

Pràctica 2 Bases de Dades Avançades

PRÀCTICA 2 – MONGO DB
PAU SANCHEZ I GUILLEM RABIONET

Introducció

L'objectiu d'aquesta pràctica és que l'alumnat es pugui familiaritzar amb els sistemes d'emmagatzematge de tipus NoSQL, en el cas ens ocupa, concretament, MongoDB, alhora que es pren contacte amb apache-flume, una eina BigData que s'utilitzarà per a dur a terme la ingesta de dades. I es farà de dues maneres:

- Des de fitxers de dades
- Ingerint les dades en 'temps real'

La ingesta de dades en temps real, a més, es farà de diverses formes, tals com poden ésser 'telnet' o 'Packet Sender', inserint directament línies de text o bé carregant en temps real un fitxer.

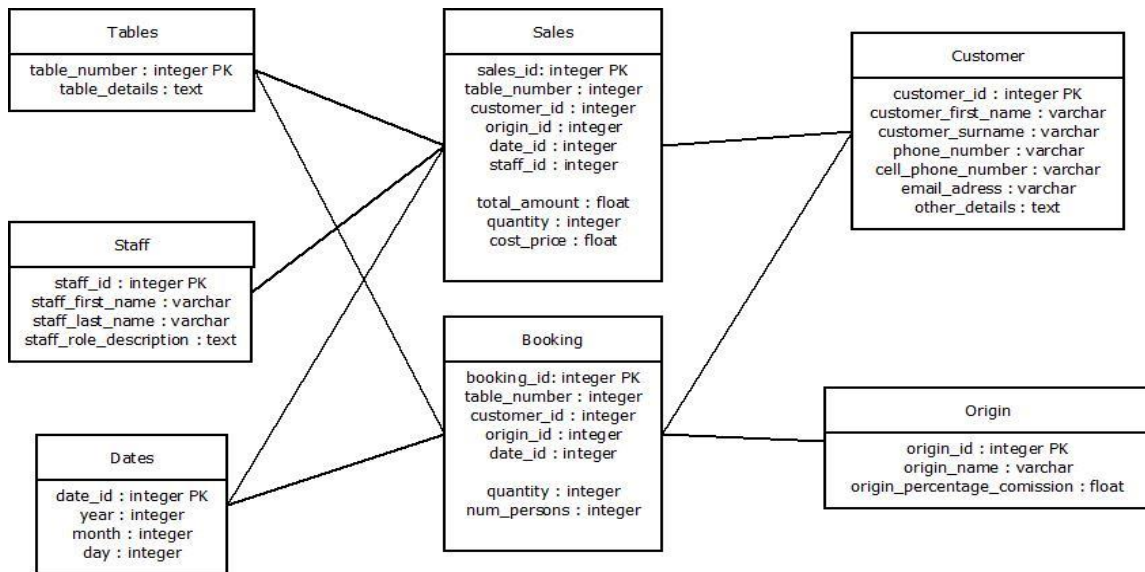
La idea de l'entrega no és altra que demostrar la capacitat de l'alumnat alhora de recrear les taules del datamart l'entrega anterior modelades en MongoDB, a més de la ingesta, explicada al paràgraf anterior.

MongoDB

MongoDB és un sistema de bases de dades NoSQL orientat a documents i desenvolupat sota el concepte de codi obert. La particularitat és que enlloc de guardar les dades en taules, com es fa a les bases de dades relacionals, Mongo guarda estructures de dades en documents de tipus JSON (JavaScript Object Notation) amb un esquema dinàmic BSON fent que la integració de les dades en certes aplicacions sigui més fàcil i ràpida.

Datamart de MenjaUB

Tal i com proposa l'enunciat de la pràctica, el que cal realitzar, no és altra cosa que el modelat de la base de dades MenjaUB realitzada a la pràctica anterior. Es mostra a continuació:



Val a dir que MongoDB no obliga a tenir estructures de d'atributs fixes sinó que permet diferents estructures de fila en una mateixa taula, per exemple, fixant la mirada a la taula 'Tables' del diagrama anterior, es podria perfectament afegir una fila que tingués tres atributs enlloc de dos i no passaria res.

Amb la particularitat anterior es podria gairebé afirmar que el modelat en MongoDB depèn de les files i no de les taules, en altres paraules, Mongo s'empassa tot allò que se li doni per emmagatzemar sense queixar-se, sempre i quan reconegui el format.

Configuració de MongoDB

MongoDB: per configurar MongoDB, es pot editar el fitxer de configuració mateixa carpeta, es troba una arxiu .bat que executa la següent línia de comanda: config setup.cfg, cridant al fitxer mencionat anteriorment següent informació:

```

systemLog:
  destination: file
  path: ..\data\database.log
storage:
  dbPath: ..\data\db
  directoryPerDB: true
  journal:
    enabled: true
  engine: mmapv1
  
```

Aquest fitxer permet configurar l'adreça destí del log i del l'storage, així com altres paràmetres de la configuració.

La instrucció bàsica per fer ús de l'anterior setup alhora d'executar mongo és la següent:

mongod --config setup.cfg

Una possibilitat és col·locar-la dins un fitxer d'extensió '.bat' per executar-la amb més facilitat, en el cas de la pràctica: *start.bat*

Configuració d'Apache-Flume

El primer que cal fer és adreçar el fitxer *start-agent.bat* a la ruta del java pertinent a la computadora en la qual s'està executant. A més, en aquest mateig fitxer té lloc el mapeig de l'agent que utilitzarà flume per enviar la informació. La següent imatge ho mostra de manera explícita:

```
set JAVA_HOME=C:\Program Files\Java\jre1.8.0_91
flume-ng agent -n agent1 -f ..\conf\flume-conf.properties|
```

La imatge anterior mostra un agent anomenat 'agent1'. Aquest serà l'encarregat d'ingerir les dades des de fitxers, per gestionar la ingesta en temps real s'ha creat un altre fitxer *start-agent2.bat* el qual mapa un altre agent per a dur a terme la ingesta en temps real, s'anomena 'agent2':

```
set JAVA_HOME=C:\Program Files\Java\jre1.8.0_91
flume-ng agent -n agent2 -f ..\conf\flume-conf.properties|
```

Aquests fitxers, però, no fan altra cosa que executar dues línies que bé es podrien escriure directament per línia de comandes, ja que tenen extensió '.bat'. El fitxer que realment conté els

aspectes més rellevants de la configuració no és altre que *flume-conf.properties*. Aquest és el que permet fer les ingestes de dades via fitxers o via Telnet en temps real, configurant adientment, és clar, les fonts, els canals i els agents.

Ingesta de dades des de FITXERS

Configurant les fonts, els canals i els sinks de l'agent 1:

Primer cal indicar quines fonts, canals i sinks tindrà l'agent:

```
##### Agent MONGODB ###  
agent1.sources = sourceTables sourceCustomers sourceStaff sourceDates sourceOrigin sourceBooking sourceSales  
agent1.channels = channelTables channelCustomers channelStaff channelDates channelOrigin channelBooking channelSales  
agent1.sinks = sinkMongoTables sinkMongoCustomers sinkMongoStaff sinkMongoDates sinkMongoOrigin sinkMongoBooking sinkMongoSales
```

El pas següent és configurar les fonts. Cal indicar el tipus de directori 'spooldir', el canal a utilitzar i per últim, el fitxer des d'on es farà la ingesta. Cal repetir aquest pas per cada canal i cada taula tal i com es mostra a continuació:

```
# sourceBooking config  
agent1.sources.sourceBooking.type = spooldir  
agent1.sources.sourceBooking.channels = channelBooking  
agent1.sources.sourceBooking.spoolDir = ../data_menjaub/facts/booking  
  
# sourceSales config  
agent1.sources.sourceSales.type = spooldir  
agent1.sources.sourceSales.channels = channelSales  
agent1.sources.sourceSales.spoolDir = ../data_menjaub/facts/sales  
  
# SourceTables config  
agent1.sources.sourceTables.type = spooldir  
agent1.sources.sourceTables.channels = channelTables  
agent1.sources.sourceTables.spoolDir = ../data_menjaub/dimensions/tables  
  
# sourceCustomers config  
agent1.sources.sourceCustomers.type = spooldir  
agent1.sources.sourceCustomers.channels = channelCustomers  
agent1.sources.sourceCustomers.spoolDir = ../data_menjaub/dimensions/customers  
  
# sourceStaff config  
agent1.sources.sourceStaff.type = spooldir  
agent1.sources.sourceStaff.channels = channelStaff  
agent1.sources.sourceStaff.spoolDir = ../data_menjaub/dimensions/staff  
  
# sourceDates config  
agent1.sources.sourceDates.type = spooldir  
agent1.sources.sourceDates.channels = channelDates  
agent1.sources.sourceDates.spoolDir = ../data_menjaub/dimensions/dates  
  
# sourceOrigin config  
agent1.sources.sourceOrigin.type = spooldir  
agent1.sources.sourceOrigin.channels = channelOrigin  
agent1.sources.sourceOrigin.spoolDir = ../data_menjaub/dimensions/origin
```

Com es pot apreciar, s'ha creat un directori data_menjaub que conté els fitxers a ingerir.

Ara cal configurar els canals d'enviament. Es fa de la següent manera:

```
# sourceBooking config
agent1.sources.sourceBooking.type = spooldir
agent1.sources.sourceBooking.channels = channelBooking
agent1.sources.sourceBooking.spoolDir = ../data_menjaub/facts/booking

# sourceSales config
agent1.sources.sourceSales.type = spooldir
agent1.sources.sourceSales.channels = channelSales
agent1.sources.sourceSales.spoolDir = ../data_menjaub/facts/sales

# SourceTables config
agent1.sources.sourceTables.type = spooldir
agent1.sources.sourceTables.channels = channelTables
agent1.sources.sourceTables.spoolDir = ../data_menjaub/dimensions/tables

# sourceCustomers config
agent1.sources.sourceCustomers.type = spooldir
agent1.sources.sourceCustomers.channels = channelCustomers
agent1.sources.sourceCustomers.spoolDir = ../data_menjaub/dimensions/customers

# sourceStaff config
agent1.sources.sourceStaff.type = spooldir
agent1.sources.sourceStaff.channels = channelStaff
agent1.sources.sourceStaff.spoolDir = ../data_menjaub/dimensions/staff

# sourceDates config
agent1.sources.sourceDates.type = spooldir
agent1.sources.sourceDates.channels = channelDates
agent1.sources.sourceDates.spoolDir = ../data_menjaub/dimensions/dates

# sourceOrigin config
agent1.sources.sourceOrigin.type = spooldir
agent1.sources.sourceOrigin.channels = channelOrigin
agent1.sources.sourceOrigin.spoolDir = ../data_menjaub/dimensions/origin
```

Ara només cal sincronitzar els canals amb les taules de la base de dades de MongoDB mitjançant els sinks:

```
# Flume NG MongoDB sinkMongoTables Config
agent1.sinks.sinkMongoTables.channel = channelTables
agent1.sinks.sinkMongoTables.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoTables.host = localhost
agent1.sinks.sinkMongoTables.port = 27017
agent1.sinks.sinkMongoTables.model = SINGLE
agent1.sinks.sinkMongoTables.db = menjaUB
agent1.sinks.sinkMongoTables.collection = tables
agent1.sinks.sinkMongoTables.batch = 100
agent1.sinks.sinkMongoTables.timestampField = "yyyy-MM-dd HH:mm:ss"

# Flume NG MongoDB sinkMongoCustomers Config
agent1.sinks.sinkMongoCustomers.channel = channelCustomers
agent1.sinks.sinkMongoCustomers.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoCustomers.host = localhost
agent1.sinks.sinkMongoCustomers.port = 27017
agent1.sinks.sinkMongoCustomers.model = SINGLE
agent1.sinks.sinkMongoCustomers.db = menjaUB
agent1.sinks.sinkMongoCustomers.collection = customers
agent1.sinks.sinkMongoCustomers.batch = 100
agent1.sinks.sinkMongoCustomers.timestampField = "yyyy-MM-dd HH:mm:ss"

# Flume NG MongoDB sinkMongoStaff Config
agent1.sinks.sinkMongoStaff.channel = channelStaff
agent1.sinks.sinkMongoStaff.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoStaff.host = localhost
agent1.sinks.sinkMongoStaff.port = 27017
agent1.sinks.sinkMongoStaff.model = SINGLE
agent1.sinks.sinkMongoStaff.db = menjaUB
agent1.sinks.sinkMongoStaff.collection = staff
agent1.sinks.sinkMongoStaff.batch = 100
agent1.sinks.sinkMongoStaff.timestampField = "yyyy-MM-dd HH:mm:ss"

# Flume NG MongoDB sinkMongoDates Config
agent1.sinks.sinkMongoDates.channel = channelDates
agent1.sinks.sinkMongoDates.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoDates.host = localhost
agent1.sinks.sinkMongoDates.port = 27017
agent1.sinks.sinkMongoDates.model = SINGLE
agent1.sinks.sinkMongoDates.db = menjaUB
agent1.sinks.sinkMongoDates.collection = dates
agent1.sinks.sinkMongoDates.batch = 100
agent1.sinks.sinkMongoDates.timestampField = "yyyy-MM-dd HH:mm:ss"
```

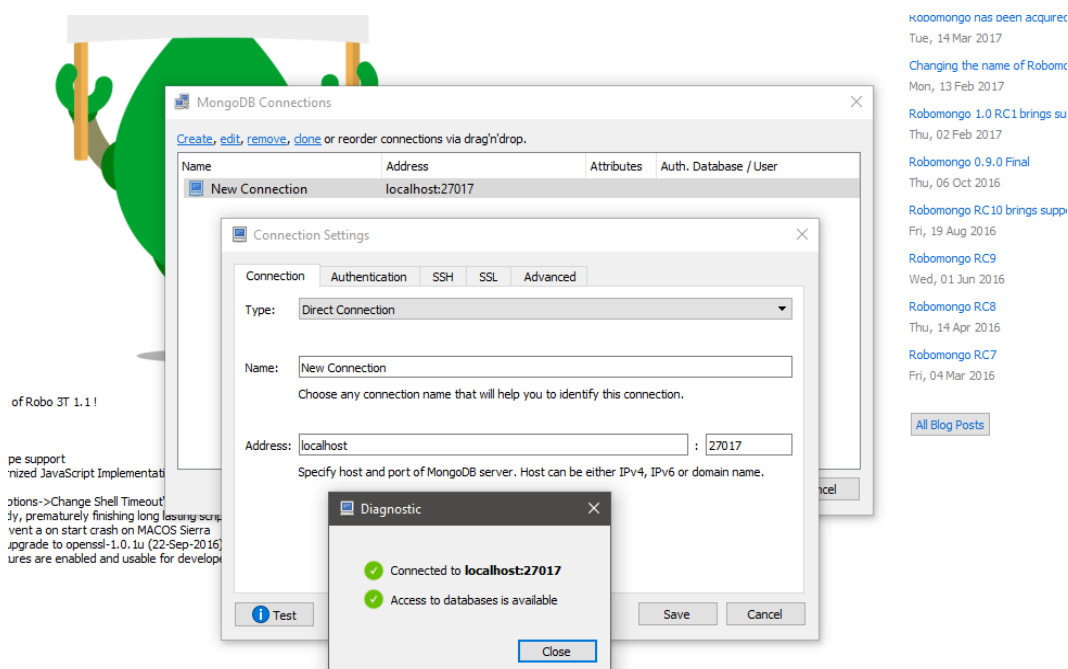
```
# Flume NG MongoDB sinkMongoOrigin Config
agent1.sinks.sinkMongoOrigin.channel = channelOrigin
agent1.sinks.sinkMongoOrigin.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoOrigin.host = localhost
agent1.sinks.sinkMongoOrigin.port = 27017
agent1.sinks.sinkMongoOrigin.model = SINGLE
agent1.sinks.sinkMongoOrigin.db = menjaUB
agent1.sinks.sinkMongoOrigin.collection = origin
agent1.sinks.sinkMongoOrigin.batch = 100
agent1.sinks.sinkMongoOrigin.timestampField = "yyyy-MM-dd HH:mm:ss"

# Flume NG MongoDB sinkMongoBooking Config
agent1.sinks.sinkMongoBooking.channel = channelBooking
agent1.sinks.sinkMongoBooking.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoBooking.host = localhost
agent1.sinks.sinkMongoBooking.port = 27017
agent1.sinks.sinkMongoBooking.model = SINGLE
agent1.sinks.sinkMongoBooking.db = menjaUB
agent1.sinks.sinkMongoBooking.collection = booking
agent1.sinks.sinkMongoBooking.batch = 100
agent1.sinks.sinkMongoBooking.timestampField = "yyyy-MM-dd HH:mm:ss"

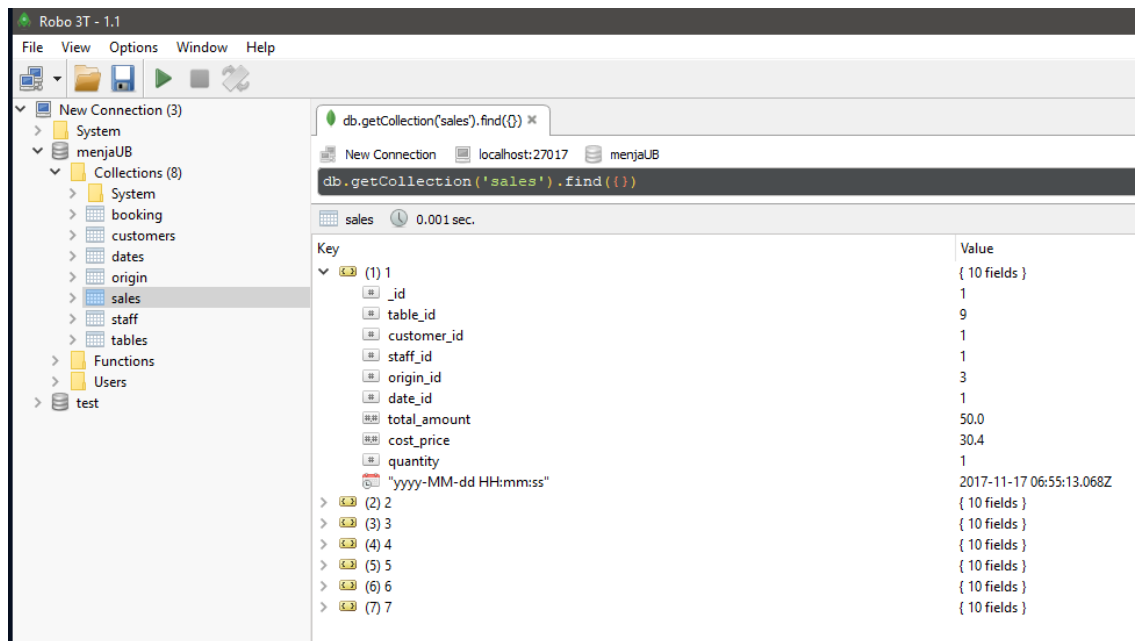
# Flume NG MongoDB sinkMongoSales Config
agent1.sinks.sinkMongoSales.channel = channelSales
agent1.sinks.sinkMongoSales.type = org.riderzen.flume.sink.MongoSink
agent1.sinks.sinkMongoSales.host = localhost
agent1.sinks.sinkMongoSales.port = 27017
agent1.sinks.sinkMongoSales.model = SINGLE
agent1.sinks.sinkMongoSales.db = menjaUB
agent1.sinks.sinkMongoSales.collection = sales
agent1.sinks.sinkMongoSales.batch = 100
agent1.sinks.sinkMongoSales.timestampField = "yyyy-MM-dd HH:mm:ss"
```

Per veure el funcionament complet del que s'ha explicat al paràgraf anterior, el que cal ara és posar en marxa el programa RoboMongo, anomenat ara Robo 3T i comprovar que la ingesta funciona.

El primer que cal és connectar la base de dades al port indicat al fitxer de configuració i testar que la connexió funciona correctament:



Després cal crear la base de dades i les collections (taules) que contindrà.



Una vegada fet això i executat el flume, podem veure que ha funcionat! A la imatge anterior es pot apreciar la taula 'sales' plena de la informació que ha xuclat MongoDB via Flume dels fitxers de dades de tipus JSON.

Ara el que es pot fer per corroborar l'èxit és buidar la taula i eliminar l'extensió '.COMPLETED' del fitxer d'ingesta per veure que, efectivament, les dades es tornen a inserir via flume.

Mantenint el flume obert s'aprecia que l'extensió '.COMPLETED' torna a generar-se automàticament tot i indicant na nova ingesta per flume.

Ingesta de dades en temps real via TELNET i PACKET SENDER

El primer es mostra és la connexió per Telnet a un dels ports assignats al fitxer flume-conf.properties. Per aquest exemple s'utilitza el port 44443 per afegir dades a la taula 'Tables' de la base de dades:


```

Microsoft Windows [Versión 10.0.15063]
(c) 2017 Microsoft Corporation. Todos los derechos reservados.

C:\Users\pausanchezv>telnet localhost 44443

```

Una vegada la connexió s'ha establert es procedeix a fer la prova d'ingerir dades en temps real:

```

OK
{"_id":"222","attr":"Atribut de prova"}
OK

```

Com bé s'ha esmentat anteriorment, aquest exemple corrobora el fet que MongoDB és una base de dades heterogènia. S'afegeix un document (fila) amb només dos atributs i tot seguit es mostra com ha funcionat i la fila (document) ha estat inserit:

Key	Value
> (1) 1	{ 3 fields }
> (2) 2	{ 3 fields }
> (3) 3	{ 3 fields }
> (4) 4	{ 3 fields }
> (5) 5	{ 3 fields }
> (6) 6	{ 3 fields }
> (7) 7	{ 3 fields }
> (8) 8	{ 3 fields }
> (9) 9	{ 3 fields }
> (10) 10	{ 3 fields }
▼ (11) 222	
_id	222
attr	Atribut de prova

Tot seguit es mostra la configuració afegida al fitxer flume-conf.properties que ha fet possible la connexió i ingesta anteriors.

Configuració de l'agent-2

```
##### Agent2 MONGODB ###  
agent2.sources = sourceTables sourceCustomers sourceStaff sourceDates sourceOrigin sourceBooking sourceSales  
agent2.channels = channelTables channelCustomers channelStaff channelDates channelOrigin channelBooking channelSales  
agent2.sinks = sinkMongoTables sinkMongoCustomers sinkMongoStaff sinkMongoDates sinkMongoOrigin sinkMongoBooking sinkMongoSales
```

Configuració de les fonts

```
### Sources config  
  
agent2.sources.sourceTables.type = netcat  
agent2.sources.sourceTables.channels = channelTables  
agent2.sources.sourceTables.bind = localhost  
agent2.sources.sourceTables.port = 44443  
  
agent2.sources.sourceCustomers.type = netcat  
agent2.sources.sourceCustomers.channels = channelCustomers  
agent2.sources.sourceCustomers.bind = localhost  
agent2.sources.sourceCustomers.port = 44444  
  
agent2.sources.sourceStaff.type = netcat  
agent2.sources.sourceStaff.channels = channelStaff  
agent2.sources.sourceStaff.bind = localhost  
agent2.sources.sourceStaff.port = 44445  
  
agent2.sources.sourceDates.type = netcat  
agent2.sources.sourceDates.channels = channelDates  
agent2.sources.sourceDates.bind = localhost  
agent2.sources.sourceDates.port = 44446  
  
agent2.sources.sourceOrigin.type = netcat  
agent2.sources.sourceOrigin.channels = channelOrigin  
agent2.sources.sourceOrigin.bind = localhost  
agent2.sources.sourceOrigin.port = 44447  
  
agent2.sources.sourceBooking.type = netcat  
agent2.sources.sourceBooking.channels = channelBooking  
agent2.sources.sourceBooking.bind = localhost  
agent2.sources.sourceBooking.port = 44448  
  
agent2.sources.sourceSales.type = netcat  
agent2.sources.sourceSales.channels = channelSales  
agent2.sources.sourceSales.bind = localhost  
agent2.sources.sourceSales.port = 44449
```

Configuració dels canals

```
# ChannelsConfig  
  
agent2.channels.channelTables.type = memory  
agent2.channels.channelTables.capacity = 1000000  
agent2.channels.channelTables.transactionCapacity = 80000  
  
agent2.channels.channelCustomers.type = memory  
agent2.channels.channelCustomers.capacity = 1000000  
agent2.channels.channelCustomers.transactionCapacity = 80000  
  
agent2.channels.channelStaff.type = memory  
agent2.channels.channelStaff.capacity = 1000000  
agent2.channels.channelStaff.transactionCapacity = 80000  
  
agent2.channels.channelDates.type = memory  
agent2.channels.channelDates.capacity = 1000000  
agent2.channels.channelDates.transactionCapacity = 80000  
  
agent2.channels.channelOrigin.type = memory  
agent2.channels.channelOrigin.capacity = 1000000  
agent2.channels.channelOrigin.transactionCapacity = 80000  
  
agent2.channels.channelBooking.type = memory  
agent2.channels.channelBooking.capacity = 1000000  
agent2.channels.channelBooking.transactionCapacity = 80000  
  
agent2.channels.channelSales.type = memory  
agent2.channels.channelSales.capacity = 1000000  
agent2.channels.channelSales.transactionCapacity = 80000
```

Configuració dels 'sinks'

```
# Sinks conig

agent2.sinks.sinkMongoTables.channel = channelTables
agent2.sinks.sinkMongoTables.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoTables.host = localhost
agent2.sinks.sinkMongoTables.port = 27017
agent2.sinks.sinkMongoTables.model = SINGLE
agent2.sinks.sinkMongoTables.db = menjaUB
agent2.sinks.sinkMongoTables.collection = tables
agent2.sinks.sinkMongoTables.batch = 100
agent2.sinks.sinkMongoTables.timestampField = "yyyy-MM-dd HH:mm:ss"

agent2.sinks.sinkMongoCustomers.channel = channelCustomers
agent2.sinks.sinkMongoCustomers.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoCustomers.host = localhost
agent2.sinks.sinkMongoCustomers.port = 27017
agent2.sinks.sinkMongoCustomers.model = SINGLE
agent2.sinks.sinkMongoCustomers.db = menjaUB
agent2.sinks.sinkMongoCustomers.collection = customers
agent2.sinks.sinkMongoCustomers.batch = 100
agent2.sinks.sinkMongoCustomers.timestampField = "yyyy-MM-dd HH:mm:ss"

agent2.sinks.sinkMongoStaff.channel = channelStaff
agent2.sinks.sinkMongoStaff.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoStaff.host = localhost
agent2.sinks.sinkMongoStaff.port = 27017
agent2.sinks.sinkMongoStaff.model = SINGLE
agent2.sinks.sinkMongoStaff.db = menjaUB
agent2.sinks.sinkMongoStaff.collection = staff
agent2.sinks.sinkMongoStaff.batch = 100
agent2.sinks.sinkMongoStaff.timestampField = "yyyy-MM-dd HH:mm:ss"

agent2.sinks.sinkMongoDates.channel = channelDates
agent2.sinks.sinkMongoDates.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoDates.host = localhost
agent2.sinks.sinkMongoDates.port = 27017
agent2.sinks.sinkMongoDates.model = SINGLE
agent2.sinks.sinkMongoDates.db = menjaUB
agent2.sinks.sinkMongoDates.collection = dates
agent2.sinks.sinkMongoDates.batch = 100
agent2.sinks.sinkMongoDates.timestampField = "yyyy-MM-dd HH:mm:ss"
```

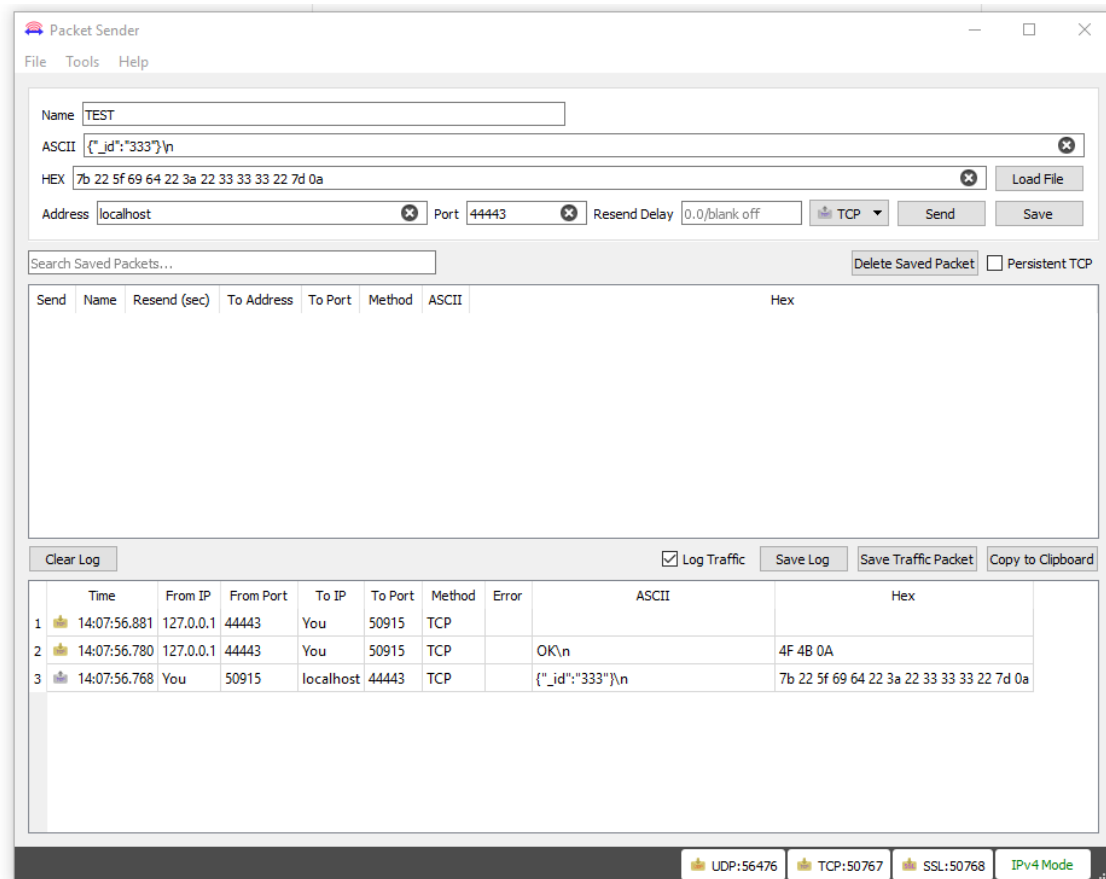
```
agent2.sinks.sinkMongoOrigin.channel = channelOrigin
agent2.sinks.sinkMongoOrigin.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoOrigin.host = localhost
agent2.sinks.sinkMongoOrigin.port = 27017
agent2.sinks.sinkMongoOrigin.model = SINGLE
agent2.sinks.sinkMongoOrigin.db = menjaUB
agent2.sinks.sinkMongoOrigin.collection = origin
agent2.sinks.sinkMongoOrigin.batch = 100
agent2.sinks.sinkMongoOrigin.timestampField = "yyyy-MM-dd HH:mm:ss"

agent2.sinks.sinkMongoBooking.channel = channelBooking
agent2.sinks.sinkMongoBooking.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoBooking.host = localhost
agent2.sinks.sinkMongoBooking.port = 27017
agent2.sinks.sinkMongoBooking.model = SINGLE
agent2.sinks.sinkMongoBooking.db = menjaUB
agent2.sinks.sinkMongoBooking.collection = booking
agent2.sinks.sinkMongoBooking.batch = 100
agent2.sinks.sinkMongoBooking.timestampField = "yyyy-MM-dd HH:mm:ss"

agent2.sinks.sinkMongoSales.channel = channelSales
agent2.sinks.sinkMongoSales.type = org.riderzen.flume.sink.MongoSink
agent2.sinks.sinkMongoSales.host = localhost
agent2.sinks.sinkMongoSales.port = 27017
agent2.sinks.sinkMongoSales.model = SINGLE
agent2.sinks.sinkMongoSales.db = menjaUB
agent2.sinks.sinkMongoSales.collection = sales
agent2.sinks.sinkMongoSales.batch = 100
agent2.sinks.sinkMongoSales.timestampField = "yyyy-MM-dd HH:mm:ss"
```

Una vegada tenim Telnet funcionant, el que cal és ingerir les dades via Packet Sender tal i com va proposar el professor de l'assignatura, l'Enric Biosca. L'objectiu d'això és poder ingerir en temps real via text i via fitxers.

Vegem, doncs, la ingesta via text a través de Packet Sender en primer lloc:



El que s'ha fet ha estat escriure el port i l'ascii corresponent al fragment de text que es pretén enviar. Lògicament el text ha de tenir un format que l'interpret de flume i mongo sigui capaç d'entendre, aquest no és pas altre que JSON.

Un detall que ha donat molts mals de cap ha estat el fet que s'ha d'afegir obligatòriament el caràcter de finalització de línia '\n' perquè el programa sigui capaç d'enviar de rebre les dades, doncs enviar-se s'enviarien igual sense el caràcter.

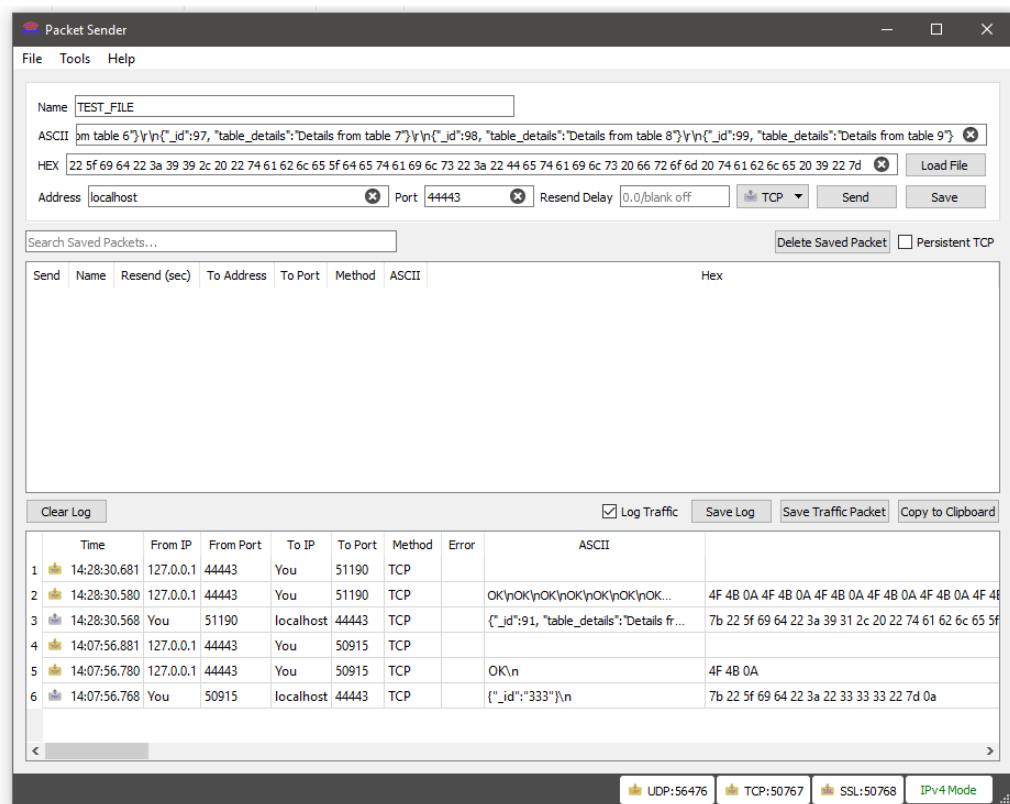
Ingesta en temps real des de fitxers utilitzant Paquet Sender

Una vegada testat el bon funcionament del programa Paquet Sender, el que cal és ser capaços de transferir grans volums de dades en temps real inserint directament des de fitxers.

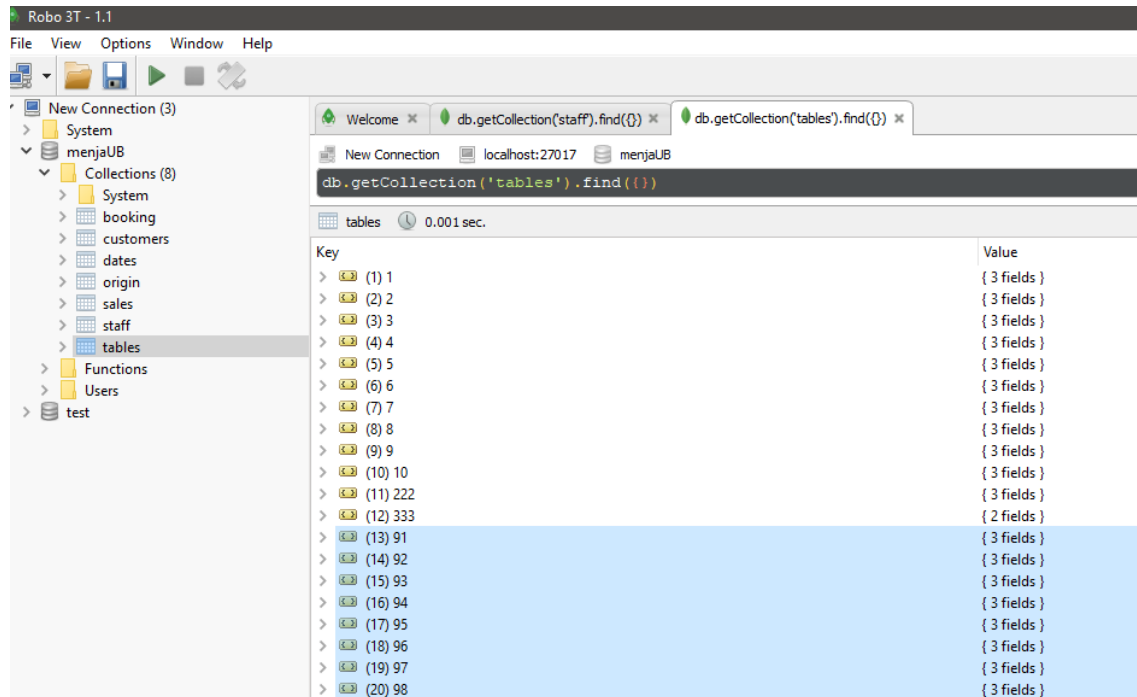
Per això hem deixat preparat un joc de proves perquè el professorat pugui testar la ingesta a partir de fitxers amb el packet sender. Aquest joc es troba a la carpeta RT_test que s'adjunta amb la documentació de la pràctica, i conté els fitxers amb extensió JSON que seran ingerits mitjançant paquet sender.

Per seguir amb l'exemple, utilitzarem la taula 'tables' per comprovar, en aquest document, que la ingesta via fitxers funciona correctament.

Una vegada seleccionat el fitxer des de packet sender, aquest és absorbit i transformat en text, tal i com mostra el camp ascii de la següent imatge:



Una vegada s'ha fet click sobre el botó 'Send' es comprova que la ingesta en temps real des de packet sender ha funcionat mirant directament la base de dades MongoDB:



Els documents (files) marcadés en blau són els que s'han ingerit des de fitxers via Packet Sender.

