Recueil du Symposium 2018 de Statistique Canada
Combiner pour vaincre : innovations dans l'utilisation de multiples sources de données

Normes de données ouvertes pour les données administratives

Ryan Mackenzie White

Résumé

Le cadre de sciences de données Artemis de Statistique Canada est un cadre basé sur le traitement de données par lots et la norme Apache Arrow de format des données ouvertes, pour la production de données administratives de haute qualité à des fins d'analyse. Plusieurs organisations statistiques sont en train de s'orienter vers une approche qui priorise l'usage des données administratives pour la production de statistiques officielles. La production de données administratives de haute qualité et adaptées à leur utilisation doit préserver les données brutes durant le cycle de vie des données (ingestion, intégration, gestion, traitement et analyse). Les formats et cadres de production de données doivent appuyer des changements dans la charge de travail analytique, qui utilise les données d'une façon différente des enquêtes traditionnelles, et qui nécessite une utilisation itérative efficiente des données à chaque étape du cycle de vie des données et des outils statistiques pour l'évaluation continue de la qualité des données et leur adaptation à l'usage prévu. Essentiellement, le prototype de cadre de sciences des données Artemis de Statistique Canada repose sur un format de données bien défini et indépendant du langage utilisé, qui accélère les traitements de données à des fins d'analyse grâce à une architecture informatique moderne.

Mots-clés: données administratives, logiciel libre, formats de données

1. Introduction

1.1 Description

En matière de données, toute initiative repose sur un modèle de données cohérent et bien défini. Le modèle de données appuie la stratégie de gestion des données sur le long terme et informe les exigences en matière de puissance informatique, de stockage et d'analyse de données. Le choix du modèle de données doit refléter les types d'analyse de l'utilisateur, et appuyer les charges de travail analytique pour la production des données.

Le cadre de traitement de données Artemis de Statistique Canada démontre l'usage du format de données Apache Arrow (Arrow 2017) pour le traitement en mémoire de données en colonnes et des techniques informatiques modernes pour l'ingestion, la gestion et l'analyse de données provenant d'ensembles de données très volumineux. Le cadre permet une charge de travail analytique à partir de sources de données administratives qui nécessite une écriture et plusieurs lectures, et des requêtes analytiques basées sur des façons communes d'accéder à des données tabulaires. Le cadre repose sur le traitement continu par lots d'enregistrements, inspiré du cadre de traitement de données par événement qui est utilisé dans la physique des particules à haute énergie (Berger, et al. 2015), pour l'usage efficient des ressources informatiques avec des économies d'échelle horizontales et verticales.

Le prototype Artemis de Statistique Canada s'appuie sur Apache Arrow dès le départ, en exploitant les mémoires tampons Arrow pour produire des ensembles de données de haute qualité. Le prototype réalise ceci en offrant un cadre pour effectuer des opérations sur les tables Arrow (un moteur pour l'exécution) à partir d'algorithmes et d'outils fournis par l'utilisateur. La fonctionnalité principale est un contrôle programmable du traitement pour produire des ensembles de données comprenant une ou plusieurs tables Arrow. Le format de données Arrow est indépendant du langage utilisé et permet de partager les données entre les processus ou les librairies, en mémoire, sans copies et sans sérialisation.

Les principaux objectifs du prototype du cadre de travail sont:

1. Produire des jeux de données avec un seul format cohérent qui permet des interactions efficientes avec des jeux de données volumineux, dans un environnement multicoeur sur une seule machine.

2. Appuyer des analyses en continu de lots d'enregistrements, qui peuvent ne pas résider entièrement en mémoire.
3. Exécuter des processus opérationnels complexes sur des lots d'enregistrements pour transformer les données.
4. Incorporer des outils de qualité des données et d'adaptation à l'utilisation comme parties à part entière du processus de production des données.

1.2 Exigences générales pour le traitement des données

Les cadres de conception et de production des données, qui supportent les besoins des utilisateurs, doivent mettre l'accent sur quatre fonctionnalités essentielles.

Performance – L'indicateur de performance habituel est le temps de rotation requis pour effectuer tous les traitements lorsque de nouvelles exigences ou de nouvelles données sont soumises. Le temps d'exécution est limité par le transfert des données, le chargement des données entre les étapes successives (ou flux successifs), et la conversion entre les différents formats à utiliser à travers des systèmes fragmentés. Le logiciel doit minimiser le nombre d'opérations de lecture ou d'écriture sur les données et faire le plus de traitement possible en mémoire. Les différentes étapes doivent être modulaires, de sorte qu'il soit possible de les exécuter à nouveau rapidement en cas de changements.

Facilité d'entretien – Un logiciel modulaire avec une séparation claire entre le code algorithmique et la configuration facilite l'introduction de nouveau code, qui peut être intégré sans perturber la configuration des processus existants. Un code modulaire appuie aussi la structure du code et encourage la réutilisation. Les formats de données communs séparent les entrées-sorties et la désérialisation du code algorithmique, et ils offrent des recettes pour le traitement ainsi qu'une structure de code prête à l'emploi pour introduire de nouveaux processus dans le système.

Fiabilité – Le système est conçu à partir de librairies de logiciel libre bien entretenues, avec des fonctionnalités pour introduire facilement des librairies existantes de procédures d'analyse courantes.

Flexibilité – La réutilisation de processus courants est facilitée par du code algorithmique programmable. L'usage d'un format de données commun pour le traitement en mémoire simplifie l'introduction de nouvelles fonctionnalités, de variables et de structures de données dans les jeux de données.

2. Norme Apache Arrow pour les données en colonnes

2.1 Normes pour les données ouvertes

Des normes ouvertes permettent aux systèmes, aux processus et aux librairies de communiquer entre eux. Une communication directe basée sur des protocoles et des formats de données respectant les normes simplifie l'architecture du système, réduit la fragmentation de l'écosystème, améliore l'interopérabilité entre les processus, et élimine la dépendance aux systèmes exclusifs. Plus important, des formats de données communs facilitent la réutilisation du code, le partage, la collaboration efficace et l'échange de données, avec pour résultat des algorithmes et des librairies utilisés par une grande communauté ouverte. Un format de données commun, qui définit les types primitifs de données observés en science des données, dans les sciences sociales et dans les entreprises, garantira que l'état brut des données est préservé lorsque les données sont consommées par des organisations. L'organisation de données tabulaires sous forme de colonnes en mémoire permet aux applications d'éviter les entrées et sorties superflues et accélère le traitement des analyses sur les unités centrales de traitement (CPUs, Central Processing Units) et les processeurs graphiques (GPUs, Graphical Processing Units) modernes.

Les communautés de science des données et des sciences sociales travaillent souvent avec des données tabulaires, qui sont sous diverses formes, le plus couramment appelées trames de données (*DataFrames*). Le concept de trame de données et la sémantique utilisés dans divers systèmes sont communs aux divers trames de données. Toutefois, la représentation sous-jacente sous forme d'octets varie d'un système à l'autre. La différence de représentation en mémoire empêche le partage de code algorithmique à travers divers systèmes et langages de programmation. Aucune norme n'existe pour les données tabulaires en mémoire, toutefois, les données tabulaires sont omniprésentes. Les données tabulaires sont courantes en SQL (Structured Query Language) et dans les systèmes Spark et Hive développés

par la communauté des données massives, et les trames de données en mémoire se retrouvent dans plusieurs langages de science de données. R, Python et Julia supportent tous les données tabulaires basées sur des trames de données en mémoire, et sont couramment utilisés par les analystes.

2.2 Principaux avantages de Apache Arrow

Figure 2.2-1
Arrow en mémoire

Le projet Apache Arrow (Arrow, 2017) résout le problème de trames de données non-portable en offrant une plateforme de développement interlangage pour les données en mémoire, qui spécifie une norme de format de mémoire en colonnes indépendant du langage pour les données unidimensionnelles et hiérarchiques, organisée pour des opérations analytiques efficientes sur de l'équipement informatique moderne. Arrow fournit des librairies informatiques, une messagerie en continu zéro-copie et de la communication interprocessus. Arrow est un interface de données commun aux processus locaux et éloignés.

L'objectif d'Apache Arrow est de fournir une plateforme de développement pour des systèmes de sciences des données, qui découple l'intégration verticale des composantes du traitement des données : sérialisation-désérialisation et entrées-sorties performantes, stockage en mémoire conforme aux normes, et moteur de calcul intégré. Apache Arrow déconstruit l'architecture de données classique sous forme de pile qui est intégrée verticalement, fournissant des interfaces de programmation des applications (API, Application Programming Interface) publics pour chaque composante.

Rapide – Il permet aux moteurs d'exécution de tirer partie des dernières opérations du type entrée unique pour plusieurs sorties (SIMD, Single input multiple data) sur des processeurs modernes, pour l'optimisation vectorisée du traitement analytique des données. Le format en colonnes est optimisé pour des données stockées localement pour une meilleure performance. Le format Arrow permet des lectures zéro-copie pour un accès rapide aux données sans sérialisation.

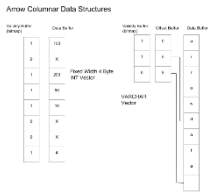
Flexible – Arrow se comporte comme une interface de haute performance entre divers systèmes et permet une grande variété de langages propres à certains domaines, incluant des implémentations natives en C++, Java, Javascript, Rust et Go, avec des liens basés sur des librairies C++ pour Python, Ruby et R.

Norme – Apache Arrow est appuyé par des développeurs clés de 13 projets majeurs de logiciel libre, des scientifiques travaillant à la frontière de la science des données, de l'apprentissage profond, des développeurs de logiciels basés sur des GPUs, incluant Calcite, Cassandra, Drill, Hadoop, Hbase, Ibis, Impala, Kudu, Pandas, Parquet, Phoenix, Spark, Storm et CERN, devenant de ce fait une norme pour l'analytique en mémoire de données en colonnes.

2.3 Format de données Apache Arrow

Le format de données complet Apache Arrow définit des types primitifs de données pour les scalaires de longueur fixe ou variable aussi bien que des types complexes tels que des unions (denses ou éparses), des structures et des listes. La longueur variable des données permet les formats UTF8 ou varchar ainsi que varbinary. Les types complexes permettent des données hiérarchiques imbriquées, par exemple, pour représenter des données JSON imbriquées.

Figure 2.3-1
Tampons Arrow



- Types primitifs de longueur fixe : nombres, booléens, date et temps, binaire de longueur fixe, décimaux, et d'autres valeurs qui peuvent s'écrire à l'aide d'un nombre donné
- Types primitifs de longueur variable : binaire, chaîne de caractères
- Types imbriqués : liste, structure et union
- Type dictionnaire : un type catégoriel codé

Le format Arrow orienté en colonnes et en mémoire permet la sérialisation-désérialisation et appuie la persistance de divers systèmes dorsaux et formats orientés en colonne. Le choix d'un format orienté en colonnes est basé sur les avantages obtenus du point de vue de la performance. Quelques façons courantes d'accéder aux données qui bénéficient de l'accès aux données orienté en colonnes sont l'accès aux éléments de colonnes adjacentes de façon consécutive et l'accès à des colonnes spécifiques. Les données en colonnes permettent les algorithmes basées sur les SIMD, les algorithmes vectorisés, et la compression en colonnes.

3. Le cadre de science des données Artemis

3.1 Aperçu

Le cadre de travail prototype Artemis de Statistique Canada s'appuie sur les fonctionnalités de la plateforme de développement Apache Arrow et met l'emphase sur le traitement et l'analyse des données de façon collaborative et reproductible. L'API de Arrow, qui ne dépend pas du serveur frontal, nous permet de définir un modèle de données pour gérer le partage des données tabulaires à travers des séquences d'algorithmes, qui décrivent divers (parfois disparates) processus opérationnels dans une seule tâche de traitement de données en mémoire. Les algorithmes décrivent divers processus opérationnels pour le même jeu de données, et les algorithmes peuvent être réutilisés pour différents jeux de données avec des exigences communes de prétraitement et de traitement. Des tâches courantes et des fonctionnalités essentielles doivent être appuyées par le cadre de travail :

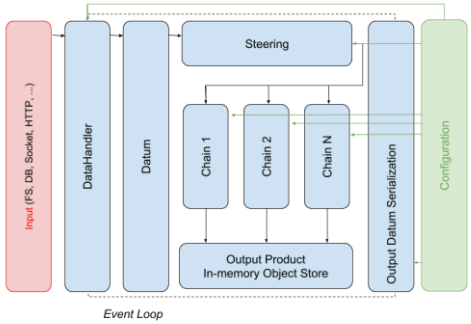
- Un soutien complet et robuste pour la lecture et l'écriture d'une diversité de formats de stockage en nuage, sur place, en entreposage de données local ou distribué (CSV, JSON, Parquet, données héritées, etc.).
- La capacité à identifier le schéma de données au moment de la lecture soit par le schéma fourni, soit par une inférence sur le type de données, et à produire un schéma bien défini pour le jeu de données.
- La capacité à extraire les métadonnées et de cataloguer les ensembles de données d'une façon indépendante du langage et de l'application.
- Un soutien pour les analyses itératives qui nécessitent une seule écriture et plusieurs lectures, avec un nombre minimal de conversions des données, d'entrées-sorties et de sérialisations.
- Une gestion efficace des données.
- Un filtrage efficace des données, par ex. la sélection de colonnes d'un jeux de données maître.
- La capacité à effectuer des opérations analytiques, par ex. des projections, des filtres, des agrégations, et des jointures.
- La collecte de statistiques sur les jeux de données, par ex. les distributions marginales, la moyenne, le minimum, le maximum.

L'hypothèse primaire pour la production de données est que des morceaux de données brutes peuvent être lus dans les tampons Arrow suivi du traitement sur des lots d'enregistrements en flux. Le traitement pourrait comprendre plusieurs sortes d'opérations du type filtrage-projection-agrégation sur des jeux de données très volumineux avec seulement un petit lot d'enregistrements en mémoire. Le traitement des enregistrements par lots peut être parallélisé sur plusieurs processeurs ou avec une grappe d'ordinateurs, avec pour résultat une économie d'échelle verticale et horizontale pour les ressources informatiques.

3.2 Contrôle de la séquence des opérations du prototype Artemis de Statistique Canada

L'objectif premier du prototype Artemis de Statistique Canada est la production de jeux de données qui utilisent les ressources de stockage et de traitement efficacement pour accélérer le traitement analytique des données sur une machine muticoeur avec un seul noeud. Le traitement des données peut être répliqué sur des parties indépendantes du jeu de données en parallèle sur plusieurs processeurs ou sur plusieurs noeuds d'une grappe d'ordinateurs d'une façon orientée envers les lots. Le jeu de données résultant comprend un ensemble de fichiers de sortie, chaque fichier étant organisé en un ensemble de lots. Chaque lot d'enregistrements comprend le même nombre d'enregistrements avec un schéma fixe et équivalent. Dans un fichier, l'ensemble des lots d'enregistrements peut être vu comme une table. La structure de données en colonnes est très compressible et conserve le schéma des données autant que l'intégralité de la charge utile dans un fichier donné. Arrow permet la lecture de lots d'enregistrements à la fois par flux et de façon aléatoire, avec pour résultat une gestion efficace et efficiente des données.

Figure 3.2-1
Séquence des opérations dans Artemis



De l'ingestion des données à la production des jeux de données Arrow, les étapes sont les suivantes. Les jeux de données brutes comprennent une ou plusieurs données (*Datum*), telles que des fichiers, des tables de bases de données ou toute autre partition de données. Pour organiser les données en ensembles comportant un nombre fixe de lots d'enregistrements à gérer en mémoire, chaque donnée est divisée en morceaux avec une taille fixe en octets. La donnée est directement lue dans des tampons natifs Arrow et tout le traitement est effectué sur ces tampons. Les tampons natifs Arrow en mémoire sont rassemblés et organisés en ensemble de lots d'enregistrements, par des algorithmes de conversion des données, de façon à construire de nouveaux jeux de données sur le disque dur à partir du flux de lots d'enregistrements. Afin de permettre toute transformation arbitraire définie par l'utilisateur, le prototype du cadre de travail définit un ensemble commun de classes de base pour des chaînes (*Chain*) définies par l'utilisateur, représentant des processus opérationnels, comme un ensemble ordonné d'algorithmes et d'outils qui transforment les données. Les algorithmes définis par l'utilisateur héritent des méthodes qui sont invoquées par l'application du prototype, de sorte que les chaînes sont gérées par un algorithme d'aiguillage (*Steering*). Le prototype du cadre gère le traitement des données de la boucle des événements (*Event Loop*), fournit les données aux algorithmes, et gère la sérialisation des données et la finition de la tâche. La conception est influencée par le système ATLAS High-Level Trigger, un logiciel pour filtrer les événements en temps réel, qui est utilisé pour sélectionner les événements d'intérêt en physique des hautes énergies (ATLAS, 2003).

3.3 Modèle de processus opérationnel

La conception du prototype Artemis de Statistique Canada sépare la définition du modèle de processus opérationnel (BPM, Business Process Model) de l'exécution de ces processus sur les données. Les modèles des processus opérationnels sont définis par l'utilisateur et conservés dans les métadonnées. La flexibilité de définir, de conserver et de stocker le BPM dans les métadonnées permet l'usage de configurations variées avec les mêmes données, permet à la tâche d'être reproductible, et facilite la validation des données ainsi que les tests de code.

Le BPM peut être représenté par un graphe orienté, décrivant les relations entre les données, leurs dépendances, et les processus à appliquer aux données. L'utilisateur définit les intrants, les extrants, les processus à appliquer aux intrants, et les algorithmes qui constituent un processus opérationnel distinct. Chaque processus opérationnel consomme des données et produit de nouvelles données, où les nouvelles données sont les extrants de un ou plusieurs algorithmes qui transforment les données. Une fois que les processus opérationnels sont représentés par une série d'algorithmes, les processus doivent être transformés d'un graphe orienté à une séquence linéaire de processus (où chaque processus est une liste d'algorithmes utilisant le même intrant). La séquence des algorithmes doit garantir le respect des dépendances avant l'exécution d'un algorithme.

La séquence d'exécution algorithmique est résolue avec un algorithme de tri topologique. Le tri topologique d'un graphe orienté est une séquence linéaire des sommets de sorte que pour chaque arc dirigé *uv* du sommet *u* vers le sommet *v*, *u* vient avant *v* dans la relation d'ordre. Les utilisateurs doivent seulement s'assurer que leur pipeline définit les intrants, la séquence d'algorithmes pour traiter ces intrants, et les extrants. Le graphe orienté des relations des données et la séquence d'exécution sont définis dans les métadonnées Artemis.

Une fois que les données brutes ont été transformées en lots d'enregistrements Arrow, le cadre de travail doit fournir les bonnes données intrantes aux algorithmes de sorte que les lots d'enregistrements finaux sont les produits de l'application du BPM aux données. Le cadre de travail a un algorithme principal, l'*aiguillage*, qui sert de moteur d'exécution pour les algorithmes définis par l'utilisateur. L'algorithme d'aiguillage gère les dépendances entre les données pour l'exécution du BPM en fournissant les données intrantes requises à chaque *noeud* dans un *arbre* via un *élément*. La structure de données de l'arbre est un graphe dirigé généré à partir du menu BPM spécifié par l'utilisateur. L'aiguillage tient et gère l'ensemble de la structure de l'arbre, qui comprend les éléments de chaque noeud, les relations entre les noeuds, et tous les tampons Arrow associées aux éléments. Les éléments sont un moyen de communication entre les algorithmes. Les tampons Arrow (tables de données) sont attachés aux éléments et sont accessibles aux algorithmes subséquents.

3.4 Modèle de métadonnées

Le modèle de métadonnées du prototype Artemis de Statistique Canada a trois composantes principales:

1. La définition de la tâche de traitement de données, c.-à-d. toutes les métadonnées requises pour exécuter le processus opérationnel.
2. Les métadonnées des tâches de traitement, c.-à-d. les métadonnées rassemblées durant l'exécution du BPM et la provenance des données.
3. Les métadonnées sommaires, c.-à-d. l'information statistique rassemblée durant le traitement des données.

Le modèle de métadonnées Artemis est défini dans un format de données basé sur le protocole Google de messagerie. Les tampons de protocole (Google) sont indépendants du langage et de la plateforme, avec un mécanisme extensible pour sérialiser les structures de données – pensez XML, mais plus petit, plus rapide et plus simple. Les tampons de protocole ont été développés par Google pour appuyer leur infrastructure de service et de communication entre les applications. Le format du message de tampons de protocole fournit une façon de définir la structure des métadonnées une seule fois, ensuite du code source spécifique est généré pour facilement écrire et lire les métadonnées structurées entre une diversité de flux de données en utilisant une variété de langages. Les messages de tampons de protocole peuvent être lus par reflet sans schéma, ce qui en fait un format extrêmement flexible en terme de développement d'application, d'usage et de persistance. Les messages peuvent être conservés simplement comme des fichiers sur un système local de fichiers, ou pour la gestion améliorée des métadonnées, les messages peuvent être catalogués dans une structure clé-valeur. Les applications peuvent persister et lire les configurations à l'aide d'une clé.

3.5 Qualité des données

La qualité des données est inhérente à l'analyse des données, et de ce fait doit être prise en compte durant les étapes clés de la conception du prototype. L'importance de la qualité des données transcende les domaines de la science, de l'ingénierie, du commerce, de la médecine, de la santé publique et des politiques publiques. Traditionnellement, la qualité des données peut être prise en compte en contrôlant les processus de mesure et de collecte de données et à travers la propriété des données. L'usage accru des sources de données administratives pose un problème à la fois

pour la propriété et le contrôle des données. Les outils, les techniques et les méthodologies de qualité des données devront évoluer (Keller et coll., 2017).

Dans le domaine scientifique, la qualité des données est généralement considérée comme faisant implicitement partie du rôle de l'utilisateur de données. L'infrastructure statistique devra appuyer la capacité des utilisateurs à mesurer la qualité des données tout au long du cycle de vie des données. Les outils statistiques requis pour mesurer la qualité des données doivent être développés pour résoudre les problèmes de qualité des données administratives. L'outil principal d'analyse statistique de la qualité des données du prototype du cadre de travail est l'histogramme.

Les histogrammes sont essentiels pour l'analyse statistique et la visualisation des données, et des outils clés pour le contrôle de la qualité (Freedman, 1981). Ce sont des outils statistiques parfaits pour décrire les données, qui peuvent être utilisés pour résumer de grands jeux de données en conservant à la fois les fréquences et les erreurs. L'histogramme est une représentation précise de la distribution de données numériques (ou de données catégorielles codées sous forme de dictionnaire), et il représente de façon graphique la relation entre une fonction de densité de probabilité $f(x)$ et un ensemble de n observations, $x_1, x_2, \ldots, x_n$ (Cowan, 1998). Le cadre de travail conserve les distributions associées au traitement et au coût global du traitement pour une surveillance centralisée des services. Il permet aussi des histogrammes définis par l'utilisateur dans les algorithmes. Les histogrammes pour la surveillance centralisée incluent la distribution des temps de rotation pour chaque étape de traitement et pour chaque algorithme du BPM, les tailles des charges utiles incluant les données et les morceaux, l'utilisation de la mémoire, la distribution des lots d'enregistrements traités, et les statistiques sur les erreurs de traitement.

Des applications distinctes de surveillance basée sur des histogrammes peuvent être développées comme une étape de post-traitement, et s'appliqueront rapidement à de grands jeux de données, puisque les données intrantes sont des lots d'enregistrements Arrow. Le profilage automatique des distributions de données est aussi envisageable en tant qu'intrant pour l'étape de post-traitement des algorithmes de qualité des données.

Résumé

Plusieurs organisations statistiques évoluent vers une approche qui priorise l'usage des données administratives pour la production de statistiques officielles. Pour la production de données administrative de haute qualité et adaptées à leur utilisation, il est essentiel que les analystes puissent utiliser les données, mesurer la qualité des données, et développer des applications des données et des outils statistiques de façon itérative. Les jeux de données administratives deviennent de plus en plus grands et complexes, représentant un défi pour notre approche d'analyse et de traitement des données. Le format de données en colonnes en mémoire Apache Arrow fournit la capacité de développer l'infrastructure pour résoudre les défis de qualité de données et d'adaptation à l'utilisation des données administratives; mieux définir les exigences informatiques en matière d'usage efficient du CPU et de la mémoire; construire un écosystème de données durable, sécuritaire, cohérent et capable d'interopérabilité; et favoriser un environnement de collaboration à travers la réutilisation de code. Le prototype du cadre de traitement de données Artemis de Statistique Canada démontre la facilité de concevoir, de construire et de modifier l'échelle d'un système de science des données à partir d'une seule librairie de logiciel libre pour livrer des jeux de données de haute qualité, conçus pour le traitement efficient sur un seul noeud multicoeur.

Bibliographie

Apache Arrow Project (2017), "Apache Arrow, a cross language development platform for in-memory data", retrieved from http://arrow.apache.org.

Berger, et. Al. (2015), "ARTUS – A Framework for Event-based Data Analysis in High-Energy Physics", Berger, J.; Colombo, F.; Friese, R.; Haitz, D.; Hauth, T.; Müller, T.; Quast, G.; Siber, G., arXiv:1511.00852

Atlas Collaboration, "ATLAS high-level trigger, data-acquisition and controls: Technical Design Report", CERN-LHCC-2003-022; ATLAS-TDR-16.

Google, "Google Protocol Buffers", retrieved from *https://developers.google.com/protocol-buffers/*

Van Der Walt, Stefan; Colbert, S. Chris; Varoquaux, Gael (2011), “The NumPy array: a structure for efficient numerical computation”, *Computing in Science and Engineering* 13, 2 pp. 22-30.

S. Keller; G. Korkmaz; M. Orr; A. Schroeder; and S. Schipp (2017), “The Evolution of Data Quality: Understanding the Transdisciplinary Origins of Data Quality Approaches” , *Annu. Rev. Stat Appl*. 4:85-108

Freedman, David; Diaconis, Persi (December 1981). "On the histogram as a density estimator: L2 theory", Probability Theory and Related Fields. HeidelbergL: Springer Berlin. 57

Glen Cowan (1998), *Statistical Data Analysis*, Oxford Press.