

RatheDG Cachalot System 1.2.0

Руководство пользователя

© 2001-2003 Rathe Development Group. Все права защищены.
Использование является предметом лицензионного соглашения.

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ.....	2
1. ОБЩЕЕ ОПИСАНИЕ.....	2
1.1. НАЗНАЧЕНИЕ СИСТЕМЫ.....	2
1.2. ОПРЕДЕЛЕНИЯ.....	2
1.3. ПЕРЕЧЕНЬ ВОЗМОЖНОСТЕЙ.....	3
2. АРХИТЕКТУРА СИСТЕМЫ.....	4
2.1. СТРУКТУРА СИСТЕМЫ.....	4
2.2. МОДУЛЬ РЕГИСТРАЦИИ СОБЫТИЙ.....	4
2.3. МОДУЛЬ РЕГИСТРАЦИИ ТРАФИКА.....	4
2.4. МОДУЛЬ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ.....	5
3. ТРЕБОВАНИЯ.....	5
4. РЕАЛИЗАЦИИ.....	5
4.1. МОДУЛЬ РЕГИСТРАЦИИ СОБЫТИЙ.....	5
4.1.С. Реализация МРС/С.....	5
4.1.К. Реализации МРС/К.....	5
4.1.К.1. Java клиент командной строки.....	5
4.1.К.2. Платформо-зависимый клиент командной строки.....	5
4.1.К.3. Windows-интегрированный клиент.....	5
4.2. МОДУЛЬ РЕГИСТРАЦИИ ТРАФИКА.....	6
4.3. МОДУЛЬ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ.....	6
4.4. СВОДКА ПЛАТФОРМ.....	6
5. РУКОВОДСТВО ПО ИНСТАЛЛЯЦИИ.....	7
5.1. СТРУКТУРА И СОДЕРЖИМОЕ ПОСТАВКИ СИСТЕМЫ.....	7
5.2. ПОСЛЕДОВАТЕЛЬНОСТЬ УСТАНОВКИ.....	7
5.3. БАЗА ДАННЫХ.....	8
5.4. МОДУЛЬ РЕГИСТРАЦИИ СОБЫТИЙ.....	9
5.4.С. Инсталляция МРС/С.....	9
5.4.К. Инсталляция МРС/К.....	11
5.4.К.1. Клиенты командной строки.....	11
5.4.К.2. Windows-интегрированный клиент.....	11
5.5. МОДУЛЬ РЕГИСТРАЦИИ ТРАФИКА.....	12
5.6. МОДУЛЬ ИНТЕРФЕЙСА ПОЛЬЗОВАТЕЛЯ.....	12
5.6.1. Задание конфигурации через конфигурационный файл.....	13
5.6.2. Задание конфигурации через JNDI.....	13
5.6.3. Параметры конфигурации.....	13
5.6.4. Установка в контейнер.....	15
6. БЕЗОПАСНОСТЬ.....	16
ПРИЛОЖЕНИЯ.....	16
Приложение А. КОНФИГУРАЦИЯ ЛОГА.....	16
Приложение Б. КОНФИГУРАЦИЯ СОЕДИНЕНИЯ С БАЗОЙ ДАННЫХ.....	16

Лицензионное соглашение.

ЛИЦЕНЗИОННОЕ СОГЛАШЕНИЕ ДЛЯ RATHEDG CACHALOT SYSTEM

Это лицензионное соглашение ("Соглашение") является юридическим соглашением между вами, пользователем RatheDG Cachalot System ("Система"), и Rathe Development Group ("RatheDG"). Пользователем Системы является физическое или юридическое лицо, имеющее копию Системы или любой из ее частей на жестком диске (или любом другом носителе) или использующее работающий экземпляр Системы или любой из ее частей.

Если вы не согласны с данным Соглашением, вы должны прекратить использовать систему и удалить все копии Системы. Любое использование, неразрешенное Соглашением, означает немедленное и автоматическое прекращение действия Соглашения и может преследоваться по закону.

Вы не можете декомпилировать, производить инженерный анализ, дизассемблировать или как-либо иначе изучать код Системы. Вы не можете модифицировать Систему (включая файлы документации или любые другие файлы, поставляемые с системой, за исключением конфигурационных и командных файлов). Вы не можете сдавать в наем или аренду или продавать Систему. Вы не можете распространять Систему ни в каком виде, кроме как в виде исходного немодифицированного дистрибутива.

СИСТЕМА ПОСТАВЛЯЕТСЯ "КАК ЕСТЬ". НИКАКИХ ГАРАНТИЙ НЕ ПРИЛАГАЕТСЯ И НЕ ПРЕДУСМАТРИВАЕТСЯ. ВЫ ИСПОЛЬЗУЕТЕ ЕЕ НА СВОЙ РИСК. RATHEDG НЕ БУДЕТ ОТВЕЧАТЬ НИ ЗА КАКИЕ ПОТЕРИ ИЛИ ИСКАЖЕНИЯ ДАННЫХ, ЛЮБУЮ УПУЩЕННУЮ ВЫГОДУ ИЛИ ЛЮБОЙ ДРУГОЙ ВИД ПОТЕРИ В ПРОЦЕССЕ ИСПОЛЬЗОВАНИЯ, НЕПРАВИЛЬНОГО ИСПОЛЬЗОВАНИЯ ИЛИ НЕВОЗМОЖНОСТИ ИСПОЛЬЗОВАНИЯ СИСТЕМЫ.

RatheDG оставляет за собой право изменять Соглашение без какого-либо уведомления вас.

Все права, не предоставленные здесь явно, сохранены за RatheDG.

1. Общее описание.

1.1. Назначение системы.

Существуют ситуации, когда организациям требуется собирать статистические данные об использовании сети и/или компьютеров, подключенных к другой сети, и наблюдать изменения в реальном времени. Типичные примеры таких организаций – так называемые Интернет кафе и провайдеры Интернета. Статистические данные включают информацию о сессиях пользователей, оплате, сетевом трафике и т.п.

Cachalot System предоставляет возможность собирать статистические данные и иметь удобный доступ к этим данным и предоставляет возможность наблюдения за событиями по сети. Система не ограничена конкретной сетью, она использует чистый TCP/IP и HTTP протоколы и может быть использована и для Интернета, и для локальной сети.

Система требует ручной установки и поддержки системным администратором.

Java FAQ – <http://www.ibiblio.org/javafaq/javafaq.html>.

1.2. Определения.

В этой главе даются описания сущностей и понятий, вовлеченных в систему.

Хост – компьютер, подключенный к сети. Хост может включаться и выключаться.

Внутренний хост – хост, зарегистрированный в системе.

Внешний хост – хост, не зарегистрированный в системе.

Пользователь – лицо, использующее внутренний хост. Пользователь может входить на хост и выходить с хоста. В каждый момент времени только один пользователь может быть вошедшим на конкретный внутренний хост.

Событие – произошедшее на конкретном внутреннем хосте некоторое действие, опционально связанное с конкретным пользователем. Используются следующие события: включение и выключение хоста, вход и выход пользователя.

Группа – множество пользователей. Пользователь принадлежит одной группе. Каждая группа имеет роль, назначенную администратором. В системе используются три роли:

- Администратор – может управлять пользователями, группами и хостами, Администратор также является Обозревателем и Пользователем.
- Обозреватель – может наблюдать за событиями и смотреть статистику, Обозреватель также является Пользователем.
- Пользователь – не может делать ничего в смысле системы, кроме как быть относящимся к событиям.

Трафик это данные, передаваемые через сеть между внутренним и внешним хостом. Обычно это входящие/исходящие данные в/из Интернета/локальной сети, передаваемые через роутер (может быть использована любая другая конфигурация, включающая роутер).

Два последовательных события вход и выход пользователя на хосте определяют **сессию**. Включение, выключение и вход трактуются как выход, если выход не был предыдущим событием. Весь трафик, переданный с/на этот хост в течение сессии, ассоциируется с этой сессией.

Часто есть необходимость считать стоимость сессии пользователя, исходя из продолжительности сессии и ее трафика. Предполагается, что стоимость C определяется

$$C = C_h * P_h * D + C_i * P_i * T_i + C_o * P_o * T_o$$

где C_h , C_i и C_o – коэффициенты, определенные для пользователя или группы; P_h , P_i и P_o – цены часа сессии, мегабайта входящего трафика и мегабайта исходящего трафика; D , T_i и T_o – продолжительность сессии, размер общего входящего трафика и общего исходящего трафика сессии. Т.о. величина C может быть посчитана для каждой сессии и термин ‘посчитанная стоимость сессии’ означает C , вычисленную для конкретной сессии.

Операция – некоторая абстрактная услуга, которая может быть измерена целым числом единиц (экземпляров). У операции есть назначенная цена – цена одной единицы. Примеры операций: распечатка страницы, запись CD, приготовление чашки кофе и т.п. (Функциональность ‘операции’ реализована как модуль расширения и является необязательной).

Внутренние хосты, пользователи, группы и операции требуют ручной регистрации в системе администратором.

1.3. Перечень возможностей.

Ниже представлен перечень возможностей системы. Уделите внимание будущим планам, если вы сомневаетесь, отвечает ли система вашим требованиям. Возможности перечислены кратко, посетите онлайн-демо для большего понимания: <http://www.rathedg.com/products/cachalot/demo/>.

- Регистрация сессий пользователей.
- Сбор информации о трафике, проходящем через роутер.
- Интерфейс для Администратора к управлению пользователями, группами, хостами, операциями и настройками системы.
- Интерфейс для Обозревателя к наблюдению активных сессий.
- Интерфейс для Обозревателя к регистрации оплаты за окончившуюся сессию.
- Интерфейс для Обозревателя к регистрации операции.
- Интерфейс для Обозревателя к просмотру статистики. Чтобы представить, что представляет собой отчет, смотрите суммарный список столбцов отчета по сессиям (т.е. базирующегося на информации о сессиях):
 - имя пользователя
 - группа, к которой принадлежит пользователь
 - хост, на который зашел пользователь
 - время входа/выхода
 - продолжительность сессии
 - размер входящего/исходящего трафика для всей сессии/по внешним хостам/по внешним портам
 - вычисленная стоимость
 - оплаченная стоимость

Список столбцов отчета по трафику:

- внутренний хост
- внутренний порт
- внешний хост
- внешний порт
- размер входящего/исходящего/суммарного трафика

Список столбцов отчета по операциям:

- операция
- кассир (Обозреватель, зарегистрировавший операцию)
- дата
- количество единиц
- сумма оплаты

Интерфейс отчета гибок и позволяет получить различные варианты статистики. Присутствует возможность включать/выключать столбцы и указывать конкретные хосты, пользователей и т.д., для которых генерировать отчет. Есть возможность создавать предустановки для часто используемых отчетов.

- Интерфейс для Администратора к удалению зарегистрированных данных из базы данных.

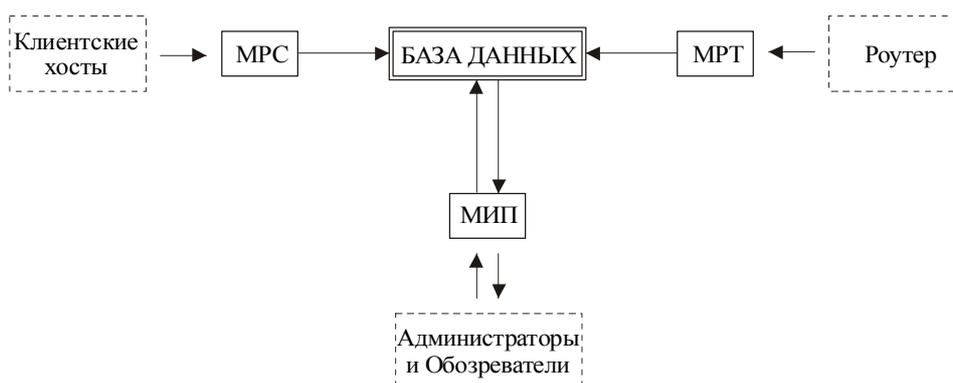
2. Архитектура системы.

2.1. Структура системы.

Система состоит из трех модулей: модуль регистрации событий (МРС), модуль регистрации трафика (МРТ) и модуль интерфейса пользователя (МИП). Назначение каждого модуля:

- МРС – регистрирует события (включение, вход, выход, выключение) в базе данных
- МРТ – регистрирует трафик в базе данных
- МИП – интерфейс пользователя для управления, наблюдения и генерации статистических отчетов.

МРС и МИП должны работать на одном компьютере. МРТ может работать как на другом компьютере, так и на том же.



Присутствует возможность устанавливать модули расширения в МИП. В любой момент индивидуальный модуль расширения может быть разработан для удовлетворения ваших специфических нужд в интерфейсе пользователя.

2.2. Модуль регистрации событий.

МРС выполнен в архитектуре клиент-сервер. Его серверная часть (МРС/С) работает на любом компьютере, который доступен по ТСР/ІР с клиентских компьютеров. Серверная часть должна иметь доступ к базе данных. Клиентская часть (МРС/К) работает на каждом компьютере и должна иметь возможность связи с МРС/С по ТСР/ІР. Все коммуникации между МРС/С и МРС/К осуществляются по ТСР/ІР через один ТСР/ІР порт.

Перед тем как событие произойдет, МРС/К связывается с МРС/С, уведомляет, что событие сейчас произойдет, и предоставляет всю информацию, касающуюся события. МРС/С регистрирует событие в базе данных.

2.3. Модуль регистрации трафика.

МРТ отслеживает информацию о трафике (ІР пакетах) и вносит ее в базу данных. МРТ отслеживает информацию только о трафике между внутренними и внешними хостами, т.е. оставляет без внимания трафик внутренний-внутренний и внешний-внешний.

МРТ должен иметь доступ к базе данных.

2.4. Модуль интерфейса пользователя.

МИП предоставляет интерфейс пользователя для Администраторов и Обозревателей: генерация статистики, наблюдение, управление системой. Это веб-интерфейс, т.е. работает веб-сервер и пользователь использует веб-браузер для доступа к интерфейсу. МИП должен иметь доступ к базе данных.

МИП имеет поддержку различных языков. В данный момент имеются английский и русский переводы, и в любой момент могут быть добавлены переводы на другие языки (это может потребовать дополнительную помощь заказчика, заинтересованного в конкретном языке).

3. Требования.

Эта глава не рассматривает требования к аппаратному обеспечению. Система не предъявляет никаких особых требований к аппаратному обеспечению. Эта глава описывает требования к программному обеспечению, которое необходимо системе, и окружению, в котором работает система.

Очень важной частью системы является база данных. Система не ограничена использованием с конкретной базой данных, и любая база данных, отвечающая требованиям системы, может быть использована. Она должна иметь SQL интерфейс (необходим SQL/92 entry level); соответствующий JDBC или ODBC драйвер должен быть доступен.

Требования к окружению каждого модуля:

- MPC
 - MPC/К – возможность связываться с MPC/С через один TCP/IP порт
 - MPC/С – возможность слушать один TCP/IP порт, доступ к базе данных
- МРТ – необходимые для прослушивания трафика привилегии, доступ к базе данных
- МИП – JSP контейнер, доступ к базе данных.

4. Реализации.

4.1. Модуль регистрации событий.

Т.к. MPC имеет архитектуру клиент-сервер, клиент и сервер могут иметь независимые варианты реализаций.

4.1.С. Реализация MPC/С.

MPC/С реализован на Java и требует JRE 1.3.0 или совместимое. Чтобы узнать требования к аппаратному обеспечению, смотрите документацию JRE для вашей платформы.

4.1.К. Реализации MPC/К.

Клиенты, реализованные как приложения командной строки, могут быть запущены из сценариев (например, сценарий, выполняемый при включении хоста, и сценарий входа пользователя).

4.1.К.1. Java клиент командной строки.

Java клиент командной строки MPC/К требует JRE 1.3.0 или совместимое.

Использование: *Client хост порт пароль <таймаут мс> <число попыток> действие [имя пользователя]*, где *действие* – одно из *startup, shutdown, logon, logoff*. *Имя пользователя* указывается только для действия *logon*. *Хост* и *порт* должны определять работающий MPC/С. *Пароль* – пароль, который должен предоставить клиент для аутентификации, этот пароль – часть конфигурации MPC/С.

4.1.К.2. Платформено-зависимый клиент командной строки.

Синтаксис использования платформено-зависимого клиента командной строки MPC/К идентичен синтаксису использования Java клиента командной строки.

Платформено-зависимый клиент командной строки, поставляемый с этой версией Cachalot System, скомпилирован под x86 MS Windows 32bit; он может быть в любой момент откомпилирован для другой платформы, возможно за дополнительную плату.

4.1.К.3. Windows-интегрированный клиент.

Windows-интегрированный MPC/К клиент реализован как Windows DLL, которая работает на месте одной из системных Windows DLL. Он может быть использован в MS Windows NT4.0 или совместимой ОС (в данный момент Windows 2000, Windows XP).

Преимущество интеграции в том, что не нужно искать место, где произвести вызов клиента командной строки. DLL осуществляет отслеживание и отсылание события автоматически, написание дополнительных сценариев или разработка не требуются. Windows-интегрированный MPC/К может быть сконфигурирован записывать события и результаты их отсылания в локальный лог-файл.

4.2. Модуль регистрации трафика.

MPT реализован как приложение на Java. MPT основан на библиотеке JРсар, которая является Java-интерфейсом к известным WinPсар/LibPсар библиотекам. Фактически, регистрация трафика осуществляется нижележащими библиотеками. MPT обрабатывает информацию о трафике, предоставляемую библиотекой, суммирует (если сконфигурирован) и вносит в базу данных. Ниже приведен алгоритм суммирования, за счет которого достигается значительное уменьшения объема базы данных и увеличение быстродействия системы.

Трафик между внутренними хостами отбрасывается, за исключением случаев, когда номер порта на одном из хостов (1) входит в список портов, которые необходимо учитывать для этого хоста (обозначим этот список D). Этот список задается в МИП для каждого хоста. В таком случае тот внутренний хост (1), чей порт необходимо учитывать, условно считается внешним хостом (и таковым фигурирует в статистике). Типичными примерами таких особых портов являются: HTTP сервер на порту 8080, HTTP прокси, туннели.

Если в течение интервала накопления трафика (см. конфигурацию MPT), зарегистрированы пакеты, имеющие одинаковые параметры (протокол, исходные и конечные хосты, порты), то они считаются одним пакетом, длина которого равна сумме длин соответствующих реальных пактов.

Для лучшего суммирования, каждый пакет (TCP и UDP) перед суммированием подвергается сбрасыванию номеров портов в 65535 по специально подобранным правилам. Общий смысл этого сбрасывания – не хранить неинформативных номеров портов; это порты хоста-клиента, открывшего соединение. Как правило, это порты, большие 1024. Точное правило сбрасывания номеров портов представлено таблицей:

	внешний хост	порт<=1024	порт>1024	порт из Dext
внутренний хост				
порт<=1024		внутренний	внешний	-
порт>1024		внутренний	-	внутренний
порт из Dint		-	внешний	внутренний

В двух случаях (из возможных семи) сбрасывается номер порта внутреннего хоста (соединение, установленное с внутреннего хоста на внешний), в двух – внешнего хоста (соединение, установленное с внешнего на внутренний хост), в остальных – номера портов не сбрасываются, т.к. нельзя их заведомо считать неинформативными. Dint – порты внутреннего хоста, большие 1024, указанные учитываемыми, Dext – порты условно внешнего хоста (для случая трафика между двумя внутренними хостами), указанные учитываемыми.

Сбрасывание номеров портов может быть полностью отключено для отдельных хостов через МИП.

Необходимо отметить, что при нехватке мощности компьютера, на котором работает MPT, для регистрации и обработки всех пакетов, будет происходить потеря информации о пакетах, в этом случае число пропущенных пакетов будет писаться в лог.

4.3. Модуль интерфейса пользователя.

МИП реализован как веб-приложение, использующее технологию JSP. МИП требует JSP контейнер (JSP 1.1 и Java Servlet 2.3 совместимый).

Большая часть интерфейса прозрачна и проста для использования, и не требует специального руководства. Руководство по интерфейсу пользователя, поставляемое с документацией, содержит замечания для нескольких наиболее непростых частей.

ВНИМАНИЕ: модули MPT и MPC должны быть перезапущены после того, как были изменены цены или сущности (созданы, изменены или удалены хост, пользователь и т.п.).

4.4. Сводка платформ.

MPC/С и МИП платформо-независимы; они требуют наличия JRE 1.3.0 или совместимого.

MPC/К требует одно из следующего:

- JRE 1.3.0 или совместимое (платформо-независимо)

- x86 MS Windows 32bit (платформо-зависимый клиент командной строки может быть поставлен скомпилированным для другой платформы по дополнительному запросу).

MPT платформо-независим; он требует наличия JRE 1.3.0 или совместимого. Т.к. MPT использует JPcap он также требует наличия одного из следующего:

- x86/Alpha MS Windows 32bit с установленными WinPcap+JPcap
- Любой Unix с установленными LibPcap+JPcap.

5. Руководство по инсталляции.

5.1. Структура и содержимое поставки системы.

Cachalot System поставляется как архив, содержащий следующие папки:

- bin – командные скрипты для запуска/остановки модулей системы
 - вспомогательные: environment.cmd, environment.sh
 - инсталляционные: cachalot.cmd, cachalot.sh
 - MPC: eru_client.cmd, eru_client.sh, eru_jclient.cmd, eru_jclient.sh, eru_start.cmd, eru_start.sh, eru_stop.cmd, eru_stop.sh
 - MPT: tru_listdevices.cmd, tru_listdevices.sh, tru_start.cmd, tru_start.sh, tru_stop.cmd, tru_stop.sh
 - МИП: uiu_start.cmd, uiu_start.sh, uiu_stop.cmd, uiu_stop.sh.
- conf – конфигурационные файлы:
 - erus.conf
 - rdgcswic.conf
 - tru.conf
 - uiu.conf.
- doc – документация:
 - license.en.html, license.ru.html
 - manual.en.pdf, manual.ru.pdf
 - install.linux.en.html, install.linux.ru.html, install.winnt.en.html, install.winnt.ru.html
 - faq.en.html, faq.ru.html
 - uinotes.en.html, uinotes.ru.html
 - changes.ru.html, changes.en.html
 - upgrade.en.html, upgrade.ru.html.
- system
 - client – исполняемый код клиентской части системы
 - database – скрипты базы данных:
 - create_general.tmp.sql, create_operations.tmp.sql
 - create_general.mssql65-8.sql, create_operations.mssql65-8.sql
 - create_general.pgsql, create_operations.pgsql.sql
 - drop_general.sql, drop_operations.sql
 - server – исполняемый код серверной части системы
 - install.cmd, install.sh.

Юридическое замечание. jpcap.dll, libjpcap.so и net.sourceforge-jpcap-0.1.13.jar поставляются на правах лицензии Mozilla Public License версии 1.1. Исходный код поставляемого исполняемого кода доступен на <http://jpcap.sourceforge.net/> или по ссылке <http://www.rathedg.com/products/cachalot/download/net.sourceforge.jpcap-0.01.13.tar.gz>

Этот продукт включает программное обеспечение, разработанное Apache Software Foundation <http://www.apache.org/>.

5.2. Последовательность установки.

Оптимальная последовательность установки и конфигурации:

- создать базу данных (см. главу «База данных»)
- сконфигурировать и запустить МИП (см. главу «Модуль интерфейса пользователя»)
- зарегистрировать через МИП хосты, пользователей, установить цены
- сконфигурировать и запустить MPC/C (см. главу «Модуль регистрации событий»)
- установить на клиентские хосты клиента MPC (см. главу «Модуль регистрации событий»), сконфигурировать
- сконфигурировать и запустить MPT (см. главу «Модуль регистрации трафика»).

5.3. База данных.

Для того, чтобы установить базу данных, используйте файлы create*.sql, расположенные в папке db поставки системы. Главные файлы – create*.tmpl.sql, они являются шаблонами, содержащими метаобозначения вместо названий типов данных. Система была протестирована с MSSQLServer 6.5, MSSQLServer 2000 и PostgreSQL 7.2.3, и поставка включает сценарии sql¹ для этих баз данных: create_*.mssql65-8.sql² и create_*.pgsql.sql. Если вы желаете использовать систему с другой базой данных X, вам нужно создать ваши собственные сценарии create_*.x.sql заменой в файлах create_*.tmpl.sql метаобозначений значениями соответственно типам данных.

метаобозначение	значение
%VARCHAR%	строка переменной длины
%BYTE%	целое число, 2 ⁸ значений
%SHORT%	целое число, 2 ¹⁶ значений
%FLOAT%	вещественное число, может быть использован любой тип
%DATETIME %	дата и время
%INTEGER%	целое число, 2 ³² значений
%CHAR%	символ (один байт)
%BIT%	булевый тип или бит (или любое целое число, если база не имеет ни булевого типа, ни бита).
%ALLOWNULL%	строка (часть языка DDL), которая сообщает базе, что поле может содержать NULL значения (например, строка 'NULL' для MSSQL и PostgreSQL)
%TRUE%	константа 'истина' (обычно 'true' или '1', в зависимости от типа %BIT%)
%FALSE%	константа 'ложь' (обычно 'false' или '0', в зависимости от типа %BIT%)

Вы можете выбрать типы данных, диапазон которых или внутреннее представление значений в базе данных отличается от требуемого³ (действуя на свой страх и риск), но учтите следующее:

- выбирая больший диапазон, вы не получите больших значений, т.к. это не изменит того, как система обрабатывает эти значения
- вы можете выбрать меньшие диапазоны (например, %BYTE% вместо %SHORT%) для уменьшения размера базы данных, но учтите, что некоторые ODBC и JDBC драйвера могут не смочь преобразовать типы данных, если они отличаются от запрошенных приложением (это может быть как осознанное поведение драйвера, так и ошибка в драйвере), и это вызовет ошибку (например, когда приложение пытается получить %SHORT% из результата выборки, в то время как результат выборки содержит %BYTE%).

Для того, чтобы установить базу данных, вам нужно выполнить последовательно два сценария: create_general.x.sql, create_operations.x.sql. Не исполняя второй из них, вы не установите модуль расширения 'Операции'. Поставка также содержит drop*.sql сценарии, они могут быть полезны для удаления всех данных и таблиц системы (порядок исполнения обратный созданию).

Для того, чтобы ваша база данных была доступна системе, вам нужно иметь соответствующий JDBC драйвер, знать его название и синтаксис его JDBC url. Ниже приведен список некоторых драйверов:

¹ Здесь термин сценарий используется в смысле последовательности SQL запросов.

² Заметьте, что MSSQL Query Analyser не может быть использован для исполнения файла сценария полностью из-за его ограничения на выполнение CREATE VIEW.

³ Если вы делаете эти изменения, вы действуете на свой страх и риск, и RatheDG оставляет за собой право решать, предоставлять ли техническую поддержку в таких случаях.

база данных	драйвер	url
MSSQLServer 7 MSSQLServer 2000	MS JDBC Driver for MSSQL (успешно протестирован с MSSQLServer 2000)	http://www.microsoft.com/sql/downloads/2000/jdbc.asp
MSSQLServer 6-7 Oracle Informix4	BEA jDriver 5.1.0 (успешно протестирован с MSSQLServer 6.5)	http://commerce.beasys.com/downloads/weblogic_server.jsp#wlsjdbc http://www.beasys.com/products/weblogic/drivers.shtml
PostgreSQL	PostgreSQL JDBC (версии 7.2 и 7.3 beta 3 не смогли произвести некоторые требующиеся преобразования типов данных; похоже на ошибки драйвера)	http://jdbc.postgresql.org/
PostgreSQL	jxDBCCon-doc-0.9z (успешно протестирован с PostgreSQL 7.2.3)	http://sourceforge.net/projects/jxdbccon
PostgreSQL	PostgreSQL JDBC 7.3 (успешно протестирован с PostgreSQL 7.2.3)	http://jdbc.postgresql.org/
Oracle	Oracle JDBC	http://otn.oracle.com/software/tech/java/sqlj_jdbc/content.html
Others		http://industry.java.sun.com/products/jdbc/drivers

Если для вашей базы нет подходящего JDBC драйвера, или он содержит слишком много ошибок, вы можете использовать JdbcOdbc драйвер, который является частью JRE. Популярная база имеет, по крайней мере, один ODBC драйвер. Это позволяет работать из JRE с ODBC datasource. Чтобы использовать JdbcOdbc драйвер, вам нужно создать DSN. Затем используйте

'sun.jdbc.odbc.JdbcOdbcDriver' как название драйвера и

'jdbc:odbc:<DSN>;UID=<username>;PWD=<password>;' как JDBC url (без угловых скобок).

5.4. Модуль регистрации событий.

5.4.С. Инсталляция MPC/C.

Инсталляция MPC/C заключается в его конфигурации. Ниже описание параметров конфигурационного файла:

параметр	описание	возможные значения
port	TCP/IP порт для принятия соединений	1..65535
address	адрес для принятия соединений	IP адрес
connections	максимальная длина очереди входящих соединений ⁴	1..65535
threads	число зарезервированных потоков ⁵	1..65535
shutdown.password	пароль для остановки сервера	любая строка
log.*	Конфигурация лога. Смотрите приложение А .	
socket.timeout	таймаут в/в сокета, мс, 0 – бесконечность	0..2147483647
socket.linger	linger time сокета, мс	0..2147483647

⁴ socket backlog

⁵ т.е. предварительно созданных и многократно используемых

socket.nodelay	опция NODELAY сокета	<i>true, false</i>
password	пароль для аутентикации клиента	непустая строка
db.*	Конфигурация связи с базой данных. Смотрите приложение Б .	
cachalot.keyfile	Полное имя ключа. Пример: "c:\cachalot\key.keyfile"	имя файла

Конфигурация требует некоторого понимания TCP/IP и JDBC. Ниже приведены наиболее важные советы для выбора правильных значений некоторых параметров:

параметр	советы
connections	<p>Определяет количество одновременных соединений с сервером. Если достигнут максимум, клиент сразу же получит ошибку "connection refused". Это число прямо пропорционально ожидаемому максимальному числу одновременных клиентских запросов (т.е. событий).</p> <p>Рекомендуется не сохранять ресурсы за счет очереди соединений. Если клиенты часто получают ошибку "connection refused", это прямая причина увеличить ограничение на число соединений.</p>
threads	<p>Число зарезервированных потоков. Оно определяет количество одновременных запросов, которые могут быть выполнены без выделения емкого ресурса – потока. Если достигнут максимум, обработчик будет создавать новые потоки, т.о. запросы будут обслуживаться несколько медленнее, чем, если максимум не был бы достигнут. Т.е. увеличение числа зарезервированных потоков ведет к ускорению обработки запросов за счет использования памяти. Это число прямо пропорционально ожидаемому типичному числу одновременных запросов и времени обработки запроса. Когда сервер создает поток, а не использует зарезервированный, он пишет сообщение в лог-файл, т.о. возможно определить, нужно ли увеличивать число потоков.</p> <p>Не имеет смысла устанавливать число потоков выше, чем число соединений, указанное в параметре 'connections'.</p>
socket.timeout	<p>Рекомендуется не использовать бесконечный таймаут. Таймаут зависит только от скорости сети (которая зависит от загрузки сети) и загрузки сервера. В случае локальной сети без крупных нагрузок и не загруженном сервере 1000 должно быть достаточно. Для медленной сети или частых крупных загрузках сети или сервера рекомендуемый минимум – 3000.</p>
socket.linger	<p>То же самое, что и для таймаута (т.к. МРС не пересылает большого количества данных).</p>
socket.nodelay	<p>Для локальной сети без крупных загрузок лучше установить <i>true</i>. В случае частых крупных загрузок сети или в случае медленной сети лучше поставить <i>false</i>.</p>

Если вы сомневаетесь насчет некоторых параметров вашей конфигурации или используете менее надежные значения, чем рекомендованные, рекомендуется обследовать лог-файлы сервера и клиентов на предмет ошибок, чтобы увидеть нуждаются ли конфигурации в улучшении.

После того, как вы заполнили конфигурационный файл значениями вашей конфигурации, вы можете запустить МРС/С из командной строки, используя предоставленный сценарий запуска/остановки (отредактировав переменные в начале сценария). Если вы желаете запустить МРС/С как сервис Windows NT, вам нужно использовать третьесторонний продукт, который позволяет запускать обычные приложения как сервисы (например, srvany).

5.4.К. Инсталляция MPC/К.

5.4.К.1. Клиенты командной строки.

Java и платформо-зависимая реализация клиента командной строки MPC/К не требуют инсталляции. Для описания параметров обратитесь к параметрам 'server.*' конфигурации Windows-интегрированного клиента, они такие же.

5.4.К.2. Windows-интегрированный клиент.

Для инсталляции Windows-интегрированного клиента MPC/С используйте инсталлятор rdgcswic.exe. Инсталлятор управляется параметрами командной строки и выполняет четыре функции:

первый параметр	функция	командная строка
view	Отображение текущей конфигурации.	rdgcswic.exe view
install	Установка (возможна установка поверх более ранней версии). В качестве второго параметра должно быть указано имя файла конфигурации.	rdgcswic.exe install rdgcswic.conf
configure	Изменение конфигурации. В качестве второго параметра должно быть указано имя файла конфигурации.	rdgcswic.exe configure rdgcswic.conf
uninstall	Деинсталляция.	rdgcswic.exe uninstall

При установке файл rdgcswic.dll должен находиться в том же каталоге, что и rdgcswic.exe. Файл конфигурации содержит значения параметров конфигурации, описанных ниже. В поставку включен пример файла конфигурации – rdgcswic.conf. После установки или изменения конфигурации необходимо перезагрузить компьютер, чтобы изменения вступили в силу.

параметр	описание	возможные значения
log.normal	Имя лог-файла, в который будет помещаться информация о событиях. Если путь не указан, файл будет помещен в %SystemRoot%\system32\.	имя файла
log.error	Имя лог-файла, в который будет помещаться информация об ошибках. Если путь не указан, файл будет помещен в %SystemRoot%\system32\.	имя файла
server.name	имя хоста для связи с MPC/С	адрес хоста
server.port	номер порта для связи с MPC/С	1..65535
server.password	пароль для аутентикации в MPC/С	любая строка
server.timeout	таймаут в/в сокета, мс	1..65535
server.attempts	число попыток послать событие перед отказом в случае ошибки	1..65535
neterror.allow	Имена пользователей и/или групп, которым позволено входить в случае ошибки при послании события входа на сервер. Остальным пользователям (не указанным и не принадлежащим ни одной из указанных групп) будет запрещен вход. Оставьте пустым, чтобы любой пользователь смог войти в случае ошибки при послании события входа на сервер (например, при отключенном сетевом кабеле). Пример: 'administrator, mydomain\Admin, mydomain\Managers'.	Список разделенных запятыми имен пользователей и/или групп

Наиболее важные советы для выбора значений некоторых параметров:

server.timeout	Таймаут зависит только от скорости сети (которая зависит от загрузки сети) и загрузки сервера. В случае локальной сети без крупных нагрузок и не загруженном сервере 1000 должно быть достаточно. При медленной сети или частых крупных загрузках сети или сервера рекомендуемый минимум – 3000.
server.attempts	Число максимальных попыток послать событие на сервер. Рекомендуемый минимум – 3.

Если вы сомневаетесь насчет некоторых параметров вашей конфигурации или используете менее надежные значения, чем рекомендованные, рекомендуется обследовать лог-файлы сервера и клиентов на предмет ошибок, чтобы увидеть, нуждаются ли конфигурации в улучшении.

5.5. Модуль регистрации трафика.

MPT требует наличия библиотеки Sourceforge JPCap 0.01.13.

Инсталляция MPT заключается в его конфигурации. Ниже приведено описание параметров конфигурационного файла:

параметр	описание	возможные значения
device.name	Имя устройства сетевого адаптера. Запустите <code>tru_listdevices.cmd</code> или <code>tru_listdevices.sh</code> , чтобы увидеть имена доступных устройств. Вместо имени можно указать IP-адрес, соответствующий адаптеру (цифрами) Пример: <code>'\Device\Packet NdisWan4', 'eth0', '192.168.0.1'</code> .	имя устройства
device.promiscuous	открывать ли устройство в promiscuous режиме	<i>true</i> или <i>false</i>
interval	Число секунд интервала накопления и суммирования информации о трафике перед внесением ее в базу данных. Укажите 0 для немедленного внесения информации о каждом пакете.	0..65535
log.*	Конфигурация лога. Смотрите приложение А .	
db.*	Конфигурация связи с базой данных. Смотрите приложение Б .	

Советы для выбора значения параметра 'interval': большее значение означает менее точную информацию о трафике сессии (т.к. информация о трафике будет, возможно, внесена в базу данных после того, как размер сессионного трафика был окончательно посчитан), но более быструю работу всей системы, т.к. в базе данных будет значительно меньший объем данных (это может быть хорошо заметно по времени генерации отчетов МИП) и MPT не будет непрерывно работать с базой.

После того, как вы заполнили конфигурационный файл значениями вашей конфигурации, вы можете запустить MPT из командной строки, используя предоставленный сценарий запуска/остановки (отредактировав переменные в начале сценария). Если вы желаете запустить MPT как сервис Windows NT, вам нужно использовать третьесторонний продукт, который позволяет запускать обычные приложения как сервисы (например, `sgvany`).

5.6. Модуль интерфейса пользователя.

Установка МИП заключается в его конфигурации и установке в JSP контейнер.

МИП можно конфигурировать двумя способами – через конфигурационный файл или через JNDI.

5.6.1. Задание конфигурации через конфигурационный файл.

Для задания конфигурации через конфигурационный файл необходимо определить переменную JVM UIU_CONF так, чтобы ее значением было полное имя конфигурационного файла (например, UIU_CONF=c:\cachalot\uiu.conf). Этот файл содержит параметры конфигурации МИП.

Некоторые контейнеры предусматривают возможность определить переменную JVM для контейнера. Для остальных контейнеров необходимо добавить параметр '-D' в строку запуска java. Переменная JVM задается добавлением в строку запуска java параметра '-Dname=value', т.е. '-DUIU_CONF=filename'. Например 'java -DUIU_CONF=c:\cachalot\uiu.conf mycontainerclassname'.

5.6.2. Задание конфигурации через JNDI.

Чтобы задать конфигурацию через JNDI, нужно чтобы переменная UIU_CONF была неопределена или имела пустое значение.

Для задания значений параметров через JNDI можно воспользоваться различными способами; эти способы зависят от конкретного контейнера. Наиболее распространенными являются следующие три способа:

- интерфейс пользователя для JNDI, предоставляемый контейнером
- указание параметров в файле конфигурации контейнера
- указание параметров в web.xml (файле конфигурации веб-приложения).

За подробности по первым двум способам обратитесь к документации контейнера.

Ниже приводится описание третьего способа.

xml-файл WEB-INF/web.xml архива uiu/uiu.war может содержать секции <env-entry> следующей структуры:

```
<env-entry>
  <env-entry-name>имя параметра</env-entry-name>
  <env-entry-value>значение параметра</env-entry-value>
  <env-entry-type>тип параметра</env-entry-type>
</env-entry>
```

Тип параметра должен быть 'java.lang.String'. Т.о., чтобы воспользоваться этим способом, нужно поместить для каждого параметра конфигурации такую секцию. Например:

```
<env-entry>
  <env-entry-name>cachalot.log.console</env-entry-name>
  <env-entry-value>off</env-entry-value>
  <env-entry-type>java.lang.String</env-entry-type>
</env-entry>
```

Для того, чтобы отредактировать файл WEB-INF/web.xml, необходимо распаковать, а после запаковать, файл веб-приложения uiu/uiu.war (при помощи unzip/zip или jar).

5.6.3. Параметры конфигурации.

В обоих способах задания конфигурации набор параметров и их назначение одинаковы, они описаны ниже:

параметр	описание	возможные значения
cachalot.db.*	Конфигурация связи с базой данных. Смотрите приложение Б (нужно добавить к именам параметров 'cachalot.').	
cachalot.db.capacity	Число зарезервированных соединений с базой данных. Оно определяет количество одновременных запросов к базе данных, которые могут быть выполнены без создания емкого ресурса – соединения с базой данных. Если достигнут максимум, при следующей необходимости будет создано новое соединение, и т.о. операция будет выполнена медленнее, чем, если максимум не был бы достигнут. Т.е. увеличение	0..65535

	<p>числа зарезервированных соединений ускоряет скорость выполнения операций за счет памяти и других ресурсов. Это число прямо пропорционально ожидаемому типичному числу одновременных запросов к МИП и времени выполнения запросов. Когда сервер создает дополнительное соединение, он пишет сообщение в лог-файл, т.о. можно увидеть, нужно ли увеличить число соединений.</p> <p>0 полностью отменяет резервирование соединений с базой данных.</p>	
cachalot.db.bigInt	<p>Функция SQL вашей базы данных для конвертирования целого числа в целочисленный тип с большим диапазоном (для того, чтобы сумма целых значений была корректным значением для больших суммируемых значений). На месте числа для конвертирования должно быть '%0%'.</p> <p>Если нужно использовать в функции символ '%', то используйте два символа '%', т.е. '%%' будут преобразованы в '%'. Например, в случае MSSQL это 'CONVERT(NUMERIC,%0%)', в случае PostgreSQL – '%0%'.</p>	строка, содержащая '%0%'
cachalot.db.dateDiff	<p>Функция SQL вашей базы данных для вычисления разности двух значений типа DATETIME в миллисекундах. '%0%' и '%1%' должны быть на месте значений типа DATETIME (%0% – более ранняя дата).</p> <p>Если нужно использовать в функции символ '%', то используйте два символа '%', т.е. '%%' будут преобразованы в '%'. Например, для MSSQL это 'DATEDIFF(ms,%0%,%1%)', в случае PostgreSQL – 'date_part('epoch',%1%-%0%)*1000'.</p>	строка, содержащая '%0%' и '%1%'
cachalot.log.events	<p>Коды событий, разделенные запятой. Определяет, какие события записывать в лог-файл. Коды:</p> <ul style="list-style-type: none"> - 'auth.success', 'auth.failure' – успешный/неудачный вход (на странице аутентикации МИП) - 'logout' – выход пользователя (из МИП) - 'entity' – управление (создание, модификация, изменение) сущностью (например пользователем) - 'system' – изменение системных параметров (например цены) - 'data.removed' – использование страницы 'Удаление данных'. <p>Пример: 'entity, system, data.removed'.</p>	коды событий, разделенные запятыми
cachalot.log.*	Конфигурация лога. Смотрите приложение А (нужно добавить к именам параметров 'cachalot.').	
cachalot.keyfile	<p>Полное имя ключа.</p> <p>Пример: "c:\cachalot\key.keyfile"</p>	имя файла

Изначально (после создания базы) в МИП зарегистрирован один пользователь с именем 'admin' и пустым паролем.

5.6.4. Установка в контейнер.

Процесс установки (deployment) зависит от конкретного контейнера. Система была протестирована с Apache Tomcat 4.0 и 4.1.18 (<http://jakarta.apache.org/tomcat/>) и Jetty 4.2.5 (<http://jetty.mortbay.com/jetty/index.html>).

Ниже обзорно описан процесс установки в эти контейнеры. Для установки в другой контейнер, обратитесь к его документации.

Установка в Tomcat 4.x.

Если вы используете задание конфигурации через конфигурационный файл, то вам потребуется определить переменную окружения CATALINA_OPTS со значением '-DUIU_CONF=путь к конфигурационному файлу'.

Вы можете установить МИП в Tomcat 4.x несколькими способами, в т.ч.:

- (Просто) Скопировав файл uiu/uiu.war в %CATALINA_HOME%/webapps. После запуска сервера, МИП будет доступен по адресу <http://yourhost:port/uiu/>. Если вы хотите, чтобы он был доступен прямо от корня сервера, то переименуйте лежащую в webapps папку ROOT, например в orig, а uiu.war в ROOT.war. После запуска сервера, МИП будет доступен по адресу <http://yourhost:port/>.
- Добавив такую секцию <Context> в секцию <Host> файла %CATALINA_HOME%/conf/server.xml:

```
<Context
  path="/cachalot"
  docBase="c:/RatheDG/cachalot/uiu.war"
/>
```

где *path* – часть URL МИП относительно корневого URL вашего сервера, *docBase* – местонахождение файла uiu.war. После запуска сервера, МИП будет доступен по адресу <http://yourhost:port/cachalot/>.

Используемый вами JDBC драйвер можно скопировать в %CATALINA_HOME%/lib (для Tomcat 4.0.x) или %CATALINA_HOME%/shared/lib (для Tomcat 4.1.x), тогда он будет автоматически доступен МИП для использования.

За более подробной информацией об установке обратитесь к документации

%CATALINA_HOME%/webapps/tomcat-docs/config/context.html или <http://jakarta.apache.org/tomcat/tomcat-4.1-doc/index.html>

Установка в Jetty 4.2.5.

Для данной версии Jetty невозможно задание конфигурации через JNDI. Вы можете установить МИП в Jetty 4.2.x несколькими способами, в т.ч.:

- (Просто) Скопировав файл uiu/uiu.war в %JETTY_HOME%/webapps. После запуска сервера, МИП будет доступен по адресу <http://yourhost:port/uiu/>. Если вы хотите, чтобы он был доступен прямо от корня сервера, то переименуйте лежащую в webapps папку root, например в root.orig, а uiu.war в root.war. После запуска сервера, МИП будет доступен по адресу <http://yourhost:port/>.
- Добавив такую секцию <Call> в секцию <Configure> файла %CATALINA_HOME%/etc/jetty.xml:

```
<Call name="addWebApplication">
  <Arg>/cachalot</Arg>
  <Arg>c:/RatheDG/cachalot/uiu.war</Arg>
  <Set name="extractWAR">>false</Set>
</Call>
```

где первый элемент *Arg* – часть URL МИП относительно корневого URL вашего сервера, второй элемент *Arg* задает местонахождение файла uiu.war. После запуска сервера, МИП будет доступен по адресу <http://yourhost:port/cachalot/>.

Используемый вами JDBC драйвер можно скопировать в %JETTY_HOME%/ext, тогда он будет автоматически доступен МИП для использования.

После чего, находясь в %JETTY_HOME%, запустите Jetty следующим образом:

```
java -DUIU_CONF=конфигурационный_файл -jar start.jar etc/jetty.xml
```

За более подробной информацией об установке обратитесь к документации

<http://jetty.mortbay.com/jetty/index.html>

6. Безопасность.

Следующее предполагается об окружении:

- неавторизованное лицо не имеет доступа к файлам сервера
- неавторизованное лицо не имеет доступа к базе данных
- неавторизованное лицо не имеет доступа к клиентскому паролю МРС, который хранится на клиентских машинах
- неавторизованное лицо не может узнать клиентский пароль МРС сканированием пакетов сети (sniffing)
- неавторизованное лицо не может изменять IP адрес клиентского компьютера
- пароль пользователя МИП известен только пользователю и никому больше.

Если все эти условия выполнены, система остается полностью безопасной, неавторизованное лицо не может: изменить настройки системы, изменить данные, предоставить ложные исходные данные и т.п. Аутентикация МИП устойчива к лобовому подбору пароля, однако рекомендуется периодически просматривать лог-файлы для того, чтобы увидеть попытки подбора.

Приложения

Приложение А. Конфигурация лога.

Параметры лог-файла:

log.filename	Имя лог-файла, может содержать путь; оставьте пустым или закомментированным, чтобы вывод в лог-файл не производился. Пример: 'c:\logs\eru.log' или '/opt/cachalot/logs/eru.log'.	имя файла
log.period	период ротации лог-файла	hour, day, month
log.prefix	добавлять ли дату перед именем файла	true, false
log.usefilesseparator	использовать ли файл-сепаратор в дате в имени файла, вместо '.', не имеет значения при log.prefix=false	true, false
log.console	дублировать ли лог-вывод на консоль	on, off

Советы для выбора значений некоторых параметров:

log.period	Меньший период облегчает регулярные просмотры лог-файлов и поиск конкретного места в лог-файле за счет очень небольшого расхода ресурсов.
log.usefilesseparator	Если true, лог-файлы будут автоматически создаваться в древовидной структуре папок, например, год/месяц/день для периода ротации hour.
log.console	Полезно для целей отладки и анализа. Рекомендуется выключать при запуске сервера фоновым процессом.

Приложение Б. Конфигурация соединения с базой данных.

Параметры конфигурации соединения с базой данных:

db.driver	имя класса JDBC драйвера базы данных, класс должен быть доступен в CLASSPATH. Пример: 'weblogic.jdbc.mssqlserver4.Driver' или 'org.postgresql.Driver'.	имя класса
db.url	JDBC url базы данных, она должна подходить для указанного драйвера и корректно определять базу данных. Проконсультируйтесь с документацией вашего JDBC драйвера насчет синтаксиса url.	JDBC url

	Пример: 'jdbc:weblogic:mssqlserver4:localhost:1433;user=cachalot;password=cachalotpass;' или 'jdbc:postgresql:net//pguser:pgpass@myhost:5438/cachalotdb'.	
db.transaction	Уровень изоляции транзакций базы данных	<i>none, read_uncommitted, read_committed, repeatable_read, serializable</i>

Ниже приведено объяснение параметра 'db.transaction'.

Обычно изменение уровня изоляции транзакций ведет к различному соотношению производительности и риска несообразности данных. Очевидно, большая производительность ведет к увеличению риска, и меньший риск ведет к меньшей производительности. Предполагается, что в Cachalot System нет сверхважных данных. Максимальная потеря – потеря событий в случае каких-то аппаратных или программных сбоев; это не причиняет значительного ущерба.

Текущая реализация Cachalot System работает в режиме auto-commit, и предполагается, что уровень изоляции транзакций "none", т.е. нет транзакций. Но некоторые базы данных не поддерживают этот уровень (например, MSSQL). В конфигурации есть параметр для каждого модуля, который должен быть определен администратором, согласно возможностям базы данных и/или собственному желанию.

Набор поддерживаемых уровней изоляции транзакций специфичен для конкретной базы данных. Ниже приведены технические описания пяти широко используемых уровней изоляции транзакций в порядке увеличения надежности и уменьшения производительности. Администратор должен указать один из них в конфигурации каждого модуля.

- isolation level "none"
Indicates that transactions are not supported.
- isolation level "read_uncommitted"
Dirty reads, non-repeatable reads and phantom reads can occur. This level allows a row changed by one transaction to be read by another transaction before any changes in that row have been committed (a "dirty read"). If any of the changes are rolled back, the second transaction will have retrieved an invalid row.
- isolation level "read_committed"
Dirty reads are prevented; non-repeatable reads and phantom reads can occur. This level only prohibits a transaction from reading a row with uncommitted changes in it.
- isolation level "repeatable_read"
Dirty reads and non-repeatable reads are prevented; phantom reads can occur. This level prohibits a transaction from reading a row with uncommitted changes in it, and it also prohibits the situation where one transaction reads a row, a second transaction alters the row, and the first transaction rereads the row, getting different values the second time (a "non-repeatable read").
- isolation level "serializable"
Dirty reads, non-repeatable reads and phantom reads are prevented. This level includes the prohibitions in TRANSACTION_REPEATABLE_READ and further prohibits the situation where one transaction reads all rows that satisfy a WHERE condition, a second transaction inserts a row that satisfies that WHERE condition, and the first transaction rereads for the same condition, retrieving the additional "phantom" row in the second read.

© 2001-2003 Rathe Development Group. Все права защищены.

Использование является предметом лицензионного соглашения.

<http://www.rathedg.com/>
support@rathedg.com