

УДК 004.75

Оценка эффективности интеграции высоконагруженных банковских систем с использованием промышленной интеграционной шины

Шилин Г.М., студент

*Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Информационные системы и телекоммуникации»*

Научный руководитель: Недашковский В.М., к.т.н, доцент

Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана

Shilingri@gmail.com

1 Введение

Всё большее расширение бизнеса и области предоставляемых услуг банков приводит к росту различных новых информационных систем и модулей банка и, как следствие, возникает задача интеграции их между собой и уже существующими банковскими системами. Не секрет, что сбой банковской системы может стоить очень дорого как клиентам, так и репутации банка, поэтому самое пристальное внимание в процессе внедрения любой новой системы необходимо уделять ее грамотной интеграции с уже существующей банковской инфраструктурой.

2 Причины и предпосылки использования ESB

По мере расширения спектра предоставляемых банком услуг увеличивается и количество специализированных ИТ-систем, автоматизирующих те или иные бизнес-процессы. Это ведет к необходимости интеграции существующих систем между собой и с новыми внедряемыми решениями и бизнес-приложениями. Однако, банки имеют в своем арсенале внушительный набор разнородных приложений, которые в свое время разрабатывались и внедрялись без учета каких-либо требований по их возможной интеграции с внешними системами. В таких условиях интеграция становится крайне трудоёмким, долгосрочным и дорогостоящим проектом, что, в условиях жесткой конкуренции на финансовом рынке, абсолютно недопустимо. Существует много подходов к решению этой проблемы, однако все больше компаний склоняются в сторону использования Enterprise Service Bus (ESB) – Сервисной Шины Предприятия, передовой на сегодняшний день технологии в области интеграции бизнес-процессов и приложений.

Для начала необходимо разобраться в особенностях автоматизации в сфере банковских бизнеса. С самого начала автоматизация российского банковского сектора происходила сколь бурно, столь и стихийно. Системы, призванные решить конкретные бизнес-задачи, выбирались (и выбираются до сих пор) по результатам тендеров. Таким образом, банк внедряет программное обеспечение, руководствуясь задачами бизнеса и финансовыми возможностями. Но, наряду с процессом специализации ИТ-инфраструктуры банков, происходит ее неконтролируемый рост и обременение обособленным, разнородным программным обеспечением сторонних разработчиков. Это ведет за собой увеличение расходов на содержание и эксплуатацию плохо совместимых друг с другом приложений, а также увеличение времени на обработку бизнес-задач и снижение надежности системы в связи с ростом количества возможных точек отказа.

Тенденции к укрупнению банков за счет слияния, образования банковских и финансово-промышленных групп, заставляет обратиться к решению задачи интеграции приобретенных систем между собой, а также с новыми внедряемыми решениями и бизнес-приложениями. Необходимо обеспечить согласованную работу большого количества разнородных, в том числе самописных, приложений, которые в свое время создавались без учета каких-либо требований, к их возможной интеграции.

В результате естественного хода автоматизации банк получает внушительный набор разнородных информационных систем, которые необходимо объединить в результативно работающую систему. Одним из решений проблемы использования большого количества разнообразных ИТ-решений является внедрение интеграционной платформы. Внедрение унифицированных, гибких и масштабируемых решений превращается в жизненную необходимость. Идеальным решением здесь может стать использование ESB (Enterprise Service Bus) – Сервисной Шины Предприятия.

3 Нагрузочное тестирование интеграционного решения

Решая задачу интеграции важно уделять внимание не только функциональным возможностям, но и нагрузочным показателям, так, чтобы шина при своём полном функциональном покрытии не стала узким местом в плане пропускной способности.

Для того чтобы оценить эффективность внедрения разработанной интеграционной системы, было проведено нагрузочное тестирование интеграционного комплекса и его компонент.

Работы по нагрузочному тестированию комплекса Business Universe RS и его компонент выполнялись в рамках комплексного внутреннего проекта на базе продуктов

компании R-Style SoftLab. Тесты выполнялись на оборудовании, предоставленном компанией-партнером Oracle в лаборатории в Шотландии (Oracle Solution Centre, Blackness Rd, Linlithgow, EH49 7LR, Scotland).

Непосредственно проведению тестов предшествовали работы по подготовке, включающие:

- Внутреннее функциональное тестирование;
- Отладка и оптимизация;
- Подготовка инфраструктуры и программного обеспечения (далее – ПО).

Тестируемая информационная система банка включает в себя следующие компоненты:

- FrontOffice (далее – Фронт или Фронт-офис);
- RS-Bank V.6, подсистема RS-Loans (далее – RS-Bank);
- Oracle Service Bus (далее – OSB, ESB или шина).

Ключевым требованием к моделированию нагрузочных воздействий является первичное наполнение баз данных компонент необходимым объемом данных:

Таблица 1

Объем данных для компонент системы

Система	Объект	Целевой объем
FrontOffice	Клиенты	9000000
FrontOffice	Заявки	13000000
RS-Bank	Клиенты	6000000.
RS-Bank	КД	9000000
RS-Bank	ДО	5670000
ESB	ID клиентов	9000000
ESB	ID договоров	15000000

Целевые показатели

Компонента	Требование	Показатель
FrontOffice	145 000 кредитных заявок за день, что соответствует 100 000 кредитных договоров в день	В среднем 7 250 заявок за час в многопользовательском интеграционном режиме
RS-Bank		В среднем 5 000 договоров за час
ESB		Обработка за час 7 250х11 синхронных запросов и 7 250х10 асинхронных (минимум 5 000 из 7 250 заявок – с созданием договора)
FrontOffice	Поддержка 20 000 одновременных пользовательских сессий	20 000 сессий при среднем отклике не более 10 сек
RS-Bank	Пакетная обработка 10 млн. кредитных договоров за 4 часа (выполнение всех операций сопровождения в БО)	10000000 КД за 4 часа

3.1 Описание тестового стенда

Стенд представляет собой набор серверов, принципиально состоящий из компонент:

- Общий для всех систем сервер базы данных (далее – БД) Oracle – SuperCluster T5-8 Full Rack,
- Сервер для сервер – приложения (далее – СП) Weblogic для FrontOffice – Exalogic X2-2 Half Rack,
- 2х Сервер Sun x4800 для виртуальных машин СП RS-Bank, СП OSB и станций нагружения и мониторинга.

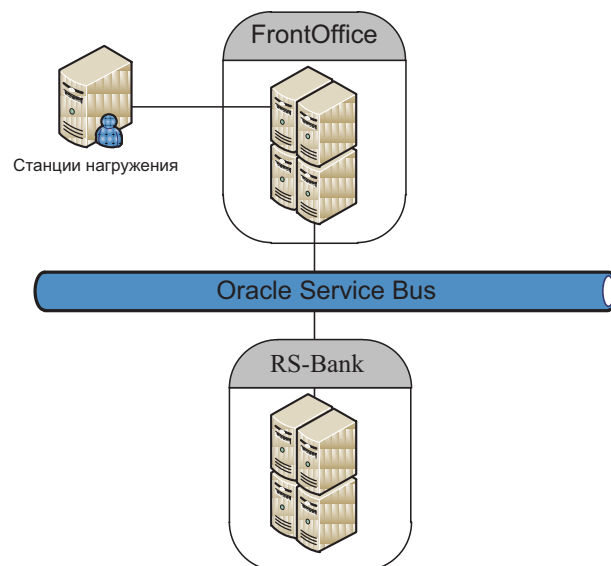


Рис. 1. Общая архитектура стенда

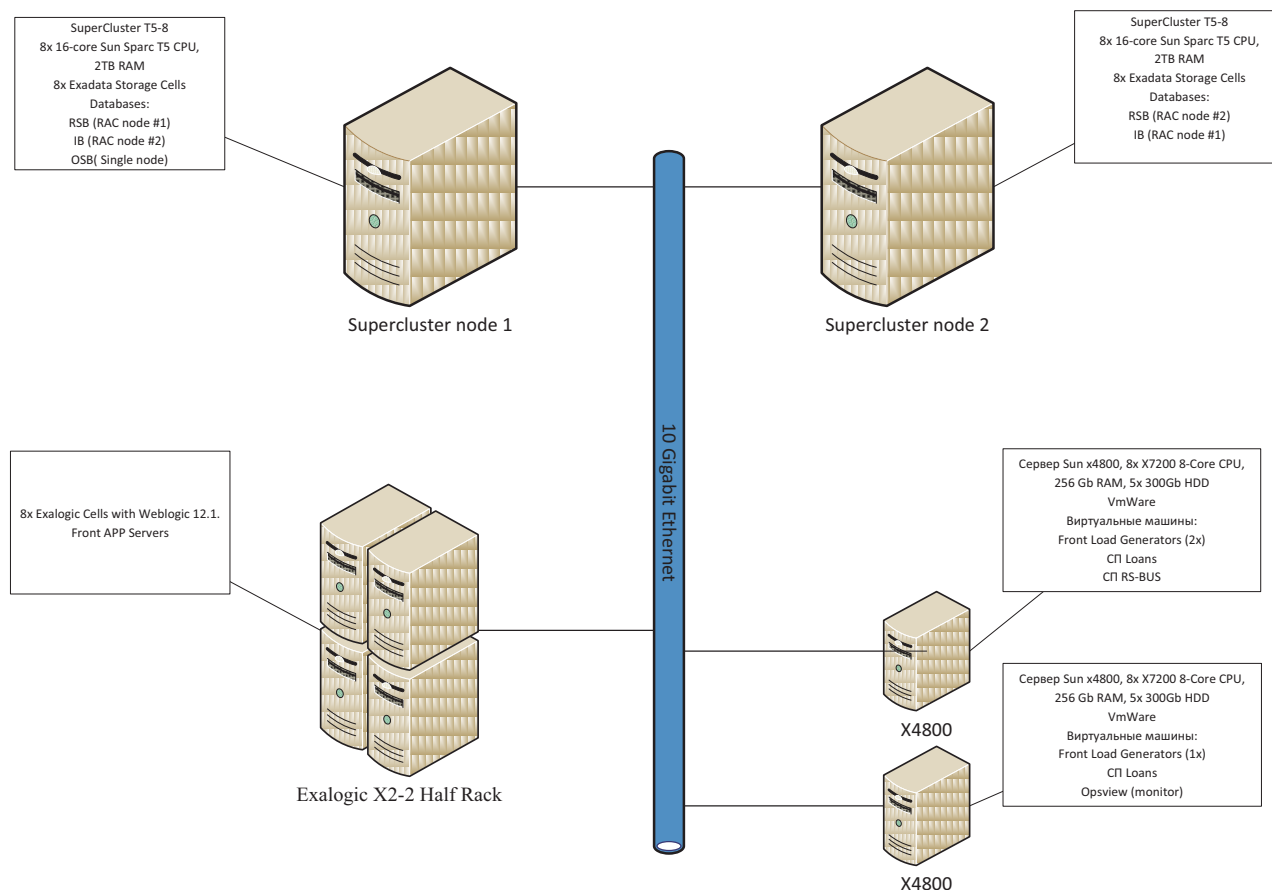


Рис. 2. Топология серверов

3.2 FrontOffice

FrontOffice – это фронт-офисная система обслуживания клиентов комплекса. Назначение FrontOffice – автоматизация деятельности сотрудников фронт-офисных подразделений по выдаче и обслуживанию кредитов. Программная архитектура FrontOffice строится на 4-звенной архитектуре со следующими выделенными компонентами:

- Сервер БД
- Сервер приложения
- Веб-сервер
- В качестве пользовательского терминала используется стандартный Web-браузер (поддерживается Microsoft Internet Explorer и Mozilla Firefox, Opera, Google Chrome).

3.2.1 Архитектура ПАК FrontOffice

Архитектура программно-аппаратного комплекса (далее – ПАК) FrontOffice состоит из выделенного сервера БД, фермы серверов приложения, балансировщика нагрузки, фермы станций нагружения.

3.2.2 Сервер БД FrontOffice

- Сервер БД FrontOffice должен располагаться на Oracle SuperCluster T5-8.
- Операционная система сервера БД – совместимая с Oracle Database 11g Enterprise Edition.
- В качестве системы управления базами данных (далее – СУБД) применяется промышленная СУБД Oracle 11g (11.2.0.3 с установленным обновлением патча не ниже 4 (11.2.0.3.4)).
- Редакция СУБД – Enterprise Edition

3.2.3 Сервер приложения FrontOffice

- Для сервера приложения FrontOffice применяется промышленный сервер приложения Oracle Weblogic Server версии 12.1.1.0.
- Редакция WebLogic Server –Enterprise Edition
- Сервер приложения FrontOffice работает на платформе Oracle Exalogic.
- В качестве операционной системы используется RedHat Enterprise Linux 6.1
- Ферма серверов приложения с Oracle Weblogic Server работает в режиме независимых узлов с кластеризацией на уровне балансировщика нагрузки

➤ Балансировка нагрузки осуществляется путем распределения сессий между серверами приложения с помощью программных или аппаратных средств балансировки нагрузки

3.2.4 Балансировщик нагрузки серверов приложения

➤ Балансировщик нагрузки на серверы приложения может располагаться как на платформе Oracle Exalogic (тогда будет использоваться штатный балансировщик Oracle Traffic Director), так и на выделенном сервере (в этом случае будет использоваться альтернативный балансировщик)

3.2.5 Станция нагрузки FrontOffice

- Специализированное ПО для моделирования пользовательской нагрузки
- Оформлено в виде виртуальной машины VmWare

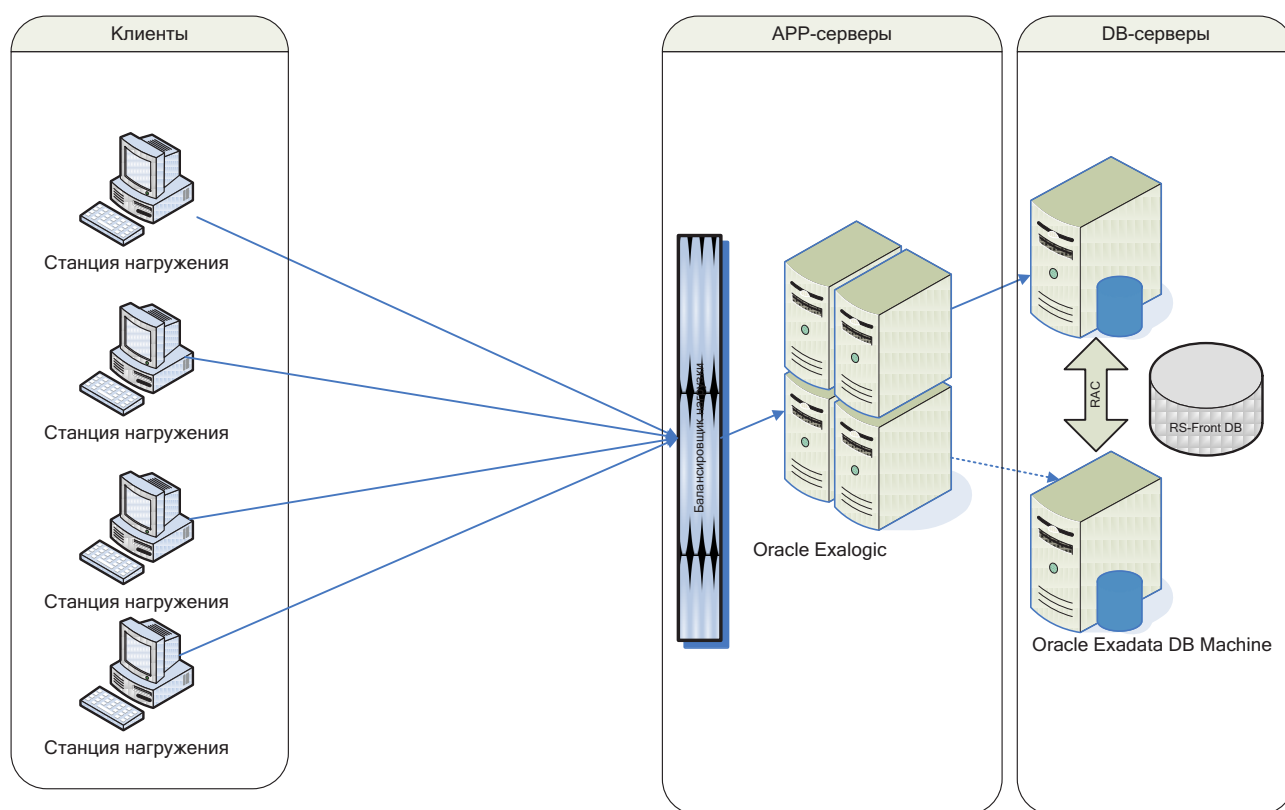


Рис. 3. Принципиальная схема размещения компонент FrontOffice

3.3 RS-bank

RS-Loans (модуль в составе RS-Bank V.6) – это бэк-офисная система обслуживания клиентов комплекса. Назначение RS-Bank – автоматизация деятельности

сотрудников бэк-офисных подразделений Банка по обслуживанию кредитов. Программная архитектура RS-Bank строится на 3-х звенной архитектуре со следующими выделенными компонентами:

- Сервер БД
- Сервер приложения
- Web-сервер транспортной компоненты для интеграции через шину
- В качестве пользовательского терминала используется терминал собственной разработки

3.3.1 Архитектура ПАК RS-Bank

Архитектура ПАК RS-Bank состоит из выделенного сервера БД, сервера приложения с транспортной компонентой.

3.3.2 Сервер БД RS-Bank

- Сервер БД RS-Bank должен располагаться на Oracle SuperCluster T5-8.
- В качестве СУБД применяется промышленная СУБД Oracle 11g (11.2.0.3) с установленным обновлением патча не ниже 4 (11.2.0.3.4)
- Редакция СУБД – Enterprise Edition
- ПАК RS-Bank должен поддерживать работу СУБД Oracle в режиме Real Application Cluster

3.3.3 Сервер приложения RS-Bank

- Для сервера приложения RS-Bank применяется собственный сервер приложения
- Сервер приложения RS-Bank работает на платформе x86 (amd x64).
- В качестве операционной системы используется Windows Server 2008 R2
- Сервер приложения поставляется для тестов в виде виртуальной машины VmWare

3.3.4 Web-сервер транспортной компоненты

- Для Web-сервера транспортной компоненты RS-Bank применяется Microsoft Internet Information Server
- Web-сервер транспортной компоненты RS-Bank работает на сервере приложения RS-Bank
- В качестве операционной системы используется Windows Server 2008 R2

3.3.5 Пользовательский терминал RS-Bank

- Собственная разработка
- Операционная система Windows XP/Windows 7

- Устанавливается на сервер приложения RS-Bank

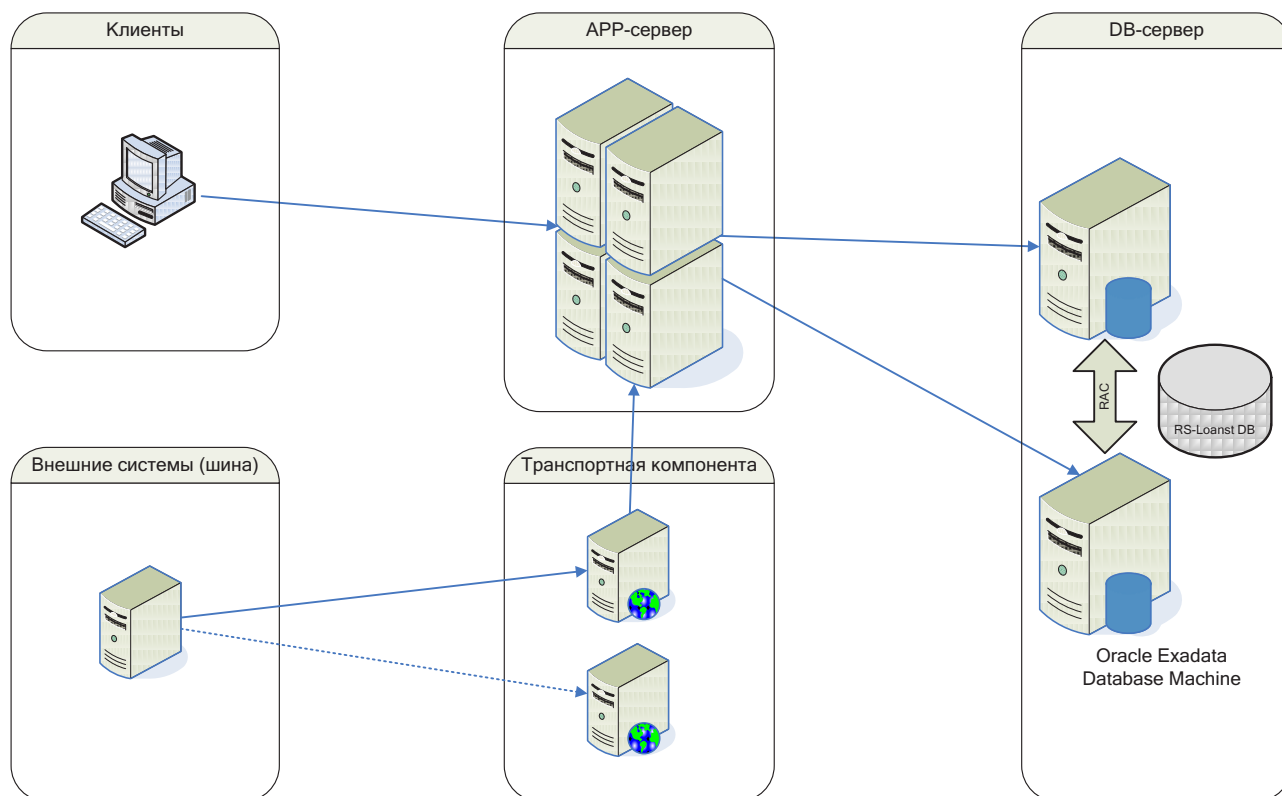


Рис. 4. Принципиальная схема размещения компонент RS-Bank

3.4 ESB

ESB – это интеграционная шина на базе программного продукта Oracle Service Bus, связывающая между собой другие системы, входящие в единый программный комплекс. Программная архитектура ESB состоит из:

- Сервер БД
- Сервер приложений

3.4.1 Архитектура ПАК ESB

Архитектура ПАК ESB состоит из выделенного сервера БД и серверов приложений.

3.4.2 Сервер БД ESB

- Сервер БД ESB должен располагаться на Oracle SuperCluster T5-8.
- В качестве СУБД применяется промышленная СУБД Oracle 11g (11.2.0.3) с установленным обновлением патча не ниже 4 (11.2.0.3.4)

- Редакция СУБД – Enterprise Edition
- ПАК ESB должен поддерживать работу сервера БД в режиме Single Instance

3.4.3 Сервер приложений

- ESB использует промышленный сервер приложения Oracle Weblogic Server с установленным программным продуктом Oracle Service Bus
- Версия ПО Weblogic 12C Enterprise Edition
- Версия ПО Oracle Service Bus 11.1.6
- Сервер приложений ESB работает виртуальной машине под управлением VMWare

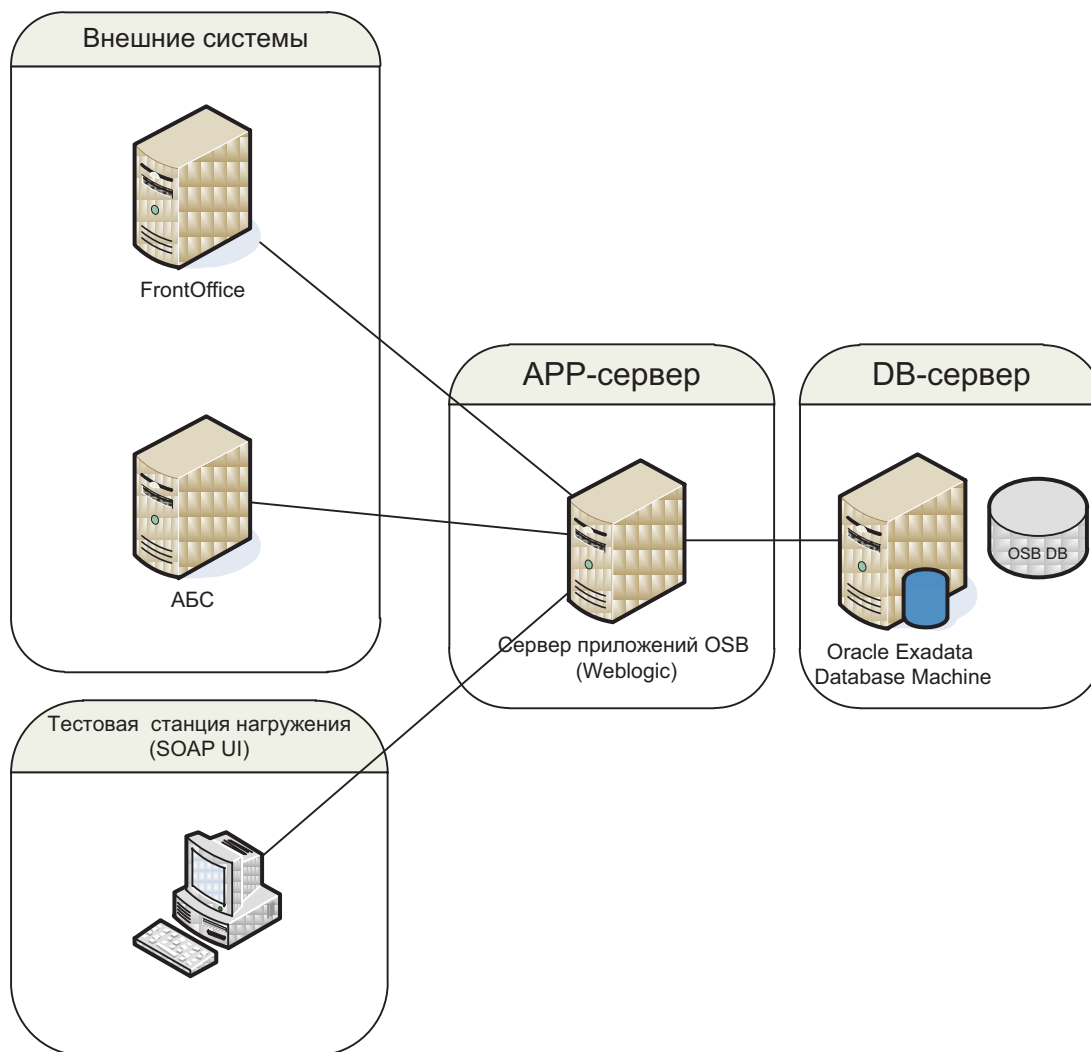


Рис. 5. Принципиальная схема размещения компонент ESB

3.5 Результаты тестирования

Целью данного тестирования являлось получение временных и нагрузочных показателей для операций выдачи кредита в FrontOffice в интеграционном режиме с RS-Bank с использованием шины – Oracle SB (при этом другие операции – сопровождение и групповые процедуры в RS-Bank – не выполнялись).

Все события запускаются из системы FrontOffice, остальные системы фиксируют показатели производительности в пассивном режиме.

Проведена серия запусков тестовых сценариев (см. таблицу 3) и получены следующие результаты - среднее время на выполнения цикла обработки заявки/договора (для FrontOffice – это общее время работы теста, поделенное на количество созданных кредитных заявок, при условии соблюдения ограничения среднего времени отклика).

Таблица 3

Сценарии, результаты и показатели

№	Сценарий	Время теста	Компонента	Показатель №1 ¹ «Кол-во объектов»	Показатель №2 ² «Среднее время»
1.	Скрипт эмуляции работы пользователей во FrontOffice состоит из 40-минутного цикла на ввод одной кредитной заявки (интенсивность соответствует 5000 КД в час) – 5000 пользовательских сессий	60 минут	FrontOffice	5 000	1 143 мс
			Oracle SB	122 893	619 мс / 21%
			RS-Bank	5 000	657 мс
2.	Скрипт эмуляции работы пользователей во	30	FrontOffice	6 661	900 мс

¹ Показатель 1:

Для FrontOffice – количество кредитных заявок, созданных и выгруженных в шину за час (при условии соблюдения ограничения среднего времени отклика)

Для Oracle SB – кол-во обработанных запросов за час

Для RS-Bank – количество созданных КД за час

² Показатель 2: среднее время выполнения всех запросов

Для FrontOffice – среднее время отклика системы (только прикладной функционал + интеграционный)

Для OSB – среднее время, включающее работу RS-Bank / процент чистого времени на шину

Для RS-Bank – среднее время работы сервисов

	FrontOffice запущен в стрессовом режиме (т.е. без временных задержек в сценарии) – 500 пользовательских сессий	минут	Oracle SB	104 354	1 356 мс / 36%
			RS-Bank	6660	1 036 мс

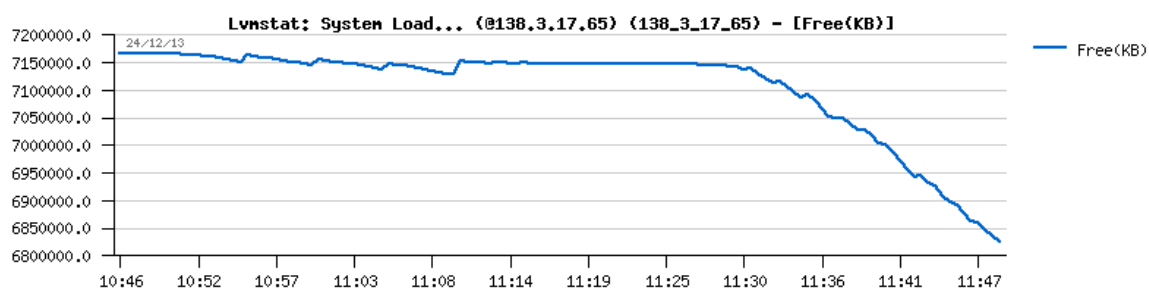


Рис. 6. Загрузка памяти для показателя №1

Первый сценарий повторяет работу среднестатистического пользователя Фронт-офиса банка, который вводит заявку за 40 минут. В начале часа идет интенсивный вызов сервиса поиска клиента, в середине – идет массовая генерация договоров, в остальное время – нагрузка минимальна.

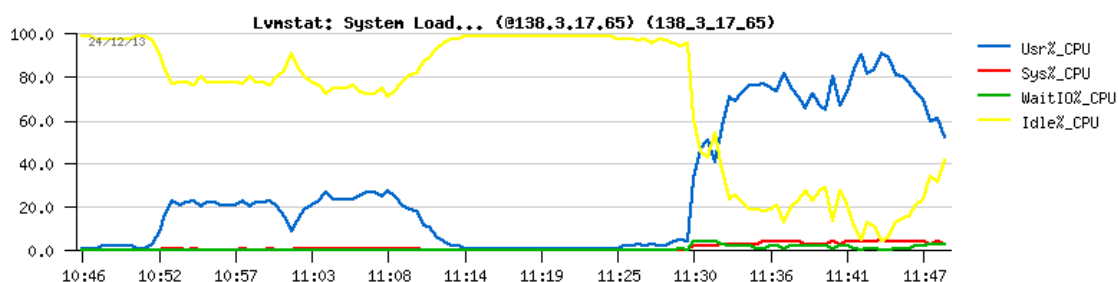


Рис. 7. Загрузка ЦП для показателя №1

Более равномерную нагрузку позволяет создать второй из двух тестов, который предусматривает ввод заявки за 2 минуты, что практически мгновенно приводит к случайному порядку вызова сервисов шины и соответственно вызывает равномерную нагрузку.

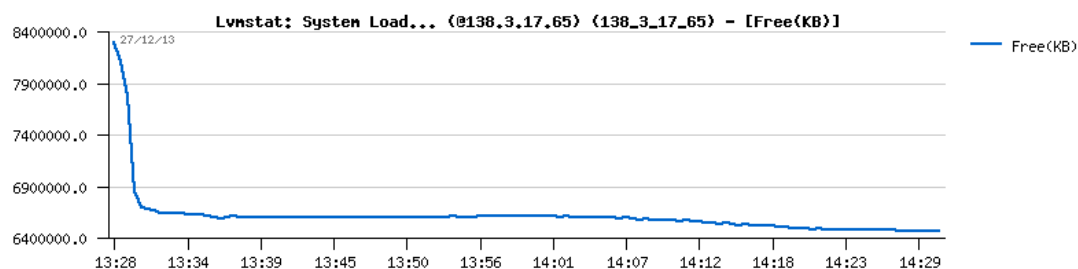


Рис. 8. Загрузка памяти для показателя №2

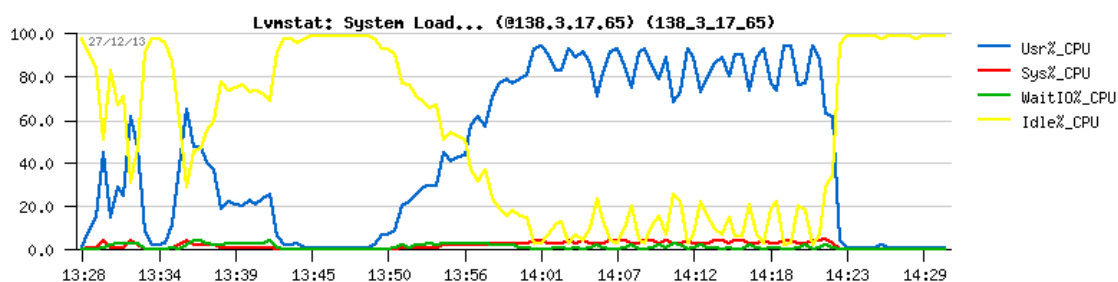


Рис. 9. Загрузка ЦП для показателя №2

Тест №1, 5000 заявок в час, когда заявка создается в течение 40 минут, вызвал существенную нагрузку на стороне шины только в завершающей стадии, когда после длительного промежутка бездействия от FrontOffice начинают приходить пакеты документов на создание договоров. Нагрузка ниже целевых показателей на 40%.

Тест №2, 500 пользователей в "стрессе", когда цикл создания заявки на стороне FrontOffice значительно сокращен и, таким образом, нагрузка на шину происходит более равномерно и меньшим числом одновременных пользователей. Данный тест превышает целевые показатели на 40%. По количеству заявок тест превышал показатели теста №1 почти в 2 раза. Тем не менее, загрузка процессора на стороне шины в активной фазе теста выросла только на 10%, но тем не менее находилась уже на пределе возможного и дальнейший рост нагрузки проводить нецелесообразно.

4. Анализ результатов и выводы

Через шину прошло более 200 000 запросов за час, при этом выполнение асинхронных запросов – с использованием очереди сообщений – не приводило к

накоплению очереди. Удалось загрузить ресурсы шины на 60-80%, т.е. существенное увеличение показателей возможно лишь при увеличении ресурсов.

Таким образом, можно заключить, что интеграционное решение было спроектировано правильно не только с точки зрения функциональности системы, но и с точки зрения нагрузочных показателей. Оно не только удовлетворяет текущим требованиям, но и позволяет увеличить нагрузку в будущем до 20% без увеличения ресурсов, и, при этом, не станет узким местом в системе.

Список литературы

1. Брешенков А.В. Базы данных. Проектирование баз данных на основе информации табличного вида. Saarbrücken: LAP LAMBERT Academic Publishing, 2011. 394 с.
2. Коптелов А. Системы управления процессами и SOA // Компьютер Пресс. 2008. № 8. С. 141-143.
3. Когаловский М.Р. Методы интеграции данных в информационных системах // Сб. трудов Третьей Всероссийской конференции "Стандарты в проектах современных информационных систем". Москва, 23-24 апреля 2003. С. 114-118.
4. Дубова Н. Краткий курс интеграции данных // Открытые системы. 2007. №9. С. 46-51.
5. Черняк Л. Интеграция данных: синтаксис и семантика // Открытые системы. 2009. №10. С. 15-23.
6. Thomas E. Service Oriented Architecture: Concepts, Technology and Design. Indiana: Pearson Education, 2003. 171 p.
7. Ключкин В. От традиционных способов интеграции – к SOA // Банковские технологии. 2008. №9. С. 28-30.
8. Шаппелл Д. Сервисная шина предприятия. Банковские технологии. СПб.: БХВ – Петербург, 2008. 368 с.