

# **REIMPLEMENTARE PDSH/PDCP**

## **Autori**

**Dendiu Denis-Cristian**

**Nicolescu Radu-Mihai**

**Ursu Arsenie**

Universitatea din Bucuresti

Departamentul de Informatica

Instrumente si tehnici de baza in informatica

2025-2026

## **Documentație Proiect**

### **1. Descrierea Problemei**

**De ce este necesar?**

- Operarea simultană și în masă a **mai multor sisteme/noduri** (management de cluster).
- **Exemplu de caz de utilizare:**
  - Servicii de servere (ex: Amazon EC2).
  - Configurarea multiplă de sisteme într-un mediu de birou.

### **Soluții Posibile**

Reimplementarea utilitatilor standard pentru execuția paralelă de comenzi și transferul de fișiere:  
\* **pdsh**: Rularea de comenzi în paralel.  
\* **pdcp**: Încărcarea (upload) de fișiere în paralel.  
\* **rpdcp**: Descărcarea (download) de fișiere în paralel.

### **2. Specificația Soluției**

#### **Obiective**

- **Deployment ușor** al nodurilor de Mașini Virtuale (VM-uri).
- **Reimplementarea** funcționalităților cheie ale comenzilor pdsh/pdcp/rpdcp.

#### **Mediu de Rulare**

- Solutia rulează în **terminal**.

## Cerințe Tehnice

- Sistem de operare: **Linux**.
- Shell: **Bash**.
- Limbaj de programare: **Python**.
- Protocol: **SSH/SCP** (pentru conectare securizată).
- Virtualizare: **KVM și QEMU**.
- Server DHCP: dnsmasq

## Limitări

- Configurarea nodurilor **nu** adaugă automat hostname-uri în fișierul `/etc/hosts`.
- Sistemul de *logging* (monitorizare) este **probabil limitat**.

## Plan de Evaluare

- Dezvoltarea unui **script de teste** bazat pe cazuri de utilizare (use case-uri).

## 3. Design

### Explicații de Design

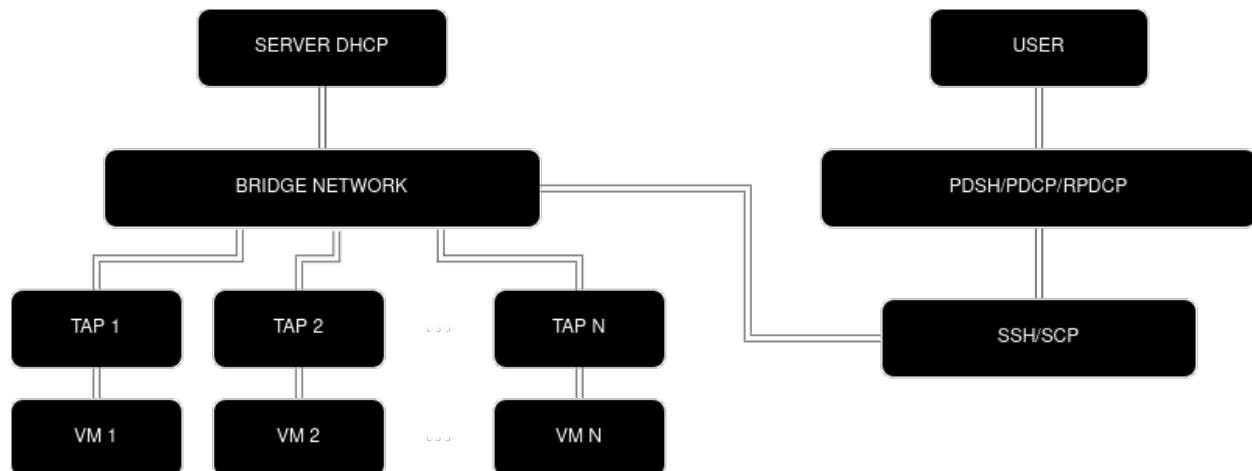


Figure 1: Diagram

- **Rețeaua Bridge și DHCP:** Se folosește un **bridge network** cu un **server DHCP** pentru a asigura un **IP static și unic** pentru fiecare VM în mod automat.
  - *Motivație:* Fără un IP unic și cunoscut, VM-urile nu pot fi accesate.
- **Alegerea Metodei:** Această metodă a fost aleasă în detrimentul `qemu-guest-agent` deoarece permite o **conectare universală** atât la VM-uri, cât și la sisteme externe, din afara specificațiilor inițiale ale proiectului.

## Descrierea Mecanismelor

- **DHCP Server:** Rulează pe `bridge network` pentru a forța VM-urile să obțină un **IP static și unic**, făcându-le accesibile pentru **USER**.
- **Comenzi PDSH/PDCP/RPDCP:** Utilizează protocolele `SSH/SCP` pentru o conectare securizată și **paralelă** la toate nodurile.
  - **Python (sau AWK)** este folosit pentru **manipularea și prelucrarea argumentelor** primite de scripturile Bash.

## 4. Implementare

### Stack Tehnologic

- **Python (AWK?)**: Pentru manipularea sirurilor de caractere (stringuri).
- **SSH/SCP**: Conectare securizată la noduri.
- **QEMU**: Interfață pentru rularea VM-urilor folosind **KVM**.
- **Alpine Linux**: Distribuție *lightweight* aleasă pentru VM-uri.
- **Dnsmasq**: Pentru crearea serverului DHCP.
- **Git/GitHub**: Pentru controlul versiunilor (*version control*).

### Probleme Apărute & Soluții

| Problemă                                 | Soluție  |
|--|--|
| Atribuirea IP-urilor unice pentru VM-uri | Obținerea unui IP static (Soluție: <code>got good</code> ) |
| Serverul DHCP blocat de firewall         | Adăugarea unei excepții în firewall pentru serviciul DHCP  |

## MOD DE FOLOSIRE

!!! Funcționalitate limitată, work in progress !!!

### Setup

#### `copyVM`

Scriptul `./copyVM` creează un număr specificat de copii ale unei imagini VM date. Clonele sunt indexate de la 1 la n. În cazul în care scriptul este rulat de mai multe ori,

indexul continuă de la ultimul clone creat.

```
./copyVM -i [path/to/image] -n [number0fClones]
```

-i : locația imaginii VM

-n : numărul de clone

## startVMs

Scriptul ./startVMs pornește toate VM-urile dintr-un director specificat.

De asemenea, permite configurarea unui **bridge network** și a unor **tap-uri** pentru fiecare nod VM.

!!! Important

Pentru a te asigura că fiecare nod VM primește automat un **ip static**, trebuie să ai un **server dhcp** care rulează pe **bridge network**.

Scriptul nu include mapări în /etc/hosts pentru fiecare nod (nu este planificat momentan, sorry :3).

**Bridge network-ul** nu este persistent.

```
./startVMs -d [directory] -ns -b [bridgeName]
```

-d : locația nodurilor

-ns : setup pentru bridge network (dacă este omis, va fi necesar setup manual pentru fiecare nod)

-b : numele pentru bridge network (dacă este omis, valoarea implicită este **megatron**)

## Usage

### mypdsh

Scriptul ./mypdsh permite rularea de comenzi tip bash pe multiple noduri în paralel prin o singura comanda.

```
./mypdsh -w [hostnames] -x [hostnames] [commands]
```

-w : specifica nodurile la care dorim executia comenzilor

-x : specifica nodurile care să fie excluse din executie

%u : username la care se conecteaza ssh

%h : hostname nod

## **mypdcp**

Scriptul ./mypdcp copiaza fisiere de pe systemul local pe noduri

```
./mypdcp -w [hostnames] -x [hostnames] [localFile] [remotePath]
```

-w : specifica nodurile la care dorim transferul

-x : specifica nodurile care sa fie excluse din transfer

%u : username la care se conecteaza ssh

%h : hostname nod

## **myrpdcp**

Scriptul ./myrpdcp este procesul mypdcp inversat. Transfera fisiere remote pe masina locala

```
./myrpdcp -w [hostnames] -x [hostnames] [localFile] [remotePath]
```

-w : specifica nodurile la care dorim transferul

-x : specifica nodurile care sa fie excluse din transfer

%u : username la care se conecteaza ssh

%h : hostname nod