

PREZENTAREA ARHITECTURII COMMON MEDIA SERVER

ing. Mihai Constantin

“Luxoft România”, Calea Floreasca 167, București, România
Student Ph.D. la “Universitatea Politehnica” din București
Email: mconstantin@luxoft.com

Rezumat: Scopul acestui studiu este de a prezenta un MediaServer care să poată interacționa cu un Software IP PBX (IPSmallOffice), pentru a oferi capabilități de codec VoIP. MediaServer trebuie să înlocuiască o componentă hardware VCM (Voice Compression Module, conține 2 DSP-uri) care furnizează acest suport IP PBX-ului.

Common Media Server are următoarele capabilități:

- Conferință
- Detectare de digiti DTMF
- Generare de ton, 'music on hold', generare de digiti DTMF

Acest document este o privire de ansamblu la nivel înalt a arhitecturii de Common Media Server, descriind modulele importante.

Cuvinte cheie: Common Media Server, IPC, MSML, SIP, TCP, Procese.

Abstract: The goal of this study is to present a MediaServer which can interact with a software IP PBX (IPSmallOffice) to provide VoIP codec capabilities. MediaServer is to replace a hardware component VCM (Voice Compression Module contains two DSP's) which provide that support.

Common Media Server has the following capabilities:

- Conference
- DTMF digits detection
- Tone generation, Music on hold, DTMF digits generation

This document is a high level overview of Common Media Server architecture, describing the main modules.

Keywords: Common Media Server, IPC, MSML, SIP, TCP, Processes.

1. Componentele Common Media Server

Arhitectura Common Media Server este compusă din mai multe componente care comunică între ele pe baza mecanismului de IPC Message Queue. Fiecare componentă Common Media Server este pornită ca un proces normal și poate controla unul sau mai multe threaduri.

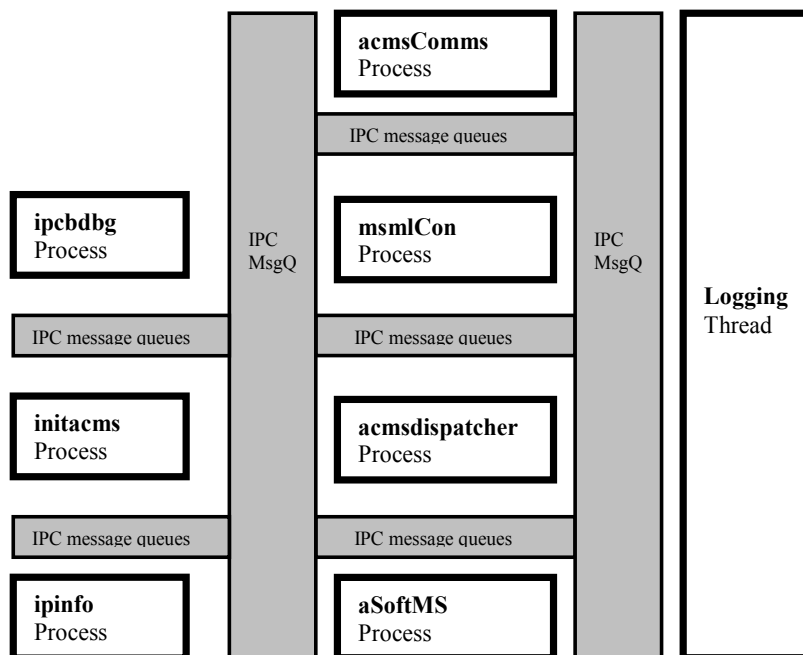


Figura 1. Componentele Common Media Server.

Comunicarea între Common Media Server și IPSmallOffice PBX (Fig. 2) se face folosind mesaje SIP învelite în pachete MSML. Când IPSmallOffice este pornit, acesta solicită un grup de porturi pe care se va asculta pentru datele RTP / RTCP. Pentru a obține acest grup de porturi, IPSmallOffice trimite la Common Media Server o serie de mesaje SIP INVITE (inclusiv un „dummy”codec și flux SDP audio setat la inactiv), iar apoi Common Media Server creează sesiuni MSML care deschid socketuri pentru aceste porturi pre-allocate. Așa că, în momentul când o conexiune de telefon sosește în IPSmallOffice, atunci IPSmallOffice întoarce un port pre-allocat în cadrul negocierii.

Atunci când un codec este configurat, IPSmallOffice va trimite un mesaj RE-INVITE la Common Media Server cu codec adecvat. Acest mesaj RE-INVITE va activa portul de conexiune în Common Media Server (portul nu va fi schimbat ca urmare a RE-INVITE). Obiectele MSML se indexează pe baza câmpului „tag” din SIP header.

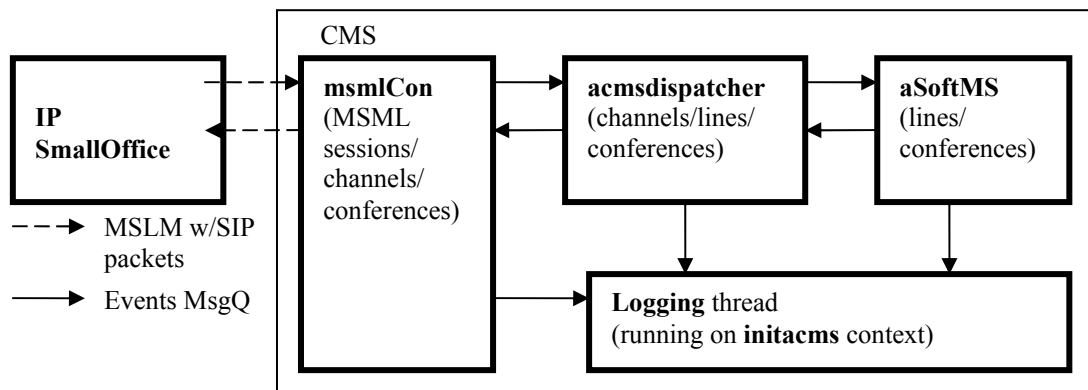


Figura 2. Comunicarea Common MediaServer si IPSmallOffice

Există un script de inițializare pentru pornirea Common Media Server (CMS). Acesta este situat în /etc/rc.d/init/ACMS. De asemenea, acest script este folosit pentru setarea mediului pentru procesele Common Media Server. Depinde de setările de configurare, dimensiunea maximă a memoriei partajate și atributele cozilor mesaj IPC din kernel space trebuie să fie stabilite în mod corespunzător (de exemplu, pentru 2000 de canale dimensiunea memoriei partajate ar trebui să fie setat la 650 000 000). După ce setările de mediu sunt efectuate, acest script „acms” pornește procesul "initacms".

2. Prezentarea procesului InitAcms

2.1 Primul proces Common Media Server început.

Scopul principal al acestui proces este acela de a gestiona procesele Common Media Server implicate în procedura de comunicare cu IPSmallOffice. Sunt descrise mai jos o parte din alte roluri:

- citește setările de Common Media Server din fișierele de configurare și furnizează aceste informații atunci când acestea sunt solicitate de către procesele Common Media Server;
- aloacă și inițializează configurabil cozile mesaj IPC utilizate pentru comunicații inter-procese de către componentele Common Media Server;
- setează limitele resurselor kernel la RLIM_INFINITY și RLIM_CORE;
- deschide „pin-holes” pentru portul de ascultare SIP/MSML în aplicația Firewall executând câteva reguli iptable.

2.2 Prezentare

Algoritmul de pornire a altor procese și threaduri ale Common Media Server gestionate de acest proces este descris mai jos:

- creează și pornește threadul de Logging (scrie mesajele de log într-un fișier);
- pornește binarul „acmsComm”;
- pornește restul proceselor Common Media Server în ordinea în care au fost predefinite în fișierul processTable.cfg;
- creează și pornește un thread care se ocupă de actualizarea schimbărilor (la fiecare 5 minute acest thread verifică dacă fișierul de configurare a fost modificat și dacă modificările sunt necesare, va trimite mesaje de notificare pentru alte procese Common Media Server):
 - creează și pornește un thread să se ocupe de evenimentele IPC trimise de el însuși sau alt proces Common Media Server;
 - creează și pornește un thread timer care, periodic (valoarea implicită este de 11 secunde), trimite evenimente IPC pentru căutare după cererea de închidere;
 - creează și pornește un thread timer (valoarea implicită este de 30 de secunde) pentru verificarea sistemului de logare ca să nu fie blocat;
 - stă într-o buclă while și așteaptă semnale care pot închide un proces, sau ”child signal” pentru a loga un mesaj care explică de ce ”child process died”.

3. Prezentarea procesului AcmsComms

3.1 Prezentare

Acest proces este folosit la trimiterea evenimentelor IPC peste conexiuni TCP. Procesul ascultă pentru conexiunile TCP pe un port configurabil (valoarea implicită este # 10334). Poate fi util atunci când unele dintre componentele Common Media Server sunt distribuite (acestea nu se găsesc pe aceeași mașină).

3.2 Algoritm de funcționare

- inițializează și setează nivelurile de log; citește interfața de rețea utilizată pentru comunicarea prin sockets; configurează propria coadă de mesaje IPC și inițializează interfața (IPC coadă) cu procesul „initacms”;
- creează și începe un thread pentru a asculta conexiunile TCP;
- stă într-o buclă while de așteptare pentru evenimente IPC care trebuie să fie transmise la alte endpoints.

4. Prezentarea procesului Msm1Con

4.1 Prezentare

Acest proces comunică cu IPSmallOffice folosind pachete MSML care conțin mesajele SIP în interior. IPSmallOffice pot trimite mesaje ca SIP INVITE, INFO, ACK și Common Media Server pot răspunde cu mesajele SIP ca 100 Try, 200 OK, BYE. De asemenea, acest proces comunică cu un alt proces „acmsdispatcher” folosind cozile de mesaje IPC.

4.2 Mecanismul de funcționare al procesului

Când un mesaj SIP INVITE este primit, SIP Stack încearcă să găsească un DCSession corespondent care se ocupă cu acest mesaj (se caută DCSession utilizând o cheie compusă dintr-o tranzacție sau un dialog; dacă nu există nici o DCSession găsită, atunci este alocată una nouă). După ce se face corespondența cu DCSession, această sesiune SIP este verificată pentru INVITE-ul inițial care înseamnă să vadă că nu există alocată SessionUser, și pentru cazul nostru special căutam sesiunea MSML. Prin urmare, pentru orice nou mesaj INVITE inițial, o nouă sesiune de MSML va fi alocată și setată ca SessionUser pentru DCSession și, pentru orice mesaj, re-INVITE SessionUser trebuie să fie deja acolo, în interiorul DCSession (altfel se produce o eroare). Sesiunile MSML sunt folosite pentru translatarea comenzilor SIP în evenimente IPC. De asemenea, fiecare pachet MSML este parsat de Common Media Server folosind o bibliotecă numită LMX dezvoltată de Codalogic.

Figura 3 prezintă comenzile SIP și evenimente IPC implicate într-un algoritm de bază de pre-alocarea pentru o nouă conexiune în Common Media Server de către IPSmallOffice:

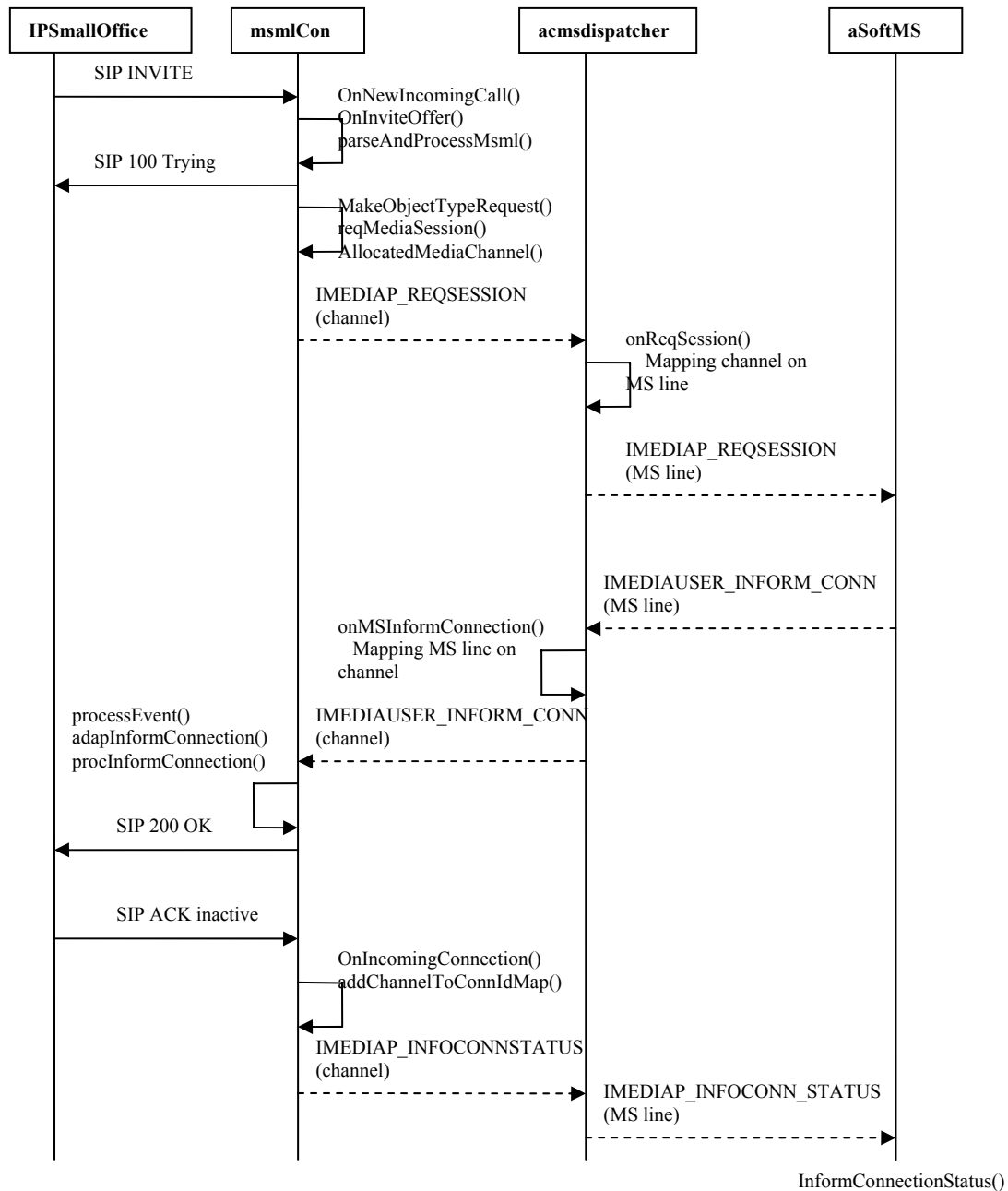


Figura 3.

MsmmlConference și obiectele MsmmlMediaChannel sunt gestionate de către MsmmlObjManager și sunt indexate utilizând o cheie pe baza valorii atributului „tag” luate din SDP body al mesajului SIP.

4.3 Algoritmul procesului

Algoritmul principal al acestui proces este descris mai jos:

- creează și pornește un thread timer care, periodic (valoarea implicită este de 20 de secunde), verifică DTMF time-outs;
- inițializează fișierele log și setează nivelurile log; cere setările de configurație de la „initacms” sau le citește din fișierul mediaServerInterface.cfg; configurează propria IPC message queue și inițializează interfețele (IPC message queue) cu procesele „initacms”;

„acmsdispatcher” ;

- alocă și inițializează managerul de obiecte MSML care este pasat către stiva SIP;
- creează și inițializează stiva SIP;
- creează și începe un thread timer care, periodic (la fiecare 1 secundă), trimite evenimente IPC către sine;
- creează și începe un thread pentru procesarea evenimentelor IPC;
- stă într-o buclă while pentru ascultarea și prelucrarea tuturor evenimentelor SIP trimise de IPSmallOffice.

5. Prezentarea procesului AcmsDispatcher

5.1 Scopul procesului

Scopul principal al acestui proces este acela de a face translatarea între canalele MSML și liniile Media Server (fiecare canal MSML trebuie să aibă asociată o linie MS). De asemenea, acest proces este proiectat pentru a gestiona resursele de la mai mult de un media server (în mod implicit există suport pentru 32 media servere și, probabil, se poate reduce acest număr). Fiecare proces al Media Server trebuie să se înregistreze la acest proces pentru a declara cât de multe canale și conferințe care pot fi gestionate. Deoarece aceste resurse sunt păstrate în memorie partajată, atunci acestea nu sunt eliberate când acest proces este oprit. Aceasta este util pentru cazul în care procesul este repornit și atunci nu este necesar să petreci timp pentru realocarea resurselor.

5.2 Algoritm de funcționare

Algoritm de funcționare al acestui proces este descris mai jos:

- inițializează fișierele log și setează nivelurile log; cere setările de configurație de la „initacms” sau le citește din mediaServerInterface.cfg; fișierul system.cfg; configurează propria IPC message queue și inițializează interfețele (IPC message queue) cu procesele „initacms”, „msmlCon”, și „aSoftMS”; alocă și inițializează managerul de canale folosind memorie partajată; în prezent nu sunt alocate prea multe resurse pentru un media server virtual (5000 de canale și 10000 de linii); în mod implicit sunt alocate un număr de 1214 conferințe (probabil se poate reduce acest număr);
- creează și începe un thread timer;
- stă într-o buclă while și procesează evenimente IPC;

6. Prezentarea procesului ASoftMS

6.1 Scopul procesului

Scopul principal al acestui proces este de a alocă și manipula canale și conferințe DSP atunci când acestea sunt solicitate. Un alt rol este de a deschide, pin-holes’ pentru porturile RTP/RTCP în aplicația Firewall utilizând reguli iptable.

6.2 Algoritm de funcționare

Algoritm principal al acestui proces este descris mai jos:

- inițializează fișierele log și setează nivelurile log; cere setări de configurație de la „initacms” sau le citește din fișierele system.cfg, audioPreferences.cfg, și softMediaServer.cfg; alocă și inițializează obiecte de control pentru gestionarea canalelor și conferințelor; valoarea implicită pentru MAX_NUMBER_CONFERENCES este de 1214 (probabil această valoare ar putea fi redusă);
- creează și începe un thread pentru procesarea evenimentelor IPC în „background”;

- stă într-un ciclu for și execută unele metode definite în obiecte de control la o anumită perioadă de timp; aceste metode sunt utilizate pentru procesare de streamuri audio și video.

7. Prezentarea procesului Ipinfo

Este folosit de scriptul „acms” pentru a regăsi statusul proceselor CommonMediaServer.

8. Prezentarea procesului IpcbDbg

Este un instrument de debug utilizat pentru a trimite evenimente IPC la un proces al Common Media Server.

Lista de abrevieri:

DTMF = Dual Tone Multi-Frequency

IPC = Inter-Process Communication

MS = Media Server

MSML = Media Server Markup Language

RTCP = Real Time Control Protocol

RTP = Real Time Protocol

SDP = Session Description Protocol

SIP = Session Initiation Protocol

CMS = Common Media Server

REFERINȚE

1. Inter Process Communications (<http://users.actcom.co.il/~choo/lupg/tutorials/multi-process/multi-process.html> for a tutorial).
2. Session Description Protocol (The SDP describes multimedia sessions for the purpose of session announcement, session invitation and other forms of multimedia session initiation.) <http://www.javvin.com/protocolSDP.html> for details.
3. SIP- Session Initiation Protocol. <http://www.ietf.org/html.charters/sip-charter.html> for details.
4. RFC 2327 - Internet Official Protocol Standard for RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals. This standard describes how to carry dual-tone multi-frequency (DTMF) signaling, other tone signals and telephony events in RTP packets. See details at <http://www.faqs.org/rfcs/rfc2833.html> .
5. RFC 2833 - Internet Official Protocol Standard for RTP Payload for DTMF Digits, Telephony Tones and Telephony Signals. This standard describes how to carry dual-tone multi-frequency (DTMF) signaling, other tone signals and telephony events in RTP packets. See details at <http://www.faqs.org/rfcs/rfc2833.html> .
6. RFC 3261 - Internet Official Protocol Standard for Session Initiation Protocol (SIP), an application-layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. These sessions include Internet telephone calls, multimedia distribution, and multimedia conferences. See details at <http://www.faqs.org/rfcs/rfc3261.html> .
7. JOHNSON, A. B.: SIP- Understanding the Session Initiation Protocol, Artech House, Boston, 271 p.