

Безносик Юрій Олександрович, кандидат технічних наук, доцент, кафедра кібернетики хіміко-технологічних процесів, Національний технічний університет України «Київський політехнічний інститут», Україна, e-mail: yu_beznosyk@ukr.net.
Корінчук Катерина Олексіївна, аспірант, відділ теплофізичних процесів в котлах, Інститут технічної теплофізики НАНУ, Україна, e-mail: engecology@gmail.com.
Кіржнер Дмитро Авраамович, інженер, відділ теплофізичних процесів в котлах, Інститут технічної теплофізики НАНУ, Україна, e-mail: dimavto@nbi.com.ua.

Логвин Валерій Александрович, аспірант, кафедра кібернетики хіміко-технологічних процесів, Національний технічний університет України «Київський політехнічний інститут», Україна.

Безносик Юрій Александрович, кандидат технічних наук, доцент, кафедра кібернетики хіміко-технологічних процесів,

Національний технічний університет України «Київський політехнічний інститут», Україна.

Корінчук Катерина Алексеевна, аспірант, отдел теплофизических процессов в котлах, Институт технической теплофизики НАНУ, Украина.

Киржнер Дмитрий Авраамович, инженер, отдел теплофизических процессов в котлах, Институт технической теплофизики НАНУ, Украина.

Logvyn Valeriy, National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine, e-mail: logvyn_valerii@mail.ru.

Beznosyk Yuriy, National Technical University of Ukraine «Kyiv Polytechnic Institute», Ukraine, e-mail: yu_beznosyk@ukr.net.

Kotinchuk Kateryna, Institute of Engineering Thermophysics NASU, e-mail: engecology@gmail.com.

Kirzhner Dmytro, Institute of Engineering Thermophysics NASU, Ukraine, e-mail: dimavto@nbi.com.ua

УДК 004.416.6:004.031.4

Лященко А. О.

ВИКОРИСТАННЯ СИСТЕМ ДЕДУПЛІКАЦІЇ ДАНИХ

У статті розглядаються особливості використання систем дедуплікації даних на конкретному прикладі. Використана файлова система ZFS, що має вбудовані механізми дедуплікації, стискування та кризну перевірку цілісності даних. Реалізація системи керування контентом виконана за допомогою системи *ownCloud*.

Ключові слова: система дедуплікації, зберігання повторюваних даних.

1. Вступ

Проблема зберігання великих обсягів даних на сьогодні стає все актуальнішою. У зв'язку з технічним прогресом та бурхливим розвитком інформаційної галузі збільшуються об'єми інформаційних масивів, підтримка та обслуговування ЦОД для їх збереження вимагає чималих витрат. Водночас з цим збільшуються об'єми даних що повторюються (дублюються) і вилучення таких надлишкових даних може суттєво зменшити витрати на обслуговування.

2. Постановка задачі

Існують кілька типів дедуплікації даних, це бітова, файлова та блочна дедуплікації. Робота будь-якого алгоритму дедуплікації зводиться до пошуку унікальних та повторюваних наборів даних, де другі замінюються посиланнями на перші. Інакше кажучи алгоритм намагається зберігати тільки унікальні дані, замінюючи повторювані дані посиланнями на унікальні. Таким чином досягається вилучення збитковості даних.

Раніше найбільш розповсюдженою була файлова дедуплікація [1]. Назву даного типу дедуплікації обумовив рівень її роботи — усі операції виконуються на рівні файлів. Якщо пригадати історію продуктів компанії Microsoft, то даний підхід раніше неодноразово використовувався у Microsoft Exchange Server та Microsoft System Center Data Protection Manager, цей механізм мав назву S.I.S. (Single Instance Storage) [1]. В продуктах сімейства Exchange від нього відмовились із міркувань швидкодії, але у Data Protection Manager

його продовжують використовувати [2]. Оскільки файловий рівень найвищий у системах збереження даних, то ефект застосування файлової дедуплікації є мінімальний у порівнянні з іншими типами дедуплікації. Область застосування файлової дедуплікації в основному зводиться до зберігання архівних даних та організації корпоративних файлових сховищ.

Блочна дедуплікація має більш досконалий механізм, який працює на субфайловому рівні, а саме — на рівні блокових даних. Цей тип дедуплікації, як правило притаманний для промислових систем зберігання даних, також саме він застосовується у Windows Server 2012 [3]. Механізми блочної дедуплікації схожі на механізми файлової дедуплікації, але працюють вони на рівні блоків. Сфера застосування блочної дедуплікації більш широка і розповсюджується не тільки на архівні дані, але і на віртуалізовані середовища, що досить логічно особливо для VDI (Virtual Desktop Infrastructure). Якщо брати до уваги, що VDI — це набір повторюваних образів віртуальних машин, в яких є відмінності, то файлова дедуплікація для цих систем не ефективна, а блочна — цілком ефективно працює з цим типом даних.

Бітова дедуплікація — найглибший тип дедуплікації даних. Вона наділена найвищим ступенем ефективності, але при цьому потребує найбільше ресурсів. Використання систем бітової дедуплікації є економічно невигідним. За цих обставин на сьогоднішній день системи бітової дедуплікації мають лише теоретичні реалізації і не використовуються на практиці.

У даній статті розглядається блочна дедуплікація, яка на даний момент є найбільш ефективною.

3. Програмно-апаратна база для дедуплікації

Для реалізації системи дедуплікації зазвичай використовують багатоядерні системи з великим об'ємом оперативної пам'яті з технологією ECC (error-correcting code, код корекції помилок), а також спеціалізованим програмним забезпеченням для забезпечення функцій дедуплікації. На сьогоднішній день найпоширеніші такі програмні рішення: Sun Microsystems ZFS, механізми дедуплікації Microsoft Windows Server 2012 та програмно-апаратні комплекси фірми NetApp.

Sun Microsystems ZFS (Zettabyte File System) розроблена для операційної системи Solaris. Ця файлова система підтримує великі об'єми даних. Головна перевага ZFS — це її повний контроль над фізичними та логічними носіями. ZFS може забезпечити високу швидкість доступу, контроль цілісності, а також мінімізацію фрагментації даних, спираючись на інформацію розташування їх на дисках. Це дозволяє динамічно виділяти або звільняти дисковий простір на одному або кількох носіях для логічної файлової системи.

Ефективне використання дедуплікації потребує великих об'ємів ОЗУ. Рекомендований діапазон оперативної пам'яті від 1 Гб на кожен додатковий 1 Тб сховища даних до 2 Гб для кожного наступного терабайту (при розмірі блоку 128 Кб). Недостатність фізичної пам'яті або замалий кеш ZFS може спричинити надмірне використання віртуальної пам'яті. Для підвищення продуктивності для зберігання таблиць дедуплікації може використовуватись SSD [4]. Станом на квітень 2013 відомі проблеми з швидкодією, що виникають за умови малої кількості ОЗУ. Деякі виробники систем зберігання даних модифікують код ZFS і використовують у своїх продуктах для досягнення більших коефіцієнтів дедуплікації, наприклад Greenbyte вдалося стиснути 5 000 віртуальних машин у 4 ТБ [5].

Microsoft Windows Server 2012. Починаючи з версії Windows Server 2012 компанія Microsoft включила до списку можливостей своїх серверних ОС можливість використання механізму дедуплікації даних. Microsoft надає статистику використання (рис. 1) своєї системи дедуплікації для трьох найбільш розповсюджених сценаріїв використання:

- 1) сервера збірки білдів ПЗ — у Microsoft кожен день збирається велика кількість білдів різних програмних продуктів. Навіть незначна зміна в коді приводить до процесу зборки — отже створюється дуже багато дубльованих даних;
- 2) ресурси спільного доступу з дистрибутивами продуктів на реліз. У Microsoft для цього використовуються спеціальні сервери, де зберігаються всі версії та мовні редакції, це теж досить ефективний сценарій, в якому ефективність від застосування дедуплікації може сягати до 70 %;

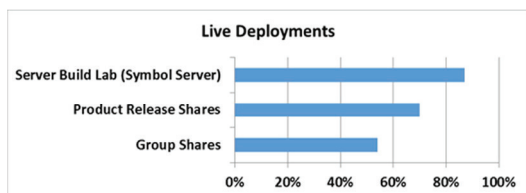


Рис. 1. Коефіцієнт дедуплікації даних для різних наборів даних [3]

3) групові ресурси спільного доступу — це поєднання документів та файлів розробників, а також їх пересувні профілі та перенаправлені директорії, що зберігаються у єдиному центральному просторі [3].

Дедуплікація в продуктах фірми NetApp. На відміну від ZFS та Windows Server 2012 продукти компанії NetApp використовують механізм «офлайн» дедуплікації. При використанні «офлайн» дедуплікації вона дає низку переваг:

- 1) можливість використання більш ефективних та точних (повільніших) алгоритмів виявлення дублікатів даних;
- 2) можливість аналізувати та обробляти значно більші обсяги даних, оскільки у випадку «офлайнової» дедуплікації для аналізу та використання доступний весь простір збереження, а не лише безпосередньо порція даних, що записується;
- 3) дедуплікація може виконуватись саме тоді і там, де це необхідно.

Таким чином зрозуміло, чого системи збереження даних від NetApp вибрали для використання саме «офлайн» спосіб, адже він дозволив робити дедуплікацію з мінімальним впливом на власну дискову швидкість системи.

На сьогоднішній день NetApp єдиний виробник систем зберігання, що використовує дедуплікацію і офіційно рекомендує використовувати для основних робочих даних, а не тільки для архівів.

Принцип функціонування систем NetApp базується на використанні «нестандартного розміру сектору» рівному 520, замість 512 байт (512 б даних + 8 б CRC) (рис. 2).

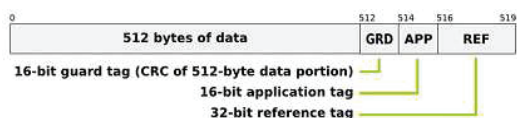


Рис. 2. Будова сектору в системах NetApp [6]

NetApp використовує додаткові 8 байт на сектор для організації свого механізму дедуплікації даних. Блок даних у WAFL займає 4096 байт. Блок даних, це те, що в файлових системах іноді називають «дисковим кластером», одна адресована порція даних. Цей блок складається з 8 секторів по 512 байт. У кожного з них є по 8 додаткових байт контрольної суми. Тобто у одному блоці WAFL зберігається 64 байти контрольної суми.

У контрольній сумі існує перевага — її досить швидко можна підрахувати. Але недоліком є вірогідність «hash-колізії», ситуації, коли два різних за змістом блоки мають однаковий результат хешу. Якщо орієнтуватись тільки на результати хешів, то можна визнати ідентичними два блоки різного змісту. Ця вірогідність не є великою, але існує.

Фоновий процес дедуплікації, складає базу хешів всіх блоків дискового тому, а потім формує список кандидатів на дедуплікацію. Над блоками з однаковим хешем система проводить операцію побайтового порівняння. Після цього решта ідентичних блоків замінюються вказівником inode. Цей механізм схожий на механізм посилань у UNIX файлових системах, але застосований до блоків даних файлової системи [6].

4. Налаштування систем дедуплікації даних на прикладі сервера з ОС Ubuntu Server

Продемонструємо використання технології дедуплікації на прикладі створення файлового сховища бази такої апаратної платформи: сервер на базі чотирьохядерного процесору Intel Core i7, 8 ГБ оперативної пам'яті та жорсткий диск ємністю 1,5 ТБ.

Оскільки файлова система ZFS реалізована тільки у операційній системі Solaris, яка у свою чергу є комерційним продуктом, в статті використано її портовану версію з відкритим вихідним кодом — ZFSOnLinux [7]. У даному прикладі використовувалась операційна система Ubuntu 12.04, хоч ZFSOnLinux також розроблений і для інших Linux систем. Для організації веб-доступу та керуванням змісту використана система ownCloud, що також є системою з відкритим вихідним кодом. На час написання цієї статті найновіша версія цієї системи була 5.0.5, вона ж використана і у прикладі [8].

Розглянемо налаштування програмного забезпечення для використання системи дедуплікації. Продемонструємо це на прикладі сервера з ОС Ubuntu Server 12.04 amd64 [9].

Перший етап — установка та налаштування файлової системи ZFS.

1. Спочатку у систему додаємо репозитарії програмних пакетів ZFSOnLinux за допомогою наступних команд:

```
sudo apt-get install python-software-properties
sudo add-apt-repository ppa:zfs-native/stable
sudo apt-get update
```

2. Далі встановлюємо програмний пакет ubuntu-zfs:

```
sudo apt-get install ubuntu-zfs
```

Для того щоб нові модулі ядра почали коректно працювати, перезавантажуємо ОС за допомогою такої команди:

```
sudo reboot
```

3. Створення дискового пулу для організації файлової системи ZFS пароводимо так:

```
sudo mkdir /pool1
sudo zpool create -m /pool1 pool1 /dev/sdb
```

4. Увімкнення механізму дедуплікації виконуємо командою:

```
sudo zfs set dedup=on pool1
```

Другим етапом є установка СКБД MySQL:

1. Напочатку проводимо встановлення програмних пакетів:

```
sudo apt-get install mysql-server
```

2. Далі — встановлення паролю для користувача root згідно інструкцій на екрані.

Третім етапом є установка та налаштування HTTP сервера Apache з підтримкою інтерпретатора PHP [10]:

1. Проводимо встановлення програмних пакетів:

```
sudo apt-get install apache2 libapache2-mod-php5
php5-mysql php5-gd
```

2. Необхідно також налаштувати каталог зберігання файлів веб-серверу:

```
sudo service apache2 stop
sudo mkdir /pool1/www
sudo rm -rf /var/www
sudo chown -R www-data:www-data /var/www
```

Четвертим етапом є розгортання та налаштування системи ownCloud:

1. Проводимо завантаження програмного пакету:

```
wget http://download.owncloud.org/community/
owncloud-5.0.5.tar.bz2
```

2. Розгортання файлів системи:

```
tar jfx owncloud-5.0.5.tar.bz2
sudo cp -rf ./owncloud /var/www
sudo chown -R www-data:www-data /var/www/
owncloud
```

3. У браузері користувача переходимо на веб-сторінку:

```
http://<адреса серверу>/owncloud
```

4. Вказуємо у полях необхідні дані.

В результаті наведеної низки налаштувань отримаємо багатокористувацьку систему, що має підтримку версій файлів та забезпечує автоматичну дедуплікацію даних.

5. Висновки

У статті розглянуто можливість створення систем зберігання даних з дедуплікацією на базі вже існуючих технологій. Однак слід зазначити, що ці технології ще не набули широкого застосування у корпоративному секторі у зв'язку з їх недавньою появою і можливою нестабільністю. Без детального відпрацювання використання розглянутих технологій може призвести до пошкодження або втрати даних. Але важлива сама можливість створення таких систем з відносно невеликими витратами, що відкриває перед розробниками та користувачами чималі перспективи.

Література

1. Microsoft Windows Server — Single Instance Storage (SIS) [Електронний ресурс]. — Режим доступу: \www/ URL: [http://technet.microsoft.com/en-us/library/gg232683\(v=ws.10\).aspx](http://technet.microsoft.com/en-us/library/gg232683(v=ws.10).aspx) — 12.05.2013 р. — Заголовок з екрану.
2. Microsoft System Center — What's new in System Center 2012 — Data Protection Manager [Електронний ресурс]. — Режим доступу: \www/ URL: <http://technet.microsoft.com/en-us/library/hh848299.aspx> — 12.05.2013 р. — Заголовок з екрану.
3. The Storage Team at Microsoft — File Cabinet Blog [Електронний ресурс]. — Режим доступу: \www/ URL: <http://blogs.technet.com/b/filecab/archive/2012/05/21/introduction-to-data-deduplication-in-windows-server-2012.aspx> — 12.05.2013 р. — Заголовок з екрану.

4. Уинзор, Дж. Solaris. Руководство системного администратора. [Текст] : пер. с англ./ Дж. Уинзор. — 3-издание. — Питер, 2003. — 444 с.
5. The Register — GreenBytes brandishes full-fat clone VDI rumpster [Електронний ресурс]. — Режим доступу: \www/ URL: http://www.theregister.co.uk/2012/10/12/greenbytes_chairman — 14.05.2013 р. — Заголовок з екрану.
6. Habrahabr — Дедуликация данных — подход NetApp [Електронний ресурс]. — Режим доступу: \www/ URL: <http://habrahabr.ru/company/netapp/blog/110482/>. — 14.05.2013 р. — Заголовок з екрану.
7. «Native ZFS for Linux» team — ZFS Stable Releases for Ubuntu [Електронний ресурс]. — Режим доступу: \www/ URL: <https://launchpad.net/~zfs-native/+archive/stable> — 12.05.2013 р. — Заголовок з екрану.
8. ownCloud — About ownCloud [Електронний ресурс]. — Режим доступу: \www/ URL: <http://owncloud.org/about/> — 14.05.2013 р. — Заголовок з екрану.
9. Ubuntu — Ubuntu Server — for scale out computing [Електронний ресурс]. — Режим доступу: \www/ URL: <http://www.ubuntu.com/server> — 14.05.2013 р. — Заголовок з екрану.
10. How-To Geek — Installing PHP5 and Apache on Ubuntu [Електронний ресурс]. — Режим доступу: \www/ URL: <http://www.howtogeek.com/howto/ubuntu/installing-php5-and-apache-on-ubuntu/> — 12.05.2013 р. — Заголовок з екрану.

ИСПОЛЬЗОВАНИЕ СИСТЕМ ДЕДУЛИКАЦИИ ДАННЫХ

В статье рассматриваются особенности использования систем дедуликации данных на конкретном примере. Использована файловая система ZFS, которая имеет встроенные механизмы дедуликации, сжатия и сквозную проверку целостности данных. Реализация системы управления контентом выполнена с помощью системы ownCloud.

Ключевые слова: система дедуликации, сохранение повторяющихся данных.

Лященко Андрій Олександрович, кафедра інформаційних систем та медичних технологій, Черкаський національний університет імені Богдана Хмельницького, Україна, e-mail: as.2mcl@gmail.com.

Лященко Андрей Александрович, кафедра информационных систем и медицинских технологий, Черкасский национальный университет имени Богдана Хмельницкого, Украина.

Lyashchenko Andrey, Cherkasy National University named after Bogdan Khmelnytsky, Ukraine, e-mail: as.2mcl@gmail.com

УДК 681.12

**Білінський Й. Й.,
Юкиш С. В.,
Сухоцька І. В.**

ЗАСІБ ВИЗНАЧЕННЯ КУТОВИХ ПАРАМЕТРІВ НАКОНЕЧНИКІВ

Представлено засіб визначення кутових параметрів наконечників, який оснований на виділенні контурів та виконанні автоматичного визначення порогу бінаризації за методом Оцу, що дозволяє вирівняти інтенсивність в контурах об'єктів зображення, зменшити шумові складові та підвищити точність за рахунок високого ступеня автоматизації процесу.

Ключові слова: контур, наконечник, виділення контуру, кутові параметри.

1. Вступ

На сьогодні різець є одним з найпоширеніших металоріжучих інструментів. При виготовленні різця враховують цілий ряд вимог, що забезпечують точність обробки і високу продуктивність його роботи, а саме матеріал ріжучої частини різця її геометрія, міцність і вібростійкість ріжучих кромок, форма і розміри платівки інструментального матеріалу, розміри, шорсткість, геометрія і конструкція гнізда для кріплення пластилини інструментального матеріалу та кутів параметри різця [1–5].

Важливим параметром різців є кут заточки, оскільки він визначає стійкість інструменту до зношування та впливає на точність обробки виробу. Тому важливою задачею є його вимірювання та контроль в процесі виробництва.

2. Аналіз літературних даних і постановка проблеми

З метою спрощення процесу вимірювання мікроскопічних об'єктів створюється програмне забезпечення, яке пропонує широкий спектр можливостей роботи

з цифровими знімками та дозволяє розраховувати геометричні та лінійні параметри [6, 7].

На сьогодні вимірювання мікроскопічних структур здійснюється шляхом використання методів візуалізації геометричних і фотометричних параметрів зображень об'єктів. При цьому розміри зображень попередньо корегуються відповідно масштабу об'єкта-мікрометра. Недоліками такого підходу є наявність похибки, яка виникає в результаті постобробки і яка не завжди задовольняє поставленим вимогам.

Для точного визначення геометричних параметрів об'єктів шляхом візуалізації необхідно виконати процедуру локалізації країв об'єктів, тобто виконати оконтурювання його. Проте контури, отримані існуючими методами, мають розриви, наявність ліній, що не відповідають досліджуваному об'єкту, при цьому контурні лінії є широкими, розмитими й нечіткими, що унеможлиблює їх розпізнавання [8–12]. Описані недоліки суттєво впливають на точність визначення геометричних параметрів різців.

Також на точність визначення геометричних параметрів об'єктів впливають методи вимірювання. На сьогодні відомі контактні (ручні) та безконтактні методи [8]. При ручному вимірюванні кути різця знаходять