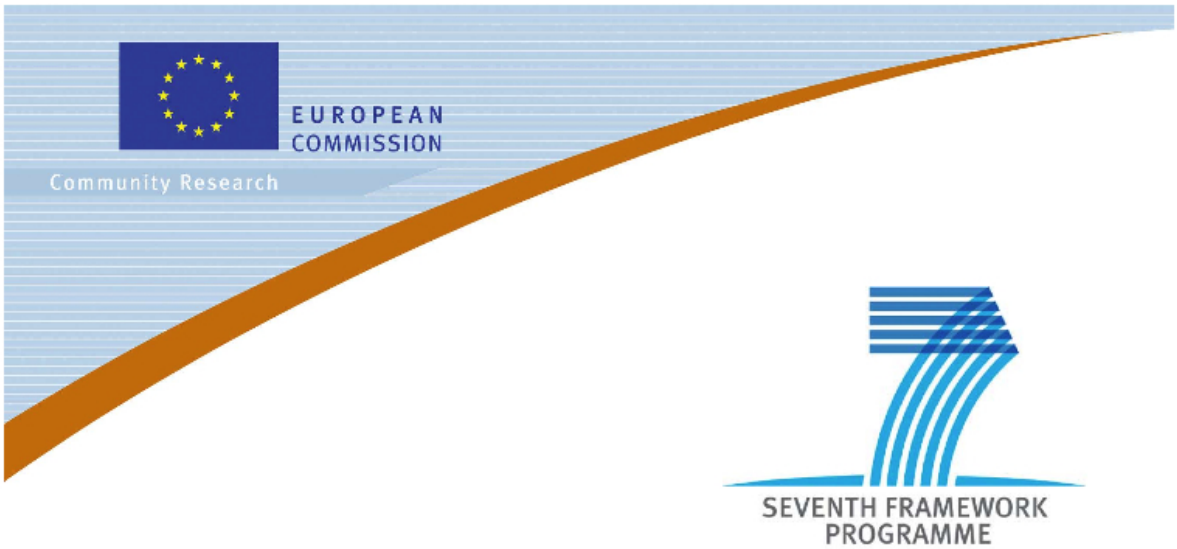
Application Testing Support Tools front page



Large-scale Integrated Project (IP)

D9.4: Application Testing Support Tools

Project acronym: FI-WARE

Project full title: Future Internet Core Platform

Contract No.: 285248

Strategic Objective: FI.ICT-2011.1.7 Technology foundation: Future Internet Core Platform

Project Document Number: ICT-2011-FI-285248-WP9-D9.4a

Project Document Date: 30 May 2012

Deliverable Type and Security: Public

Author: Matteo Melideo (ENG)

Contributors: Davide Dalle Carbonare (ENG), Roozbeh Farahbod (SAP), Tim Jonischkat (UDE), Osama Sammodi (UDE), Marcel Zalmanovici (IBM-IL)

Abstract: This deliverable provides a subset of reference tools to test and analyze specific metrics related to the run-time behavior of Future Internet application.

Keyword list: FI-WARE, PPP, DevComE, PROSA, TraceAnalyzer, Software Performance Cockpit, test, performance, Quality of Service, QoS, System Activity Report, TCPDump, Eclipse

Contents

Articles

References

Application Testing Support Tools

Software Components

This deliverable is mainly composed by software prototypes that are listed here and briefly described. More information on the tool, together with how to obtain it, are available trough the reported links.

PROSA

: PROSA is an online testing framework targeted to ensure the constant availability of QoS monitoring data for a given service (used by a FI App).

Trace Analyzer

: Trace Analyzer is an Eclipse plugin for graphical and numerical analysis of performance traces.

SoPeCo

: Software Performance Cockpit is a framework for systematic performance evaluations of software systems, based on systematic measurements, statistical methods, and machine learning.

PROSA

Introduction

PROSA is an online testing framework targeted to ensure the constant availability of QoS monitoring data for a given service (used by a FI App). The Runtime solution monitors the (real) usage of a given service and additionally performs additional service invocations (online testing) where necessary. PROSA thereby ensures that a minimum set of QoS data points is available for the given service for every time interval.

PROSA will be running as a background service inside the FI-WARE platform (or testbed), monitoring the QoS data of outbound service calls of service compositions in a service composition engine. By specifying monitoring parameters for specific services, PROSA is provided with the relevant data to ensure that the requested monitoring interval is maintained. PROSA therefore completes the set of monitoring data gathered from actual service utilization with data generated from online test executions (performed by the PROSA framework itself).

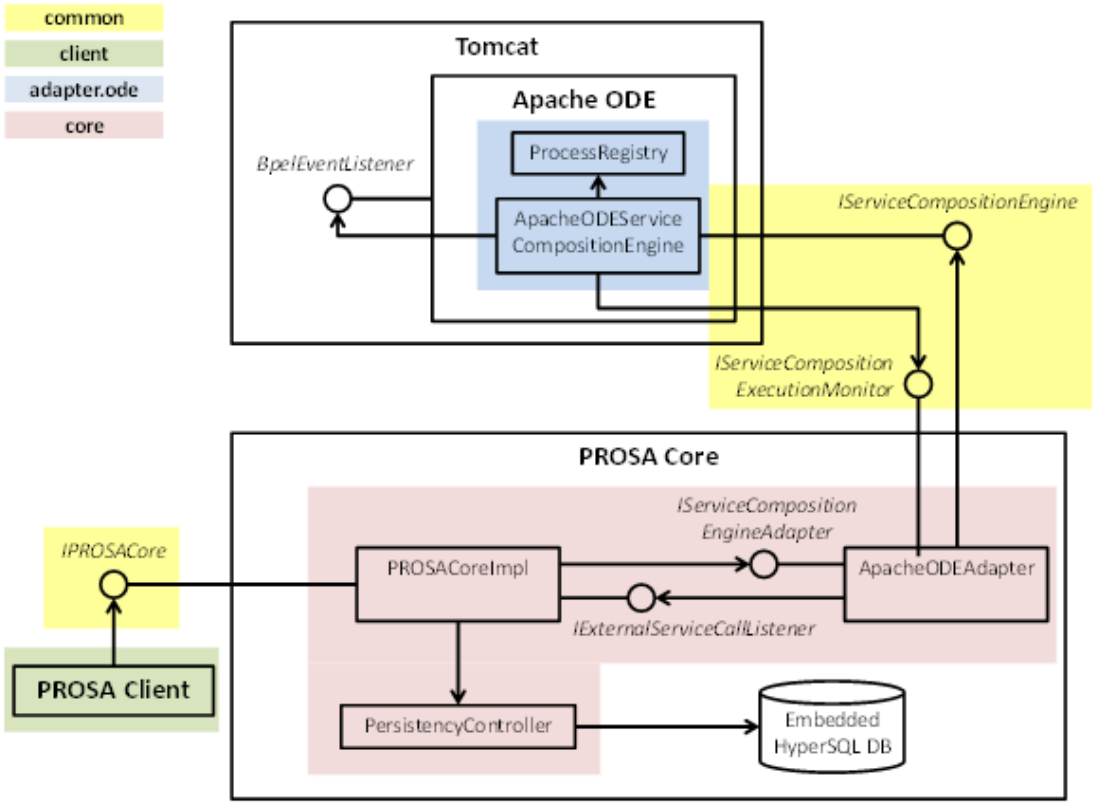
The Management Console (Interface) of PROSA will be provided as a client component inside the FI-WARE IDE providing the user the functionality to (1) manage the services / service composition engines monitored by the PROSA platform, (2) specify monitoring parameters for the monitored services and (3) view/analyze the historical QoS data.

Information

Name: PROSA
Version: 1.0.0
License: Open-source license. Detailed license information will be provided soon
Documentation: PROSA User Manual [1]
Download: Source code and binary [2]
Scope: Performance Testing and Monitoring

Architecture

The following figure describes the technical architecture (technical components and their interactions) of the current implementation of PROSA.



Key components are the Apache ODE service composition engine (running on Tomcat), the PROSA Core, and PROSA Client.

Service compositions implemented in BPEL are deployed and run in the Apache ODE service composition engine.

The PROSA Client is used to specify parameters for connecting PROSA Core with Apache ODE.

The PROSA Core interfaces with the Apache ODE engine to collect information such as deployed processes and events resulting from executing the processes. Monitoring data (e.g., service response times) obtained from the collected events are stored in the data base.

The PROSA Client visualizes the stored monitoring QoS data.

References

[1] https://forge.fi-ware.eu/frs/download.php/38/PROSA_user_manual_v1.0.pdf
[2] https://forge.fi-ware.eu/frs/?group_id=15

Trace Analyzer

Introduction

Trace Analyzer is an Eclipse plugin for graphical and numerical analysis of performance traces. Trace Analyzer reads in traces from various sources, and displays time-based graphical visualization of the program execution as well as a list of trace contents and the event details for selection.

On Linux, supported data sources include System Activity Report (SAR) and TCPDump.

Information

Name: Trace Analyzer

Version: 1.3.0

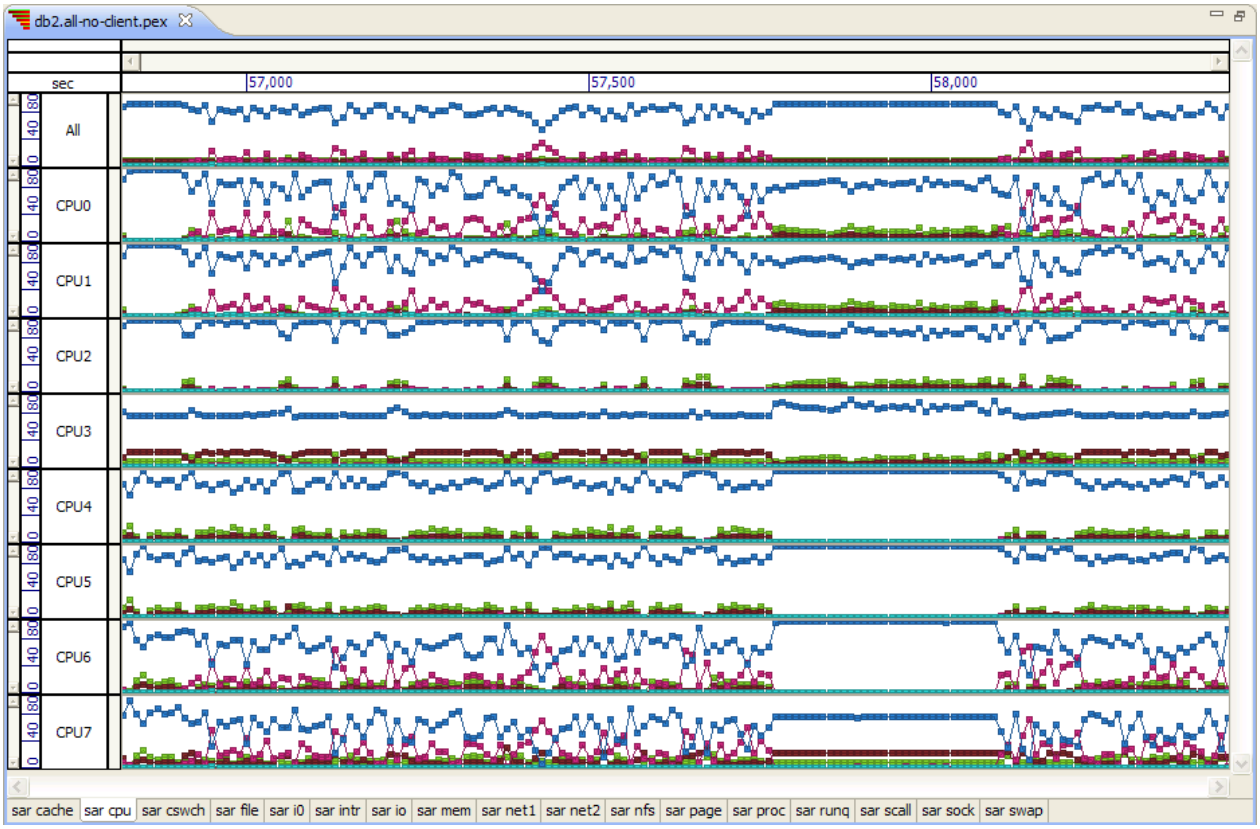
License: IBM License File [1]

Documentation: Press F1 in Eclipse and navigate to Trace Analyzer

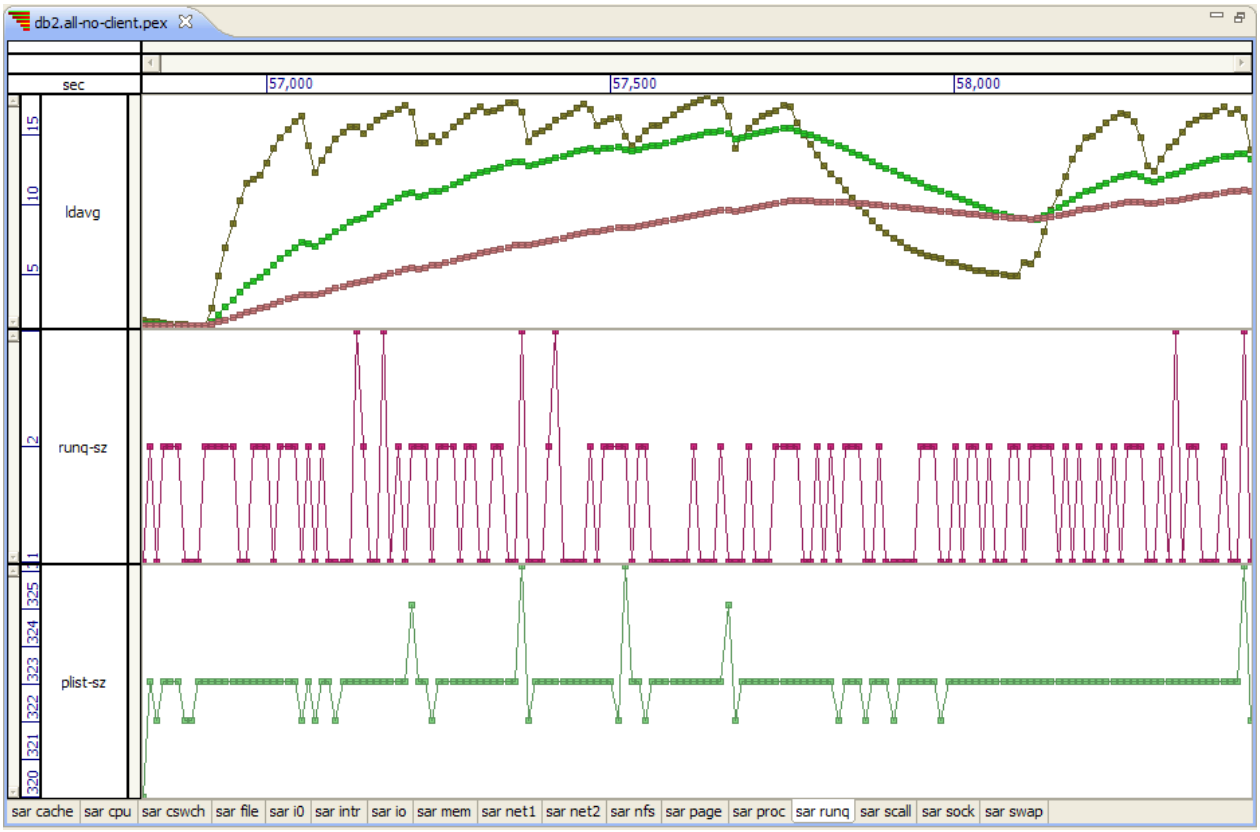
Download: Please send an email to Marcel Zalmanovici (marcel [at] il.ibm.com) for a link to the Eclipse plugins.

Examples

SAR

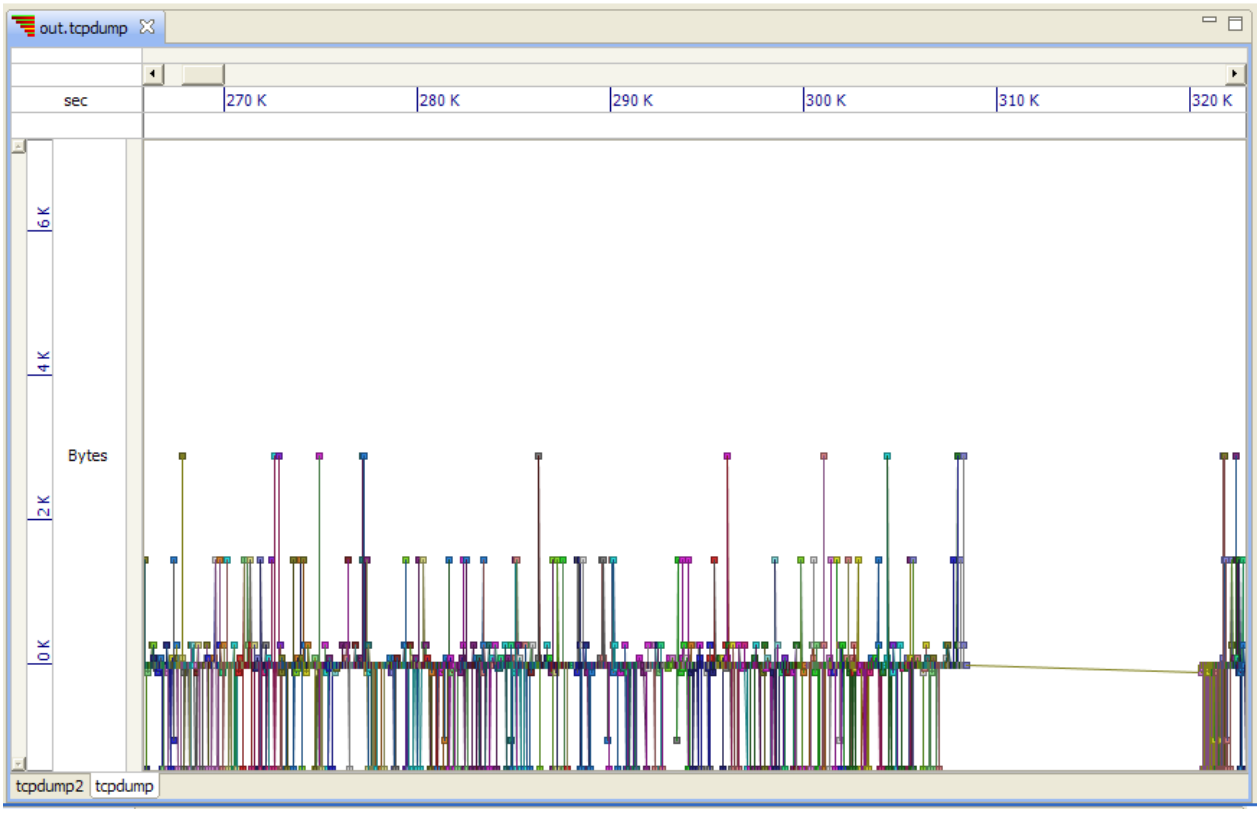


This view shows the CPU utilization data: time spent in user- and system-level code, idle time, idle time waiting for I/O, time executing with nice priority, involuntary wait time due to hypervisor servicing another request. The view shows a row per CPU; the top row shows the average for all CPUs.



This view shows the run queue statistics. The top row shows one-, five, and fifteen-minute load averages. The second row shows the run queue length, and the third row shows number of processes and threads in the process list.

TCPDump



This view shows the amount of data being passed in time; the *x* coordinate is time, and the *y* coordinate the amount of data. The amount is positive if data is being sent and negative if data is being received. Each series in the graph corresponds to a single session, identified by the local and remote ports and the remote host name.

References

[1] https://forge.fi-ware.eu/frs/download.php/36/TAlicense.dat

SoPeCo

Introduction

SoPeCo is a framework for systematic performance evaluations of software systems, based on systematic measurements, statistical methods, and machine learning. For more details visit http:// www. sopeco. org. Being improved and adapted for FI-WARE, SoPeCo plays an essential role in FI-WARE by enabling the performance-aware development of FI Applications including performance-based comparison and selection of GE implementations.

Information

Name: Software Performance Cockpit
Version: 0.1
License: (in progress) either BSD or EPL (Eclipse Public License)
Documentation: https://github.com/sopeco/Software-Performance-Cockpit/wiki/HowTo
Download: Please send an email to Roozbeh Farahbod (roozbeh.farahbod [at] sap.com) for a link to the Eclipse plugins.
Scope: Measurement-based Performance Testing and Analysis

Components

SoPeCo is developed based on Eclipse and the OSGi framework. Its binary distribution is available from a pair of eclipse update sites:

- SoPeCo Core, which contains the core components of SoPeCo
- SoPeCo Libs, which contains required libraries

The update sites are currently for FI-WARE internal try-out only. To obtain the links, please send an email to roozbeh.farahbod [at] sap.com.

Documentation

A basic tutorial that covers the main concepts of SoPeCo is available here: https:/ / github. com/ sopeco/ Software-Performance-Cockpit/wiki/HowTo

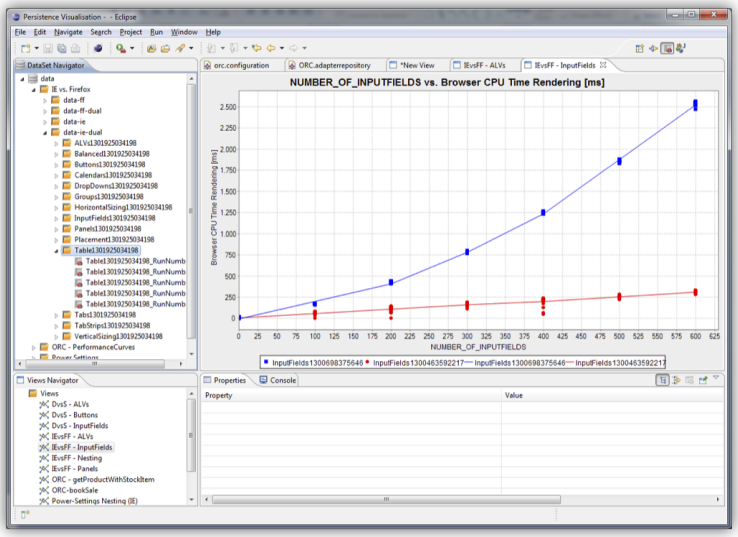
Current Version

In light of FI-WARE use-cases, the design and implementation of the original SoPeCo framework is being revised mainly to achieve the following goals:

- simplifying the configuration and application of SoPeCo in development
- providing a pure plugin-based architecture that enables future extensions as required by FI-WARE (or similar) use-cases

The latest release of SoPeCo is version 0.1.

Screen Shots



alt text