Rozproszone systemy plików wprowadzenie

Autor: Łukasz Polak

Spis treści

Co to jest rozproszony system plików?

System plików określa sposób, w jaki dane abstrakcyjne mają swoje odzwierciedlenie na fizycznym nośniku, którym może być np. dysk twardy, dyskietka, pamięć komputerowa. Składa się na to kilka elementów. Po pierwsze, sposób w jaki dane, są umieszczane na wybranym nośniku – które bloki mają zajmować, w jaki sposób odszukiwać wolne bloki, itp. Prócz tego określa jak mają być zapamiętywane różne informacje o tych plikach, czyli gdzie jest on zapisany, jak się do niego dostać, gdyż jest to potrzebne do spełnienia innych wymagań stawianych systemowi plików – musi umożliwiać możliwie szybki i łatwy dostęp do plików, manipulacji nimi oraz ich usuwania.

Systemy plików dzielą się na kilka rodzajów. Najpopularniejszy z nich to dyskowy system plików, czyli opisujący system plików dla np. twardych dysków. Prócz tego istnieje również kilka innych, jak np. poznany przez nas linuxowy /proc. Wśród nich istnieje także podtyp zwany rozproszonym systemem plików, którym będziemy się właśnie zajmować.

Rozproszone systemy plików można by uznać za protokoły, gdyż określają one sposób, w jaki użytkownicy mogą przesyłać polecenia do serwera przez sieć, wykonywać operacje na odległość oraz otrzymywać inne informacje z powrotem z serwera do klienta. Serwerów tych może być więcej niż jeden, oraz użytkowników komunikujących się z tymi serwerami może być oczywiście również więcej niż jeden. Co więcej – użytkownicy powinni móc współbieżnie móc korzystać z danego rozproszonego systemu plików. Jednak dzięki tzw. wymogowi przezroczystości użytkownik powinien odnieść wrażenie, iż korzysta ze zwykłego lokalnego systemu plików – nie powinien widzieć żadnych podziałów na serwery, ani tego że ktoś inny korzysta z danego pliku, a jedynie widzieć drzewo katalogów, wyglądające analogicznie do tych z dyskowych systemów plików. System, który spełnia te kilka podstawowych zasad może być uznany za rozproszony system plików.

Zalety

Oczywiście rozproszone systemy plików nie powstały bez potrzeby, a dlatego iż posiadają one szereg zalet, których nie potrafią zapewnić (lub w mniejszym stopniu) zwykłe systemy plików.

Sama możliwość dostępu do systemu przez wielu użytkowników jest sama w sobie ogromną zaletą. Dzięki temu nie potrzebujemy wielu dysków twardych, na wszystkich których jest zainstalowane te same oprogramowanie, lub przynajmniej na których część oprogramowania jest taka sama u kilku użytkowników. Prócz trudności podczas samej instalacji, zwiększa się znacznie koszt takiego rozwiązania. Dzięki takiemu dostępowi wystarczy że zainstalujemy jedną kopię oprogramowania na danym serwerze i już wszyscy użytkownicy rozproszonego systemu plików mogą korzystać z tej aplikacji.

Prócz współdzielenia aplikacji, sami użytkownicy mogą dzielić między sobą różne dokumenty, pliki. Eliminuje to problem kopiowania jednej kopii danego dokumentu i wysyłania jej do wszystkich zainteresowanych osób. Z rozproszonym systemem plików wystarczy wrzucić jedną kopię na serwer i już pozostali będą mieli do niej dostęp.

Dzięki scentralizowaniu przechowywanych danych można osiągnąć większa niezawodność, np. przez dołączenie nowego poziomu RAID dla serwerów, co przy kilku lokalnych systemach plików sprawiło by duże problemy. Scentralizowanie ułatwia też administrację – znacznie łatwiej jest administrować jednym czy kilkoma serwerami, niż całą siecią komputerów użytkowników.

Problemy

Tak jak ze wszystkim, tak i z rozproszonymi systemami plików nie jest tak bardzo wesoło i istnieje szereg problemów, z którymi muszą sobie poradzić programiści tworzący takie systemy.

Jak zapewnić żeby każdy klient widział takie same pliki i katalogi, przy jednoczesnym zachowaniu dobrej wydajności podczas operacji na systemie plików?

A jak powinien zachować się system w przypadku awarii jednego lub więcej serwerów? W jaki sposób zachować ciągły dostęp do plików przez użytkowników?

Jak walczyć z przeciążeniami sieci, które mogą skutecznie uniemożliwić lub znacznie spowolnić pracę użytkowników?

Co robić przy równoczesnej pracy kilku użytkowników nad jednym plikiem, a co gorsza – jednoczesnym zapisem do tego pliku?

I wreszcie jak stworzyć bezpieczny dostęp do plików przez różne grupy użytkowników tak, aby odpowiedni użytkownicy mogli korzystać jedynie z plików (aplikacji, dokumentów, itp.) do korzystania z których mają uprawnienia?

Cechy

Z tych wszystkich problemów, na które napotykają się twórcy rozproszonych systemów, można stworzyć pewną listę cech, jakie może, lub wręcz czasami musi posiadać, dobry rozproszony system plików.

Przezroczystość

Było już o niej wspomniane podczas definiowania rozproszonych systemów plików. W dużym skrócie, przezroczystość ma powodować, iż maskowane będzie rozproszenie, użytkownicy będą odnosić wrażenie iż korzystają ze zwykłego, dyskowego systemu plików. Jest wiele różnych rodzajów przezroczystości, które powinien posiadać dobry rozproszony system plików:

1.Dostępu – nie ma rozróżnienia pomiędzy dostępem do plików lokalnych a plików z systemu rozproszonego. Istnieje jeden zestaw operacji, których używa się zarówno do uzyskiwania dostępu dla plików z nośników w komputerze, jak i tych zdalnych.

2.Położenia – nazewnictwo plików, ścieżki, nie powinny nic mówić o rzeczywistym położeniu pliku (np. na jakim serwerze jest). Powinnie być jeden, ujednolicony sposób zapisu. W szczególności, w jednym katalogu w systemie plików rozproszonym mogą znajdować się pliki, które w rzeczywistości znajdują się na dwóch różnych serwerach fizycznych.

3.Wydajności – operacje na rozproszonym systemi plików powinny zachowywać pewną prędkość wykonania nawet w przypadku przeciążenia (do pewnego stopnia) sieci. Nie może zajść sytuacja w której użytkowanie kilku użytkowników na raz powoduje że jeden z nich ma trudności z wykonywaniem operacji na systemie plików.

4.Mobilności – zmiana lokalizacji danych nie powinna zmuszać od strony użytkowników żadnej modyfikacji posiadanych programów.

Równoległość

Jak było już wspominane, systemy rozproszone muszą umożliwiać równoległość, to znaczy możliwość jednoczesnej operacji na danym pliku przez kilku użytkowników. Ważne

jest, aby zmiany wprowadzane do pliku przez jednego użytkownika nie przeszkadzały innym w próbie dostępu do tego pliku. Jest to znany problem współbieżności. Najlepszym rozwiązaniem tego problemu jest wprowadzanie blokad na plik, w przypadku jego użytkowania.

Powielanie

Inaczej replikacja, czyli dopuszczenie sytuacji, w której dany jeden plik w systemie rozproszonym jest reprezentowany jako kilka kopii w różnych miejscach na rzeczywistych serwerach. Takie rozwiązanie powoduje, że zwiększa się odporność na błędy nośnika – gdy jedna kopie zostanie zamazana, zostają inne. Poza tym powoduje to, iż obciążenie może być rozrzucane na kilka serwerów, polepszając tym samym wydajność całego systemu. Oczywiście, takie rozwiązanie musi być odpowiednio przemyślane, gdyż nieodpowiednia implementacja może spowodować problemy ze spójnością – różne wersje pliku na różnych serwerach będą mięc inną zawartość.

Heterogeniczność

Zdefiniowanie interfejsów tak, że zarówno aplikacja kliencka jak i serwer mogą być zaimplementowane na dowolnym systemie operacyjnym, a także na dowolnym komputerze, architekturze, itp.

Odporność na awarie

Chodzi tutaj o awarie serwerów. Oczywiste jest, iż pomimo awarii jednego z nich cały system powinien nadal bez problemów działać i udostępniać użytkownikom pliki. Najoczywistszym rozwiązaniem problemu jest duplikacja – wtedy w przypadku awarii jednego z serwerów żądania do niego skierowane mogą zostać przekazane innym serwerom, posiadającym kopie plików znajdujących się na serwerze, który aktualnie padł ofiarą awarii

Spójność

Każdy plik w sieci w danej chwili, powinien mieć identyczną zawartość dla każdego klienta sieci. Widać, że głównym zagrożeniem tutaj są wszelkie zabiegi podczas których powielany jest plik – duplikacja, cache'owanie, itp. Podczas ich wykonywania powstaje kilka wersji tego samego pliku i wówczas należy uważać, aby podczas uaktualniania pliku przeprowadzić aktualizację również wszelkich jego fizycznych kopii dostępnych zarówno na serwerach, jak i użytkowników.

Bezpieczeństwo

Na tę cechę składają się wszystkie możliwe mechanizmy, które mają uchronić pliki z rozproszonego systemu przed dostępem osób, które nie powinny takiego dostępu uzyskać. Pierwsza kwestia, to ustalanie praw dostępu dla poszczególnych plików. System powinien dawać możliwość ustawienia, kto może odczytywać dany plik, kto zapisywać, uruchamiać, oraz egzekwować te zasady. Druga kwestia to szyfrowanie przesyłanych danych. Ponieważ rozproszone systemy plików operują przez sieć, toteż istnieje ryzyko przechwycenia danych przez niepowołane osoby, podczas gdy przesyłane są one od klienta do serwera, bądź odwrotnie. Aby zapobiec czemuś takiemu stosowane jest szyfrowanie.

Skalowalność

System powinien umożliwiać w miarę rozrostu w miarę bezbolesną możliwość przyłączenia do już istniejącego systemu nowych komputerów, nowego sprzętu, itp.

Warto zauważyć, że tak jak już w niektórych punktach to zaznaczyłem, pomiędzy wymaganiami istnieje ścisła sieć powiązań, która sprawia że wzrost jakości w jednej z wymaganych sfer powoduj spadek tejże w innej, np. zastosowanie jakiegoś systemu cache'owania spowoduje zwiększenie wydajności, ale może się to odbyć kosztem spójności

Historia

Początki rozproszonych systemów plików sięgają lat siedemdziesiątych. Już w roku 1973 zainstalowany na maszynie PDP-10 Datacomputer udostępniał podobny do FTP protokół, który umożliwiał komputerom nie posiadającym dużej przestrzeni dyskowej na korzystanie z jego dysku.

Dwa lata później, w instytucie Xerox PARC powstał IFS, czyli Interim File Server. umożliwiał on organizowanie publicznych i osobistych plików użytkowników w formie drzewa. Ten sam instytut kilka lat później stworzył Woodstock File Server. Można w nim było uzyskiwać dostęp do pojedynczych stron danego pliku, co umożliwiało tworzenie klientów, nie posiadających dysków i używających wirtualnej pamięci poprzez sieć.

Lata osiemdziesiąte to rozkwit zainteresowania rozproszonymi systemami. Powstało wtedy wiele systemów, jak np. XDFS, LOCUS, SWALLOW, ACORN. W latach tych były opracowywane również trzy popularne, używane nawet do dziś systemy, czyli NFS, AFS (AndrewFS) i CODA.

Rozwój rozproszonych systemów plików trwa do dziś, powstają nowe takie systemy jak GoogleFS czy też zFS.

Global File System

GFS był początkowo tworzony jako projekt potrzebny do pracy dyplomowej, pisanej na Uniwersytecie Minnesoty w 1997. Początkowo tworzony na system operacyjny IRIX, jednak szybko (1998) przeniesiono go na Linuxa. Na przełomie lat 1999 / 2000 pracą nad nim zajęła się firma Sistina Software, która początkowo udostępniała go na zasadzie Open-Source, jednak w 2001 roku uczyniła go komercyjnym produktem z zamkniętymi źródłami. Powstał wtedy odłam, bazowany na ostatniej dostępnej wersji kodu, zwanym OpenGFS.

W grudniu 2003 roku firma Red Hat wykupiła Sistinę i po około pół roku ponownie otworzyła źródła dla tego systemu. Od tego czasu rozwój skupił się na eliminowaniu błędów i stabilizacji.

Obecnie GFS jest częścią Linuxów Fedora i Centos, a także może być zakupiony dla Linuxa Red Hat.

Cechy

Brak podziału na klient serwer

W systemie GFS nie ma rozróżniania na serwery które współdzielą pliki i klientów z nich korzystających, każdy węzeł może równocześnie udostępniać pliki dla sieci jak i mieć dostęp do zawartości całej sieci.

Przezroczystość dostępu i położenia

GFS udostępnia dostęp do plików poprzez ujednolicone drzewo katalogów, bez rozróżniania poszczególnych klastrów.

Wydajność

Umożliwia tworzenie sieci używając technologii iSCSI i Fibre Channel, które dostarczają wysoką przepustowość (np. w technologii Fibre Channel jest to nawet 10 gigabitów.

Odporność na awarie

GFS umożliwia zwiększanie dostępności, poprzez łatwe tworzenie kopii każdego elementu systemu. Ponadto począwszy od wersji 4 jest dostępny mechanizm księgowania (journaling), które to znacznie zmniejsza ryzyko błędów w skutek awarii sprzętu.

Współbieżność i spójność

GFS umożliwia równoległy dostęp do każdego klastra, który jest dzielony w sieci. Jednocześnie używa on rozbudowanego systemu blokad, dzięki którym maszyny w sieci koordynują swoje operacje wejścia-wyjścia, w celu zachowania maksymalnej spójności danych. Są tutaj możliwe różne mechanizmy do wykorzystania, obecnie preferowany system to Distributed lock manager. GFS pilnuje także, aby zawsze zmiany wprowadzone przez jedną maszynę miały natychmiast swoje odzwierciedlenie na reszcie maszyn.

Skalowalność

System ten jest bardzo łatwo skalowalny, dołączenie nowego komputera to jedynie zamontowanie go do systemu plików. GFS posiada również ogromne możliwości co do ilości połączonych maszyn, których mogą być nawet setki.

Brak pełnej heterogeniczności

Niestety, poważną wadą GFSa jest jego nastawienie na systemy Linuxowe – jest on rozwijany tylko na nie i nie jest dostępny na inne systemy. Jest za to dostępny na wiele różnych architektur, był np. pierwszym systemem klastrowym działającym na architekturach 64 bitowych.

Informacje dotyczące innych implementacji

W drugiej części slajdów.