Engineer

RAID

☰ кілька дисків, що працюють як один, переважно для надійності

Що це таке

RAID об’єднує кілька дисків в один блоковий пристрій, щоб група була швидшою, більшою або живучішою, ніж будь-який її окремий диск. Назва походить із технічного звіту Каліфорнійського університету в Берклі 1987 року, який написали Девід Паттерсон, Гарт Гібсон і Ренді Кац: *A Case for Redundant Arrays of Inexpensive Disks (RAID)*. Linux пише рівні малими літерами й без пробілу, raid1 чи raid5, — саме так їх показує драйвер md.

Звіт починає з розриву. Процесори й пам’ять швидшали, і анотація застерігає, що цей виграш пропаде, якщо введення-виведення не встигатиме за ними. Звіт протиставляє Single Large Expensive Disk, SLED, чия місткість швидко зростала, а продуктивність покращувалася лише помірно, масивам із дешевих дисків, зроблених для персональних комп’ютерів, і очікує, що такі масиви приблизно вдесятеро переважать великий диск за швидкістю, надійністю, споживанням енергії та запасом для зростання. Порівняння цілком конкретне: IBM 3380 для мейнфреймів, Fujitsu Super Eagle для мінікомп’ютерів і Conner CP 3100 на 100 МБ для персональних комп’ютерів.

Далі — заковика, у розділі, який автори самі називають поганими новинами. Якщо диски відмовляють незалежно один від одного і з постійною інтенсивністю, а звіт зауважує, що саме це припускають виробники, коли вказують середній час до відмови (MTTF), то MTTF масиву дорівнює MTTF одного диска, поділеному на кількість дисків. Сто дисків CP 3100, кожен із заявленими 30 000 годин, дають 300 годин, тобто менше двох тижнів; тисяча дає 30 годин. Звіт робить прямий висновок: великі масиви дешевих дисків без відмовостійкості надто ненадійні, щоб від них була користь.

Відповідь — надлишковість. Диски ділять на групи надійності, і в кожній є додаткові контрольні диски з надлишковою інформацією. Коли диск відмовляє, його замінюють і відтворюють його вміст на новому диску з цієї інформації; час, який на це йде, — середній час відновлення (MTTR), а гарячі резервні диски (hot standby spares), що вмикаються електронно, щойно диск помирає, його скорочують. Потім звіт описує п’ять способів організувати диски й називає кожен рівнем. Рівень 1 дзеркалить кожен диск. Рівень 2 виправляє помилки кодом Геммінга, як це робили системи пам’яті. Рівню 3 досить одного контрольного диска на групу. Рівень 4 дає змогу дрібним операціям читання й запису йти на різні диски одночасно. Рівень 5 розподіляє дані й контрольну інформацію по всіх дисках, щоб жоден окремий контрольний диск не гальмував запис.

Практика доповнила цей перелік. mdadm, інструмент для керування програмним RAID у Linux, приймає, серед інших, linear, raid0 (або stripe), raid1, raid4, raid5, raid6 і raid10. Розподілу даних без надлишковості (striping), який нині звуть рівнем 0, у класифікації 1987 року немає: вона починається з дзеркальних дисків. Звіт залишив відкритим питання, будувати масив апаратно чи програмно, і назвав його суто питанням витрат і вигоди. Linux md — програмний варіант; плата контролера, що показує операційній системі один великий диск, — апаратний.

Застосуйте поняття до рішення

Застосуйте «Що таке RAID?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Резервні копії, які перевірено, а не просто вважають робочими, і записаний план на той ранок, коли сервер не ввімкнеться.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Автоматизувала 100 критичних резервних копій — 100 критичних резервних копій автоматизовано в усіх базах

Автоматизувала 100 критичних резервних копій Зв'язатися з нами →

Чому це важливо

Диск зношується, і на сервері з одним диском його відмова зупиняє машину й залишає дані на милість тієї резервної копії, яка є. Розділ звіту про надійність починається саме з цього: диски настільки ненадійні, що системні адміністратори щоночі роблять резервну копію всього. З надлишковістю відмова диска стає ремонтом, а не аварією. Масив працює далі на дисках, що лишилися, у стані, який Linux називає degraded (деградований), і повертається до повної сили, коли заміну додано й відбудовано.

Цей захист триває, лише поки тримаються решта дисків. Розрахунки звіту вважають дані втраченими, коли в тій самій групі відмовляє ще один диск, перш ніж відремонтовано перший, тож тривалість вікна ремонту важить не менше, ніж кількість контрольних дисків. З гарячими резервами й коротким ремонтом цифри вражають: за підрахунком звіту, масив рівня 5 із 1 000 дисків, у групах по десять і з 5 відсотками резервних дисків, і далі мав би MTTF у 90 років. В іншому місці звіт попереджає, що землетрус або стрибок напруги можуть змусити диски одного масиву відмовити разом, а його формули виходять із того, що такого не стається.

Вибір рівня — це компроміс між місткістю, швидкістю й ціною. Власний висновок звіту такий: дзеркальні диски пасують обробці транзакцій, яка використовує не більше половини сховища, а рівень 5 виглядає найкраще, коли потрібна більша частка місткості, для роботи суперкомп'ютерів або для поєднання такої роботи з обробкою транзакцій.

Коли надлишковість вичерпано, лишається відновлення даних. Сторінка портфоліо про відновлення даних з SD-карт, HDD, SSD і RAID розглядає зруйнований масив RAID як окреме завдання, відмінне від мертвого SSD чи несправного диска з пластинами, і підхід там залежить від носія.

Як це виглядає на практиці

Візьмімо сервер, у якого на кожному з двох дисків є по розділу, `/dev/sda1` і `/dev/sdb1`, і їх треба віддзеркалити під том для даних. З Linux md три команди будують масив, створюють на ньому файлову систему й записують його в конфігурацію:

```
mdadm --create /dev/md0 --level=1 --raid-devices=2 /dev/sda1 /dev/sdb1
mke2fs -t ext4 -L data /dev/md0
mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Перша створює `/dev/md0`, масив RAID1 із двох розділів, і ядро запускає ресинхронізацію (resync), що перераховує надлишкову інформацію, як і після будь-якого створення. Друга форматує масив точнісінько як окремий диск; файлова система бачить один пристрій. Третя дописує рядок з описом масиву до конфігураційного файлу mdadm, який лежить у `/etc/mdadm.conf` або `/etc/mdadm/mdadm.conf`, а посібник просить переглянути цей вивід перед використанням, бо в ньому можуть опинитися зайві подробиці.

Стан масиву показують три файли:

```
cat /proc/mdstat
cat /sys/block/md0/md/degraded
cat /sys/block/md0/md/sync_action
```

`/proc/mdstat` перелічує всі активні пристрої md з їхнім станом, і mdadm читає його, щоб знаходити масиви й стежити за відбудовою. Кожен масив має ще й каталог під `/sys/block/`, і в ньому файл `degraded` містить кількість відсутніх учасників: 0 для цілого масиву, 1, коли одного диска немає. `sync_action` показує `idle`, коли нічого не відбувається, `resync` після створення або некоректного вимкнення, `recover`, поки гарячий резерв переймає вміст утраченого учасника, і `check` чи `repair` під час повної перевірки. Якщо записати в нього `check`, почнеться ця перевірка, що читає кожен блок масиву:

```
echo check > /sys/block/md0/md/sync_action
cat /sys/block/md0/md/mismatch_cnt
```

Після неї `mismatch_cnt` показує кількість секторів, які було перезаписано або було б перезаписано. Документація ядра зауважує, що більшість рівнів працює зі сторінками, а не секторами, тож це число може бути більшим за кількість справжніх помилок.

Про відмову ще має хтось дізнатися. `mdadm --monitor` періодично опитує масиви й перетворює помічене на події, які може надіслати поштою на адресу або передати програмі. Для системи під керуванням systemd посібник називає основним шляхом `mdmonitor.service`, а для пошти — параметр `MAILADDR` у `mdadm.conf`. Пошту надсилають лише п’ять подій: `Fail`, `FailSpare`, `DegradedArray`, `SparesMissing` і `TestMessage`. Остання існує, щоб довести, що канал працює:

```
MAILADDR ops@example.com
```

```
mdadm --monitor --scan --oneshot --test
mdadm --detail --test /dev/md0
```

Перша команда надсилає тестове повідомлення для кожного знайденого масиву й завершується. Друга згодиться для скрипта: з `--test` її код виходу 0 для справного масиву, 1, коли відмовив щонайменше один пристрій, 2, коли відмовило стільки, що масив непридатний, і 4, коли mdadm натрапив на помилку, збираючи відомості про масив.

Масив захищає лише від однієї події — виходу диска з ладу. Сторінка портфоліо про автоматизацію резервного копіювання та тестового відновлення описує окрему схему відновлення: надсилання зашифрованих знімків restic із дедуплікацією до сховища іншого провайдера через SFTP, таймери systemd та щомісячний процес, який відновлює відомий файл у тимчасове місце й порівнює його байти. Це передбачена схемою перевірка відновлення, а не доказ того, що її вже виконували. У записі про реалізацію від 2026-10-27 зазначено, що таймери були вимкнені за відсутності налаштованого репозиторію; жодного завершеного реального відновлення не зафіксовано.

Де виникають проблеми

Масив будують під одну подію, відмову одного з його дисків, і більшість способів утратити дані лежить зовсім поруч із нею.

Масив сприймають як резервну копію. Дзеркало записує кожну зміну на обидва диски одночасно, зокрема й випадково видалений файл, таблицю, знищену невдалою міграцією, чи том, зашифрований програмою-вимагачем. Тоді кожен учасник несе ту саму шкоду. Вихід — окрема резервна копія поза хостом і відновлення з неї, яке перевіряють за розкладом, щоб масив захищав від збоїв обладнання, а резервна копія — від усього іншого.

Деградований масив ніхто не помічає. Деградований масив і далі віддає дані, і ніщо в ньому не виглядає інакше, аж поки не відмовить наступний диск. mdadm не надсилає пошти, якщо в нього немає ні адреси, ні програми: з `--scan` без жодної з них, каже посібник, він нічого не відстежує. До того ж `DegradedArray` стосується лише масивів, які вже деградовані, коли mdadm бачить їх уперше; диск, що помирає пізніше, породжує подію `Fail`. Вихід — задати `MAILADDR`, переконатися, що `mdmonitor.service` працює, і надіслати повідомлення з `--test`, щоб довести, що пошта доходить.

Другий диск відмовляє під час відбудови. Відбудова відтворює відсутній диск з усіх інших, і модель звіту втрачає дані саме тоді, коли ще один диск групи відмовляє в межах часу ремонту. Мертвий диск, який лишили на місці на тиждень, розтягує вікно до тижня. Вихід — скоротити це вікно: тримати в масиві гарячий резерв, який mdadm може додати ще під час створення через `--spare-devices`, швидко міняти мертві диски й запускати `check` за розкладом, щоб диск із проблемами знайшовся, поки масив ще здатен їх виправити.

Диски не незалежні. Розрахунок MTTF припускає, що диски відмовляють кожен сам по собі, і звіт сам називає землетрус і стрибок напруги випадками, коли це не так. Два диски в одному корпусі ділять блок живлення, контролер, кімнату й будівлю. Вихід — сприймати розрахункову цифру як верхню межу, а від спільних причин захищатися чимось поза цим корпусом, наприклад копією в іншому місці.

Масив RAID 5 чи 6 водночас «брудний» і деградований. Якщо машина зупинилася некоректно, поки бракувало учасника, парності довіряти не можна, і деякі блоки не вдасться відбудувати, тож документація ядра попереджає про пошкодження, яке неможливо виявити, а md відмовляється запускати масив. Примусовий запуск через `mdadm --assemble --force`, а для кореневої файлової системи — через параметр модуля `start_dirty_degraded`, означає прийняти цей ризик. Вихід — заздалегідь вибрати політику узгодженості: md може записувати операції запису на окремий пристрій-журнал для рівнів 4, 5 і 6, а для рівня 5 пропонує partial parity log, який, за документацією, закриває так звану write hole. Якщо примус — єдиний спосіб дістатися даних, їх слід вважати сумнівними й звірити з резервною копією.

RAID 0 вважають захистом. md зараховує масиви raid0 і linear до тих, що не мають надлишкової інформації, а монітор mdadm позначає їх як `Wrong-Level`, бо вони не підтримують жодної з операцій відмови, резерву й ресинхронізації, за якими він стежить. Відсутнього учасника немає з чого відбудувати. Вихід — використовувати рівень 0 лише для даних, які можна викинути або відтворити, і ніколи не рахувати його за RAID у тому сенсі, який вкладав у це слово звіт 1987 року.

Апаратний контролер ховає диски. За платою контролера операційна система бачить один логічний пристрій, і перевірка стану, спрямована на нього, доходить до контролера, а не до накопичувачів. Наприклад, smartctl потребує `-d megaraid,N`, щоб дістатися диска N за контролером MegaRAID, і,

за його посібником, це охоплює й карти Dell PERC. Вихід — стежити за кожним фізичним диском через власний інтерфейс контролера, щоб учасника, який починає відмовляти, помітили, поки масив ще цілий.

Що почитати далі

- A Case for Redundant Arrays of Inexpensive Disks (RAID) — сторінка технічного звіту з Берклі з анотацією та PDF.
- UCB/CSD-87-391, повний текст звіту (PDF) — розрахунок MTTF, групи надійності й п’ять рівнів.
- RAID arrays, документація ядра Linux — файли md у `/sys`, «брудні» деградовані масиви й політики узгодженості.
- mdadm(8) — створення масивів, події монітора, пошта й коди виходу `--detail --test`.
- smartctl(8), smartmontools — доступ до дисків за RAID-контролерами через параметр `-d`.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке RAID?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/raid/