Pivotal

BUILT FOR THE SPEED OF BUSINESS

Massively Parallel Processing with Procedural Python

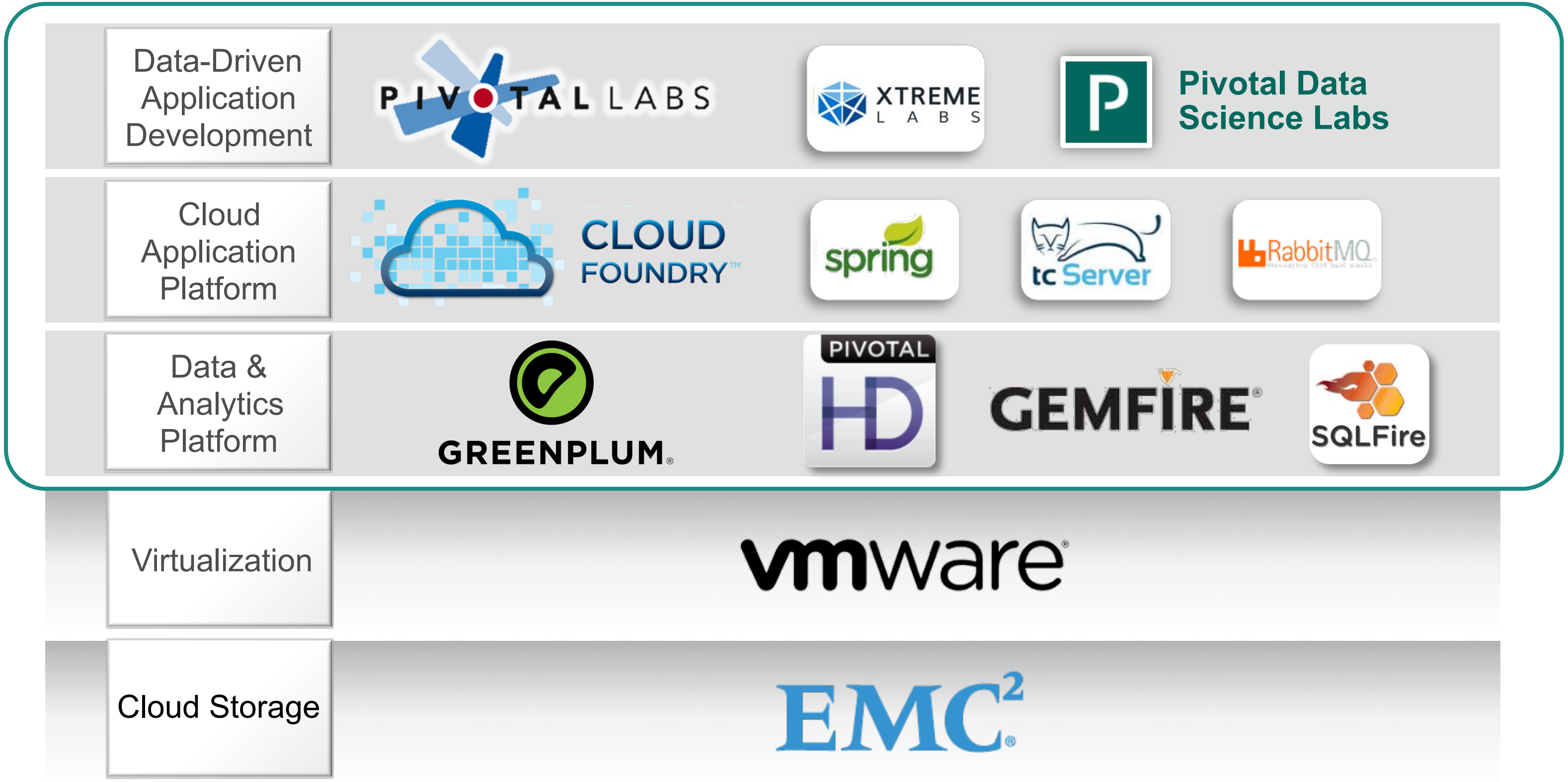
How do we use the PyData stack in data science engagements at Pivotal?

Ian Huston, @ianhuston
Data Scientist, Pivotal

Some Links for this talk

- Simple code examples: https://github.com/ihuston/plpython_examples
- IPython notebook rendered with nbviewer: http://tinyurl.com/ih-plpython
- More info (written for PL/R but applies to PL/Python): http://gopivotal.github.io/gp-r/
- Traffic Disruption demo (if we have time) http://ds-demo-transport.cfapps.io

About Pivotal



Data-Driven Application Development

Cloud Application Platform

Data & Analytics Platform

Virtualization

Cloud Storage

What do our customers look like?

- Large enterprises with lots of data collected
  - Work with 10s of TBs to PBs of data, structured & unstructured
- Not able to get what they want out of their data
  - Old Legacy systems with high cost and no flexibility
  - Response times are too slow for interactive data analysis
  - Can only deal with small samples of data locally
- They want to transform into data driven enterprises

NYSE · verizon · BlueCross BlueShield · SILVERSPRING NETWORKS · UNITEDHEALTH GROUP · WELLPOINT

Fidelity INVESTMENTS · T··Mobile· · Sears Home Services · CareCore NATIONAL · GENSCAPE · comSCORE · HAVAS DIGITAL · ORBITZ · SIEMENS

Open Source is Pivotal

spring
redis
Apache Tomcat
MADlib
CLOUD FOUNDRY™
RabbitMQ™
GRAILS
Groovy
hadoop
Open Chorus Project

Pivotal's Open Source Contributions

Lots more interesting small projects:

- PyMADlib – Python Wrapper for MADlib
  https://github.com/gopivotal/pymadlib
- PivotalR – R wrapper for MADlib
  http://github.com/madlib-internal/PivotalR
- Part-of-speech tagger for Twitter via SQL
  http://vatsan.github.io/gp-ark-tweet-nlp/
- Pandas via psql
  (interactive PostgreSQL terminal)
  https://github.com/vatsan/pandas_via_psql

Typical Engagement Tech Setup

- Platform:
  - Greenplum Analytics Database (GPDB)
  - Pivotal HD Hadoop Distribution + HAWQ (SQL DB on Hadoop)
- Open Source Options (http://gopivotal.com):
  - Greenplum Community Edition
  - Pivotal HD Community Edition (HAWQ not included)
  - MADlib in-database machine learning library (http://madlib.net)
- Where Python fits in:
  - **PL/Python running in-database**, with nltk, scikit-learn etc
  - IPython for exploratory analysis
  - Pandas, Matplotlib etc.

PIVOTAL DATA SCIENCE TOOLKIT

1 Find Data

Platforms
- Greenplum DB
- Pivotal HD
- Hadoop (other)
- SAS HPA
- AWS

3 Run Code

Interfaces
- pgAdminIII
- psql
- psycopg2
- Terminal
- Cygwin
- Putty
- Winscp

2 Write Code

Editing Tools
- Vi/Vim
- Emacs
- Smultron
- TextWrangler
- Eclipse
- Notepad++
- IPython
- Sublime

Languages
- SQL
- Bash scripting
- C
- C++
- C#
- Java
- Python
- R

4 Write Code for Big Data

In-Database
- SQL
- PL/Python
- PL/Java
- PL/R
- PL/pgSQL

Hadoop
- HAWQ
- Pig
- Hive
- Java

5 Implement Algorithms

Libraries
- MADlib

Java
- Mahout

R
- (Too many to list!)

Text
- OpenNLP
- NLTK
- GPText

C++
- opencv

Python
- NumPy
- SciPy
- scikit-learn
- Pandas

Programs
- Alpine Miner
- Rstudio
- MATLAB
- SAS
- Stata

6 Show Results

Visualization
- python-matplotlib
- python-networkx
- D3.js
- Tableau
- GraphViz
- Gephi
- R (ggplot2, lattice, shiny)
- Excel

7 Collaborate

Sharing Tools
- Chorus
- Confluence
- Socialcast
- Github
- Google Drive & Hangouts

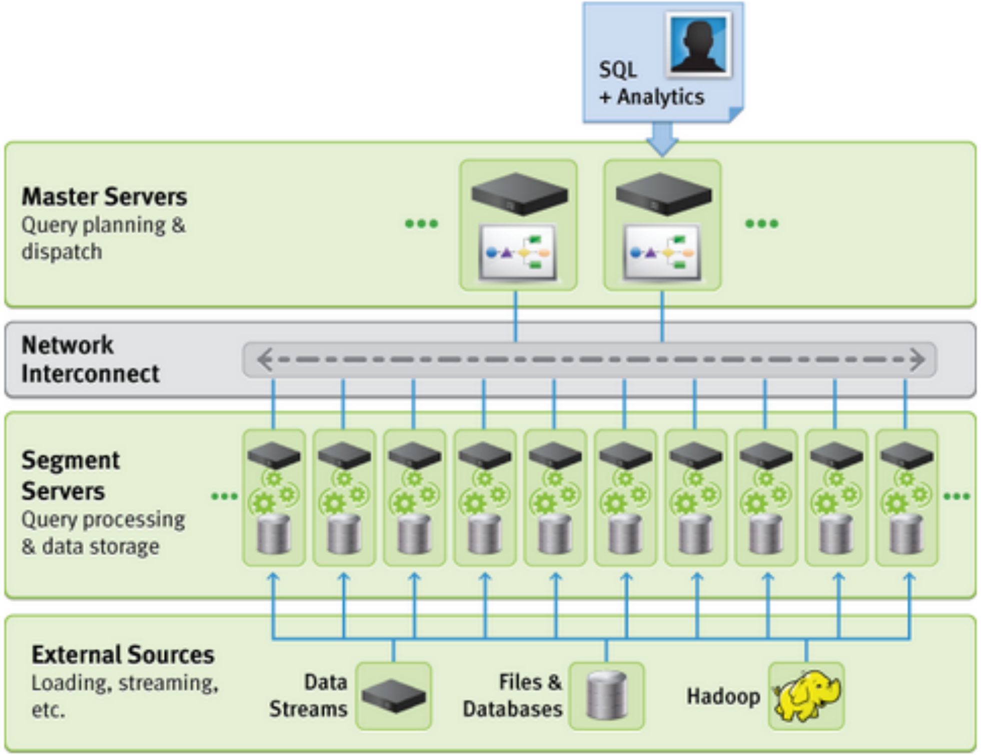
A large and varied tool box!

@ianhuston

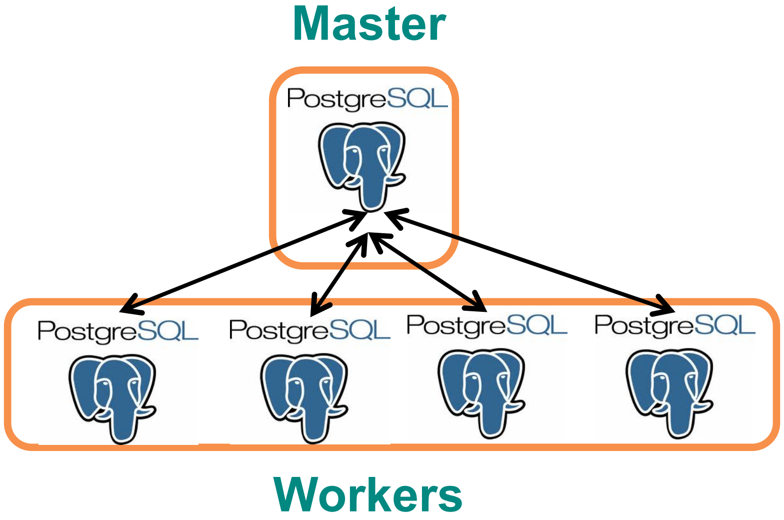
Pivotal

PL/Python

MPP Architectural Overview



Think of it as multiple PostGreSQL servers



Data Parallelism

- Little or no effort is required to break up the problem into a number of parallel tasks, and there exists no dependency (or communication) between those parallel tasks.
- Examples:
  - Measure the height of each student in a classroom (explicitly parallelizable by student)
  - MapReduce
  - map() function in Python

User-Defined Functions (UDFs)

- PostgreSQL/Greenplum provide lots of flexibility in defining your own functions.
- Simple UDFs are SQL queries with calling arguments and return types.

Definition:

```
CREATE FUNCTION times2(INT)
RETURNS INT
AS $$
    SELECT 2 * $1
$$ LANGUAGE sql;
```

Execution:

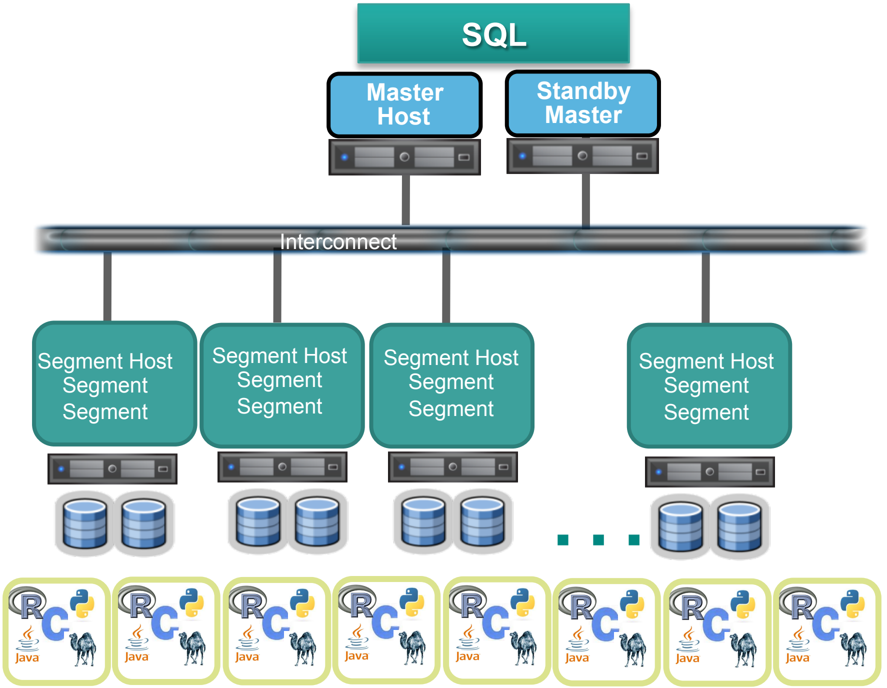
```
SELECT times2(1);
 times2
--------
      2
(1 row)
```

PL/X : X in {pgsql, R, Python, Java, Perl, C etc.}

- Allows users to write Greenplum/ PostgreSQL functions in the R/Python/ Java, Perl, pgsql or C languages

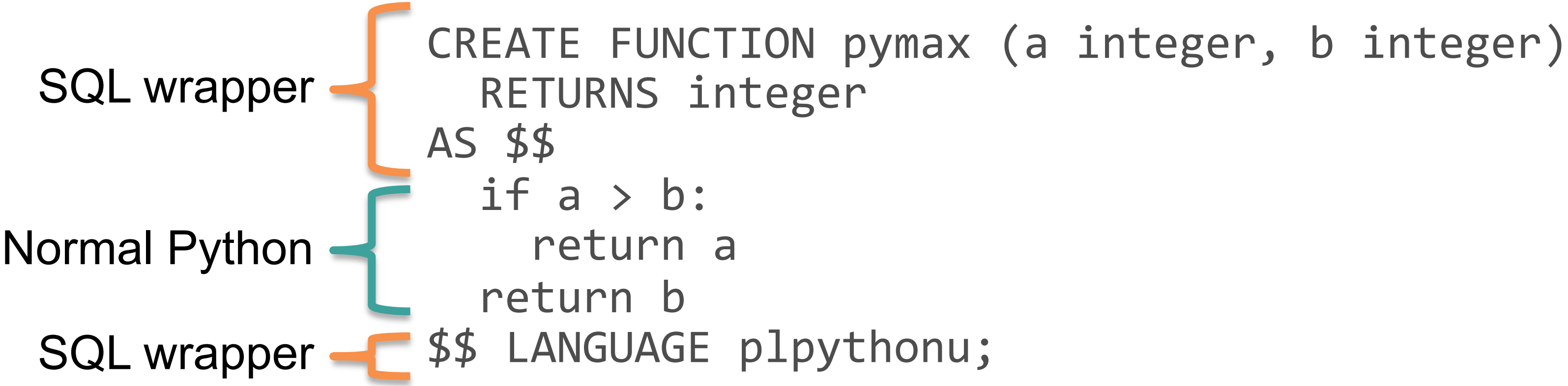
- The interpreter/VM of the language 'X' is installed on each node of the Greenplum Database Cluster

- Data Parallelism:
  - PL/X piggybacks on Greenplum's MPP architecture



Intro to PL/Python

- Procedural languages need to be installed on each database used.
- Name in SQL is plpythonu, ‘u’ means untrusted so need to be superuser to install.
- Syntax is like normal Python function with function definition line replaced by SQL wrapper. Alternatively like a SQL User Defined Function with Python inside.



SQL wrapper

Normal Python

SQL wrapper

```
CREATE FUNCTION pymax (a integer, b integer)
  RETURNS integer
AS $$
  if a > b:
    return a
  return b
$$ LANGUAGE plpythonu;
```

Examples

Returning Results

- Postgres primitive types (int, bigint, text, float8, double precision, date, NULL etc.)
- Composite types can be returned by creating a composite type in the database:

```
CREATE TYPE named_value AS (
  name  text,
  value  integer
);
```

- Then you can return a list, tuple or dict (not sets) which reference the same structure as the table:

```
CREATE FUNCTION make_pair (name text, value integer)
  RETURNS named_value
AS $$
  return [ name, value ]
  # or alternatively, as tuple: return ( name, value )
  # or as dict: return { "name": name, "value": value }
  # or as an object with attributes .name and .value
$$ LANGUAGE plpythonu;
```

- For functions which return multiple rows, prefix “`setof`” before the return type

Returning more results

You can return multiple results by wrapping them in a sequence (tuple, list or set), an iterator or a generator:

Sequence

```
CREATE FUNCTION make_pair (name text)
  RETURNS SETOF named_value
AS $$
  return ([ name, 1 ], [ name, 2 ], [ name, 3])
$$ LANGUAGE plpythonu;
```

Generator

```
CREATE FUNCTION make_pair (name text)
  RETURNS SETOF named_value  AS $$
  for i in range(3):
      yield (name, i)
$$ LANGUAGE plpythonu;
```

Accessing Packages

- On Greenplum DB: To be available packages must be installed on the individual segment nodes.
  - Can use “parallel ssh” tool `gpssh` to conda/pip install
  - Currently Greenplum DB ships with Python 2.6 (!)

- Then just import as usual inside function:

```
CREATE FUNCTION make_pair (name text)
  RETURNS named_value
AS $$
  import numpy as np
  return ((name,i) for i in np.arange(3))
$$ LANGUAGE plpythonu;
```

Benefits of PL/Python

- Easy to bring your code to the data.
- When SQL falls short leverage your Python (or R/Java/C) experience quickly.
- Apply Python across terabytes of data with minimal overhead or additional requirements.
- Results are already in the database system, ready for further analysis or storage.

MADlib

Going Beyond Data Parallelism

- Data Parallel computation via PL/Python libraries only allow us to run ‘n’ models in parallel.
- This works great when we are building one model for each value of the group by column, but we need parallelized algorithms to be able to build a single model on all the available data
- For this, we use MADlib – an open source library of parallel in-database machine learning algorithms.

MADlib: The Origin

UrbanDictionary

mad *(adj.): an adjective used to enhance a noun.*
1- dude, you got skills.
2- dude, you got mad skills

- First mention of MAD analytics was at VLDB 2009

  MAD Skills: New Analysis Practices for Big Data
  J. Hellerstein, J. Cohen, B. Dolan, M. Dunlap, C. Welton
  (with help from: Noelle Sio, David Hubbard, James Marca)
  http://db.cs.berkeley.edu/papers/vldb09-madskills.pdf

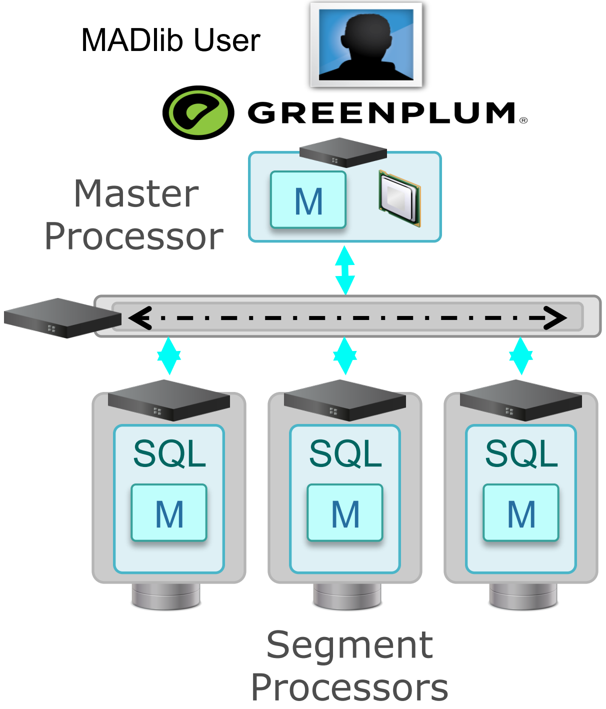
- MADlib project initiated in late 2010:

  Greenplum Analytics team and Prof. Joe Hellerstein

- Open Source!
  https://github.com/madlib/madlib
- Works on Greenplum DB, PostgreSQL and also HAWQ & Impala
- Active development by Pivotal
  - Latest Release: v1.4 (Nov 2013)
- Downloads and Docs:
  http://madlib.net/

MADlib

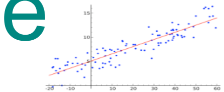
MADlib Executes Algorithms In-Place

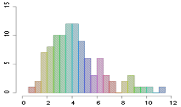


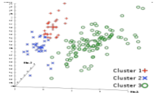
MADlib Advantages

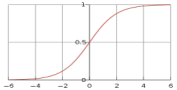
- No Data Movement
- Use MPP architecture's full compute power
- Use MPP architecture's entire memory to process data sets

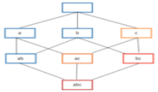
MADlib In-Database Functions











Predictive Modeling Library

Generalized Linear Models
- Linear Regression
- Logistic Regression
- Multinomial Logistic Regression
- Cox Proportional Hazards
- Regression
- Elastic Net Regularization
- Sandwich Estimators (Huber white, clustered, marginal effects)

Matrix Factorization
- Single Value Decomposition (SVD)
- Low-Rank

Machine Learning Algorithms
- Principal Component Analysis (PCA)
- Association Rules (Affinity Analysis, Market Basket)
- Topic Modeling (Parallel LDA)
- Decision Trees
- Ensemble Learners (Random Forests)
- Support Vector Machines
- Conditional Random Field (CRF)
- Clustering (K-means)
- Cross Validation

Linear Systems
- Sparse and Dense Solvers

Descriptive Statistics

Sketch-based Estimators
- CountMin (Cormode-Muthukrishnan)
- FM (Flajolet-Martin)
- MFV (Most Frequent Values)

Correlation

Summary

Support Modules

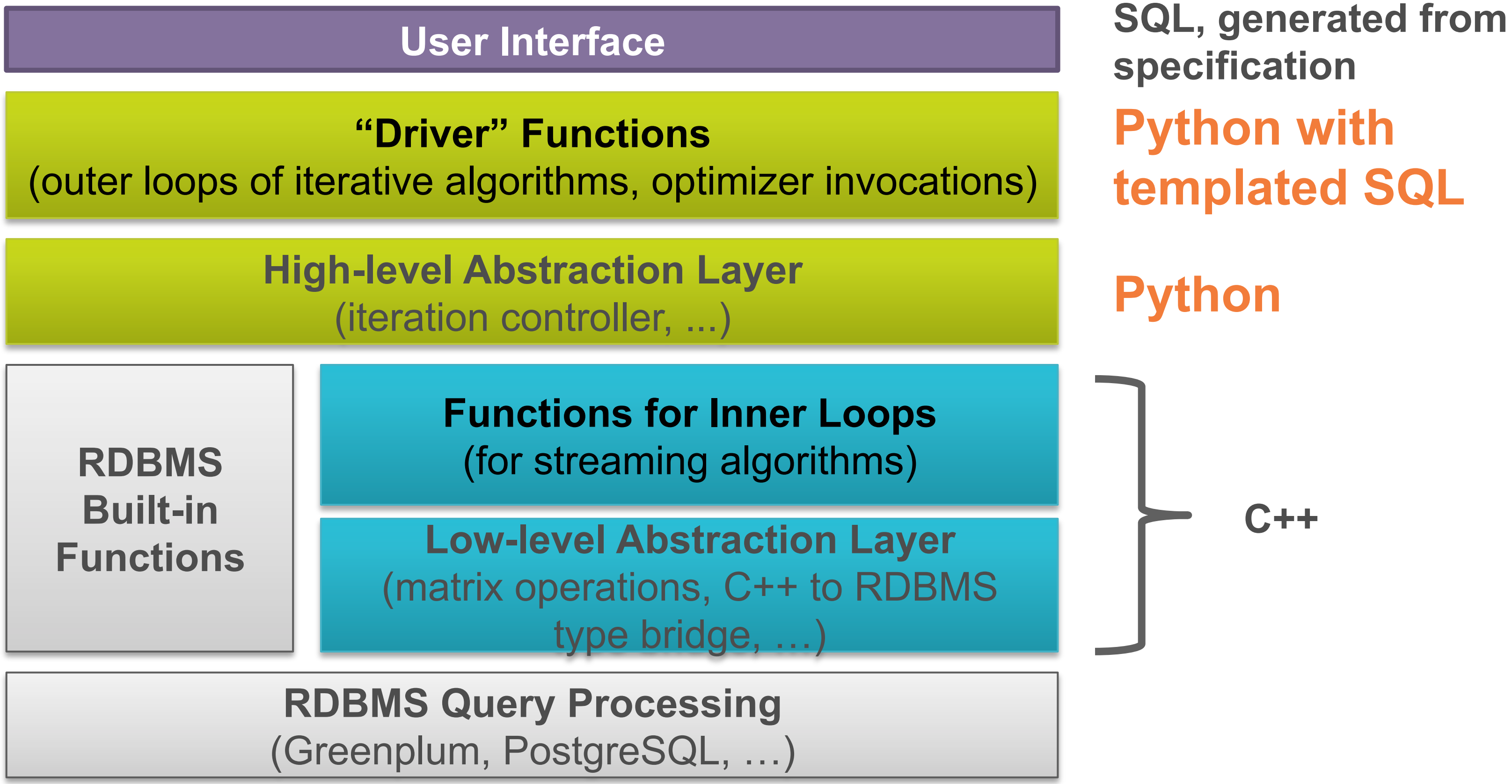
Array Operations

Sparse Vectors

Random Sampling

Probability Functions

Architecture



User Interface

"Driver" Functions
(outer loops of iterative algorithms, optimizer invocations)

High-level Abstraction Layer
(iteration controller, ...)

RDBMS Built-in Functions

Functions for Inner Loops
(for streaming algorithms)

Low-level Abstraction Layer
(matrix operations, C++ to RDBMS type bridge, …)

RDBMS Query Processing
(Greenplum, PostgreSQL, …)

SQL, generated from specification

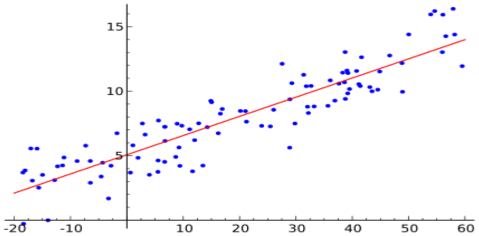
Python with templated SQL

Python

C++

How does it work ? : A Linear Regression Example

- Finding linear dependencies between variables
  - $y \approx c_0 + c_1 \cdot x_1 + c_2 \cdot x_2$?



```
# select y, x1, x2  from unm limit 6;

   y    |  x1   | x2
--------+-------+-----
  10.14 |     0 | 0.3
  11.93 |  0.69 | 0.6
  13.57 |   1.1 | 0.9
  14.17 |  1.39 | 1.2
  15.25 |  1.61 | 1.5
  16.15 |  1.79 | 1.8
```

Vector of dependent variables y

Design Matrix X

Reminder: Linear-Regression Model

- $E[Y \mid \boldsymbol{x}] = \boldsymbol{x}^T \boldsymbol{c}$
- If residuals i.i.d. Gaussians with standard deviation *σ*:
  - max likelihood ⇔ min sum of squared residuals

$$f(y \mid \boldsymbol{x}) \propto \exp\left(-\frac{1}{2\sigma^2} \cdot (y - \boldsymbol{x}^T \boldsymbol{c})^2\right)$$

- First-order conditions for the following quadratic objective (in ***c***)

$$(\boldsymbol{y} - X\boldsymbol{c})^T (\boldsymbol{y} - X\boldsymbol{c})$$

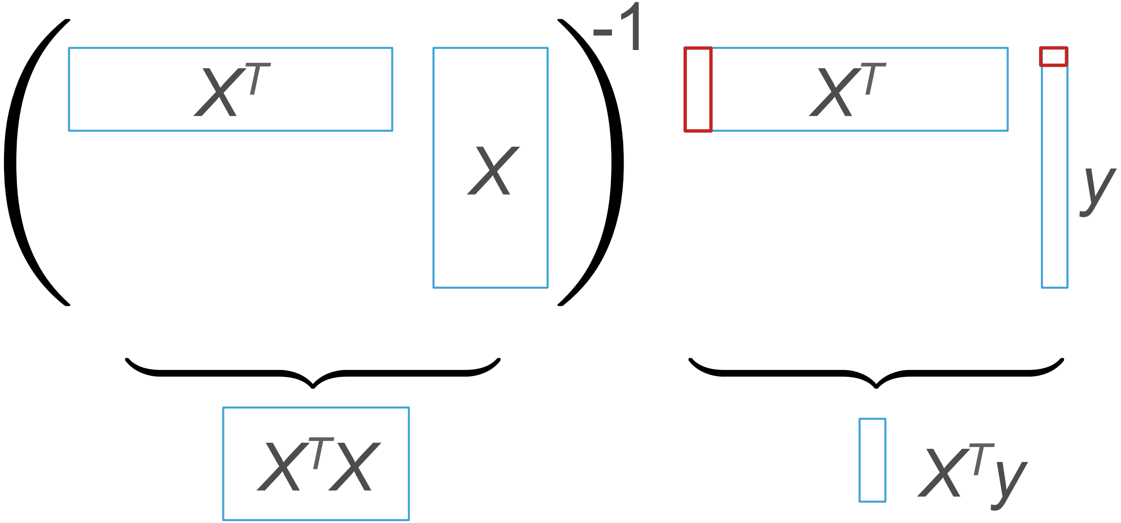
yield the minimizer

$$\widehat{\boldsymbol{c}} = (X^T X)^{-1} X^T \boldsymbol{y}$$

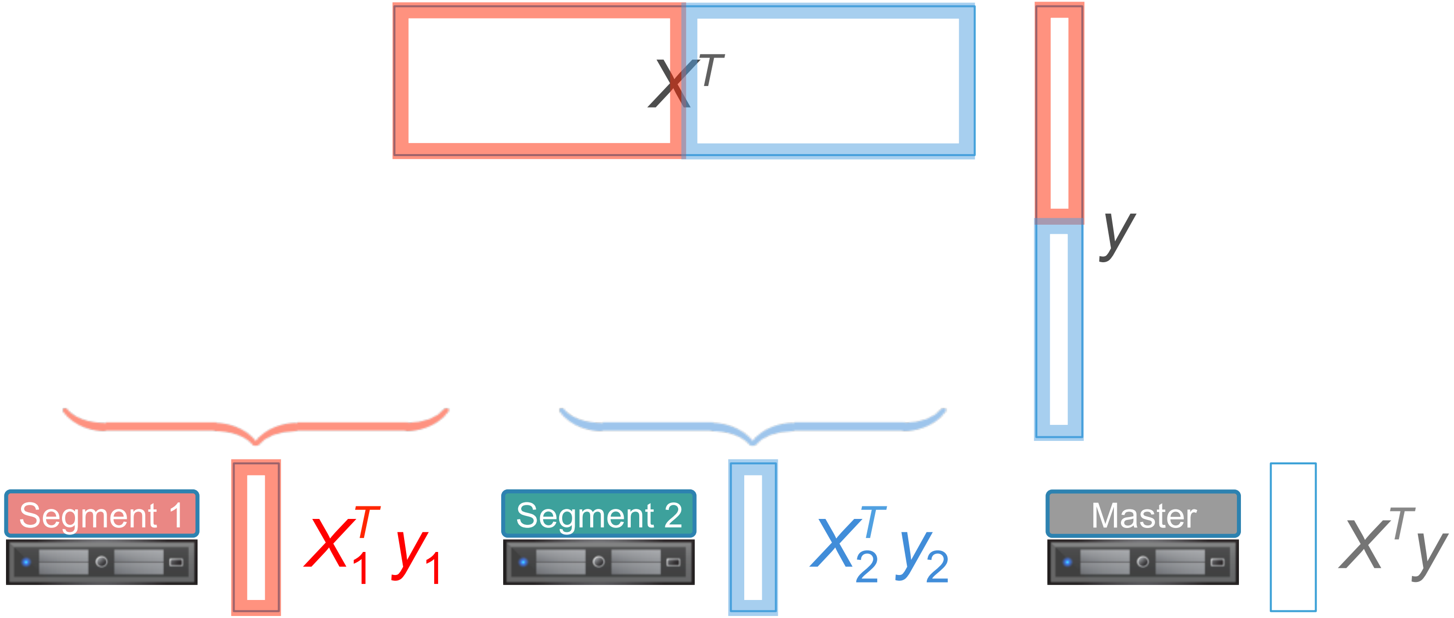
Linear Regression: Streaming Algorithm

- How to compute with a single table scan?

$$\widehat{\boldsymbol{c}} = (X^T X)^{-1} X^T \boldsymbol{y}$$



Linear Regression: Parallel Computation

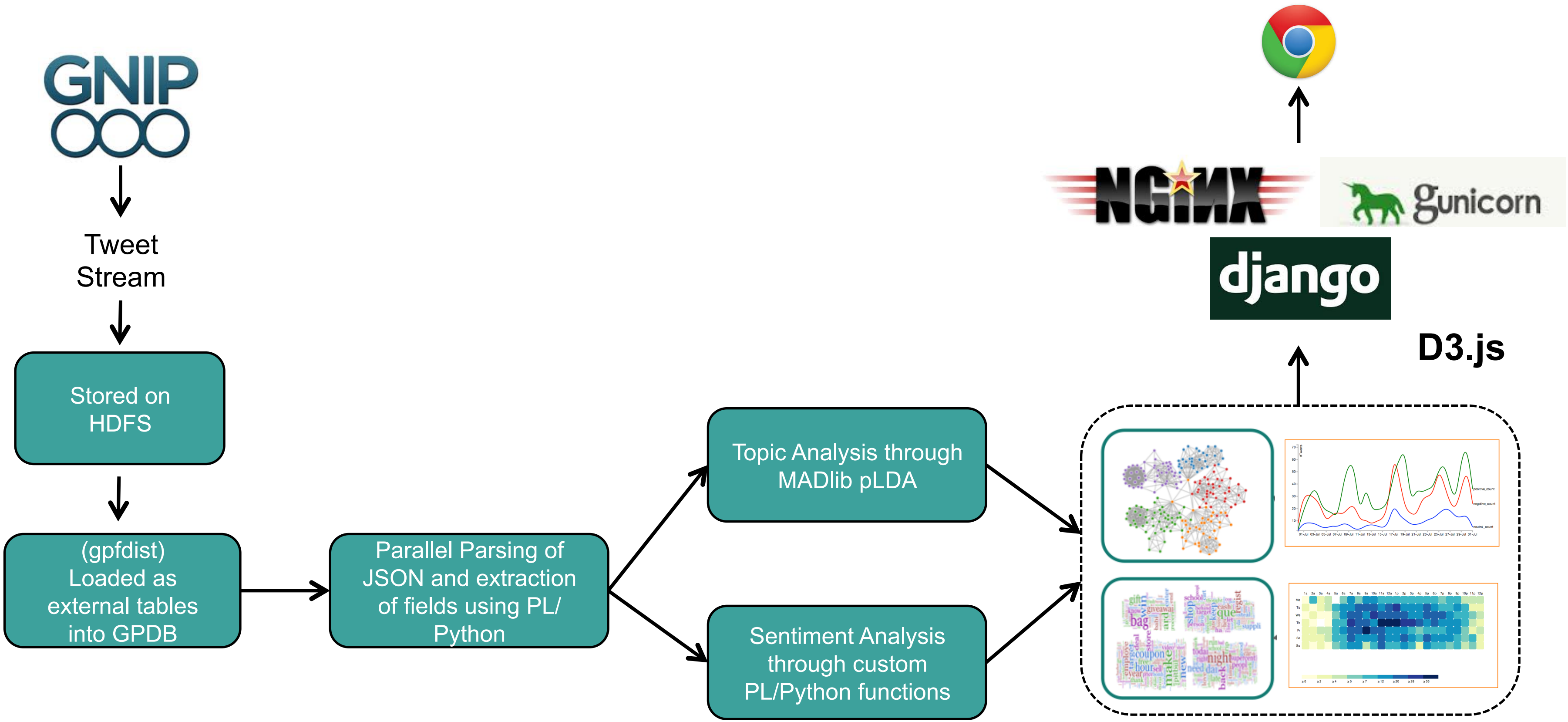


$$X^T \boldsymbol{y} = (X_1^T \;\; X_2^T) \begin{pmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \end{pmatrix} = \sum X_i^T \boldsymbol{y}_i$$

Demos

- We built demos to showcase our technology pipeline, using Python technology.
- Two use cases:
  - Topic and Sentiment Analysis of Tweets
  - London Road Traffic Disruption prediction

Topic and Sentiment Analysis Pipeline



GNIP

Tweet Stream

Stored on HDFS

(gpfdist) Loaded as external tables into GPDB

Parallel Parsing of JSON and extraction of fields using PL/ Python

Topic Analysis through MADlib pLDA

Sentiment Analysis through custom PL/Python functions

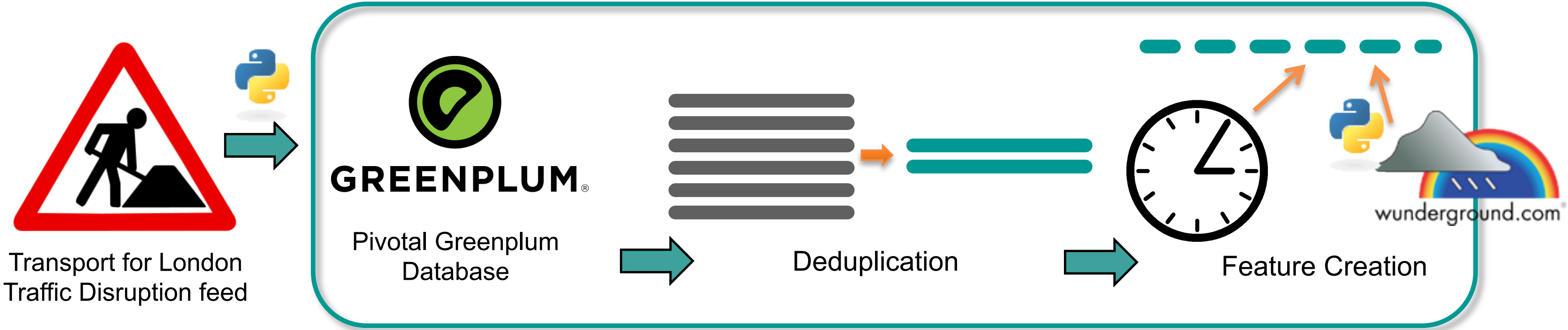
D3.js

django

NGINX

gunicorn

Transport Disruption Prediction Pipeline

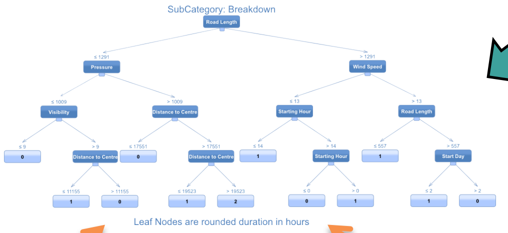


Transport for London Traffic Disruption feed

Pivotal Greenplum Database

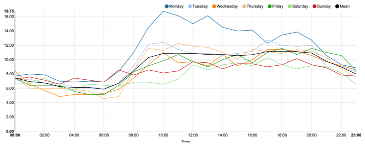
Deduplication

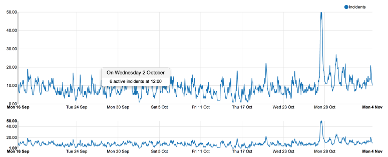
Feature Creation



python™ scikit learn

Modelling & Machine Learning





d3.js & NVD3

Interactive SVG figures

CLOUD FOUNDRY™

@ianhuston

Pivotal™

Get in touch

Feel free to contact me about PL/Python, or more generally about Data Science and opportunities available.

@ianhuston

ihuston @ gopivotal.com

http://www.ianhuston.net

Pivotal

BUILT FOR THE SPEED OF BUSINESS