

Apache Arrow

A cross-language development platform
for in-memory data

Kouhei Sutou

ClearCode Inc.

SciPy Japan Conference 2019
2019-04-23

Me

Ruby committer since 2004

2004年からRubyコミッター

Why do I talk at SciPy?

なぜSciPyで話しているのか？

To introduce
Apache Arrow
Apache Arrowを紹介するため

Apache Arrow

“ A cross-language development
platform for in-memory data
インメモリーデータ向け多言語対応開発プラットフォーム
[cited from `https://arrow.apache.org/'] ”

Cross-language

多言語対応

- ✓ C, C++, C#, Go, Java,
- ✓ JavaScript, MATLAB, **Python**,
- ✓ R, **Ruby** and Rust

Development platform

開発プラットフォーム

Apache Arrow ...

- ✓ specifies standards and
標準化
- ✓ provides implementations
実装

to advance cooperation by many people
多くの人が協力できるように

For in-memory data インメモリーデータ

Apache Arrow focuses on ↓ **for now**

Apache Arrowは**今のところ**は↓に注力

✓ sharing columnar/tensor data

カラムナーデータ・テンソルデータの共有

✓ analyzing columnar data

カラムナーデータの分析

✓ RPC for columnar data

カラムナーデータのRPC

Apache Arrow and Python

Apache ArrowとPython

✓ As pickle replacement

pickleの代替

✓ PySpark does

PySparkはすでにやっている

✓ As dataframe library

データフレームライブラリー

✓ pandas and Vaes use Apache Arrow a bit

pandasとVaesはApache Arrowを少し使っている

Apache Arrow and me

Apache Arrowと私

- ✓ A release manager (リリースマネージャー)
 - ✓ 0.11.0 and 0.13.0 (the latest release/最新リリース)
- ✓ An active developer (アクティブな開発者)



Feature (1)

機能 (1)

Effective serialization

効率的なシリアライズ

Why effective?

なぜ効率的なのか

- ✓ Don't parse data
データをパースしないから
- ✓ Use data directly
データをそのまま使うから

Data format: Number

データフォーマット：数値

Contiguous data (Same as C array)

連続データ (Cの配列と同じ)

32bit integer: [1, 2, 3]

0x01 0x00 0x00 0x00 0x02 0x00 0x00 0x00 0x03 ...

Compare to JSON

JSONと比較

"[1, 2, 3]"



"1" → 1 (String → Number)

"2" → 2 (String → Number)

"3" → 3 (String → Number)

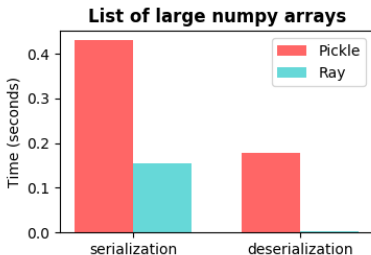
Merit of direct data use

データを直接使うことのメリット

- ✓ Zero copy cost
コピーコストをなくせる
- ✓ Copy is costly for large data
大きなデータではコピーはコストが高い
- ✓ (Nearly) zero parse cost
(ほぼ) パースコストをなくせる
- ✓ Only need to parse metadata
メタデータをパースするだけでよい

Performance

性能



[Fast Python Serialization with Ray and Apache Arrow](#)

Apache License 2.0: (c) 2016-2019 The Apache Software Foundation

Zero copy and large data

ゼロコピーと大きなデータ

- ✓ pandas can't process large data
pandasは大きなデータを扱えない
- ✓ Because it needs to allocate memory
メモリーを確保する必要があるから
- ✓ Apache Arrow supports memory mapping
Apache Arrowはメモリーマッピング対応
- ✓ Can use data in file directly without copy
ファイル内のデータをコピーせずに使える

Effective string representation

効率的な文字列表現

✓ pandas: Array of strings

pandas : 文字列の配列

✓ Use discontinuous memory

非連続なメモリーを使う

✓ Apache Arrow: Data and array of lengths

Apache Arrow : データと長さの配列

✓ Use contiguous memory: Fast

連続したメモリーを使う : 速い

Data format: String

データフォーマット：文字列

Data bytes + length array

UTF-8 string: ["Hello", "", "!"]

Data bytes: "Hello!"

Length array: [0, 5, 5, 6]

i-th length: lengths[i+1] - lengths[i]

i-th data: data[lengths[i]:lengths[i+1]]

Feature(?) (2)

機能 (?) (2)

Specify data format

データフォーマットを仕様化

Why do Arrow specify?

なぜArrowは仕様化するのか

Effective data exchange

効率的なデータ交換のため

Effective data exchange

効率的なデータ交換

- ✓ Use common format widely
みんなが同じフォーマットを使うこと
- ✓ No format conversion reduces resource usage
フォーマットを変換しなくてよいならリソース使用量を減らせる
- ✓ Use low {,de}serialize cost format
シリアライズコストが低いフォーマットを使うこと
- ✓ Fast

Who uses Arrow format?

Arrowフォーマットをだれが使っているか

- ✓ RAPIDS: For NVIDIA GPU
- ✓ Fletcher, InAccel: For FPGA
- ✓ Spark: For interprocess data exchange
Spark: プロセス間のデータ交換のために

CPU and GPU

- ✓ Can't share data on memory
メモリー上のデータを共有できない
- ✓ Need to copy between CPU and GPU
CPUとGPU間でコピーする必要がある
- ✓ Effective data exchange improves performance
データ交換を効率化することで高速化

CPU and FPGA

- ✓ Can't share data on memory
メモリー上のデータを共有できない
- ✓ Need to copy between CPU and FPGA
CPUとFPGA間でコピーする必要がある
- ✓ Effective data exchange improves performance
データ交換を効率化することで高速化

Spark

- ✓ **Process large data**
大きなデータを処理
- ✓ **Need to pass data to worker processes**
ワーカースタンプセスにデータを渡す必要がある
- ✓ **Effective data exchange improves performance**
データ交換を効率化することで高速化

PySpark

- ✓ Worker by Python

ワーカーはPython

- ✓ Use pickle to exchange data

データ交換にpickleを使用

- ✓ Spark supports Arrow for data exchange

Arrowを使ったデータ交換をサポート

- ✓ Disabled by default

デフォルトでは無効

PySpark with Arrow

```
In [2]: %time pdf = df.toPandas()  
CPU times: user 17.4 s, sys: 792 ms, total: 18.1 s  
Wall time: 20.7 s
```

```
In [3]: spark.conf.set("spark.sql.execution.arrow.enabled", "true")
```

```
In [4]: %time pdf = df.toPandas()  
CPU times: user 40 ms, sys: 32 ms, total: 72 ms  
Wall time: 737 ms
```

[Speeding up PySpark with Apache Arrow](#)

Feature (3)

機能 (3)

Optimized data processing modules

最適化されたデータ処理モジュール

Optimized data processing

最適化されたデータ処理モジュール

- ✓ Apache Arrow targets large data
Apache Arrowは大きなデータを対象にしている
- ✓ Performance is important
性能は重要
- ✓ How to get high performance...?
どうすれば速くできる。。。？

High performance (1)

高速化 (1)

Data locality

データを局所化

Data locality

データを局所化

✓ Minimize cache misses

キャッシュミスを減らす

✓ Storage is very slow

ストレージはすごく遅い

✓ Memory is slow

メモリーは遅い

✓ CPU cache is fast

CPUキャッシュは速い

High performance (2)

高速化 (2)

SIMD

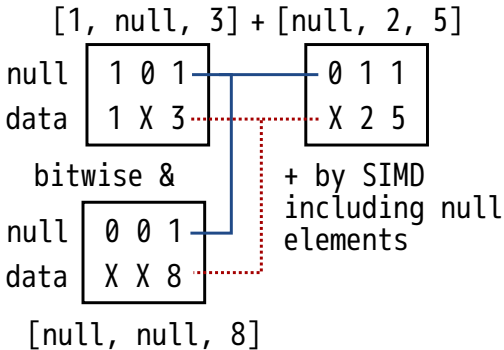
Single Instruction Multi Data
一気に複数のデータを処理する方法

SIMD

- ✓ Data must be contiguous and aligned
データは連続していてアラインされていないといけない
- ✓ Arrow format is SIMD ready
ArrowフォーマットはSIMDを使える
- ✓ No condition branch
条件分岐がないこと
- ✓ Use bitmap instead of "missing" for null
nullを表現するために「欠損値」ではなく別途ビットマップを使う
[Is it time to stop using sentinel values for null / NA values?](#)

No condition branch

条件分岐なし



FYI: null

参考情報: null

- ✓ All data types support null in Arrow
Arrowはすべての型でnullをサポート
- ✓ Some types only support null in NumPy
NumPyは一部の型でnullをサポート
[欠損値の制約 - PythonとApache Arrow](#)

High performance (3)

高速化 (3)

Thread

Thread

- ✓ Use multi-cores in single process
シングルスレッドで複数コアを使う
- ✓ Minimize resource conflict
リソースの競合をなくすこと
- ✓ Locking to avoid conflict reduces performance
競合を避けるためにロックすると性能劣化
- ✓ Approaches (アプローチ)
 - ✓ Read only or copy (shared nothing)
リードオンリーにするかコピー (なにも共有しない)

Apache Arrow and thread

Apache Arrowとスレッド

- ✓ Data is read only
データはリードオンリー
- ✓ Share data in threads without lock overhead
ロックのオーバーヘッドなしでスレッド間でデータを共有
- ✓ Avoid both locking and copying
ロックもコピーも避ける
- ✓ They reduce performance
どちらも性能劣化するから

High performance (4)

高速化 (4)

Compute kernels

計算カーネル

Compute kernels

計算カーネル

- ✓ SIMD ready primitive operations
SIMDを使ったプリミティブな演算
- ✓ Projection, Filter, Aggregation, ...
射影とかフィルターとか集計とかとか
- ✓ compare, take, mean, ...
比較とか行選択とか平均とか

High performance (5)

高速化 (5)

Subgraph compiler

サブグラフコンパイラー

Subgraph compiler: Gandiva

サブグラフコンパイラー：Gandiva

- ✓ Compile operator graphs at run-time
実行時に演算グラフをコンパイル
- ✓ Operator graph: combined multiple operations
演算グラフ：演算のまとめ
- ✓ `table.a + table.b < table.c && ...`
- ✓ Usable for query engine backend
クエリーエンジンのバックエンドとして使える

High performance (6)

高速化 (6)

Query engine

クエリーエンジン

Query engine

クエリーエンジン

✓ For single node
シングルノード向け

✓ Dataflow-style operator execution
データが流れるように演算を実行

✓ scan → project → filter → aggregate → ...
データ取得→射影→フィルター→集計→...

[Apache Arrow Query Engine for C++](#)

Query engine from Python

Pythonからクエリーエンジンを使う

✓ With pandas (pandasを使う)

- ✓ Large data → execute → to_pandas()
大きなデータ→実行→to_pandas()

✓ With Dask (Daskを使う)

- ✓ Dask will be able to use this as backend
Daskのバックエンドで使えるかも？

High performance (7)

高速化 (7)

Datasets

データセット

Datasets

データセット

- ✓ Scan data from storage/database
ストレージ・データベースからデータ取得
 - ✓ File systems: local, HDFS, ...
 - ✓ Formats: CSV, Parquet, ...
 - ✓ Databases: MySQL, PostgreSQL, ...

[Apache Arrow C++ Datasets](#)

Fast datasets

高速なデータセット

- ✓ Predicate pushdown
条件のプッシュダウン
- ✓ Scan only needed data
必要なデータのみ取得
- ✓ Parallel scan
並列取得

Feature (4)

機能 (4)

RPC

RPC: Arrow Flight

- ✓ Fast RPC framework for Arrow
Arrow用の高速なRPC
- ✓ Based on gRPC with low-level extensions
gRPCベースでいくつか低レベルの拡張をしている

[Apache 0.11.0 Release](#)

Wrap up

まとめ

- ✓ Arrow is useful for SciPy community
SciPyコミュニティにArrowは有用
- ✓ in not only Python but also other languages
Pythonだけでなく他の言語でも有用
- ✓ Join Apache Arrow development!
Apache Arrowの開発に参加しよう！
- ✓ Ask me how to start
なにから始めればよいかは私に相談してね

Next step

次の一歩

- ✓ [Mailing list](mailto:dev@arrow.apache.org): dev@arrow.apache.org
- ✓ Chat in Japanese:
 - ✓ <https://gitter.im/apache-arrow-ja/community>
- ✓ Apache Arrow Tokyo Meetup 2019 this summer?
 - ✓ See also: [Apache Arrow Tokyo Meetup 2018](#)