Apache Arrow：一个跨平台的内存数据交换格式

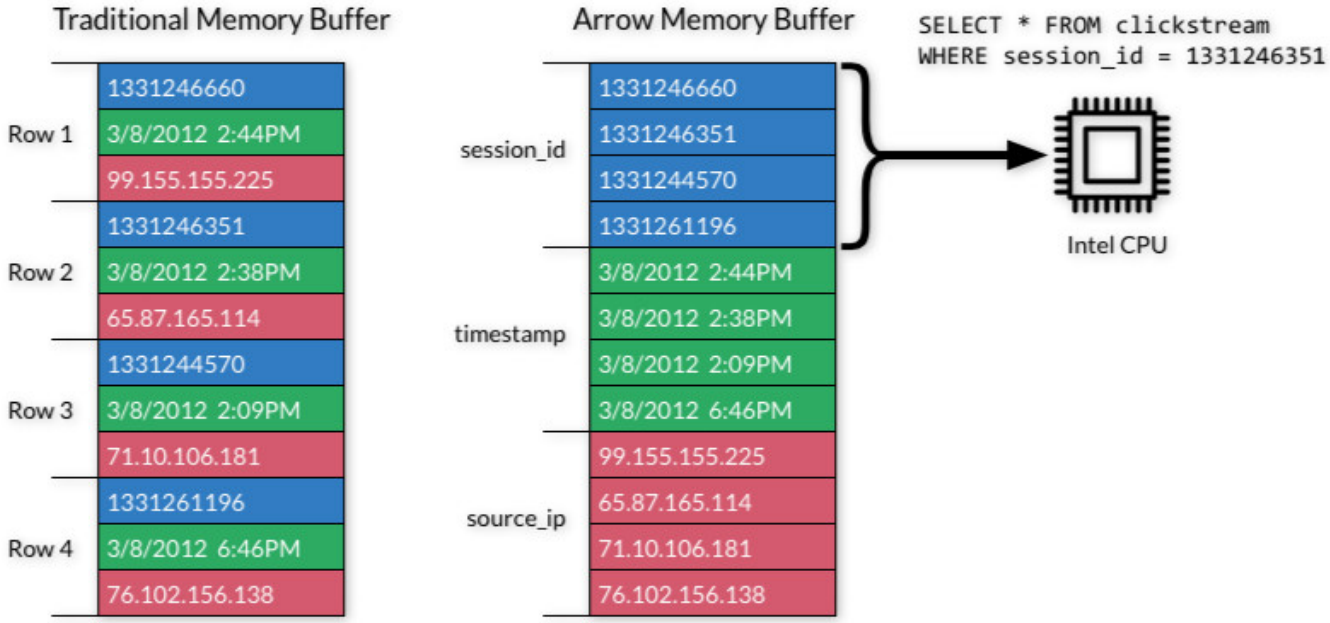
Apache Arrow是Apache基金会下一个全新的开源项目，同时也是顶级项目。它的目的是作为一个跨平台的数据层来加快大数据分析项目的运行速度。

用户在应用大数据分析时除了将Hadoop等大数据平台作为一个经济的存储和批处理平台之外也很看重分析系统的扩展性和性能。过去几年开源社区已经发布了很多工具来完善大数据分析的生态系统，这些工具涵盖了数据分析的各个层面，比如列式存储格式(Parquet/ORC)、内存计算层(Drill、Spark、Impala和Storm)以及强大的API接口(Python和R语言)。Arrow则是最新加入的一员，它提供了一种跨平台跨应用的内存数据交换格式。

提高大数据分析性能的一个重要手段是对列式数据的设计和处理。列式数据处理借助向量计算和SIMD使我们可以充分挖掘硬件的潜力。Apache Drill这一大数据查询引擎无论是在硬盘还是在内存中数据都是以列的方式存在的，而Arrow就是由Drill中Value Vector这一数据格式发展而来。除了列式数据，Apache Arrow也支持关系型和动态数据集，这使它成了处理物联网等数据时的理想格式选择。

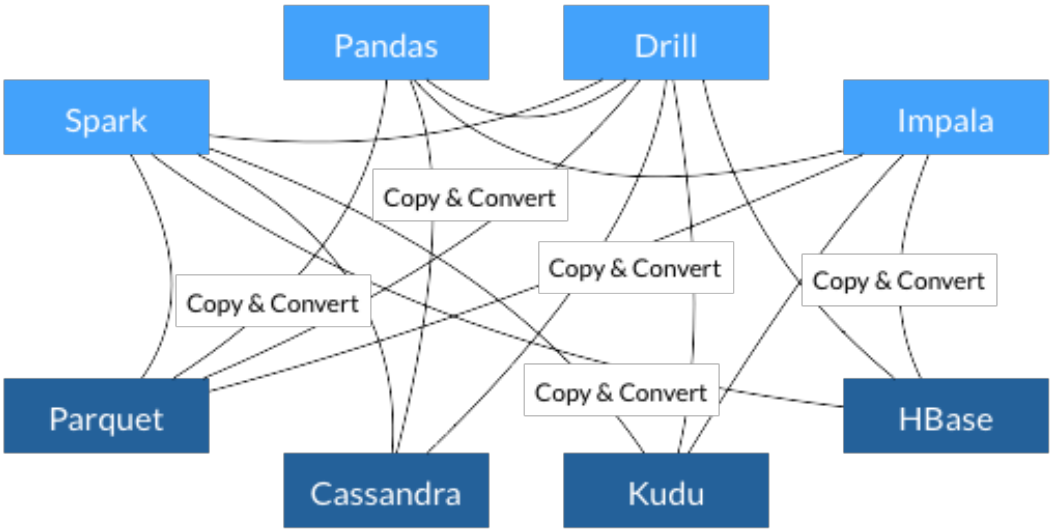
Apache Arrow为大数据生态系统带来了可能性是无穷的。有Apache Arrow做为今后的标准数据交换格式，各个数据分析的系统和应用之间的交互性可以说是上了一个新的台阶。过去大部分的CPU周期都花在了数据的序列化和反序列化上，现在我们则能够实现不同系统之间数据的无缝共享。这意味着用户在将不同的系统结合使用时再也不用为数据格式多花心思了。下图是列式存储格式和行式存储格式之间的区别：

	session_id	timestamp	source_ip
Row 1	1331246660	3/8/2012 2:44PM	99.155.155.225
Row 2	1331246351	3/8/2012 2:38PM	65.87.165.114
Row 3	1331244570	3/8/2012 2:09PM	71.10.106.181
Row 4	1331261196	3/8/2012 6:46PM	76.102.156.138

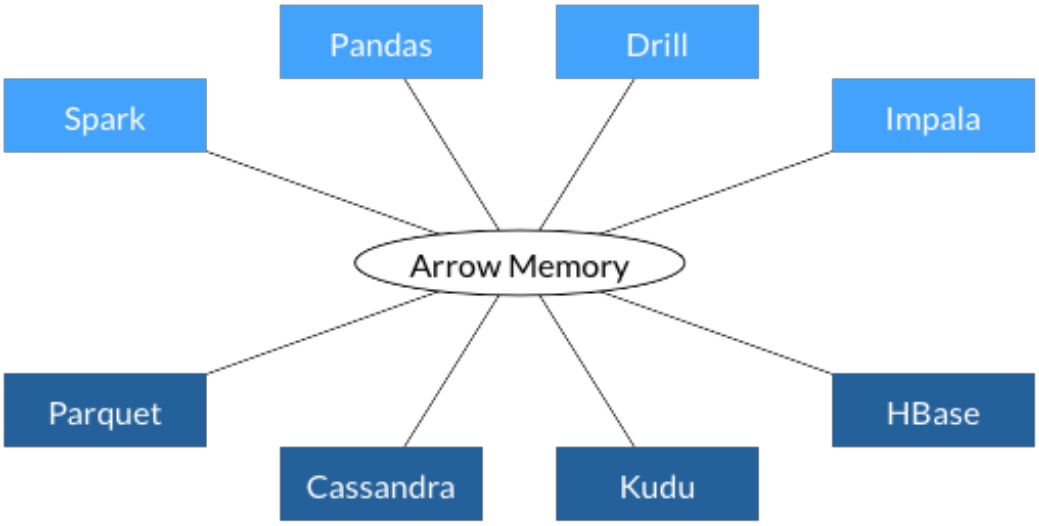


使用Apache Arrow前后优势对比

在不使用Apache Arrow项目的系统数据交互过程中：（1）、每个系统都有自己内部的内存格式；（2）、70-80%的CPU浪费在序列化和反序列化过程；（3）、类似功能在多个项目中实现，没有一个标准。如下图所示：



在使用Apache Arrow项目的系统数据交互过程中：（1）、所有系统都使用同一个内存格式；（2）、避免了系统间通信的开销；（3）、项目间可以共享功能（比如Parquet-to-Arrow reader）如下图所示：



本文转载自：http://geek.csdn.net/news/detail/58169；

英文原文：Apache Arrow – New Level of Performance and Interoperability for Big Data Analytics：https://www.mapr.com/blog/apache-arrow-new-level-performance-and-interoperability-big-data-analytics

本博客文章除特别声明，全部都是原创！
原创文章版权归过往记忆大数据（过往记忆）所有，未经许可不得转载。
本文链接:【】（）