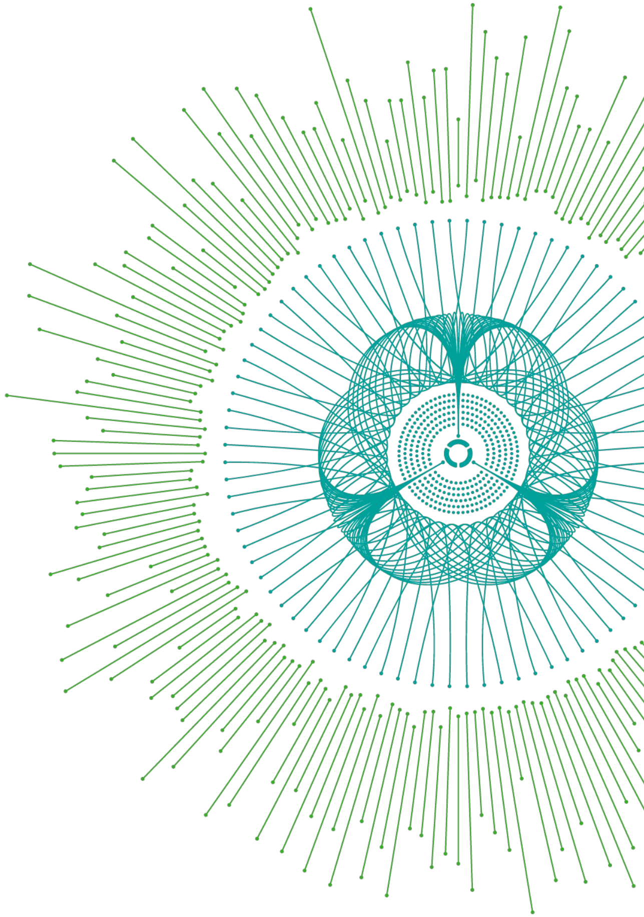
ANACONDA®



Working with Data in the Cloud

Martin Durant

October 2020

Contents

Introduction:

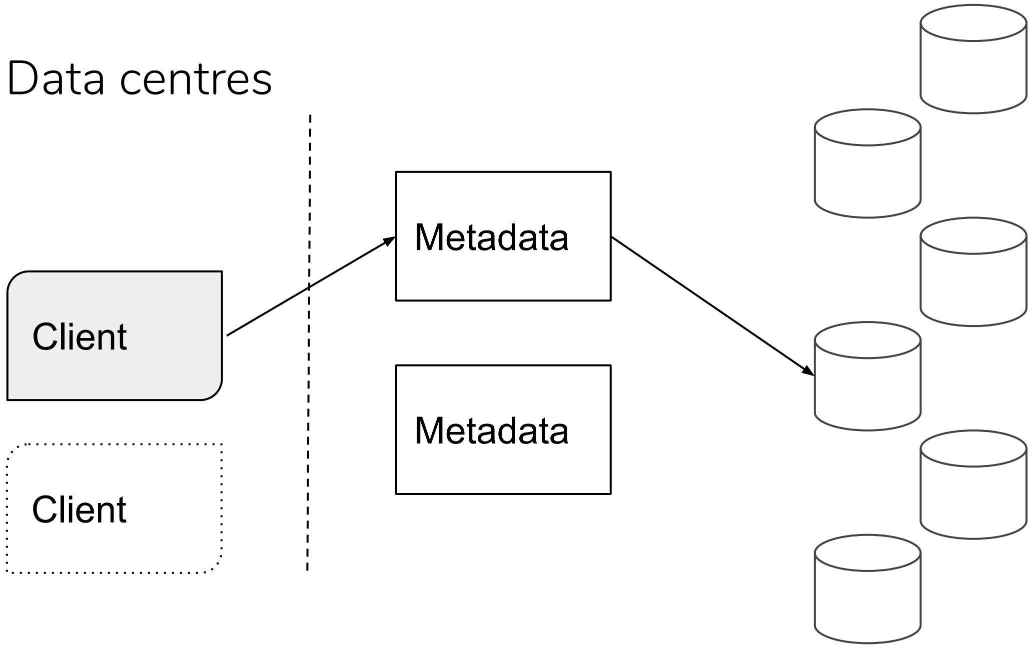
Big data, Bigger data

Big, Bigger

- We are drowning in ever more data (TB->PB)
- Storing in memory is not an option
- Downloading is not an option
- Stored in variety of data farms:
    - Cloud provider object stores
    - Cluster storage (HDFS)
    - Institutional servers (NFS etc)
- Auth and access

Big, Bigger

- Data centres



Big, Bigger: caveats

- Object stores are not filesystems
- Every system has its own ideas about auth
- Access from within the network very different from without

fsspec

fsspec: introduction

- Consistent API over many storage backend
- Explore any store like local files
- Makes file-like objects
- Integrates with python ecosystem

fsspec: introduction

- Implements:
    - local
    - http
    - (s)ftp
    - (web/http)HDFS
    - github, jupyter, git
    - AWS S3
    - GCS
    - MS SMB
    - Azure datalake/blob
    - dropbox, gdrive

fsspec: examples

fsspec: bonus

- Convenience functions
- Path expansion
- Bulk operations
- Async/concurrency
- Caching and buffering
- Compression and text mode
- Cloud friendly

Cloud formats

Cloud-friendly binary formats

- A well-designed file format has the following features:
  - Human-readable metadata, with descriptive attributes
  - Binary format for efficiency and compactness
  - Choice of compression and filter operations: CPU versus size
  - Explicit data types
  - Chunks/fields of data can be accessed independently
  - **Works seamlessly with remote/cloud storage**

Cloud-friendly?

- JSON { }
- XML < / >
- CSV , ,
    - with schema?
- Excel
- HDF
- pickle
- avro/thrift/protobufs
- proprietary/custom

Cloud-friendly? Excel??

```
import pandas as pd
pd.read_excel(
    "zip://FinancialSample.xlsx::"
    "s3://mymdtemp/FinancialSample.xlsx.zip"
)
```

Cloud-friendly: parquet and zarr

- Efficient binary encoding and compression
- Metadata stored separately
- Strongly typed
- Natural chunking - load what you need
- Filename conventions

Cloud formats: examples

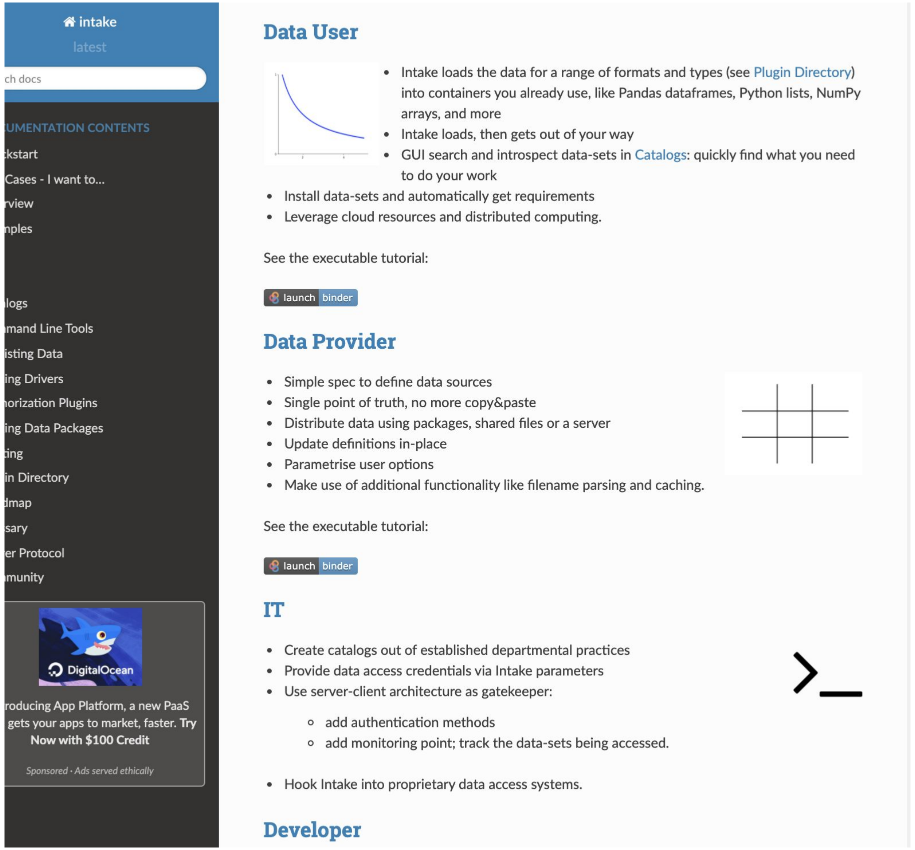
Cataloging

Cataloging

INTAKE

- Consistent API to all data sources
- Hierarchical, searchable catalogue trees
- Metadata lives with dataset declaration
- Intake allows data as declarative code, you can
    - Share (files, remote)
    - Update in place
    - Version control
    - Group and structure
    - Package and distribute

Cataloging



intake
latest

ch docs

UMENTATION CONTENTS
kstart
Cases - I want to...
rview
nples

logs
mand Line Tools
isting Data
ing Drivers
orization Plugins
ing Data Packages
ting
in Directory
dmap
sary
er Protocol
munity

DigitalOcean
roducing App Platform, a new PaaS gets your apps to market, faster. **Try Now with $100 Credit**

Sponsored · Ads served ethically

Data User

- Intake loads the data for a range of formats and types (see Plugin Directory) into containers you already use, like Pandas dataframes, Python lists, NumPy arrays, and more
- Intake loads, then gets out of your way
- GUI search and introspect data-sets in Catalogs: quickly find what you need to do your work
- Install data-sets and automatically get requirements
- Leverage cloud resources and distributed computing.

See the executable tutorial:

launch binder

Data Provider

- Simple spec to define data sources
- Single point of truth, no more copy&paste
- Distribute data using packages, shared files or a server
- Update definitions in-place
- Parametrise user options
- Make use of additional functionality like filename parsing and caching.

See the executable tutorial:

launch binder

IT

- Create catalogs out of established departmental practices
- Provide data access credentials via Intake parameters
- Use server-client architecture as gatekeeper:
  - add authentication methods
  - add monitoring point; track the data-sets being accessed.
- Hook Intake into proprietary data access systems.

Developer

INTAKE

Making a catalogue

- Figure out how to load
- Encode the spec
- Save to file
- Add detail
- Done
- Update!

Share catalogue

- Send the file
- Upload the file, add access or make public
  - network drive
  - github
  - dropbox/gdrive
  - cloud providers (azure, aws, gcp)
  - file server
- Build a package

Intake Example

Compute to data

- Code is small
- Results are (usually) small
- Easily scale up or out
- Bandwidth
- Familiar browser UI
- Sharing

Summary

Summary

- Big data:
    - store in the cloud or network resources
    - `fsspec` allows uniform experience for all backends
- Cloud friendly formats
    - chunking
- Metadata and cataloging
    - `Intake` to reference all your data sources
- Compute to data

Thank You!

ANACONDA®