APACHECON North America
Montréal, Canada
24-27 September, 2018

MAPD

Interacting with Billions of National Water Model (NWM) Predictions using Apache Kafka and Arrow with MapD

ApacheCon | Montreal | September 26, 2018

slides: https://speakerdeck.com/mapd

MAPD

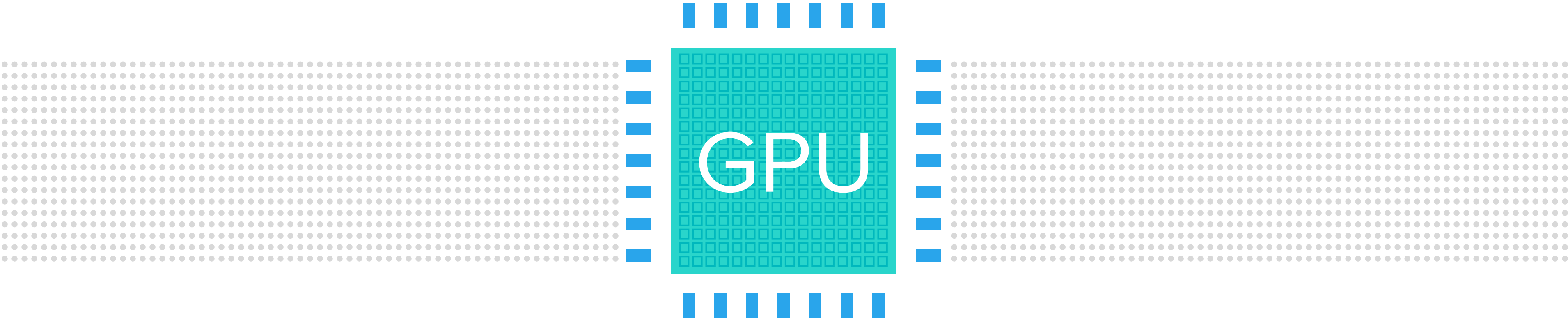
Benjamin Lewis
Geospatial Technology Manager at Harvard
Center for Geographic Analysis (CGA)

@bgreenelewis
blewis@cga.harvard.edu
/in/benjamin-lewis-9844207/
/blewis

Aaron Williams
VP of Global Community at MapD

@_arw_
aaron@mapd.com
/in/aaronwilliams/
/williamsaaron

Center for Geographic Analysis
Harvard University



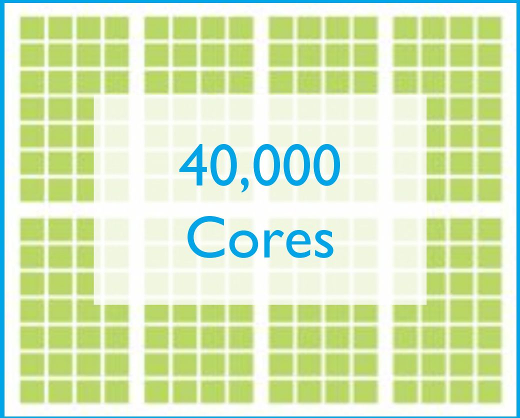
HARNESS GPUs

The Fastest Software
Designed for the Fastest Hardware

CPU Processing

20 Cores

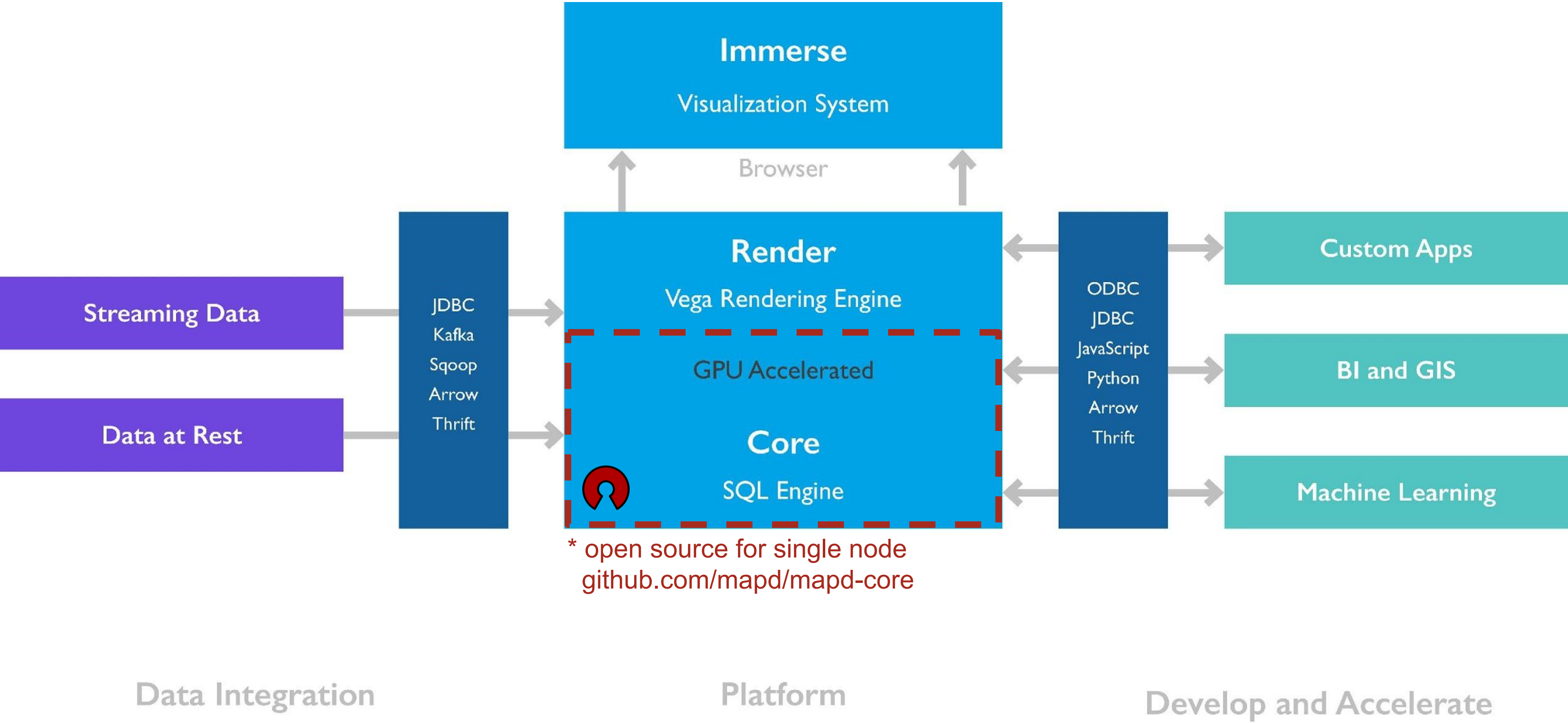
GPU Processing



	Latency	**Throughput**
CPU	1 ns per task	(1 task/ns) x (20 cores) = 20 tasks/ns
GPU	10 ns per task	(0.1 task per ns) x (40,000 cores) = 4,000 task per ns

Latency: Time to do a task. | Throughput: Number of tasks per unit time.

*fictitious example


Immerse
Visualization System
Browser
Render
Vega Rendering Engine
GPU Accelerated
Core
SQL Engine
Streaming Data
Data at Rest
JDBC
Kafka
Sqoop
Arrow
Thrift
ODBC
JDBC
JavaScript
Python
Arrow
Thrift
Custom Apps
BI and GIS
Machine Learning
* open source for single node
github.com/mapd/mapd-core
Data Integration
Platform
Develop and Accelerate

MAPD DEMO

https://www.mapd.com/demos/

Advanced memory management

Three-tier caching to GPU RAM for speed and to SSDs for persistent storage

Hot Data
Speedup = 1500x to 5000x
Over Cold Data

GPU RAM (L1)
24GB to 256GB
1000-6000 GB/sec

COMPUTE LAYER

Warm Data
Speedup = 35x to 120x
Over Cold Data

CPU RAM (L2)
32GB to 3TB
70-120 GB/sec

Cold Data

STORAGE LAYER

SSD or NVRAM STORAGE (L3)
250GB to 20TB
1-2 GB/sec

Data Lake/Data Warehouse/System Of Record

MapD Core: Query Compilation with LLVM

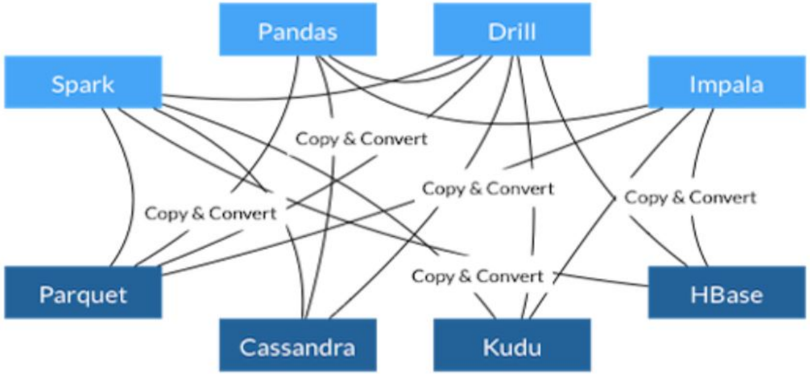
Traditional DBs can be highly inefficient

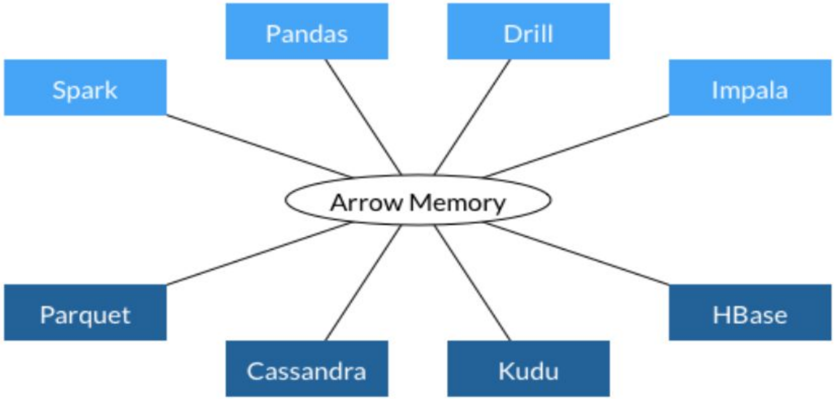
- each operator in SQL treated as a separate function
- incurs tremendous overhead and prevents vectorization

MapD compiles queries w/LLVM to create one custom function

- Queries run at speeds approaching hand-written functions
- LLVM enables generic targeting of different architectures (GPUs, X86, ARM, etc).
- Code can be generated to run query on CPU and GPU simultaneously

APACHE ARROW





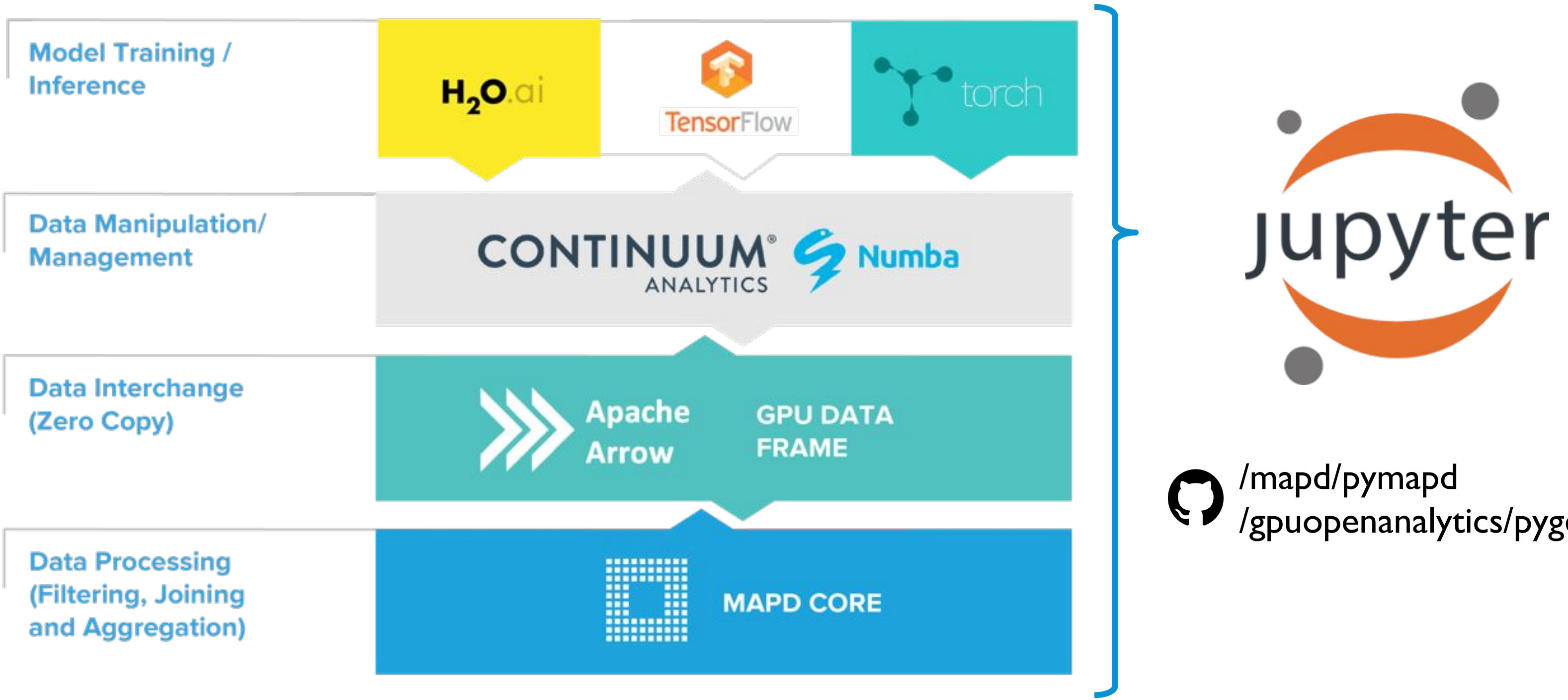
- Each system has its own internal memory format
- 70-80% computation wasted on serialization and deserialization
- Similar functionality implemented in multiple projects

- All systems utilize the same memory format
- No overhead for cross-system communication
- Projects can share functionality (eg, Parquet-to-Arrow reader)

Source: https://arrow.apache.org/

The GPU Open Analytics Initiative (GOAI)

Creating common data frameworks to accelerate data science on GPUs



/mapd/pymapd
/gpuopenanalytics/pygdf

ML Examples

We've published a few notebooks showing how to connect to a MapD database and use an ML algorithm to make predictions

/gpuopenanalytics/demo-docker

We also have notebooks from an example we created with Volkswagen

/mapd/mapd-ml-demo

/watch?v=SOXdRUKUWoE

README.md

mapd-ml-demo

Can be run with nvidia-docker-compose. This depends on two containers:

Name	Use	Dockerfile location
`mapd-core`	MapD Database	Defaults to Community Edition on Docker Hub `mapd/mapd-ce-cuda`
`mapd/ml`	Demo notebooks	Dockerfile in top-level of `mapd-ml-demo` repo

Build

mapd-ml-demo Server

To build the `mapd-ml-demo` container, clone the repo and `cd` into it.

To build the container, run:

```
docker build -t mapd/ml .
```

Exporting

If you need to move the containers to a new machine, run:

```
docker save -o mapd-ce-cuda.tar mapd/mapd-ce-cuda
docker save -o mapd-ml.tar mapd/ml

gzip mapd-ce-cuda.tar
gzip mapd-iml.tar
```

You will then have files which can be moved to the new machine: `mapd-ce-cuda.tar.gz`, `mapd-ml.tar.gz`. You will also want to grab the `docker-compose.yml` file (but probably not the `nvidia-docker-compose.yml` one).

Run

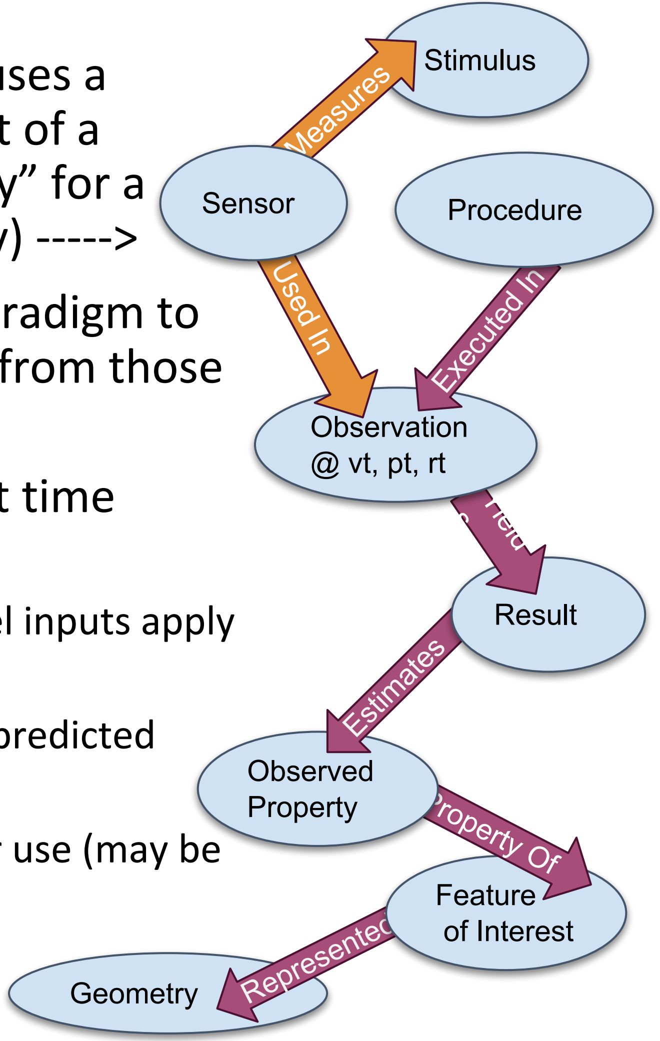
Interacting with National Water Model Predictions

“Big” geospatial data: Not just the number of features

- Most sets of geographic features are modest: thousands to millions in size. But...
- Increasing spatial resolution is changing this: e.g. National Hydro Datasets Medium Res -> ~3M reaches, High Res -> ~30M reaches. Similar for gridded data, e.g. 10m DEM -> 1m Lidar-based 3DEP increases volume 100x.
- Time is changing this: multiple observations and predictions for multiple feature properties quickly combine into billions of records.
- Traditional GIS software struggles to access and visualize, let alone analyze such scales and structures of datasets.
- Datasets with 1-100 billion records are becoming common in academic, business, and government domains.
- Traditional GIS data model of 1 feature + 1 geometry + n attributes is increasingly inadequate to large-scale observations & prediction

Model Simulation and Prediction Data

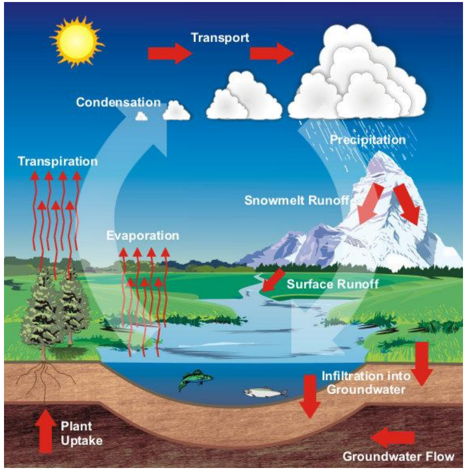
- Most observation data is already “model-based” and uses a computational “procedure” to convert a measurement of a “stimulus” into an estimation of an “observed property” for a “feature of interest” (O&M model, SOSA/SSN ontology) ----->
- Simulation and prediction models extend this same paradigm to generate properties at places and/or times that differ from those at which measurements are made.
- Model outputs are characterized by at least 3 different time senses:
   a. Valid Time - the time or time interval within which the model inputs apply and the output is therefore valid.
   b. Phenomenon Time - the time of the observed / simulated / predicted property estimate
   c. Result Time - the time at which model output is available for use (may be some time after the Valid Time for lengthy computations).



High-performance model interpretation

- Computing needs are at the scale of the volume, velocity, variety, verisimilitude of the model output and other data to be processed, juxtaposed, or compared.
- Needs may also vary according to the specific hypotheses to be tested, methods to be employed, and the number of interpreters working with a given model output.
- Parallel computing can address volume but may not produce the throughput to support interactive interpretation nor be cost effective to scale for many users.
- GPU-based computing can increase throughput through efficient “parallelism in place” with fast execution of certain operations on thousands of inexpensive processor cores, if the data fit into GPU memory.
- Specific computational components assembled into tool chains provide flexibility for evolving model analysis and visualization needs.

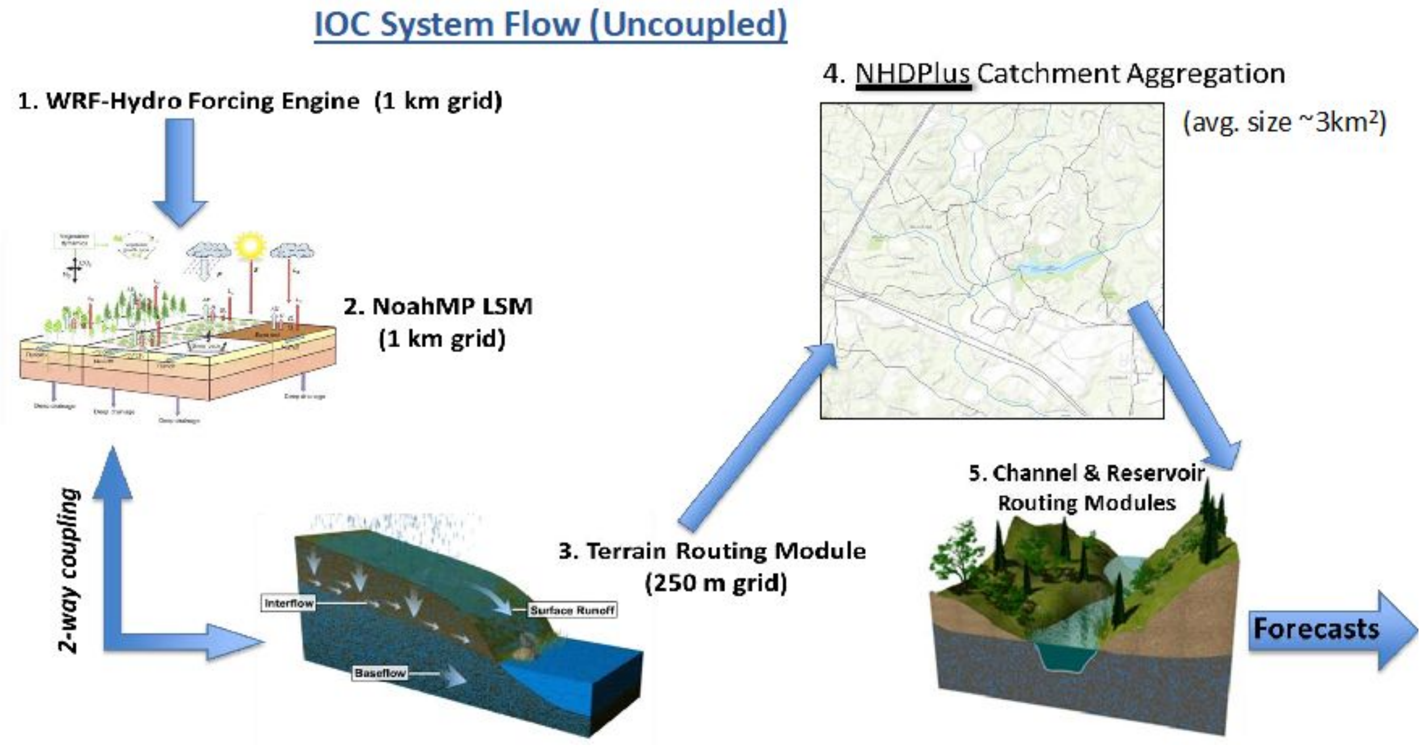
The National Water Model



- U.S. National Water Model (NWM) models run up to hourly on a Cray XC40 supercomputer.
- Input data from ~3600 river / reservoir gauges, along with weather model outputs and other data sources (forcing), generates predictions (present, 0-18-hr, 0-10-day, or 0-30-day) of hydrologic conditions
- Predictions for 2.7 million stream reaches, 1260 reservoirs, and~300M surface grid points across the U.S. (1km & 250m spacings).
- NWM outputs ~90gb / day (1gb present conditions, 18gb shortrange, 65gb / day medium range, ~4gb / day long range).
- A viewer is available for pre-generated images of present model output and another for pre-generated grouped streamflow features.

WRF-Hydro Model

- A community-based hydrologic modeling framework supported by NCAR
- Not dependent on a particular forcing data source or choice of LSM
- Able to operate over multiple scales and with multiple physics options



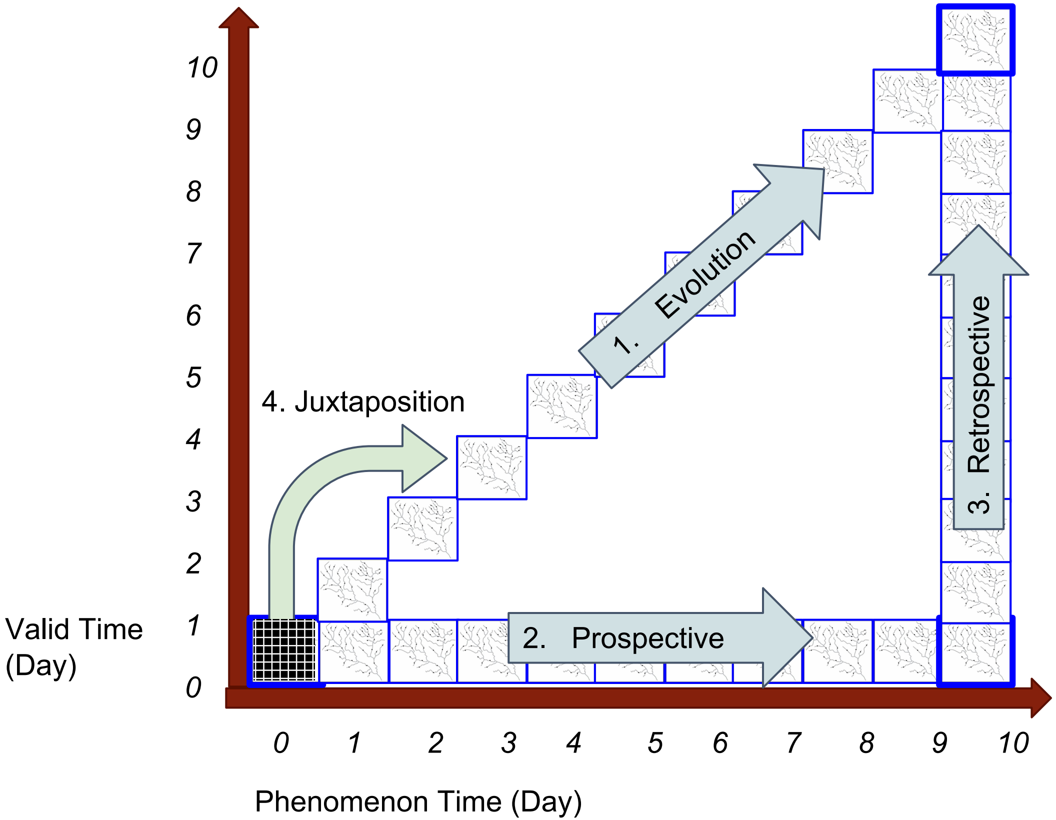
Data Flow from NWM to MapD

- Harvesting
  - NWM output files in NetCDF format downloaded from website.
- Storage
  - Initially: 2-month test dataset at 6-hour intervals of present and predicted conditions.
  - Presently: rolling 1-2 month time window drawn from Kafka streams defined on the NWM datafiles.
- Preprocessing / loading to MapD
  - NetCDF -> Xarray -> Pandas Dataframe -> PyMapD -> MapD table
  - Geometric coordinates stored separately from model output parameters
- Limitations
  - Data are initially loaded onto disk, then column-wise into (limited) CPU and/or GPU memory for query and/or rendering (1 K-80 GPU -> ~11gb data memory).

Initial work with NWM and MapD

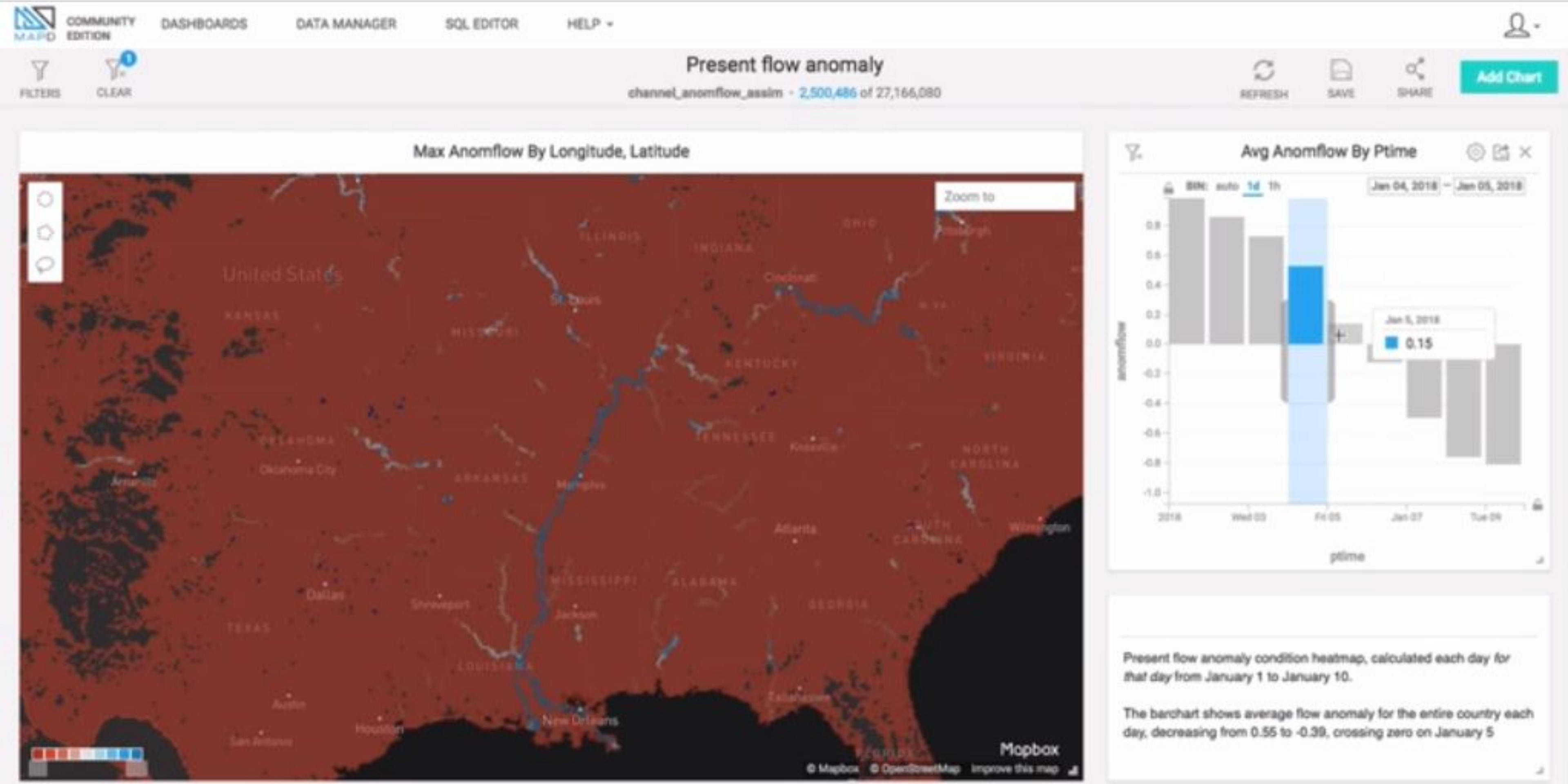
- Develop data download, storage, database loading procedures
- Configure and install MapD on 1-GPU EC2 instances
- Load nationwide stream reach mid-points and 1km (down-sampled from 250m to fit in GPU memory) grid point geometries to MapD store.
- Load stream flow / velocity and soil inundation / saturation outputs for once-daily present conditions and 1-10 day predictions over a 10-day period at the beginning of 2018. Learn virtues of pandas dataframe.
- Construct SQL views to join stream and point locations with model output values. Discover that (some) views work differently than equivalent queries in MapD.
- Develop dashboard views in MapD Immerse client to visualize the results. Learn hidden tricks, GPU utilities, and undocumented configuration switches.

Time perspectives in NWM output data



1. Evolution of present conditions over 10 days
2. Prospective prediction over 10 days from start of period
3. Retrospective evolution of 10th day predictions from 0-day prediction to present conditions
4. Juxtaposition and time offset for influence of surface / subsurface flow routing on nearby river inputs and flow (not shown in this presentation)

Demo



COMMUNITY EDITION
DASHBOARDS
DATA MANAGER
SQL EDITOR
HELP
FILTERS
CLEAR
Present flow anomaly
channel_anomflow_assim · 2,500,486 of 27,166,080
REFRESH
SAVE
SHARE
Add Chart
Max Anomflow By Longitude, Latitude
Zoom to
Mapbox
Avg Anomflow By Ptime
BIN: auto 1d 1h
Jan 04, 2018 – Jan 05, 2018
0.15
anomflow
ptime
Present flow anomaly condition heatmap, calculated each day *for that day* from January 1 to January 10.
The barchart shows average flow anomaly for the entire country each day, decreasing from 0.55 to -0.39, crossing zero on January 5

Additional Work with NWM and MapD

- Work with stream line and watershed geometries.
- Integrate additional parameters such as precipitation forcing data
- Juxtapose additional critical features such as roads and bridges to connect model predictions with emergency response planning
- Develop custom applications for interactive interpretation, model validation, and decision support using NWM outputs

Current Activity

Geospatial Data Types and Functions

- Points, Lines, Polygons
- Distance, Point-in-polygon operators

LIDAR Data

- Initial dataset from the state of Florida with several billion points
- Input to future capabilities for images and other coverage datasets

Open Source MapD Core

- Hosted on Github
- Opportunity for extended capabilities in processing and rendering through GoAI in-memory IPC
- Illustrates need for standard geospatial datatypes for big data ecosystem tools such as Apache Arrow

Conclusions

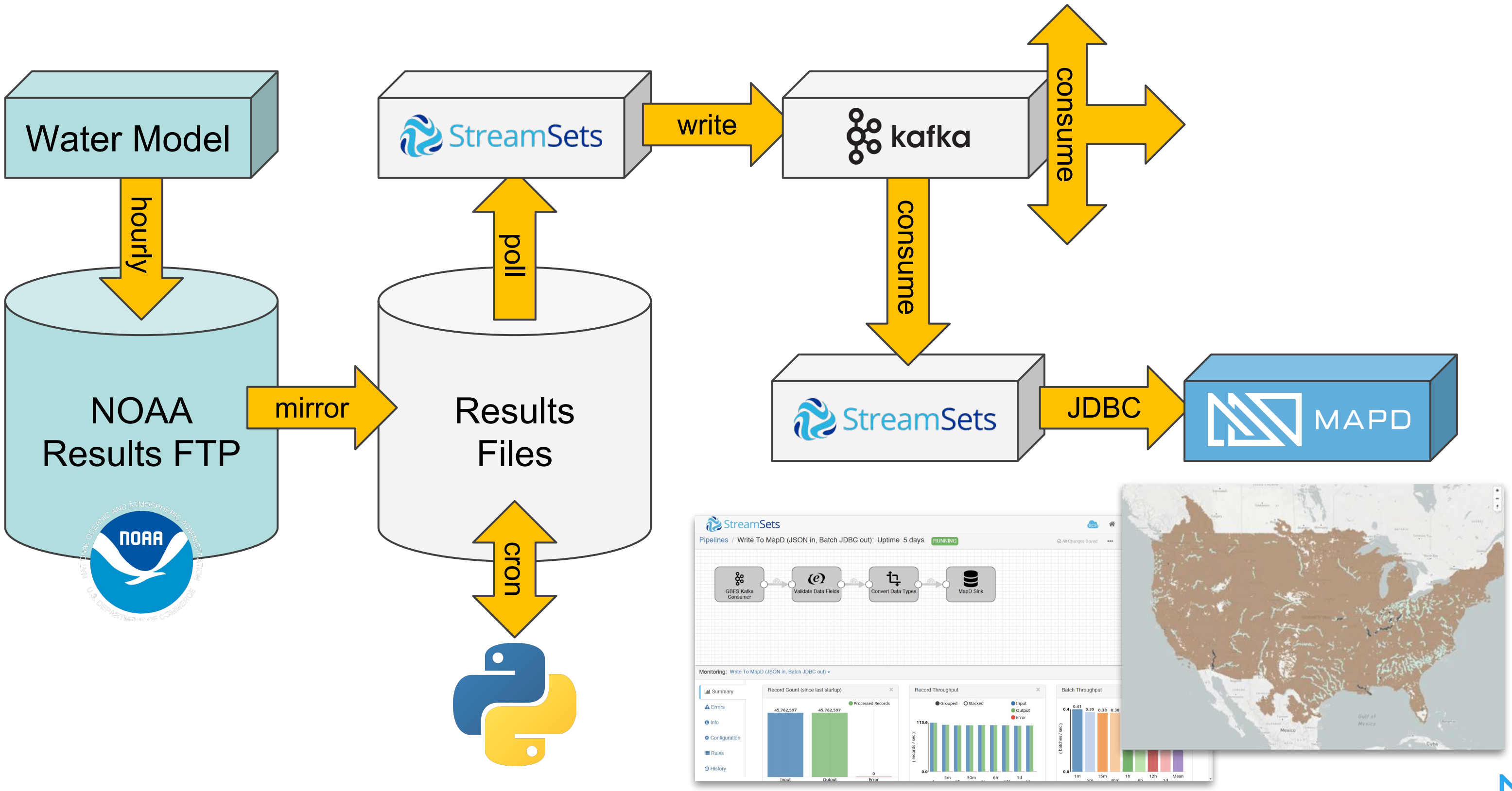
- Interpretation of simulation / prediction model outputs for geographic entities can be a big data challenge that is both significant and separate from that of running the models. Without adequate tools to interpret this scale of data, the usefulness of creating and running the models themselves is reduced.
- GPU-based data analysis and visualization tools such as MapD offer good possibilities for addressing this challenge with fast data interaction, cost effective deployment, and flexible integration with other tools.
- DBMS’ such as MapD still require significant expertise to use effectively when “pushing the envelope” on new capabilities.
- CGA has learned much already from working with MapD and NWM model outputs and plan to apply this to other use cases and domains.

Links

- More detail on the Project Wiki https://github.com/cga-harvard/HPC_on_MOC/wiki
- MapD Core code https://github.com/mapd/mapd-core
- Collaboration announcements http://gis.harvard.edu/announcements/renewed-collaboration-between-cga-and-mapd-accelerate-research-gpus

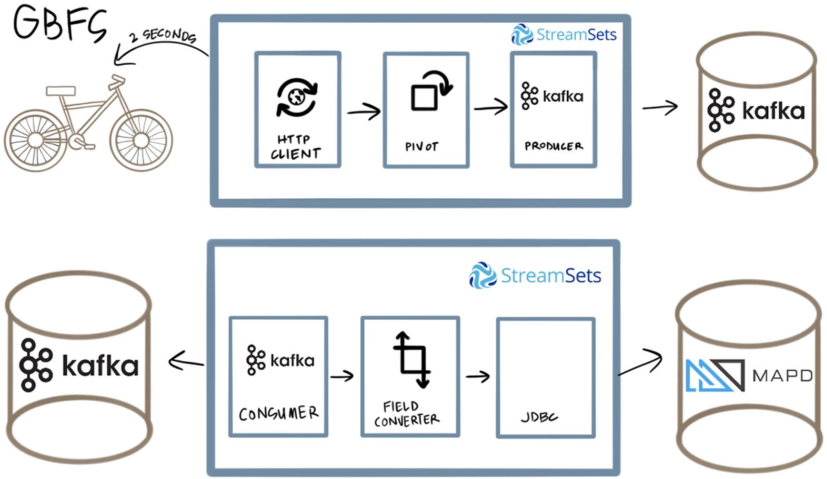
Apache Kafka Streaming

Adding Apache Kafka Streaming



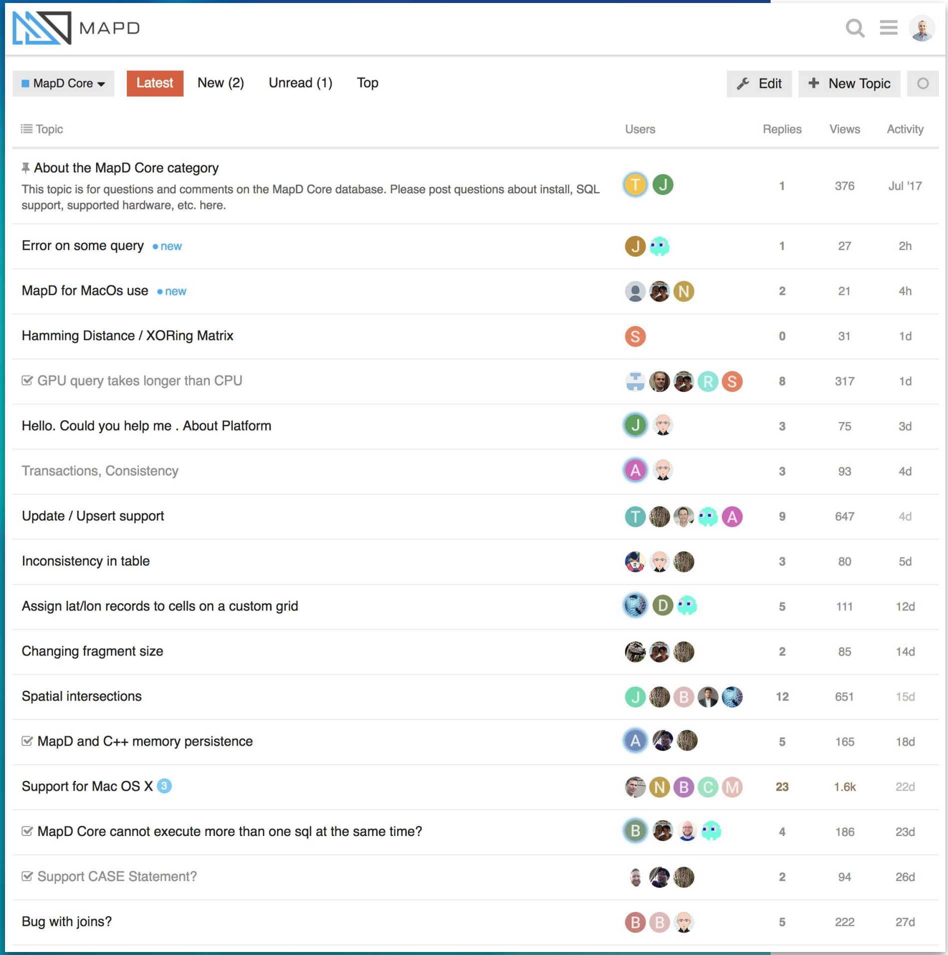
Another Kafka Example

kafka



https://www.jowanza.com/blog/2018/9/8/real-time-station-tracking-ford-gobike-and-mapd

twitter: @jowanza



Next Steps

- **mapd.com/demos**
  Play with our demos
- **mapd.cloud**
  Get a MapD instance in less than 60 seconds
- **mapd.com/platform/downloads/**
  Download the Community Edition
- **community.mapd.com**
  Ask questions and share your experiences

Thank you! Questions?

Benjamin Lewis
Geospatial Technology Manager at Harvard
Center for Geographic Analysis (CGA)

@bgreenelewis
blewis@cga.harvard.edu
/in/benjamin-lewis-9844207/
/blewis

Aaron Williams
VP of Global Community at MapD

@_arw_
aaron@mapd.com
/in/aaronwilliams/
/williamsaaron

Center for Geographic Analysis
Harvard University