Streszczenie

Notatka opisuje kilka optymalnych algorytmów kompresji stosowanych do tworzenia archiwów. W szczególności pełniących rolę kopii zapasowej lub nośnika danych przekazywanych innym.

Spis treści

Motywacja

Najczęściej stosowane algorytmy kompresji danych w systemach z rodziny UN*X takie jak bzip2 i gzip ze względu na swój wiek stały się nieefektywnym narzędziem archiwizacyjnym. Wprowadzenie na początku lat 2000 nowocześniejszych metod kompresji opartych o algorytm LZMA umożliwiło drastyczne (nawet do 70%) zmniejszenie przestrzeni dyskowej koniecznej do przechowywania tej samej ilości danych. Niesie to ze sobą możliwości obniżenia zarówno kosztów przechowywania danych, jak i obniżenia wagi szkodliwych substancji, które emitowane są do środowiska podczas ich transferu. Według uśrednionych szacunków pochodzących z badań[1] [2] przesłanie 1MB danych generuje około 10g CO2.

Proponowane metody kompresji

LZIP

Lzip jest algorytmem dostępnym na zasadach Powszechnej Licencji Wolnego Oprogramowania GNU (GPL). Został wdrożony do powszechnego użytku przez dystrybucje systemu GNU/Linux takie jak GNU Guix i Dragora; organizację IANA, która wykorzystuje go do dystrybucji bazy informacji o strefach czasowych oraz Parlament Europejski, który z jego wykorzystaniem publikuje zrzuty swojej bazy danych w formacie JSON.

Podstawowa implementacja `lzip` nie posiada obsługi wielu wątków procesora, wydłużając tym samym czas kompresji i dekompresji przetwarzanych danych. W repozytoriach wielu dystrybucji udostępniono pakiet `plzip`, który pozwala na równoległą kompresję archiwów z wykorzystaniem wielu wątków procesora.

Wśród narzędzi konsumenckich algorytm ten jest obsługiwany przez narzędzie kompresji środowiska graficznego GNOME, Midnight Commander oraz zmodyfikowaną wersję archiwizatora 7-zip dostępną również na platformę MS Windows.

Przykłady zastosowania By skompresować dane za pomocą `lzip` stosujemy komendę:

```
tar cv katalog_z_danymi | lzip -c -f - > archiwum.tar.lz
```

Archiwum to możemy rozpakować za pomocą komendy:

```
tar xf archiwum.tar.lz
```

Stosując kompresję wielowątkową z `plzip` dobrze jest unikać mieszania z jednowątkowym `tar`, dlatego też kompresję wykonamy za pomocą komend:

```
tar cv katalog_z_danymi > archiwum.tar
plzip -c -f archiwum.tar > archiwum.tar.lz
```

a dekompresję wykonując:

```
plzip -d archiwum.tar.lz
tar xf archiwum.tar
```

Zstandard

Jest to algorytm opracowany przez pracowników Facebooka i opublikowany na zasadach licencji Otwartego Oprogramowania BSD. Wśród wdrażających go organizacji znajdziemy dystrybucję systemu GNU/Linux Archlinux. Jest on również rozważany jako domyślny format kompresji archiwów `*.deb` dystrybucji GNU/Debian i Ubuntu.

W repozytoriach dystrybucyjnych obecny jest jako pakiet `zstd`.

Przykład zastosowania W celu utworzenia archiwum z wykorzystaniem `zstd` wykonujemy komendę:

```
tar cv katalog_z_danymi | zstd -c -z -q - > archiwum.tar.zst
```

Skompresowane pakiety rozpakowujemy za pomocą:

```
tar -I zstd -xvf archiwum.tar.zst
```

Podobnie jak algorytm Lzip, obsługiwany jest przez narzędzie archiwizacyjne środowiska GNOME, zmodyfikowaną wersję 7-zip oraz bazy danych AWS Redshift i RocksDB.

7-zip

Narzędzie wprowadzone na początku lat 2000, które jako jedno z pierwszych stosowało algorytm LZMA. Jest ono powszechnie dostępne na platformach MS Windows (na zasadach licencji LGPL/BSD/unRAR), macos (od wersji 1.0 jest to oprogramowanie własnościowe) oraz GNU/Linux (LGPLv2). Stanowi to ogromną zaletę jeśli kompresujemy nasze dane, by przekazać je współpracownikom i podwykonawcom.

Przykład zastosowania By uzyskać siłę kompresji archiwum porównywalną z zaprezentowanymi wcześniej algorytmami, kompresję z wykorzystaniem 7zip przeprowadzimy za pomocą komendy:

```
tar cv katalog_z_danymi | 7z a -t7z -m0=lzma -mx=9 \
                           -mfb=64 -md=32m -ms=on \
                           -si archiwum.tar.7z
```

Archiwum dekompresujemy wywołując:

```
7z x archiwum.tar.7z
tar xf archiwum.tar
```

Porównanie skuteczności kompresji

Dla paczki logów osiągi wymienionych wcześniej komend prezentują się następująco:

```
473M    archiwum.tar
26M     archiwum.tar.gz
11M     archiwum.tar.lz
14M     archiwum.tar.zst
11M     archiwum.tar.7z
```

Dla zróżnicowanych plików użytkowników jednego z serwisów, zawierających dużo danych w postaci binarnej (pliki obrazów/PDF):

```
19G     files/
18G     files.tar
16G     files.tar.gz
12G     files.tar.lz
13G     files.tar.zst (najkrótszy czas kompresji)
11G     files.tar.7z
```

Kompresji do archiwum gzip dokonano z wykorzystaniem komendy:

```
tar cv katalog | gzip --best > archiwum.tar.gz
```

Pozostałe uwagi

Należy zauważyć, że ewentualne wdrożenie, opisywanych, algorytmów do kompresji logów odkładanych na serwerach może skutkować utratą kompatybilności z narzędziami takimi jak `zcat`, `bzcat` oraz `fgrep`, służącymi do szybkiego ich przeglądania i przeszukiwania z uwzględnieniem paczek po rotacji.