






T^FM Control Autogeneración

Sistema para el procesamiento y análisis de información de sistemas de autogeneración energética

**Trabajo fin de master - 8º Master de
arquitectura Big Data**

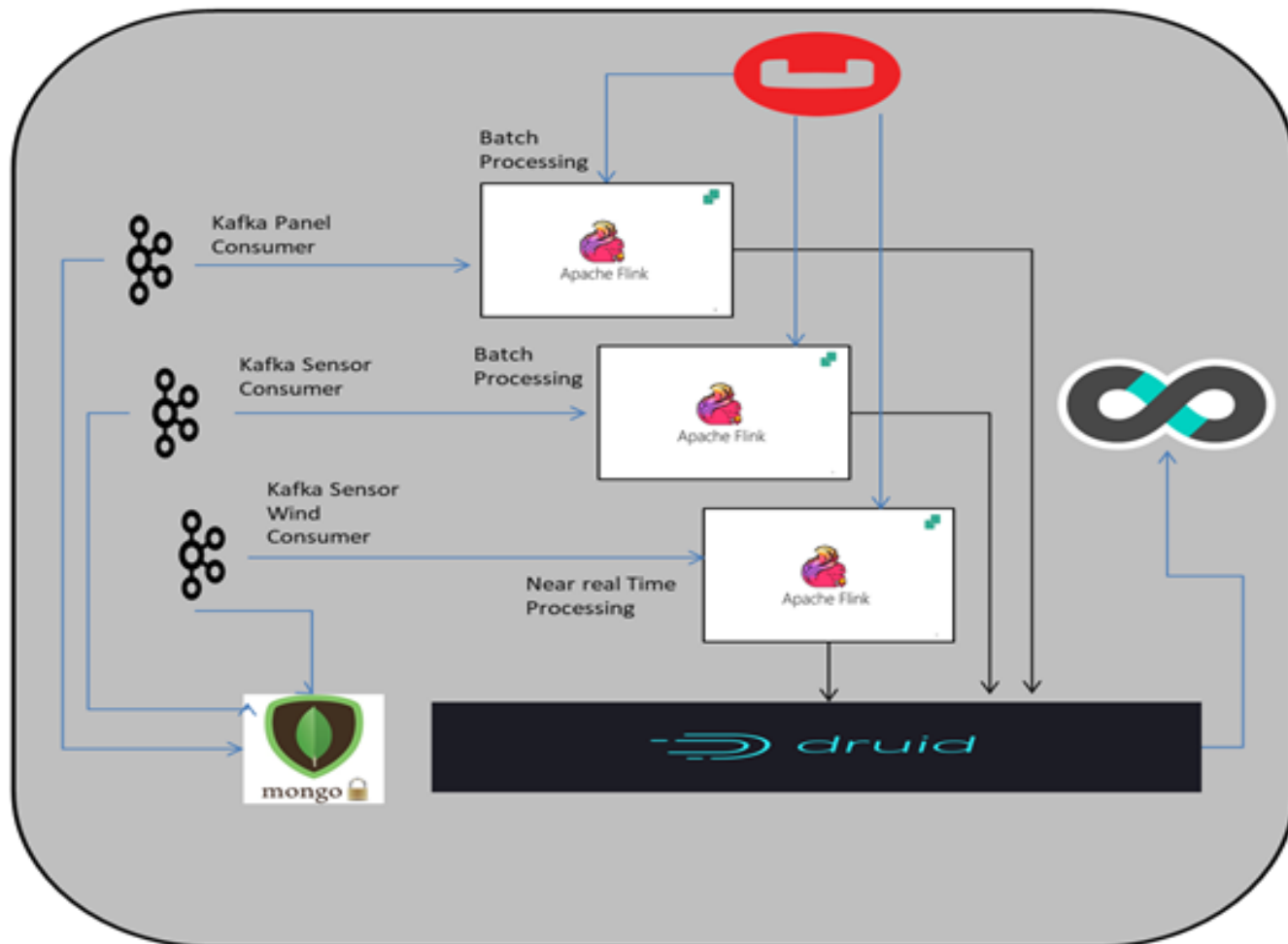
Requisitos

Se quiere disponer un sistema con el que poder:

-  Recibir información de diferentes sistemas para autogeneración energética. El análisis de la información posee un componente en tiempo real y otro en modo batch. Los elementos tenidos en cuenta en el presente TFM son placas solares, sensores de luz y anemómetros.
-  Ejecutar consultas de la potencia en W generada por placas solares. Esta información se generará por tramos de 1 hora. La información obtenida en tiempo real se agrupará de esta forma para obtener puntos de control que tengan un sentido práctico y sirva de comparación entre tramos y fechas.
-  Ejecutar consultas sobre la información aportada por los sensores de luz. Estos sensores no proporcionan, a priori, información directa sobre la cantidad de energía que se podría producir, necesitan de un cálculo previo que da una aproximación a dicho fin. Estos sensores pueden dar información sobre el ángulo óptimo y dirección de las posibles placas a colocar en su lugar para el mejor aprovechamiento de las mismas.
-  Ejecutar consultas sobre la información aportada por los sensores de luz. Estos sensores no proporcionan, a priori, información directa sobre la cantidad de energía que se podría producir, necesitan de un cálculo previo que da una aproximación a dicho fin. Estos sensores pueden dar información sobre el ángulo óptimo y dirección de las posibles placas a colocar en su lugar para el mejor aprovechamiento de las mismas.
-  Consultar la información sobre el rendimiento del sistema en la entrada de la información de los diferentes topic's.
-  Toda la información que se consulte referente a los tres tipos de fuentes irá enriquecida con datos de ubicación, modelo de fuente, propietario del equipo, fecha de fabricación del equipo, fecha de instalación del mismo, etc.

Arquitectura

- ❑ Para el almacenamiento y procesamiento de los datos enviados por las diferentes fuentes en tiempo real: MongoDB.
- ❑ Para el almacenamiento y procesamiento del histórico: Druid.
- ❑ Para almacenamiento información elementos e instalaciones: Couchbase.
- ❑ Motor de procesamiento en near real time: Apache Flink
- ❑ Motor de procesamiento batch: Apache Flink
- ❑ Para la visualización: Apache Superset.



Kafka - procesos

- ❑ StreamingSensorWind.py: tiene como objetivo procesar los mensajes enviados, en formato JSON por los anemómetros. La información enviada a través del topic, se enriquece y se almacena en MongoDB.
- ❑ StreamingSensorLight.py: tiene como objetivo procesar los mensajes enviados, en formato JSON por los sensores de luz. La información enviada a través del topic, se enriquece y se almacena en MongoDB.
- ❑ StreamingPanel.py: tiene como objetivo procesar los mensajes enviados, en formato JSON por las placas solares. La información enviada a través del topic, se enriquece y se almacena en MongoDB.

Procesos near Real Time Flink

- ❑ RichSensorWind.py: tiene como objeto enriquecer desde Couchbase con los datos de ubicación y timestamp del evento la información recibida desde el topic. Este proceso es casi en real time, tomaremos ventanas de 1 segundo.
- ❑ WriteSensorWind.py: tiene como objeto el almacenar en Druid la información ya enriquecida desde el proceso anterior.

Procesos Batch Flink

- ❑ RichSensorLight.py: tiene como objeto enriquecer desde Couchbase con los datos de ubicación y timestamp del evento la información recibida desde el topic. Este proceso es batch, tomaremos ventanas de 1 hora. La información estará en disco.
- ❑ WriteSensorLight.py: tiene como objeto el almacenar en Druid la información ya enriquecida desde el proceso anterior.
- ❑ RichPanel.py: tiene como objeto enriquecer desde Couchbase con los datos de ubicación y timestamp del evento la información recibida desde el topic. Este proceso es batch, tomaremos ventanas de 1 hora. La información estará en disco.
- ❑ WritePanel.py: tiene como objeto el almacenar en Druid la información ya enriquecida desde el proceso anterior.

Procesos Alta Couchbase

- ❑ AddNewInstall.py: tiene como objeto añadir información a las tablas de Couchbase sobre nuevos elementos instalados con la información necesaria para los procesos en near real time o batch.
- ❑ AddNewElement.py: tiene como objeto añadir información a las tablas de Couchbase sobre nuevos tipos de sensores, placas o anemómetros.

Kafka - procesos

Proceso	Periodicidad
StreamingSensorWind.py	Real Time
StreamingSensorLight.py	Real Time
StreamingPanel.py	Real Time

Procesos near Real Time Flink

Proceso	Periodicidad
RichSensorWind.py	Real Time
WriteSensorWind.py	Real Time

Procesos Batch Flink

Proceso	Periodicidad
RichSensorLight.py	Batch
WriteSensorLight.py	Batch
RichPanel.py	Batch
WritePanel.py	Batch

Procesos Alta Couchbase

Proceso	Periodicidad
AddNewInstall.py	Real Time
AddNewElement.py	Real Time

Modelo de datos - Batch (DRUID) 1/2

panel_report	
campo	descripción
client_id	código cliente
panel_id	identificación placa
address	dirección cliente
postal code	código postal
name	nombre del cliente
apell_1	apellido del cliente
apell_2	apellido del cliente
power_gen	energía generada última hora
timestamp	timestamp de generación dato
diff_last_hour	delta de generación de la ultima hora a la que está tratando en ese momento
diff_last_day	delta de generación de la misma hora del día anterior
diff_last_month	delta de generación de la misma hora del mes anterior
diff_last_quar	delta de generación de la misma hora un trimestre antes
diff_last_six	delta de generación de la misma hora un semestre antes
diff_last_year	delta de generación de la misma hora de un año antes
acum_day	acumulado del día
acum_month	acumulado del mes
acum_quarter	acumulado del trimestre
acum_semestre	acumulado del semestre
acum_year	acumulado del año

Modelo de datos - Batch (DRUID) 2/2

panel_alarm		sensor_alarm	
campo	descripción	campo	descripción
client_id	código cliente	client_id	código cliente
panel_id	identificación placa	light_id	identificación placa
address	dirección cliente	address	dirección cliente
postal code	código postal	postal code	código postal
name	nombre del cliente	name	nombre del cliente
apell_1	apellido del cliente	apell_1	apellido del cliente
apell_2	apellido del cliente	apell_2	apellido del cliente
power_low	diferencia de generación	power_low	diferencia de generación
text_alarm	texto asociado a la alarma	text_alarm	texto asociado a la alarma
timestamp		timestamp	

Modelo de datos - Real Time (DRUID) 1/1

wind_alarm	
campo	descripción
client_id	código cliente
wind_id	identificación placa
address	dirección cliente
postal code	código postal
name	nombre del cliente
apell_1	apellido del cliente
apell_2	apellido del cliente
wind_speed	velocidad del viento
text_alarm	texto asociado a la alarma
timestamp	

Modelo de datos - Real Time (MongoDB) 1/1

topic		topic_alarm	
campo	descripción	campo	descripción
topic_id	nombre del topic de kafka	topic_id	nombre del topic de kafka
timestamp	timestamp	timestamp	timestamp
msg_value	datos del mensaje del topic	msg_value	datos del mensaje del topic
		text_alarm	texto asociado a la alarma

Modelo de datos - Enriquecimiento (Couchbase) 1/2

client		panel		sensor_light		sensor_wind	
campo	descripción	campo	descripción	campo	descripción	campo	descripción
client_id	código cliente	panel_id	id del panel	light_id	id sensor	wind_id	id anemómetro
address	dirección cliente	model	modelo panel	model	modelo panel	model	modelo panel
postal code	código postal	manufacture_date	fecha fabricación	manufacture_date	fecha fabricación	manufacture_date	fecha fabricación
client_signature	identificación cliente	install_date	fecha instalación	install_date	fecha instalación	install_date	fecha instalación
open_date	fecha alta	power_cap	potencia del panel	power_cap	potencia del panel	variable_data	información adicional
end_date	fecha baja	panel_price	precio de compra	variable_data	información adicional	sensor_price	precio de compra
client_type	tipo de cliente	panel_sold	precio de venta	sensor_price	precio de compra	live_cicle	ciclo de vida útil
name	nombre del cliente	live_cicle	ciclo de vida útil	live_cicle	ciclo de vida útil	end_date	fecha de baja
apell_1	apellido del cliente	variable_data	información adicional	end_date	fecha de baja		
apell_2	apellido del cliente	end_date	fecha de baja				

Modelo de datos - Enriquecimiento

(Couchbase) 2/2

clients_rel	
campo	descripción
client_id	id cliente
panel_id	id panel
light_id	id sensor
wind_id	id anemómetro
variable_data	información adicional
init_panel_date	fecha de alta relación cliente-panel
end_panel_date	fecha de baja relación cliente-panel
init_sensor_date	fecha de alta relación cliente-sensor
end_sensor_date	fecha de baja relación cliente-sensor
init_wind_date	fecha de alta relación cliente-wind
end_wind_date	fecha de baja relación cliente-wind

Superset

Historico empresas ☆

Ingresos totales

456M

Filtro compañías

Since

or

Until

or

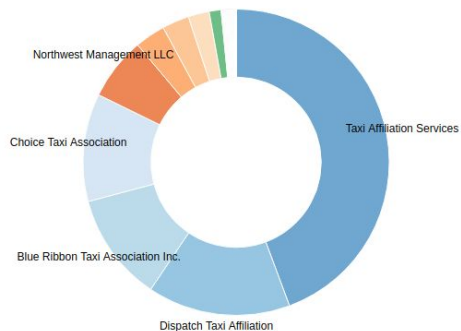
Time Grain

month

company

Select [company]

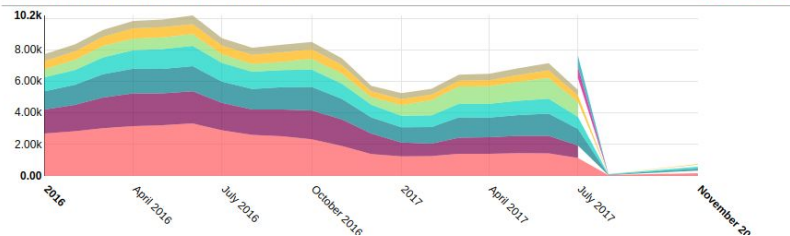
Distribución de los ingresos por empresas



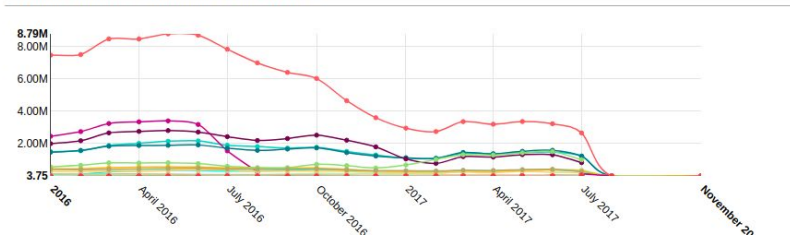
Empresas por ingresos

company	sum_trip_totals
Taxi Affiliation Services	106M
Dispatch Taxi Affiliation	36.1M
Blue Ribbon Taxi Association Inc.	30.3M
Choice Taxi Association	29.2M
Chicago Elite Cab Corp. (Chicago Carriag	20.1M
Northwest Management LLC	16.0M
KOAM Taxi Association	7.66M
Top Cab Affiliation	7.20M
Chicago Medallion Leasing INC	5.44M
Suburban Dispatch LLC	5.01M
Chicago Medallion Management	3.00M
T.A.S. - Payment Only	1.15M
Blue Cab Co	210k
Dispatch Taxi Affiliation (credit hold)	163k
0118 - 42111 Godfrey S.Awir	141k
5129 - 87128	134k
6743 - 78771 Luhak Corp	134k
Star North Management LLC	119k

Media mensual de ingresos por empresa

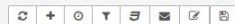


Evolución ingresos mensuales por empresa

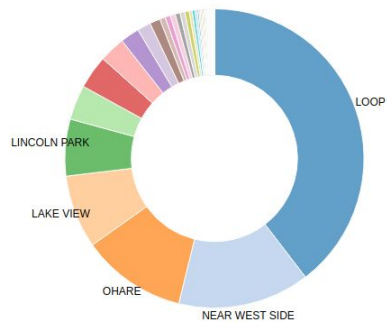


Superset

Histórico áreas ☆



Distribución de los viajes por área de inicio



Filtro - áreas

Since Until

Time Grain

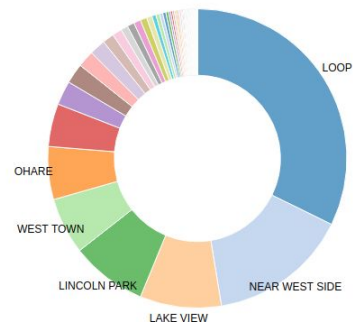
dropoff_community_area_name

Select [dropoff_community_area_name]

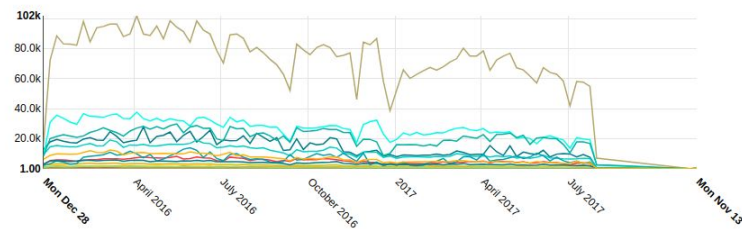
Número total de viajes

27.6M

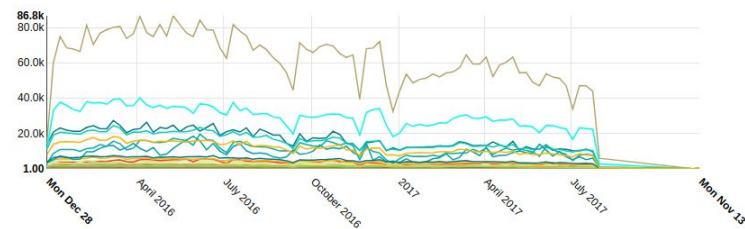
Distribución de los viajes por área de fin



Evolución de número de viajes por área de inicio



Evolución número de viajes por área de fin



Superset

Real time ☆



Filtro - real time

Since **2 days ago** Until **on**

Time Granularity
1 minute

company
Select [company]

pickup_community_area_name
Select [pickup_community_area_name]

dropoff_community_area_name
Select [dropoff_community_area_name]

Evolución del número de viajes



Número total de viajes

1.22k

Áreas por número de viajes



Problemas / Soluciones

- ❑ Para paliar el tráfico de red entre Kafka y MongoDB ubicaremos en el mismo nodo ambos elementos.
- ❑ Para paliar el tráfico de red entre Druid y Superset ubicaremos en el mismo nodo ambos elementos.

DEMO