



(12) CERERE DE BREVET DE INVENȚIE

(21) Nr. cerere: a 2015 00006

(22) Data de depozit: 09/01/2015

(41) Data publicării cererii:
30/08/2016 BOPI nr. 8/2016

(71) Solicitant:
• IXIA, A CALIFORNIA CORPORATION,
26601 WEST AGOURA ROAD,
CALABASAS, CA, US

(72) Inventatori:
• CORNEA TUDOR,
STR.VASILE GHERGHEI NR.98,
BUCUREȘTI, B, RO;

• COMĂNESCU GEORGE,
STR.MARINARILOR NR.23-25, BL.5/2,
AP.26, BUCUREȘTI, B, RO;
• STANCIU ADRIAN, ȘOS.COLENTINA
NR.16, BL.B3, AP.51, SECTOR 2,
BUCUREȘTI, B, RO;
• BELOV KONSTANTIN, 2697
REGINA AVENUE, THOUSAND OAKS,
CALIFORNIA, US

(74) Mandatar:
RATZA ȘI RATZA SRL, B-DUL A.I. CUZA,
NR. 52-54, SECTOR 1, BUCUREȘTI

(54) METODE, SISTEME ȘI SUPTOR CITIBIL PE CALCULATOR
PENTRU IMPLEMENTAREA UNEI MAȘINI VIRTUALE (VM)

(57) Rezumat:

Invenția se referă la o metodă, la un sistem și la un suport citibil de către calculator, pentru implementarea unei mașini virtuale. Metoda conform invenției include o etapă (302) de recepționare a unei cereri de creare a unei mașini virtuale (VM), și o etapă (304) de creare a mașinii virtuale (VM), folosind un sistem de operare (OS) a mașinii virtuale, stocat într-o memorie read-only. Sistemul pentru implementarea unei mașini virtuale, conform invenției, cuprinde o memorie read-only și o platformă de calcul configurată pentru a recepționa o cerere de creare a unei mașini virtuale (VM), și pentru a crea mașina virtuală (VM) folosind un sistem de operare (OS) al mașinii virtuale, stocat în memoria read-only. Suportul citibil de către calculator, conform invenției, este un suport netranzitoriu, având stocate instrucțiuni executabile de către calculator, care, atunci când sunt executate de către un procesor, efectuează etapele metodei conform invenției.

Revendicări: 20
Figuri: 3

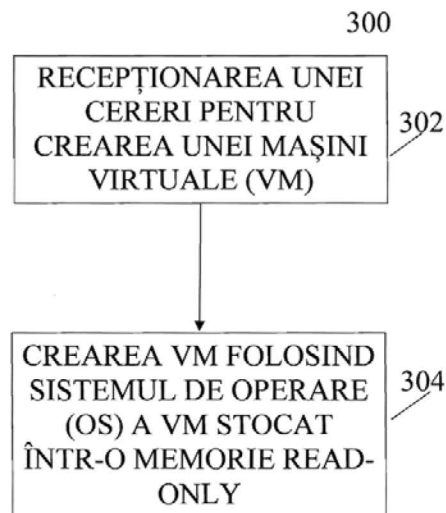
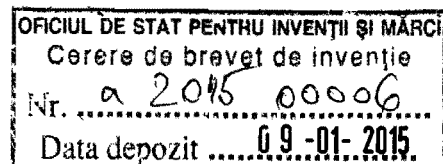


Fig. 3





DESCRIERE

METODE, SISTEME SI SUPORT CITIBIL DE CALCULATOR PENTRU IMPLEMENTAREA UNEI MAȘINI VIRTUALE (VM)

5

DOMENIUL TEHNIC

Obiectele prezentei invenții se referă la informatizarea virtualizării. Mai precis, obiectele se referă la metode, sisteme si suport citibil de calculator pentru implementarea unei mașini virtuale (VM)

10

STADIUL ANTERIOR AL TEHNICII

Un centru de date este o facilitate utilizată la sistemele de calcul personale și la componentele asociate (de exemplu, sistemele de stocare). Centrele de date oferă, de obicei, fiabilitate și securitate ridicată și de obicei includ resurse partajate de mai mulți clienți ai operatorului centrului de date. Centrele mari de date oferă operațiuni la scară industrială care folosesc energie cât un oraș mic. Diferite centre de date pot utiliza virtualizarea. De exemplu, un server fizic într-un centru de date poate implementa mai multe mașini virtuale (VM), de exemplu, servere virtuale, care folosesc un server sau un nod fizice, în centrul de date. În acest exemplu, fiecare VM poate executa un sistem de operare și alte software-uri, caz în care fiecare VM poate apărea ca un server fizic pentru utilizatorii finali.

În general, atunci când sunt implementate una sau mai multe mașini virtuale pe un server fizic, este folosit un hipervisor pentru a gestiona și a facilita VM-ile. De exemplu, un hipervisor poate imita diferite caracteristici hardware disponibile la VM-uri. În acest exemplu, software-ul (de exemplu, un sistem de operare), care, executat pe o VM, poate avea acces la hardware, cum ar fi la video, tastatură, mediu de stocare și/sau la interfețele de rețea, imitate de hipervisor. De asemenea, hipervisorul poate spara VM-ile, una de alta, astfel că o operație în interiorul unei VM, este menținută în cadrul acelei VM și nu este vizibilă sau modificabilă de la o altă VM.

În general, implementarea VM convențională are probleme de viteză și de scalabilitate. În special, implementarea VM convențională necesită, în general, o cantitate semnificativă de timp, resurse, și/sau expertiză.

În consecință, există o necesitate de metode, sisteme si suport citibil de calculator pentru implementarea unei mașini virtuale (VM), îmbunătățite.

EXPUNEREA PE SCURT A INVENȚIEI

Sunt dezvăluite metode, sisteme și suport citibil de calculator pentru implementarea unei mașini virtuale (VM). Conform invenției, metoda include recepționarea unei cereri de a crea o VM. Metoda, de asemenea, mai include crearea VM folosind un sistem de operare a VM (OS), stocat într-o memorie read-only.

Conform invenției, sistemul include o memorie read-only și o platformă de calcul configurate pentru a recepționa o cerere de crearea unei VM și pentru a crea VM folosind un sistem de operare VM OS stocat în memoria read-only.

Obiectele prezentei invenții pot fi implementate în software-ul în combinație cu hardware-ul și/sau firmware-ul. De exemplu, obiectele prezentei invenții pot fi implementate în software-ul executat de un procesor. Conform unui exemplu de implementare, obiectele prezentei invenții pot fi implementate folosind un suport non-tranzitoriu care poate fi citit de calculator, care are stocate instrucțiuni executabile de calculator, care atunci când sunt executate de către procesor, comandă calculatorul să efectueze etapele metodei. Suportul citibil de calculator, adecvat pentru punerea în aplicare a obiectelor prezentei invenții, include dispozitive non-tranzitorii, cum ar fi dispozitive de memorie pe disc, dispozitive de memorie cip, dispozitive logice programabile, circuite integrate digitale configurabile de către utilizator și circuite integrate specifice aplicației. În plus, un suport care poate fi citit de calculator, care implementează obiectele prezentei invenții, poate fi amplasat pe un singur dispozitiv sau platformă de calcul sau poate fi distribuit pe mai multe dispozitive sau platforme de calcul.

Așa cum este utilizat în prezenta descriere, termenul 'nod' se referă la o platformă fizică care include mai multe procesoare, interfețe de rețea și memorie.

Așa cum sunt utilizați în prezenta descriere, fiecare dintre termenii "funcția", "motor" și "modul" se referă la hardware, firmware sau software în combinație cu hardware și/sau firmware pentru implementarea caracteristicilor descrise aici.

DESCRIEREA PE SCURT A DESENELOR EXPLICATIVE

Obiectele prezentei invenții vor fi explicate în continuare în legătură cu figurile anexate, care reprezintă:

Figura 1 este o diagramă care ilustrează o platformă de calcul pentru implementarea unei mașini virtuale (VM), în conformitate cu un exemplu de realizare a prezentei invenții;

Figura 2 este o diagramă care ilustrează comunicațiile asociate cu implementarea VM, în conformitate cu un exemplu de realizare a prezentei invenții; și

Figura 3 este o diagramă care ilustrează o metodă de implementarea VM, în conformitate cu un exemplu de realizare a prezentei invenții.

DESCRIEREA DETALIATĂ

5 Obiectele prezentei invenții se referă la sisteme și suport citibil de calculator pentru implementarea unei mașini virtuale (VM). Implementarea VM convențională implică de obicei un hipervizor sau o altă entitate de crearea uneia sau mai multor VM, unde fiecare VM este alocată unei porțiuni separate de memorie, pentru stocarea unui sistem (OS) de operare a VM și/sau a altor informații. Cu toate acestea, pot apărea probleme, atunci când numărul de
10 VM este semnificativ. În special, un VM OS și/sau alte informații pot fi copiate și stocate într-o porțiune de memorie relevantă pentru fiecare VM, ceea ce poate avea un impact semnificativ asupra timpului de implementare și asupra resurselor de memorie, pe măsură ce numărul mașinilor virtuale crește.

În conformitate cu unele aspecte ale prezentei invenții, tehnicile de implementare
15 VM pot include crearea de una sau mai multe mașini virtuale VM care folosesc memoria read-only (RO) pentru anumite date (de exemplu, un VM OS) și, eventual, memoria citire-scriere (read-write (RW)) pentru alte date (de exemplu, parole, script-uri, informații de configurare, și/sau informații persistente). În unele exemple de realizare, memoria RO poate fi partajată între o multitudine de mașini virtuale. În unele exemple de realizare, memoria RW
20 poate fi un disc virtual de orice dimensiuni creat prin tehnologia 'thin provisioning', astfel încât spațiul de stocare să fie alocat funcție de necesități.

În conformitate cu unele aspecte ale prezentei invenții, pot fi utilizate diverse optimizări pentru îmbunătățirea creerii de VM și/sau a timpului de implementare. De exemplu, folosind o memorie RO partajată (SRAM) și memorii RW prin tehnologia 'thin
25 provisioning, un timp de implementare VM asociat cu aspectele prezentei invenții, poate fi aproape un ordin de mărime (de exemplu, șapte până la zece ori) mai rapid. Mai mult, astfel de optimizare nu este limitată la un anumit mediu sau hardware și se poate aplica în orice scenariu de implementare, în cazul în care VM-uri similare urmează să fie implementate rapid și/sau cu impact minim asupra resurselor de memorie (de exemplu, spațiul de stocare).

30 În conformitate cu unele aspecte ale prezentei invenții, tehnicile de implementare VM pot include actualizarea uneia sau mai multor mașini virtuale VM care folosesc memoria RO, prin modificarea memoriei RO, de exemplu, prin modificarea unui VM OS stocat într-un SRAM, prin intermediul unui hipervizor. Prin modificarea memoriei RO (de exemplu, prin intermediul unui hipervizor), conectivitatea cu una sau mai multe mașini virtuale VM poate fi

opțională în procesul de actualizare, permițând astfel ca actualizările VM să fie efectuate mai eficient.

Se vor detalia în continuare exemple de realizare ilustrative ale obiectelor prezentei invenții, exemple care sunt ilustrate în Figurile anexate. Ori de câte ori este posibil, vor fi utilizate aceleași numere de referință din Figuri, care să se refere la același element sau la părți similare.

Figura 1 este o diagramă care ilustrează o platformă de calcul 100 pentru implementarea unei VM, în conformitate cu un exemplu de realizare a prezentei invenții. Platforma de calcul 100 poate reprezenta orice entitate sau entități corespunzătoare (de exemplu, o platformă de testare, un instrument de testare, un dispozitiv, un nod, și/sau una sau mai multe platforme de calcul) pentru furnizarea de diferite resurse legate de virtualizare, cum ar fi resursele legate de unul sau mai multe servere, hipervizori, VM-uri, routere virtuale, switch-uri virtuale, și/sau alt "nor" ("cloud") sau centru de date. În unele exemple de realizare, platforma de calcul 100 poate fi configurată să efectueze unul sau mai multe aspecte legate de testarea unuia sau mai multor dispozitive, rețele sau servicii. În unele exemple de realizare, platforma de calcul 100 poate fi un instrument de sine stătător, un dispozitiv de testare, sau software executat pe un procesor sau pe mai multe procesoare. În unele exemple de realizare, platforma de calcul 100 poate fi un singur nod sau poate fi distribuită în mai multe platforme de calcul sau noduri.

În unele exemple de realizare, platforma de calcul 100 poate include funcționalitatea de a comunica cu motorul de configurare 106 și/sau cu diverse entități. Motorul de configurare 106 poate reprezenta orice entitate corespunzătoare (de exemplu, un nod, modul, sau dispozitiv) pentru generarea și/sau transmiterea de informații de configurare asociate cu VM-urile sau a informațiilor legate de resurse. De exemplu, motorul de configurare 106 poate furniza o interfață sau consolă de comunicație pentru comunicația cu un utilizator (de exemplu, un operator de testare sau un administrator VM). În acest exemplu, utilizatorul poate selecta și/sau determina diverse setări legate de VM și/sau configurații similare pentru implementarea VM, de exemplu, de la platforma de calcul 100.

Motorul de configurare 106 poate include o memorie 112. Memoria 112 poate reprezenta orice entitate corespunzătoare (de exemplu, un suport non-tranzitoriu care poate fi citit de calculator sau un dispozitiv de memorie) pentru stocarea datelor asociate cu implementarea VM și/sau a informațiilor legate de configurarea VM. De exemplu, memoria 112 poate include un sistem de operare VM OS 124 și/sau alte informații. În acest exemplu, atunci când se inițiază crearea VM și/sau actualizarea VM, motorul de configurare 106 poate

trimite VM OS 124 și/sau alte informații stocate în memoria 112 de la platforma de calcul 100.

Platforma de calcul 100 poate include un hipervizor 118, o mașină virtuală VM 1 120, și o mașină virtuală VM 2 122. Hipervizorul 118 poate reprezenta orice entitate adecvată pentru gestionarea (de exemplu, crearea, ștergerea, migrarea, repornirea, și/sau oprirea VM-
5 lor) și/sau facilitarea VM 1 120 și VM 2 122. De exemplu, hipervizorul 118 poate aloca și/sau imitata resursele fizice asociate cu VM 1 120 și cu VM 2 122. În acest exemplu, hipervizorul 118 poate fi configurat pentru a recepționa comenzi de la un controler hipervizor (de exemplu, motorul de configurare 106) sau de la o altă entitate care instruește hipervizorul 118 cu
10 privire la modul de a gestiona și/sau configura VM 1 120 și VM 2 122.

Fiecare dintre VM 1 120 și VM 2 122 poate reprezenta orice entitate adecvată (de exemplu, software-ul executat pe un procesor) pentru imitarea unui sistem de calcul. Fiecare dintre VM 1 120 și VM 2 122 poate fi asociată cu resursele fizice și/sau virtuale pentru executarea unui sistem de operare VM OS și/sau a altui software asociat cu imitarea unui
15 sistem de calcul și/sau cu efectuarea de funcționalități conexe. De exemplu, hipervizorul 118 poate configura VM 1 120 și VM 2 122 pentru imitarea de servere web și/sau pentru efectuarea funcționalităților serverului web, de exemplu, recepționarea cererilor prin protocolul HTTP (hypertext transfer protocol) și oferirea de răspunsuri HTTP.

VM 1 120 poate include sau accesa un disc logic 110. Discul logic 110 poate reprezenta orice entitate adecvată pentru furnizarea stocării informațiilor asociate cu VM 1
20 120. De exemplu, discul logic 110 poate reprezenta un dispozitiv logic care conține una sau mai multe porțiuni ale unuia sau mai multor dispozitive fizice de memorie. În acest exemplu, fiecare porțiune de disc logic 110 poate fi configurată diferit și/sau poate cuprinde diferite tipuri de medii de stocare, de exemplu, memorie cu acces aleatoriu (RAM), memorie flash, discuri compacte (CD-uri) sau discuri fizice.
25

Discul logic 110 poate include o SROM 102, de asemenea, menționată ca un disc logic RO, și o memorie RW 108, de asemenea, menționată ca un disc logic RW. SROM 102 poate reprezenta orice entitate adecvată (de exemplu, unul sau mai multe suporturi care pot fi citite de calculator sau porțiuni ale acestora) pentru stocarea datelor. În unele realizări, SROM
30 102 poate fi partajată între mai multe mașini virtual VM, de exemplu, VM 1 120 și VM 2 122. În unele aplicații concrete, SROM 102 poate fi configurată să permită sau să efectueze operații de citirea memoriei, dar nu și operații de scrierea memoriei. De exemplu, SROM 102 poate fi montată ca și un spațiu de stocare read-only, astfel încât nu există date stocate în aceasta care să poată fi modificate și nu există date suplimentare care să se poată scrie în

SROM 102. În acest exemplu, **SROM 102** poate evita coruperea sistemului de fișiere sau alte probleme asociate cu stocarea care permite operații de scriere memorie.

În unele exemple de realizare, **SROM 102** poate fi un disc virtual de orice dimensiuni creat prin tehnologia ‘thin provisioning’. De exemplu, hipervizorul **118** poate prevedea **VM 1 120** cu doi gigabyte (GB) de **SROM 102**. În acest exemplu, spațiul de stocare asociat cu **SROM 102** poate fi rezervat sau alocat și/sau formatat, de exemplu, cu zerouri leneși sau dinamici’ (lazy zeros or eager zeros), spațiu de stocare fizic, la crearea și/sau înainte de utilizare. Într-un alt exemplu, în cazul în care **SROM 102** este partajată între o multitudine de VM-uri, hipervizorul **118** poate configura **VM 1 120** să utilizeze o **SROM 102** preexistentă (de exemplu, pre-alocată și/sau pre-formatată).

Memoria **RW 108** poate reprezenta orice entitate adecvată (de exemplu, unul sau mai multe suporturi care pot fi citite de calculator sau porțiuni ale acestora) pentru stocarea datelor asociate cu **VM 1 120**. Memoria **RW 108** poate fi configurată pentru a permite sau a efectua operații de citirea memoriei și operații de scriere memorie. De exemplu, pot fi scrise parole asociate cu performanța VM, la memoria **RW 108** și care pot fi citite din memoria **RW 108**.

VM 2 120 poate include sau accesa un disc logic **116**. Discul logic **116** poate reprezenta orice entitate adecvată pentru care efectuează stocare de informații asociate cu **VM 2 122**. De exemplu, discul logic **116** poate reprezenta un dispozitiv logic care conține una sau mai multe porțiuni ale unuia sau mai multor dispozitive fizice de memorie. Discul logic **116** poate include o memorie **RW 114** și o **SROM 102**. În unele aplicații concrete, **SROM 102** poate fi partajată între **VM 1 120** și **VM 2 122**. În unele aplicații concrete, fiecare dintre **VM 1 120** și **VM 2 122** poate include o memorie **RO** separată sau distinctă care poate efectua operații de citire memorie. Memoria **RW 114** poate reprezenta orice entitate adecvată (de exemplu, unul sau mai multe suporturi care pot fi citite de calculator sau porțiuni ale acestora) pentru stocarea datelor asociate cu **VM 2 122**. Memoria **RW 114** poate fi configurată pentru a permite sau pentru a efectua operații de citire memorie și operații de scriere memorie.

În unele exemple de realizare, memoria **RW 108** și/sau memoria **RW 114** pot fi un disc virtual de orice dimensiuni creat prin tehnologia ‘thin provisioning’, astfel că spațiul de pe disc nu poate fi alocat în avans. De exemplu, la etapa de creare VM, hipervizorul **118** poate prevedea **VM 1 120**, cu patru GB de memorie **RW 108**; dar nu poate aloca de fapt patru GB de memorie fizică la **VM 1 120**. În schimb, în acest exemplu, hipervizorul **118** poate aloca 200 de megabyte (MB) din cei patru GB prevăzuți și poate aloca mai mult spațiu fizic de memorie, după cum este necesar, de exemplu, în pachete de 200 MBs, astfel că utilizarea memoriei atinge un prag de utilizare (de exemplu, 90% din memoria **RW** este folosită).

În unele exemple de realizare, memoria RW 108 și/sau memoria RW 114 pot fi amplasate pe un alt dispozitiv de stocare și/sau pe un tip diferit de spațiu de stocare decât SROM 102. De exemplu, hipervizorul 118 poate crea 20 de VM-uri, în care fiecare VM utilizează SROM 102, dar fiecare folosește o memorie separată RW. În acest exemplu, SROM 102 poate include o memorie flash sau RAM, în timp ce fiecare memorie RW poate include o porțiune din spațiul de stocare asociată cu un disc fizic. Într-un alt exemplu, în cazul în care fiecare VM utilizează SROM 102, dar fiecare VM utilizează o memorie separată RW, SROM 102 poate include spațiu fizic de stocare pe disc și fiecare memorie RW poate include o porțiune din spațiul de stocare în memoria flash. În acest exemplu, fiecare memorie RW poate fi un disc virtual de orice dimensiuni creat prin tehnologia 'thin provisioning' și cu dimensiunea discului asociată cu fiecare memorie RW, și poate depinde de cât de multe informații se scrie. În unele variante, dacă utilizarea spațiului de stocare pentru o anumită memorie RW devine prea mare, poate fi alocat un spațiu de stocare suplimentar, de exemplu, de la un dispozitiv diferit de stocare sau de la un alt tip de dispozitiv de stocare.

În unele exemple de realizare, memoria RW 108, memoria RW 114, și/sau SROM 102 pot include memorie persistentă, memorie non-volatilă, și/sau memorie configurată pentru a stoca informații persistente. De exemplu, memoria RW 108 poate stoca informații care pot fi preluate, chiar dacă VM 1 120 și/sau platforma de calcul 100 sunt resetate (de exemplu, repornite).

În unele exemple de realizare, memoria RW 108, memoria RW 114, și/sau SROM 102 pot include memorie non-persistentă, memorie volatilă, și/sau memorie configurată pentru a stoca informațiile pentru o scurtă perioadă de timp. De exemplu, memoria RW 108 poate stoca informații care pot fi extrase în timpul unei sesiuni VM curente, dar care nu pot fi recuperate dacă VM 1 120 și/sau platforma de calcul 100 s-au resetat (de exemplu, repornite).

În unele exemple de realizare, SROM 102 poate include sau stoca VM OS 104 și/sau alte informații utilizabile de VM 1 120 și/sau VM 2 122. De exemplu, în cazul în care SROM 102 este partajată de către VM 1 120 și VM 2 122, fiecare VM poate executa VM OS 104 sau versiuni ale acestuia din SROM 102.

În unele exemple de realizare, VM OS 104 și/sau alte software-urile legate de VM pot fi stocate ca o imagine sau ca un fișier în SROM 102 sau ca un depozit. De exemplu, VM OS 104 poate reprezenta o anumită versiune sau tip de sistem de operare Linux, cum ar fi CentOS, și/sau componentele (de exemplu, software) pentru efectuarea uneia sau mai multor funcții legate de VM.

În unele exemple de realizare, în cazul în care mai multe mașini virtuale VM partajează (de exemplu, bootează din) SROM 102, resursele de memorie pot fi conservate. De exemplu, dacă motorul de configurare 106 instruește hipervizorul 118 să genereze o sută de mașini virtuale, care includ un GB de date identice, hipervizorul 118 poate stoca un GB de date în SROM 102, și poate configura fiecare din cele o sută VM-uri să folosească SROM 102. În acest exemplu, prin partajarea SROM 102, hipervizorul 118 poate conserva aproximativ 99 GB de date în comparație cu cantitatea de stocare utilizată dacă hipervizorul 118 copiază un GB de date, într-un spațiu de stocare separat pentru fiecare VM.

În unele exemple de realizare, în cazul în care mai multe VM partajează SROM 102, crearea VM și/sau timpul de implementare pot fi reduse în mod semnificativ. De exemplu, prin partajarea SROM 102 între mai multe mașini virtuale VM, hipervizorul 118 poate economisi timp prin faptul că nu trebuie să copieze informațiile stocate în SROM 102 într-o zonă de stocare separată pentru fiecare VM creată. În acest exemplu, presupunând că 100 de VM urmează să fie create și că menționata copiere a informațiilor stocate în SROM 102 necesită zece secunde pentru fiecare operație de copiere, hipervizorul 118 poate finaliza procedura de creare VM cu 990 secunde (de exemplu, în jur de șaisprezece minute) mai rapid decât tehnicile care folosesc spații de stocare separate.

În unele exemple de realizare, în cazul în care mai multe mașini virtuale partajează SROM 102, timpul necesar pentru efectuarea unei actualizări a VM poate fi semnificativ redus. De exemplu, în timpul unui proces de actualizare pentru VM 1 120 și VM 2 122, hipervizorul 118 poate obține (de exemplu, de la motorul de configurare 106) un alt sistem de operare VM OS 124 și poate suprascrive și/sau înlocui VM OS 104 stocat în SROM 102 cu sistemul de operare VM OS 124. În acest exemplu, conectivitatea cu VM 1 120 sau VM 2 122, poate să nu fie necesară. În schimb, după modificarea SROM 102, fiecare VM poate fi repornită (de exemplu, prin hipervizorul 118), astfel încât, după repornire, fiecare VM utilizează VM OS 124 sau o versiune a acestuia. Prin faptul că nu există comunicație între VM 1 120 și VM 2 122, procesul de actualizare poate fi completat mai eficient, deoarece o entitate de cerere actualizare (de exemplu, motorul de configurare 106) comunică cu o singură entitate (de exemplu, hipervizorul 102), în loc să comunice cu fiecare VM, separat.

În unele exemple de realizare, una sau mai multe tehnici de unificare pot fi folosite pentru a prezenta o memorie RO (ex SROM 102) și o memorie RW (de exemplu, memorie RW 108), ca un singur sistem de fișiere, un dispozitiv de stocare, un disc, și/sau o structură de directori. De exemplu, VM 1 120 și/sau VM OS 104 pot fi configurate să utilizeze o montare unificată, de exemplu, un sistem de fișiere UnionFS, pentru a prinde împreună SROM 102 și

memoria RO **108** ca și o singură structură director, unificată. În acest exemplu, VM 1 **120** și/sau VM OS **104** pot monitoriza, și, dacă este necesar, pot redirecționa operațiile legate de memorie la memoria RW **108**, de exemplu, astfel că utilizatorii nu sunt conștienți de diferitele capacități sau configurații ale memoriilor care stau la bază.

5 În unele exemple de realizare, prin utilizarea unei montări unificate și/sau alte tehnici unificatoare, operațiile de scrierea memoriei pot fi efectuate (de exemplu, prin intermediul memoriei RW **108** sau memoriei RW **114**) pentru fiecare VM, permițând astfel unui utilizator să personalizeze o anumită VM, și să nu corupă alte mașini virtuale. Mai mult, prin utilizarea unor astfel de tehnici unificatoare, un utilizator poate efectua personalizarea VM (de exemplu,
10 scrierea de script-uri de testare utilizator sau menținerea parolelor de performanță), fără să știe un punct de montare asociat cu memoria RW respectivă.

Se va aprecia că Figura 1 este în scop ilustrativ și că diverse entități descrise, locațiile lor, și/sau funcțiile lor, descrise mai sus în legătură cu Figura 1, pot fi schimbate, modificate, adăugate sau eliminate. De exemplu, motorul de configurare **106** poate fi situat la platforma
15 de calcul **100** sau la un nod separat și/sau distinct de platforma **100** de calcul.

Figura 2 este o diagramă care ilustrează comunicațiile asociate cu implementarea VM, în conformitate cu un exemplu de realizare a obiectelor prezentei invenții. În unele exemple de realizare, motorul de configurare **106** sau altă entitate (de exemplu, un operator de testare sau utilizator) poate interacționa cu platforma de calcul **100**. De exemplu, motorul de
20 configurare **106** poate genera comenzi care instruește hipervizorul **118** pentru a efectua implementarea VM și/sau alte acțiuni, cum ar actualizarea VM și/sau migrațiile VM.

Conform cu Figura 2, la etapa **2001**, un mesaj pentru creerea unei sau mai multor mașini virtuale VM poate fi trimis de la motorul de configurare **106** la hipervizorul **118**. De exemplu, un mesaj de crearea VM (ilor) poate indica faptul că două VM urmează să fie
25 create, în care fiecare VM este pentru a executa VM OS **104** sau o versiune a acestuia.

La etapa **2002**, ca răspuns la recepționarea mesajului, hipervizorul **118** poate crea una sau mai multe mașini virtuale, în care fiecare dintre una sau mai multe VM poate utiliza o memorie RO (ex SROM **102**) pentru stocarea VM OS **104** și o memorie RW pentru stocarea datelor de utilizator și/sau a altor informații. De exemplu, hipervizorul **118** poate crea VM 1
30 **120** configurată să utilizeze SROM **102** și memoria RW **108** și poate crea VM 2 **122** configurată să utilizeze SROM **102** și memoria RW **114**.

La etapa **2003**, după ce una sau mai multe VM sunt create și/sau implementate, poate fi trimis un mesaj pentru recunoașterea sau raportarea creerii uneia sau mai multor VM, de la hipervizorul **118** la motorul de configurare **106**.

La etapa **2004**, un mesaj de actualizarea uneia sau mai multor mașini virtuale VM poate fi trimis de la motorul de configurare **106** la hipervizorul **118**. De exemplu, un mesaj de actualizarea VM (ilor) poate indica faptul că două VM urmează să fie actualizate cu VM OS **124**.

5 La etapa **2005**, ca răspuns la recepționarea mesajului, hipervizorul **118** poate actualiza una sau mai multe VM, prin înlocuirea tuturor variantelor sistemului de operare VM OS cu un sistem de operare VM OS diferit. De exemplu, hipervizorul **118** poate înlocui VM OS **104** în SROM **102** cu VM OS **124**. În acest exemplu, după înlocuirea VM OS **104** în SROM **102** cu VM OS **124**, hipervizorul **118** poate reporni orice mașină virtuală VM afectată, astfel încât, 10 atunci când repornește, fiecare VM folosește VM OS **124**. Întrucât hipervizorul **118** modifică informațiile stocate în SROM **102**, una sau mai multe VM pot fi actualizate fără necesitatea ca VM-urile să comunice cu motorul de configurare **106**.

La etapa **2006**, după ce una sau mai multe VM sunt actualizate, un mesaj pentru recunoașterea sau raportarea actualizării uneia sau mai multor VM poate fi trimis de la 15 hipervizorul **118** la motorul de configurare **106**.

Se va aprecia că aceste comunicații și/sau acțiuni reprezentate în Figura 2 sunt în scopi ilustrativ și că comunicații și/sau acțiuni diferite și/sau suplimentare în raport cu cele prezentate în Figura 2 pot fi utilizate pentru implementarea VM și/sau actualizarea VM. Se va aprecia, de asemenea, că diversele comunicații și/sau acțiuni descrise aici, pot apărea simultan 20 sau într-o ordine sau secvență diferită.

Figura 3 este o diagramă care ilustrează un exemplu de metodă **300** pentru implementarea unei VM, în conformitate cu un exemplu de realizare a prezentei invenții. În unele exemple de realizare, metoda **300**, sau porțiuni din aceasta, poate fi efectuată de către sau la platforma de calcul **100**, motorul de configurare **106**, și/sau un alt nod sau modul. În 25 unele exemple de realizare, metoda **300** poate include etapele **302** și/sau **304**.

Conform metodei **300**, la etapa **302**, poate fi recepționată o cerere de a crea o VM. De exemplu, hipervizorul **118** sau o altă entitate, la platforma de calcul **100**, poate recepționa o cerere pentru crearea VM **1 120** de la motorul de configurare **106**.

La etapa **304**, VM poate fi creată folosind un VM OS stocat într-o memorie RO. De 30 exemplu, hipervizorul **118** poate crea VM **1 120** și poate configura VM **1 120** să partajeze SROM **102** și/sau să execute variante VM OS **104** stocat în ea.

În unele exemple de realizare, o VM (ex VM **1 120** și/sau VM **2 122**) poate fi configurată pentru a utiliza o memorie RW (de exemplu, memoria RW **108** și/sau memoria RW **114**) pentru stocarea informațiilor de utilizator, parolelor, script-urilor de utilizator sau

informațiilor persistente. De exemplu, hipervizorul 118 poate crea VM 1 120, poate configura VM 1 120 să utilizeze SROM 102 pentru executarea variantelor de VM OS 104 stocat. În acest exemplu, VM 1 120 poate fi, de asemenea, configurată pentru a utiliza o memorie RW separată și/sau distinctă pentru stocarea datelor, de exemplu, parolele de sistem sau informațiile generate de utilizator.

În unele variante, o VM (ex VM 1 120 și/sau VM 2 122) poate fi actualizată prin modificarea unei memorii RO (ex SROM 102 sau altă memorie RO). În astfel de exemplu, conectivitatea cu VM poate fi opțională, în timpul actualizării. De exemplu, motorul de configurare 106 poate trimite VM OS 124 la hipervizorul 118 și hipervizorul 118 poate înlocui VM OS 104 stocat în SROM 102 cu VM OS 124. În acest exemplu, motorul de configurare 106 nu poate comunica în mod direct cu orice mașină virtuală afectată. În schimb, motorul de configurare 106 poate solicita sau instrui hipervizorul 118 pentru a actualiza VM-ile și hipervizorul 118 poate reporni VM-ile după efectuarea actualizării.

În unele exemple de realizare, o VM (ex VM 1 120 și/sau VM 2 122) poate fi configurată să utilizeze o montare unificată sau alte tehnici pentru a prezenta o memorie RO și o memorie RW ca un singur disc logic sau un sistem de fișiere, pentru un utilizator. De exemplu, prin utilizarea unei montări unificate, VM 1 120 poate prezenta SROM 102 și RW 108 ca un sistem de fișiere și/sau zonă de stocare unice. În acest exemplu, din perspectiva unui utilizator, utilizatorul pare a fi interacțiunea cu un singur sistem de fișiere sau cu o singură zonă de stocare.

În unele exemple de realizare, operațiile de scriere memorie sunt redirecționate către o memorie RW (de exemplu, memoria RW 108 și/sau memoria RW 114). De exemplu, sistemul de operare VM OS 104 sau altă entitate asociată cu VM 1 120 pot fi configurate pentru a redirecționa automat operațiile de scriere memorie la memoria RW 108. În acest exemplu, VM OS 104 sau altă entitate asociată cu VM 1 120 pot permite ca operațiile citire memorie să fie gestionate de memoria apropiată, de exemplu, fie memoria RW 108 fie SROM 102. Într-un alt exemplu, în loc de a folosi montarea unificată (de exemplu, în cazul în care VM OS 104 redirecționează operațiile de scriere memorie), aplicațiile individuale asociate cu VM 1 120 pot să fie conștiente de memoria RW 108 și SROM 102 și pot direcționa sau redirecționa operațiile de scriere memorie și/sau de citire memorie în zona de stocare corespunzătoare.

În unele exemple de realizare, o memorie RO (ex SROM 102 sau altă memorie RO) poate fi partajată între o multitudine de VM (de exemplu, VM 1 120 și VM 2 122).

În unele exemple de realizare, o memorie RO (ex SROM **102** sau altă memorie RO) poate include un suport care poate fi citit de calculator configurat pentru a permite operații de citire memorie, dar nu și operații de scriere memorie.

5 În unele exemple de realizare, o memorie RW (de exemplu, memoria RW **108** și/sau memoria RW **114**) poate include un suport care poate fi citit de calculator configurat pentru a permite operații de citire memorie și operații de scriere memorie.

În unele exemple de realizare, o memorie RW (de exemplu, memoria RW **108** și/sau memoria RW **114**) sau o memorie RO (ex. SROM **102** sau altă memorie RO) pot include o memorie persistentă, de exemplu, o memorie non-volatilă cu acces aleatoriu (NVRAM).

10 În unele variante, o memorie RW (de exemplu, memoria RW **108** și/sau memoria RW **114**) poate fi un disc virtual de orice dimensiuni creat prin tehnologia 'thin provisioning', astfel încât spațiul de stocare să fie alocat funcție de necesități.

Se va aprecia că metoda **300** este în scopi ilustrativ și că pot fi utilizate acțiuni diferite și/sau suplimentare. Se va aprecia, de asemenea că, diferite acțiuni descrise aici pot apărea
15 într-o ordine sau secvență diferită.

Trebuie remarcat faptul că motorul de configurare **106**, platforma de calcul **100** și/sau funcționalitățile descrise aici, pot constitui un dispozitiv de calcul cu scop special. Mai mult, motorul de configurare **106**, platforma de calcul **100** și/sau funcționalitățile descrise aici, pot
20 îmbunătăți domeniul tehnologiei de testare noduri de rețea, prin oferirea unor mecanisme pentru implementarea unei sau mai multor mașini virtuale care folosesc memoria RO, de exemplu, SROM **102**, și, opțional, memoria RW .

Obiectele prezentei invenții pentru implementarea unei VM îmbunătățesc funcționalitățile platformelor de testare și/sau a instrumentelor de testare, prin oferirea unor
25 mecanisme pentru implementarea unei sau mai multor mașini virtuale care folosesc memoria RO, de exemplu, SROM **102**, și, opțional, memoria RW. Mai mult, prin utilizarea unei memorii RO pentru a stoca VM OS **104** sau alte informații, pot fi implementate rapid una sau mai multe mașini virtuale (de exemplu, șapte până la zece ori mai rapid decât în tehnicile convenționale care nu utilizează memoria RO). De asemenea, trebuie remarcat faptul că o platformă de calcul care pune în aplicare obiectele prezentei invenții poate cuprinde un
30 dispozitiv de calcul cu scop special util pentru a implementa o VM și/sau actualiza o VM.

Se înțelege că diferite detalii ale obiectelor descrise aici pot fi modificate, fără a ne îndepărta de la scopul prezentei invenții. Mai mult decât atât, descrierea de mai sus este în scop numai de ilustrare și nu în scopul limitării, astfel că obiectele invenției sunt definite de revendicări, așa cum sunt enunțate mai jos.

REVENDICĂRI

1. Metodă pentru implementarea unei mașini virtuale (VM) care include:
 - a. recepționarea unei cereri de a crea o mașină virtuală (VM);
 - b. crearea VM folosind un sistem de operare a VM (OS), stocat într-o memorie read-only.
2. Metodă, conform revendicării 1, **caracterizată prin aceea că** VM este configurată să utilizeze o memorie de citire-scriere pentru stocarea informațiilor de utilizator, parolelor, script-urilor de utilizator sau a informațiilor persistente.
3. Metodă, conform revendicării 1, **caracterizată prin aceea că** mai include:
 - i. actualizarea VM prin modificarea memoriei read-only, în care conectivitatea cu VM este opțională, în timpul actualizării.
4. Metodă, conform revendicării 2, **caracterizată prin aceea că** VM este configurată să utilizeze o montare unificată sau alte tehnici de prezentarea memoriei read-only și a memoriei de citire-scriere ca un disc logic sau ca un sistem de fișiere, la un utilizator.
5. Metodă, conform revendicării 2, **caracterizată prin aceea că** operațiile de scriere memorie sunt redirecționate către memoria de citire-scriere.
6. Metodă, conform revendicării 2, **caracterizată prin aceea că** memoria read-only este partajată între o multitudine de mașini virtuale.
7. Metodă, conform revendicării 2, **caracterizată prin aceea că** memoria citire-scriere include un suport care poate fi citit de calculator configurat pentru a permite operații de citire memorie și operații de scriere memorie și **prin aceea că** memoria read-only include un suport care poate fi citit de calculator configurat pentru a permite operații de citire memorie, dar nu operații de scriere memorie.
8. Metodă, conform revendicării 2, **caracterizată prin aceea că** memoria de citire-scriere sau memoria read-only include o memorie persistentă.

9. Metodă, conform revendicării 2, **caracterizată prin aceea că** memoria citire-scriere este un disc virtual de orice dimensiuni creat prin tehnologia ‘thin provisioning’, astfel încât spațiul de stocare este alocat funcție de necesități.

5 10. Sistem pentru implementarea unei mașini virtuale, sistemul cuprinzând:

- i. o memorie read-only; și
- ii. o platformă de calcul configurată pentru a recepționa o cerere pentru a crea oVM și pentru a crea o VM folosind un sistem de operare VM stocat în memoria read-only.

10

11. Sistem, conform revendicării 10, **caracterizat prin aceea că** VM este configurată să utilizeze o memorie de citire-scriere pentru stocarea informațiilor de utilizator, parolelor, script-urilor de utilizator sau a informațiilor persistente.

15

12. Sistem, conform revendicării 10, **caracterizat prin aceea că** platforma de calcul este configurată să actualizeze VM prin modificarea memoriei read-only, în care conectivitatea cu VM este opțională în timpul actualizării.

20

13. Sistem, conform revendicării 11, **caracterizat prin aceea că** VM este configurată să utilizeze o montare unificată sau alte tehnici de prezentarea memoriei read-only și a memoriei de citire-scriere ca un disc logic sau ca un sistem de fișiere, la un utilizator.

25

14. Sistem, conform revendicării 11, **caracterizat prin aceea că** VM este configurată să redirecționeze operațiile de scriere memorie la memoria de citire-scriere.

30

15. Sistem, conform revendicării 11, **caracterizat prin aceea că** memoria read-only este partajată între o multitudine de mașini virtuale.

16. Sistem, conform revendicării 11, **caracterizat prin aceea că** memoria citire-scriere include un suport care poate fi citit de calculator configurat pentru a permite operații de citire memorie și operații de scriere memorie și **prin aceea că** memoria read-only include un suport care poate fi citit de calculator configurat pentru a permite operații de citire memorie, dar nu operații de scriere memorie.

17. Sistem, conform revendicării 11, **caracterizat prin aceea că** memoria de citire-scriere include o memorie persistentă.
- 5 18. Sistem, conform revendicării 11, **caracterizat prin aceea că** memoria citire-scriere este un disc virtual de orice dimensiuni creat prin tehnologia ‘thin provisioning’, astfel încât spațiul de stocare este alocat funcție de necesități.
- 10 19. Suport non-tranzitoriu care poate fi citit de calculator, care are stocate instrucțiuni executabile de calculator implementate în suportul citibil de calculator, care atunci când sunt executate de către procesorul unui calculator, efectuează etapele care includ:
- a. recepționarea unei cereri de a crea o mașină virtuală (VM);
 - b. crearea VM folosind un sistem de operare a VM (OS), stocat într-o memorie read-only.
- 15 20. Suport non-tranzitoriu care poate fi citit de calculator, conform revendicării 19, **caracterizat prin aceea că** VM este configurată să utilizeze o memorie de citire-scriere pentru stocarea informațiilor de utilizator, parolelor, script-urilor de utilizator sau a informațiilor persistente.

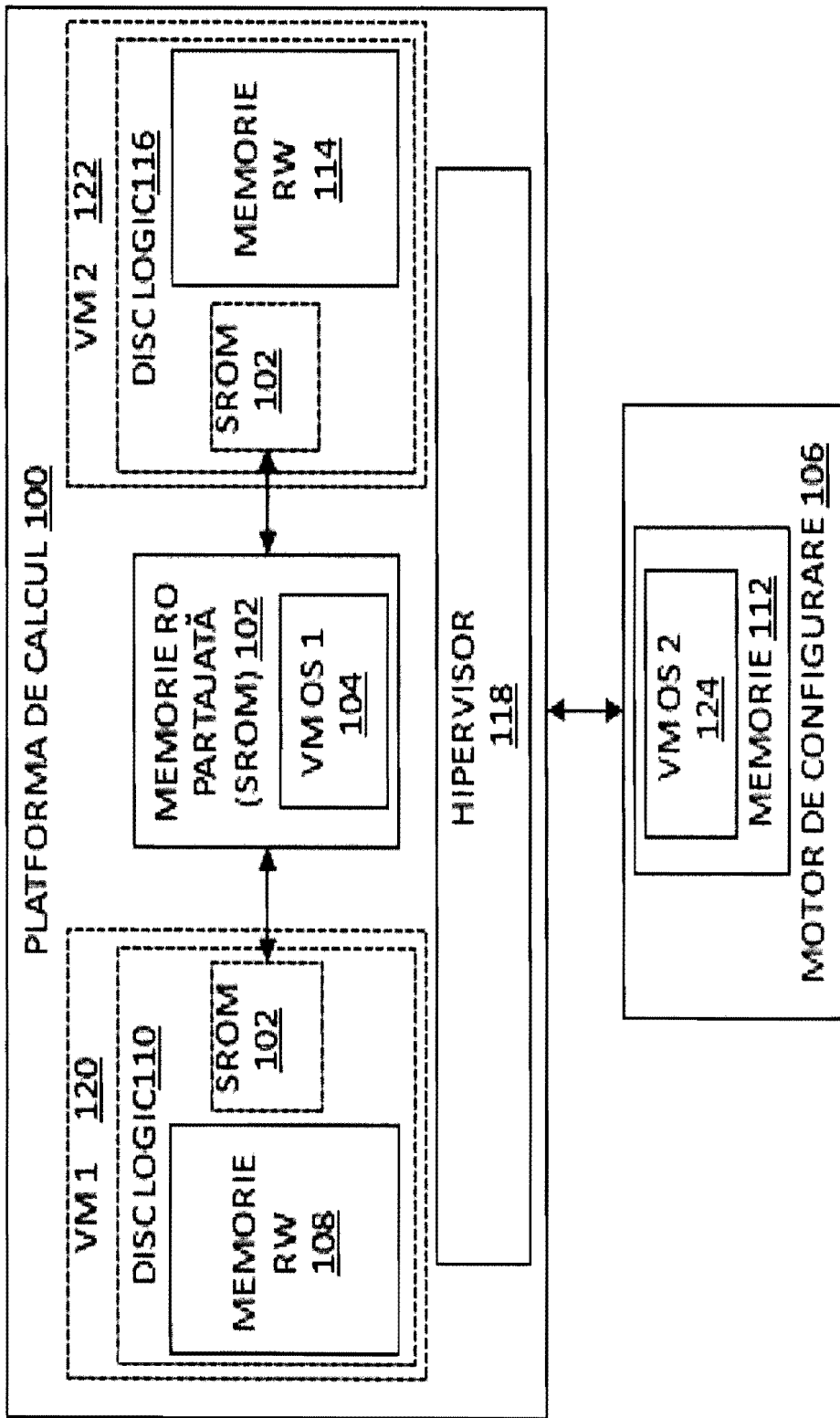


FIG. 1

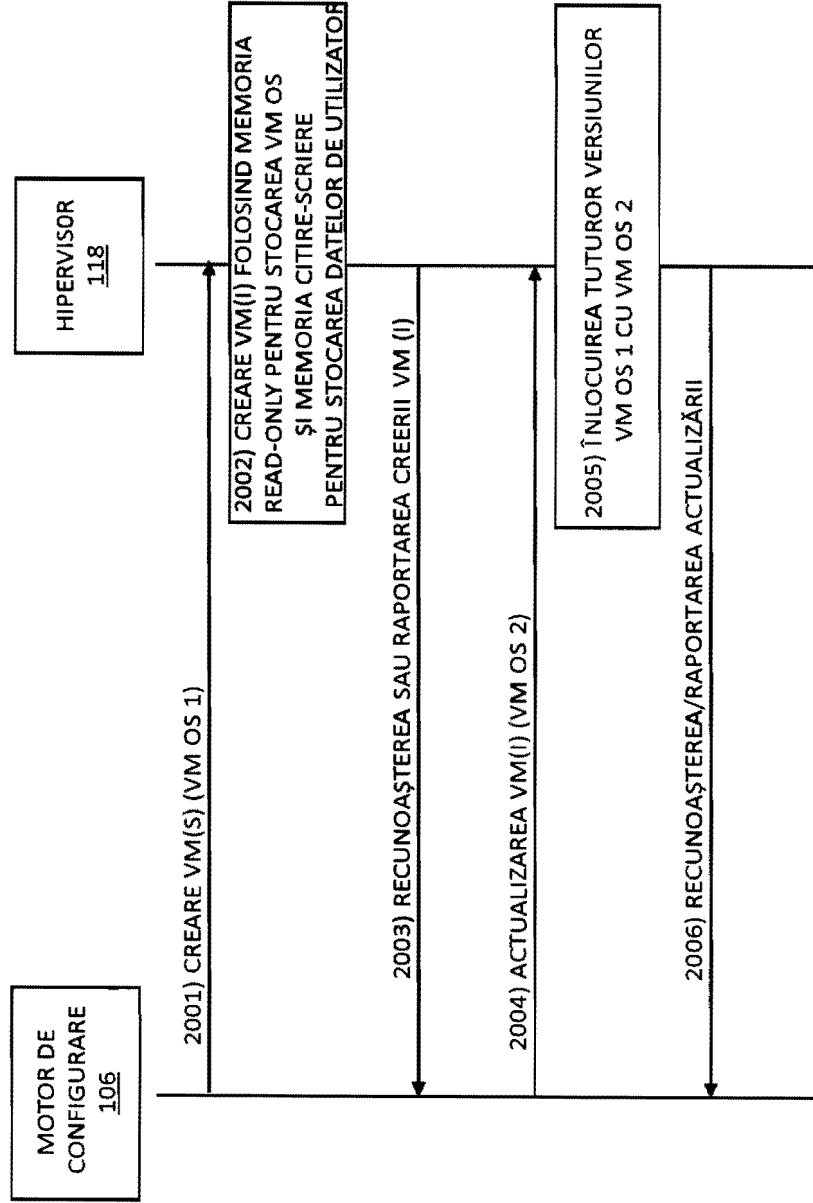


FIG. 2

300

5
10
15
20
25
30
35
40

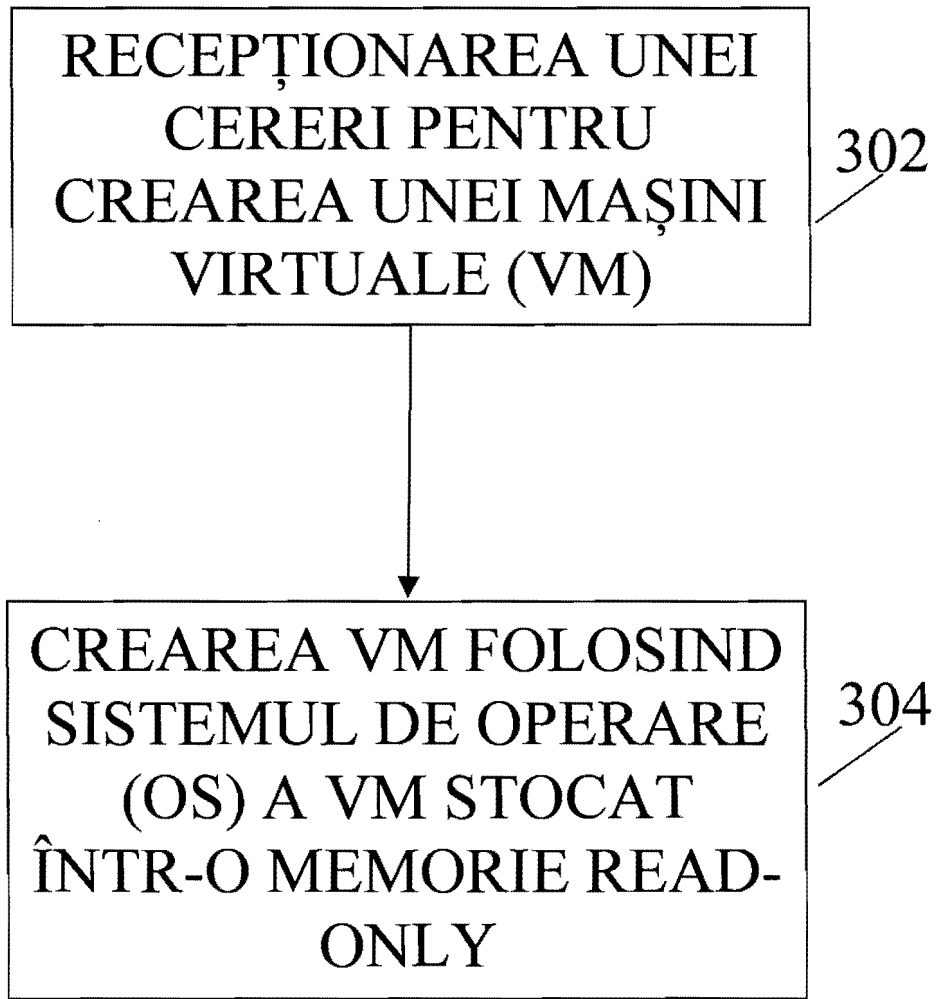


FIG. 3