Zagadnienie równoważenia obciążeń w środowisku OpenSSI

Dokumentacja końcowa do projektu z przedmiotu RSO

Karol Rzońca
krzonca@elka.pw.edu.pl

16 czerwca 2005

Spis treści

1 Migracja

Aby w ogóle można było mówić o równoważeniu obciążenia, niezbędny jest mechanizm migracji procesów między węzłami. W OpenSSI migracja procesu polega na przeniesieniu procesu i jego kontekstu w całości na inny węzeł. W jądrze starego węzła nie zostają żadne informacje o migrującym procesie.

Taka realizacja możliwa jest poprzez nadawanie procesom identyfikatorów (pid) unikalnych w skali klastra, dzięki czemu proces nie zmienia pid'u przy przenoszeniu go na inny węzeł. System plików HA-CFS daje możliwość przezroczystego dostępu do plików. Dodatkowo, podobnie jak procesy, wszystkie obiekty IPC posiadają unikalne identyfikatory w skali klastra, przez co są dostępne z dowolnego węzła. Istnieje kilka ograniczeń technicznych i logicznych zakazów migracji, które nie pozwalają przenosić następujących procesów:

- proces init i wszystkie procesy jądra
- procesy, które zaczęły być niszczone
- procesy, które zostały stworzone przez vfork i nie wykonaly rexec
- procesy, które dzielą struktury mm
- procesy, które dzielą struktury fs

- procesy, które dzielą struktury pliku
- procesy, które dzielą struktury sygnałów
- procesy systemowe
- procesy, które posiadają zablokowane strony w pamięci
- procesy, które śpią na semaforze

W celu zrównoważenia obciążenia migrować mogą tylko te procesy, które zostały oznaczone przez umieszczenie jedynki w pliku /proc/¡pid¿/loadlevel. Zero w tym pliku zakazuje wybierania procesu przez algorytm równoważenia obciążenia, czyli nie podlega on równoważeniu. Jedynka ta może się pojawić automatycznie, jeśli program umieszczony jest na specjalnej liście znajdującej się w pliku /proc/cluste/loadleverootl (lista ta jest inicjowana plikiem /cluster/etc/loadlevel podczas uruchamiania usługi loadlevel). Pojawia się ona również automatycznie, gdy proces jest tworzony przez proces, który ma wpisana jedynkę, następuje jej dziedziczenie w momencie tworzenia procesu. Istnieje też możliwość ręcznego wstawienia jedynki lub zera za pomocą komendy loadlevel.

Ponadto w pliku /proc/¡pid¿/pin zaznacza się, czy proces jest przypięty do węzła, na którym jest uruchomiony. Jedynka w tym pliku oznacza, że proces, niezależnie od innych ustawień, nie podlega równoważeniu. Jeśli włączone będzie równoważenie dla procesu przypiętego, to jego dzieci również będą miały włączone równoważenie, a ponieważ przypięcie nie jest dziedziczone, to jego dzieci nie będą przypięte i będą równoważone.

2 Obliczanie obciążenia

OpenSSI podczas wyliczania obciążenia węzłów korzysta z wyników prac projektu MOSIX. Zapożyczone zostały tylko algorytmy wyliczania obciążenia węzłów, wybierania procesów do migracji i wybierania miejsca przeznaczenia migrującego procesu. Pozostałe czynności potrzebne do realizacji równoważenia obciążenia zostały zaimplementowane od nowa z uwagi na inny charakter OpenSSI.

Podstawowym problemem, z jakim wszystkie algorytmy równoważenia obciążenia muszą sobie poradzić jest fakt, iż dostępne zasoby takie jak pamięć, zużycie procesora, komunikacja międzyprocesowa i wiele innych branych pod uwagę , są praktycznie nieporównywalne. Przez to wyznaczenie optymalnego węzła docelowego jest zadaniem bardzo skomplikowanym. W projekcie MOSIX, na podstawie zużycia procesora, wolnej pamięci, liczby operacji I/O i wykorzystania IPC, oblicza się dla każdego procesu wartość obciążenia danego węzła przez ten proces. Obciążenie wyrażone jest w jednej wspólnej jednostce, co pozwala w łatwy sposób stwierdzić, czy węzeł jest bardziej obciążony od innych, czy mniej.

W MOSIX, oprócz normalnego algorytmu wyznaczania obciążenia węzła, istnieje jeszcze priorytetowy algorytm, uaktywniający się w momencie, gdy na danym węźle następuje zbyt duża wymiana stron pamięci (na węźle brakuje pamięci operacyjnej). Gdy zostanie wykryta taka sytuacja, wybierany jest proces i następuje próba jego migracji na węzeł z wystarczającą ilością pamięci. Dzieje się tak nawet, jeśli migracja miałaby doprowadzić do nierównomiernego obciążenia. Do wyznaczenia węzła docelowego podczas migracji i stwierdzenia, czy węzeł jest obciążony na tle innych,

potrzebny jest stan całego klastra. Dzielenie się informacjami miedzy węzłami odbywa się poprzez jeden wyznaczony węzeł zwany master , który odpytuje wszystkie węzły i zbiera tylko dynamiczne informacje, takie jak obciążenie i wolna pamięć. Następnie węzeł master rozsyła te informacje do wszystkich węzłów. Informacje statyczne, takie jak ilość procesorów, ich prędkości i całkowita pamięć, rozsyłane są podczas podłączenia węzła do klastra. Przy wyznaczaniu węzła docelowego brany jest również pod uwagę czas potrzebny do zakończenia procesu, rozmiar procesu oraz ilość operacji wejścia/wyjścia.

3 Równoważenie przychodzących połączeń sieciowych

Bardzo dużą zaletą OpenSSI jest fakt, że potrafi wystawić na świat interfejs sieciowy, który jest przez każdy węzeł widziany w ten sam przezroczysty sposób, dlatego przychodzące połączenia na ten interfejs mogą obsługiwać dowolne węzły klastra. Przy specyfikacji takiego interfejsu sieciowego można podać algorytm wybierania węzła obsługującego przychodzące połączenie. Następujące algorytmy szeregowania przychodzących połączeń sieciowych są zaimplementowane w jądrze: Robin Robin(rr), Weighted Round Robin(wrr), Least Connection(lc), Weighted Least-Connection(wlc), Locality-Based Least-Connection(lblc), Locality-Based Least-Connection with Replication(lblcr), Destination Hashing(dh), Source Hashing(sh), Source Hashing(sed), Never Queue(nq). Ich opisy można znaleźć w manualu dla ipvsadm(8).

4 Konfiguracja i użytkowanie

W pliku /cluster/etc/loadlevellist znajduje się lista programów, które po uruchomieniu mają automatycznie włączone równoważenie. Wartości z tego pliku przy uruchomieniu usługi równoważenia obciążenia ładowane są do pliku /proc/cluste/loadlevellist. Algorytmem równoważenia obciążenia można sterować za pomocą komendy loadlevel, która uruchamiana przez root'a pozwala włączać i wyłączać równoważenie obciążenia na wybranych węzłach. Za pomocą tej samej komendy można włączać równoważenie dla wybranych procesów podając ich PID lub nazwę. Polecenie loads pozwala sprawdzić obciążenie węzłów. Komenda migrate służy do ręcznej próby migracji procesu lub procesów na wybrany węzeł. Należy pamiętać o opisanych wcześniej ograniczeniach migracji procesów. Możliwe jest także konfigurowanie równoważenia obciążeń sieciowych przez odpowiednie wpisy w pliku /etc/cvip.conf w znacznikach ¡scheduler¿¡/scheduler¿.

5 Testowanie równoważenia obciążenia

Testowane będzie równoważenie obciążenia na problemach w których procesy komunikują się ze sobą za pomocą mechanizmów IPC. Programem, a właściwie zestawem programów, dostarczonych wraz z OpenSSI. Są to następujące programy:

- demo-proclb - program uruchamiający test, uruchamia procesy, przydziela im fragment zasobu, którym jest podany w argumencie plik, i odbiera od tych procesów, za pomocą kolejki wiadomości IPC, informacje o stanie przetworzenia zasobu

- demo-proclb-child - program uruchamiany przez demo-proclb jako pod proces dziecko, proces ten czyta kawałek po kawałku przydzielony zasób i zgłasza przez kolejki wiadomości IPC
- demo-proclb-monitor - program monitorujący prace programu testującego, proces czyta z pliku logów programu testującego, numery procesów i wyświetla na którym węźle klastra znajduje znajdują się procesy, wyświetla również informacje /proc/loadavg dla każdego węzła (średnie obciążenie węzła)

Charakterystyka tego problemu testowego polega na tym iż istnieje jeden proces z któremu pozostałe procesy zdają raport ze swojej pracy. Wszystkie procesy dzieci pracują na tym samym pliku (zasobie), każdy na przydzielonym tylko niemu fragmencie tego pliku. Procesy dzieci czytają fragment pliku, symulują prace nad czymś przez wykonanie pustej pętli określoną ilość razy, następnie informują program testujący o wykonanym fragmencie pracy po czym czytają kolejny kawałek pliku i tak aż skończą przetwarzać cały przydzielony im kawałek.

Proces testowania polega na uruchomieniu programu testującego z różną liczbą procesów, z włączonym równoważeniem obciążenia i z wyłączonym równoważeniem obciążenia, i mierzeniu czasu wykonania testu. Dodatkowo na innej konsoli można oglądać rozkład procesów na poszczególnych węzłach jak i ich obciążenie.

Podczas obserwacji testów można łatwo zauważyć iż bez włączonego równoważenia obciążenia, testy wykonują się tyle razy wolniej ile jest węzłów w klastrze. Z obserwacji wynika również że procesy przemieszczają się z węzła najbardziej obciążonego i najmniej wydajnego na węzły szybsze i mniej obciążone. Węzły z szybszymi procesorami, wykonują więcej testowych procesów.

Drugim testem było uruchomienie programu symulującego ruch ciał w dwuwymiarowej przestrzeni grawitacyjnej. Program ten ma w celu przyspieszenia obliczeń dzieli zadaną przestrzeń na NxN podprzestrzeni, następnie tworzy NxN procesów i przydziela każdej podprzestrzeni dokładnie jeden proces obsługujący. Każdy proces oblicza ruchy ciał które znajdują się mu przydzielonym fragmencie przestrzeni, gdy ciało opuszcza jego fragment, zostaje przekazane prze z pamięć dzieloną IPC do odpowiedniego procesu. Dodatkowo każda iteracja obliczeń, synchronizowana jest przez jeden proces synchronizujący, przez co praca wszystkich procesów uzależniona jest od pracy pozostałych i występuje tu bardzo częsta komunikacja z procesem synchronizującym. Podczas obserwacji pracy tego programu, z włączonym równoważeniem obciążenia nie można zaobserwować żadnych zmian w stosunku do pracy bez włączonego równoważenie obciążenia. Wynika z tego iż proces równoważenie obciążenia stwierdza iż nie opłacalne jest migrowanie procesów bardzo często komunikujących się ze sobą.

Podczas obu testów, bardzo rzadko, z włączonym równoważeniem obciążenia, zdarzała się sytuacja w której operacja podniesienia semafora zwracała błąd wykonania, co skutkuje tym iż w swoich programach należy uwzględniać fakt iż czasem operacja semaforowa może się nie udać i należy na ten fakt odpowiednio reagować.