



## Distributed Database Access and Data Stream Management

### Data Stream Management—An Implementation Report<sup>1</sup>

|                   |                     |
|-------------------|---------------------|
| Deliverable       | D4.8                |
| Authors           | Tobias Scholl       |
| Editors           | Tobias Scholl       |
| Date              | July 31, 2008       |
| Document Version  | 1.0.0               |
| Current Version   | 1.0.0               |
| Previous Versions | 0.1.0, 0.2.0, 0.4.0 |

#### **A: Status of this Document**

Deliverable D8 of working group 4.

#### **B: Reference to project plan**

Eighth deliverable of working group *Distributed Database Access and Data Stream Management*.

---

<sup>1</sup>This work is part of the AstroGrid-D project and D-Grid. The project is funded by the German Federal Ministry of Education and Research (BMBF).

**C: Abstract**

This deliverable gives an implementation report for the data stream management system.

**D: Change History**

| <b>Version</b> | <b>Date</b> | <b>Name</b>   | <b>Brief summary</b>                               |
|----------------|-------------|---------------|--|
| 1.0.0          | 31.07.2008  | Tobias Scholl | First release. Incorporated the comments.          |
| 0.4.0          | 16.07.2008  | Tobias Scholl | Submitted draft for comments to the whole project. |
| 0.2.0          | 30.06.2008  | Tobias Scholl | Added details on the individual module changes.    |
| 0.1.0          | 20.08.2007  | Tobias Scholl | Initial draft.                                     |

---

E:

## Contents

|  |          |
|--|----------|
| <b>Abstract</b>                              | <b>2</b> |
| <b>Change History</b>                        | <b>3</b> |
| <b>1 Overview</b>                            | <b>5</b> |
| <b>2 Dependency changes</b>                  | <b>5</b> |
| <b>3 Module starglobe-operators</b>          | <b>5</b> |
| 3.1 Package streamglobe.ogsadai . . . . .    | 5        |
| 3.2 Documentation . . . . .                  | 5        |
| <b>4 Module streamglobe-clients</b>          | <b>5</b> |
| <b>5 Module streamglobe-middleware</b>       | <b>6</b> |
| <b>6 Module streamglobe-services</b>         | <b>6</b> |
| 6.1 Package streamglobe.ws.cp.impl . . . . . | 6        |
| 6.2 Other Changes . . . . .                  | 7        |
| <b>7 Module streamglobe-tools</b>            | <b>7</b> |
| <b>References</b>                            | <b>8</b> |

## 1 Overview

This deliverable summarizes the major development changes since the previous deliverables on the data stream management system [1, 3].

## 2 Dependency changes

Two new dependencies were added to the data stream management *pom.xml* in the course of implementing the OGSA-DAI WSRF-2.2 integration in StarGlobe conducted in Deliverable D4.7 [2]. The new jar files are:

**ogsadai-activities** The various activities provided by OGSA-DAI.

**ogsadai-core** The core library of OGSA-DAI

Both libraries are available on the OGSA-DAI download website.<sup>2</sup>

## 3 Module starglobe-operators

This module was updated to contain the OGSA-DAI connectivity.

### 3.1 Package streamglobe.ogsadai

This package contains all classes used for connecting the data stream management with OGSA-DAI. A new content server, *streamglobe.ogsadai.RelationalContentServer*, allows sending data streams directly from a OGSA-DAI service, while *streamglobe.ogsadai.RelationalStreamIterator* provides the interface for writing data stream into a relational database via OGSA-DAI. For examples and a detailed description how these components are configured we refer to the corresponding deliverable [2].

### 3.2 Documentation

Additional example query execution plans and scenario files are added to the usecase documentation (in the directory *doc/usecases*).

## 4 Module streamglobe-clients

Several client classes which we use for setting up a StarGlobe network were internally changed to reflect the enhancements in the configuration mechanism for ContentProviders. These changes are described in more detail in Section 6.1.

---

<sup>2</sup>[http://www.ogsadai.org.uk/downloads/archive/ogsa-dai\\_wsrf\\_2.2/](http://www.ogsadai.org.uk/downloads/archive/ogsa-dai_wsrf_2.2/)

Moreover, the client for ContentProviders, *streamglobe.client.cpcc.ContentProviderControlClient*, now also supports the HTTPS protocol. Thereby, we can now also control data streams which are served by a secured Globus container.

## 5 Module streamglobe-middleware

Within this module, the major changes were simplifications and generalizations of the *ContentProvider* WSDL-interface. In the course of the updates, the method *provideFileContent* has been generalized to *provideContent* and the method *setSleepTime* has been removed from the general *ContentProvider* interface.

Additionally, minor improvements of the build and deploy process for GT4 have been added to *build-gt4.xml*.

Finally, we enhanced the *PeerServiceReference* by a method for retrieving the IP address for the peer.

## 6 Module streamglobe-services

### 6.1 Package streamglobe.ws.cp.impl

When implementing the OGSA-DAI support and integrating non-Java applications on the *ContentServer* side, interfaces and implementation details have changed slightly.

General constants for content servers are collected in *ContentServerConstants*. The *ContentServer* interface was changed to the only method *configure(Properties)*. The former methods implied that content was served from a file.

The new interface only expects Java Properties (key-value pairs) to be specified. The existing *ContentServer* implementations (*FileContentServer*, *XMLFileContentServer*) basically reused the existing methods within the body of the configuration method.

For *ControlledContentServers* the functionality of the *setOptions(int)* method is also performed within the configure method of the *ContentServer*.

In order to ease the development of new *ContentServers* that basically follow the same processing pattern as in the *XMLFileContentServer* (using a *DefaultHandler* for XML parsing), an abstract supertype was constructed (*AbstractXMLContentServer*). The extension point for subclasses is the *initializeReader* method which encapsulates the actual source for the input data (be it a file or a TCP socket).

Using the *AbstractXMLContentServer*, creating a new *ContentServer* for reading data from a *ServerSocket* is now very simple. In addition to the the parameters of an *XMLFileContentServer*, two further parameters, *stream.input.host* and

*stream.input.port* are specified as in the following example.

```
<streams>
  <kindDefinition>
    <kind name="socket" class="streamglobe.ws.cp.impl.XMLSocketContentServer"/>
  </kindDefinition>
  <stream sid="stream-0" type="socket">
    <dtd filename="/tmp/dsm-example/nbody_nested.dtd"/>
    <param name="stream.input.host">localhost</param>
    <param name="stream.input.port">33631</param>
    <param name="stream.server.port">9009</param>
    <param name="stream.sleep.time">1</param>
    <param name="stream.injection.mode">manual</param>
  </stream>
</streams>
```

With the `XMLSocketContentServer`, `StarGlobe` provides now the mechanism integrate third party content providers that send XML-data on a TCP socket.

## 6.2 Other Changes

We slightly changed the managing of aliases, i.e., different names for the same data stream, within the *streamglobe.services.management.ReuseInfoManager*.

Within the *StreamServerManager* of the *streamglobe.servivces.p2p.engine* package, we introduced an explicit variable for specifying the number of tries for a new TCP port in case the randomly chosen port from the *TCP\_PORT\_RANGE* should be already in use.

Besides fixing a bug in the *ResultDispatcher* implementation (package *streamglobe.services.p2p.engine.operator*), we enhanced all operator classes within the *streamglobe.services.plan* package by adding a method *appliesTransformation* which allows us to determine whether the operators apply transformations.

Support for statistics about the local network interface was added to *streamglobe.statistiX.hardware.BasicMonitor*.

Finally, the implementations for the *SpeakerPeerResource* (within the package *streamglobe.ws.speaker.impl*), *Peer*, and *PeerResource* were updated in order to provide operator statistics (both in *streamglobe.ws.peer.impl*).

## 7 Module streamglobe-tools

No changes.

**F: References / Bibliography****References**

- [1] A. Carlson and T. Scholl. Distributed Database Access and Data Stream Management: Prototype for Manual Query Execution Plans. Deliverable D4.2, AstroGrid-D project, August 2006. <http://www.gac-grid.de/project-documents/deliverables/wp4/wg4-d2-1.0.0.pdf>.
- [2] A. Reiser and T. Scholl. Distributed Database Access and Data Stream Management: Integration of Streaming and Persistent Data Management. Deliverable D4.7, AstroGrid-D project, February 2008.
- [3] T. Scholl and A. Reiser. Distributed Database Access and Data Stream Management: Distributed Function Providers. Deliverable D4.4, AstroGrid-D project, February 2007. <http://www.gac-grid.de/project-documents/deliverables/wp4/wg4-d4-1.0.0.pdf>.