DS 7347
High-Performance Computing (HPC) and Data Science
Session 20

Robert Kalescky
Adjunct Professor of Data Science
HPC Research Scientist

June 30, 2022

Research and Data Sciences Services
Office of Information Technology
Center for Research Computing
Southern Methodist University

Outline

Help Sessions

Session Question

File Formats

Readings and Assignments

Help Sessions

Help Sessions

- 30 minutes before and after Thursday's (6/30/22) session
- 6:00 - 6:30 PM and 8:00 to 8:30 PM (Central)

Session Question

Session Question: Tuesday

What are some different ways the following data could be stored or encoded?

Table 2
Catalogue of the largest known mafic eruptive units from LIPs (≥1300 km^3) ordered in terms of eruptive volume. The Mesozoic to Cenozoic LIPs are the best studied and preserved and the catalogue is biased to these more modern examples. For the basaltic eruptions, the Columbia River flood basalt province may contain more than 300 individual basalt lava flows that have an average volume of 500–600 km^3 (Tolan et al., 1989). Eruptive volumes are dense rock equivalent. Eruption magnitude is based on Pyle (1995, 2000) using a magma density of 2700 kg m^{-3}. Note that the eruptive volume for the Mahabaleshwar–Rajahmundry Traps eruptive unit is an upper end of a plausible range of eruptive volumes for this pahoehoe lava flow field that reached across the Indian subcontinent (see Self et al., 2008).

Eruptive Unit	LIP	Eruptive Age (Ma)	Minimum Eruptive Volume (km^3)	Lithology and Thickness (m)	Magnitude	Composition (wt.% SiO2)	References
Mahabaleshwar–Rajahmundry Traps (Upper)	Deccan	64.8	9300	Basalt lava (20–50)	9.40	High-Ti tholeiitic Basalt (48.1)	Self et al. (2008)
McCoy Canyon flow (Sentinel Bluffs Member, Grande Ronde N_2)	Columbia River	15.6	4278	Basalt lava (10–60)	9.06	Tholeiitic Basalt (53.6)	Reidel (2005); Landon and Long (1989)
Umtanum flow (Grande Ronde N_2)[1]	Columbia River	~15.6	~2750	Basalt lava (~50)	8.87	Tholeiitic Basalt (54.7)	Reidel et al. (1989)
Sand Hollow flow (Frenchmans Springs member, Wanapum Basalt)	Columbia River	15.3	2660	Basalt lava (~40)	8.86	Tholeiitic Basalt (51.8)	Beeson et al. (1985) Tolan et al. (1989)
Pruitt Draw flow (Teepee Butte Member, Grande Ronde R_1)	Columbia River	16.5	2350	Basalt Lava (30–100)	8.80	Tholeiitic Basalt (53.0)	Reidel and Tolan (1992); Reidel (1983)
Museum flow (Sentinel Bluffs Member, Grande Ronde N_2)	Columbia River	15.6	2349	Basalt Lava (10–80)	8.80	Tholeiitic Basalt (54.2)	Reidel (2005); Landon and Long (1989)
Rosalia flow (Priest Rapids Member, Wanapum Basalt)	Columbia River	14.5	1900	Basalt lava (~50)	8.70	Tholeiitic Basalt (50.5)	Tolan et al. (1989)
Joseph Creek flow (Teepee Butte Member, Grande Ronde R_1)	Columbia River	16.5	1850	Basalt Lava (20–90)	8.70	Tholeiitic Basalt (52.3)	Reidel and Tolan (1992)
Ginkgo Basalt (Frenchmans Springs member)	Columbia River	15.3	1600	Basalt lava (30–>150)	8.64	Tholeiitic Basalt (51.5)	Tolan et al. (1989); Reidel et al. (1994); Beeson et al. (1985)
Rosa Member (Wanapum Basalt)	Columbia River	14.5	1300	Basalt lava (3–50)	8.55	Tholeiitic Basalt (50.2)	Tolan et al. (1989); Self et al. (1997)
Stember Creek flow (Sentinel Bluffs Member, Grande Ronde N_2)	Columbia River	15.6	1192	Basalt lava (5–50)	8.51	Tholeiitic Basalt (53.5)	Reidel (2005); Landon and Long (1989)

1. Reidel et al. (1989) recognised the Umtanum unit comprised 2 lava flow units with a cumulative volume of >5500 km^3, but an average of 2750 km^3 for each lava flow.

Session Question: Thursday

What makes inference unique as compared to training on an HPC system?

File Formats

Commonly Used Plain Text File Formats

- Plan text
- CSV
- TSV
- JSON

Commonly Used Binary File Formats

- Tape Archives (tar)
- HDF5
- netCDF
- Apache ORC (Optimized Row Columnar)
- Apache Arrow:
  - Feather
  - Parquet
- Binary JSON

Compressed File Formats

- Used alongside text and binary formats
- Common compression tools:
  - Gzip
  - Bzip
  - Xz

Advantages and Disadvatnages

Text Files

- Easy to inspect
- Easy to edit
- Can be slow and large
- Can be error prone
- Everything is a string being reinterpreted

Binary Files

- Can be harder to inspect
- Hard to edit
- Can be fast
- Can be smaller and faster
- Can preserve data types

Hierarchical Data Format (HDF)

- Originally developed at the National Center for Supercomputing Applications (NCSA)
- Now supported by The HDF Group
- Features:
  - High performance with options for parallel reading and writing
  - Explicit data types
  - Multiple datasets
  - Compression
  - Mutable or static
  - Many official and unofficial interfaces

HDF5 Interfaces

- Official support for C, C++, Fortran, .Net, and Java
- Modern C++:
  - HighFive
  - h5cpp
  - ESS h5cpp
- Python:
  - PyTables
  - Pandas (Via PyTables)
  - h5py

Apache Parquet

- https://parquet.apache.org
- Part of the Apache Hadoop ecosystem
- Available in many Hadoop-adjacent tools as well
- Features:
  - High performance
  - Explicit data types
  - Good for columnar data storage
  - Compression

Apache Feather

- Part of Apache Arrow
- Features:
  - High performance
  - Explicit data types
  - Good for columnar data storage
  - Compression
  - Many official interfaces, *C, C++, C#, Java, Python, etc.*

Pandas and Dask Support

- Pandas IO Tools
- Dask

Group: Comparison of File Formats

- Time reading and writing

  `/scratch/group/oit_research_data/hansard/hansard_20191119.{tsv,.tsv.gz,parquet,feather}`
- On M2, via HPC Portal or SSH:
- Bootstap Conda environment: `module load python/3 && conda create --name intel -c intel python pandas pyarrow jupyterlab`
- JupyterLab Options
  - Partition: `medium-mem-1-s`
  - Environment: `intel`
  - Time: `2`
  - Nodes: `1`
  - Cores: `36`
  - GPUs: `0`
  - Memory: `750`

Comparing File Formats Hints

- Use IPython time magic
- Use `sep='\t '` for reading TSV files
- Does copying the data to `/dev/shm` first help?
- Compare different compression options

Readings and Assignments

Readings and Assignments

Readings

None

Project

- Explore various file formats for your data and compare performance
- Commit the results to your project repo