Apache Arrow 1.0

A cross-language development platform for in-memory data

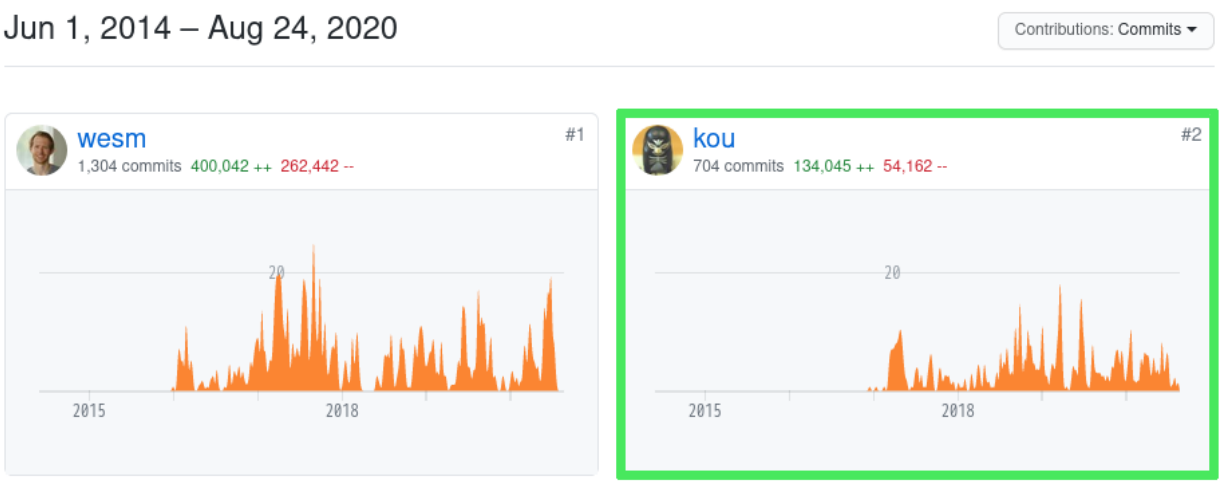
Sutou Kouhei

ClearCode Inc.

SciPy Japan 2020
2020-10-30

Me

- ✓ Apache Arrow PMC member
- ✓ #2 Apache Arrow contributor



Apache Arrow and You

Your data size is ...
データサイズが…

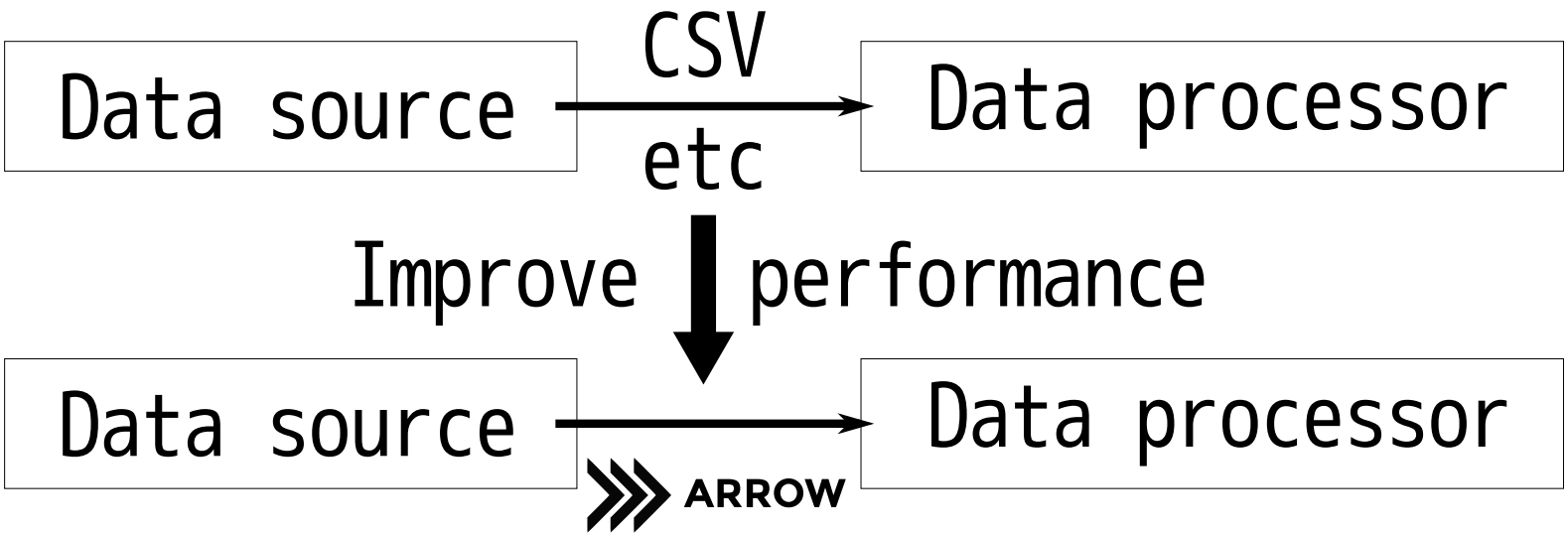
- ✓ small:
  小さい
  - ✓ You don't need Apache Arrow
    Apache Arrowを使う必要はない
- ✓ large:
  大きい
  - ✓ Apache Arrow will help you!
    Apache Arrowを使うといいよ！

How does Apache Arrow help you?

どうしてApache Arrowを使うといいの？

Improves **data interchange** performance

データ交換のパフォーマンス改善



Why does Apache Arrow focus on data interchange?

どうしてApache Arrowはデータ交換に注力しているの？

Because it's a bottleneck

ボトルネックだから

...Really?

…本当？

Large data transfer case
大規模データを転送するケース

Don't Hold My Data Hostage –
A Case For Client Protocol Redesign

Mark Raasveldt
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
m.raasveldt@cwi.nl

Hannes Mühleisen
Centrum Wiskunde & Informatica
Amsterdam, The Netherlands
hannes@cwi.nl

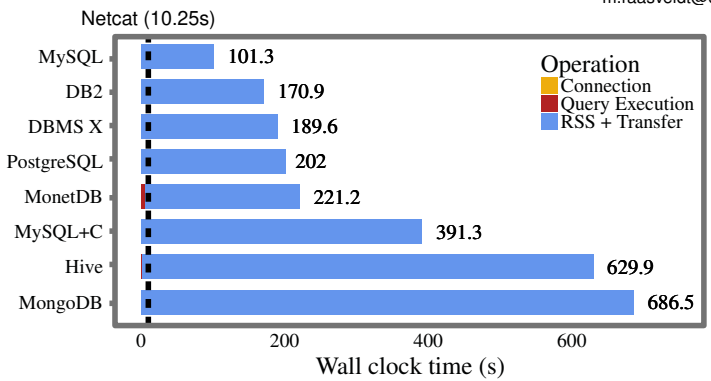


Figure 1: Wall clock time for retrieving the `lineitem` table (SF10) over a loopback connection. The dashed line is the wall clock time for netcat to transfer a CSV of the data.

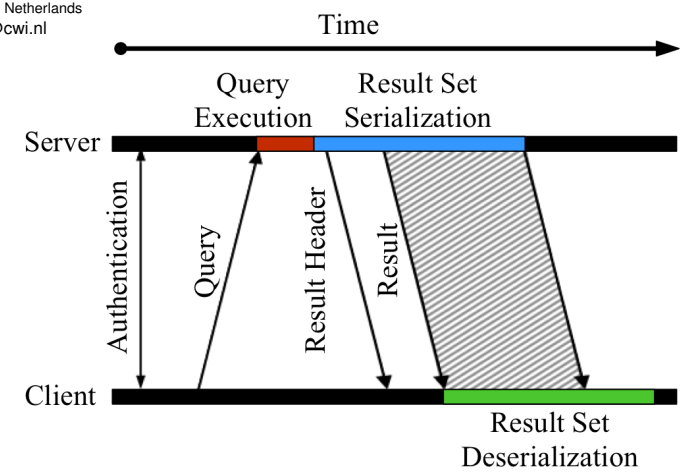
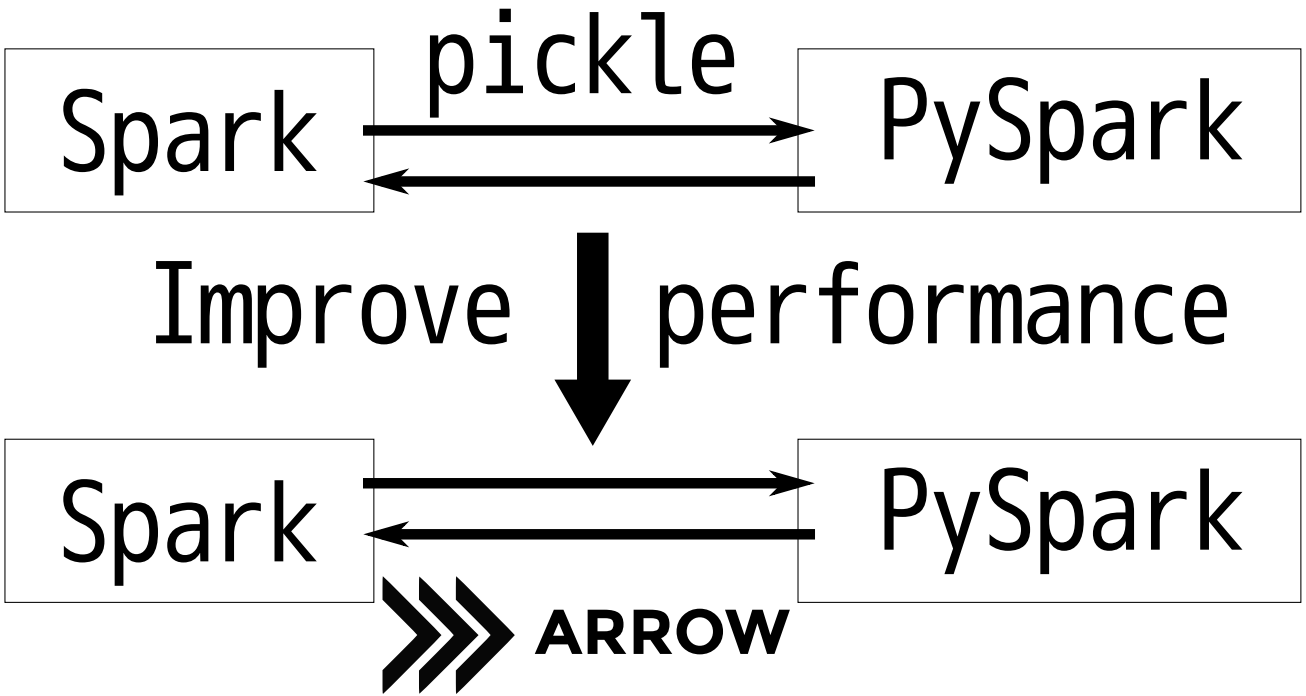
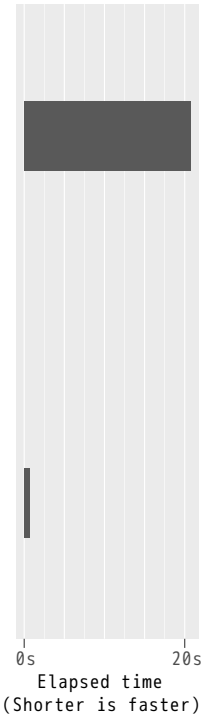


Figure 2: Communication between a client and a server

https://hannes.muehleisen.org/p852-muehleisen.pdf

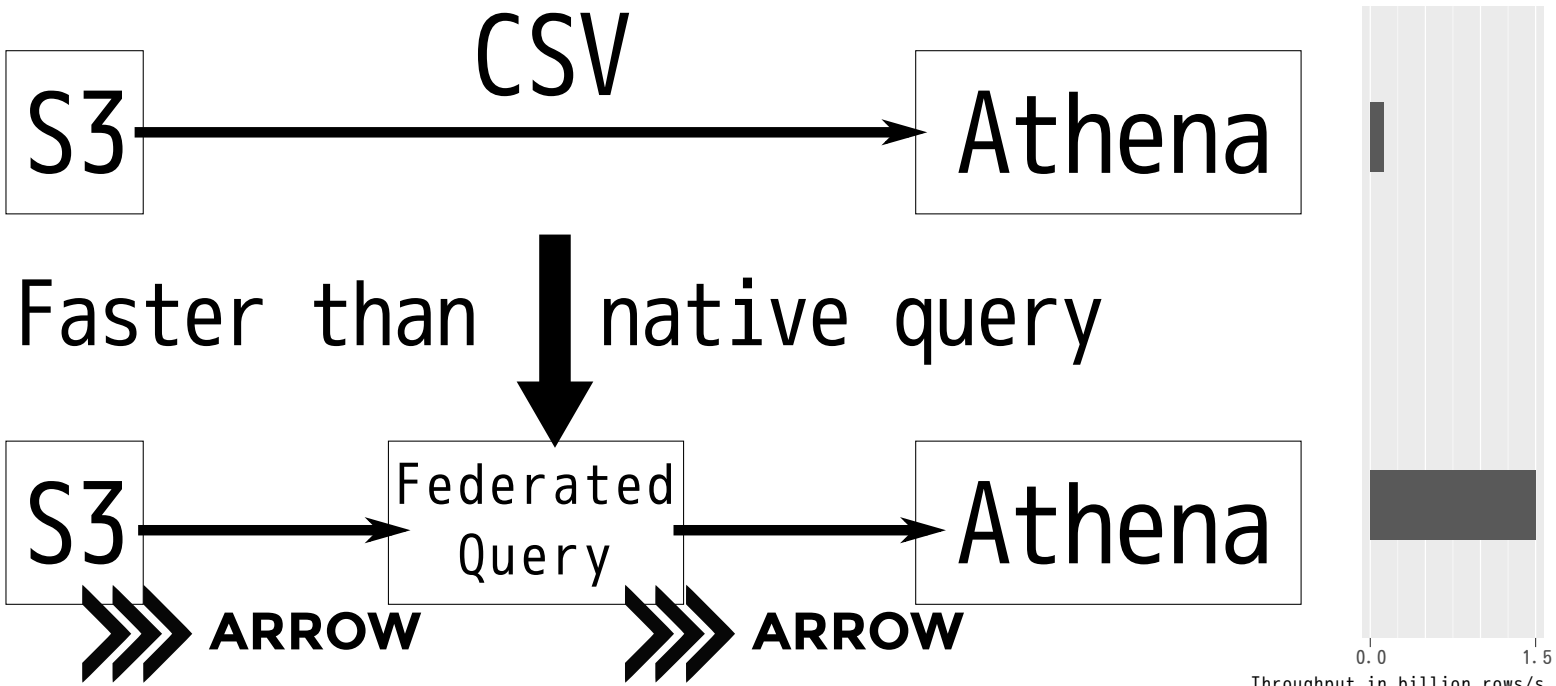
Use case: Apache Spark





Data at https://arrow.apache.org/blog/2017/07/26/spark-arrow/

Use case: Amazon Athena



Data at https://github.com/awslabs/aws-athena-query-federation/tree/master/athena-federation-sdk#performance

Why is Apache Arrow fast?

どうしてApache Arrowは速いの？

ClearCode

Apache Arrow

A cross-language development platform
for in-memory data

Kouhei Sutou

ClearCode Inc.

SciPy Japan Conference 2019
2019-04-23

https://slide.rabbit-shocker.org/authors/kou/scipy-japan-2019/

Wow! Awesome!

すごいじゃん！

Should I always use Apache Arrow?

いつもApache Arrowを使うべき？

...No

…いやいや

Apache Arrow is designed for
Apache Arrowの設計方針

- ✓ In-memory analytics
  メモリー内での分析
- ✓ Cross-language
  - ✓ Can be used from all languages
    プログラミング言語に依存せずに使えること
- ✓ Large data
  大規模データ
- ✓ Fast data transfer
  高速なデータ転送

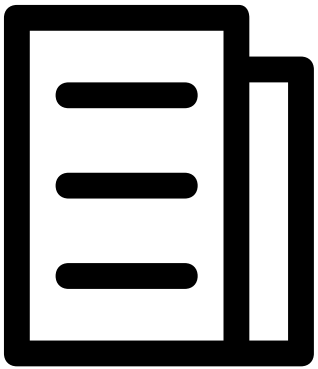
In-memory analytics
メモリー内での分析

- ✓ For in-memory: not for storage
  メモリー内用：保管用ではない
  - ✓ For storage: Apache Parquet, ...
    保管用にはApache Parquetなどいろいろある
- ✓ For analytics: not for transaction
  分析用：トランザクション用ではない
  - ✓ Not for OLTP use
    OLTP用ではない

In-memory（メモリー内）

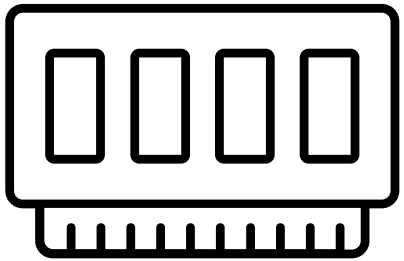
- ✓ Apache Arrow format means:
  Apache Arrowフォーマットは↓ということ
  - ✓ File format: *.arrow files（ファイルフォーマット）
  - ✓ Memory layout（メモリー上のデータの配置方法）
- ✓ With Apache Parquet (for storage):
  （保管用の）Apache Parquetと一緒に使う場合：
  - ✓ Reading Apache Parquet data (file format)
    Apache Parquetデータ（ファイルフォーマット）を読む
  - ✓ as Apache Arrow data (memory layout)
    読んだデータはApache Arrowデータ（メモリー上の配置）にする

Apache Parquet and Apache Arrow



Created by Muhammad Yafinuha
from the Noun Project

Read



Created by Philip Sheffield
from the Noun Project

Parquet

ARROW

Analytics
分析

- ✓ Column based operations are important
  カラム単位の操作が重要
  - ✓ Aggregate, sort, filter, ...
    集約、ソート、絞り込み…
- ✓ For fast column based operations
  高速なカラム単位の操作に必要なもの
  - ✓ Columnar layout（カラムごとにまとまった配置）
  - ✓ Contiguous data for vectorization
    ベクトル化用に連続して配置したデータ

Columnar layout
カラムごとにまとまった配置

- ✓ Type per column
  カラムごとに型がある
- ✓ Data frame is suitable
  データフレームは適切
- ✓ Mixed type column data aren't suitable
  カラム内で型が混在しているデータは不向き
  - ✓ Like data in MongoDB (MongoDBに入っているようなデータ)
  - ✓ Union can be used for it but ...
    unionを使って表現できなくはないだろうけど…

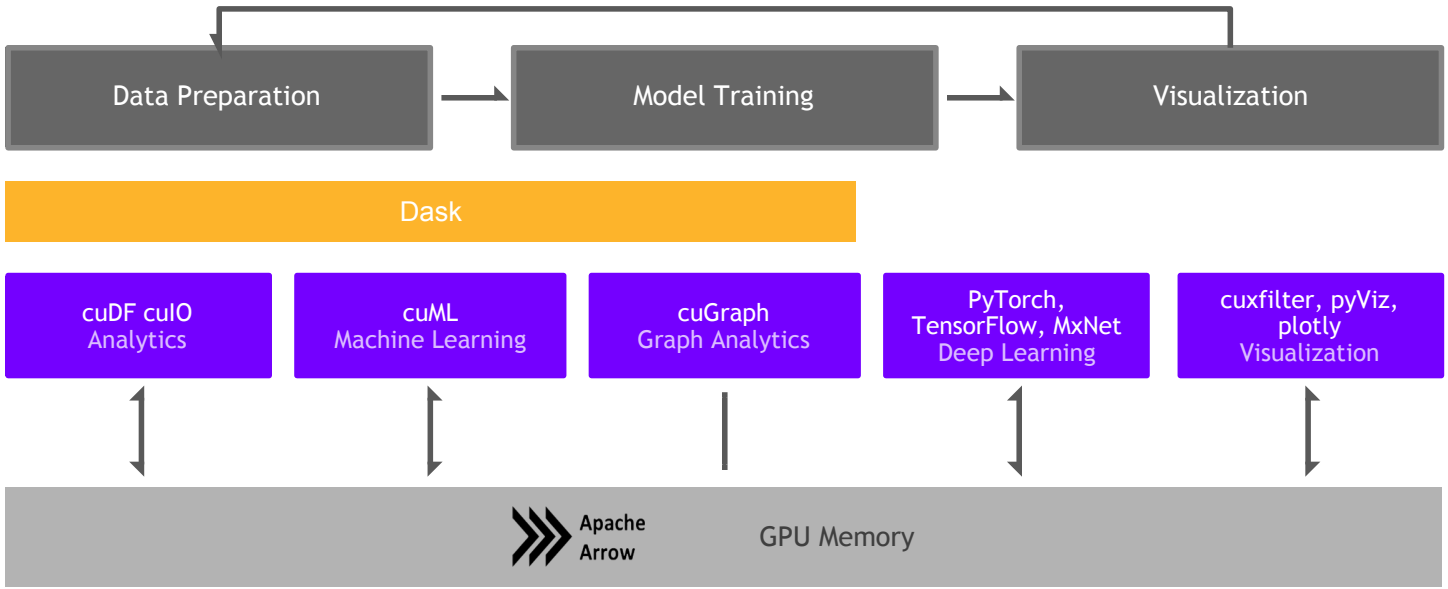
Cross-language

- ✓ All your modules use only NumPy
  すべてのモジュールでNumPyしか使っていない場合
  - ✓ NumPy format is enough（NumPyフォーマットで十分）
- ✓ Your system uses multiple languages
  複数のプログラミング言語を使っている場合
  - ✓ Apache Arrow will be useful（Apache Arrowの出番）
  - ✓ Apache Spark, Python and R, ...

Use case: RAPIDS

RAPIDS

End-to-End GPU Accelerated Data Science

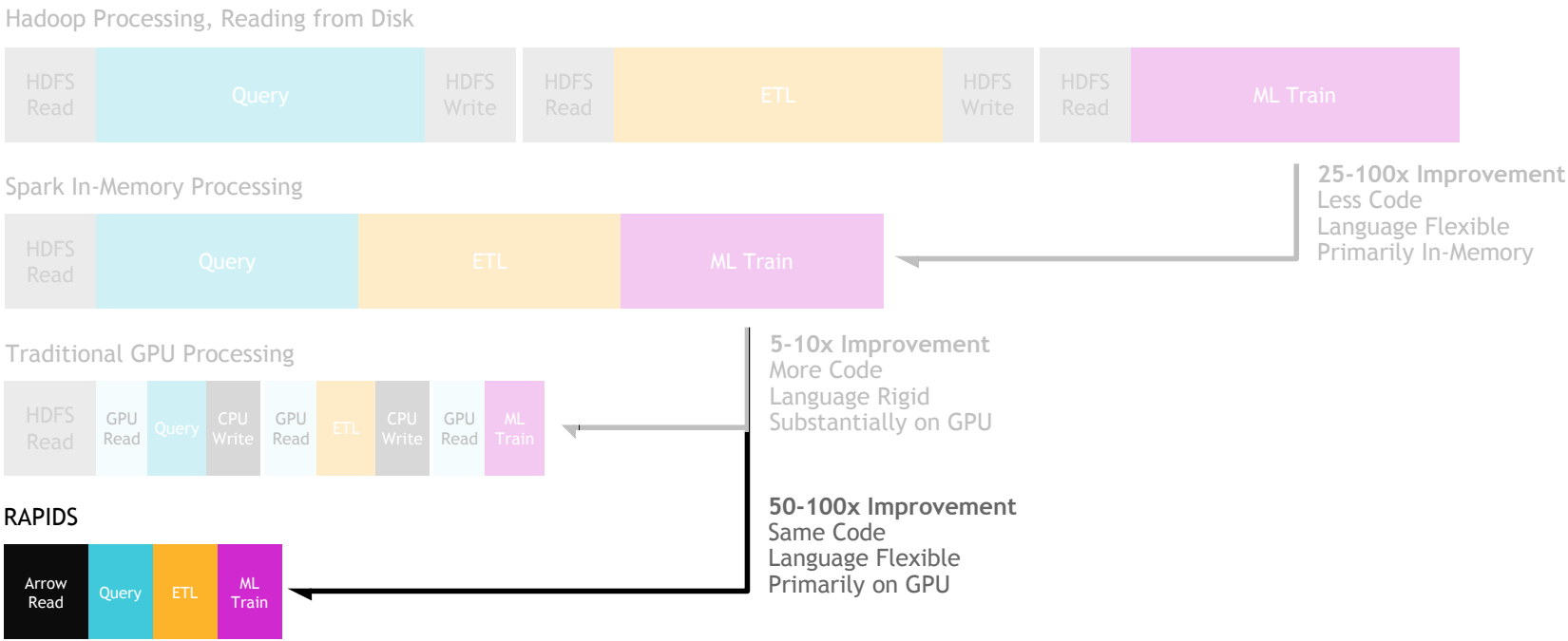


https://docs.rapids.ai/overview/RAPIDS%200.15%20Release%20Deck.pdf#page=3

Use case: RAPIDS

Data Processing Evolution

Faster Data Access, Less Data Movement



https://docs.rapids.ai/overview/RAPIDS%200.15%20Release%20Deck.pdf#page=8

Data size
データサイズ

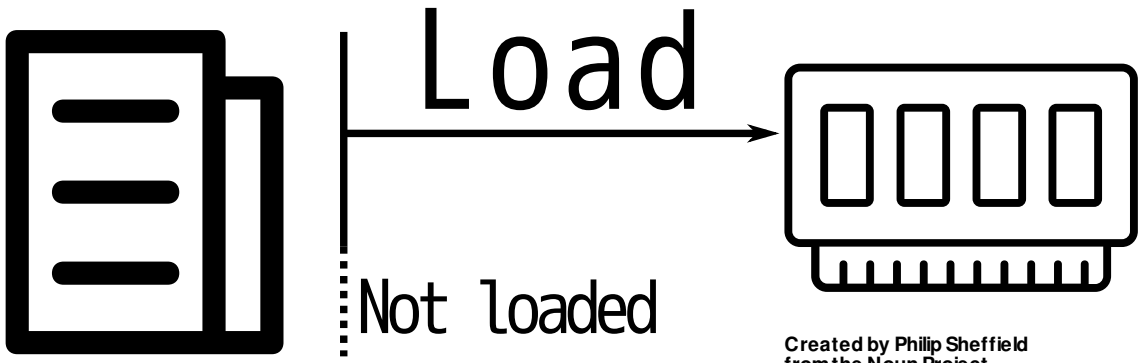
- If you have only small data
  小さなデータの場合
  - Any format will work well
    なにを使ってもいい
- We need something to process large data
  大規模データを処理するには工夫が必要

Large data with Apache Arrow

Apache Arrowで大規模データ

- ✓ Memory mapping（メモリーマップ）
  - ✓ can process large data without loading all data on memory at once
    大規模データを一度にすべてメモリー上に載せなくても処理できる
- ✓ Record batch（レコードバッチ）
  - ✓ can process large data with batches
    大規模データをバッチに分割して処理できる
  - ✓ can process large data as stream
    大規模データをストリームとして処理できる

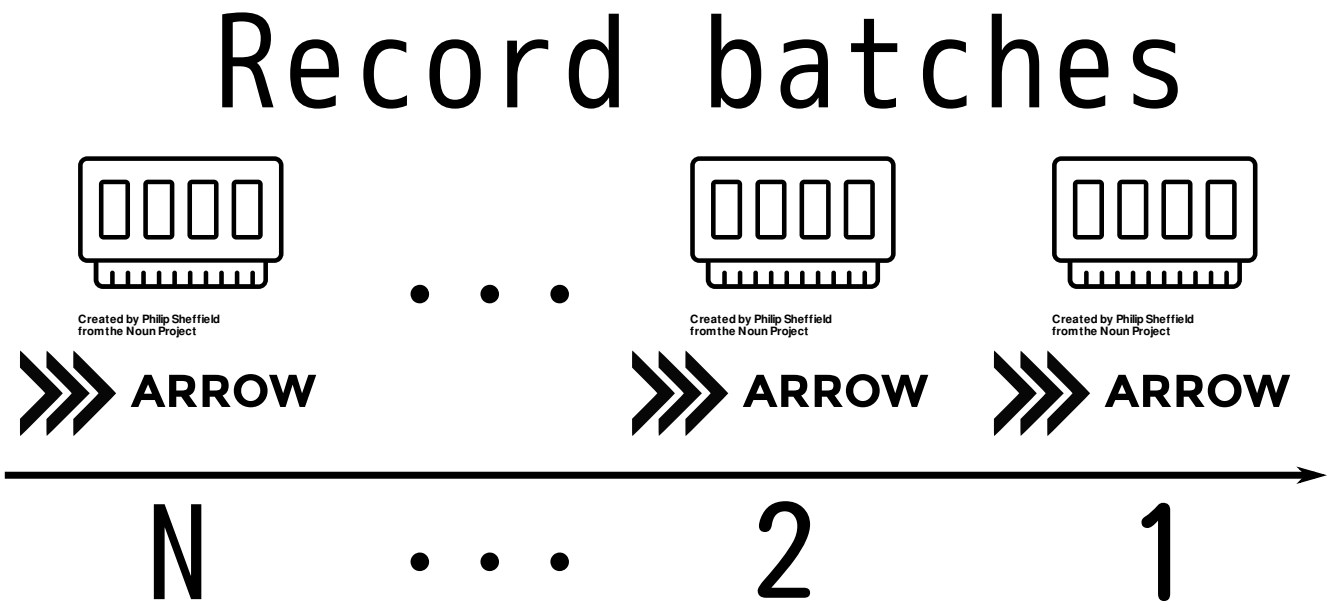
Memory mapping
メモリーマップ



Created by Muhammad Yafinuha
from the Noun Project

Created by Philip Sheffield
from the Noun Project

Record batch
レコードバッチ



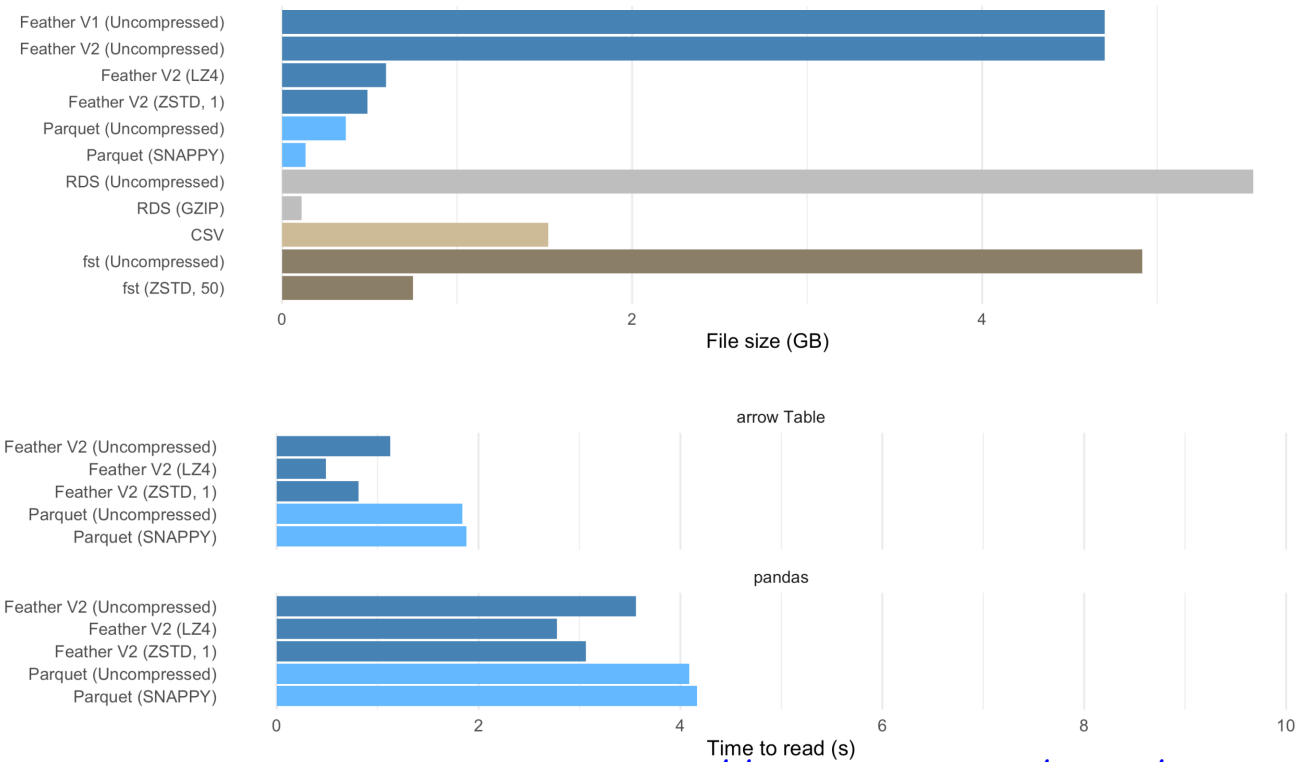
Fast data transfer
高速データ転送

- ✓ Enough network bandwidth case:
  十分なネットワーク帯域がある場合
  - ✓ Fast {,de}serialize improves performance
    高速なシリアライズ機能で高速転送が可能
- ✓ Network bottleneck case:
  ネットワーク帯域がボトルネックになるケース
  - ✓ Other small size format may be better
    よりサイズが小さくなる別のフォーマットの方がいいかも

Network bottleneck
ネットワーク帯域がボトルネック

- ✓ The Amazon Athena use case:
  前述のAmazon Athenaの事例
  - ✓ Apache Parquet is faster than Apache Arrow
    Apache ArrowよりApache Parquetの方が高速
  - ✓ Because run-length encoding works well
    ランレングス圧縮がすごく効いていたから
- ✓ Apache Arrow supports compression
  Apache Arrowは圧縮をサポートしている
  - ✓ Maybe, the Amazon Athena use case doesn't use it
    たぶん前述のAmazon Athena事例では使っていない

Apache Arrow and compression



https://ursalabs.org/blog/2020-feather-v2/

When should I use Apache Arrow
いつApache Arrowを使えばよいか

- ✓ In-memory analytics
  メモリー内での分析
  - ✓ Not for storage（保管用ではない）
  - ✓ Not for transaction（トランザクション用ではない）
- ✓ Columnar layout data
  カラムごとにまとまったデータ
  - ✓ Not for data in document DB
    ドキュメント指向DBに入っているようなデータは向いていない

When should I use Apache Arrow
いつApache Arrowを使えばよいか

- ✓ Cross-language system
  複数のプログラミング言語を使うシステム
  - ✓ Many data analytics system use multiple languages
    多くのデータ分析システムは複数のプログラミング言語を使っている
- ✓ Large data（大規模データ）
- ✓ Fast data transfer（高速データ転送）
  - ✓ Compression may resolve network bottleneck
    ネットワーク帯域がボトルネックになるときは圧縮するといいかも

Current & Future

Apache Arrow 1.0.0

2020-07-24

What does 1.0.0 mean?

1.0.0になったということはどういうこと？

Start following Semantic Versioning

セマンティックバージョン開始

Semantic Versioning
セマンティックバージョン

- ✓ Backward compatible（後方互換）
  - ✓ In the same major version（メジャーバージョンが同じ間）
  - ✓ Newer can be used with older
    新しいバージョンは古いバージョンと一緒に使える
- ✓ Forward compatible（前方互換）
  - ✓ In the same major.minor version
    メジャー・マイナーバージョンが同じ間
  - ✓ Older can be used with newer w/o new features
    古いバージョンは新機能を使わないなら新しいバージョンと一緒に使える

Versions in Apache Arrow
Apache Arrowのバージョン

- ✓ Apache Arrow format（Apache Arrowフォーマット）
  - ✓ File format（ファイルフォーマット）
  - ✓ Memory layout（メモリー上でのデータの配置方法）
- ✓ Apache Arrow libraries such pyarrow
  pyarrowのようなApache Arrowライブラリー
  - ✓ Read/write Apache Arrow file format
    Apache Arrowファイルフォーマットを読み書き
  - ✓ Process Apache Arrow memory layout data
    メモリー内のApache Arrowデータの処理

Release cycle
リリースサイクル

Per 3-4 months
3-4ヶ月ごと

Release and version
リリースとバージョン

- Apache Arrow format's version
  - It's bumped only when needed
    必要なときだけバージョンアップ
- Apache Arrow libraries' version
  - Major version is always bumped
    常にメジャーバージョンアップ
  - 2020-07: 1.0.0
  - 2020-10: 2.0.0

Major version up
メジャーバージョンアップ

It may break compatibility
互換性がなくなるかもしれない

You should pin pyarrow version

pyarrowのバージョンは固定するべき

requirements.txt:

```
pyarrow~=1.0.*
```

You may not use pyarrow directly
直接pyarrowを使わないかもしれない

- ✓ Your dependencies may use pyarrow internally instead
  依存しているモジュールが内部で使っているかも
  - ✓ pandas uses pyarrow to read Apache Parquet
    pandasはApache Parquetを読み込むためにpyarrowを使っている
  - ✓ PySpark uses pyarrow to communicate Spark
    PySparkはSparkと通信するためにpyarrowを使っている
- ✓ Modules use pyarrow will be increased
  pyarrowを使うモジュールは増えていくはず

Ecosystem
エコシステム

- The more many modules support Arrow, the more we can improve performance
  Apache Arrow対応モジュールが増えるほど性能アップ
  - Because we don't need to convert to Arrow
    Apache Arrowフォーマットに変換する必要がなくなるから
- Many modules start supporting Arrow!
  多くのモジュールがApache Arrowの対応を始めている！
  - Because 1.0.0 was released
    1.0.0がリリースされたから

BigQuery Storage API

```
from google.cloud import bigquery_storage_v1




client = bigquery_storage_v1.BigQueryReadClient()
...
stream = client.read_rows(...)
arrow_table = stream.to_arrow(...)
```

See also: https://medium.com/google-cloud/announcing-google-cloud-bigquery-version-1-17-0-1fc428512171

pandas, Dask and Vaex

- ✓ Use pyarrow to RW Apache Parquet
  Apache Parquetフォーマットを読み書きするためにpyarrowを使用
- ✓ Not use Apache Arrow for internal memory layout
  内部のデータの持ち方はApache Arrowではない
  - ✓ cuDF does but not use pyarrow to process data
    cuDFはApache Arrowデータとして持っているがpyarrowでは処理していない

Current pyarrow use cases
現時点でのpyarrowの利用例

- ✓ Apache Parquet reader/writer
- ✓ Apache Arrow data representation
  Apache Arrowデータの表現
- ✓ Not used for internal memory layout
  内部のデータ表現としては使われていない
  - ✓ NumPy's memory layout is used for numeric
  - ✓ Apache Arrow's memory layout is compatible with NumPy's one
  - ✓ Why?（どうして？）

Apache Arrow native data processing

Apache Arrowデータの処理

Work In Progress

実装中

Data processing
データ処理

- ✓ Current: (現在)
  - ✓ Gandiva and compute functions
    https://arrow.apache.org/docs/cpp/api/compute.html
    Gandivaと計算関数がある
  - ✓ Missing some features such as sort related
    ソート関連などまだ機能が足りない
- ✓ Future: (将来)
  - ✓ Apache Arrow native data frame
    Apache Arrowデータ用データフレーム

Apache Arrow native data frame
Apache Arrowデータ用データフレーム

- File system module（ファイルシステムモジュール）
  - Read/write files with the same API
    同じAPIでファイル操作
  - Current: Local, HDFS and S3
- Dataset module（データセットモジュール）
  - Read/write semantic datasets stored in different locations and formats
    様々な場所・フォーマットのデータを論理的なデータセットとして読み書き
  - Current: Apache Arrow, Apache Parquet and CSV

Apache Arrow native data frame
Apache Arrowデータ用データフレーム

- ✓ Query engine module (クエリーエンジンモジュール)
  - ✓ Process query against dataset
    データセットに対してクエリー実行
  - ✓ Current: Not started (現在：未着手)
- ✓ Data frame module (データフレームモジュール)
  - ✓ Provide higher level API for these modules
    これらのモジュールを使う高レベルのAPI
  - ✓ Current: Not started (現在：未着手)

Wrap up
まとめ

- ✓ Apache Arrow 1.0.0 was released!
  Apache Arrow 1.0.0がリリースされた！
  - ✓ Semantic versioning（セマンティックバージョン）
- ✓ You will use Apache Arrow implicitly
  知らないうちにApache Arrowを使うことが増えるはず
- ✓ You can't use pyarrow for data processing yet
  pyarrowでデータ処理をするにはまだ機能不足
  - ✓ But I want to use ASAP!
    でも、すぐに使いたい！

Join us!
おいでよ！

- The Apache Arrow project welcomes all!
  https://arrow.apache.org/community/
  Apache Arrowプロジェクトは歓迎するよ！
- You can get features what you want quickly if you join developing them
  開発に参加すると欲しい機能がより早く手に入るよ！
  - Community in Japanese: Red Data Tools
    https://gitter.im/red-data-tools/ja
    日本語で相談したい人はRed Data Toolsへ
  - ClearCode Inc. supports it:
    https://www.clear-code.com/services/apache-arrow.html
    クリアコードは商用サポートを提供しているよ