Old-School Cool - Utilizing SQL to Simplify and Optimize National Water Model Data Pipelines

Tyler Schrag[1], Corey Krewson[2], Fernando R. Salas[2]

1 NOAA Affiliate, Lynker, Geospatial Intelligence Division, Office of Water Prediction, National Water Center, Tuscaloosa, AL.

2 NOAA Federal, Geospatial Intelligence Division, Office of Water Prediction, National Water Center, Tuscaloosa, AL.

OWP | OFFICE OF WATER PREDICTION

NOAA

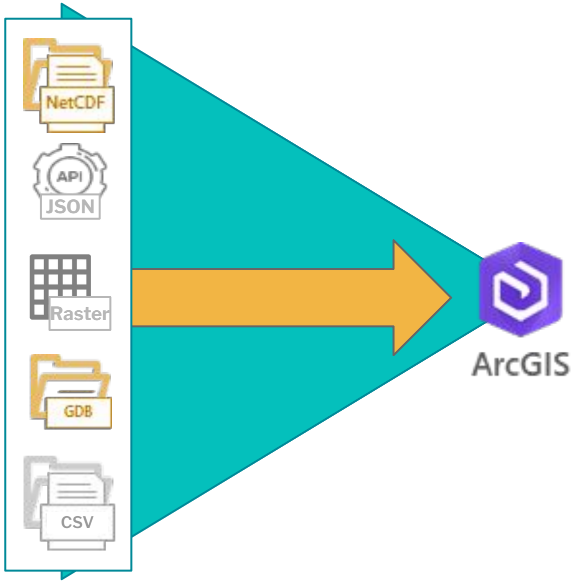
OWP Visualization Pipeline

~39 T Records

~34 TB Processed

~630 Pipelines

Per Day



~65 Map Services

3-10 M Total Rows

3-10 GB Total Size

Depending on Weather

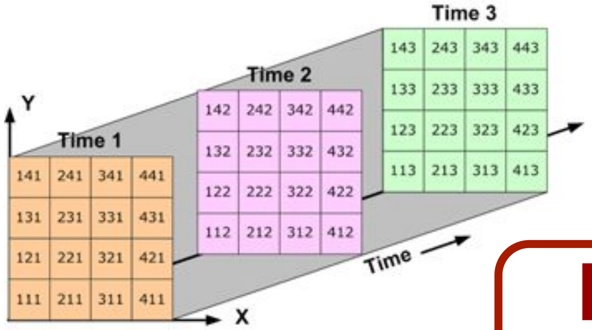
Primary Data Inputs

- National Water Model (NWM)
- Flood Inundation Mapping (FIM) Models
- Replace & Route (RnR) Models
- River Forecast Center Forecasts (AHPS)

Viz Pipeline - Cloud Design Challenge

How Best to Quickly ETL Multidimensional NetCDF Data?

Multi-Dimensional Data Model



Credit: ESRI

NetCDF

THREDDS | STAC ...

Python: NetCDF4, Xarray, Zarr, Dask ...

NEW SCHOOL

Two-Dimensional Data Model

X	Y	Time	Value
1	1	1	111
1	1	2	112
1	1	3	113
1	2	1	121
1	2	2	122
1	2	3	123
1	3	1	131
1	3	2	132
1	3	3	133
1	4	1	141
	...		

JSON

MongoDB | DynamoDB | Cassandra ...

NoSQL

OLD SCHOOL

Flat File (CSV)

RDBMS (MS SQL | PostgreSQL | Oracle ...)

SQL

Viz Pipeline - RDBMS Concerns

Is the 2D Data Too Big?

In some cases, maybe.

Is the Frequency of Pipeline Data Too Heavy on DB I/O?

With some data type tweaks, no.

Our SQL-Migrated Pipelines have seen:

~20 X Faster Processing

~85% Less Code

*Not quite an apples-to-apples comparison due to different environments.

Benefits of SQL

- **Relationships and intuitive queries**
  - **All of our data products use NWM Feature ID, NWS LID, and/or HUC keys.**

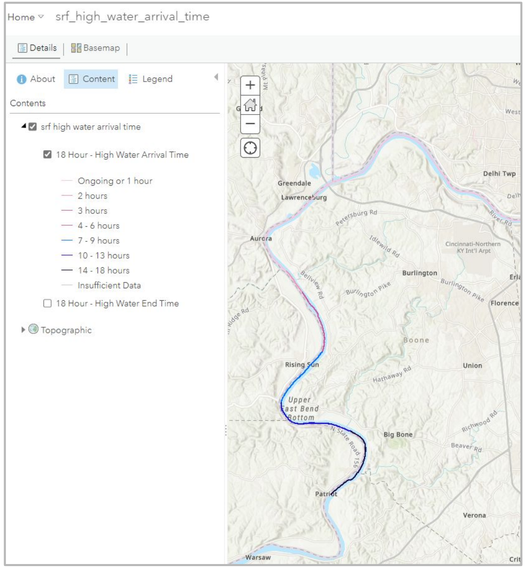
Query   Query History

```sql
SELECT
    forecasts.feature_id,
    min(forecast_hour) AS high_water_arrival_hour,
    max(streamflow) AS max_flow
FROM ingest.nwm_channel_rt_srf AS forecasts
JOIN derived.recurrence_flows_conus AS thresholds ON forecasts.feature_id = thresholds.feature_id
WHERE streamflow >= high_water_threshold
GROUP BY forecasts.feature_id
```

Data output   Messages   Notifications

	feature_id integer	high_water_arrival_hour integer	max_flow double precision
1	12776	1	0.01
2	12918	1	0.01
3	12934	1	0.01
4	83474	1	0.02
5	83726	7	3.64

~ 5-10 seconds



High Water Arrival Time

Benefits of SQL

- **Modular code and easy to test.**

Query | Query History

```sql
WITH floodstart AS (
    -- Add a rank value by feature_id, so we can look up info for the flood start later with 'rnk = 1'
    WITH pct_change_by_hour AS
        (
        WITH forecasts_full AS
            (
            WITH series AS -- Calculate a full 18 hour series for every feature_id, so that unadjacent hours aren't compared
                (SELECT channels.feature_id, generate_series(1,18,1) AS forecast_hour
                 FROM derived.channels_conus channels JOIN cache.max_flows_srf as mf on channels.feature_id = mf.feature_id
                 WHERE channels.strm_order <= 4
                )
            SELECT series.feature_id, series.forecast_hour, CASE WHEN streamflow is NOT NULL THEN (streamflow * 35.315) ELSE 0.001 END AS streamflow -- Set streamflow to 0.01 in cases where it is missing, so we don't get a
            FROM series
            LEFT OUTER JOIN ingest.nwm_channel_rt_srf AS forecasts ON series.feature_id = forecasts.feature_id AND series.forecast_hour = forecasts.forecast_hour -- Left outer join to the forecasts table (so that all hours
            ORDER BY forecasts.feature_id, series.forecast_hour
            )
        SELECT feature_id, forecast_hour, streamflow AS flow,
        ( -- Use the lag funtion to calucate percent change for each reach / forecast hour timestep
            (streamflow) - (lag(streamflow, 1) OVER (PARTITION BY feature_id ORDER BY forecast_hour)))/ --Numerator: current streamflow - last hour streamflow
            (lag(streamflow, 1) OVER (PARTITION BY feature_id ORDER by forecast_hour) -- Denominator: last hour streamflow
        ) AS pct_chg
        FROM forecasts_full
        )
    SELECT *, rank() OVER (PARTITION BY feature_id ORDER BY forecast_hour) as rnk
    FROM pct_change_by_hour
    WHERE pct_chg >= 1)
-- Select the forecast related fields for the attribute table
SELECT
    forecasts.feature_id,
    min(forecasts.forecast_hour) AS flood_start_hour,
    max(forecasts.forecast_hour) AS flood_end_hour,
    max(forecasts.forecast_hour) - min(forecasts.forecast_hour) AS flood_length,
    min(floodstart.flow) AS flood_flow,
    min(floodstart.pct_chg) AS flood_percent_increase,
    max(high_water_threshold) AS high_water_threshold
FROM ingest.nwm_channel_rt_srf AS forecasts
```

Data output | Messages | Notifications

	feature_id [PK] integer	forecast_hour integer	flow double precision	pct_chg double precision	rnk bigint
1	857	16	0.7062999999999999	705.3	1
2	865	11	0.35314999999999996	352.15	1
3	893	4	3.8846499999999997	2.6666666666666665	1
4	893	11	2.8251999999999997	3.0000000000000004	2
5	937	5	1.4125999999999999	1	1

~ 30 seconds

Rapid Onset Flooding

Benefits of SQL

- ~~ArcPy~~ PostGIS for basic spatial joins and geospatial operations

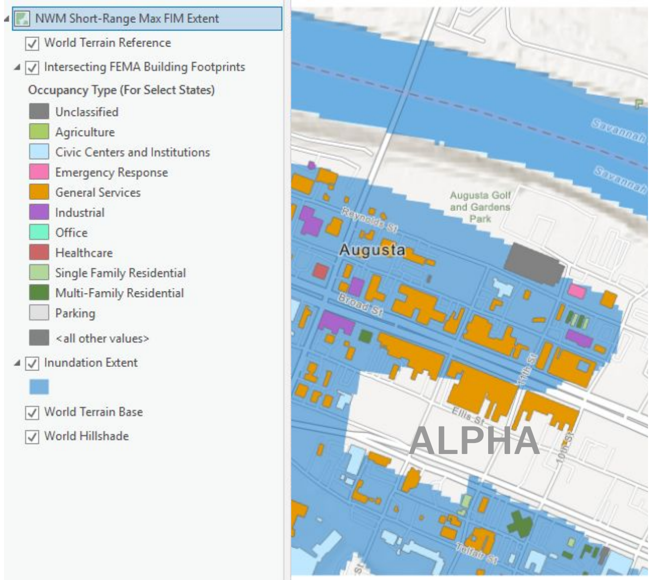
Query | Query History

```sql
SELECT
    ROW_NUMBER() OVER (ORDER BY uuid)::integer AS oid,
    buildings.build_id,
    buildings.occ_cls,
    buildings.prim_occ,
    buildings.prop_st,
    buildings.sqfeet,
    buildings.height,
    buildings.prod_date,
    buildings.source,
    buildings.val_method,
    fim.hydro_id,
    buildings.geom
FROM external.building_footprints_fema AS buildings
JOIN publish.ana_inundation as fim ON ST_Intersects(fim.geom, buildings.geom);
```

Data output | Messages | Notifications

	oid integer	build_id bigint	occ_cls text	prim_occ text	prop_st text	sqfeet double precision	height double precision	prod_date text	source text	val_method text	hydro_id integer	geom geometry
1	1	31895963	Residential	Single Fa...	FLORIDA	1306.33703613281	3.20000004768371	2021-07-2...	NGA	Unverified	12380088	0106000020110...
2	2	30367048	Residential	Single Fa...	FLORIDA	2461.64038085937	3.09999990463256	2021-07-2...	NGA	Unverified	12500487	0106000020110...
3	3	30419399	Residential	Multi - Fa...	FLORIDA	1029.88232421875	5.26999998092651	2021-07-2...	NGA	Unverified	12500119	0106000020110...
4	4	30369229	Residential	Single Fa...	FLORIDA	3920.41357421875	3.23000001907348	2021-07-2...	NGA	Unverified	12500489	0106000020110...
5	5	31900210	Residential	Single Fa...	FLORIDA	3360.18481445312	4.80000019073486	2021-07-2...	NGA	Unverified	12380088	0106000020110...
6	6	30202623	Residential	Single Fa...	FLORIDA	2631.07641601562	3.19000005722045	2021-07-2...	NGA	Unverified	12500460	0106000020110...

~ 20 seconds



Inundation FIMpact Building Footprints

Benefits of SQL

- **Foundation for more robust visualization, intelligence & analytics**

Query | Query History

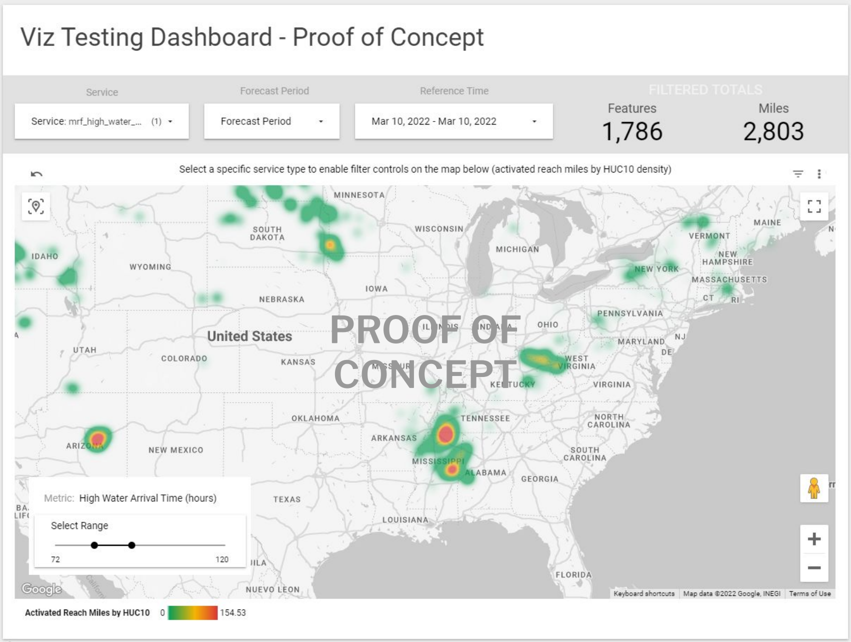
```sql
SELECT
    region_name,
    count(ana.feature_id) AS flooded_features_now,
    count(srf.feature_id) AS flooded_features_18hour,
    count(mrf.feature_id) AS flooded_features_10day
FROM derived.huc2_rf_thresholds AS huc2
JOIN derived.featureid_huc_crosswalk AS xwalk ON huc2.huc2 = xwalk.huc2
LEFT OUTER JOIN publish.ana_high_flow_magnitude ana ON xwalk.feature_id = ana.feature_id
LEFT OUTER JOIN publish.srf_max_high_flow_magnitude srf ON xwalk.feature_id = srf.feature_id
LEFT OUTER JOIN publish.mrf_max_high_flow_magnitude mrf ON xwalk.feature_id = mrf.feature_id
GROUP BY region_name
```

Data output | Messages | Notifications

	region_name text	flooded_features_now bigint	flooded_features_18hour bigint	flooded_features_10day bigint
1	Arkansas-White-Red River Basins	645	682	924
2	California Region	658	968	1215
3	Great Basin	270	270	298
4	Great Lakes Region	4343	4615	7616
5	Lower Colorado River Basin	5833	5839	5950
6	Lower Mississippi River Basin	1384	1484	1941
7	Mid-Atlantic Region	1790	1848	3563
8	Missouri River Basin	3073	3078	3141

~ 1 second

Easy Summaries



Google Data Studio

Thank You!

Tyler Schrag

tyler.schrag@noaa.gov

https://water.noaa.gov

OWP OFFICE OF WATER PREDICTION

Lynker

Looking for more OWP presentations at AGU?

Scan the QR code to the right to see a full schedule of events!