

Providing large trade data sets for research using Apache Parquet and R Shiny

NTTS2021

David Zenz and Oliver Reiter

wiiw (wiiw.ac.at)

2021-02-04



Apache Arrow and Parquet

Apache Arrow

- ▶ platform for data analytics
- ▶ in-memory column-oriented layout
- ▶ a lot of language bindings
- ▶ major release v3.0.0 (Jan 25, 2021)

Apache Parquet

- ▶ column-oriented storage format
- ▶ uses enhanced in-memory performance
- ▶ Hadoop ecosystem
- ▶ separates metadata from data
- ▶ compressed & efficient data representation

Use case for large databases

Previous setup with Stata

- ▶ Trade data saved in huge number of separate files
- ▶ Long reading and writing times, cumbersome updating

Easy setup with Apache Parquet

```
write_dataset(data, path = base_output_path,  
              format = "parquet",  
              partitioning = c("Classification", "Year",  
                               "TradeFlow", "ReporterISO"))
```

Write size comparison

	CSV	Stata	Parquet		CSV	Stata
file size	~900GB	~1.2TB	55GB	ratio	16.36	21.82
				savings	93.89 %	95.42 %

Benchmark setup

- ▶ Debian 10.6 64bit / Linux 4.19.0-11-amd64 running on a 4-core AMD A10-7800 Radeon R7 with 32GB RAM and a 200GB SSD
- ▶ R v4.0.2, Shiny Server v1.5.14.948; Stata/MP 16 (8 core)
- ▶ Test data: UN Comtrade Classification HS12 (2012-2019)

	CSV	Stata	rds	rds (xz/9)	fst	fst (100)	Parquet
file size (GB)	41	80	57	1.3	14	3.96	3.8

Apache Parquet as compared to

	CSV	Stata	rds	rds (xz/9)	fst	fst (100)
ratio	10.79	21.05	15.00	0.34	3.68	1.04
savings	90.73%	95.25%	93.33%	none	72.86%	4.04%

Results: Queries

Queries

- ▶ 5 different queries, from simple to complex
- ▶ designed to imitate practical use-cases at our institute
- ▶ different subsets (flows, reporters, partners, years, commodities)

Timings (in seconds)

Query	CSV	Stata	rds	rds (xz/9)	fst	fst (100)	Parquet
Q1	0.89	0.32	1.44	2.70	0.44	0.50	3.77
Q2	4.08	1.42	6.71	12.07	3.44	2.96	4.40
Q3	18.43	11.11	31.77	55.56	10.16	12.74	4.63
Q4	18.60	11.73	32.09	60.52	11.37	13.54	5.44
Q5	139.10	96.19	180.23	455.23	79.63	89.62	8.32

Shiny GUI wiiw client



Dataset & Class Reporter/Partner & Products Country Aggregates Product Aggregates **Period & Flow & Unit** Download data

Range: 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 2016 2017

Note: NCU/EUR rate is period average, Source: Eurostat

Flow: IM

Unit: EUR TH
EUR
EUR TH
EUR MN
USD
USD TH
USD MN

RESET ALL FILTERS

Last update: 2018-09-13 15:56:14

Show 10* entries

Search:

Reporter	Partner	Flow	nace_rev2_2	nace_rev2_desc	unit	2005	2006	2007	2008	2009	2010
CZE	DACH Region	IM	01	Products of agriculture, hunting and related services	EUR TH	228,031.18	276,631.90	287,416.26	334,450.25	268,407.76	375,872.43
CZE	DACH Region	IM	02	Products of forestry, logging and related services	EUR TH	27,243.70	26,421.17	21,471.83	14,399.05	16,752.61	18,654.04
CZE	AUT	IM	01	Products of agriculture, hunting and related services	EUR TH	18,776.16	17,791.11	20,151.09	30,282.33	33,954.02	38,364.41
CZE	AUT	IM	02	Products of forestry, logging and related services	EUR TH	3,424.66	1,806.81	1,818.80	1,886.35	2,161.37	2,855.11
CZE	CHE	IM	01	Products of agriculture, hunting and related services	EUR TH	18.03	51.63	61.63	316.99	88.64	99.77
CZE	CHE	IM	02	Products of forestry, logging and related services	EUR TH	1.52	2.09	18.13	207.11	116.47	219.80
CZE	DEU	IM	01	Products of agriculture, hunting and related services	EUR TH	209,237.00	258,789.16	267,203.54	303,850.94	234,365.10	337,408.26
CZE	DEU	IM	02	Products of forestry, logging and related services	EUR TH	23,817.53	24,612.27	19,634.90	12,305.58	14,474.76	15,579.13

Showing 1 to 8 of 8 entries

Previous 1 Next

Conclusions

File size

- ▶ massive reduction in file size UN Comtrade database: from ~900GB to 55GB (93.89% space savings)
- ▶ allowing cheap storage of large & big data sets on modern PC hardware

Parquet performance

- ▶ outperforms CSV in all tested bigger queries
- ▶ when queries get bigger, Parquet constant, other struggle (except fst)
- ▶ best trade-off between timings & file size

Other advantages

- ▶ hides distributed file system
- ▶ enables user to write easier understandable code
- ▶ all other file-based databases require loop constructs

Thank you for your attention!

David Zenz: zenz@wiiw.ac.at | [@DavidZenz](https://twitter.com/DavidZenz)

Oliver Reiter: reiter@wiiw.ac.at | [@olreit](https://twitter.com/olreit)

Code available at:

<https://gitlab.com/DavidZenz/ntts2021-large-datasets-apache-parquet>

References

Romain François, Jeroen Ooms, Neal Richardson and Apache Arrow (2021). arrow: Integration to 'Apache' 'Arrow'. <https://CRAN.R-project.org/package=arrow>

Matt Dowle and Arun Srinivasan (2020). data.table: Extension of data.frame. R package version 1.13.6. <https://CRAN.R-project.org/package=data.table>

Hadley Wickham and Evan Miller (2020). haven: Import and Export 'SPSS', 'Stata' and 'SAS' Files. <https://CRAN.R-project.org/package=haven>

Winston Chang, Joe Cheng, JJ Allaire, Carson Sievert, Barret Schloerke, Yihui Xie, Jeff Allen, Jonathan McPherson, Alan Dipert and Barbara Borges (2021). shiny: Web Application Framework for R. <https://CRAN.R-project.org/package=shiny>

Barret Schloerke and Jeff Allen (2020). plumber: An API Generator for R. <https://CRAN.R-project.org/package=plumber>

Leo Lahti, Janne Huovari, Markus Kainu, Przemyslaw Biecek. Retrieval and analysis of Eurostat open data with the eurostat package. R Journal 9(1):385-392, 2017. <http://ropengov.github.io/eurostat>

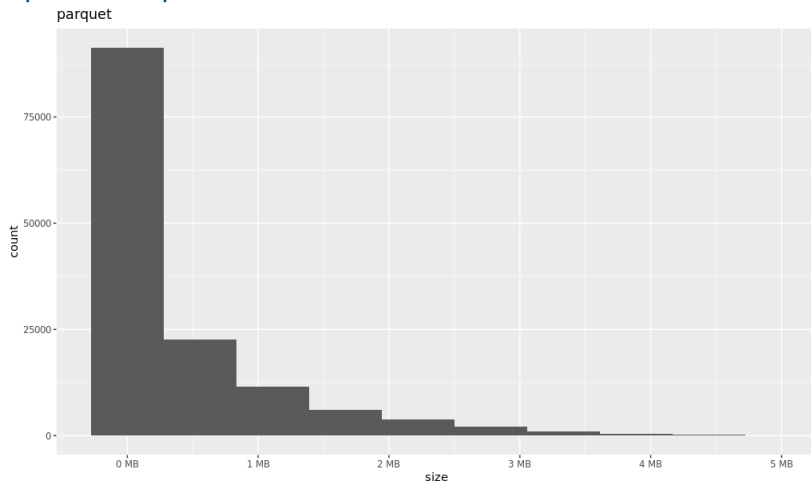
Timmer, M., Erumban, A.A., Gouma, R., Los, B., Temurshoev, U., de Vries, G.J., Arto, I.A., Genty, V.A.A., Neuwahl, F., Francois, J. and Pindyuk, O., 2012. The world input-output database (WIOD): contents, sources and methods (No. 20120401). Institute for International and Development Economics.

Boufe, A., Finkers, R., van Kaauwen, M., Kramer, M. and Athanasiadis, I.N., 2017, December. Managing variant calling files the big data way: Using HDFS and Apache Parquet. In Proceedings of the Fourth IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (pp. 219-226).

Mark Klik (2020). fst: Lightning Fast Serialization of Data Frames. <https://CRAN.R-project.org/package=fst>

Appendix

Apache Parquet database file size



summary statistics on file size (MB)

Apache Parquet

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	0.001140	0.007606	0.061394	0.455608	0.539806	11.188237

Stata/CSV

	flows	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	Stata	0.015851	8.525011	63.34861	184.16788	257.06717	1906.8167
2:	csv	0.001270	4.492057	32.07498	92.84871	131.01025	968.6813

	years	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	Stata	0.014635	12.126676	38.48883	64.00299	85.93286	405.6438
2:	csv	0.000232	6.340455	20.13017	32.37135	42.91962	202.5043

rds/fst default

Note: using default compression settings (rds: none, fst: 50%)

	flows	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	rds	0.003012	6.425210	46.13478	131.97573	187.05686	1360.7449
2:	fst	0.002638	2.432267	13.47476	30.84753	44.50416	283.6544
	years	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	rds	0.520376	20.102097	36.65086	56.89523	76.36421	289.3247
2:	fst	0.166317	5.489484	9.407384	13.30107	17.63233	57.18726

rds/fst compressed

	flows	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	rds	0.000560	0.198800	1.13725	2.95559	4.12263	30.7941
2:	fst	0.002373	0.533577	3.30211	8.94283	12.42210	92.0948
	years	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	rds	0.018792	0.5412400	0.910820	1.334852	1.788831	5.825256
2:	fst	0.034790	1.4354630	2.535617	3.857012	5.167992	17.314611

Example HS12 - Stata

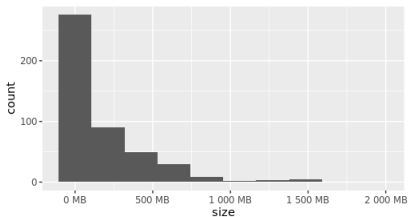
Summary statistics on filesize (MB) in *Stata* format

	partitioning	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1:	rough	401.718175	1020.475650	1265.857303	1191.66729	1400.213654	1672.2994
2:	less rough	0.010534	1.977533	5.341943	14.51347	11.273093	1471.0017
3:	granular-2d	0.011069	0.444637	1.944555	4.65645	5.365782	64.9372
4:	granular-6d	0.012687	3.315144	20.663137	68.26704	86.959175	1003.2492

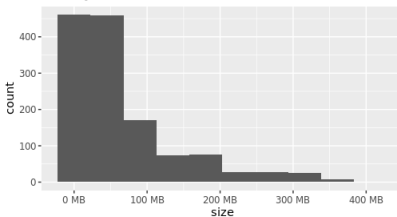
Benchmark partitioning

Stata/CSV

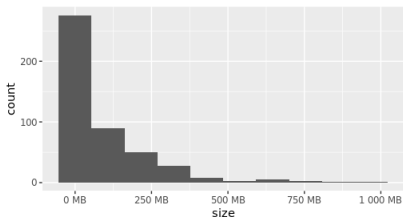
Stata flows



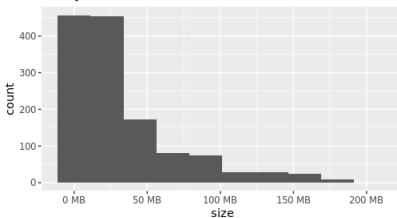
Stata years



CSV flows

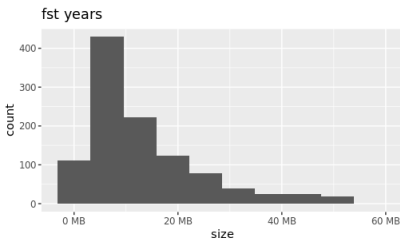
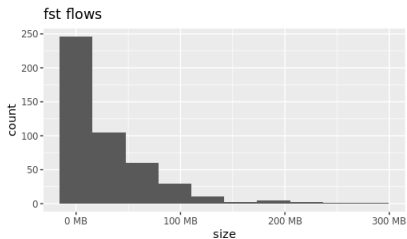
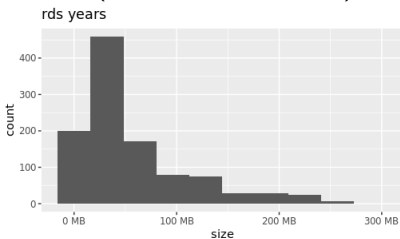
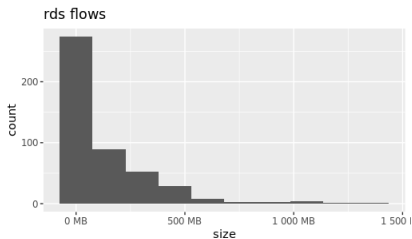


CSV years



rds/fst default compression

Note: using default compression settings (rds: none, fst: 50%)



rds/fst compressed

note: using compression settings (rds: xz/9, fst: 100%)

