OdysseusNet

(this page is currently work-in-progress)

OdysseusNet (or Odysseus.net) is a extension for Odysseus which enables distributed data stream processing using a network of interconnected machines (e.g., LANs). It is a follow-up project of OdysseusP2P.

Installation and Starting

OdysseusNet is a feature which is installed like the other features (see How to install new features). However due to technical limitations, OdysseusNet is (currently) not applicable to the client-version of Odysseus. If you use the server-version, install the OdysseusNet Server Feature. For monolithic Odysseus, install the monolithic version of the OdysseusNet Feature. After restarting Odysseus, OdysseusNet is integrated and ready.

If you develop with Odysseus and checked out its sourcecode, you can start OdysseusNet by adding the plugin `de.uniol.inf.is.odysseus.net.monolithic.feature` (if you're developing with the monolithic version of Odysseus) or the plugin `de.uniol.inf.is.odysseus.net.server.feature` (server-version of Odysseus) to your product- or launch-definitions. These features contain all underlying OdysseusNet-Features currently available. A more detailed description which feature encapsulates which function can be found in Developing with OdysseusNet.

Docker

OdysseusNet is also available as docker image: odysseusol/odysseusnet.

Remark: Currently, we are developing a **master/worker based version of OdysseusNet**. In this case, only the master node needs to be an OdysseusNet image. The worker images could be standard Odysseus images. To allow easisier result handling, we provide an image for the worker nodes with an integrated kafka handling: odysseusol/odysseus_kafka

See OdysseusNet Docker Compose Example

Remark: In cases where the workers should communicate with each other, it is important, that the master and the worker are in the same network, i.e. in case of using docker, you cannot use a master from outside the docker network. To communicate with the master node, you should use WebStudio. Another solution would be to use e.g. kafka (that is visible to all the nodes) to communicate between the nodes.

See e.g. https://docs.docker.com/network/ or https://www.tutorialworks.com/container-networking/

Network structure of OdysseusNet: OdysseusNodes and NodeGroups

After starting Odysseus, OdysseusNet is – by default – deactivated. In this case, Odysseus behaves like normal. When OdysseusNet is activated (explained in the next section), the Odysseus-Instance becomes an *OdysseusNode.* A OdysseusNode is a machine running Odysseus with OdysseusNet activated and participates in the distributed data stream processing. Each node has a human-readable non-unique name (set in the OdysseusNet Configuration) and a generated unique *NodeID* (needed for developers).

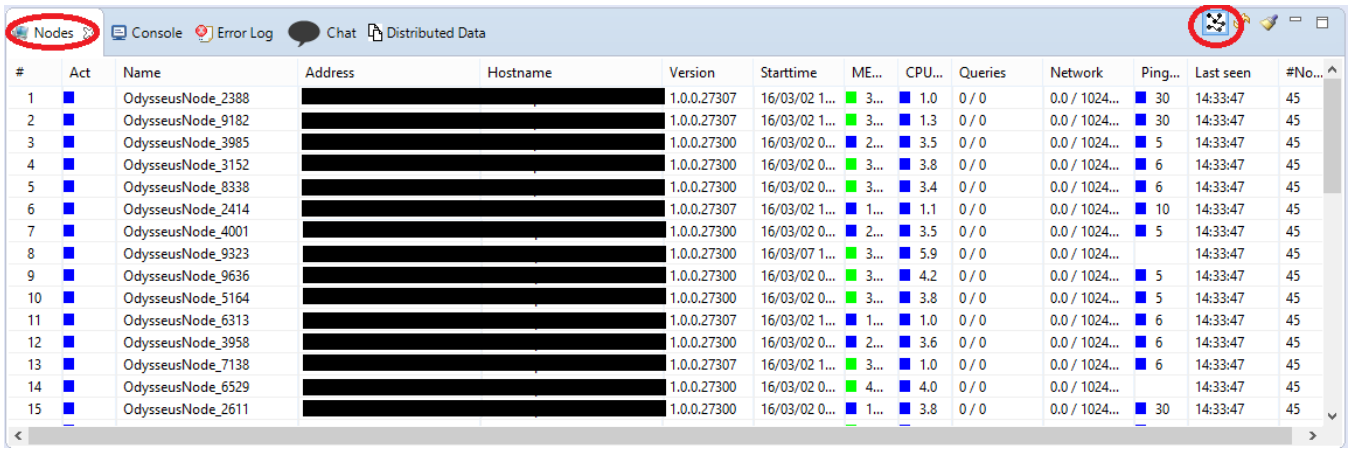
By default, two nodes are connecting to each-other if they are physically reachable (e.g., LAN), forming a network of interconnected Odysseus-Instances. To divide the network in logical subnetworks, each OdysseusNode can be assigned to a *NodeGroup*. Only nodes, which are in the same group, are connected and can communicate with each-other. By default, each OdysseusNode is in the same group (called `OdysseusGroup`). However, this can be changed in the OdysseusNet Configuration of OdysseusNet.

With this default behaviour, a decentralized and unstructured network of nodes is being built. See Discovery of OdysseusNodes and Connection to OdysseusNodes if you want more details and to modify this.

In case of the master/worker approach, worker must be assigned to the master and could not be found automatically.

Activating/Deactivating OdysseusNet

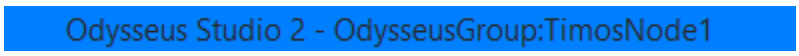
Only when OdysseusNet is activated, the functionality for distributed data stream processing is available. There are multiple possibilities to activate it. If the graphical user interface OdysseusStudio is available, the user can switch to the OdysseusNet Perspective. There in the NodeView, the user can click on the button in the top-right corner to activate OdysseusNet (marked with red circles in the picture below).



#	Act	Name	Address	Hostname	Version	Starttime	ME...	CPU...	Queries	Network	Ping...	Last seen	#No...
1		OdysseusNode_2388	[redacted]		1.0.0.27307	16/03/02 1...	3...	1.0	0 / 0	0.0 / 1024...	30	14:33:47	45
2		OdysseusNode_9182	[redacted]		1.0.0.27307	16/03/02 1...	3...	1.3	0 / 0	0.0 / 1024...	30	14:33:47	45
3		OdysseusNode_3985	[redacted]		1.0.0.27300	16/03/02 0...	2...	3.5	0 / 0	0.0 / 1024...	5	14:33:47	45
4		OdysseusNode_3152	[redacted]		1.0.0.27300	16/03/02 0...	3...	3.8	0 / 0	0.0 / 1024...	6	14:33:47	45
5		OdysseusNode_8338	[redacted]		1.0.0.27300	16/03/02 0...	3...	3.4	0 / 0	0.0 / 1024...	6	14:33:47	45
6		OdysseusNode_2414	[redacted]		1.0.0.27307	16/03/02 1...	1...	1.1	0 / 0	0.0 / 1024...	10	14:33:47	45
7		OdysseusNode_4001	[redacted]		1.0.0.27300	16/03/02 0...	2...	3.5	0 / 0	0.0 / 1024...	5	14:33:47	45
8		OdysseusNode_9323	[redacted]		1.0.0.27300	16/03/07 1...	3...	5.9	0 / 0	0.0 / 1024...		14:33:47	45
9		OdysseusNode_9636	[redacted]		1.0.0.27300	16/03/02 0...	3...	4.2	0 / 0	0.0 / 1024...	5	14:33:47	45
10		OdysseusNode_5164	[redacted]		1.0.0.27300	16/03/02 0...	3...	3.8	0 / 0	0.0 / 1024...	5	14:33:47	45
11		OdysseusNode_6313	[redacted]		1.0.0.27307	16/03/02 1...	1...	1.0	0 / 0	0.0 / 1024...	6	14:33:47	45
12		OdysseusNode_3958	[redacted]		1.0.0.27300	16/03/02 0...	2...	3.6	0 / 0	0.0 / 1024...	6	14:33:47	45
13		OdysseusNode_7138	[redacted]		1.0.0.27307	16/03/02 1...	3...	1.0	0 / 0	0.0 / 1024...	6	14:33:47	45
14		OdysseusNode_6529	[redacted]		1.0.0.27300	16/03/02 0...	4...	4.0	0 / 0	0.0 / 1024...		14:33:47	45
15		OdysseusNode_2611	[redacted]		1.0.0.27300	16/03/02 0...	1...	3.8	0 / 0	0.0 / 1024...	30	14:33:47	45

The user can execute the Console Command `startOdysseusNet` to start OdysseusNet in the console (useful in the server-version of Odysseus). Alternatively, it is possible to start OdysseusNet automatically when Odysseus is started (see OdysseusNet Configuration of OdysseusNet).

After OdysseusNet is activated successfully, the title bar of OdysseusStudio shows the name and the NodeGroup of the own OdysseusNode aside the OdysseusStudio 2 title (separated with ":").



In the console, the message `OdysseusNet started` is printed.

To finally stop OdysseusNet, the user has to click the button in the toolbar of the NodeView again, execute the console command `stopOdysseusNet` or has to stop Odysseus entirely.

Features of OdysseusNet

OdysseusNet is a sum of multiple smaller OdysseusNet-features. Each of them encapsulates specific functions, possibilities and interfaces, all useful for "typical" Odysseus-Users and Odysseus-Developers. A (not complete) list of features is as follows:

For users of Odysseus, OdysseusNet provides the following features:

- Integration of OdysseusNet in OdysseusStudio. There, OdysseusNet has its own OdysseusNet Perspective.
- Configuration of each OdysseusNode due to a separate configuration file (see OdysseusNet Configuration)
- Controlling OdysseusNodes with the OSGi-Console (see Console Commands)
- View the connection quality with PingMaps
- Distributing data sources definitions into the network
- Distributing queries into the network of OdysseusNodes

For developers, OdysseusNet provides additional in-depth features (e.g., interfaces) to make it possible to alter and/or extend OdysseusNet aside the extensibility of Odysseus (see Developing with OdysseusNet):

- Interfaces for accessing remote OdysseusNodes (see Management of OdysseusNodes)
- Interfaces for easy communication with remote OdysseusNodes (see Communication with OdysseusNodes)
- Receiving log messages of remote nodes (see Logging)
- Deeper control of configuration settings (without interfering with the configuration of Odysseus) (see Configuration for developers)
- Network structure control. This contains interfaces for discovery of OdysseusNodes (see Discovery of OdysseusNodes) and selection-strategies for connections that need to be established (see Connection to OdysseusNodes).
- Easy methods to distribute static data across the network (see Distribution of (static) data)
- Interfaces to add strategies for query distribution (see Query distribution strategies)
- Interfaces to access information of current resource usages (processor, memory, etc.) of remote OdysseusNodes (see Resource management)