УДК 004.42

Е. В. Ерофеев, руководитель группы разработки, Группа компаний "АйТи", Москва,
e-mail: EErofeev@it.ru,
И. Е. Пелепелин, канд. физ.-мат. наук, вед. науч. сотр., Национальный исследовательский университет "Высшая школа экономики",
e-mail: IPelepelin@blogic20.ru

Анализ производительности протокола CMIS на примере его реализации в Alfresco и IBM FileNet

Обсуждаются результаты тестирования скорости выполнения базовых операций работы с документами, реализованных на основе протокола CMIS версии 1.0, в сравнении с использованием Native API хранилищ ECM-систем Alfresco и IBM FileNet. Результаты показывают значительное снижение производительности при использовании CMIS. Даны практические рекомендации по определению границ, в рамках которых целесообразно использование протокола CMIS.

Ключевые слова: *CMIS, производительность, ECM-система, Alfresco, IBM FileNet*

Введение

Открытый стандарт CMIS [1] призван обеспечить интероперабельность веб-сервисов по отношению к существующим системам класса *Enterprise content management* (ECM) и хранилищам неструктурированной информации, в первую очередь для систем-лидеров, которые определяются согласно исследованиям Гартнера [2]. Разработанный стандарт позволяет поддерживать одновременную работу с большим числом хранилищ, делая доступным контент, который уже имеется на предприятии. Таким образом, он призван снизить ограничения, связанные с невозможностью работы одновременно с несколькими хранилищами неструктурированных данных в мультивендорных и мультирепозитарных средах крупных организаций, для которых типично наличие сразу нескольких ECM-систем. По данным опросов [3], если бы ECM-решения могли интегрироваться со сторонними информационными системами, они играли бы значительно более важную роль на производстве.

В течение последних двух лет CMIS был реализован на ряде промышленных ECM-платформ, таких как Alfresco, EMC Documentum, IBM FileNet, MS SharePoint. Однако для того чтобы начать использовать CMIS в промышленных разработках, необходимо убедиться, что конкретная его реализация на конкретной ECM-платформе обеспечивает необходимые параметры производительности.

В данной статье приведены новые результаты исследований, на основании которых каждый желающий сможет сделать для себя определенные выводы о применимости конкретных реализаций протокола CMIS для работы с документами в хранилищах неструктурированной информации.

В настоящее время в сообществе разработчиков сложилось общее понимание того, что реализация протокола CMIS на базе основных ECM-платформ не позволяет добиться нужной эффективности по быстродействию. Однако количественных результатов сравнительного анализа, подтверждающих это мнение, пока никто не опубликовал. В настоящей статье представлены именно количественные результаты, позволяющие устранить этот недостаток.

Особенностью проведенных исследований является то обстоятельство, что сравнение проводилось, прежде всего, между реализацией протокола CMIS на промышленной ECM-платформе и эталонной реализацией тех же самых функций с использованием "родного" программного интерфейса данной ECM-платформы, так называемого *"Native API"*. Кроме этого, в статье представлены замеры быстродействия работы исследуемых функций в зависимости от типа опера-

ционной системы (сравнивались MS Windows и Linux).

Необходимо особо подчеркнуть, что в рамках данного исследования не проводилось сравнительного тестирования производительности CMIS между платформами Alfresco и IBM FileNet. Полученные в ходе тестирования результаты для этих двух платформ также не подлежат сравнению, поскольку тестовые узлы, на которых было развернуто программное обеспечение Alfresco и IBM FileNet, имеют разную конфигурацию аппаратной части с разной производительностью. В связи с этим любое сравнение между ними является некорректным.

Постановка задачи

Для того чтобы ответить на вопрос о границах применимости протокола CMIS, потребовалось провести ряд тестов и получить фактические данные, которые можно анализировать и делать соответствующие выводы. Работа по проведению серии тестов включала в себя, прежде всего, развертывание инфраструктуры двух стендов (узлов): Alfresco и IBM FileNet. Следует еще раз отметить, что аппаратное обеспечение этих стендов было разным. По этой причине сравнительных тестов между Alfresco и IBM FileNet не проводилось.

Для того чтобы обеспечить сравнимость результатов тестов в рамках одной ECM-системы, было разработано специальное программное обеспечение, позволяющее легко переключаться между режимами тестирования, реализуя тем самым единый программный код для всех вариантов, а также корректность сравнения полученных результатов. Тесты проводили на двух массивах текстовых документов. Первый массив содержал 10 документов, второй массив — 1000 документов. Время, которое было затрачено на реализацию каждой отдельной операции при работе с документами хранилища, измеряли в миллисекундах. Для получения общей картины, характеризующей производительность реализации протокола CMIS в каждом из вариантов, были протестированы следующие основные операции по работе с документами хранилища.

- Создание документа с атрибутами. Под документом хранилища здесь подразумевается предметно-ориентированная сущность, которая содержит характеризующие ее атрибуты (поля).
- Создание документов с контентом. В качестве контента здесь может быть любой бинарный файл, например, изображение в каком-либо формате, файл в формате MS Word и т.п.
- Создание связей между документами. Связи могут быть как атрибутивные[1], так и ассоциативные[2]. Тип связи зависит от конкретной реализации хранилища. Протокол CMIS поддерживает оба вида связи, но жестко не регламентирует каким из них пользоваться. Под созданием связи подразумевались следующие действия: получение документа; присвоение (атрибуту или помещение в коллекцию связанных) идентификатора другого документа; сохранение.
- Получение документов с атрибутами — извлечение из хранилища карточки документа со всеми его полями.
- Получение документов с контентом — извлечение из хранилища документа с вложенным бинарным файлом.
- Получение связей между документами — извлечение из хранилища информации о связях между документами.

Производительность реализованного протокола CMIS сравнивалась с производительностью Native API, т.е. "родных" интерфейсов, встроенных в ECM-системы Alfresco и IBM FileNet. Для Alfresco, кроме того, было проведено сравнительное тестирование одних и тех же функций работы с документами для случаев, когда сервер приложений Alfresco работает под управлением операционных системы Windows и Linux.

Основная задача, которая решалась данным исследованием, заключалась в необходимости определения возможных потерь и рисков неблагоприятных событий, связанных со снижением производительности при использовании протокола CMIS в собственных разработках. Речь идет о потерях, которые могут возникать:

- на стороне протокола SOAP при интенсивной передаче данных, например, при экспорте/импорте больших массивов документов;
- на стороне ECM-системы, при неэффективном преобразовании понятий протокола CMIS во внутренние понятия ECM-системы и обратно.

Для решения поставленной задачи необходимо было провести замеры быстродействия выполнения следующих основных операций при работе с документами с точки зрения программного интерфейса: создание документа, извлечение документа из хранилища, создание связей между документами и извлечение этих связей из хранилища.

Конфигурация стенда и программная реализация

Для проведения экспериментов по замеру скорости выполнения функций работы с хранилищем документов, перечисленных выше, был собран программно-аппаратный комплекс, состоящий из двух узлов.

Узел тестирования CMIS в реализации Alfresco состоял из двух независимых серверов, параметры которых представлены в табл. 1.

Узел тестирования CMIS в реализации IBM FileNet состоял из трех взаимосвязанных серверов (табл. 2).

[1] Атрибутивная связь — связь, при которой один документ имеет атрибут, ссылающийся на другой документ. Например, документ "Поручение" имеет атрибут "Исполнитель", который ссылается на документ "Сотрудник".

[2] Ассоциативная связь — связь, при которой документ имеет ссылки на документы без привязки к конкретному атрибуту.

Таблица 1

Состав узла тестирования CMIS в реализации Alfresco

Параметр	Сервер Alfresco (Windows)	Сервер Alfresco (Linux)
Операционная система	Windows XP 32бит	CentOS ver. 5.5
Оперативная память	4 Гбайта	4 Гбайта
Дисковое пространство	40 Гбайт	20 Гбайт
Процессор	2-ядерный Intel Xeon Core I7	2-ядерный Intel Xeon Core I7
Прикладное ПО	Alfresco v3.4.0 (d 3370), Tomcat 6.0.26, MySQL	Alfresco v3.4.0 (d 3370), Tomcat 6.0.26, MySQL

Таблица 2

Состав узла тестирования CMIS в реализации IBM FileNet

Параметр	Сервер FileNet	Сервер — контроллер домена	Сервер СУБД
Операционная система	Windows Server 2008 R2 64бит	Windows Server 2008 R2 64бит	Windows 7 64бит
Оперативная память	8 Гбайт	4 Гбайт	4 Гбайта
Дисковое пространство	40 Гбайт	40 Гбайт	40 Гбайт
Процессор	2-ядерный Intel Xeon Core I7	2-ядерный Intel Xeon Core I7	2-ядерный Intel Xeon Core I7
Прикладное ПО	IBM FileNet P8	—	Microsoft SQLServer 2008 R2

Для проведения тестов на языке Java было разработано универсальное клиентское приложение, которое в зависимости от заданных параметров обращается либо к хранилищу документов Alfresco, либо к IBM FileNet. Также в зависимости от заданных параметров используется либо интерфейс протокола CMIS, либо Native API соответствующего хранилища.

Каждый из рассмотренных вендоров реализовал протокол CMIS с отклонениями от спецификации стандарта [1]. Так, например, в случае работы с реализациями CMIS для IBM FileNet и Alfresco пришлось учитывать специфику именования типов и работы со связями для каждой из платформ.

Что касается Native API, то различий, с точки зрения реализации, оказалось гораздо больше, поскольку вендоры не связаны никакими стандартами. Тем не менее с точки зрения выполняемых функций работы с хранилищем документов, Native API реализуются сходным образом.

Следует отметить, что при выполнении операции извлечения документов из хранилища через CMIS возвращаются только указатели на документы, но нет ни атрибутов, ни вложений (контента). Для того чтобы получить атрибуты, необходимо обратиться к каждому документу в цикле, что увеличивает общие затраты по времени. В противоположность к этому подходу, при получении документов через Native API возвращается коллекция документов сразу с атрибутами. Ниже будет дано разъяснение положительных и отрицательных аспектов этих двух подходов.

Полученные результаты и их анализ

В ходе экспериментов, помимо двух разных ECM-платформ, исследовалось также влияние на результат используемой на сервере операционной системы. В частности, система Alfresco была развернута на серверах с операционной системой как Windows, так и Linux. Несмотря на тот факт, что тесты проводились на массивах, состоящих из 10 и 1000 документов, различия в результатах, пересчитанных на один документ, оказались незначительными. По этой причине в настоящей статье приведены результаты только для массива из 10 документов.

В табл. 3 представлены результаты измерений выполнения функций по работе с документами при обращении к хранилищу Alfresco, развернутому на базе операционной системы Windows XP. Результаты получены на коллекции из 10 тестовых документов и представлены в трех столбцах. В столбце CMIS показано время (мс) при обращении через протокол CMIS. В столбце Native API приведены результаты, полученные при обращении через API системы Alfresco. В столбце Сравнительный анализ представлено отношение (частное от деления) времени из столбца CMIS ко времени из столбца Native API. Для сравнения в табл. 4 представлены аналогичные результаты тестов по производительности, но уже для хранилища Alfresco, которое работает под управлением Linux (параметры сервера представлены выше в табл. 1). Данные результаты также получены на коллекции из 10 тестовых документов.

Таблица 3

Результаты тестов для Alfresco (Windows)

Операция над документом	Параметр измерения	CMIS, мс	Native API, мс	Сравнительный коэффициент
Создание документов с атрибутами	Среднее время выполнения теста для одного документа	1279,2	133,1	9,61
Создание документов с контентом		1322,9	352,8	3,75
Создание связей между документами	Среднее время выполнения теста для одной связи	1637,4	313,4	5,22
Получение документов с атрибутами	Среднее время выполнения теста для одного документа	322,7	40,3	8,01
Получение документов с контентом		170,2	149,1	1,14
Получение связей между документами	Среднее время выполнения теста для одной связи	236,3	80,6	2,93

Таблица 4

Результаты тестов для Alfresco (Linux)

Операция над документом	Параметр измерения	CMIS, мс	Native API, мс	Сравнительный коэффициент
Создание документов с атрибутами	Среднее время выполнения теста для одного документа	812	79,3	10,24
Создание документов с контентом		138,8	134,2	1,03
Создание связей между документами	Среднее время выполнения теста для одной связи	1574,8	148,8	10,58
Получение документов с атрибутами	Среднее время выполнения теста для одного документа	326	22,9	14,24
Получение документов с контентом		280,8	97,1	2,89
Получение связей между документами	Среднее время выполнения теста для одной связи	415,4	47,9	8,67

В табл. 5 представлены результаты сравнительного тестирования скорости выполнения базовых операций с документами из хранилища FileNet, которое развернуто под операционной системой Windows. Параметры данного стенда были приведены в табл. 2.

На рис. 1 (см. вторую сторону обложки) представлены данные по производительности, взятые из табл. 3 и 5. Обе платформы развернуты под операционной системой Windows. Для каждой из шести функций работы с документами показано по четыре результата:

- первая колонка — CMIS в реализации FileNet;
- вторая колонка — Native API FileNet;
- третья колонка — CMIS в реализации Alfresco;
- четвертая колонка — Native API Alfresco.

На рис. 1 хорошо видно, что производительность CMIS одинаково сильно (для Alfresco в среднем в 8 раз, для FileNet в 3 раза) проигрывает по сравнению с производительностью Native API. Наиболее затратными являются функции, связанные с созданием документов, где снижение производительности получено примерно в 10 и 4 раза соответственно.

На рис. 2 (см. вторую сторону обложки) представлены данные по производительности, взятые из табл. 3 и 4. Одна и та же ECM-платформа Alfresco была развернута под операционными системами Linux и Windows, и для каждой из шести функций работы с документами также показано по четыре результата:

- первая колонка — CMIS в реализации Alfresco под Linux;
- вторая колонка — Native API Alfresco под Linux;
- третья колонка — CMIS в реализации Alfresco под Windows;
- четвертая колонка — Native API Alfresco под Windows.

По результатам тестирования хорошо видно, что функции работы с документами, реализованные через CMIS, по производительности значительно проигрывают аналогичной реализации, полученной через Native API ("родной" для ECM-платформы программный интерфейс). В случае Alfresco отличие почти на порядок, в случае FileNet разница меньше, всего в 3—4 раза, однако тоже заметная. Наибольшая разница наблюдается в операциях по созданию документа, созданию связи между документами и извлечению документа из хранилища вместе с его атрибутами.

Существенным может являться тот факт, что при использовании CMIS, в случае, когда необходимо по-

Таблица 5

Результаты тестов для IBM FileNet (Windows)

Операция над документом	Параметр измерения	CMIS, мс	Native API, мс	Сравнительный коэффициент
Создание документов с атрибутами	Среднее время выполнения теста для одного документа	264,7	60,2	4,40
Создание документов с контентом		530,2	191,2	2,77
Создание связей между документами	Среднее время выполнения теста для одной связи	591,9	138	4,29
Получение документов с атрибутами	Среднее время выполнения теста для одного документа	74,2	23,9	3,10
Получение документов с контентом		75,1	52,9	1,42
Получение связей между документами	Среднее время выполнения теста для одной связи	119,5	53,1	2,25

лучить список документов, сначала нужно получить перечень указателей на искомые документы, а затем в процессе перечисления поэлементно "подтянуть" необходимые данные по атрибутам. В противовес этому, в случае использования Native API вместе с запуском запроса на получение списка документов сразу же приходят данные по атрибутам.

Пакетная (постраничная) загрузка данных оказывается эффективнее при большом числе результатов запроса. Причина в том, что здесь важно не только быстро выполнить запрос, но и быстро отобразить его результаты, т.е. вывести список документов, для чего требуется подгрузить их атрибуты. В этом случае Native API оказывается в 3—10 раз быстрее CMIS.

Неожиданные результаты показали тесты для одного и того же хранилища, но под разными операционными системами. Так, в частности, для платформы Alfresco, развернутой под Linux, функция создания контента документа через CMIS работает также быстро, как и в случае обращения через Native API. При этом по остальным тестируемым функциям особых различий между операционными системами не наблюдалось.

Выводы

Представленные результаты тестовых экспериментов наглядно показывают, что производительность выполнения основных базовых операций по работе с документами из хранилища, реализованных на основе версии протокола CMIS 1.0, значительно хуже реализации через "родной" программный интерфейс (Native API).

В связи с этим использование CMIS целесообразно в кросс-платформенных приложениях, где необходимо обеспечить максимальную независимость от вендора, где не возникает потребность в работе с большими списками документов и/или не нужно обрабатывать большие массивы документов. В приложениях, где наоборот нужно обеспечить максимальную эффективность и быстродействие, рекомендуется использовать Native API, или самостоятельно разрабатывать адаптеры, скрывающие от сервисов более высокого уровня особенности обращений к хранилищу ECM-платформы различных вендоров.

Заключение

В статье описана методика тестирования и представлен анализ результатов тестирования основных базовых функций работы с документами и хранилищем документов через протокол CMIS. Результаты позволяют определить границы разумного использования протокола CMIS. Вместе с этим показано, что для ряда задач, где быстродействие становится наиболее критичным параметром, могут быть использованы функции, реализованные через Native API.

Принимая во внимание общий вектор развития систем управления контентом, использование CMIS приведет к тому, что ECM-хранилища станут в большей степени отчуждаемы от конкретной ECM-платформы. Вероятно, крупные вендоры хорошо понимают, что этот факт ослабляет их связи с корпоративными заказчиками. В силу этого они пока не стремятся вкладывать средства в развитие реализации CMIS на своих платформах. В связи с этим не следует ожидать быстрых изменений в производительности реализаций протокола CMIS [4]. С точки зрения функциональности, CMIS, безусловно, не покрывает все возможности, которые дает использование Native API. Однако стандарт развивается и готовится к выходу версия 1.1, а также идет работа над версией 2.0. При этом остается в силе тот факт, что качество и эффективность реализации CMIS целиком и полностью зависят от вендора, и это в первую очередь влияет на производительность приложений, работающих на протоколе CMIS.

Вместе с тем CMIS не обойден вниманием также и Open Source сообщества. Реализация проекта Apache Chemistry [5] позволяет всем вендорам JCR-совместимых приложений поддерживать CMIS.

Низкая производительность CMIS и, соответственно, слабый интерес к разработкам на основе CMIS косвенно подтверждают локальные опросы пользователей ECM-систем, проведенные корпорацией Generis в 2011 и 2012 гг. Их результаты свидетельствуют о том, что знание пользователей о протоколе CMIS в течение года не изменилось и, скорее, пошло на убыль [6]. Вместе с тем этот факт может трактоваться как традиционный для вывода на рынок новой технологии этап спада "информационного шума", который следует за "пиком завышенных ожиданий" и предшествует обычно наступлению зрелости технологии.

Данное исследование осуществлялось при финансовой поддержке Правительства Российской Федерации (Минобрнауки России) в рамках договора № 13.G25.31.0096 о "Создании высокотехнологичного производства кросс-платформенных систем обработки неструктурированной информации на основе свободного программного обеспечения для повышения эффективности управления инновационной деятельностью предприятия в современной России".

Список литературы

1. **OASIS** Standard Specification. Content Management Interoperability Services (CMIS) Version 1.0. 2010. URL: http://docs.oasis-open.org/cmis/CMIS/v1.0/os/cmis-spec-v1.0.pdf

2. **Ваулин В.** Оправдает ли CMIS надежды рынка СЭД? URL: http://www.cnews.ru/reviews/free/dms2010/articles/articles11.shtml.

3. **Waldhauser S.** 5 myths about the CMIS standard. URL: http://www.digitallandfill.org/2011/08/5-myths-about-the-cmis-standard.html.

4. **Apache** Software Foundation. Apache Chemistry. URL: http://incubator.apache.org/projects/chemistry.html.

5. **Bell T., Shegda K. M., Gilbert M. R., Chin K.** Magic Quadrant for Enterprise Content Management. Gartner, 2010. URL: http://www.ecmforhighereducation.com/wp-content/uploads/2011/09/Gartner-MQ-report-2010.pdf.

6. **Content** Management Plans for 2012. A Short Survey on CMS and CMIS. Generis. February 2012. URL: http://www. generis-corp.com/docs/CMIS and CMS 2012 Plans — Survey Responses.pdf

Software Quality Assurance Days

30 ноября - 1 декабря 2012, Минск, Беларусь

SQA Days является конференцией №1 на пространстве СНГ и одним из главных событий в Восточной Европе, посвященных тематике тестирования и обеспечения качества программного обеспечения.

Конференция охватит широкий спектр профессиональных вопросов в области обеспечения качества, ключевыми из которых являются:

- Методики и инструменты тестирования ПО
- Автоматизация тестирования ПО
- Подготовка, обучение и управление командами тестировщиков
- Процессы обеспечения качества в компании
- Управление тестированием и аутсорсинг
- Совершенствование процессов тестирования и инновации

Организаторы конференции приглашают к сотрудничеству докладчиков, партнёров и других заинтересованных лиц.

SQA DAYS#12

Twitter лента конференции: **#sqadays12**

WWW.SQADAYS.COM

проект компании “Лаборатория тестирования” (www.sqalab.ru)