

JuxMem: Juxtaposed Memory

Table of contents

1 Overview.....	2
2 Layers of a JuxMem peer.....	2
3 Bindings.....	2

1. Overview

Entities of JuxMem

The above figure shows the hierarchy of the entities defined in the architecture of JuxMem. This architecture is made up of a network of peer groups (cluster groups A, B and C), which generally correspond to clusters at the physical level. All the groups are inside a wider group which includes all the peers which run the service (the juxmem group). Each cluster group consists of a set of nodes which provide memory for data storage. We will call these nodes **providers**. In each cluster group, a node is in charge of managing the memory made available by the providers of the group. This node is called **cluster manager**. Finally, a node which simply uses the service to allocate and/or access data blocks is called **client**. It should be stressed that a node may at the same time act as a cluster manager, a client, and a provider. However, each node only plays a single role in the example illustrated on the figure for the sake of clarity.

When allocating memory, the client has to specify on how many clusters the data should be replicated, and on how many nodes in each cluster. This results into the instantiation of a set of data replicas. The allocation operation returns a global data ID. This ID can be used by other nodes in order to identify existing data. To obtain read and/or write access to a data block, the clients only need to use this ID. It is JuxMem's responsibility to localize the data, and then perform the necessary data transfers.

2. Layers of a JuxMem peer

Layers of a JuxMem peer

The above figure shows all the layers of a JuxMem peer (client, provider or cluster manager, that does not matter). The goal of the JuxMem core layer is to hide JXTA to upper layers, therefore making them independent of the P2P library used, and also to adapt JXTA to grid architectures. This layer is developed by Mathieu Jan. The goal of the self-organizing group and glue layer is to transparently allow consistency protocol to be fault-tolerant. This layer is developed by Sébastien Monnet. The consistency protocol layer implements consistency models available in JuxMem. Currently JuxMem implements an entry consistency model through a hierarchical consistency protocol developed by Jean-François Deverge.

3. Bindings

There is 2 bindings of JuxMem: a Java binding ([JuxMem-J2SE](#)) and a C binding ([JuxMem-C](#)). Both implements all the entities of JuxMem, but if you want to mix peers from both bindings, you can't use Java clients with C providers. Failure detectors are only

implemented in the Java Binding, which features several consistency protocols, whereas only the entry consistency protocol is implemented in the C binding. Current development in JuxMem is mostly done on the C binding, for performance reasons.