Klimatic: A Virtual Data Lake for Harvesting and Distribution of Geospatial Data

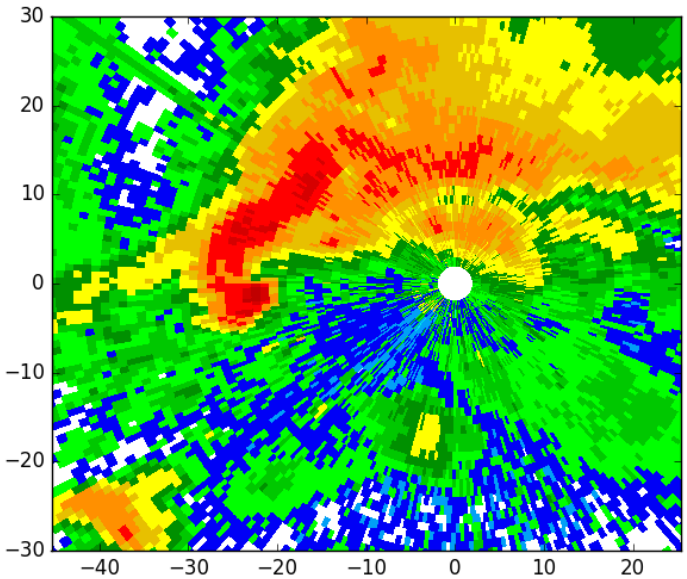
Tyler J. Skluzacek, Kyle Chard, Ian Foster

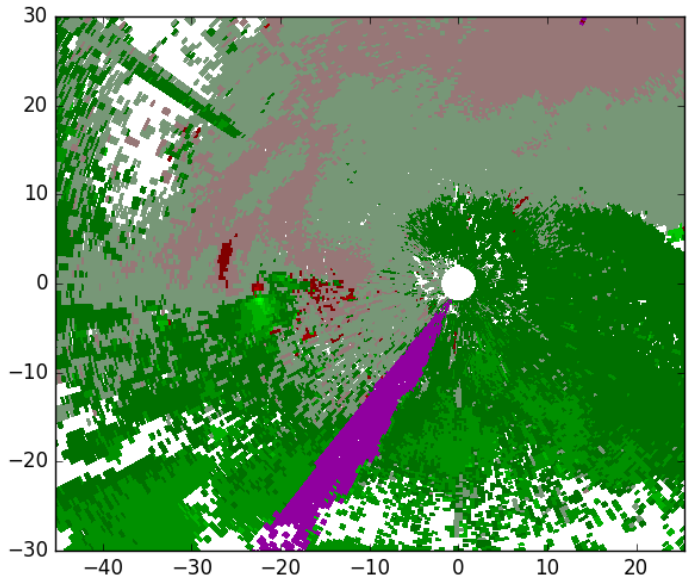
PDSW-DISCS 2016

November 14, 2016

Motivation

- Disparate research datasets stored in dark, siloed repositories.
- Researchers want robustness.
- Data hidden across HTTP and FTP servers (Globus GridFTP).
- Scalable architecture needed to find, index, integrate, and distribute.
- Geospatial data especially inaccessible to users (format, size, complexity).
  - Ex: NetCDF





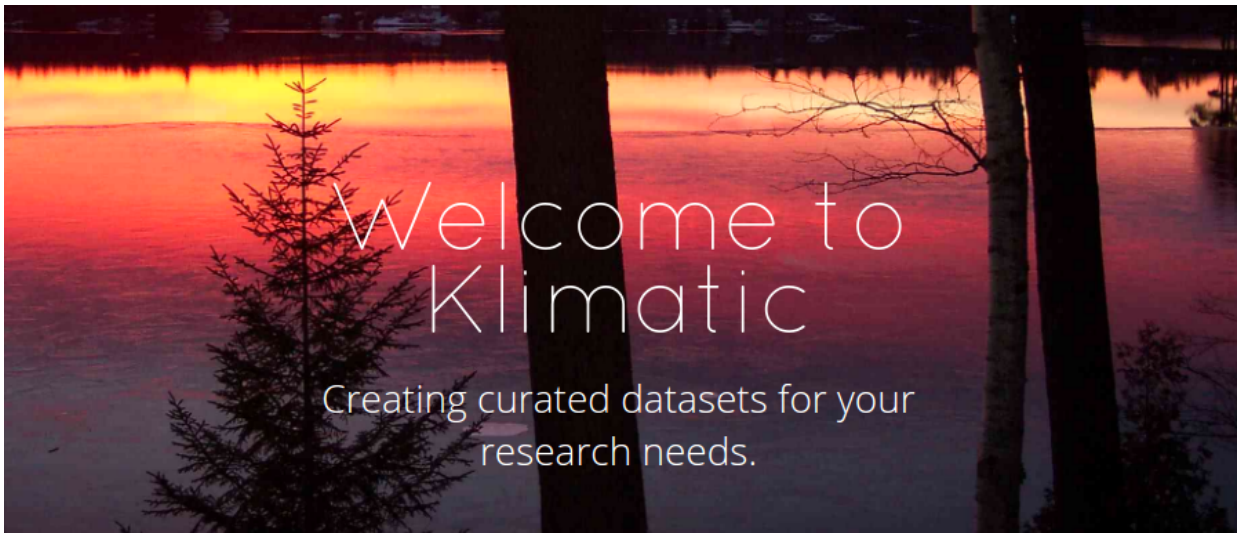
Problem Constraints

- Data integrity must be upheld.
  - *i.e.*, data in system = data in wild
- Non-standard naming and coding conventions.
- Available data storage.
- Must be scalable.
- Intuitive queries (or lack thereof for lay(wo)man).
- The process should be automated.

Proposed Solution

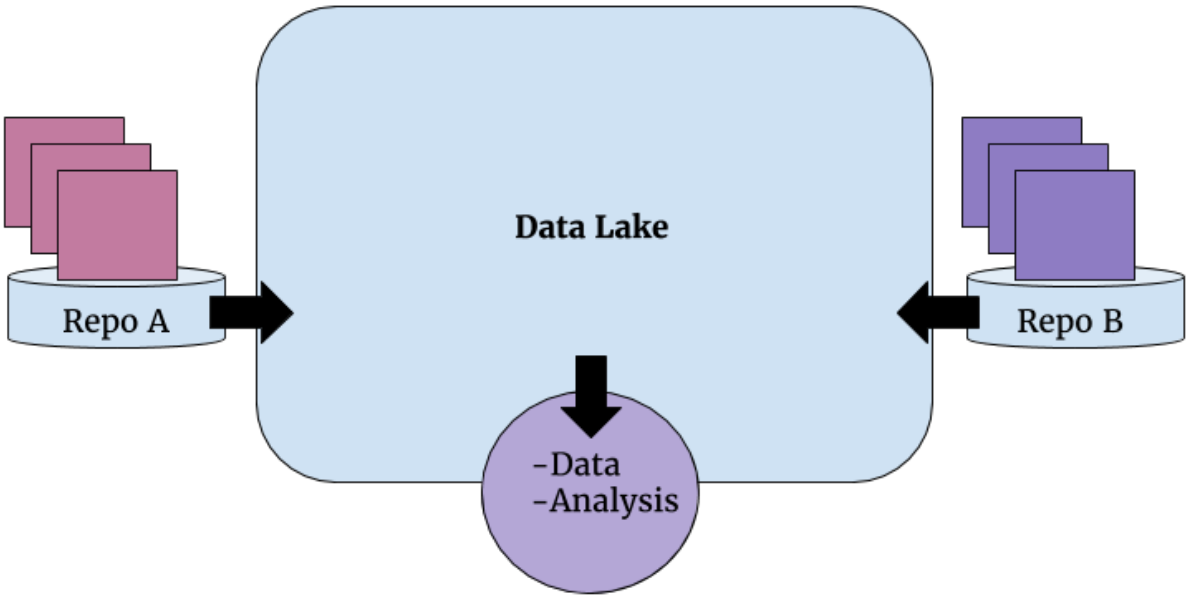
Enter Klimatic

- Quick access for researchers to search a world of data.
- Allows simple querying across datasets.
- Automated integration of compatible datasets into necessity-sized chunks.
- Introduction of the container-based *Virtual Data Lake*.



The Data Lake

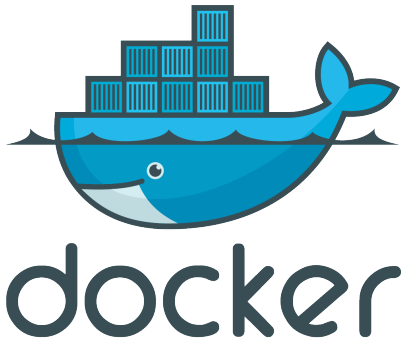
"The term data lake has been coined to convey the concept of a centralized repository containing virtually inexhaustible amounts of raw (or minimally curated) data that is readily made available anytime to anyone".



[1] I. Terrizano, et al. *Data Wrangling: The Challenging Journey from the Wild to the Lake*. *CIDR*, 2015.

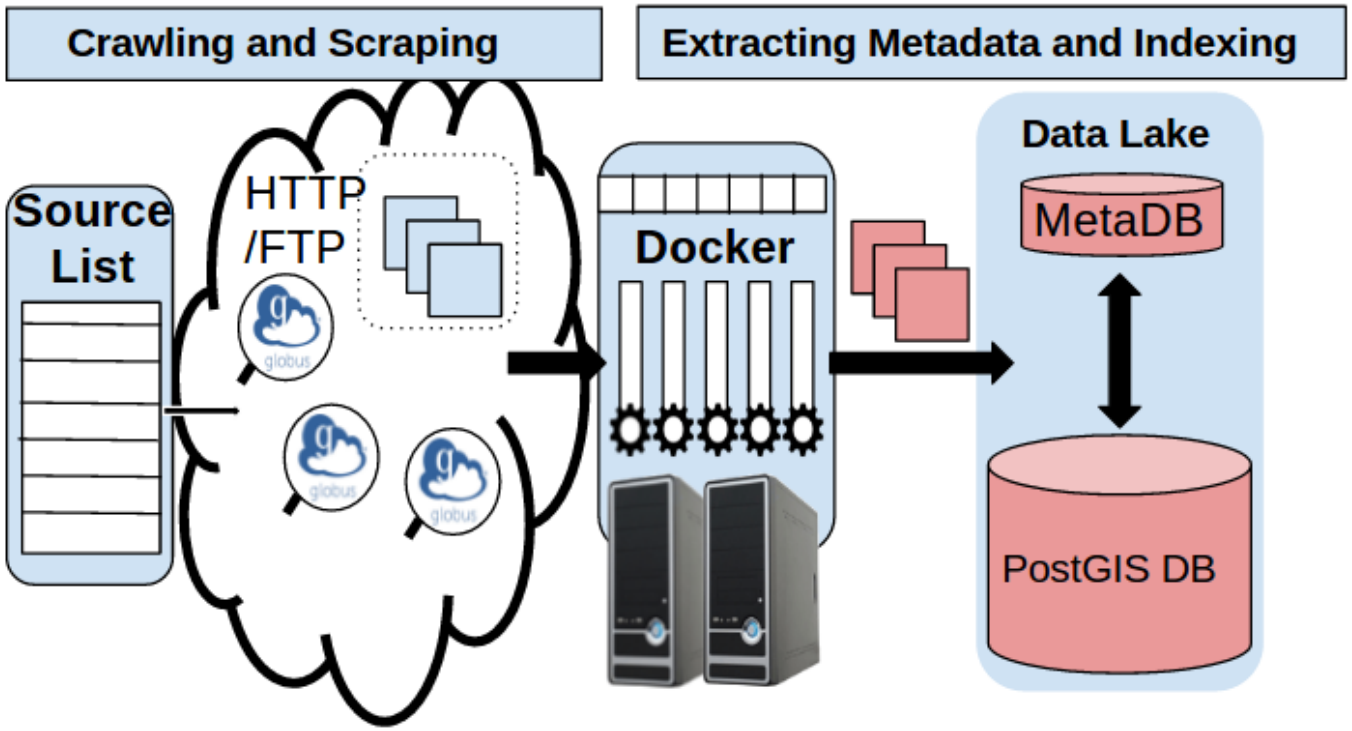
The "Virtual" Data Lake

- Data Lake that conflates locally stored data with remote, indexed data
- Metadata for all data stored locally; only important* raw data stored.
- *Importance based on relevance (-rel), size (-sz), and provider (-prv).
- Container Model: Extractor instances run in Docker containers.



Architecture: Collection

- Set number of data extraction instances (Docker containers).
- Extraction instance scans source from list—extracts all files.
- Check *HTTP/FTP* for nested repos/links. Append to list.
- Create searchable *TS_Vector* with metadata attributes of data.
- Store metadata, consider dataset for raw storage vs. eviction.



Scraping HTTP vs. Globus GridFTP

HTTP:

- Utilize existing tools (Scrapy) and in-house tools to pull data.
- Trump wingdings and Javascript-embedded files.
- Scrape *context* in addition to content (*in early stages*).

Globus:

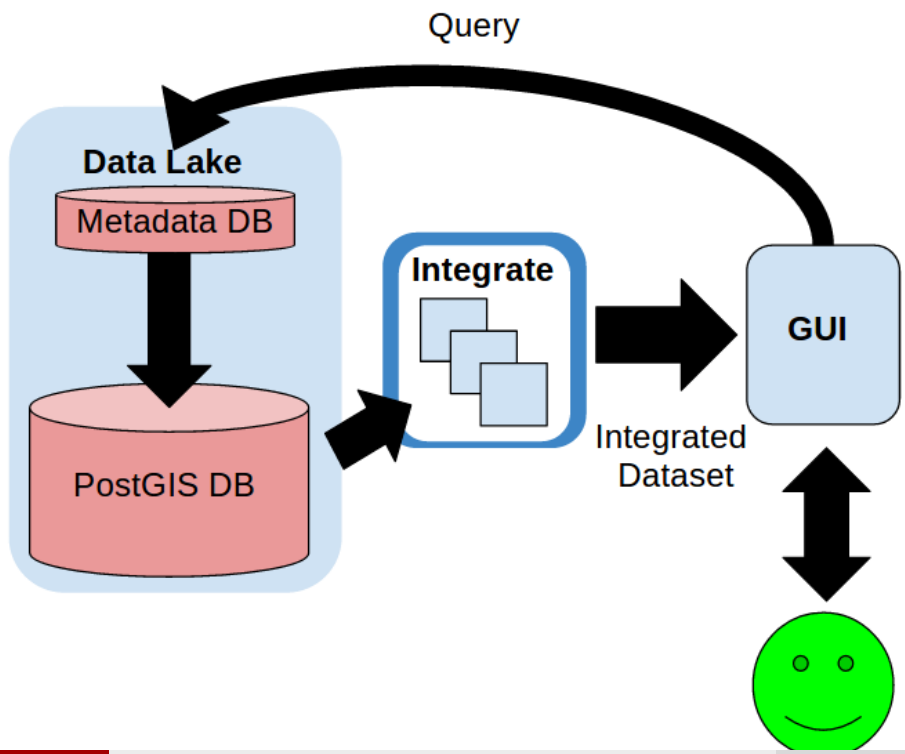
- Spawn list of candidate files stored in publicly-accessible endpoints.
- Use Globus Transfer API (Python) to pull all candidate datasets from the repositories.
- Path is "Globus User ID" followed by file system's path to data.

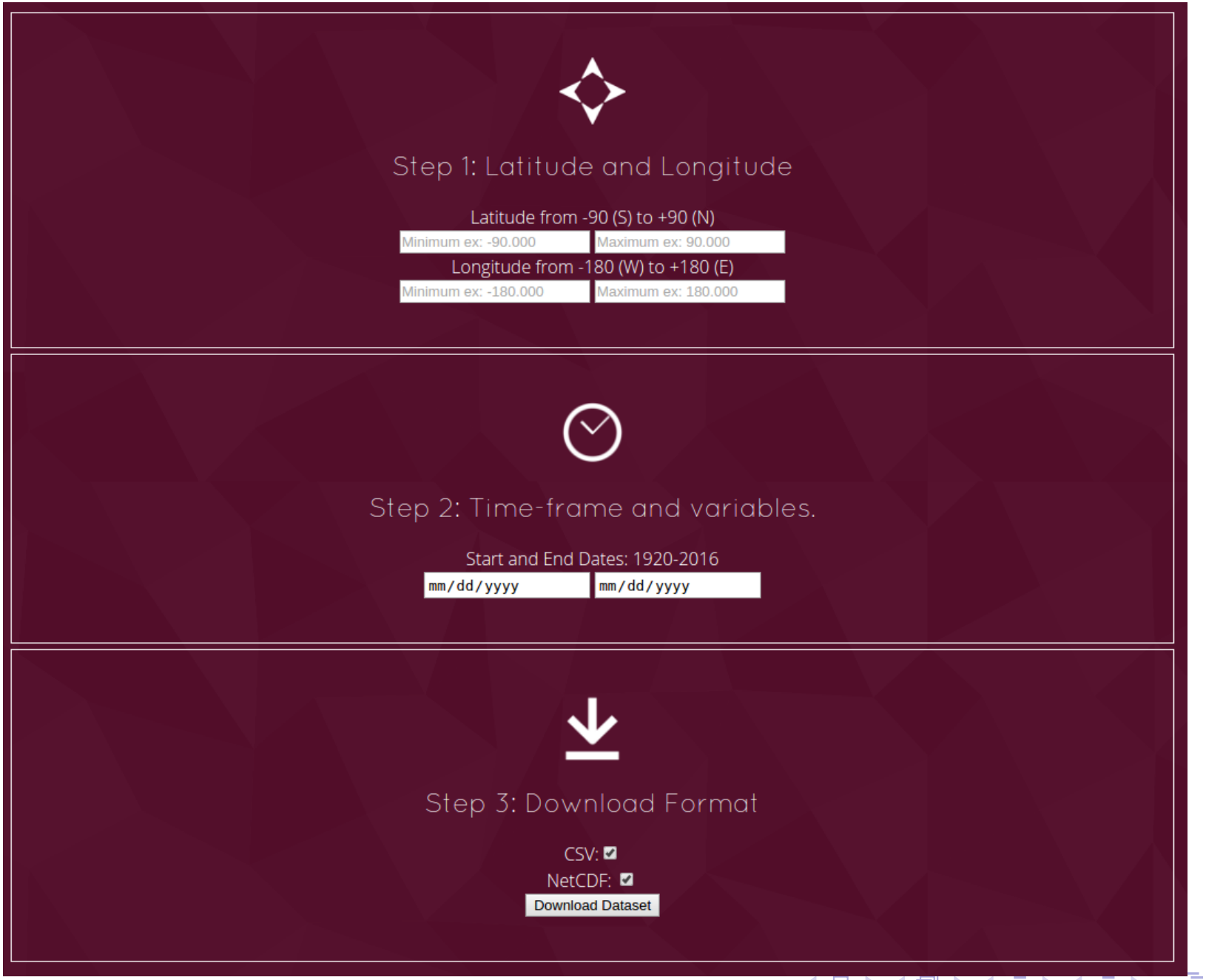
Metadata Extraction

- Open each file to find key attributes.
- Search header for human-collected keywords.
  - Latitude: stlat, lats, lati, stdlat, y, lt, north, NS, and N.
- Standardize attributes before insertion into metaDB.
- Create searchable indexed string:
  latMin55.232latMax66.000lonMin0.000lonMax180.000resolution12km...
- Compute and compare checksums.
- Evict or store?

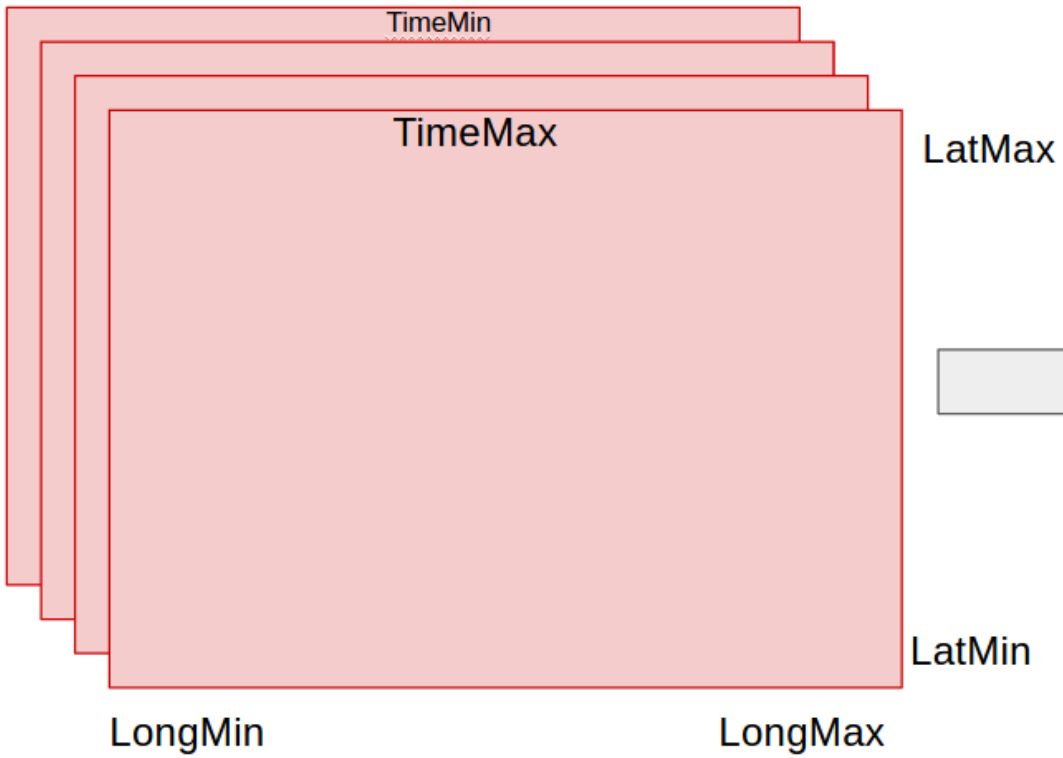
Architecture: Distribution

- User requests desired traits of data from GUI. Query sent to data lake.
- If possible, pull all candidate for an integrated dataset.
  - Requested datasets in vector-format fitted to grid.
- Datasets integrated on 'snap-to-larger' basis.
- Delivery in desired format.




Step 1: Latitude and Longitude
Latitude from -90 (S) to +90 (N)
Minimum ex: -90.000
Maximum ex: 90.000
Longitude from -180 (W) to +180 (E)
Minimum ex: -180.000
Maximum ex: 180.000
Step 2: Time-frame and variables.
Start and End Dates: 1920-2016
mm/dd/yyyy
mm/dd/yyyy
Step 3: Download Format
CSV:
NetCDF:
Download Dataset

Query: Building a Bounding Box

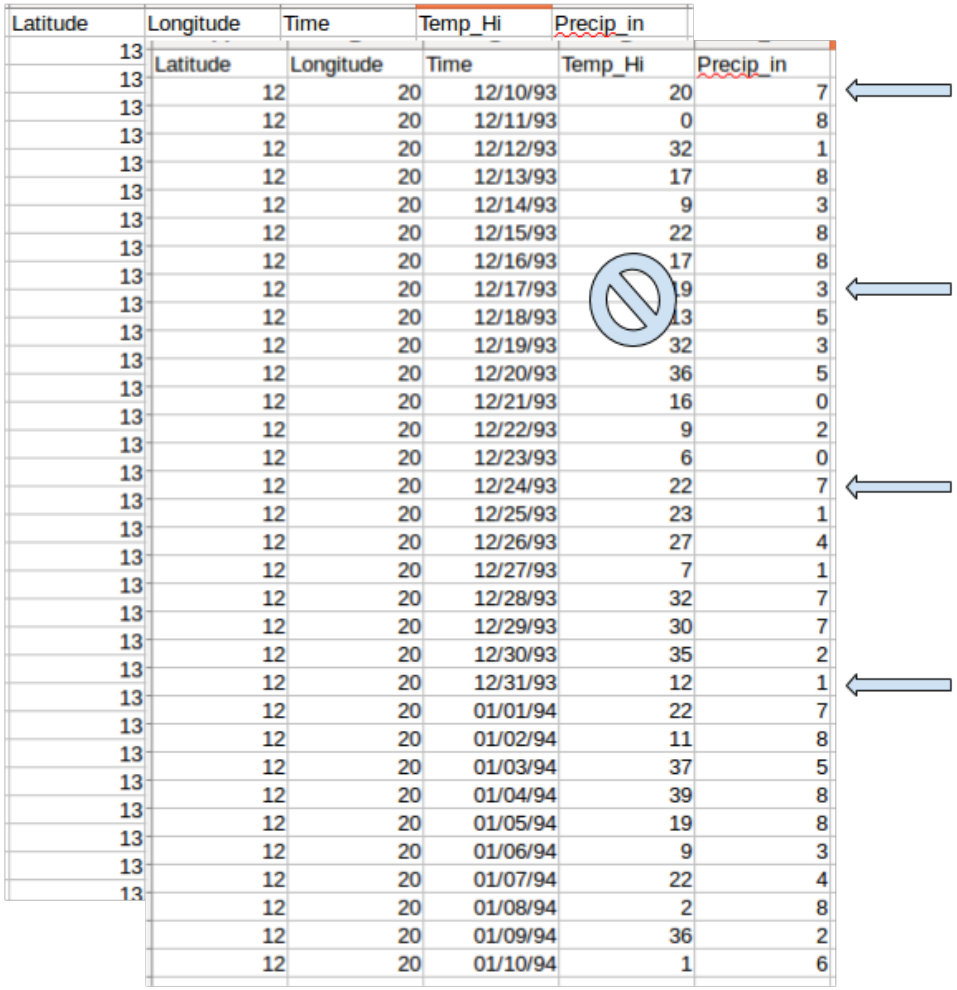


Provide Options Based on Availability

-Covariates
  -Resolution(s)
-Provider
-Output format (.nc/.csv)

Integration: Identify Necessary Data

"I want precipitation data for every Tuesday in December on/after the 10th, at latitudes 11-13 and longitude 20".



Latitude	Longitude	Time	Temp_Hi	Precip_in
12	20	12/10/93	20	7
12	20	12/11/93	0	8
12	20	12/12/93	32	1
12	20	12/13/93	17	8
12	20	12/14/93	9	3
12	20	12/15/93	22	8
12	20	12/16/93	17	8
12	20	12/17/93	19	3
12	20	12/18/93	13	5
12	20	12/19/93	32	3
12	20	12/20/93	36	5
12	20	12/21/93	16	0
12	20	12/22/93	9	2
12	20	12/23/93	6	0
12	20	12/24/93	22	7
12	20	12/25/93	23	1
12	20	12/26/93	27	4
12	20	12/27/93	7	1
12	20	12/28/93	32	7
12	20	12/29/93	30	7
12	20	12/30/93	35	2
12	20	12/31/93	12	1
12	20	01/01/94	22	7
12	20	01/02/94	11	8
12	20	01/03/94	37	5
12	20	01/04/94	39	8
12	20	01/05/94	19	8
12	20	01/06/94	9	3
12	20	01/07/94	22	4
12	20	01/08/94	2	8
12	20	01/09/94	36	2
12	20	01/10/94	1	6

Time	Latitude	Longitude	Precip_in
12/10/93	11	20	3
12/17/93	11	20	4
12/24/93	11	20	7
12/31/93	11	20	5
01/07/94	11	20	5
01/14/94	11	20	4
01/21/94	11	20	6
01/28/94	11	20	2
02/04/94	11	20	2
02/11/94	11	20	7
02/18/94	11	20	4
02/25/94	11	20	1
03/04/94	11	20	8
03/11/94	11	20	4
03/18/94	11	20	5
03/25/94	11	20	1
04/01/94	11	20	3
04/08/94	11	20	1
04/15/94	11	20	0
04/22/94	11	20	1
04/29/94	11	20	3
05/06/94	11	20	2
05/13/94	11	20	4
05/20/94	11	20	3
05/27/94	11	20	5
06/03/94	11	20	8

Integration: Snap to Standard Grid and Merge

- Snap to the grid of the less-granular data.
- Reduced datasets merged into one.
- Accompanied by header to ensure integrity.

Latitude	Longitude	Time	Precip_in
13	Latitude	Longitude	Time	Precip_in
13	12	20	12/10/93	2
13	12	20	12/17/93	1
13	12	20	12/24/93	2
	12	20	12/31/93	8

Candidate A: reduced

Latitude	Longitude	Time	Precip_in
11	20	12/10/93	0
11	20	12/17/93	2
11	20	12/24/93	3
11	20	12/31/93	6

Candidate B: reduced

Latitude	Longitude	Time	Precip_in
13	Latitude	Longitude	Time	Precip_in
13	12	Latitude	Longitude	Time	Precip_in
13	12	11	20	12/10/93	0
13	12	11	20	12/17/93	2
	12	11	20	12/24/93	3
		11	20	12/31/93	6

New Dataset

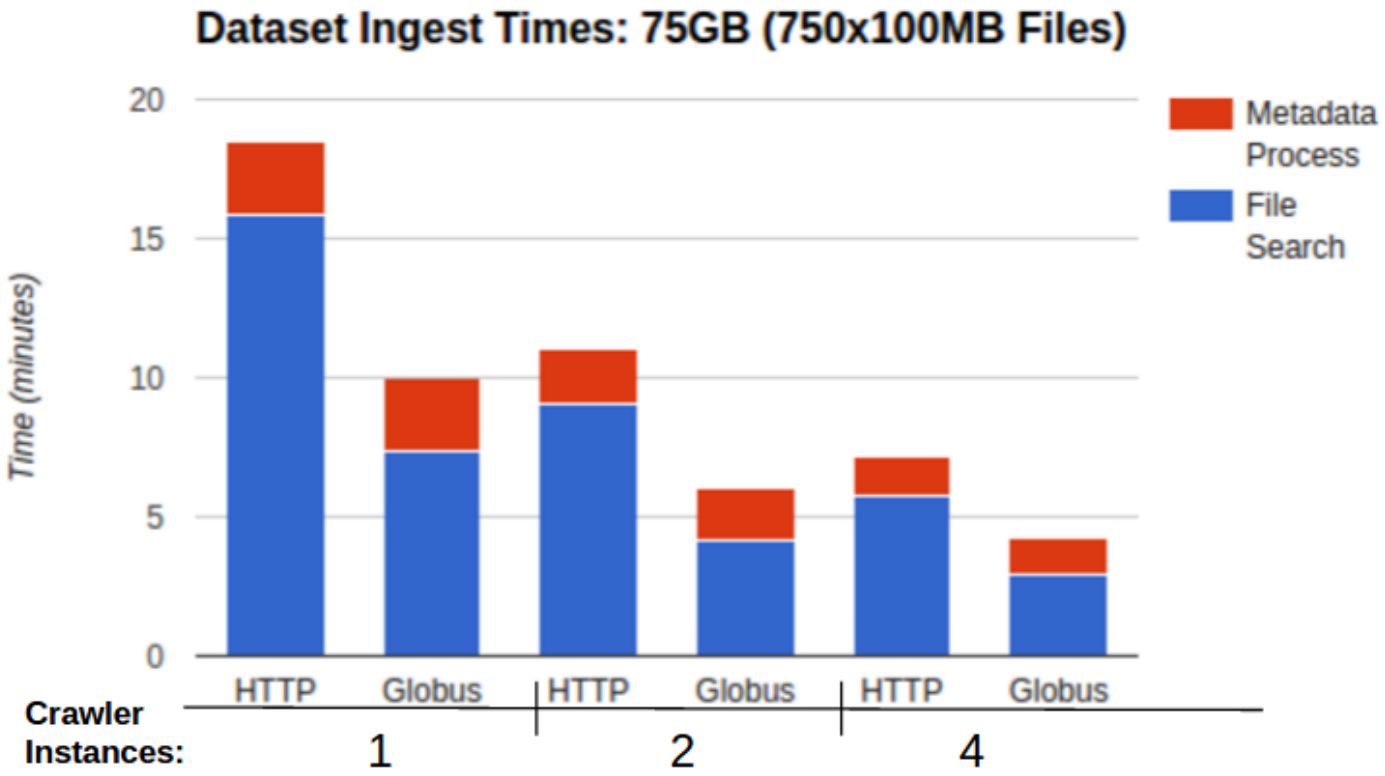
Origin
A
B
C
…
Distribution

Header.txt: Provenance-tracking

Evaluation

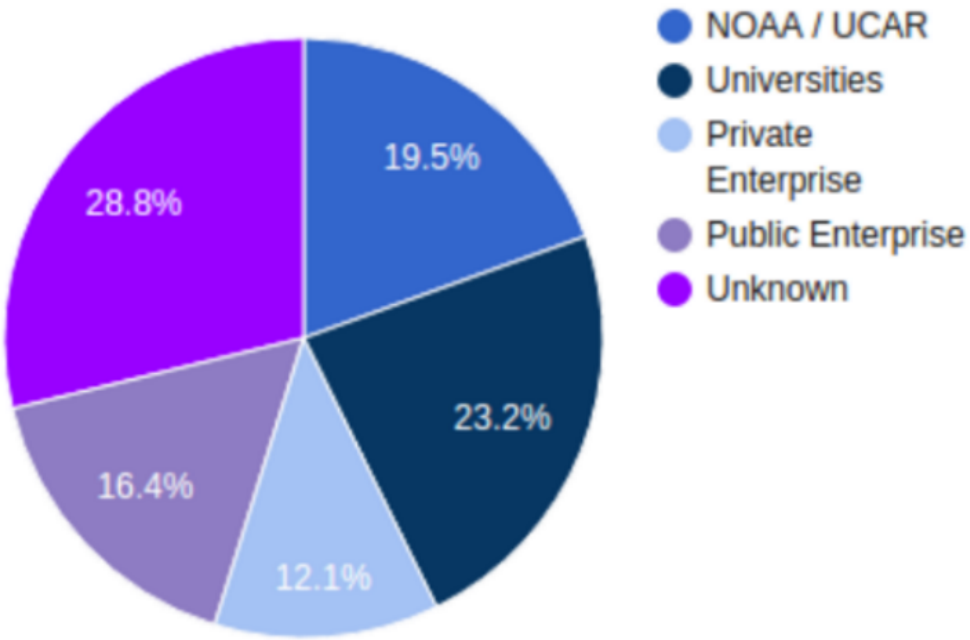
Preliminary Results: Evaluation

- Run in curated experimental sandbox with known access paths.
- 1-4 Docker containers instantiated in a single Linux 14.04 VM (16GB, 500GB).



Preliminary Results: Coverage

- 10,002 datasets extracted (∼11.5 TB).
- Every continent (included Anarctica) has at least 1,000 datasets.
- 20,000 world carbon data datasets ready for indexing (∼30,000 total).



Conclusion

Conclusions and Future Work

- Klimatic is an effective architecture for large scientific data.
- Container-model is scalable across containers across nodes.
- Robust coverage thus far.

Next Steps:

- Expand to other sciences' data needs (first up: materials science).
- Implementation of event-based update engine for Globus GridFTP.
- Add support for shapefiles (bounding-box becomes "bounding-shape")
- Classify data based on *content* and *context*.

Questions?