Taller Apache Cassandra

Curso Big Data

David Martínez Casas

david.martinez.casas@usc.es

Centro Singular de Investigación en Tecnoloxías da Información

UNIVERSIDADE DE SANTIAGO DE COMPOSTELA

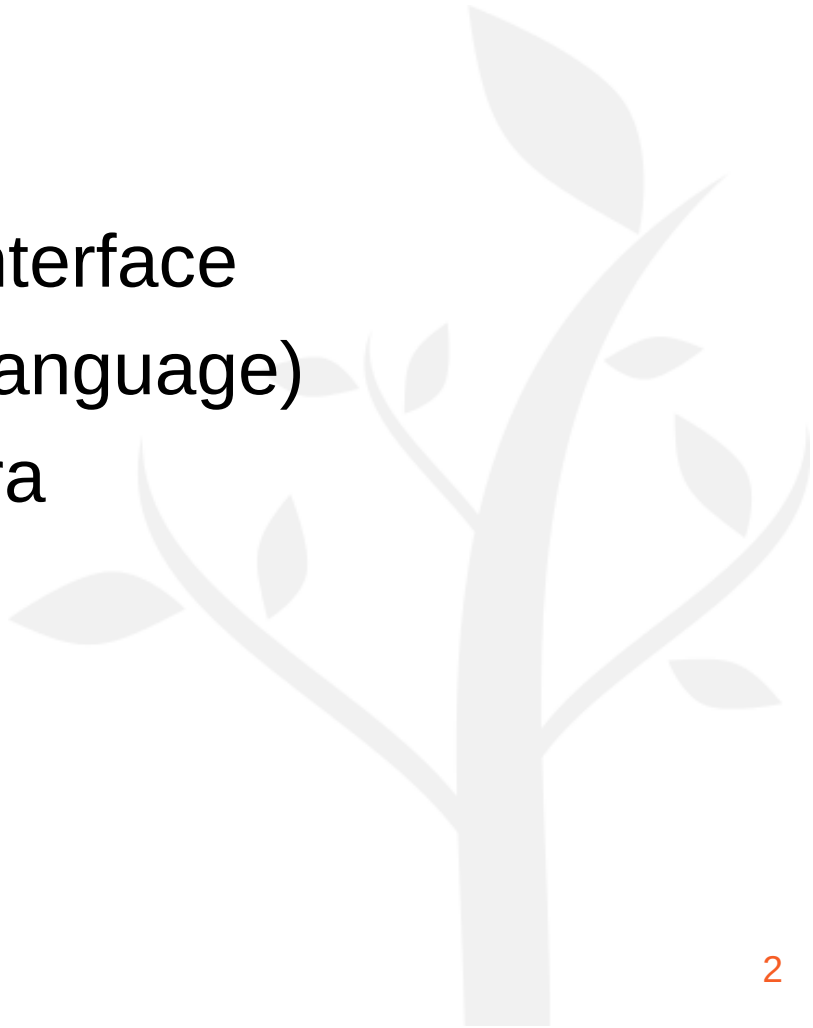
citius.usc.es

USC UNIVERSIDADE DE SANTIAGO DE COMPOSTELA CAMPUS VIDA

CiTiUS Centro Singular de Investigación en Tecnoloxías da Información

Índice

- Introducción a Cassandra
- Modelo de datos
- Instalación y configuración
- Cassandra command line interface
- CQL 3 (Cassandra Query Language)
- Clientes Cassandra: kundera
- Ejemplos
- Cluster multinodo



CiTUS

Que es Apache Cassandra

Introducción

- Apache Cassandra es un motor de bases de datos NoSQL, Open Source e implementado en Java.
- Fue originalmente creada por Facebook y donada a Apache como software libre en 2009.
- Es una de las base de datos NoSQL más relevantes a nivel mundial: Netflix, eBay, Twitter, Urban Airship, Constant Contact, Reddit, Cisco, OpenX, Digg, CloudKick, Ooyala, …
- Cassandra puede manejar varios terabytes de datos si lo necesita y puede, fácilmente, manejar millones de ficheros, incluso en un clúster pequeño (Big Data).
- La información en las bases de datos relacionales, se almacenan en forma de filas, pero en Cassandra la información se almacena en columnas con pares key-value.

Características de Cassandra I

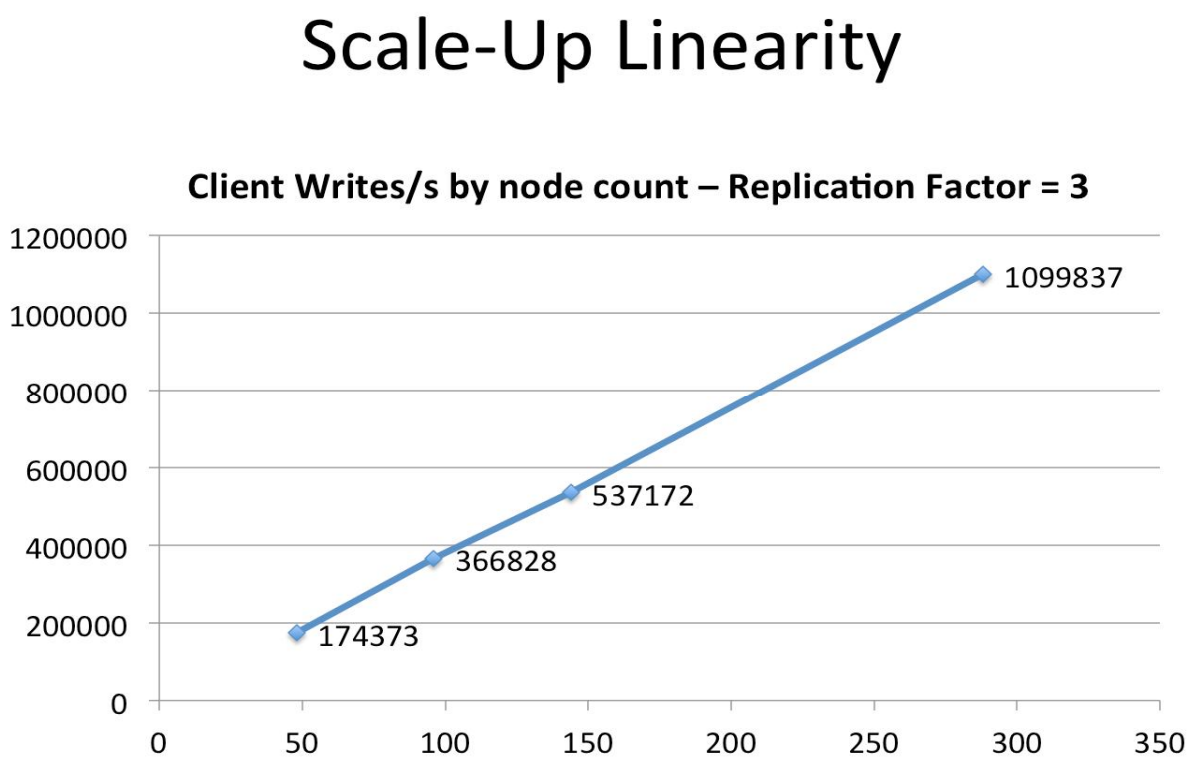
Introducción

- Esquema dinámico. El esquema que define la estructura de los datos puede cambiar en tiempo de ejecución.
- No hay un único punto de fallo. Los datos se replican automáticamente a varios nodos. Perder un nodo no causa la baja del clúster.
- Alta disponibilidad. Los datos estás disponibles la mayor parte del tiempo gracias a la redundancia que introduce la replicación de datos.
- Particionamiento de los datos. La topología de Cassandra es la de un anillo a través del cual se distribuyen los datos para minimizar cuellos de botella en el acceso a los mismos.
- Escalabilidad horizontal. Hasta un alto número de máquinas la capacidad de cómputo aumenta linealmente con el número de máquinas.
- Capacidad para manejar cientos de gigabytes de datos.

Características de Cassandra II

Introducción

- Soporte profesional: varias empresas dan soporte y construyen productos sobre Cassandra: Datastax, Acunu, …



CiTUS

NETFLIX

Terminología de Cassandra

Modelo

- **Column**. Es la unidad mas básica en el modelo de datos de Cassandra. Una column es un triplete de un key (un nombre) un value (un valor) y un timestamp. Los valores son todos suministrados por el cliente. El tipo de dato del key y el value son matrices de bytes de Java, el tipo de dato del timestamp es un long primitive.
  - Las column son inmutables para evitar problemas de multithreading.
  - Las column se organizan dentro de las columns families.
  - Las column se ordenan por un tipo, que pueden ser uno de los siguientes:
    - AsciiType
    - BytesType
    - LongType
    - TimeUUIDType
    - UTF8Type

CiTUS

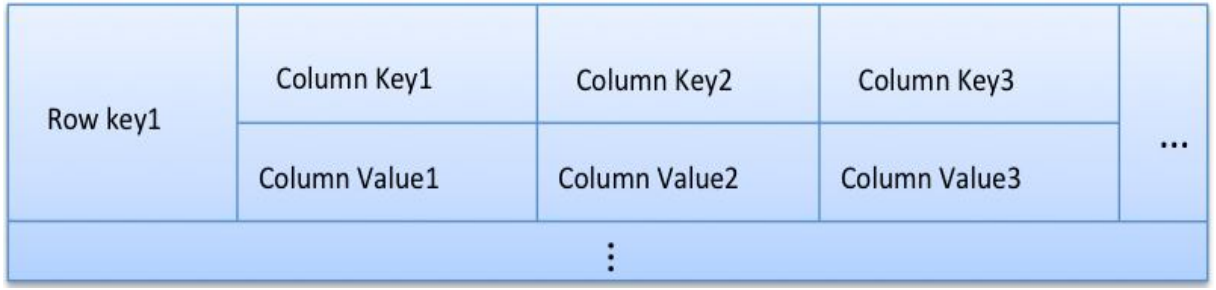
Terminología de Cassandra

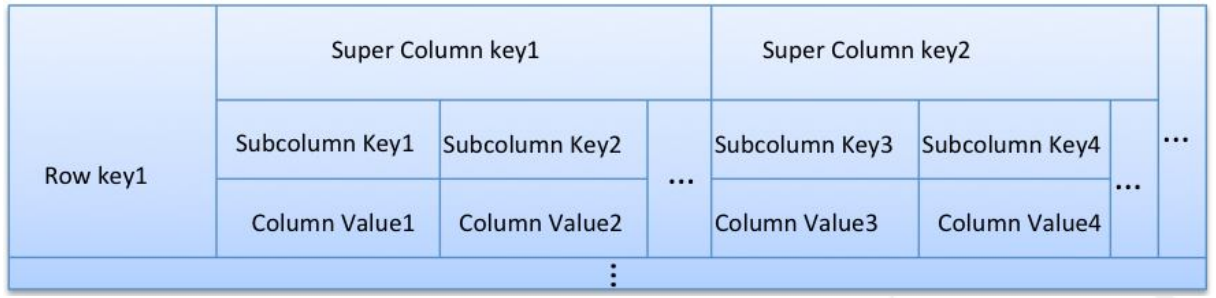
Modelo

- **SuperColumn**. Es una column cuyos values son una o más columns, que en este contexto se llamaran subcolumns. Las subcolumns están ordenadas, y el numero de columnas que se puede definir es ilimitada. Las Super columns, a diferencias de las columns, no tienen un timestamp definido.
- **Column Family**. Es mas o menos análogo a una tabla en un modelo relacional. Se trata de un contenedor para una colección ordenada de columns. Cada column family se almacena en un archivo separado
- **Keyspace**. Es el contenedor para las column family. Es mas o menos análogo a una base de datos en un modelo relacional, usado en Cassandra para separar aplicaciones. Un keyspace es una colección ordenada de columns family.
- **Clúster**. Conjunto de máquinas que dan soporte a Cassandra y son vistas por los clientes como una única máquina.

Modelo de datos

Modelo



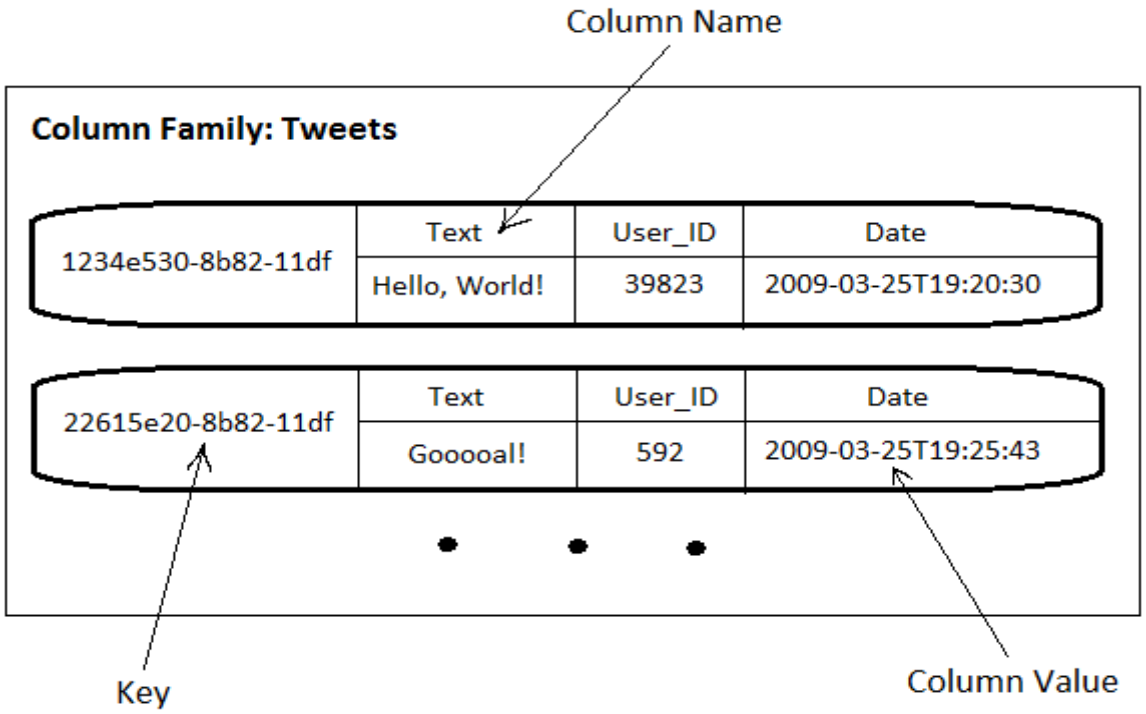


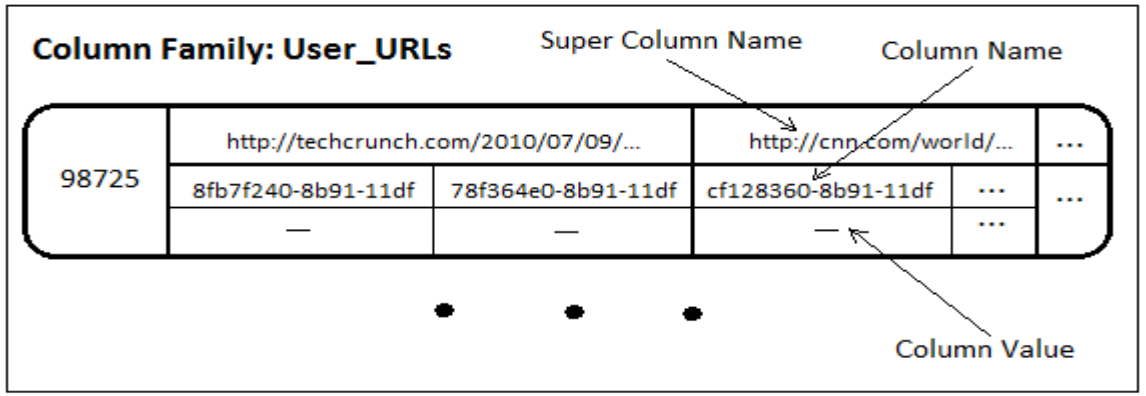
Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family (CF)
Primary key	Row key
Column name	Column name/key
Column value	Column value

CiTUS

Modelo de datos

Modelo





Relaciones

Modelo

- Entidad
  - People: {
    John Dow: {
    twitter: jdow,
    email: jdow@example.com,
    bio: bla bla bla,
    ...
    },
    }

People				
John Dow	Twitter	Email	bio	...
	Jdow	jdow@	Bla, bla	...

- One to one
  - People: {
    John Dow: {
    twitter: jdow,
    email: jdow@example.com,
    bio: bla bla bla,
    marriedTo: Mary Kay,
    ...
    },
    Mary Kay: {
    marriedTo: John Dow,
    ...
    },
    ...
    }

CiTUS

Relaciones

Modelo

- One to many
  - Children: {
    John Dow: {
    01/18/1976: John Dow Jr,
    05/27/1982: Kate Dow
    },
    ...
    }
  - People: {
    John Dow Jr: {
    father: John Dow,
    ...
    },
    ...
    }

- Many to many
  - Friendship: {
    John Dow: {
    10: Mark Seldon,
    8: Julian Hendrix,
    ...
    },
    Mark Seldon: {
    9: John Dow,
    ...
    },
    ...
    }

CiTIUS

Descargar e instalar

Instalación

- Se obtiene la última versión de Apache Cassandra (actualmente 1.2.6) de http://cassandra.apache.org/download/
- Se descarga al ordenador y se descomprime dentro de un directorio:
  - **tar xvf apache-cassandra-1.2.6-bin.tar.gz**
- Se modifica el fichero de configuracion:
  - **<cassandra_path>/conf/cassandra.yaml**
    - *listen_address*. Por defecto localhost
    - *rpc_address*. Por defecto localhost en el puerto 9160
    - *data_file_directories*. Permisos de escritura
    - *commitlog_directory*. Permisos de escritura
    - *saved_caches_directory*. Permisos de escritura
- Se necesita Java instalado (1.5+).
- Se ejecuta el script de inicio:
  - **<cassandra_path>/bin/cassandra.sh**
  - *“Startup completed! Now serving reads.”*

CiTUS

Cassandra-cli

Introducción

- <cassandra-path>/bin/cassandra-cli –host localhost –port 9160
  - Nos permite conectarnos a un nodo de un cluster. Por defecto se conecta a *localhost:9160*.
- Principales comandos:
  - help
  - connect
  - create/drop keyspace <keyspace_name>
  - use <keyspace_name>
  - create/drop column family <column_family>
  - set
  - get
  - list
  - update
  - truncate
  - show

CiTUS

Cassandra-cli

Introducción

- CREATE KEYSPACE demo;
- USE demo;
- CREATE COLUMN FAMILY users
  WITH comparator = UTF8Type
  AND key_validation_class=UTF8Type
  AND column_metadata = [
  {column_name: full_name, validation_class: UTF8Type}
  {column_name: email, validation_class: UTF8Type}
  {column_name: state, validation_class: UTF8Type}
  {column_name: gender, validation_class: UTF8Type}
  {column_name: birth_year, validation_class: LongType}
  ];
- CREATE COLUMN FAMILY blog_entry
  WITH comparator = TimeUUIDType
  AND key_validation_class=UTF8Type
  AND default_validation_class = UTF8Type;

Cassandra-cli

Introducción

- SET users['A']['full_name']='Robert Jones';
- SET users['A']['email']='bobjones@gmail.com';
- SET users['A']['state']='TX';
- SET users['A']['gender']='M';
- SET users['A']['birth_year']='1975';

- SET users['B']['full_name']='Cathy Smith';
- SET users['B']['state']='CA';
- SET users['B']['gender']='F';
- SET users['B']['birth_year']='1969';

- SET blog_entry['user'][timeuuid()] = 'I love my new shoes!';

- LIST users;
- LIST blog_entry;

Cassandra-cli

Introducción

- CREATE COLUMN FAMILY page_view_counts
  WITH default_validation_class=CounterColumnType
  AND key_validation_class=UTF8Type AND comparator=UTF8Type;
- INCR page_view_counts['www.datastax.com'][home] BY 1;
- LIST page_view_counts;

- GET users[utf8('A')][utf8('full_name')];
- UPDATE COLUMN FAMILY users WITH comparator = UTF8Type
  AND column_metadata =
  [{column_name: birth_year,
  validation_class: LongType,
  index_type: KEYS
  }
  ];
- GET users WHERE birth_year = 1969;

CiTUS

Cassandra-cli

Introducción

- DEL users [‘B’][‘gender’];
- GET users [‘B’];
- DEL users [‘B’];
- DROP COLUMN FAMILY users;
- DROP COLUMN FAMILY blog_entry;
- CREATE COLUMN FAMILY SuperExample with column_type = Super
  AND comparator= UTF8Type AND subcomparator = UTF8Type;
- SET SuperExample[utf8('Key1')]['A']['B']=utf8('Valor1');
- SET SuperExample[utf8('Key1')]['A']['C']=utf8('Valor2');
- DROP KEYSPACE demo;

CQL3

Introducción

- Cassandra Query Language
- <cassandra_path>/bin/cqlsh =>consola de línea de comandos realizada en Python para interactuar de forma similar a los sistemas SQL.
- CREATE KEYSPACE demo WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1} ;
- USE demo ;
- CREATE TABLE users(user_name varchar PRIMARY KEY ,
  password varchar ,gender varchar ,
  state varchar , birth_year bigint);

- CREATE TABLE emp (empID int, deptID int,
  first_name varchar, last_name varchar,
  PRIMARY KEY (empID, deptID));

- INSERT INTO emp (empID, deptID, first_name, last_name) VALUES (104, 15, 'jane', 'smith');
- INSERT INTO emp (empID, deptID, first_name, last_name) VALUES (105, 16, 'john', 'smith');

- SELECT * FROM emp;

CiTIUS

TTL & WriteTime

- CREATE TABLE clicks (userid int, url text,
  date timestamp, name text, PRIMARY KEY (userid, url));

- INSERT INTO clicks (userid, url, date, name)
  VALUES (1, 'http://apache.org', '2013-10-09', 'Mary')
  USING TTL 50;

- SELECT TTL (name) from clicks
  WHERE url = 'http://apache.org' ALLOW FILTERING;

- INSERT INTO clicks (
  userid, url, date, name)
  VALUES (2,'http://google.com‘, '2013-10-11', 'Bob’);

- SELECT WRITETIME (date) FROM clicks WHERE url = 'http://google.com' ALLOW FILTERING;

CiTUS

Set & List

- CREATE TABLE users (user_id text PRIMARY KEY, first_name text,
- last_name text, emails set<text>);
- INSERT INTO users (user_id, first_name, last_name, emails) VALUES('frodo', 'Frodo', 'Baggins', {'f@baggins.com', 'baggins@gmail.com'});
- UPDATE users SET emails = emails + {'fb@friendsofmordor.org'} WHERE user_id = 'frodo';
- DELETE emails FROM users WHERE user_id = 'frodo';

- ALTER TABLE users ADD top_places list<text>;
- UPDATE users
  SET top_places = ['rivendell', 'rohan'] WHERE user_id = 'frodo';
- UPDATE users
  SET top_places = ['the shire'] + top_places WHERE user_id = 'frodo';
- UPDATE users
  SET top_places = top_places + ['mordor'] WHERE user_id = 'frodo';
- UPDATE users SET top_places[2] = 'riddermark' WHERE user_id = 'frodo';
- SELECT user_id, top_places FROM users WHERE user_id = 'frodo';
- DELETE top_places[3] FROM users WHERE user_id = 'frodo';

CiTUS

Map

- ALTER TABLE users ADD todo map<timestamp, text>;
- INSERT INTO users (user_id, todo)
  VALUES ('frodo',{ '2013-9-22 12:01' : 'birthday wishes to Bilbo',
  '2013-10-1 18:00' : 'Check into Inn of Prancing Pony' });
- SELECT user_id, todo FROM users WHERE user_id = 'frodo';
- UPDATE users
  SET todo = { '2012-9-24' : 'enter mordor',
  '2012-10-2 12:00' : 'throw ring into mount doom' } WHERE user_id = 'frodo';
- SELECT user_id, todo FROM users WHERE user_id = 'frodo';
- DELETE todo['2012-9-24'] FROM users WHERE user_id = 'frodo';
- SELECT user_id, todo FROM users WHERE user_id = 'frodo';

Index

- DROP TABLE users;
- CREATE TABLE users(
  user_name varchar PRIMARY KEY ,
  password varchar ,
  gender varchar ,
  state varchar ,
  birth_year bigint);
- INSERT INTO users (user_name, state , birth_year) VALUES ('david','TX',1980);
- CREATE INDEX state_key ON users (state);
- CREATE INDEX birth_year_key ON users (birth_year);
- SELECT * FROM users WHERE state='TX' AND birth_year > 1968 ALLOW FILTERING;

CiTUS

Clientes/APIs

- Cassandra tiene clientes para diferentes tecnologías
  - Python
  - Java
  - Node.js
  - Clojure
  - .NET
  - Ruby
  - PHP
  - Perl
  - Go
  - Haskell
  - C++
- En el caso de Java una alternativa interesante es Kundera, basada en el paradigma JPA 2.0.

CiTUS

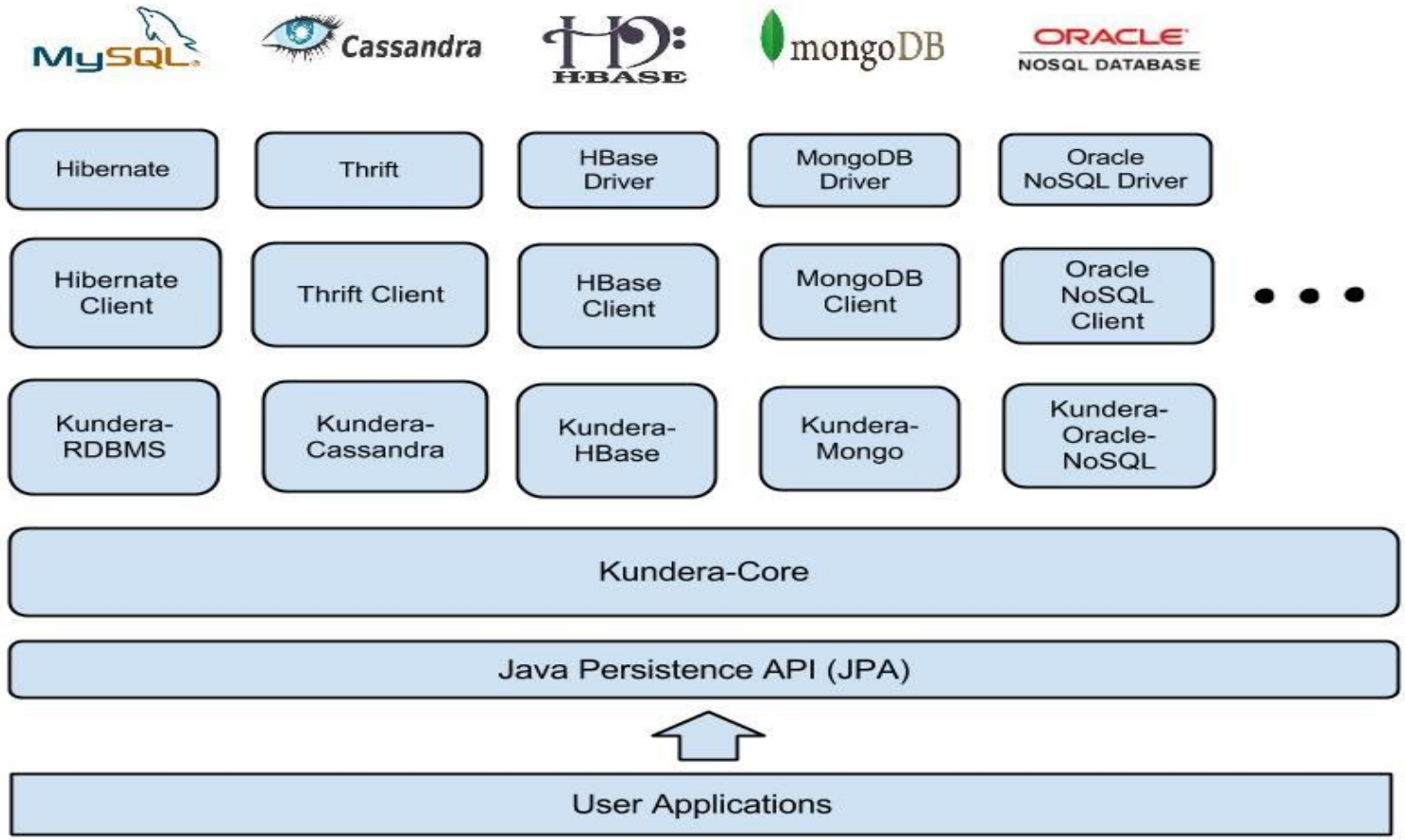
Kundera: introducción

- Kundera es una libreria java con soporte para JPA 2.0 orientada a base de datos tanto de tipo relacional comom de tipo NoSQL.
- Web: https://github.com/impetus-opensource/Kundera/wiki
- Actualmente da soporte para las siguientes DB:
  - Bases de datos relacionales: MySQL, PostgreSQL, ...
  - Apache Cassandra.
  - Apache HBase
  - MongoDB
  - Oracle NoSQL
  - Neo4j

Annotation	Cassandra	HBase	MongoDB
Table	Column Family/ Super Column Family	Table	Document (Collection)
Column	Column	Column	Column
Embeddable	Super Column	Column Family	Document (embedded in another document)

CiTIUS

Kundera: arquitectura



Kundera: configuración

- Configuracion: persistence.xml

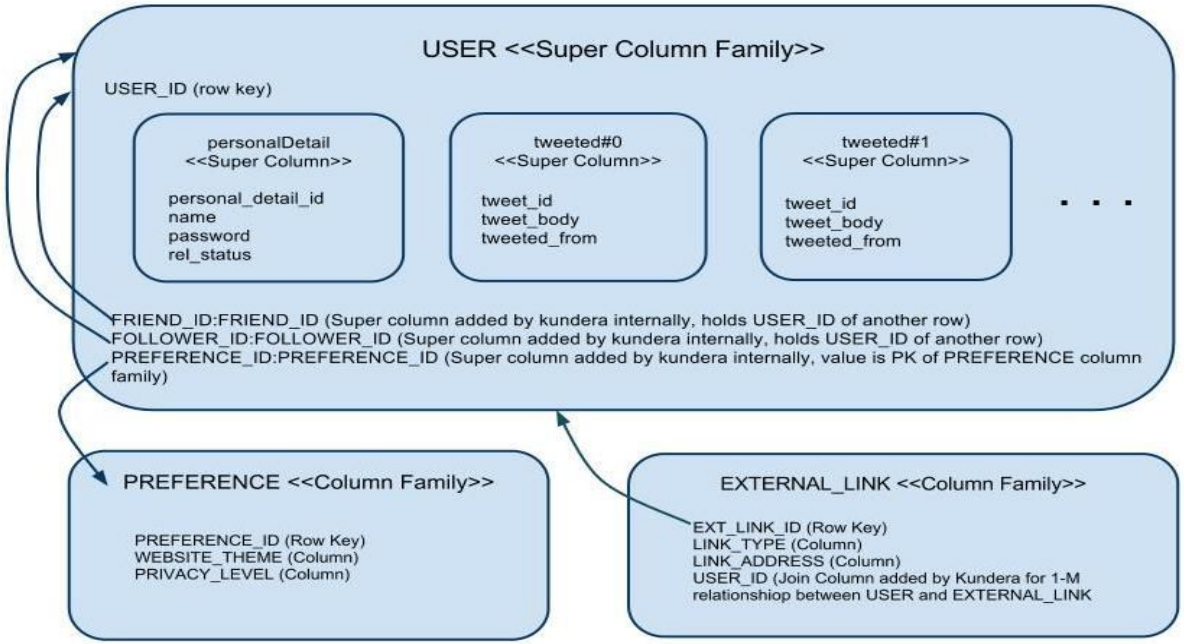
```
<persistence xmlns="http://java.sun.com/xml/ns/persistence"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/persistence
    http://java.sun.com/xml/ns/persistence/persistence_2_0.xsd"
    version="2.0">
    <persistence-unit name="cassandra_pu">
        <provider>com.impetus.kundera.KunderaPersistence</provider>
        <properties>
            <property name="kundera.nodes" value="localhost" />
            <property name="kundera.port" value="9160" />
            <property name="kundera.keyspace" value="demo" />
            <property name="kundera.dialect" value="cassandra" />
            <property name="kundera.client.lookup.class"
                value="com.impetus.client.cassandra.pelops.PelopsClientFactory" />
            <property name="kundera.cache.provider.class"
                value="com.impetus.kundera.cache.ehcache.EhCacheProvider" />
            <property name="kundera.cache.config.resource" value="/ehcache-test.xml" />
        </properties>
    </persistence-unit>
</persistence>
```

CiTIUS

Ejemplos Kundera

Ejemplos

- CRUD
  - Consultas básicas:
    - Create
    - Read
    - Update
    - Delete
- Twitter



CiTUS

Clúster Multinodo

Multinodo

- Procedimiento para inicializar un clúster multi-nodo
  - Instalar Cassandra en cada uno de los nodos.
  - Establecer un nombre para el clúster.
  - Tener disponible la IP de cada uno de los nodos.
  - Determinar cual o cuales de los nodos van a hacer de “seed”. Los “seed” se usan para determinar los nodos que forman parte del clúster y la topología por la cual están distribuidos.
  - Determinar el tipo de “snitch”. El tipo de “snitch” determina la visibilidad a nivel de “rack” y “datacenter” de los distintos nodos del clúster.
- Configuración: modificar el fichero de propiedades cassandra.yaml
  - cluster_name: <name of cluster>
  - num_tokens: <recommended value: 256>
  - -seeds: <internal IP address of each seed node>
  - listen_address: <localhost IP address>
  - rpc_address: <0.0.0.0>
  - endpoint_snitch: <name of snitch>

CiTIUS

Gracias por su atención

Unidad de Innovación: citius.idi@usc.es

citius.usc.es