

Podpora distribuovaného výpočtu

<small>Z ČWUT</small>

Obsah

- 1 Relační a presentační služby
 - 1.1 Relační
 - 1.2 Prezentací
- 2 Datové konverze, komprese a kryptografie
- 3 Meziprocesová komunikace v distribuovaných systémech
- 4 Výměna zpráv, mechanismy RPC, distribuovaná paměť
 - 4.1 Předávání zpráv
 - 4.2 RPC
 - 4.3 Distribuovaná sdílená paměť
- 5 Technologie Java RMI, CORBA a SOAP
 - 5.1 RMI
 - 5.2 CORBA
 - 5.3 SOAP
 - 5.4 Tento článek potřebuje editovat

Relační a presentační služby

Relační

- řízení dialogu, synchronizace a řízení aktivit
- session protocol data unit
- spojení: navázání - komunikace - uzavření

- řízení dialogu
 - poloduplexní přenos dat (omezení plněduplexní možnosti transportní vrstvy)
- synchronizace
 - vkládání synchronizačních bodů do komunikace
 - návrat k bodu v případě havárie na přijímací straně - žádá příjemce
 - velké - explicitně potvrzované, nelze se před ně vrátet
 - malé - nepotvrzované, lze se vrátit k libovolnému
- řízení aktivit
 - kratší úseky spojení (přenos souboru, DB transakce, ...) - aktivity
 - dovoluje uzamknout příslušná data

Prezentací

- transformace dat mezi počítači s různými zobrazeními
- komprese dat
 - statistická (RLE)
 - diferenční (PCM)
 - slovníková (LZW)
- kryptografická ochrana dat

- symetrická
 - stejný klíč na obou stranách
 - substituce - náhrada
 - transpozice - změna pořadí
 - DES
- asymetrická
 - umožňuje veřejný a soukromý klíč
 - RSA
 - hashovací funkce
- síťová reprezentace dat
 - data se před odesláním převedou na určitou reprezentaci, před přijetím se převedou na reprezentaci cíle
 - stačí menší počet překladačů - jen podle počtu různých reprezentací (nativní - síťová), ne pro všechny kombinace reprezentací
 - ASN.1

Datové konverze, komprese a kryptografie

#Prezentační služby

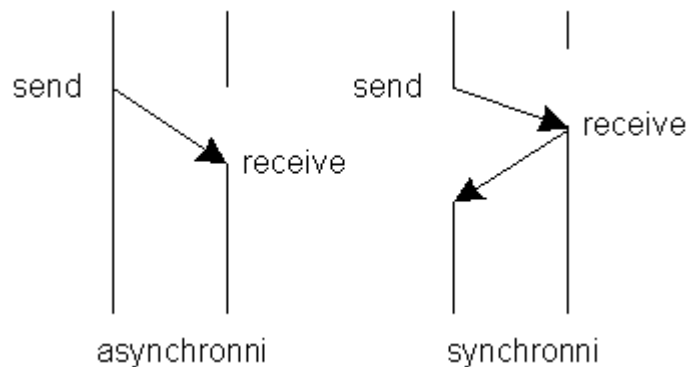
Meziprocesová komunikace v distribuovaných systémech

- výměna zpráv
- sdílená paměť

Výměna zpráv, mechanismy RPC, distribuovaná paměť

Předávání zpráv

- synchronní - odesílatel čeká až do příjmu
- asynchronní - nečeká
- oboje prostřednictvím kanálu - umožňuje multiplex - např. očekávat zprávu od více příjemců



RPC

- vzdálené volání procedur
- klient požádá, server provede a vrátí klientovi odpověď
- z hlediska zdrojového kódu vypadá jako volání běžného podprogramu - stub

Mechanismus RPC:

- stub na straně klienta - transformuje volání na zprávu serveru a zpětně odpověď serveru
- stub na straně serveru - transformuje přijatou zprávu na volání a pak odpověď na zprávu klientu
- vlastní kód RPC procedury na serveru

Distribuovaná sdílená paměť

- sdílení proměnných
- v zásadě klient vždy používá pouze metody READ a WRITE, jako by pracoval s vlastními daty. Rozdíl je v tom, že buď je před samotným provedením R/W provedena rutina, která nakopíruje sdílenou proměnnou ke klientovi (proměnlivé umístění), nebo je každé R/W volání převedeno vždy na zprávu, která je zaslána vlastníkovvi proměnné (pevné umístění).

Metody:

- pevné umístění proměnné na jednom uzlu
- migration - také jediný originál
- read-replication - jediná modifikovatelná kopie, další read-only kopie
- full replication - více plnohodnotných kopií, při aktualizaci nutno provést všude

Technologie Java RMI, CORBA a SOAP

RMI

- volání: client - stub - remote reference layer (- transport layer) - skeleton - server
- jinak jako RPC

CORBA

- na rozdíl od RPC nepotřebuje C - klient i server v různých jazycích
- ORB - na portu přijímá požadavky - má každý objekt
- OA - object adapter - server
- klient má stub, server skeleton
- podpora dynamického volání - když předem nevím, jak volání vypadá (parametry apod.)
- komunikace buď client - client stub - network transport - port mapper - server stub - server
- nebo client - cl. stub - ORB - BOA - skeleton - server

SOAP

- nástupce XML-RPC
- zprávy jsou přenášeny v XML, což má své výhody i nevýhody:
 - výhody - XML je standard. Je tedy otevřená cesta pro meziplatformní komunikaci
 - nevýhody - nutnost parserovat každou příchozí zprávu
 - výhody i nevýhody - přenášená data jsou srozumitelnější a tedy i odposlechnutelnější
- pro přenos používá HTTP, snadno tedy čelí také firewallům
- počátky SOAP se datují na rozmezí let 1998-9
- ukázka zpráv http://cs.wikipedia.org/wiki/Simple_Object_Access_Protocol#Uk.C3.A1zka
- popis WSDL

-
- Stránka byla naposledy editována v 11:45, 8. 2. 2007.
-