WissGrid

Dokumentation

Arbeitspaket 3: Langzeitarchivierung von Forschungsdaten

JHOVE2 module for ROOT files[1]

Autoren	Arbeitspaket 3: Langzeitarchivierung von Forschungsdaten
Editoren	Kalinin
Datum	5. April 2012
Dokument Status	Entwurf
Dokument Version	0.2.1

Änderungen

Version	**Datum**	Name(n)	**Kurzinfo**
0.2.1	5. April 2012	Kalinin	JHOVE2-2 and Cache info added

[1] This work is created by the WissGrid project. The project is funded by the German Federal Ministry of Education and Research (BMBF).

Inhaltsverzeichnis

1 Preview

1.1 Abstract

JHOVE2 is a framework and application for next-generation format-aware characterization of digital objects. The function of JHOVE2 is encapsulated in a series of modules that can be configured for use within the framework's plug-in architecture. The ROOT module provides characterization services for ROOT family of formats. The module has been developed in the framework of WissGrid - Grid for the Science project funded by Bundesministerium für Bildung und Forschung (German Ministry of Education and Research) at the University of Wuppertal. Experimental particle physics group from the university participates in the ATLAS experiment, which is one of the four high energy physics experiments being performed at CERN, Geneva, Switzerland. The amount of data produced by modern elementary particles experiments is as big as dozens of petabytes per year, and the major part of the data is stored in ROOT format. Such high volume of data requires characterization of stored data, and this can be achieved by using the ROOT module for JHOVE2. The characterization is based upon metadata content of ROOT files and their names. The metadata stored in files is extracted using external calls of ROOT package.

2 Design

2.1 Introduction

JHOVE2 is a framework and application for next-generation format-aware characterization of digital objects.

How is this design section organized?

This main page describes the system design in terms of packages, classes, relationships, and behavior. Several attached worksheets address specific aspects of the overall system design, such as user interface and database design.

What are the most important facts about this design for a developer?

The idea of the package is to identify ROOT files based on their signature and to externally call *root* to extract meta data(names of data objects stored in the file). The current implementation has selectable granularity starting from TTree up to atomic objects themselves.

What are the prioritized goals of this design?

1. Correctness
2. Feasibility
3. Understandability
4. Modularity

5. Extensibility

2.2 UML structural design

See sequence and class UML diagrams.

3 Technische Umsetzung

The RootModule has been implemented using external calls to ROOT package. The package provides a set of tools accessing and extracting of data stored in ROOT files. The current implementation extracts trees, branches and their names and objects themselves. For the moment only simple information is extracted for object(e.g. size) but more granular access is also possible and depends upon requirements. Unfortunately, there are no up-to-date JAVA API for ROOT files and this is exactly the reason why external shell calls are needed to extract information from ROOT files. An alternative would be to use JNI but this is by far more confusing and less flexible.

3.1 Type of Implementation

Type of system:

- Unix-style command
- Reusable library

Programming Language(s):

- Java
- ROOT
- Bash shell scripts
- XML

Data Storage:

- Flat files using XML
- Flat files JSON format
- Flat files

3.2 Configuration

As stated in the JHOVE2 Architectural Overview: JHOVE2 uses the Spring Java enterprise platform (www.springsource.org) to manage module instantiation. JHOVE2 is implemented in terms of the Spring philosophy of dependency injection: all relationships between JHOVE2 components are defined in external configuration files, not the Java source code, so they can be modified easily by a JHOVE2 user at the time of installation or invocation. Spring Config File The Spring configuration file for the RootModule and the Root

Format is `config/spring/module/format/root/jhove2-root-config.xml` The Spring bean with id="RootModuleïs used to create instances of the module. Two other beans with id="RootFormatänd id="RootIdentifieräre used to create instances of Format and Identifier respectively. Current implementation of the beans allows only one parsing behavior of this module.

3.3 Runtime Environment

Processes:

- Main application process
- Operating system services or drivers
- Batch system if it is being run on a cluster

Configuration Files:

- `config/droid/DROID_SignatureFile_V20.xml`: contains ROOT file signature description.
- `config/spring/module/format/root/jhove2-root-config.xml`: Spring format identifier.

Data files:

- .root: files to be processed.
- .xml or .txt: output.

Temporary files:

- /tmp/: JHOVE2 creates temporary files in this directory which can be as big as original files themselves. One has to take care of this when processing big files.

3.4 Operational Procedures

- **Install:** See the Installation guide
- **Configuration:** There is no specific configuration files if one submits jobs locally.
- **Compilation:** See the Installation guide
- **Monitor Activity:** Any favorite monitoring can be used.
- **Periodic Cleanup:** All temporary files are automatically removed.

3.5 Performance and Scalability

It takes about 6 seconds in average to process a ROOT file on a relatively modern machine. A test has been performed in Wuppertal to estimate scalability. A test has been run on Pleiades 5 cluster. The computing facility includes around 900 CPU cores, a storage system having 14 disk servers each equipped with 10 Gb network interfaces. The ROOT files were first copied from the storage system to the local disks and then processed by JHOVE2. The average file size was estimated to be 400 MB and the average processing time was 14 seconds per file. That means that one million of such files can be processed in less than 5 hours.

The current version also uses TTreeCache objects which are used to cache ROOT file structure for further access optimization. This is especially useful if one has to extract the finest granularity of ROOT files. A typical Tier2 has several millions of ROOT files and this caching mechanism is essential.

4 Installation und Benutzung

4.1 Introduction

This document describes how to install and get started with JHOVE2 module for ROOT files.

4.2 Minimal System Requirements

System requirements are described in the release notes.

4.3 Installation

What other software must be installed first?

Before you can install this product, you must install the following packages:

- Java SDK
- Maven
- ROOT

How do I install JHOVE2 module for ROOT files?

Please follow the steps described in the User's guide for JHOVE2. A version of JHOVE2 with ROOT module can be downloaded from WissGrid SVN repository.

What if I encounter problems?

Please see the troubleshooting section in the FAQ.

4.4 Getting Started

How can I quickly get started using JHOVE2 module for ROOT files?

1. Compile JHOVE2 with ROOT module
2. Copy `target/jhove2-2.0.0.jar` to `lib/`
3. Setup ROOT environment
4. Execute `jhove2.sh [filename]`

5 Benutzer FAQ

5.1 General Information

What is JHOVE2 module for ROOT files?

It is a JHOVE2 module for ROOT files. Read our JHOVE2 module for ROOT filesoverview.

Who should use JHOVE2 module for ROOT files?

Anyone who wants to identify, characterize, validate and to assess ROOT files. Read more about our target audience and benefits.

5.2 Download and Upgrade

How can I obtain JHOVE2 module for ROOT files?

JHOVE2 module for ROOT filesincluding JHOVE2 can be downloaded from here.

How do I upgrade from an older version of JHOVE2 module for ROOT files?

Installation instructions are available on-line here.

5.3 Troubleshooting

In case of problems see JHOVE2 documentation.

6 JHOVE2 module for ROOT files 0.2.1: Versionsinformationen

6.1 Introduction

This document contains the release notes for JHOVE2 module for ROOT filesversion 0.0.2. The following sections describe the release in detail and provide late-breaking or other information that supplements the main documentation.

This is a developer release for internal evaluation only. Please report any issues via the internal issue tracker.

This is an early access release for evaluation and usage by select partners. Your feedback is important to us, please help us make this the best product possible.

This is an early access release for wide evaluation and usage. Your feedback is important to us, please help us make this the best product possible. Keep in mind that we are continuing to work on JHOVE2 module for ROOT filesand things may change in the future.

This is the first full release of this product. See the product website for a complete description.

6.2 Minimum system requirements

- **System Processor:** 800MHz
- **System Memory:** 256MB
- **Free Disk Space:** The maximum file size to be processed
- **Operating System:** Windows 2000, Windows XP, Mac OS X, Linux (kernel 2.4)
- **Networking:** Internet access
- **Existing Software:** java, bash, root

6.3 Recent Changes

6.4 Known Problems and Workarounds