Enabling Data Transport between Web Services through alternative protocols and Streaming

Spiros Koulouzis, Edgar Meij, M. Scott Marshall, Adam Belloum
Informatics Institute, University of Amsterdam, The Netherlands

4th IEEE International Conference on e-Science 2008,University Place Indianapolis Indiana USA

Introduction

e-Science Applications can be implemented as Web Services (WS), as they may offer interoperability and flexibility in a large scale distributed environment. WS can be combined in a workflow so that more complex operations may be achieved, but any workflow implementation is potentially faced with a *data transport problem:*

•In service orchestration, **all data is passed to the workflow engine** before delivered to a consuming WS

•Data transfers are made through **SOAP**, which **is unfit for large data transfers**

•Third party file transfer results in **unnecessary intermediate transfers** that slow down workflow execution and place excessive demand o storage resources.

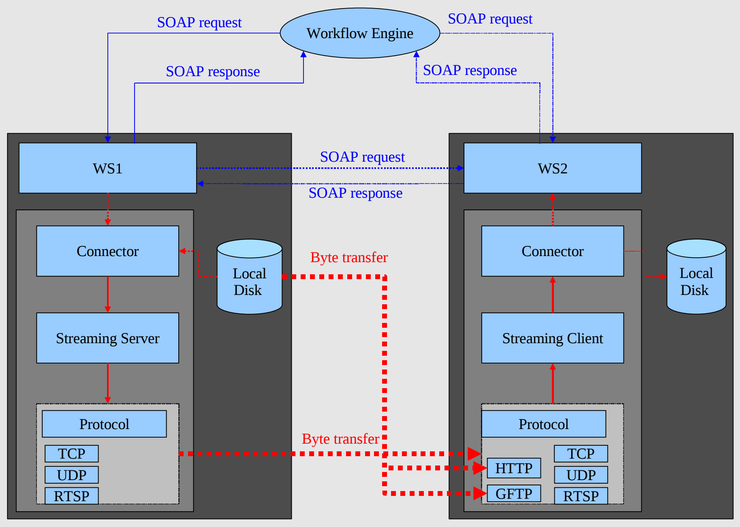
A Streaming Library

Address the *data transport problem* the use of streaming between WS through the implementation of a a Java library is introduced.

• The library **delivers data to a consuming WS** with alternative protocols to SOAP (TCP,HTTP,GridFTP, etc.), thus addressing the first two problems.

•One solution to the third problem, **streaming** is described as **a way to deliver data to a web service** without the need for intermediate file transfers.

Our Streaming library is a modular, client/server design that uses **SOAP as a control channel while the actual data transport is accomplished by the various protocol implementations**.

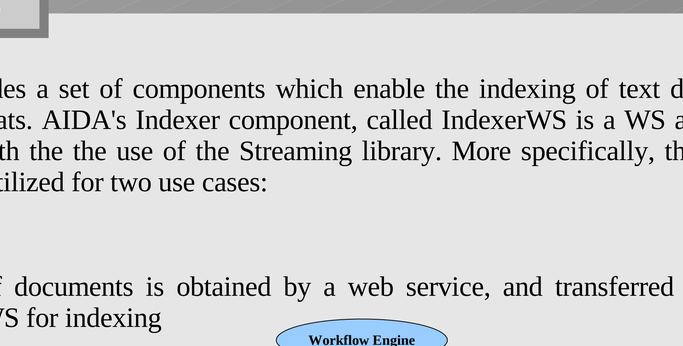


Use Case

AIDA provides a set of components which enable the indexing of text documents in various formats. AIDA's Indexer component, called IndexerWS is a WS able to index document with the the use of the Streaming library. More specifically, the Streaming library was utilized for two use cases:

•A set of documents is obtained by a web service, and transferred to the IndexerWS for indexing

•A PDF DocumentHandler (A component responsible of extracting content from various document formats) is implemented as a web service, for extracting text from a set of PDF files. This text is streamed directly to the IndexerWS for indexing.



Results

The results are obtain using our proposed approach on two tasks: file and streaming transport.

File Transport

•**GridFTP**: Stable behavior, easy copes with large file sizes.

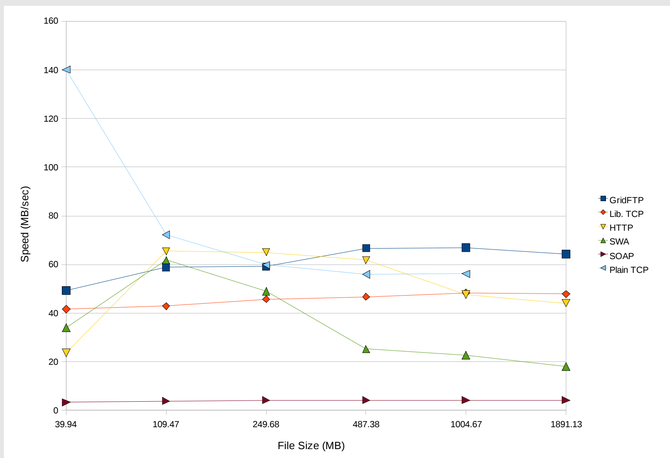
•**HTTP**: faster only for file sizes of 109.47 and 259.48 MB, possibly because of Tomcat's behavior.

•**SwA**: For large file sizes, disk I/O introduces significant overhead.

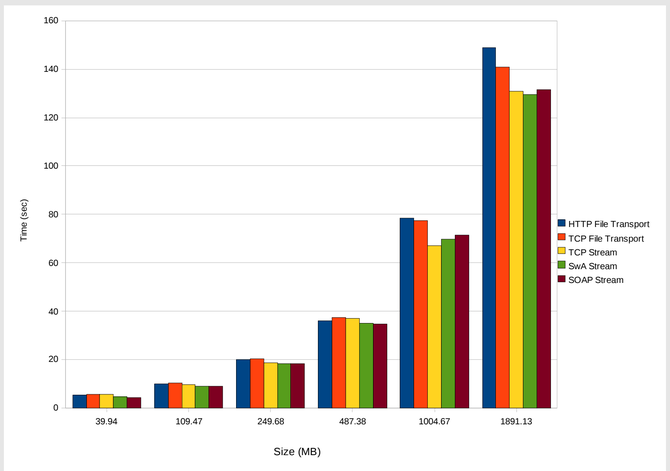
•**TCP:** fails to match HTTP's performance, something that can be attributed to the lack of TCP tuning parameters.

•**SOAP**: Is the slowest because “chunking” is used, preventing crashing as well as full bandwidth utilization

•All of the protocols exhibit a speed reduction for file sizes of 1891.13 MB. This is because this experiment concerns 5 files



File Transport Speed



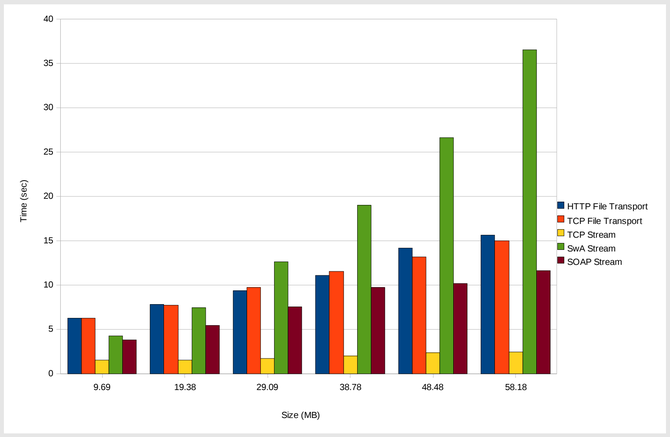
Workflow Execution Time

Streaming Transport

In this experiment the execution time of the indexing workflow is measured.

•**SOAP streaming** is faster than **HTTP file transfer**, (experiment with file size of 1891.13 MB) by approximately 20 sec.

•With **no loading and processing overhead** streaming is even faster.



Workflow Execution Time

Discussion

Streaming data directly to web services can benefit workflow execution time and storage demands.

Streaming may not be applicable when a consuming WS must obtain the entire data set before operating on it. Nevertheless this approach enables WS to work on large data sets, without excessive storage demands

We have identified and addressed the problem of transferring large data sets between web services.

We have described a modular and extensible Streaming library able to transfer large files, as well as connect web services in a continuous data pipeline as an alternative approach to data transport.

In our proposed approach, **SOAP is used as a control channel**, while data is transferred using the most suitable protocol for either file or streaming transfers.

In addition, the use of **streaming can speed up workflow execution time** by eliminating disk I/O latency and enabling WS to work on data as it is generated, rather than waiting for an entire file to be delivered.

This work was carried out in the context of the Virtual Laboratory for e-Science project. This project is supported by a BSIK grant from the Dutch Ministry of Education, Culture and Science (OC&W) and is part of the ICT innovation program of the Ministry of Economic Affairs (EZ).