

Distributed Emergent Software: Assembling, Perceiving and Learning Systems at Scale – Experiments Replication

June 13th, 2019

1 Introduction

Welcome and thank you for using our distributed emergent software framework prototype. This prototype was described in the paper “*Distributed Emergent Software: Assembling, Perceiving and Learning Systems at Scale*” submitted to **SASO 2019**. This manual contains instructions to execute the framework with the datacentre software and the pervasive system described in the paper.

Please, download and install Dana programming language¹. Feel free to drop us a line in the forum if you have any questions, comments or suggestions. Also, consult Dana’s Programming Guide², for instructions to properly install Dana’s toolchain and to better understand the syntax of the code distributed with this manual. This project should be executed in Ubuntu to better replicate the results presented in the paper.

2 Datacentre Software

2.1 Project Structure

The folder structure of this project follows that of all Dana projects. The *resources* folder contains all of the interfaces for the components. The implementation of the interfaces are provided in the folders outside of *resources*. For example, the interface that defines Assembly functionality is in *resources/composition*, in **Assembly.dn** interface. The components are located in the folders with the same name as the interface’s folders outside *resources*, in this example *composition/Assembly.dn*.

The framework and web platform code are located in different folders. The web platform main components are located in *http*, *cache* and *compression* inside of *repository* folder. The main framework components (assembly, perception and learner) are presented in *composition*, *monitoring* and *learning* inside of *metacom* folder.

¹<http://www.projectdana.com>

²<http://www.projectdana.com/dana/guide/doku.php>

2.2 Required Software and Configurations

As previously described you will need to have the Dana programming language in its 225 version installed³. Other than the Dana programming language, you will also need to install the MySQL database⁴.

After installing all necessary software (Dana and MySQL), you will need to setup the database scheme. In order to do that, you will need to compile and execute the script located in `repository/danapedia` named *DanaPediaDBCcreator.dn*. Before compiling the file, in line 7, add your database info (ip address, user and password):

```
MySQL mysql = new MySQL("IP", 0, "user", "pass", "")
```

To compile the script that sets the database scheme type:

```
dnc DanaPediaDBCcreator.dn
```

After the script is compiled, execute it with the following command:

```
dana DanaPediaDBCcreator
```

Also, you will need to provide information about the database username and password to the `DanaPedia.dn` component. This component is responsible to process incoming requests to the Danapedia application, an application that runs on our web platform. The `DanaPedia.dn` component is located in `repository/danapedia/DanaPedia.dn`, and to configure it, simply change the line 21 (illustrated below) and provide the database username and password:

```
db = new DBConnector("root", "root", "danapedia")
```

Now that the required software are installed and properly configured, move forward to the next sections of this manual to execute the datacentre software and explore its self-distributing feature.

2.3 Execution

The first step to execute the code is to compile the project. In order to compile the entire datacentre project, it is necessary to execute the command “`dnc . -sp ../repository`” in *metacom* folder, and execute “`dnc . -sp ../metacom`” in *repository* folder. Please, make sure Dana’s toolchain is properly installed and working.

³<http://www.projectdana.com/dana/guide/doku.php>

⁴<http://www.mysql.com/downloads/>

2.3.1 Interactive Distributor

The Interactive Distributor is the program that allows you to control the different system composition through a command line interface. You will be able to list all available compositions, and change the system from one composition to another by inserting simple commands in the prompt. Some of the available compositions will relocate or replicate a set of components to a set of different machines in your infrastructure. The Interactive Distributor is located in *metacom* folder. After compiling the entire project as previously described (by compiling the files in both *metacom* and *repository* folders), you will need to configure the `InteractiveDistributor.dn`. In the `InteractiveDistributor.dn` you will find the following snippet of code:

```
String ips[] = new String[](  
new String("local-ip:2021"),  
new String("second-node-ip"),  
new String("third-node-ip"),  
new String("fourth-node-ip"))
```

Change that snippet of code by adding the addresses of the nodes that are part of your system. For instance:

```
String ips[] = new String[](  
new String("scc-mc1.lancs.ac.uk:2021"),  
new String("scc-mc2.lancs.ac.uk"),  
new String("scc-mc3.lancs.ac.uk"),  
new String("scc-mc4.lancs.ac.uk"))
```

As the next step, open the `Distributor.dn` component located in *metacom/distributor* folder, and change the IP address in line 16 to point to the node where you will run the `Manager.o` component. The `Manager.o` component is responsible to coordinate adaptation of the distributed system, and interact with the `ESLauncher.o` component which are responsible to execute relocated and replicated subgraphs of components (this is part of the self-distributing feature of this project). Make sure to run the `Manager.o` component on the same node you are planning on running the `InteractiveDistributor.o` component, the node that will be the entry point of the system.

After these changes are done, you will need to compile the Interactive Distributor, and the Distributor. Simply execute the `dnc . -sp ../repository` in the *metacom* folder. Also you will need to start the `Manager.dn` component located in *metacom* folder using the following command:

```
dana -sp ../repository Manager
```

It is advisable to execute the `Manager.dn` component on the entry point node, the node where you are planning on running the Interactive Distributor. Now you should

be able to start the Interactive Distributor, in *metacom* folder execute:

```
dana -sp ../repository InteractiveDistributor ../repository/TCPNetwork.o
```

And then executing the `ESLauncher.dn` in the nodes you listed in `InteractiveDistributor.dn`, in the *metacom* folder execute:

```
dana -sp ../repository ESLauncher.o
```

Finally, you will be able to access the running web server by going to a web browser and typing "http://node-address:2012/danapedia", where *node-address* is the address of the node running the Interactive Distributor. You should get our *DanaPedia* web application home page in your browser. You should also be given a command prompt to interact with the Interactive Distributor to list the available compositions, and change the system from one composition to another. Type **help** to list the commands available.

2.3.2 Learning System

The *learning module* uses the *perception module* to collect information from the executing components. It then establishes correlation between the recognised pattern and the best performing architecture (a fully description of this component is provided in the paper – including the real-time learning technique and related details).

In order to execute the learning module you have to first change the `SASOLearner.dn` component in the *metacom* folder, providing information to line 99:

```
String ips[] = new String[] (
new String("scc-mc1.lancs.ac.uk:2021"),
new String("scc-mc2.lancs.ac.uk"),
new String("scc-mc3.lancs.ac.uk"),
new String("scc-mc4.lancs.ac.uk"))
```

Add the IP address of the local node, followed by the IP addresses of the rest of the nodes that are part of the system, just like you did to configure the Interactive Distributor. Then, compile the entire project (as previously described). In the *metacom* folder, start the `Manager.o` and `ESLauncher.o` with the commands:

```
dana -sp ../repository Manager.o
```

```
dana -sp ../repository ESLauncher.o
```

You should start an instance of the `ESLauncher.dn` component in every single node you provided the address in the `SASOLearner.dn` component, and execute the `Manager` on the IP address stated in line 16 in the `Distributor.dn` component located in *metacom/distributor*. For more information refer to Interactive Distributor.

Now it is important to execute a client request pattern from a client machine to watch the learning system in action; for example by executing: ‘`dana Client`’ from *repository/ws_clients* folder. Before executing the client program, however, please remember to change the IP address of the entry point node in line 16.

3 Pervasive System

For our pervasive system example, we take a more theoretical approach to studying learning across the design space. Here we use a simulated version of our real system which has hard-wired values for compute and network, and we gain a trace from this system for learning. We then use our learning algorithm to understand convergence time under different conditions.

First, open a command terminal in the *hetper* directory, and compile everything with the command:

```
dnc .
```

You can then gain a system trace using two command terminals in this directory: one runs our standard emergent systems framework and the other explores the permutations of the system being managed by that framework:

```
dana pal.rest Server.o  
dana StatWriter.o my_trace.txt
```

We can then run our UCB1-tuned algorithm on the same system, with the command:

```
dana Learning.o my_data.txt
```

4 Final Considerations

Thank you for your interest in our research. We hope you find this manual helpful to get you started with our current version of the distributed emergent software framework. All questions, comments and feedback is greatly appreciated. Contact us on Project Dana’s forum (<http://www.projectdana.com/fora>) or over email (r.rodriguesfilho@lancaster.ac.uk).