Project Number 318763

D7.2 – Reference Implementation

Version 1.0
7 December 2015
Final

Public Distribution

aicas

Project Partners: **aicas**, **HMI**, **petaFuel**, **SOFTEAM**, **Scuola Superiore Sant'Anna**, **The Open Group**, **University of Stuttgart**, **University of York**, **Brno University of Technology**

Project Partner Contact Information

aicas Fridtjof Siebert Haid-und-Neue Strasse 18 76131 Karlsruhe Germany Tel: +49 721 66396823 E-mail: siebert@aicas.com	**HMI** Markus Schneider Im Breitspiel 11 C 69126 Heidelberg Germany Tel: +49 6221 7260 0 E-mail: schneider@hmi-tec.com
petaFuel Ludwig Adam Muenchnerstrasse 4 85354 Freising Germany Tel: +49 8161 40 60 202 E-mail: ludwig.adam@petafuel.de	**SOFTEAM** Andrey Sadovykh Avenue Victor Hugo 21 75016 Paris France Tel: +33 1 3012 1857 E-mail: andrey.sadovykh@softeam.fr
Scuola Superiore Sant'Anna Mauro Marinoni via Moruzzi 1 56124 Pisa Italy Tel: +39 050 882039 E-mail: m.marinoni@sssup.it	**The Open Group** Scott Hansen Avenue du Parc de Woluwe 56 1160 Brussels Belgium Tel: +32 2 675 1136 E-mail: s.hansen@opengroup.org
University of Stuttgart Bastian Koller Nobelstrasse 19 70569 Stuttgart Germany Tel: +49 711 68565891 E-mail: koller@hlrs.de	**University of York** Neil Audsley Deramore Lane York YO10 5GH United Kingdom Tel: +44 1904 325571 E-mail: neil.audsley@cs.york.ac.uk
Brno University of Technology Pavel Smrz Bozetechova 2 61266 Brno Czech Republic Tel: +420 54114 1282 E-mail: smrz@fit.vutbr.cz	

Contents

1 Summary **1**

2 Introduction **2**

3 Maven Repository **3**

4 Platform **4**

4.1 Offline MPI extensions 4

5 Modelling and Development Tools **5**

5.1 Modelling Environment 5

5.2 PostgreSQL Modeller 5

5.3 MongoDB Modeller 5

5.4 JUNIPER Modelio Extension 5

6 Monitoring Library **6**

7 Scheduling Advisor **7**

8 Real-Time Scheduling Analysis **8**

9 Realtime Java Platform **9**

10 Realtime Operating System **10**

11 FPGA Acceleration/caicos **11**

References **13**

Document Control

Version	Status	Date
0.1	Document outline	20 November 2015
0.2	Draft	27 November 2015
0.3	Final Draft	3 December 2015
1.0	Final	7 December 2015

1 Summary

This document constitues deliverable 7.2 – “Reference Implementation” of work package 7 of the JUNIPER project. It provides the reference Implementation of the Realtime Operating System, the Java Platform, API implementation as well as development and management tools. It does so by providing a listing of the delivered software components with short descriptions, references to the respective release deliverables and instructions how the software can be obtained, either in source or binary form. For open source components it provides information on the source repository location.

2 Introduction

The JUNIPER project consortium created a large set of software components for achieving the project's goals. To make the JUNIPER technology attractive for a wide range of use-cases the focus set on a high degree of modularity. Thus the resulting software artifacts are provided in independent modules. This document provides an overview of the individual components and how they can be obtained.

There is pre-existing and new open source software being used in the JUNIPER project together with some close source components:

- For pre-existing open source software code was fed back into the source project, if possible (e.g. OpenMPI, Linux kernel).
- For newly created open source software GitHub repositories were created under a JUNIPER umbrella project: `https://github.com/juniper-project`
- Closed source software (i.e. JamaicaVM) is considered optional and available in an evaluation version to the user.

The following sections of this document will list the individual components, provide short descriptions, license information and instructions on how to obtain the sources, if applicable.

For users who want to use the framework without rebuilding the sources can take advantage of an Apache Maven[1] repository to easily retrieve tested builds of the software components.

[1] `https://maven.apache.org/`

3 Maven Repository

For retrieving a binary release of the open source Java-based JUNIPER components an Apache Maven repository was made available to the public.

The Maven repository is available at `https://github.com/juniper-project/mvn-repo`.

To use the repository in a software project, dependencies on indidivual components can be added into Maven pom.xml files in the following format:

```
<dependency>
        <groupId>eu.juniper</groupId>
        <artifactId>juniperplatform</artifactId>
        <version>1.0</version>
</dependency>
```

The repository contains the following artifacts:

- juniperplatform
- offlinempi
- openmpi
- monitoring
- sched-advisor-deployment-executor
- sched-advisor-deployment-model
- sched-advisor-deployment-monitor
- sched-advisor-monitoring-agent
- sched-advisor-tool

In order to make the repository and the artifacts it contains accessible to a Maven installation, in addition the following entry needs to be added to the pom.xml configuration file:

```
<repositories>
    <repository>
        <id>juniper-mvn-repo</id>
        <url>https://github.com/juniper-project/mvn-repo/raw/master/repository</url>
        <snapshots>
            <enabled>true</enabled>
            <updatePolicy>always</updatePolicy>
        </snapshots>
    </repository>
</repositories>
```

4 Platform

The JUNIPER platform, the software framework for developing and executing JUNIPER applications, based on Java and MPI is available in the JUNIPER GitHub project: `https://github.com/juniper-project/platform`.

The platform consists of

1. a core runtime environment for executing JUNIPER,
2. an example application replicating Hadoops MapReduce example on the platform,
3. an identical example application that does require MPI to be installed.

The platform is open source software licensed under the BSD license, for details please refer to the license file in the GitHub repository.

4.1 Offline MPI extensions

As the JUNIPER platform aims to be modular, the usage of MPI is considered optional, hence there is a partial re-implementation of the interface for running code generated by the modelling tools offline.

The Offline MPI extensions can be opbtained from the JUNIPER GitHub project: `https://github.com/juniper-project/offline-mpi-extensions`.

The Offline MPI extensions are open source software licensed under the Apache License Version 2.0, for details please refer to the license file in the GitHub repository.

5 Modelling and Development Tools

The JUNIPER modelling environment consists of four submodules:

1. the core modelling environment,
2. a Modelio module to extend the modelling environment with PostgreSQL database support,
3. a Modelio module to extend the modelling environment with MondoDB support and
4. a Modelio extension to support the JUNIPER programming model.

The modelling tools base on SOFTEAM's Modelio Development Environment, which is available from `http://forge.modelio.org/projects/juniper/`.

The modelling and development tools are licensed under the Apache License, for details please refer to the license files in the respective GitHub repositories.

5.1 Modelling Environment

The modelling environment constitutes the API used by automatically generated code generated by the Modelio Development Environment. It allows the applications to interact with the JUNIPER platform. It can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/modelling-environment`

5.2 PostgreSQL Modeller

The PostgreSQL modeller is a Modelio module that extends the JUNIPER modelling environment with the support for PostgreSQL databases in JUNIPER applications. It can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/modelling-postgresql`

5.3 MongoDB Modeller

The MongoDB Modeller is a Modelio module that extends the JUNIPER modelling environment with the support for MongoDB databases in JUNIPER applications. It can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/modelling-mongodb`

5.4 JUNIPER Modelio Extension

The JUNIPER Modelio Extension adds support for JUNIPER's programming model for Big Data realtime applications to the Modelio development environment.

It can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/modelling-juniper-api`

6 Monitoring Library

The monitoring library is designed to support Scheduling Advisor with monitoring of existing as well as user-defined metrics. It writes and reads metrics values from an ElasticSearch database.

The sources of the monitoring library can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/monitoring-lib`

The Monitoring Library is licensed under the BSD license, for details please refer to the license file in the GitHub repository.

7 Scheduling Advisor

The Scheduling Advisor, as described in deliverable 3.9 “Final Real-time Scheduling Advisor” [4] analyses monitoring data and provides advices for the performance tuning of JUNIPER applications.

It can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/sched-advisor`

The scheduling advisor is licensed under the BSD license, for details please refer to the license file in the repository.

8 Real-Time Scheduling Analysis

The Real-time Scheduling Analyser, as described in Deliverable 5.6 “Final Integrated MDE Environment” [7] is a tool implementing Response-Time Analysis of applications modelled by Directed Acyclic Graphs. Nodes in a graph represent computations (e.g., CPU, I/O), while edges represent precedence constraints among these computations.

The final version of the tool is available from the JUNIPER GitHub project: `https://github.com/juniper-project/rt-sched-analysis`. It can be used both as a stand-alone command line tool or integrated within the JUNIPER MDE Enrivonment.

The Real-Time Scheduling Analyser is licensed under the GNU Public License in version 2; for details please refer to the license file in the GitHub repository.

9 Realtime Java Platform

The JUNIPER Realtime Java Platform is based on aicas' commercial product JamaicaVM and closed source software. The Java platform was finalized with deliverable 4.4 – "Java Platform and Tools Implementation" [5].

The release of the Java platform and tools can be obtained from aicas[2]. For installation instructions please refer to Deliverable 4.5 – "Java Platform and Tools Documentation" [6] or the JamaicaVM User Manual [1].

[2] `https://www.aicas.de/customers/University_Reviewer`, user: University_Reviewer, password: rUAt8bNWZ

10 Realtime Operating System

The Realtime Operating System, as described in Deliverable 3.6 “Final Operating System with Real-Time Support” [3]. The real-time features developed within the JUNIPER project include:

- PREEMPT_RT, a patch replacing most kernel spin-locks with mutexes that support Priority Inheritance and moving interrupts and software interrupts to kernel threads; moreover, it provides high resolution kernel timers.
- SCHED_DEADLINE, a new scheduling class providing Earlieast Deadline First scheduling together with a bandwidth researvation mechanism based on the Constant Bandwidth Server algorithm; this feature has been merged in mainline Linux since version 3.14.
- BFQ, providing fair allocation of disk I/O bandwidth; the amount of bandwidth actually reserved depends on the number of requests and their relative priorities; BFQ is highly tunable for both bandwidth and latency and it is also compatible with cgroups.

The changes to the Linux kernel are available as patches that apply to Linux kernel 4.0.4, they can obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/rt-linux-kernel`.

The Real-Time Operating System is licensed under the GNU Public License in version 2; for details please refer to the license file in the GitHub repository.

11 FPGA Acceleration/caicos

caicos is a tool designed to build hardware accelerators, suitable for use in Xilinx Vivado HLS, from C code that has been produced by Jamaica Builder from aicas. It was released as part of deliverables 2.5 – "Static Acceleration Implementation and Assessment" [2] and 2.7 – "Dynamic Acceleration Implementation and Assessment".

The sources can be obtained from the JUNIPER GitHub project: `https://github.com/juniper-project/fpgas-caicos`. Note that use of the software requires two closed source tools, Xilinx Vivado HLS and aicas JamaicaVM.

caicos is licensed under the GNU Public License in version 2, for details please refer to the license file in the GitHub repository.

References

[1] aicas GmbH. JamaicaVM User Manual. https://www.aicas.com/cms/sites/default/files/jamaicavm_6.3_manual.pdf, also available as print, 2014.

[2] The JUNIPER Project. D2.5 – Static Acceleration Implementation and Assessment, 2015.

[3] The JUNIPER Project. D3.6 – Final Operating System with Real-Time Support, 2015.

[4] The JUNIPER Project. D3.9 – Final Real-time Scheduling Advisor, 2015.

[5] The JUNIPER Project. D4.4 – Java Platform and Tools Implementation, 2015.

[6] The JUNIPER Project. D4.5 – Java Platform and Tools Documentation, 2015.

[7] The JUNIPER Project. D5.6 – Final Integrated MDE Environment, 2015.