow_int32_array_sum:** * @array: A #GArrowInt32Array. * @error: (nullable): Return location for a #GError or %NULL. **** Returns:** The value of the computed sum on success, * If an error is occurred, the returned value is untrustful value. **** Since:** 0.13.0
11. IWYU pragma: export

git_commits:

1. **summary:** ARROW-4882: [GLib] Add sum functions
message: ARROW-4882: [GLib] Add sum functions Author: Yosuke Shiro
 <yosuke.shiro615@gmail.com> Author: Kouhei Sutou <kou@clear-code.com> Closes #3933 from shiro615/glib-sum and squashes the following commits: 1fe8805c <Kouhei Sutou> Add test for empty case 4393487b <Kouhei Sutou> Add NULL check adbe5ca5 <Yosuke Shiro> Fix typos 6ebe5a02 <Yosuke Shiro> Align c1361034 <Yosuke Shiro> Fix documents 3a079dfc <Yosuke Shiro> Implement a template function that can be used by all sum functions 316d3cae <Yosuke Shiro> Add sum functions

github_issues:**github_issues_comments:****github_pulls:**

1. **title:** ARROW-4882: [GLib] Add sum functions
body:

github_pulls_comments:

1. # [Codecov](https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=h1) Report > Merging [#3933] (https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=desc) into [master] (https://codecov.io/gh/apache/arrow/commit/c707822d01ccba02c823d14426188943ce7a4cdb?)

src=pr&el=desc) will ****increase**** coverage by `0.82%`. > The diff coverage is `n/a`. [![Impacted file tree graph](https://codecov.io/gh/apache/arrow/pull/3933/graphs/tree.svg?width=650&token=LpTCFbqVT1&height=150&src=pr)](https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=tree) ``diff @@ Coverage Diff @@ ## master #3933 +/- ##
===== + Coverage 87.74% 88.56% +0.82%
===== Files 727 593 -134 Lines 89499 79916 -9583
Branches 1252 0 -1252 ===== - Hits 78532 70780
-7752 + Misses 10849 9136 -1713 + Partial 118 0 -118 `` | [Impacted Files]
(https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=tree) | Coverage Δ | | |---|---|---| |
[cpp/src/plasma/thirdparty/ae/ae.c](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-Y3BwL3NyYy9wbGFzbWEvdGhpcmRwYXJ0eS9hZS9hZS5j) | `70.75% <0%> (-0.95%)` | :arrow_down: | | [python/pyarrow/tests/test_convert_pandas.py]
(https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-cHl0aG9uL3B5YXJyb3cvdGVzdHMvdGVzdF9jb252ZXJ0X3BhbmRhcy5weQ==) | `94.82% <0%> (ø)` | :arrow_up: | | [cpp/src/arrow/util/cpu-info.cc](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-Y3BwL3NyYy9hcnJvdj91dGlsL2NwdS1pbmZvLmNj) | `75% <0%> (ø)` | :arrow_up: | | [python/pyarrow/pandas_compat.py](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-cHl0aG9uL3B5YXJyb3cvdGVzdGFzX2NvbXBhdC5weQ==) | `93.68% <0%> (ø)` | :arrow_up: | | [go/arrow/math/uint64_amd64.go](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-Z28vYXJyb3cvbWF0aC91aW50NjRfYW1kNjQuZ28=) | | | | [go/arrow/memory/memory_avx2_amd64.go](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-Z28vYXJyb3cvbWVtb3J5L2l1bW9yeV9hdngyX2FtZDY0Lmdv) | | | | [js/src/enum.ts](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-anMvc3JjL2VudW0udHM=) | | | | [go/arrow/array/builder.go](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-Z28vYXJyb3cvYXJyYXkvYnVpbGRlci5nbw==) | | | | [js/src/Arrow.node.ts](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-anMvc3JjL0Fycm93Lm5vZGUudHM=) | | | | [js/src/schema.ts](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree#diff-anMvc3JjL3NjaGVtYS50cw==) | | | | ... and [128 more](https://codecov.io/gh/apache/arrow/pull/3933/diff?src=pr&el=tree-more) | | -----
[Continue to review full report at Codecov](https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=continue). > ****Legend**** - [Click here to learn more](https://docs.codecov.io/docs/codecov-delta) > `Δ` = absolute <relative> (impact), `ø` = not affected, `?` = missing data` > Powered by [Codecov](https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=footer). Last update [c707822...1fe8805](https://codecov.io/gh/apache/arrow/pull/3933?src=pr&el=lastupdated). Read the [comment docs](https://docs.codecov.io/docs/pull-request-comments).

2. Thanks!

github_pulls_reviews:

- Can we create a template function that can be used by all `sum` functions? ``diff diff --git a/c_glib/arrow-glib/basic-array.cpp b/c_glib/arrow-glib/basic-array.cpp index deedc0fe..4db6a564 100644 --- a/c_glib/arrow-glib/basic-array.cpp +++ b/c_glib/arrow-glib/basic-array.cpp @@ -92,6 +92,28 @@
garrow_numeric_array_get_sum_value_raw(std::shared_ptr<arrow::Scalar> scalar) return arrow_numeric_scalar->value; }; +template <typename ArrowType, typename GArrowArrayType> +typename ArrowType::c_type +garrow_numeric_array_sum(GArrowArrayType array, + GError **error, + const gchar *tag, + typename ArrowType::c_type default_value) +{ + auto arrow_array = garrow_array_get_raw(GARROW_ARRAY(array)); + auto memory_pool = arrow::default_memory_pool(); + arrow::compute::FunctionContext context(memory_pool); + arrow::compute::Datum sum_datum; + auto status = arrow::compute::Sum(&context, arrow_array, &sum_datum); + if (garrow_error_check(error, status, tag)) { + using ScalarType = typename arrow::TypeTraits<ArrowType>::ScalarType; + auto arrow_numeric_scalar = + std::dynamic_pointer_cast<ScalarType>(sum_datum.scalar()); + return arrow_numeric_scalar->value; + } else { + return default_value; + } +} + G_BEGIN_DECLS /** @@ -1062,16 +1084,10 @@ gint8 garrow_int8_array_sum(GArrowInt8Array *array, GError **error) { - auto arrow_array = garrow_array_get_raw(GARROW_ARRAY(array)); - auto memory_pool = arrow::default_memory_pool(); - arrow::compute::FunctionContext context(memory_pool); - arrow::compute::Datum sum_datum; - auto status = arrow::compute::Sum(&context, arrow_array, &sum_datum); - if (garrow_error_check(error, status, "[int8-array][sum]")) { - return

```
    garrow_numeric_array_get_sum_value_raw<arrow::Int64Type>(sum_datum.scalar()); - } else { - return  
    0; - } + return garrow_numeric_array_sum<arrow::Int64Type>(array, + error, + "[int8-array][sum]", + 0);  
} ``
```

2. Should we use `gint64` instead of `gint8`?
3. "0 on error" is misleading. 0 is one of a valid value. Users should not use 0 as an error indicator.
4. Thanks! I've updated a template function.
5. Yes. I've fixed it.
6. I've fixed documents.
7. Note that it may also return NULL, and this should not be an error, so the "true" return type is
`option<T>`.
8. Thanks. We need to add a NULL check, then.
9. I've added. We use zero for the case because we can't implement `option<T>` in C for primitive types.
10. Oh, sorry. I missed it.

jira_issues:

1. **summary:** [GLib] Add "Sum" functions
description:

jira_issues_comments:

1. Issue resolved by pull request 3933 [<https://github.com/apache/arrow/pull/3933>]