

РЕАЛИЗАЦИЯ РАЗДЕЛЬНОГО ФИЗИЧЕСКОГО ХРАНЕНИЯ XML ДАННЫХ В РАМКАХ СЕРВЕРА XSS

XSS (XML Storage System) – система хранения данных в формате XML фирмы ООО «Деловые Консультации СПб» решает следующие задачи:

1. Хранение множества XML документов (на основе XML файлов, SQL2005, Native XML DB).
2. Обработка ссылок между документами с поддержкой.
 - Итеративной загрузки частей XML-документа при обращении к линкованным элементам (Итеративный XLink);
 - Работы с общим полным деревом посредством XPath.
3. Обеспечение многопользовательского доступа в части.
 - Блокировок элемента;
 - Блокировок поддерева;
 - Блокировок поддерева с элементами.
4. Обеспечение хранения истории и откатов изменений.
5. Поддержка протоколов взаимодействия с XSS из сторонних проектов.
6. Решение административных задач в рамках клиентской части системы (CXSS).

Проблема физического хранения XML данных в рамках системы XSS имеет множество решений от файлового представления документов до реляционного в рамках СУБД. Каждый из вариантов хранения имеет свои особенности в экономических, ресурсной и мощностной характеристиках. Применение конкретного варианта физического хранения обуславливается особенностями задачи решаемой на базе XSS.

Для поддержки всех этих механизмов комплексно в рамках сервера XSS разработан унифицированный протокол взаимодействия с физическим хранилищем в рамках абстрактного класса `AbstractAdapter` и механизмы настройки для каждого документа особенностей его физического хранения. Система команд унифицированного протокола включает следующий набор команд:

`П1 = (command, data): (stream, err_code)`

`command = {"get_doc" | "set_doc" | "doc_list" | "delete_doc"}`

`П1.1 = ("get_doc", xml_doc_id) : (xml_stream | null, err_code)`

`П1.2 = ("set_doc", xml_stream) : (null, err_code)`

`П1.3 = ("doc_list", null) : (xml_stream | null, err_code)`

`П1.4 = ("delete_doc", xml_doc_id) : (null, err_code)`

`П1.1` - получить XML документ. `xml_doc_id` – идентификатор документа. `xml_stream` – поток, содержащий XML документ или `null`.

`П1.2` – сохранить XML документ. `xml_stream` – поток содержащий XML документ.

`П1.3` – получить список XML документов в хранилище. `xml_stream` – XML поток, содержащий список идентификаторов существующих документов.

`П1.4` – удалить XML документ. `Xml_doc_id` – идентификатор документа.

В потомках `AbstractAdapter` инкапсулирована поддержка физического хранения по унифицированному протоколу в рамках классов `FileStorageAdapter`, `SQLStorageAdapter` и `SednaStorageAdapter`. `FileStorageAdapter` обеспечивает физическое хранение XML документов в рамках файловой системы, `SQLStorageAdapter` – на базе реляционной базы данных (Microsoft SQL Server 2005), а `SednaStorageAdapter` использует `Native Xml Db Sedna`. SQL

Server 2005 и Sedna обеспечивают транзакционность операций. SQLStorageAdapter поддерживает файловые группы и T-SQL запросы, что позволяет распределять нагрузки по ресурсам сервера, а в SednaStorageAdapter унаследована поддержка XQuery. Так же, SQLStorageAdapter позволяет использовать при хранении как серверный тип данных XML, так и текст (VARCHAR(MAX)), обеспечивая в первом случае автоматическую валидацию XML данных и возможность использования XML индексов, а во втором – максимальное быстродействие.

С помощью разработанной библиотеки появилась возможность отдельного задания для каждого документа сервера XSS особенностей его физического хранения и использования специфичных запросов. А с помощью аппарата линковки сервера XSS - возможность разрезания XML документов и разнесения их по различным физическим хранилищам. При этом, клиенты сервера получают единый линкованный документ, но могут использовать особенности уровня физического хранения для серверной поддержки реализации специфичных задач.

В результате работ был сформулирован унифицированный протокол взаимодействия сервера XSS с физическим хранилищем. Спроектирована архитектура и реализована подсистема доступа к физическому хранилищу, в рамках которой было инкапсулировано взаимодействие с файловым представлением XML, SQL Server 2005 и Native XML Db Sedna. В данный момент библиотека отдельного физического хранения находится на стадии комплексного тестирования.