УДК 004.056.5 (043.2)

ВІДМОВОСТІЙКА ФАЙЛОВА СИСТЕМА

Л. П. Галата*, *Ю. О. Максимов

Національний авіаційний університет

lili-lili@bigmir.net

Проаналізовано дискову реалізацію структури Resilient File System та досліджено переваги і недоліки архітектури, продуктивності та відмовостійкості. Проведено оцінювання ефективності відновлення файлів після збою обладнання на основі структури метаданих. Розглянуто журнали транзакцій як підхід для забезпечення узгодженості на диску. Протестована перевірка надмірності системи зберігання даних та швидкість роботи на прикладі пошуку атрибутів папки.

Ключові слова: відмовостійка файлова система; B+-дерева; метадані; журнал транзакцій; цілісні потоки.

This article analyzes the implementation of disk structure Resilient File System, explored the advantages and disadvantages of architecture, performance and fault tolerance. The estimation of efficiency of recovering files after a hardware failure occurs on the basis of metadata structures, transaction logs are considered as an approach to maintain consistency on the disk. Tested checking redundancy data storage system and the speed of the search attribute in the example folder.

Keywords: Resilient File System; B +-trees; metadata; journal of transaction; the integral streams.

Вступ

Восени 2012 р. вийшла публічна бета-версія Microsoft Windows Server 2012 з підтримкою анонсованої файлової системи ReFS (Resilient File System — відмовостійка файлова система), раніше відомої під кодовою назвою «Protogon». Ця файлова система пропонується як альтернатива файловій системі NTFS у сегменті систем зберігання даних на базі продуктів

Microsoft. ReFS створена на основі NTFS, тому в ній збереглися найважливіші можливості сумісності та одночасно вона розроблена і спроектована з урахуванням потреб нового покоління технологій і сценаріїв обробки та зберігання даних.

Постановка проблеми

«... Сьогодні система NTFS є найбільш широко використовуваною, передовою та функціонально файловою системою. Але переосмислюючи ОС Windows, а ми в даний момент модернізуємо Windows 8 — не зупиняємося на досягнутих висотах...» [1].

Виходячи зі слів екс-президента підрозділу Windows Стівена Сінофскі, компанія Microsoft як лідер програмних передових інновацій розробляла новий Windows Server із сімейства операційних систем Windows 8 з упровадженням «хмарних» технологій, файлової системи відновлення даних, технологією віртуалізації Hyper V, IIS 8.0 з підтримкою ASP.NET 3.5/4.5, новим протоколом WebSocket, оновленими можливостями Server Core та PowerShell.

Зважаючи на кількість оновлень та інновацій, запропонованих у Server 2012, можна стверджувати, що вона є найсучаснішою серверною операційною системою.

Аналіз досліджень і публікацій

Варіант файлової системи має підтримку кластерів даних розміром тільки 64 та 16 Кб. При форматуванні ФС єдиним доступним варіантом для вибору розміру кластера є 64 Кб. Такий розмір кластера є більш ніж достатнім для організації файлових систем будь-якого розміру з практично реалізованих, але водночас призводить до відчутної надмірності при зберіганні даних.

Цілі

Мета статті — опис структури нової файлової системи, її переваг і недоліків, а також аналіз технологій створення, архітектури та продуктивності.

Архітектура файлової системи

Дискова реалізація структур ReFS кардинально відрізняється від інших файлових систем Microsoft. Основними структурними елементами нової файлової системи є «B+-дерева». Всі елементи структури файлової системи представлені списками або багаторівневими «B+-деревами», що дозволяє значно масштабувати будь-який з елементів файлової системи. 64-бітна нумерація всіх елементів системи виключає появу нефрагментованої області при її зміні. Усі, крім кореневого, записи мають розмір цілого блоку метаданих (у даному випадку — 16 Кб); проміжні коди мають невеликий розмір (близько 60 байт), тому потрібна невелика кількість рівнів дерева для опису навіть дуже великих структур, що досить сприятливо позначається на загальній продуктивності системи [1].

Основним складовим елементом файлової системи є «каталог», представлений у вигляді

«B+-дерева», ключем в якому є номер об'єкта-папки. Файл у ReFS не є окремим ключовим елементом «каталогу», а лише існує у вигляді запису [1].

Для папки існують три основних типи записів: дескриптор каталогу, індексний запис і дескриптор вкладеного об'єкта. Всі такі записи складені у вигляді окремого «B+-дерева», що має ідентифікатор папки; корінь цього дерева є розгалуженням «B+-дерева», що дозволяє записувати у папку будь-яку кількість записів. На нижньому рівні «B+-дерева» папки знаходиться в першу чергу запис дескриптора каталогу, що містить основні відомості про папку (ім'я, «стандартна інформація», атрибут імені файлу і т.д.).

Структури даних мають багато спільного з прийнятими в NTFS, хоча і мають низку відмінностей, основним з яких є відсутність типізованого списку іменованих атрибутів.

У каталозі є так звані індексні записи: короткі структури, які містять дані про елементи, що містяться в папці. Порівняно з NTFS ці записи значно коротші, що певною мірою перевантажує томи метаданими. Для папок ці елементи містять ім'я папки, її ідентифікатор у «каталозі» і структуру «стандартної інформації». Для файлів ідентифікатор відсутній, але замість цього структура містить всі основні дані про файл, включаючи корінь «B+-дерева» фрагментів файлу. Відповідно файл може складатися з великої кількості фрагментів.

На диску файли розташовуються в блоках розміром 64 Кб, хоча адресуються так само, як і блоки метаданих (кластерами розміром 16 Кб). «Резидентність» даних файлу на ReFS не підтримується, тому файл розміром 1 байт на диску займе цілий блок 64 Кб, що веде до значної надмірності зберігання на дрібних файлах, з іншого боку, це спрощує керування вільним простором і виділення вільного місця під новий файл здійснюється значно швидше.

Розмір метаданих порожньої файлової системи становить близько 0,1 % від розміру самої файлової системи (тобто близько 2 ГБ на 2 ТБ). Деякі основні метадані дублюються для кращої стійкості від збоїв [1].

Захищеність від збоїв

Архітектура файлової системи володіє всіма необхідними інструментами для безпечного відновлення файлів навіть після серйозного збою обладнання. Частина структур метаданих містить власні ідентифікатори, що дозволяє перевірити належність структури; посилання на метадані містять 64-бітові контрольні суми блоків, що дозволяє оцінити цілісність прочитаного за посиланням блоку [1]. При цьому варто відзначити, що контрольні суми для користувача даних не підраховуються. З одного боку, це відключає механізм перевірки цілісності в зоні даних, з іншого ж боку, це прискорює роботу системи за рахунок мінімальної кількості змін в зоні метаданих. Зміна структури метаданих здійснюється в два етапи: спочатку створюється нова копія метаданих у вільному дисковому просторі, потім атомарною операцією поновлення проводиться переклад посилання зі старої на нову зону метаданих. Така стратегія має назву «Copy-on-Write (копіювання-при-записі)» та дозволяє обійтися без аудиту, зберігаючи автоматично цілісність даних [1].

Видалення файлу здійснюється перестроюванням структури метаданих (з використанням CoW), що зберігає попередню версію блоку метаданих на диску. Це робить відновлення видалених файлів можливим до їх перезапису новими даними.

Відмовостійкість диска

Однією з найважливіших задач проектування ФС є виявлення та виправлення пошкоджень. Всі метадані ReFS перевіряються по контрольних сумах на рівні сторінки дерева «B+», а самі контрольні суми зберігаються окремо від сторінки. Це дає змогу виявити всі форми ушкоджень диска, включаючи втрачені або збережені не в тому місці записи та бітовий розпад (погіршення стану даних на носії). Оновлення контрольних сум відбувається автоматично при записі даних, тому якщо під час запису відбувається збій даних, завжди буде системнодоступна версія файлу.

ReFS і простори зберігання даних (*Storage Spaces*) розроблялися як два взаємодоповнювальних компонента єдиної системи зберігання даних. Крім поліпшення продуктивності, простори зберігання захищають дані від часткових і повних збоїв диска за рахунок копій на декількох дисках. При збоях читання простори зберігання можуть читати змінені копії, а при збоях запису (і у випадку повної втрати носія при читанні/записі) прозоро розподіляти дані заново. У багатьох випадках збій відноситься не до носія, а виникає через пошкодження даних або через те, що записи були втрачені або збережені не в потрібному місці. Це саме ті види збоїв, які ReFS може виявити за допомогою контрольних сум. Коли система ReFS виявляє такий збій, вона зв'язується з просторами зберігання для читання всіх доступних копій даних і вибирає правильну на підставі перевірки контрольної суми. Потім система повідомляє простори зберігання про необхідність відновлення пошкоджених копій на

основі правильних копій. З погляду програми все це відбувається прозоро. Якщо ReFS працює без дзеркальних просторів зберігання, немає можливості автоматично виправити пошкодження. У цьому випадку система просто внесе події в журнал, зазначивши, що було виявлено пошкодження і, якщо це були файлові дані, читання буде неможливим. Контрольні суми завжди включені для метаданих ReFS, та за умови, що том розміщений на дзеркальному просторі зберігання, автоматичне виправлення теж завжди ввімкнено.

Надмірність зберігання даних

Для перевірки надмірності системи було встановлено Windows Server та скопійований на розділ ReFS розміром 580 ГБ. Розмір метаданих на порожній ФС становив близько 0,73 ГБ. При копіюванні встановленого Windows Server на розділ з ReFS надмірність зберігання даних файлів зросла з 0,1 % на NTFS майже до 30 % на ReFS. При цьому ще близько 10 % надмірності додалося за рахунок метаданих. У підсумку дані користувачів розміром 11ГБ (більше 70 тис. файлів) на NTFS з урахуванням метаданих зайняли 11,3 ГБ, тоді як на ReFS ті самі дані зайняли 16,2 ГБ; це означає, що надмірність зберігання даних на ReFS становить майже 50 %, але при невеликій кількості файлів великого розміру такого ефекту не спостерігається [1].

Швидкість роботи

Для пошуку атрибутів папки потрібно три операції читання блоків по 16 Кб. Для порівняння, на NTFS ця операція займе одне читання розміром 1–4 Кб. Для пошуку атрибутів файлу по папці і імені файлу в ReFS потрібно ті самі три операції читання. На NTFS вже потрібно 2 читання по 1 Кб або 3–4 читання (якщо запис про файл знаходиться в нерезидентному атрибуті «індекс»). У папках більшого розміру кількість читань NTFS зростає набагато швидше, ніж кількість читань, необхідних для ReFS.

Стратегія надійного оновлення диска

Надійне та ефективне оновлення диска є одним з найбільш важливих і складних аспектів проектування файлової системи. Для забезпечення узгодженості на диску NTFS спирається на журнал транзакцій. Цей підхід оновлює метадані на диску та використовує журнал на стороні, щоб враховувати зміни, які можна відкотити в разі помилок або збою. Однією з переваг цього підходу є збереження метаданих, що є вигідним для продуктивності при читанні [2].

Недоліки системи журналів полягають у тому, що записи можуть бути невпорядковані, і оновлення диска може пошкодити записані раніше метадані у разі збою живлення під час запису (проблема «обірваного запису»).

Для максимальної надійності та запобігання обірваних записів вибраний підхід «розміщення при записі», за якого метадані ніколи не оновлюються на місці, а записуються в іншому місці атомарним чином.

Транзакції будуються на основі підходу «розміщення при записі». Оскільки верхній рівень ReFS успадкований від NTFS, нова модель транзакцій використовує вже існуючу логічну схему відновлення після помилок, яка була перевірена та стабілізована у багатьох випусках [2].

У файловій системі ReFS метадані впорядковані для забезпечення можливості комбінування записами за допомогою меншої кількості операцій введення–виведення, що є оптимальним рішенням для дискового простору. Водночас зберігається належний ступінь безперервності читання. Тут використовується схема ієрархічного розташування.

Цілісні потоки

Цілісні потоки захищають вміст файлу від усіх видів пошкоджень даних. Ця характеристика має цінність для багатьох сценаріїв, але в деяких випадках вона непридатна. Наприклад, деякі програми використовують управління зберіганням файлів з певним сортуванням файлів на диску. Оскільки цілісні потоки перерозподіляють блоки кожен раз, компонування файлів для цих додатків занадто непередбачуване. Системи баз даних є яскравим прикладом. Як правило, такі програми самостійно ведуть облік контрольних сум вмісту файлів і мають можливість перевіряти та виправляти дані шляхом прямої взаємодії з інтерфейсами API Storage Spaces [2].

Бітовий розпад

Як було сказано вище, поєднання ReFS і просторів зберігання забезпечує високу міру стійкості даних при пошкодженнях диска та збоїв зберігання. Важче виявляти і виправляти втрати даних, які виникають через «бітовий розпад», коли невчасно виявлені пошкодження для частин диска, що рідко зчитуються, поступово розростаються. До моменту читання і виявлення пошкоджень вони можуть вже торкнутися копії, або дані можуть бути втрачені через інші збої. Щоб запобігти бітовому розпаду, була додана системна задача, яка періодично очищає всі метадані і дані цілісних потоків на томі ReFS, що знаходиться на дзеркальному просторі зберігання. У ході очищення всі надлишкові копії читаються і перевіряються на правильність за допомогою контрольних сум ReFS.

У разі розбіжності контрольних сум, копії з помилками виправляються за допомогою правильних копій.

Сумісність

Щодо сумісності із стеком зберігання даних ReFS так само взаємодіє, як будь-яка інша файлова система, з метою максимізувати сумісність. Система може легко використовувати шифрування BitLocker, списки контролю доступу для безпеки, журнал USN, повідомлення про зміни, символьні посилання, точки з'єднання, точки підключення, точки повторної обробки, знімки томів, файлові ідентифікатори та нежорсткі блоки.

Висновок

Отже, з розгляду нової файлової системи ОП Windows Server 2012 можна спостерігати початок орієнтованості на серверний сегмент, на системи віртуалізації, взаємодії із СУБД та сервера архівації, у якому швидкість та надійність роботи є головними. Розроблена «з нуля» файлова система ReFS була створена спеціально для нового покоління зберігання даних. Microsoft серйозно переглянула список функцій файлової системи, намагаючись залишити максимальну сумісність з основними функціями NTFS, при цьому було усунуто всі менш запитувані систему можливості. Розробляючи ReFS, інженери повинні були виконати такі завдання: дати системі можливість перевіряти й автоматично виправляти дані, максимізувати можливості масштабування, наділити систему можливістю безперебійної роботи за рахунок ізоляції проблемних ділянок та забезпечити роботу нової технології Storage Spaces. Основний недолік файлової системи — неефективний розподіл даних на диск. Проте ключовими характеристиками ReFS залишається забезпечення цілісності метаданих, використання цілісних потоків, копіювання під час запису, можливість обробки та запису великих розмірів файлів та каталогів.

ЛІТЕРАТУРА

1. *Файловая* система ReFS изнутри. По матеріалам SysDev Laboratories. — [Електронный ресурс]. — Режим доступа: http://rlab.ru/.

2. *Создание* нового поколения файловой системы для Windows: ReFS. Steven Sinofsky. — [Електронный ресурс]. — Режим доступа: http://blogs.msdn.com/.

Стаття надійшла до редакції 05.12.2013