SWEeneyThreads

Actorbase

a NoSQL DB based on the Actor model

User Manual

Redattori:
Maino Elia
Bortolazzo Matteo

Approvazione:
Bonato Paolo
Verifica:
Padovan Tommaso

Versione 1.0.0

16 maggio 2016

Indice

History log

Version	Date	Author	Description
1.0.0	2016-05-12	*Responsible* Bonato Paolo	Document approved.
0.1.0	2016-05-12	*Verifier* Padovan Paolo	Document verified.
0.0.2	2016-05-12	*Designer* Bortolazzo Matteo	Document completed.
0.0.1	2016-05-11	*Designer* Maino Elia	Created document structure, written the introduction and the server application section.

Tabella 1: History log

1 Introduction

1.1 Document's purpose

The present document represents the user manual for the use of *Actorbase*, a NoSQL database. All the user application's features will be described in detail. The manual is divided into three main sections, related to the main components of the product: Client, Server and Driver.

1.2 Product's purpose

The project's purpose is the development of a key-value NoSQL Database based on the actor model, with the goal of supply an appropriate technology for the development of modern applications which request very short response time, and which elaborate substantial amount of data. The development will lead to the release of the software under MIT license.

2 Actorbase

Actorbase is a key-value NoSQL database based on the actor model, which guarantees high level of scalability, resilience and performance. It allows to manage easily and flexibly your data, using the main advantages offered by the actor model, in order to support the development of modern and performing applications.

Actorbase provides a command line client interface which offers an easy way to handle data as strings. With the CLI is possible to communicate quickly and intuitively with a server.

For more flexible queries it's avaiable the *Scala* driver, integrable in every Java application.

Figura 1: Actorbase Logo

3 System requirements

The correct execution of *Actorbase* is guaranteed on machines that meet the subsequent hardware and software specifications.

OS:

- Windows 7 or newer
- OS X 10.7 or newer
- Ubuntu 14.04 or newer

Java Virtual Machine (JVM) 8 or newer.

RAM:

- Client application: 2GB minimum
- Server application: 4GB minimum, 8GB recommended

There aren't explicit requirements for the processor architecture or speed. Using a very old processor could slow down the system performances.

4 Installation

Actorbase runs on JVM, that's the reason why it doesn't need an installation procedure (just click on the launch icon).

5 Server application

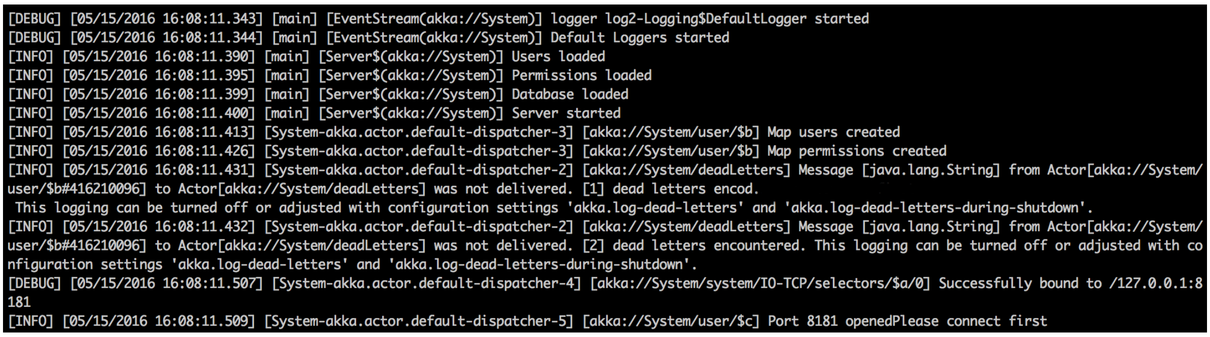
The server application allows the user to run and manage an *Actorbase* server instance on the machine. Once started, the server allows clients to connect to the machine, and to query the database.

5.1 Server configuration

The server machine's configuration is made by changing `server.conf` file. The configuration file allows the administrator to set the IP addresses and the connection ports
If the user tries to run the server application without having defined the configuration parameters, the server will display an error message and won't start.

5.2 Server interface

Once the administrator has modified the configuration file it is possible to start the application by click on the icon. The server application provides the user a command line interface which displays the operation log in real time.

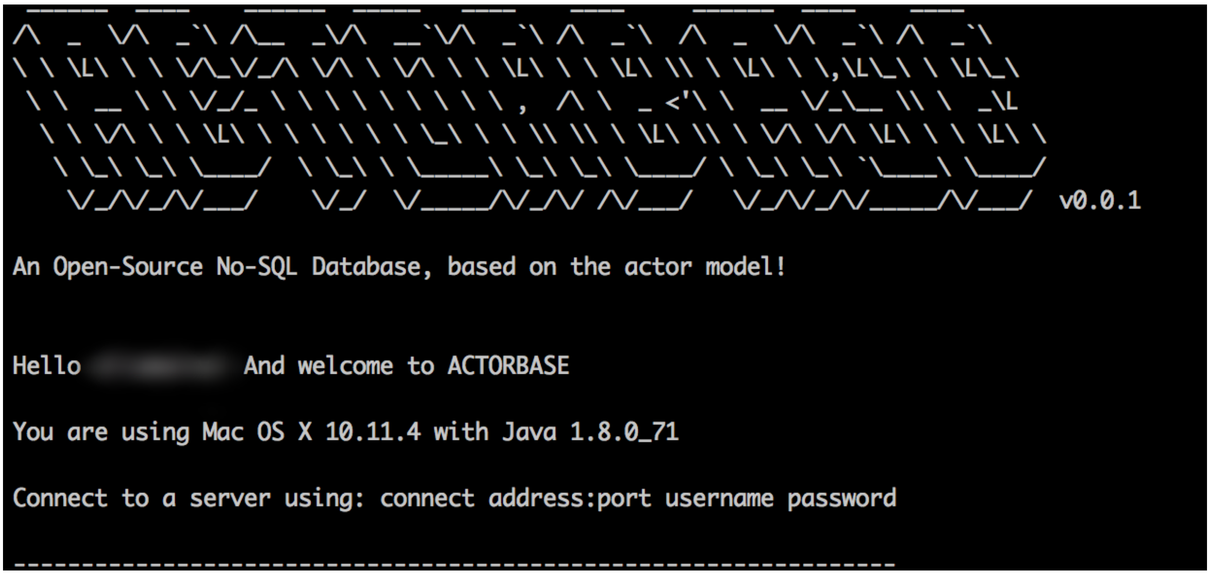


```
[DEBUG] [05/15/2016 16:08:11.343] [main] [EventStream(akka://System)] logger log2-Logging$DefaultLogger started
[DEBUG] [05/15/2016 16:08:11.344] [main] [EventStream(akka://System)] Default Loggers started
[INFO] [05/15/2016 16:08:11.390] [main] [Server$(akka://System)] Users loaded
[INFO] [05/15/2016 16:08:11.395] [main] [Server$(akka://System)] Permissions loaded
[INFO] [05/15/2016 16:08:11.399] [main] [Server$(akka://System)] Database loaded
[INFO] [05/15/2016 16:08:11.400] [main] [Server$(akka://System)] Server started
[INFO] [05/15/2016 16:08:11.413] [System-akka.actor.default-dispatcher-3] [akka://System/user/$b] Map users created
[INFO] [05/15/2016 16:08:11.426] [System-akka.actor.default-dispatcher-3] [akka://System/user/$b] Map permissions created
[INFO] [05/15/2016 16:08:11.431] [System-akka.actor.default-dispatcher-2] [akka://System/deadLetters] Message [java.lang.String] from Actor[akka://System/
user/$b#416210096] to Actor[akka://System/deadLetters] was not delivered. [1] dead letters encod.
 This logging can be turned off or adjusted with configuration settings 'akka.log-dead-letters' and 'akka.log-dead-letters-during-shutdown'.
[INFO] [05/15/2016 16:08:11.432] [System-akka.actor.default-dispatcher-2] [akka://System/deadLetters] Message [java.lang.String] from Actor[akka://System/
user/$b#416210096] to Actor[akka://System/deadLetters] was not delivered. [2] dead letters encountered. This logging can be turned off or adjusted with co
nfiguration settings 'akka.log-dead-letters' and 'akka.log-dead-letters-during-shutdown'.
[DEBUG] [05/15/2016 16:08:11.507] [System-akka.actor.default-dispatcher-4] [akka://System/system/IO-TCP/selectors/$a/0] Successfully bound to /127.0.0.1:8
181
[INFO] [05/15/2016 16:08:11.509] [System-akka.actor.default-dispatcher-5] [akka://System/user/$c] Port 8181 openedPlease connect first
```

Figura 2: Server's log interface

6 Client application

The client application provides a command line interface to connect to an *Actorbase* server and to query it. To start the client just double-click on the client's icon. Once started, the client presents a welcome banner followed by a brief description of the software's configuration used (JVM version and operating system).



```
 ______  ____    ______  _____   ____    ____     ______  ____    ____
/\  _  \/\  _`\ /\__  _\/\  __`\/\  _`\ /\  _`\  /\  _  \/\  _`\ /\  _`\
\ \ \L\ \ \ \/\_\/_/\ \/\ \ \/\ \ \ \L\ \ \ \L\ \\ \ \L\ \ \,\L\_\ \ \L\_\
 \ \  __ \ \ \/_/_ \ \ \ \ \ \ \ \ \ ,  /\ \  _ <'\ \  __ \/_\__ \\ \  _\L
  \ \ \/\ \ \ \L\ \ \ \ \ \ \ \_\ \ \ \\ \\ \ \L\ \\ \ \/\ \/\ \L\ \ \ \L\ \
   \ \_\ \_\ \____/  \ \_\ \ \_____\ \_\ \_\ \____/ \ \_\ \_\ `\____\ \____/
    \/_/\/_/\/___/    \/_/  \/_____/\/_/\/ /\/___/   \/_/\/_/\/_____/\/___/  v0.0.1

An Open-Source No-SQL Database, based on the actor model!


Hello [redacted] And welcome to ACTORBASE

You are using Mac OS X 10.11.4 with Java 1.8.0_71

Connect to a server using: connect address:port username password

------------------------------------------------------------
```

Figura 3: Client's welcome banner

The interaction with the client interface is made with textual commands. They can be composed of multiple fields, separated by a space character.

6.1 Connection management

Once the client is running the first thing to do is to connect to a server. The connection management is based on two commands: `connect` and `disconnect`. The application doesn't allow to manage more than one connection at time.

6.1.1 `connect` command

This command allows the user to connect to a server, the command structure is:

```
actorbase> connect address username password
```

The server address has to be written in the format `serverAddress:port`.

In case of success the user receives a confirm message (`"You are connected!"`), if not, the user receives an error message (`"Connection failed!"`).

6.1.2 `disconnect` command

To disconnect from the server, the user just has to insert the command `disconnect` and press enter.

6.1.3 `quit` command

To quit the client, the user just has to insert the command `quit` and press enter.

6.2 Inline help

It is possible to obtain an help for the allowed operations directly from the command line. The command `help` allows the user to obtain a generic help or a specific one.

6.2.1 Generic `help` command

Generic help command doesn't request more parameters and prints on screen the *Actorbase* commands list. Each command is followed by a brief description which explains his behaviour.

6.2.2 Specific `help` command

The specific help command has this structure:

```
actorbase> help commandName
```

It allows to request informations for a particulare command, and to obtain a description of it.

6.3 Server level operations commands

Once connected the user is at "server level". At this level the user can do operation the following operations:

- Show the databases list
- Select a database
- Create a database
- Remove a database

6.3.1 `listdb` command

This command allows the user to obtain, the list of databases of which the user has some access permission. The command does not request additional parameters.

```
actorbase> listdb
```

6.3.2 `selectdb` command

This command allows the user to select a database. Once selected a database the user can execute operations on its maps. The command structure is the sequent:

```
actorbase> selectdb databaseName
```

In case the user has the requested permissions, he receives a confirmation message: `"Database x selected"`. If not an invalid operation is reported.

6.3.3 `createdb` command

This commands allows the user to create a new database with the name specified, in case a database with that name isn't already in the server.

```
actorbase> createdb databaseName
```

If a database with the inserted name already exists the creation fails, the user receives an error message: `"A database with the requested name already exists"`.

6.3.4 `deletedb` command

This command allows the user to delete a database from the server:

```
actorbase> deletedb databaseName
```

If the user tries to remove a database than does not exist, or a database on which he does not have the modify permissions, the user receives an error message of invalid operation.

6.4 Database level operations commands

Once the user has selected a database with the `selectdb` command, he is at "database level". At that level a user can do these operations:

- Display the list of the maps which compose the database
- Map selection
- Map creation
- Map removal

6.4.1 `listmap` command

The command show every map within the selected database:

```
actorbase> listmap
```

6.4.2 `selectmap` command

This command allows to select a map using the following sintax:

```
actorbase> selectmap nomeMappa
```

If the selection is confirmed the message `"Map x selected"` is showed, otherwise an `"Invalid operation"` message is showed.

6.4.3 `createmap` command

This command allows to create a map with the specified name within the selected database:

```
actorbase> createmap nomeMappa
```

If the creation is confirmed the message `"Map x created"` is showed, otherwise an `"Invalid operation"` message is showed if the user doesn't have write permissions on the database or if a map with that name already exists.

6.4.4 `deletemap` command

This command allows to delete a map within the selected database:

```
actorbase> deletemap nomeMappa
```

If the delete is confirmed the message `"Map x deleted"` is showed, otherwise an `"Invalid operation"` message is showed if the user doesn't have write permissions on the database or if a map with that name doesn't exist.

6.5 Map level operations commands

Once selected a map via the command `selectmap`, the user is at the "map level". At this level the following operations are allowed:

- Show the list of keys in a map
- Find an item's value
- Insert an item
- Update an item's value
- Remove an item

6.5.1 `keys` command

This command shows the list of all keys within the selected map

```
actorbase> keys
```

6.5.2 `find` command

This command allows to get the value of an item in the map searching by his key.

```
actorbase> find key
```

If the map contains an item with the inserted key the client shows the value of the item otherwise an error is showed.

6.5.3 `remove` command

This command allows to remove an item within the map:

```
actorbase> remove key
```

If the remove is confirmed the message `"Item removed"` is showed, otherwise an `"No item with this key"` message is showed.

6.5.4 `insert` command

This command allows to insert an item (key-value) within the map:

```
actorbase> insert key value
```

If the insert is confirmed the message `"Item inserted"` is showed, otherwise an `"An item with this key already exists"` message is showed.

6.5.5 `update` command

This command allows to update an item's value within the map:

```
actorbase> update key value
```

If the update is confirmed the message `"Item updated"` is showed, otherwise an `"No item with this key"` message is showed.

6.6 Administrator operations commands

A user logged as admin can do the following operations:

- User list
- Add an user
- Remove an user
- User's permissions list
- Add/update a database permission to an user
- Remove a database permission to an user
- Set the max number of rows per Storekeeper
- Set the number of Ninja per Storekeeper
- Set the number of Warehouseman per Storekeeper
- Set the max number of Storekeeper per Storefinder
- Set the max number of Storefinder per Storemanager

6.6.1 `listuser` command

The command shows every user on the database.

```
actorbase> listmap
```

6.6.2 `adduser` command

This command allows to add an user:

```
actorbase> adduser username password
```

If the add is confirmed the message `"User x added"` is showed, otherwise an `"User already exists"` message is showed if an user with that username already exists.

6.6.3 `removeuser` command

This command allows to remove an user:

```
actorbase> removeuser username
```

If the remove is confirmed the message `"User x removed"` is showed, otherwise an `"User doesn't exist"` message is showed if an user with that username doesn't exist.

6.6.4 `listpermission` command

The command shows every user's permissions on the database.

```
actorbase> listpermission
```

6.6.5 `addpermission` command

This command allows to add a database permission to an user:

```
actorbase> addpermission username database permissionType
```

If the add is confirmed the message `"Permission x added to user"` is showed, otherwise an `"User doesn't exist"` message is showed if an user with that username doesn't exist or `"Database doesn't exist"` a database with that name doesn't exist.

6.6.6 `removepermission` command

This command allows to remove a database permission to an user:

```
actorbase> removepermission username database
```

If the remove is confirmed the message `"Permission x remove to user"` is showed, otherwise an `"A permission for that database doesn't exists"` message is showed if a permission for that database of that user doesn't exits.

6.6.7 `setmaxrows` command

This command allows to set the max number of rows for each Storekeeper:

```
actorbase> setmaxrows number
```

6.6.8 `setninja` command

This command allows to set the number of Ninjas for each Storekeeper:

```
actorbase> setninja number
```

6.6.9 `setwarehouseman` command

This command allows to set the number of Warehousemans for each Storekeeper:

```
actorbase> setwarehouseman number
```

6.6.10 `setmaxstorekeeper` command

This command allows to set the max number of Storekeepers for each Storefinder:

```
actorbase> setmaxstorekeeper number
```

6.6.11 `setmaxstorefinder` command

This command allows to set the max number of Storefinders for each Storemanager:

```
actorbase> setmaxstorefinder number
```

Elenco delle figure

Elenco delle tabelle