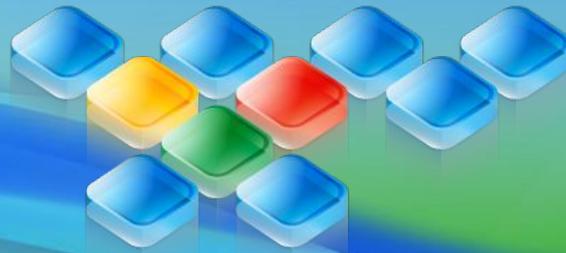


Microsoft®  
**Платформа**

0000000010010010111000111 2010 01011





# SQL Server 2008, восстановление при катастрофических сбоях

**Дмитрий Артемов**

Microsoft





Самое время для страшных сказок на ночь

# Содержание

- Отличия между 2005 и 2008
- Ошибки при попытках восстановления
- Как быстро восстановится
  - Как определить наличие сбоя
  - Как восстанавливаться
  - Правильные приемы
- Что говорят о восстановлении и что из этого верно

# Отличия между 2005 и 2008

- Database Mirroring
  - Автоматическое восстановление страниц
- Backup compression
  - Копия меньше и создается гораздо быстрее
- DBCC Checkdb EXTENDED\_LOGICAL\_CHECKS
  - Если БД имеет уровень 100 (SQL Server 2008) или выше, выполняются проверки логической целостности на имеющихся индексированных представлениях, XML индексах и spatial индексах
- BACKUP LOG with NO\_LOG и BACKUP LOG with TRUNCATE\_ONLY больше не принимаются ни под каким видом (проверьте ваши JOB'ы)

# Ошибки при попытках восстановления

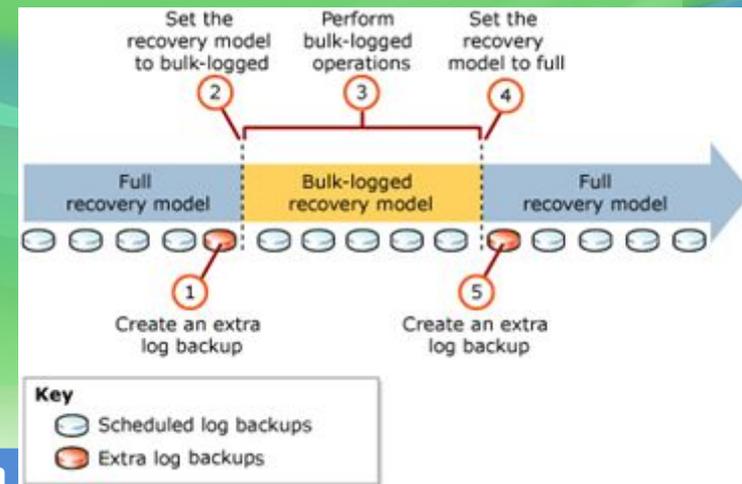
10

Некорректное использование модели восстановления

- Первое что следует определить при создании/получении БД
- Ошибка может иметь серьезные последствия
- Выбор модели влияет на:
  - Производительность
  - Время резервного копирования
  - Сложность плана восстановления
  - Способность к восстановлению
  - Администрирование

# Типы моделей восстановления

- Simple
  - Журнал особенно не растёт
  - Журнал используется только для поддержания целостности
  - Журнал не используется при резервном копировании (только full backup, differential backup, partial full и differential backup и file group backups для групп read-only)
- Bulk-Logged
  - Массивные операции журналируются минимально, остальные полностью
  - Повышает производительность массивных операций
  - Копирование \ восстановление журнала возможно, но без point in time recovery (если копия журнала содержит изменения от массивных операций)
  - **Размер копии журнала может сильно вырасти т.к. в копию попадают не только записи журнала но и все Extents, модифицированные массивными операциями**
- Full
  - Все транзакции журналируются полностью
  - Оказывает влияние на производительность всех транзакций
  - Возможно Point in time recovery



# Ошибки при попытках восстановления

## 9 Плохо продуманная политика хранения

- Где хранят копии, сколько времени нужно, чтобы их доставить...

## 8 После того как копия записана на диск, все хорошо?

- Их можно просмотреть средствами Notepad
  - Права на доступ к папкам
  - Шифрование диска\ленты
  - Encrypted backup (SQL Server 2008)

# Ошибки при попытках восстановления

7

Нужно понимать, что копия БД, это не вся система

- Копия хранит многое но...
- Не все, что может понадобится при любом аварийном восстановлении
- Связь login  user
  - Master..syslogins  login | SID
  - Userdb..sysusers  dbuser | SID
  - Нет гарантии, что SID одинаков на разных серверах
  - Доступ к БД может быть нарушен
- Полнотекстовые каталоги в SQL Server 2000
  - Теперь (SQL 2005/8) они включены в основную копию
- Восстановление БД часто только первый шаг восстановления системы

# Ошибки при попытках восстановления

## 6 Игнорируются (недооцениваются) системные БД

- А ведь они не менее важны чем пользовательские БД
- Master хранит информацию на уровне экземпляра
  - Login / password
  - Конфигурация
- MSDB хранит описания заданий
- Model – шаблон для всех новых БД
  - Может хранить набор стандартных объектов

# Ошибки при попытках восстановления

**5** Нельзя полагаться только на Backup/Restore

- Спасут ли они от логической ошибки пользователя или приложения
- BACKUP Database завершился без ошибок.

**4** Что, не о чем беспокоится?

- Н Е Т!
- Периодическое тестовое восстановление в режиме выхода из сбойной ситуации, вот что дает уверенность
- Всегда следите за ростом размера БД и временем копирования \ восстановления

**3**

# Ошибки при попытках восстановления

## 2 Прямое копирование на ленту

- Это уже не считается правильным
- Хотя лента по-прежнему – важный элемент процесса восстановления
  - Копируйте диск на ленту
  - Архив на ленту только после успешного копирования на диск

## 1 Администратор теряет контроль

- План восстановления должен обеспечивать полную независимость администратора от внешних условий

# Чего НЕ надо делать

- Копировать на тот же физический диск, где лежат файлы БД
- Игнорировать проверку на целостность перед копированием
- Игнорировать проверку успешности копирования (журнал задания SQL Server Agent и собственно запись на диск)
- Игнорировать верификацию копии (RESTORE VERIFYONLY).
- Игнорировать полномасштабное тестирование всего процесса восстановления

# Как построить план аварийного восстановления?

- **Как много вы готовы потерять?**
- Что имеется в распоряжении из технологий, повышающих отказоустойчивость?
- Сколько у вас есть времени (SLA)?
- Насколько секретны данные?
- Как проверить, что план работает?



# Как определить наличие сбоя

Как определить наличие сбоя  
Как восстанавливаться  
Правильные приемы

# Как узнать что произошло?

- Смотрите SQL Server ERRORLOG
  - Ошибки ввода \ вывода, контрольных сумм, повторные чтения
  - Ошибки штатного восстановления (Recovery) БД
  - Неожиданные рестарты SQL Server
- Смотрите журналы Windows
  - Ошибки ввода \ вывода, неожиданные рестарты сервера
- Включите дополнительный аудит
  - Страниц в памяти (TF806) и записей журнала транзакций (TF3422)
  - Контрольные суммы
- Используйте SQLIOSim, чтобы протрясти стойку
- Проведите диагностику аппаратуры и проверьте версии firmware

# Контрольные суммы

- Создаются для каждой страницы
  - Записываются как последняя операция при физической записи страницы
  - Проверяются первой операцией физического чтения
  - Увеличивают процессорную загрузку
- Не сошедшаяся сумма однозначно говорит о сбое стойки
  - (иначе начинается «сам дурак» между администраторами SQL и поставщиками оборудования)
- При создании новой БД включено по умолчанию для SQL Server 2005/8
  - Не включается для БД, перенесенных с SQL 2000
  - Enterprise edition: lazy writer проверяет контрольные суммы в памяти (выборочно)
- Если суммы не сходятся, выдается ошибка 824
  - Ошибка физического чтения на уровне ОС выдает ошибку 823
- Имейте в виду:
  - НЕ используется для TEMPDB
  - Работает только для записанных страниц (пока страница не тронута, сумма не формируется)

# Повторные чтения

- Присутствует SQL Server 2000 в ограниченных пределах
  - Только для операций сортировки
- Расширено в SQL Server 2005 для страниц данных
- При ошибке чтения попытка повторяется 4 раза
- На пятой попытке выдается ошибка ввода \ вывода
- Если попытка “удалась” делается запись в ERRORLOG:
  - “A read of the file <<FILE NAME>> at offset <<PHYSICAL OFFSET>> succeeded after failing <<RETRY COUNT>> time(s) with error: <<DETAILED ERROR INFORMATION>>. Additional messages in the SQL Server error log and system event log may provide more detail. This error condition threatens database integrity and must be corrected. Complete a full database consistency check (DBCC CHECKDB). This error can be caused by many factors; for more information, see SQL Server Books Online.”
- Тревожный звонок, следует обратить внимание на подсистему ввода \ вывода

# 82\* ошибки

- 823 – ошибка ввода вывода. SQL Server затребовал чтение страницы, но ОС не смогла его выполнить.
- 824 – ошибка чтения. На уровне ОС чтение состоялось, но SQL Server решил, что страница повреждена – например при проверке контрольной суммы
- 825 – ошибка повторного чтения. При возникновении ошибки 823 или 824, SQL делает повторные попытки и если получилось, записывает в ERRORLOG сообщение. Это тревожный звонок.

# DBCC CHECKDB

- Единственный способ прочитать все выделенные страницы БД
  - Можно использовать для принудительного расчета контрольных сумм
- Можно выбирать между полной проверкой и WITH PHYSICAL\_ONLY
- Оптимизирована для ускорения работы и выполнения **ONLINE** операций
  - По сравнению с 6.5 или 7.0
- Новые возможности SQL Server 2005/8
  - Использует Snapshot
  - Указание % исполнения
  - Фиксирует последний раз нормального завершения
  - Более не демонстрирует “плавающих” проблем: если проблема есть DBCC на нее укажет
  - Ремонт в режиме Emergency
  - Расширенные логические проверки (индексированные представления и чистота данных)

# CHECKDB: как это работает

- 1. Получает целостное представление о БД (средствами TABLOCK или Snapshot). См. KB 926070 о возможных проблемах использования Snapshot
- 2. Примитивные проверки критичных системных таблиц: тех, что хранят метаданные, используемые компонентом Storage Engine
- 3. Проверки размещения (GAM, SGAM, IAM, PFS,... pages)
- 4. Логические проверки (начинает с критичных системных таблиц. Если найдены необратимые ошибки, стоп. Иначе, проверяем остальные таблицы)
  - 4.1 Проверяем метаданные каждой таблицы в части storage engine
  - 4.2 Читаем и проверяем все страницы данных, индексов, BLOB
  - 4.3 Проверяем связность страниц
  - 4.4 Проверяем счетчики в заголовках страниц (slot, ghost record, free space)
- 5. Логически проверки более высокого уровня
  - 5.1 Проверки для Service Broker (связи между conversation, endpoint, message и queue)
  - 5.2 Перекрестные проверки метаданных (Relational Engine)
  - 5.3 Проверки Indexed view и XML index

# Как правильно

- Всегда вызывайте DBCC CHECKDB WITH ALL\_ERRORMSGCS из консоли (sqlcmd [с выводом результатов в файл]) или задания с выводом результатов в файл
- SSMS выводит только первые 1000 сообщений



# Как восстановиться

Как определить наличие сбоя  
Как восстанавливаться  
Правильные приемы

# Используйте резервную копию

- Это самое лучшее, что можно посоветовать
- SQL Server 2005/8 предлагает массу вариантов
  - Полное восстановление БД
  - Восстановление файлов
  - Восстановление страниц
  - Восстановление файловых групп
  - Только Recovery (WITH RECOVERY для БД, имеющих более одной файловой группы)
  - Восстановление журнала
  - Создание зеркала (WITH NORECOVERY)
- **Перед тем как использовать BACKUP, его нужно создать**
- **И он должен быть рабочим**
  - Используйте VERIFY ONLY, CONTINUE\_AFTER\_ERROR
  - Создавайте с CHECKSUM (одновременно проведите контрольные суммы на страницах, если они
- А если \_Е\_Г\_О\_ нет?...



Будем  
делать  
REPAIR?

# Да, будем делать REPAIR!

- Какова цель “ремонта”?
  - **Починить структуру БД, а не восстановить вам данные!!!**
- Все поддается ремонту? Нет!!
  - Критические повреждения системного каталога
  - Повреждения PFS страниц
  - Ошибки «чистоты» данных
- Ремонт всегда Offline?
- **Итого: Ремонт означает простой и, как правило, потерю данных**

# Жест отчаяния, который иногда делают первым... Перестройка журнала

- DBCC REBUILD\_LOG не документированная и опасная команда...

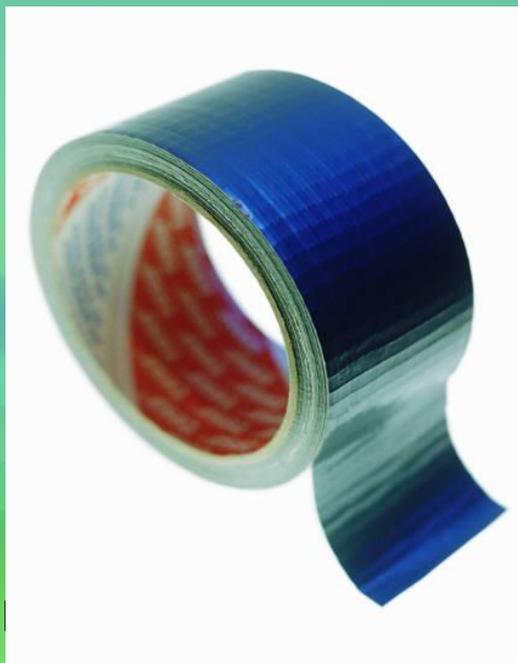


- Лог создан заново, но БД теперь теряет транзакционную целостность и считается поврежденной!!

# Жест отчаяния, который иногда делают первым...

## Database repair

- REPAIR\_ALLOW\_DATA\_LOSS его ведь не просто так назвали



- Структура БД исправлена, но потерянных данных - нет!

# Ну и если вообще ничего не помогает!! Документированный ремонт в emergency режиме

Находясь в EMERGENCY mode, вы можете использовать DBCC CHECKDB, чтобы вернуть БД online статус.

Единственно возможный вариант - REPAIR\_ALLOW\_DATA\_LOSS в усиленном режиме:

- ❖ Принудительное восстановление по журналу (если он еще цел).
- ❖ Аналог 'recovery with CONTINUE\_AFTER\_ERROR'
- ❖ Терять уже нечего, стараемся восстановить как можно больше
- ❖ Перестройка журнала - если он поврежден
- ❖ Исполнение DBCC CHECKDB в режиме REPAIR\_ALLOW\_DATA\_LOSS
- ❖ Возврат БД в ONLINE режим

Все вышеописанное - непрерывная и безоткатная операция



# Правильные приемы

Как определить наличие сбоя  
Как восстанавливаться  
Правильные приемы

# Как часто нужно запускать CHECKDB?

- Зависит от:
  - Стабильности подсистемы ввода \ вывода
  - Стратегии резервирования
  - Приемлемого простоя при сбоях
  - Приемлемых потерь при сбоях
  - Наличия административного окна под доп. нагрузку IO и CPU
  - Типа системы? (пром, тест, ...)
  - Секционирования?
- Например:
  - Нет копий и регулярные повреждения
  - VLDB с очень жесткими требованиями по простоям \ потерям

# Сколько времени это займет?

- Зависит от множества факторов:
  - Размер БД
  - Конкурентного доступа к стойке
  - Конкуренции за процессоры
  - Интенсивности модификаций
  - Возможностей стойки
  - Числа процессоров
  - Скорости дисков под TEMPDB (используется для промежуточного хранения)
    - Переполнение TempDb может привести к падению DBCC
  - Сложности схемы БД
  - Опций DBCC
  - Числа и типов повреждений

# Как быть с очень большими БД?

- DBCC CHECKDB на очень большой БД идет очень долго!!
- Как ускорить:
  - Не проверять! Да, это тоже вариант (плохой!)
  - Использовать WITH PHYSICAL\_ONLY
  - Разбивать проверки
    - CHECKALLOC
    - CHECKCATALOG
    - CHECKTABLE
  - Использовать секционирование
  - Использовать второй сервер
- **Итого: не сдавайтесь – если подумать, все ВОЗМОЖНО**

# Restore или repair?

- Торговля между потерей и скоростью восстановления (хотя, на очень большой БД ремонт может идти..., страшно сказать сколько)
- К сожалению, нет ни легкого решения, ни автоматизированного способа понять как себя вести, но вы можете принять осознанное решение
  - Есть ли резервная копия?
  - Сколько времени требуется на восстановление из копии (в различных режимах)?
  - Сколько времени занимает исполнение CHECKDB?
  - Какой объем данных вы готовы потерять?

# Стандартные неправильные советы (или “не стоит верить всему, что можно прочитать в Интернете...”)

- Просто выполните ремонт
- Просто перестройте журнал
- Detach и Attach вашу БД
  - Может просто не подключиться
- Просто восстановите из копии
- Запустите <комбинация СHECK\* команд>

Факультатив (см. в конце презентации)



Что говорят о восстановлении и что из этого верно

# Заключение

- Не ждите проблем, займите активную позицию
- Раннее обнаружение минимизирует потери и простой
- Разберитесь с режимами восстановления и умейте применять их
  - Если не делать копий, их невозможно использовать!!!
- Разработайте план аварийного восстановления, соответствующий вашему сервисному контракту
  - Если сервисный контракт предъявляет нереальные требования, не принимайте его
- Практика, практика, практика!!

# Двухдневный семинар с лабораторными работами "Обеспечение работоспособности БД SQL Server (2005/2008) при катастрофических сбоях"

## Презентации

- 00.SQL 2005 vs SQL 2008 from data recovery standpoint
- 01.SQL 2005&2008 On-disk Data Structures
- 02.Recovery & Logging
- 03.Backup Restore (Lab 01)
- 04.Disaster Recovery Mistakes (Lab 02 03 04)
- 05.Partial Database Availability (Lab 05)
- 07.Secrets Fast Detection Recovery
- 07a.Misconceptions about database repair (Lab 06 07 08)
- 08.Architectural Best Practices
- 09.End-to-End Troubleshooting for SQL Server 2005/8

## Лабораторные работы

- 01-EmergencyRestore
- 02-Recovery Failure (Deferred Transactions)
- 03-Recovery Failure (Error Handling in Media Recovery).
- 04-Recovery Failure (Restore Pending)
- 05-Partial Database Restore
- 06-Data Recovery Preventative Techniques
- 07-Restoring backup
- 08-Rebuilding System Databases

Читает Дмитрий Артемов  
Требуется класс для работы руками

Условия заключения контракта и возможные корректировки содержания оговариваются отдельно



# Вопросы?



# Что говорят о восстановлении и что из этого верно

# #1

- **Repair не ведет к потере данных.** Зависит. Если использовать REPAIR\_ALLOW\_DATA\_LOSS, потери неизбежны. НЕ зря же ее так назвали.
- **Repair следует исполнять всегда.** Нет. Сначала надо определить причину сбоя. Если у вас поврежден кластерный индекс в 1TB, Repair перестроит его. Если нет свободных 1TB, перестроение не пройдет, и вы в исходной позиции после многих часов бессмысленной активности. Возможно стоит потратить это время более продуктивно.

# #2

- **Можно выполнить ремонт не прибегая к DBCC CHECKDB.** Нет. Это одна из опций команд проверки (DBCC CHECKALLOC, DBCC CHECKTABLE или DBCC CHECKDB. DBCC CHECKFILEGROUP и DBCC CHECKCATALOG не выполняют ремонта).
- **После ремонта все станет хорошо.** Нет. После ремонта всегда следует исполнить DBCC CHECKDB для проверки качества ремонта. Иногда исправление одной ошибки дает возможность найти другую. Кроме того, ремонт мог удалить данные. Как теперь поведет себя приложение? Что делать с потерями?

# #3

- **Repair всегда все починит.** Нет. Есть случаи, когда DBCC CHECKDB не может помочь
  - Поврежденный системный каталог
  - Страницы кластерных индексов системных таблиц
  - Страницы PFS
  - Ошибки чистоты данных
- **Repair можно использовать на системных БД.** Нет. Ремонту не поддаются Master или Tempdb, так как их нельзя перевести в однопользовательский режим. Можно ремонтировать Model, но это вряд ли необходимо (обычно там нет пользовательских таблиц). Можно ремонтировать Msdb, но побочные эффекты могут быть самыми неожиданными

# #4

- **Можно выполнить ремонт в Online.** Нет. Repair всегда offline, БД должна быть в однопользовательском режиме.
- **REPAIR\_REBUILD все починит.** Нет. REPAIR\_REBUILD правит только некластерные индексы. Начиная с 2005, REPAIR\_FAST вообще ничего не делает.
- **Ремонт издателя распространяется на подписчиков.** Нет. То что делает Repair НЕ отмечается для репликации. После ремонта издателя нужно синхронизировать подписчиков.

# #5

- **Ремонт всегда правит ограничения. Нет.** Repair вообще не знает о наличии ограничений. После ремонта всегда следует выполнить DBCC CHECKCONSTRAINT для проверки.
- **Ремонт пытается спасти данные. Нет.** Основная задача – восстановление структур хранения и удаление того, что невозможно восстановить.

# #6

- **Ремонт в режиме EMERGENCY всегда поможет.** Нет. Если что-то сбилось на уровне файловой системы, ремонт может не пройти.
- **Можно откатить ремонтные операции.** Зависит. Явную транзакцию можно откатить. Ремонт в режиме EMERGENCY откату не подлежит