

Jerzy KĘDZIERA

Instytut Informatyki Teoretycznej i Stosowanej, PAN

ANALIZA WYDAJNOŚCI KOMUNIKACJI W ŚRODOWISKU SIECIOWYM NA PRZYKŁADZIE RPC

Streszczenie. Mechanizm zdalnego wywoływania procedur stosowany jest jako jeden z podstawowych mechanizmów wymiany danych w systemach przetwarzania rozproszonego. Jego użyteczną cechą jest przezroczystość, maskująca szczegóły implementacji rozwiązań komunikacyjnych. Analiza wydajności mechanizmu RPC daje podstawy doboru parametrów komunikacji dla efektywnej konstrukcji konkretnych zastosowań sieciowych.

ANALYSIS OF COMMUNICATION PERFORMANCE IN NETWORK ENVIRONMENT REPRESENTED BY RPC.

Summary. Remote procedure call interface is used in distributed computing environments as one of the basic data exchange engines. Its useful feature is transparency which allows to generate calls to a standard interface without being concerned with the underlying network implementation. The analysis of RPC communication efficiency gives us suggestions for selection of communication parameters for effective construction of specific network implementations.

1. Przetwarzanie rozproszone

Przetwarzanie rozproszone realizowane jest w środowisku składającym się z wielu węzłów połączonych siecią. U podstaw przetwarzania rozproszonego stoi wzajemne udostępnianie zasobów lokalnych poszczególnych węzłów pozostałym jednostkom w sieci komputerowej. Z tą cechą związana jest potrzeba zapewnienia właściwych metod komunikacji oraz ocena ich efektywności. Problem analizy wydajności mechanizmów komunikacji na platformie typu UNIX został zanalizowany w pracy dyplomowej [1], zrealizowanej w Instytucie Informatyki Teoretycznej i Stosowanej PAN. W trakcie jej realizacji zwrócono uwagę na

nieregularności wydajności komunikacji w zależności od długości przesyłanego pakietu, występujące w trakcie przesyłu danych z wykorzystaniem zdalnego wywołania procedur. Fakt ten przyczynił się do bardziej szczegółowego zainteresowania się problematyką wydajności zdalnego wywoływania procedur oraz podjęcia próby głębszej interpretacji i wyjaśnienia uzyskanych wyników.

2. Mechanizmy komunikacji

W środowisku rozproszonym procesy mogą być realizowane na jednej lub kilku jednostkach przetwarzania i w zależności od tego można stosować różne mechanizmy komunikacji. Procesy działające na różnych jednostkach przetwarzania komunikują się ze sobą za pośrednictwem łącza sieciowego oraz odpowiednich protokołów komunikacyjnych. Z poziomu programu użytkownika dane do procesu odległego można przysyłać jawnie przez specjalnie utworzony dla tej sesji punkt komunikacyjny. Inną metodą jest przysyłanie niejawne, kiedy program użytkownika komunikuje się z odległym procesem za pośrednictwem systemu komunikacyjnego, do którego dostęp realizowany jest w sposób identyczny, jak odwołania realizowane w autonomicznym systemie. Szczegóły implementacji rozwiązań sieciowych nie mają wtedy znaczenia.

3. Zdalne wywoływanie procedur

Zdalne wywoływanie procedur, zwane RPC (ang. *remote procedure call*), jest metodą przekazywania sterowania z programu do zdalnej procedury z gwarantowanym powrotem sterowania do miejsca wywołania procedury [3]. Nieodłączną częścią mechanizmu wywoływania procedur jest przekazywanie argumentów przez część wywołującą do części wywoływanej, a zdalna procedura ma możliwość zwrotu wartości wyjściowych.

3.1. Przeźroczystość mechanizmów RPC

Podstawową cechą mechanizmu RPC jest zamaskowanie interfejsu sieciowego przed użytkownikiem, który ma do dyspozycji interfejs wyższego rzędu, więc mechanizm RPC musi być tak zaprojektowany, by z punktu widzenia programisty operacja RPC nie różniła się od lokalnego wywołania procedury. Mechanizm jest więc przeźroczysty (ang. *transparent*) dla użytkownika.

Podstawowe znaczenie dla przeźroczystości ma sposób przekazywania argumentów zdalnej procedurze oraz odbierania jej wartości wynikowych. Aby możliwa była komunikacja

między procesami, działającymi w odmiennych systemach operacyjnych, musi istnieć ściśle określona semantyka, interpretowalna jednoznacznie przez obie strony.

3.2. Przebieg zdalnego wywoływania procedur

Klient wywołuje procedurę lokalną, nazywaną łącznikiem klienta (ang. *client stub*), która jest reprezentacją procedury zdalnej w systemie lokalnym. Argumenty wywołania, po sformowaniu w komunikaty, zostają przygotowane do wysłania do systemu zdalnego. Przygotowane komunikaty wysyłane są do systemu zdalnego przez łącza komunikacyjne. Do transmisji może być wykorzystany protokół połączeniowy lub bezpołączeniowy.

Działający w systemie zdalnym łącznik serwera (ang. *server stub*) oczekuje na żądania klientów, odbiera komunikaty i, po rozformowaniu do postaci argumentów, wywołuje właściwą procedurę lokalną, przekazując do niej argumenty przesłane przez klienta.

Po wykonaniu procedury obsługi w węźle serwera wartości wynikowe zostają przygotowywane w łączniku serwera do wysłania przez uformowanie komunikatu lub komunikatów sieciowych i w tej postaci są przesyłane do systemu klienta, gdzie po odebraniu zostają przekształcone do postaci zgodnej z deklaracją procedury. Przekazanie wartości wyjściowych stanowi drugi etap działania procedury zdalnej odwzorowany w systemie lokalnym.

3.3. Implementacja mechanizmu Sun RPC

Jedną z najczęściej implementowanych realizacji zdalnego wywoływania procedur jest otwarty system obliczeń sieciowych firmy Sun - ONC (ang. *Sun Microsystems' Open Network Computing*). Realizacja Sun RPC składa się z biblioteki procedur zapewniających poprawną obsługę protokołu, standardu opisu danych zewnętrznych XDR (ang. *eXternal Data Representation*) oraz kompilatora *rpcgen*, który, na podstawie definicji interfejsu procedury zdalnej, generuje procedury łączników oraz procedury XDR [3].

3.3.1. Organizacja usług

Jeden serwer zawiera jeden program identyfikowany przez numer programu. Numer programu nie może przybierać wartości zastrzeżonych (numery od 0 do 1FFFFFFF). O przydzielonych numerach można dowiedzieć się wykonując polecenie `rpcinfo -p`.

Jeden program ma jedną lub kilka wersji identyfikowanych przez numer wersji. Każda z wersji zawiera jedną lub kilka procedur identyfikowanych przez numer procedury, który nie może przyjmować wartości zastrzeżonej „0”.

3.3.2. Nawiązanie połączenia

Klient, w celu wywołania odległej procedury, musi uzyskać adres usługi i dlatego wysyła zapytanie do systemu serwera, w którym proces zarządcy portmap przekazuje klientowi odpowiedź na podstawie listy numerów zarejestrowanych programów oraz wersji wraz z odpowiadającymi im numerami portów. Nawiązanie połączenia klienta z systemem zdalnym nazywa się związywaniem (ang. *binding*).

3.3.3. Cechy transportu

Jako protokół transportu może zostać wykorzystany protokół TCP lub UDP. Protokół UDP narzuca ograniczenie na rozmiar danych przekazywanych jako argumenty wywołania oraz zwracane wyniki.

RPC ma określony limit czasu oczekiwania na odpowiedź serwera. W przypadku protokołu TCP, jeżeli odpowiedź nie nadejdzie przed upływem tego czasu, zgłaszany jest błąd. W przypadku protokołu UDP, zanim zostanie zgłoszony błąd, żądanie jest ponawiane określoną liczbę razy. Jednak protokół ten nie zapewnia poprawności transmisji, ponieważ nie zabezpiecza przed możliwością zgubienia datagramów. Dlatego, w przypadku protokołu UDP, konieczne jest przeprowadzanie kontroli poprawności transmisji w ramach aplikacji.

3.3.4. Standard XDR

Standard XDR umożliwia opisywanie, reprezentowanie i przesyłanie danych w sposób niezależny od komputera. W przypadku mechanizmu RPC danymi tymi są argumenty zdalnej procedury oraz wartości przekazywane przez tę procedurę. XDR opiera się na dwóch założeniach: format liczb ma postać *mniejsze wyżej*, tzn. starsze bajty mają niższe adresy, a rozmiar pola nie może być mniejszy od 32 bitów. Oznacza to, że gdy przesyłany będzie łańcuch znakowy o długości nie będącej wielokrotnością czterech bajtów, zostanie przesłany pakiet zawierający wielokrotność czterech bajtów, przy czym bajty nieznaczące zostaną wypełnione zerami.

Standard XDR opisuje dwa pojęcia: potok XDR, czyli ciąg bajtów, w którym dane są reprezentowane w formacie XDR oraz filtr XDR, którym jest procedura kodująca lub dekodująca określony typ danych. Istnieją trzy rodzaje potoków XDR, jednak dla systemu RPC ma znaczenie tylko tzw. potok w pamięci z wykorzystaniem gniazd UDP oraz potok komunikatów z wykorzystaniem gniazd TCP. Za pomocą tych potoków przesyłane są argumenty i wyniki zdalnej procedury. Z kolei, w zależności od formatu tych argumentów i wyników przekazywanych przez procedurę, wykorzystywane są odpowiednie filtry.

4. Badanie wydajności mechanizmów komunikacji

Badanie wydajności mechanizmów komunikacji dla konkretnych rozwiązań sprzętowo-programowych jest ważnym elementem, mającym na celu ocenę możliwości obliczeniowych konkretnej konfiguracji rozproszonej. Niemniej istotny jest element weryfikacji rzeczywistych zachowań systemu rozproszonego z wynikami uzyskanymi na drodze obliczeń lub symulacji. Efektywność komunikacji zależy w znacznym stopniu od drogi przepływu informacji przez elementy pośredniczące kanału komunikacyjnego, z których każdy po drodze wnosi dodatkowe obciążenie związane z przygotowaniem danych do dalszej transmisji.

4.1. Przepływ informacji dla mechanizmu RPC

Sposób organizacji wymiany danych w przypadku mechanizmu RPC oraz badania jego wydajności w odniesieniu do wydajności protokołów TCP i UDP, stanowiących dla niego warstwę transportową, wpływa na możliwość oceny kosztu związanego z działaniem samego mechanizmu RPC. Koszt jest konsekwencją działania funkcji XDR i stanowiący jest przez czas potrzebny na związanie, czyli połączenie klienta z właściwym systemem zdalnym, a w przypadku protokołu TCP dodatkowe opóźnienie wprowadzane jest przez mechanizm wyodrębniania rekordów w strumieniu bajtów. Koszt ten wiąże się także z istnieniem dodatkowych warstw mechanizmu RPC, stanowiących interfejs wyższego poziomu nad interfejsem sieciowych funkcji systemowych wykorzystywanych przez ten mechanizm.

4.2. Specyfikacja programu testującego

W celu przetestowania mechanizmu zdalnego wywoływania procedur napisano program, który ma na celu przygotowanie struktury argumentów składającej się ze znaczników pakietu oraz wypełnienia łańcuchem znaków do wymaganego rozmiaru pakietu. Program w wersji źródłowej składa się z trzech plików, w tym z pliku informacyjnego rejestracji programu. W wyniku ich kompilacji i konsolidacji wygenerowano program klienta i serwera. Zadaniem programu klienta jest wywołanie zdalnej procedury w systemie serwera. Argumentem procedury jest pakiet danych o określonej długości. Z kolei program serwera uruchamia procedurę, której jedynym zadaniem jest przekazanie otrzymanego jako argument pakietu ze zmienionym znacznikiem czasu.

4.3. Działanie programu testującego efektywność mechanizmu RPC

Działanie programu testującego polega na wysyłaniu bloku, zawierającego określoną liczbę znaków do serwera, który w odpowiedzi odsyła taki blok o tej samej liczbie znaków do

klienta. Serwer, oprócz odesłania odpowiedzi, realizuje pomiar czasu w chwili nadejścia pakietu oraz zapisuje jego wartość do odpowiedniego pola bufora danych. W odniesieniu do efektywności działania programu serwera, w wersji bez tych funkcji, stwierdzono, że spowodowany narzut czasowy realizacji pomiaru mieści się poniżej rozróżnialnej granicy błędu pomiarowego czasu realizacji transmisji. Można więc przyjąć, że na czas trwania komunikacji składa się jedynie czas transmisji przez sieć argumentów i wyników zdalnej procedury oraz czas trwania realizacji funkcji systemowych, które muszą być wykonane zawsze, niezależnie od funkcji programu. Możliwe jest zatem określenie efektywności mechanizmu RPC w zależności od ilości danych przesyłanych jako argumenty.

4.4. Środowisko testowe

W celu pomiaru efektywności mechanizmu RPC utworzono program klienta oraz serwera według wcześniejszego opisu. W programie klienta zamieszczono instrukcje realizujące obliczenie efektywności transmisji.

Testy przeprowadzono na następujących platformach sprzętowo-programowych: komputer PC 486DX4-100 z systemem operacyjnym Linux oraz komputer PC Pentium 166 i 2 komputery SUN Ultra 2 pod kontrolą systemu operacyjnego Solaris 2.5.1.

Komputery SUN połączone były ze sobą za pomocą sieci ATM (lane), a komputery typu PC korzystały z łącza Ethernet. Sieć ATM skonfigurowana została w oparciu o przełącznicę, a Ethernet z wykorzystaniem koncentratora. Wzajemne połączenie sieci miało miejsce w przełącznicy Ethernet wyposażonej w port ATM.

Testy efektywności przeprowadzono we wszystkich możliwych konfiguracjach połączeń dwu platform oraz klienta z serwerem z zastosowaniem protokołu TCP/IP.

Seria testów obejmowała pomiary czasu trwania od 100 do 3000 wywołań określonej procedury zdalnej, dla pakietów danych o długościach z zakresu od 16 B do 8192 B co 16 B. Liczba wywołań procedury zdalnej zależała od długości pakietu – więcej wywołań dla krótszych pakietów. W podanych konfiguracjach przeprowadzono po 10 serii testów, co pozwala na ocenę statystyczną uzyskanych wyników.

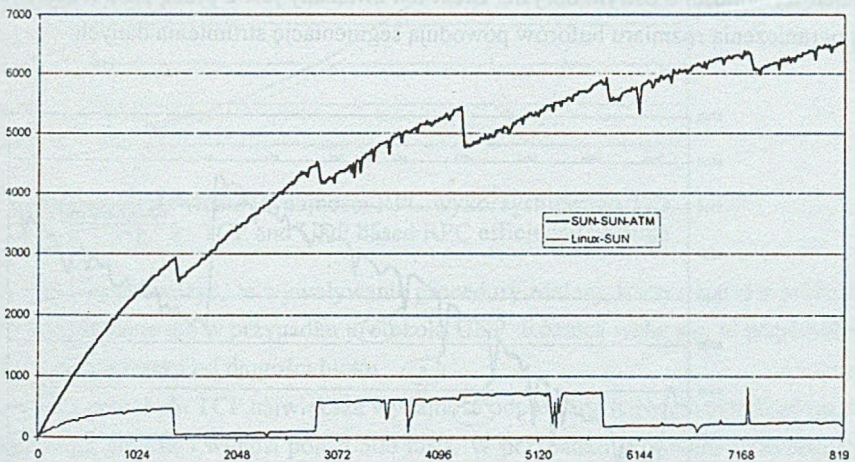
Procedura zdalna kończy się po przekazaniu pakietu zwrotnego o tej samej długości jak w argumentach, więc, w rzeczywistości, pomiar czasu obejmuje transmisję podwójnej liczby danych w stosunku do długości argumentu. Czas realizacji jest przeliczany na wydajność, której jednostką jest liczba kilobajtów na sekundę.

4.5. Wyniki testów

Na poniższych wykresach przedstawiono najbardziej charakterystyczne wyniki testów dla różnych konfiguracji platform sprzętowo-programowych. Wyniki uzyskane z 10 serii testów

dla poszczególnych długości pakietu poddano analizie statystycznej. Dla niektórych wartości długości argumentu zaobserwowano pojedyncze pomiary znacznie różniące się od wartości średniej (spadek wydajności o ok. 50%). Z tego powodu nie brano pod uwagę minimalnej wartości skrajnej pomiaru dla danej długości pakietu. Wartości maksymalne pozostawiono. Po odrzuceniu wartości skrajnej obliczono dla pomiarów wartości średnie, które zostały zamieszczone na wykresach. Odchylenie standardowe dla całego zakresu pomiarów nie przekroczyło 5%.

4.5.1. Wpływ rozmiaru argumentu na efektywność komunikacji



Rys. 1. Zależność wydajności komunikacji od długości pakietu
Fig. 1. Communication efficiency depended on packet size

Na rys. 1. pokazano wpływ długości pakietu na wydajność komunikacji dla dwóch różnych platform. Wykres SUN-SUN-ATM ilustruje wzrost wydajności komunikacji pomiędzy dwoma systemami Sun Ultra2 pracującymi w sieci ATM, a wykres Linux-SUN sporządzony został dla systemu Linux i stacji SUN. W drugim przypadku przepływ danych odbywa się od komputera PC przez łącze Ethernet, a następnie przełącznice i ATM do stacji SUN. Z obu wykresów można wywnioskować, że wydajność komunikacji rośnie wraz ze wzrostem długości bloku.

4.5.2. Wpływ środowiska operacyjnego na charakter transmisji

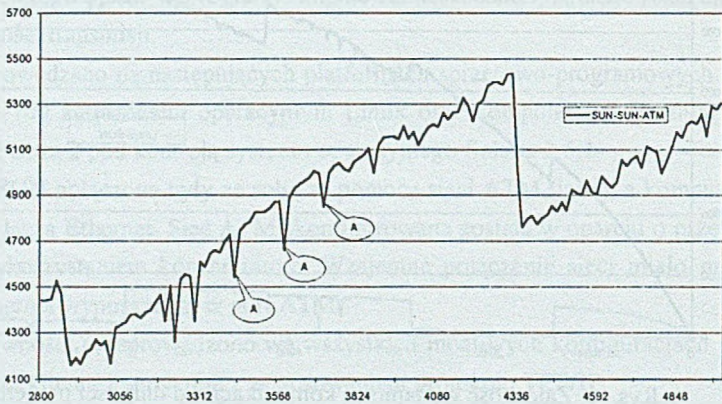
Na wykresie Linux-SUN z rys. 1. zauważalne są gwałtowne spadki wydajności komunikacji dla ograniczonego zakresu długości pakietu. Efekt ten jest związany z charakterem formowania danych w strumieniu XDR w systemie Linux. Niezależnie od tego, czy serwer uruchomiony został na platformie Linux, czy SUN, z klientem po drugiej stronie, wykresy za-

leżności wydajności nie różnią się bardziej niż przyjęty wcześniej próg 5%. Charakter przebiegu zależności wydajności, przy współpracy systemu Linux z pozostałymi systemami testowymi, nie różni się od przedstawionego na rys. 1.

W różnych konfiguracjach, niezależnie od układu klienta względem serwera, charakter przebiegu wydajności narzucony jest przez platformę o najniższej wydajności i wszelkie wahania charakterystyczne dla pozostałych węzłów zanikają. Wynika z tego, że szczegóły implementacji RPC w zależności od środowiska operacyjnego nie mają znaczenia.

4.5.3. Wpływ infrastruktury sieciowej na przebieg charakterystyki wydajności

Na wykresie SUN-SUN-ATM z rys. 1. można również zaobserwować serię regularnych wahań wydajności o ostrym obrysie. Efekt ten związany jest z pracą przełącznicy ATM, której ograniczenia rozmiaru buforów powodują segmentację strumienia danych.



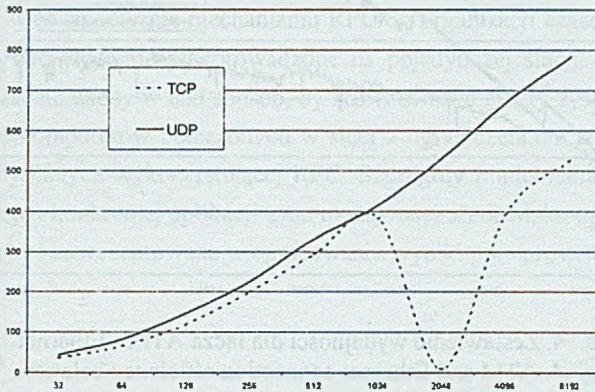
Rys. 2. Wpływ elementów sieciowych na wydajność komunikacji
Fig. 2. Influence of network components upon communication efficiency

Rys. 2. stanowi fragment wykresu SUN-SUN-ATM z rys. 1. W tym powiększeniu zauważalne są wahania wydajności związane z rozdrabnianiem pakietów na ramki ATM. Serię trzech wybranych punktów charakterystycznych została na rys.2. zaznaczona literą „A”. Na rozkład przedstawionych punktów ma wpływ rzeczywista długość przesyłanego pakietu, jego podział na komórki ATM oraz częstość pobierania próbek. Z tego powodu nie są widoczne wszystkie skoki. Szczegóły tego zachowania byłyby bardziej widoczne podczas badania protokołów niższych warstw.

4.5.4. Wpływ protokołu transportowego

Rys. 3., prezentowany w skali logarytmicznej, przedstawia zestawienie porównawcze dla protokołów transportowych UDP i TCP. Dane są wynikiem pomiarów wydajności mehani-

zmu RPC w konfiguracji Linux-SUN, gdzie zauważalny jest nagły spadek wydajności. Wyniki zostały zebrane tylko dla podanych etykiet, niezależnie od prezentowanych wcześniej pomiarów, wyłącznie w celu zestawienia charakterystyk obu porównywanych protokołów transportowych i z tego powodu wynikają różnice dla protokołu TCP w odniesieniu do poprzednich wykresów.



Rys. 3. Zestawienie wydajności RPC wykorzystującego TCP i UDP
Fig. 3. TCP and UDP based RPC efficiency balance

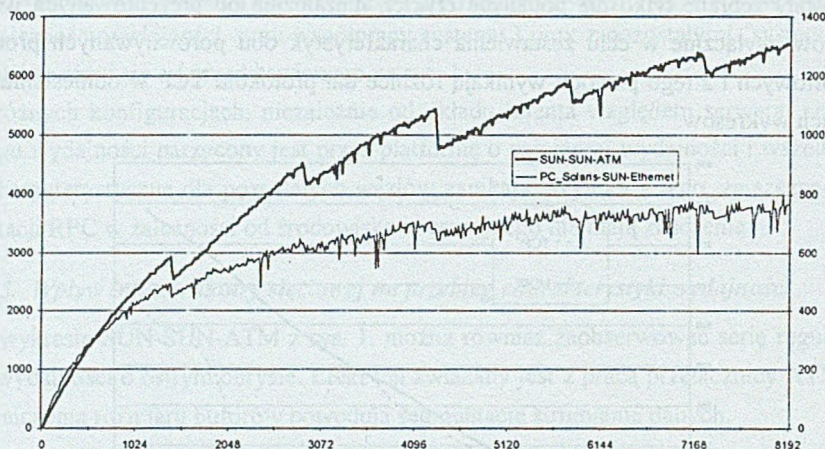
Na rys.3. można zauważyć, że wywoływanie procedury zdalnej, korzystającej z protokołu TCP, jest mniej wydajne niż w przypadku protokołu UDP. Różnica waha się, w przybliżeniu, od 10 do 40% w zależności od długości bloku.

W przypadku protokołu TCP największa wydajność odpowiada największym brany pod uwagę długościom bloków i wynosi ponad 500 kB/s. W przypadku protokołu UDP największa wydajność dochodzi do granicy 800 kB/s.

4.5.5. Porównanie implementacji ATM i Ethernet

W celu uzyskania charakterystyki wydajności dla łącza Ethernet zestawiono połączenie pomiędzy komputerem SUN i PC z systemem operacyjnym Solaris 2.5.1 (PC_Solaris-SUN-Ethernet) korzystając z konfiguracji zaprezentowanej w punkcie 4.4. Uzyskane wyniki pomiarów porównane zostały z prezentowaną uprzednio konfiguracją SUN-SUN-ATM.

Zestawienie wyników zaprezentowano na rys. 4., na którym można zaobserwować podobny charakter obu przebiegów, porównując go z systemem Linux. Maksymalna wydajność komunikacji, w konfiguracji wykorzystującej Ethernet, dochodzi do wartości 800 kB/s, w porównaniu do wydajności w sieci ATM dochodzącej do 7 MB/s. Dla konfiguracji Ethernet niezauważalne są cechy charakterystyczne, wynikające z pracy urządzeń sieciowych ATM, przedstawione w punkcie 4.5.3., pomimo to że dane pomiędzy stacjami przechodzą również przez urządzenia ATM.



Rys. 4. Zestawienie wydajności dla łącza ATM i Ethernet
 Fig. 4. ATM and Ethernet connection efficiency balance

4.5.6. Porównanie efektywności RPC z efektywnością protokołu TCP

Porównanie efektywności mechanizmu RPC, używającego protokołu transportowego TCP, z efektywnością samego protokołu TCP wskazuje na maksymalnie 18-krotną i minimalnie 3-krotną różnicę na korzyść protokołu TCP [1]. Przy czym różnica jest tym mniejsza, im większa jest długość przesyłanych łańcuchów znakowych. Jest to zrozumiałe. W przypadku długich bloków, przeważającą część czasu trwania całej komunikacji stanowi bowiem przesyłanie danych przez sieć, które jest realizowane właśnie przez protokół TCP. Natomiast w przypadku bloków krótszych w całkowitym koszcie komunikacji zwiększa się udział czynności związanych z obsługą mechanizmu RPC, co powoduje większą różnicę wydajności w stosunku do wydajności samego protokołu.

Podobna zależność występuje w przypadku protokołu UDP, jednak różnica wydajności jest tu maksymalnie 4-krotna, a najmniejsza zarejestrowana różnica pomiędzy wydajnością protokołu a wydajnością systemu RPC wynosi około 40%.

5. Posumowanie

Stosowanie mechanizmu RPC służy do przeniesienia wywołania procedury lokalnej do dziedziny komunikacji sieciowej. Wiąże się to ze spadkiem wydajności realizacji pojedynczej procedury. R. Stevens podaje [2], że zmniejszenie wydajności, wynikające ze zdalnego

wywołania procedury, jest zwykle co najmniej 10-krotne w stosunku do wywołania procedury lokalnej.

Wydajność komunikacji z zastosowaniem mechanizmu RPC może być nawet kilkanaście razy mniejsza od wydajności samej sieci przy przesyłaniu podobnej ilości danych. Zauważalne jest to w szczególności w systemie Linux.

Celowe jest więc stosowanie mechanizmu RPC do organizacji czasochłonnych obliczeń, które nie mogą być wydajnie przeprowadzone na pojedynczej stacji. Rozłożenie obliczeń należy przeprowadzić wtedy w taki sposób, by zrównoważyć efekty zyskane dzięki równoległej pracy wielu komputerów połączonych w sieci z ograniczeniami stawianymi przez wydajność wymiany danych wykorzystującej RPC. Szczegóły implementacji tego rozwiązania zależą od wymagań konkretnej aplikacji oraz możliwości środowiska operacyjnego, w czym przydatne mogą być zaprezentowane w opracowaniu wyniki pomiarów wydajności.

LITERATURA

- [1] Przyłucki P.: Analiza mechanizmów komunikacji systemu Unix w heterogenicznym rozproszonym środowisku operacyjnym, Praca Dyplomowa Pol. Śl., Gliwice 1997.
- [2] Stevens W. R.: Programowanie zastosowań sieciowych w systemie Unix. WNT, Warszawa 1995.
- [3] SunSoft: Network Programming Answer Book. SunSoft Answer Book electronic media, 1996.
- [4] O'Reilly & Associates, Inc.: UNIX Administracja systemu. Oficyna Wydawnicza RE-AD ME, Warszawa 1996.
- [5] Wolisz A.: Podstawy lokalnych sieci komputerowych. WNT, Warszawa 1990, 1992.

Recenzent: Dr inż. Mirosław Skrzewski

Wpłynęło do Redakcji 16 stycznia 1998 r.

Abstract

A distributed computing environment consists of multiple nodes connected via network. These nodes may be computers or they may consist of a variety of elements working with conjunction with computers. A computer and other elements will typically be shared widely in the networked computing environment. Distributed applications make use of cooperative

processing services. It is provided by remote execution services such as Remote Procedure Call (RPC).

The RPC model implements the procedure call generalized from a single system in a distributed architecture. Its useful feature is transparency which allows to generate calls to a standard interface without being concerned with the underlying network implementation.

The analysis of RPC communication efficiency is performed by a test program which consists of client and server parts. From client side remote procedure is called via client stub which gives control to the rest of RPC system. Procedure handler and its arguments are transferred to a server stub, where they are used by server side procedure. Results are transferred back to client part. The time between start and end of 100 remote procedure calls was measured for variable argument and result lengths. Network environment for test programs used SUN and PC computers with Solaris or PC-Linux operating systems over ATM and Ethernet based networks. The comparison between PC-Linux over Ethernet and SUN-Solaris over ATM RPC efficiency is presented on Fig. 1. The influence of ATM network elements is shown on Fig. 2. Transport layer dependency is illustrated on Fig. 3.