(19) OFICIUL DE STAT
PENTRU INVENŢII ŞI MĂRCI
Bucureşti

ROMÂNIA

(11) **RO 126508 A2**
(51) **Int.Cl.**
G06F 9/46 (2006.01)

(12) # CERERE DE BREVET DE INVENŢIE

(21) Nr. cerere: **a 2009 00665**

(22) Data de depozit: **28.08.2009**

(41) Data publicării cererii:
29.07.2011 BOPI nr. **7/2011**

(71) Solicitant:
• **DĂSCĂLIŢA DRAGOŞ**, *STR.DACIA NR.39, AP.6, IAŞI, IS, RO;*
• **MĂRGELATU IONUŢ MAXIM**, *STR.DOAMNA GHICA NR.1, CLĂDIREA NR.1, AP.198, BUCUREŞTI, B, RO;*
• **GEORGITA DRAGOŞ**, *STR.BREBU NR.1, BL.T9A, SC.A, AP.36, BUCUREŞTI, B, RO;*
• **ACHIM ALIN**, *STR.STAŢIEI, BL.11, SC.A, ET.1, AP.6, BUMBEŞTI-JIU, GJ, RO;*
• **RAU-NEACŞU LIVIU AURELIAN**, *STR.BABA NOVAC NR.23, BL.G10, SC.1, AP.31, ET.7, BUCUREŞTI, B, RO;*
• **IONESCU COSTIN**, *STR.NICOLAE SEBE NR.13, BUCUREŞTI, B, RO*

(72) Inventatori:
• **DĂSCĂLIŢA DRAGOŞ**, *STR.DACIA NR.39, AP.6, IAŞI, IS, RO;*
• **MĂRGELATU IONUŢ MAXIM**, *STR.DOAMNA GHICA NR.1, CLĂDIREA NR.1, AP.198, BUCUREŞTI, B, RO;*
• **GEORGITA DRAGOŞ**, *STR.BREBU NR.1, BL.T9A, SC.A, AP.36, BUCUREŞTI, B, RO;*
• **ACHIM ALIN**, *STR.STAŢIEI, BL.11, SC.A, ET.1, AP.6, BUMBEŞTI-JIU, GJ, RO;*
• **RAU-NEACŞU LIVIU AURELIAN**, *STR.BABA NOVAC NR.23, BL.G10, SC.1, AP.31, ET.7, BUCUREŞTI, B, RO;*
• **IONESCU COSTIN**, *STR.NICOLAE SEBE NR.13, BUCUREŞTI, B, RO*

(74) Mandatar:
CABINET ENPORA SRL -
STR. GEORGE CĂLINESCU NR. 52A, AP. 1, SECTOR 1, BUCUREŞTI

(54) # METODE ŞI SISTEM PENTRU FURNIZAREA INFORMAŢIILOR PENTRU FOLOSIRE ÎNTR-UN MEDIU CU TIMP DE RULARE OPERAŢIONAL

(57) **Rezumat:**

Invenţia se referă la o metodă şi un sistem pentru furnizarea de informaţii destinate a fi utilizate într-un mediu de rulare care restricţionează accesul la astfel de informaţii. Metoda conform invenţiei este implementată pe calculator şi cuprinde furnizarea unei aplicaţii runtime pe un dispozitiv de calcul, în careaplicaţia runtime nu poate accesa informaţii în cadrul mediului de rulare şi utilizarea aplicaţiei runtime pe dispozitivul de calcul prin lansarea unui executabil care preia informaţiile pe care aplicaţia runtime nu le poate accesa din mediul runtime şi face ca informaţiile să devină disponibile pentru a fi utilizate de aplicaţia runtime, urmată de lansarea aplicaţiei runtime care poate acum accesa informaţiile puse la dispoziţie de către un executabil. Sistemul conform invenţiei conţine un dispozitiv de calcul (**10**) care cuprinde un mediu de rulare (**14**) care rulează pe un sistem de operare (**13**) al dispozitivului de calcul (**10**) şi un pachet (**21**) care cuprinde un executabil (**22**) şi o aplicaţie runtime (**23**), în care lansarea pachetului (**21**) cuprinde lansarea executabilului (**22**) care preia informaţiile pe care aplicaţia runtime (**23**) nu le poate accesa din mediul de rulare (**14**) şi face ca aplicaţiile să devină disponibile pentru a fi utilizate de către aplicaţia runtime (**23**), urmată de lansarea aplicaţiei runtime (**23**).

Revendicări: 20
Figuri: 3

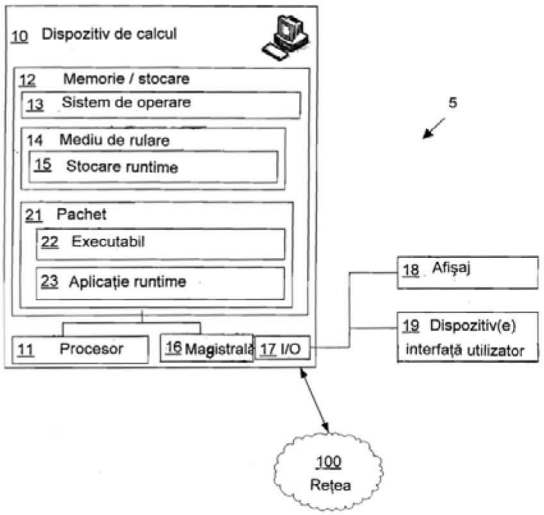

Fig. 1

RO 126508 A2

DOMENIUL

[001] Această descriere, în general, se referă la software pentru calculatoare care execută, afişează, furnizează, sau foloseşte în alt mod conţinutul electronic.

PREZENTARE GENERALĂ

[0002] Sunt disponibile multe medii de rulare (runtime environments) privind aplicaţiile pentru calculatoare. Unele medii de rulare rezidă în partea superioară a sistemului de operare general al unui calculator, pentru a oferi o colecţie de servicii software prin restricţionarea accesului la sistemul de operare şi la alte informaţii externe în timpul de rulare. Un mediu de rulare poate permite, de asemenea, de utilizarea aplicaţiilor în cadrul timpului de rulare fără a permite acestor aplicaţii să acceseze anumite informaţii de aplicaţie şi de sistem. Exemple de informaţii de sistem includ dar nu se limitează la: tipul de procesor, frecvenţa, memoria, spaţiul disponibil pe disc, detalii privind sistemul de operare şi detalii privind placa video. Exemple de informaţii de aplicaţie includ dar nu se limitează la: căile până la aplicaţiile deja instalate, informaţii privind configurarea aplicaţiilor, parametrii de iniţializare, precum şi alte detalii de configurare.

[0003] În timp ce există diverse beneficii în restricţionarea accesului la anumite informaţii de către un mediu de rulare, există anumite circumstanţe în care aceste informaţii sunt necesare în mediul de rulare. De exemplu, aplicaţiile de diagnostic care sunt utilizate pentru a evalua erorile şi alte probleme pot necesita accesul la astfel de informaţii pentru a evalua, printre altele, dacă o eroare este rezultatul resurselor de sistem inadecvate sau al configurării eronate a sistemului. În general, poate fi de dorit ca un mediu de rulare să restricţioneze în general accesul la informaţii în cadrul timpului de rulare, dar să permită sau să faciliteze accesul la astfel de informaţii în anumite alte circumstanţe. Alternativ, un mediu de rulare existent poate restricţiona accesul la anumite informaţii şi ar putea fi de dorit să se adauge capacităţi de acces la informaţii la timpul de rulare fără a schimba mediul de rulare în sine.

Rezumat

[0004] Sunt prezentate sisteme şi metode pentru furnizarea de informaţii pentru a fi utilizate într-un mediu de rulare pentru calculator. Un executabil este lansat înainte de a lansa un mediu de rulare sau o aplicaţie runtime. Executabilul preia informaţiile şi face ca

informaţiile să fie disponibile pentru utilizarea în mediul de rulare, de exemplu, prin stocarea informaţiilor într-o zonă de sistem de fişiere runtime. Mediul de rulare sau aplicaţia runtime poate accesa apoi informaţiile. De exemplu, executabilul poate obţine informaţii de la un sistem de operare gazdă şi apoi să lanseze o aplicaţie runtime care utilizează informaţiile preluate. O aplicaţie care este utilizată în cadrul timpului de rulare şi are nevoie de informaţii runtime externe poate fi furnizată ca un pachet care include un executabil şi aplicaţia însăşi. Când pachetul este lansat, executabilul preia informaţiile şi le pune la dispoziţie pentru runtime. Runtime este apoi lansat şi furnizează aplicaţia runtime care poate folosi aceste informaţii.

[0005] Un exemplu de materializare cuprinde furnizarea unei aplicaţii runtime pe un dispozitiv de calcul, la care aplicaţia runtime este utilizată în într-un mediu de rulare, ce restricţionează accesul la anumite informaţii. Aplicaţia runtime este utilizată sau lansată în alt mod pe dispozitivul de calcul prin lansarea mai întâi a unui executabil care preia informaţiile pe care aplicaţia runtime nu le poate accesa altfel, şi face ca informaţiile să fie disponibile pentru a fi utilizate de către aplicaţia runtime. Aplicaţia runtime este apoi lansată şi poate avea acces la informaţiile puse la dispoziţie de către executabil. Într-un exemplu de materializare, executabilul pune la dispoziţie informaţiile pentru utilizarea de către aplicaţia runtime prin stocarea informaţiilor într-un fişier într-o zonă de fişiere de sistem runtime accesibil de către aplicaţia runtime. Într-un alt exemplu de materializare, executabilul este pornit ca un proces şi pune la dispoziţie informaţiile prin utilizarea unor socket-uri pentru a trece informaţiile la aplicaţia runtime. Într-un exemplu de materializare, executabilul şi aplicaţia runtime sunt stocate ca un singur pachet pe dispozitivul de calcul. Lansarea acestui pachet iniţiază lansarea atât a executabilului cât şi a aplicaţiei runtime.

[0006] Aceste exemple de materializare nu sunt menţionate pentru a limita sau defini descrierea, ci pentru a oferi exemple de materializări pentru a ajuta înţelegerea acestora. Materializările sunt discutate în Descrierea detaliată, şi acolo sunt prevăzute descrieri detaliate. Avantajele oferite de diferitele materializări pot fi înţelese în continuare prin examinarea acestei specificaţii.

Scurtă descriere a figurilor

[0007] Aceste şi alte caracteristici, aspecte, precum şi avantajele prezentei comunicări sunt mai bine înţelese în cazul în care Descrierea detaliată care urmează este citită cu referire la desenele însoţitoare, în care:

Figura 1 este o schemă a sistemului care ilustrează un mediu ilustrativ de calcul;
Figura 2 este o schemă de flux care ilustrează un exemplu de metodă de furnizare a informațiilor pentru a fi utilizate într-un exemplu de mediu de calcul runtime; și
Figura 3 este o schemă care ilustrează un alt exemplu de metodă de furnizare a informațiilor pentru a fi utilizate într-un exemplu de mediu de calcul runtime.

Descrierea detaliată

[0008] Sunt prezentate sisteme și metode pentru furnizarea de informații pentru a fi utilizate într-un mediu de rulare. De exemplu, un mediu de rulare poate furniza o colecție de servicii de software care sunt disponibile în timpul utilizării sale, dar restricționează accesul la sistemul de operare și la alte informații. Un mediu de rulare poate permite, de asemenea, utilizarea aplicațiilor în cadrul runtime, dar poate restricționa accesul la anumite informații. Exemple de informații de sistem includ dar nu se limitează la: viteza și tipul procesorului, frecvența, memoria fizică, memoria RAM, spațiul disponibil pe disc, versiunea și detalii privind sistemul de operare, tipul și detalii privind placa video și/sau volumul sistemului de operare etc. Exemple de informații de aplicație includ dar nu se limitează la: căile până la aplicațiile deja instalate, informații privind configurarea aplicațiilor, parametrii de inițializare, precum și detalii de configurare.

[0009] Una dintre materializări prevede un executabil care este lansat înainte de a lansa un mediu de rulare. Executabilul preia informațiile și face ca informațiile să fie disponibile pentru utilizarea în mediul de rulare, Apoi, mediul de rulare se lansează și accesează informațiile. De exemplu, executabilul poate obține informații de la un sistem de operare gazdă și apoi să lanseze aplicația runtime. În una dintre materializări, aplicație care este utilizată în cadrul timpului de rulare și are nevoie de informații runtime externe este furnizată ca un pachet care include atât un executabil cât și aplicația însăși. Când pachetul este lansat, executabilul preia informațiile și le pune la dispoziție pentru runtime. Runtime este apoi lansat și furnizează aplicația runtime în mediul de rulare.

[0010] În general, informațiile pot fi puse la dispoziția unui runtime sau unei aplicații runtime într-o varietate de moduri. De exemplu, aceste informații pot fi stocate într-un fișier text, o bază de date, sau într-o locație de memorie care este furnizată sau pusă la dispoziția runtime. De exemplu, dacă un runtime are propriul sistem de fișiere sau o altă zonă de stocare, un fișier care conține informațiile preluate ar putea fi stocat în această zonă de stocare și apoi accesat din interiorul runtime. Ca un alt exemplu, executabilul care

preia informațiile ar putea fi a pornit ca un proces şi apoi să utilizeze socket-uri pentru a comunica cu aplicația runtime pentru a transmite informațiile preluate.

[0011] Aceste exemple ilustrative sunt date pentru a-l introduce pe cititor în subiectul general de subiect discutat în acest document şi nu sunt destinate să limiteze domeniul de aplicare a conceptelor descrise. Secțiunile care urmează descriu diferite materializări şi exemple suplimentare.

Mediul de calcul ilustrativ

[0012] Referindu-se acum la desene în care, ca cifre indica ca elemente de pe întreg teritoriul figurile de mai multe, figura 1 este un sistem de diagramă care ilustrează un mediu de calcul ilustrative 5 în funcție de anumite materializări. Materializări Alte pot fi utilizate. Mediu de calcul 5 cuprinde un dispozitiv de calcul 10, că este conectat la o rețea fără fir cu fir sau 100. Aplicații exemplară care executa pe dispozitivul de calcul 10 sunt prezentate ca funcționale sau a componentelor de stocare care au reşedința în memorie 12. De memorie 12 pot fi tranzitorii sau persistente. După cum este cunoscut la una de calificare în arta, astfel de aplicații pot fi rezident în orice computer adecvate de mediu uşor de citit şi executa pe orice procesor adecvat. De exemplu, dispozitivul de calcul 10 mai cuprinde un calculator poate fi citit-mediu, cum ar fi o memorie cu acces aleator (RAM), cuplat la un procesor de 11 de computer care execută instrucțiunile de program executabil şi / sau accesează informațiile stocate în memoria 12. O astfel de procesoare poate cuprinde un microprocesor, un ASIC, o maşină de stat, sau de alte procesor, şi poate fi oricare dintre un număr de procesoare de calculator. Aceste procesoare cuprind, sau pot fi într-un mediu de comunicare cu un mediu citibil de către calculator care stochează instrucțiunile care, atunci când sunt executate de procesor, determină procesorul să efectueze paşii descrişi aici.

[0013] Un mediu citibil de către calculator poate cuprinde, dar nu se limitează la un dispozitiv de stocare electronic, optic, magnetic sau un dispozitiv de stocare de alt tip. capabil de a furniza unui procesor instrucțiuni citibile de către calculator. Alte exemple cuprind, dar nu sunt limitate la dischete, CD-ROM, DVD, discuri magnetice, cipuri de memorie, ROM, RAM, un ASIC, un procesor configurat, benzi magnetice sau alte dispozitive de stocare magnetică sau alte medii de pe care procesorul unui calculator poate citi instrucțiunile. Instrucțiunile pot include instrucțiuni specifice procesorului, generate de un compilator şi / sau de un interpretor de cod, scrise în orice limbă adecvată de

programare pentru calculator, inclusiv, de exemplu, C, C + +, C #, Visual Basic, Java, Python, Perl, JavaScript şi ActionScript.

[0014] Reţeaua 100 prezentată cuprinde Internetul. În alte materializări, pot fi utilizate alte reţele, intraneturi, combinaţii de reţele, sau nici o reţea. Dispozitivul de calcul 10 poate cuprinde, de asemenea, un număr de dispozitive externe sau interne, cum ar fi un mouse, un CD-ROM, DVD, o tastatură, un afişaj, boxe audio, sau alte dispozitive de intrare sau ieşire. De exemplu, dispozitivul de calcul include 10 conexiunile 17de intrare / ieşire, care leagă un afişaj 18 şi diverse dispozitive de interfaţă cu utilizatorul 19. Dispozitivul de calcul 10, descris ca un sistem informatic unic, poate fi implementat ca o reţea de calculatoare, servere, sau procesoare. Exemple de dispozitive server sunt serverele, calculatoarele mainframe, calculatoarele legate în reţea, un dispozitiv bazat pe procesor, precum şi tipuri similare de sisteme si dispozitive.

[0015] Un dispozitiv de calcul, cum ar fi dispozitivul exemplificativ de calcul 10, poate utiliza componente funcţionale variate pentru a implementa una sau mai multe dintre caracteristicile descrise aici. Dispozitivul de calcul 10 are un sistem de operare 13 pentru operarea dispozitivului de calcul 10 şi un mediu de rulare 14 pentru furnizarea de funcţionalitate şi aplicaţii runtime specifice. Aparatul exemplificativ de calcul 10 include un pachet 21, care cuprinde un executabil 22 şi o aplicaţie runtime 23. Executabilul 22 poate fi utilizat direct din sistemul de operare 13, în timp ce aplicaţia runtime 23 este utilizată în runtime. Când pachetul 21 este pornit, executabilul preia orice informaţii necesare, cum ar fi informaţiile despre dispozitivul de calcul 10, sistemul de operare 13 al acestuia, sau alte componente, sau despre orice aplicaţii instalate sau utilizate anterior pe dispozitivul de calcul 10. Executabilul 22 poate furniza astfel de informaţii pentru a fi utilizate de către aplicaţia runtime 23, care poate fi altfel în imposibilitatea de a accesa aceleaşi informaţii, de exemplu, din cauza limitărilor sau restricţiilor impuse de mediul de rulare 14. Într-unul dintre exemple, executabilul 22 preia sistemul de operare şi alte informaţii şi furnizează aceste informaţii pentru stocarea într-o stocare runtime 15 asociată cu şi accesibilă din mediul de rulare 14.

[0016] Acest mediu de calcul ilustrativ 5 este oferit doar pentru a ilustra un mediu potenţial care poate fi folosit pentru a implementa anumite materializări. Alte dispozitive de calcul şi de configuraţii pot fi utilizate în mod alternativ sau suplimentar

Metode exemplificative pentru furnizarea de informaţii destinate utilizării într-un mediu de rulare pentru calcul

[0017] Anumite materializări furnizează informaţii pentru a fi utilizate într-un mediu de rulare, cum ar fi Adobe® AIR™. Adobe® AIR™ este un mediu de rulare care se află în partea superioară a unui sistem de operare şi este folosit pentru a rula o serie de aplicaţii diverse, inclusiv aplicaţiile bogate de Internet (RIA), care, de obicei, oferă experienţe mai bune din punct de vedere grafic şi interactiv, decât conţinutul browserelor web tradiţionale. Aplicaţiile Adobe® AIR™ au acces limitat la informaţiile privind funcţionarea sistemului şi la informaţiile privind setările locale. Una dintre materializările exemplificative furnizează o aplicaţie Adobe® AIR™ cu informaţii despre sistemul de operare şi / sau informaţii cu privire la una sau mai multe aplicaţii Adobe® AIR™.

[0018] O varietate de tehnici pot fi folosite pentru a transmite informaţii către o aplicaţie care funcţionează într-un runtime în care există un acces limitat la informaţiile de sistem. Astfel de informaţii pot fi furnizate înainte de lansarea runtime. De exemplu, o aplicaţie C++, o aplicaţie Java, o aplicaţie Visual Basic, sau un alt program executabil separat poate fi utilizat pentru a prelua informaţii de la sistemul gazdă, înainte de lansarea unei aplicaţii de rulare, care va utiliza apoi informaţiile. Programul executabil separat poate furniza informaţiile colectate runtime-ului sau unei aplicaţii runtime în diferite moduri. Informaţiile pot fi stocate într-o bază de date sau într-o altă zonă de memorie şi / sau transmise direct aplicaţiei. Executabilul separat ar putea transmite informaţiile într-un fişier text, într-o altă zonă de memorie de stocare, sau să plaseze informaţiile într-o zonă de fişiere de sistem specifice runtime care poate fi accesată de o aplicaţie runtime , ca exemple. Într-o materializare alternativă, executabilul separat poate fi pornit ca un proces şi pot fi folosite socket-uri pentru a comunica cu runtime-ul sau cu aplicaţia runtime pentru a transmite informaţiile.

[0019] În contextul unei aplicaţii de diagnosticare care rulează într-un mediu de rulare, poate fi necesar ca aplicaţia de diagnosticare să determine dacă dispozitivul îndeplineşte cerinţele minime de sistem. De exemplu, o astfel de informaţii poate ajuta o astfel de aplicaţie de diagnostic să stabilească dacă unele schimbări recente într-un sistem, ca de exemplu eliminarea unei părţi din RAM, fac ca sistemul să nu mai îndeplinească anumite cerinţe minime de sistem. Dacă utilizatorul nu are permisiunea de a accesa informaţiile necesare, aplicaţia va da un mesaj.

[0020] Una dintre materializări prevede utilizarea unui executabil separat cu o aplicație de diagnostic de runtime, cum ar fi unul care rulează pe Adobe® AIR™. O astfel de aplicație de diagnostic exemplificativă poate fi o aplicație desktop stocată ca un pachet care cuprinde atât aplicația de diagnostic cât şi un executabil separat. Pachetul poate fi configurat astfel încât, atunci când este lansat, lansează separat executabil înainte de lansarea runtime care rulează aplicația de diagnosticare. Executabilul separat poate obține informații pentru utilizarea ulterioară de către aplicația de diagnosticare în runtime prin furnizarea de aplicației la aplicația de diagnostic sau la o locație accesibilă prin aplicația de diagnosticare.

[0021] O aplicație exemplificativă de diagnostic utilizată într-un mediu de rulare poate fi folosită pentru a diagnostica erorile care apar în una sau mai multe alte aplicații care rulează pe calculator atât în interiorul cât şi în exteriorul mediului de rulare. În consecință, informațiile furnizate unei aplicații de diagnostic se pot referi la informația la nivel general şi la nivel de sistem, de exemplu, privind sistemul de operare şi / sau se poate referi la informații despre o aplicație care este în curs de examinare prin aplicația de diagnosticare. Informațiile cu privire la aplicația care trebuie să fie examinată, de exemplu, poate să fi fost stocate ca un fişier jurnal pe calculator. La lansare, aplicația de diagnosticare pot fi furnizată cu sau să ofere acces la astfel de informații din fişierul jurnal de către executabilul separat. Aplicația exemplificativă de diagnostic poate utiliza informațiile din fişierul jurnal, informațiile generale sau de sistem şi / sau informațiile preluate din alte surse, inclusiv, dar nu limitat la serverele externe, pentru a diagnostica şi a corecta erorile.

[0022] În cazul în care un utilizator nu este în măsură de a diagnostica sau a corecta o eroare, o aplicație de diagnostic poate fi în măsură să furnizeze informațiile jurnal şi / sau alte informații relevante pentru utilizator pentru a contacta o sursă externă de diagnostic, cum ar fi o linie de ajutor. De exemplu, fişierul jurnal şi / sau alte informații poate fi încărcat la un server aflat la distanță, iar utilizatorul poate fi în măsură să apeleze o linie de ajutor şi să discute eroarea cu un specialist care va avea acces la fişierul jurnal şi la alte informații. O aplicație de diagnosticare poate transmite astfel de informații la un punct de colectare care stochează jurnalele şi alte informații, inclusiv rezultatele diagnosticelor de la mai multe sisteme şi utilizatori. Un punct de colectare poate permite forme statistice şi alte forme de analiză care urmează să fie folosite pentru a permite unui furnizor de aplicații să înțeleagă mai bine natura şi amploarea erorilor care apar în timpul utilizării aplicațiilor lor.

[0023] Figura 2 este o schemă de flux care ilustrează o metodă exemplificativă 200 de furnizare a informațiilor destinate pentru a fi utilizate într-un mediu de calcul runtime exemplificativ. Metoda exemplificativă 200 implică utilizarea unui executabil separat 210 şi a unei aplicații runtime 220. De exemplu, un executabil separat şi o aplicație de runtime pot fi furnizate şi / sau stocate ca un pachet pe un dispozitiv de calcul, cum ar fi în pachetul exemplificativ 21 descris în figura 1.

[0024] Metoda 200 implică utilizarea aplicației runtime pornind mai întâi executabilul separat. Metoda 200 implică regăsirea informațiilor de către executabil, aşa cum se arată în blocul 230. Aceste informații pot fi o parte sau toate informațiile care nu sunt disponibile aplicației runtime. Aceste informații pot include informații generale sau alte informații de sistem, cum ar fi viteza şi tipul procesorului, frecvența, memoria fizică, memoria RAM, spațiul disponibil pe disc, versiunea şi detaliile sistemului de operare, tipul şi detaliile plăcii video şi / sau volumul sistemului de operare etc. Aceste informații pot include, alternativ sau suplimentar, informații privind căile până la aplicațiile deja instalate utilizate pe dispozitivul de calcul în interiorul sau în afara timpului de rulare, parametrii de inițializare a aplicației, precum şi alte detalii de configurare a aplicațiilor.

[0025] După preluarea informației folosind executabilul separat 210, metoda 200 implică lansarea aplicației runtime 220, aşa cum se arată în blocul 240. Acest lucru poate implica lansarea mai întâi a mediului de rulare, dacă un astfel de mediu nu este deja lansat, şi apoi lansarea aplicației runtime 220 în acel mediu. În unele materializări alternative, executabilul separat 210 şi aplicația runtime 220 sunt lansate concomitent. În alte materializări alternative, este lansată mai întâi aplicația runtime 220, iar executabilul separat este lansat dacă sunt necesare informații care nu sunt accesibile din interiorul runtime-ului.

[0026] Revenind la metoda exemplificativă 200 din figura 2, după lansarea aplicației runtime 220, aplicația runtime 220 regăseşte informațiile, după cum se arată în blocul 250. În acest exemplu, aplicația runtime 220 solicită informațiile de la executabilul separat 210 sau dintr-o locație de memorie asociată cu executabilul separat 210. Într-o materializare exemplificativă, executabilul separat continuă executarea până când informațiile sunt furnizate aplicației runtime 220. Într-o altă materializare exemplificativă, executabilul separat îşi încheie funcționarea după regăsirea informațiilor şi stocarea informațiilor într-o locație accesibilă de către aplicația runtime 220.

[0027] Informaţiile sunt furnizate aplicaţiei runtime 220, aşa cum se arată în blocul 260. Aceasta se poate întâmpla într-o varietate de moduri în funcţie de circumstanţele speciale şi de caracteristicile puse în aplicare de runtime şi / sau de aplicaţie. Executabilul separat 210 poate fi configurat pentru a furniza în mod direct informaţii la cererea aplicaţiei runtime 220. De exemplu, executabilul separat 210 ar putea fi pornit ca un proces şi ar putea fi utilizate socket-uri pentru a comunica cu aplicaţia runtime 220 pentru a transmite informaţiile.

[0028] Figura 3 este o schemă de flux care ilustrează o altă metodă exemplificativă 300 de furnizare a informaţiilor pentru a fi utilizate într-un mediu de calcul runtime exemplificativ. Metoda exemplificativă 300 implică utilizarea unui executabil separat 310, a unei zone de sistem pentru fişiere runtime 320, şi a unei aplicaţii runtime 330. Ca şi în exemplul anterior din figura 2, executabilul separat 310 şi o aplicaţie runtime 320 pot sau nu pot fi furnizate şi / sau stocate ca un pachet pe un dispozitiv de calcul, cum ar fi în pachetul exemplificativ 21 descris în figura 1. Zona de sistem de fişiere runtime 320 din acest exemplu este accesibilă din interiorul mediului de rulare care rulează aplicaţia runtime 220. De exemplu, mediul de rulare 14 din Figura 1 ar putea fi folosit pentru a rula aplicaţia runtime 23, iar stocarea runtime 15 ar putea fi folosită pentru a stoca informaţiile furnizate de către executabilul 22 pentru accesul de către aplicaţia runtime 23 şi de către alte aplicaţii din mediul de rulare14. Zona sistemului de fişiere runtime 320 descrisă în figura 3 poate fi localizată în dispozitivul de calcul pe care este furnizat runtime-ul sau poate fi situată în altă parte, de exemplu, într-o reţea cu locaţie pe Internet accesibilă din dispozitivul de calcul.

[0029] Metoda 300 poate începe într-o varietate de stadii. De exemplu, mediul de rulare folosit pentru a rula aplicaţia runtime 330 poate sau nu poate fi deja în curs de rulare. În general, separat faţă de mediul de rulare, metoda 300 începe atunci când executabilul separat 310 preia informaţiile 340 pentru a fi utilizate de aplicaţia runtime, după cum se arată în blocul 340. Executabilul separat poate accesa informaţiile privind sistemul, sistemul de operare, mediul de calcul, reţelele disponibile, dispozitivele periferice şi / sau o varietate de alte tipuri de informaţii, în orice mijloace adecvate, cunoscute în general şi utilizate de către aplicaţiile care se execută printr-un sistem de operare pe o reţea sau în alt mod. Executabilul separat poate, ca un exemplu, să acceseze un registru utilizat şi / sau menţinut de un sistem de operare de pe dispozitivul de calcul.

[0030] După ce executabilul preia informaţiile, metoda 300 presupune în continuare trimiterea de informaţii preluate de către executabilul separat 310 la zona sistemului de fişiere runtime 320, după cum arată blocul 350. Stocarea sau furnizarea de informaţii în zona sistemului de fişiere runtime 320 poate fi realizată într-o varietate de tehnici. De exemplu, informaţiile pot fi compilate în unul sau mai multe fişiere noi care sunt stocate în zona sistemului de fişiere runtime 320. Alternativ, informaţiile pot fi adăugate la fişierele sau jurnalele existente menţinute în zona sistemului de fişiere runtime.

[0031] Metoda 200 presupune lansarea aplicaţiei runtime 220, aşa cum se arată în blocul 360. Acest lucru poate avea loc înaintea sau după blocul 350, în funcţie de circumstanţe. În anumite circumstanţe, poate fi necesar ca informaţiile să fie trimise în zona sistemului de fişiere runtime 320 înainte de lansarea mediului de rulare şi / sau a aplicaţiei runtime 320. În general, lansarea aplicaţiei runtime 330 poate implica mai întâi lansarea mediului de rulare, în cazul în care un astfel de mediu nu este deja lansat, şi apoi lansarea aplicaţiei runtime 330 în acel mediu.

[0032] Odată lansată, aplicaţia runtime 330 poate prelua informaţiile, aşa cum se arată în blocul 370, iar informaţiile sunt furnizate aplicaţiei runtime, aşa cum se arată în blocul 380. Informaţiile pot fi preluate imediat, la lansarea aplicaţiei, sau regăsite pe bază de necesitate. Aplicaţia runtime poate interacţiona, de asemenea, în continuare cu zona sistemului de fişiere runtime 320 pentru a actualiza informaţiile, de exemplu, prin adăugarea de informaţii suplimentare din cadrul mediului de rulare sau din locaţia accesibilă din mediul de rulare.

[0033] În general, în plus faţă de exemplele discutate anterior, informaţiile pot fi regăsite şi furnizate unui mediu de rulare sau unei aplicaţii runtime într-o varietate de moduri, ale căror detalii pot depinde de detaliile specifice ale mediului de rulare sau ale aplicaţiei runtime.

Generalităţi

[0034] Numeroase detalii specifice sunt stabilite aici pentru a oferi o înţelegere aprofundată a subiectului revendicat. Cu toate acestea, cei calificaţi în domeniu vor înţelege că subiectul poate fi practicat fără aceste detalii specifice. În alte cazuri, metodele, aparatele sau sistemele care ar fi cunoscute de către cei cu cunoştinţe obişnuite nu au fost descrise în detaliu, astfel încât să nu se ascundă subiectul revendicat.

[0035] Unele porţiuni sunt prezentate în termeni de algoritmi sau de reprezentări simbolice de operaţiuni pe biţi de date sau semnale digitale binare stocate într-o memorie a sistemului de calcul, cum ar fi memoria unui calculator. Aceste descrieri sau reprezentări algoritmice sunt exemple de tehnici utilizate de către cei cu cunoştinţe obişnuite în domeniul prelucrării datelor pentru a transmite substanţa muncii lor altora calificaţi in domeniu. Un algoritm este o secvenţă auto-coerentă de operaţiuni de prelucrare sau similare, care să ducă la rezultatul dorit. În acest context, operaţiunile sau prelucrarea implică manipularea fizică a cantităţilor fizice. În mod tipic, dar nu necesar, astfel de cantităţi pot lua forma unor semnale electrice sau magnetice care pot fi stocate, transferate, combinate, comparate sau manipulate în alt mod. S-a dovedit convenabil în timp, în principal pentru motive de utilizare comună, să se facă referinţă la asemenea semnale ca la biţi, date, valori, elemente, simboluri, caractere, termene, numere, cifre şi altele asemenea. Trebuie să se înţeleagă însă că toţi aceşti termeni şi alţii asemenea trebuie să fie asociaţi cu cantităţi fizice corespunzătoare şi sunt doar nişte etichete convenabile. Dacă nu se afirmă în mod specific altfel, se apreciază că, în toate aceste discuţii privind specificaţia, utilizarea unor termeni ca "procesare", "calculare", "determinare" şi „identificare" sau altele asemenea se referă la acţiuni sau procese ale unei platforme de calcul, cum ar fi unul sau mai multe calculatoare şi/sau un dispozitiv sau dispozitive electronice similare de calcul care manipulează sau transformă date reprezentate ca nişte cantităţi fizice electronice sau magnetice prin memorii, registre sau alte dispozitive de stocare a informaţiilor, dispozitive de transmitere sau dispozitive de afişare ale platformei de calcul.

[0036] Diferitele sisteme discutate aici nu sunt limitate la nici o arhitectură sau configuraţie particulară de hardware. Un dispozitiv de calcul poate include orice aranjament corespunzător de componente care furnizează un rezultat condiţionat de una sau mai multe intrări. Dispozitivele adecvate de calcul includ sisteme de calculatoare multiscop bazate pe microprocesoare care accesează soft stocat care programează sau configurează sistemul de calcul de la un aparat de calcul de utilizare generală până la un aparat specializat de calcul care implementează una sau mai multe materializări ale prezentului obiect. Se poate utiliza orice limbaj sau combinaţie de limbaje de programare, scriptare sau de alte tipuri, pentru a implementa cunoştinţele cuprinse aici într-un soft destinat pentru utilizarea în programarea sau configurarea unui dispozitiv de calcul.

[0037] Materializări ale metodelor comunicate aici pot fi realizate în exploatarea unor asemenea dispozitive de calcul. Ordinea blocurilor prezentate în exemplele de mai sus

poate varia - de exemplu, blocurile pot fi reordonate, combinate şi/sau divizate în sub-blocuri. Unele blocuri sau procese pot fi realizate în paralel.

[0038] Aşa cum s-a remarcat mai sus, un dispozitiv de calcul poate accesa unul sau mai multe medii citibile de calculator care încorporează în mod sensibil instrucţiuni citibile de calculator care, atunci când sunt executate de cel puţin un calculator, determină cel puţin un calculator să implementeze una sau mai multe materializări ale prezentului obiect. Când se utilizează softul, acesta poate cuprinde una sau mai multe componente, procese şi/sau aplicaţii. Suplimentar sau alternativ la software, dispozitivul (ele) pot cuprinde circuite care fac ca dispozitivul(ele) să devină operative pentru a implementa una sau mai multe metode ale prezentului obiect.

[0039] Exemplele de dispozitive de calcul includ, dar nu sunt limitate la servere, calculatoare personale, asistenţi digitali personali (PDA), telefoane celulare, televizoare, boxe set-top de televiziune şi playere portabile de muzică Dispozitivele de calcul pot fi integrate în alte dispozitive, de ex. aparate "smart", automobile, chioşcuri şi altele.

[0040] Flexibilitatea inerentă a sistemelor bazate pe calculatoare permite o mare varietate posibilă de configuraţii, combinaţii şi diviziuni de sarcini şi funcţionalităţi între componente. De exemplu, procesele discutate aici pot fi implementate utilizând un singur dispozitiv de calcul sau mai multe dispozitive de calcul care să lucreze în combinaţie. Bazele de date şi aplicaţiile pot fi implementate pe un singur sistem sau distribuite pe mai multe sisteme. Componentele distribuite pot funcţiona secvenţial sau în paralel.

[0041] Când datele sunt obţinute sau accesate între un prim sistem şi un al doilea sistem de calculatoare sau de componente ale acestora, datele actuale pot circula direct sau indirect între sisteme. De exemplu, dacă un prim calculator accesează date dintr-un al doilea calculator, accesul poate implica unul sau mai multe calculatoare, proxi-uri etc. intermediare. Datele efective pot circula între primul şi al doilea calculator, sau primul calculator poate furniza un cursor sau un metafişier pe care cel de-al doilea calculator îl utilizează pentru a accesa datele efective dintr-un alt calculator decât primul calculator, de exemplu. Datele pot fi "trase" printr-o cerere sau "împinse" fără o cerere, în diferite materializări..

[0042] Tehnologia descrisă aici face, de asemenea, referinţă la comunicarea datelor între componente sau sisteme. Se va aprecia că o asemenea comunicare poate avea loc prin orice număr sau tip adecvat de reţele sau linkuri, incluzând dar nefiind limitată la reţele comutate, o reţea locală (LAN), o reţea lărgită (WAN), reţeaua publică de telefonie

comutată (PSTN), Internetul, un intranet sau orice combinație de linkuri de comunicare cu sau fără fir.

[0043] Orice mediu tangibil adecvat citibil de către calculator poate fi utilizat pentru implementarea sau practicarea obiectului comunicat prin prezenta, inclusiv dar nefiind limitat la dischete, unități de disc, medii de stocare magnetică, medii de stocare optică, inclusiv discurile (incluzând CD-ROM, DVD-ROM şi variante ale acestora), memorii flash, RAM, ROM, şi alte dispozitive de memorie.

[0044] Utilizarea aici a expresiilor "adaptat pentru" sau "configurat pentru" este înțeleasă ca un limbaj deschis şi inclusiv care nu rezolvă dispozitive adaptate sau configurate pentru a efectua sarcini sau paşi suplimentari. În plus, utilizarea expresiei "bazat pe" este înțeleasă a fi deschisă şi inclusivă, în sensul că, în practică, un proces, un pas, un calcul sau o altă acțiune "bazată pe" una sau mai multe condiții sau valori enumerate poate să fie bazată, în practică, pe condiții sau valori suplimentare peste cele enumerate. Anteturile, listele şi numerotările incluse aici servesc numai pentru înlesnirea explicării şi nu sunt înțelese a fi limitative.

[0045] Deşi prezentul obiect a fost descris în detaliu cu referire la materializări specifice ale acestuia, se va aprecia că cei cu cunoştințe în domeniu, după înțelegerea celor de mai sus, pot realiza cu uşurință modificări, variante şi echivalente ale unor asemenea materializări. În mod corespunzător, se va înțelege că prezenta comunicare a fost prezentată mai mult pentru exemplificare decât ca limitare şi nu împiedică includerea unor modificări, variații şi/sau adăugări la prezentul obiect, aşa cum va fi evident, cu uşurință, pentru o persoană cu cunoştințe obişnuite în acest domeniu.

a-2009-00665--
28 -08- 2009

61

REVENDICĂRI

1. O metodă implementată pe calculator, cuprinzând:
furnizarea unei aplicații runtime pe un dispozitiv de calcul, aplicația runtime utilizată în cadrul unui mediu de rulare pe dispozitivul de calcul, în care aplicația runtime nu poate accesa informațiile în cadrul mediului de rulare;
utilizarea aplicației runtime pe dispozitivul de calcul prin:
lansarea unui executabil pe un dispozitiv de calcul, în care executabilul preia informațiile pe care aplicația runtime nu le poate accesa din mediul runtime şi face ca informațiile să fie disponibile pentru a fi utilizate de aplicația runtime; precum şi lansarea aplicației runtime, în care aplicația runtime poate accesa informațiile puse la dispoziție de către executabil.

2. Metoda conform revendicării 1, în care executabilul face ca informațiile să fie disponibile pentru utilizarea de către aplicația runtime prin stocarea informațiilor într-o zonă a sistemului de fişiere runtime accesibilă de către aplicația runtime.

3. Metoda conform revendicării 1, în care executabilul este pornit ca un proces, în care executabilul face ca informațiile să fie disponibile pentru utilizarea de către aplicația runtime folosind socket-uri pentru a transmite informațiile către aplicație runtime.

4. Metoda conform revendicării 1, în care executabilul şi aplicația runtime sunt stocate ca un singur obiect pe dispozitivul de calcul şi în care lansarea obiectului unic inițiază lansarea atât a executabilului cât şi a aplicației runtime.

5. Metoda conform revendicării 1, în care informațiile specifică: viteza procesorului, tipul procesorului, mărimea memoriei fizice, mărimea memoriei RAM, spațiul disponibil pe disc, versiunea sistemului de operare sau tipul plăcii video.

6. Metoda conform revendicării 1, în care informațiile specifică o cale până la o altă aplicație pe dispozitivul de calcul.

7. Metoda conform revendicării 1, în care lansarea executabilului are loc înainte de lansarea aplicației runtime.

8. Metoda conform revendicării 1, în care lansarea executabilului are loc în timp ce mediul de rulare nu este în execuție.

9. Metoda conform revendicării 1, în care mediul de rulare este Adobe AIR care rulează în partea superioară a unui sistem de operare de pe dispozitivul de calcul şi în care informațiile reprezintă informații privind sistemul de operare.

10. Metoda conform revendicării 1, în care aplicația runtime este o aplicație de diagnostic care necesită informațiile pentru a diagnostica erori.

11. Un aparat de calcul care cuprinde:
un mediu de rulare specificat de instrucțiunile din aparatul de calcul, în care mediul de rulare care rulează pe un sistem de operare al aparatului de calcul restricționează accesul la informații şi;
un pachet specificat de instrucțiunile din aparatul de calcul, în care lansarea pachetului cuprinde:
lansarea unui executabil pe un aparat de calcul, după care executabilul preia informațiile pe care o aplicație runtime nu le poate accesa din mediul de rulare şi face ca informațiile să fie disponibile pentru utilizarea de către aplicația runtime, şi
lansarea aplicației runtime, după care aplicația runtime poate accesa informațiile puse la dispoziție de executabil.

12. Aparat de calcul conform revendicării 11, în care lansarea pachetului cuprinde mai departe rularea mediului de rulare după lansarea executabilului.

13. Aparat de calcul conform revendicării 11, în care executabilul face ca informațiile să fie disponibile pentru utilizarea de către aplicația runtime prin stocarea informațiilor într-o zonă a sistemului de fişiere runtime accesibilă de către aplicația runtime.

14. Aparat de calcul conform revendicării 11, în care executabilul este pornit ca un proces, la care executabilul face ca informațiile să fie disponibile pentru utilizarea de către aplicația de runtime folosind socket-uri pentru a transmite informațiile la aplicația runtime.

15. Aparat de calcul conform revendicării 11, în care informațiile specifică: viteza procesorului, tipul procesorului, mărimea memoriei fizice, mărimea memoriei RAM, spațiul disponibil pe disc, versiunea sistemului de operare sau tipul plăcii video.

16. Aparat de calcul conform revendicării 11, în care mediul de rulare este Adobe AIR şi în care informațiile reprezintă informații privind sistemul de operare.

17. Aparat de calcul conform revendicării 11, în care aplicația runtime este o aplicație de diagnostic care necesită informațiile pentru a diagnostica erori.

18. Un mediu citibil de către calculator pe care este codat un cod program, codul program cuprinzând:
cod program pentru furnizarea unei aplicații runtime pe un dispozitiv de calcul, aplicația runtime fiind utilizată în cadrul unui mediu de rulare pe dispozitivul de calcul, în care aplicația runtime nu poate accesa informațiile în mediul de rulare;
cod program pentru lansarea aplicației runtime pe dispozitivul de calcul prin:
lansarea unui executabil pe un dispozitiv de calcul, în care executabilul preia informațiile pe care aplicația runtime nu le poate accesa din mediul runtime şi face ca informațiile să fie disponibile pentru utilizarea de către aplicația runtime; precum şi
lansarea aplicației runtime, după care aplicația runtime poate accesa informațiile puse la dispoziție de către executabil.

19. Mediu citibil de către calculator conform revendicării 18, în care executabilul face ca informațiile să fie disponibile pentru utilizarea de către aplicația runtime prin stocarea informațiilor într-o zonă a sistemului de fişiere runtime accesibilă de către aplicația runtime.

20. Mediu citibil de către calculator conform revendicării 18, în care executabilul este pornit ca un proces, în care executabilul face ca informațiile să fie disponibile pentru

utilizarea de către aplicația de runtime folosind socket-uri pentru a transmite informațiile la aplicația runtime.

========

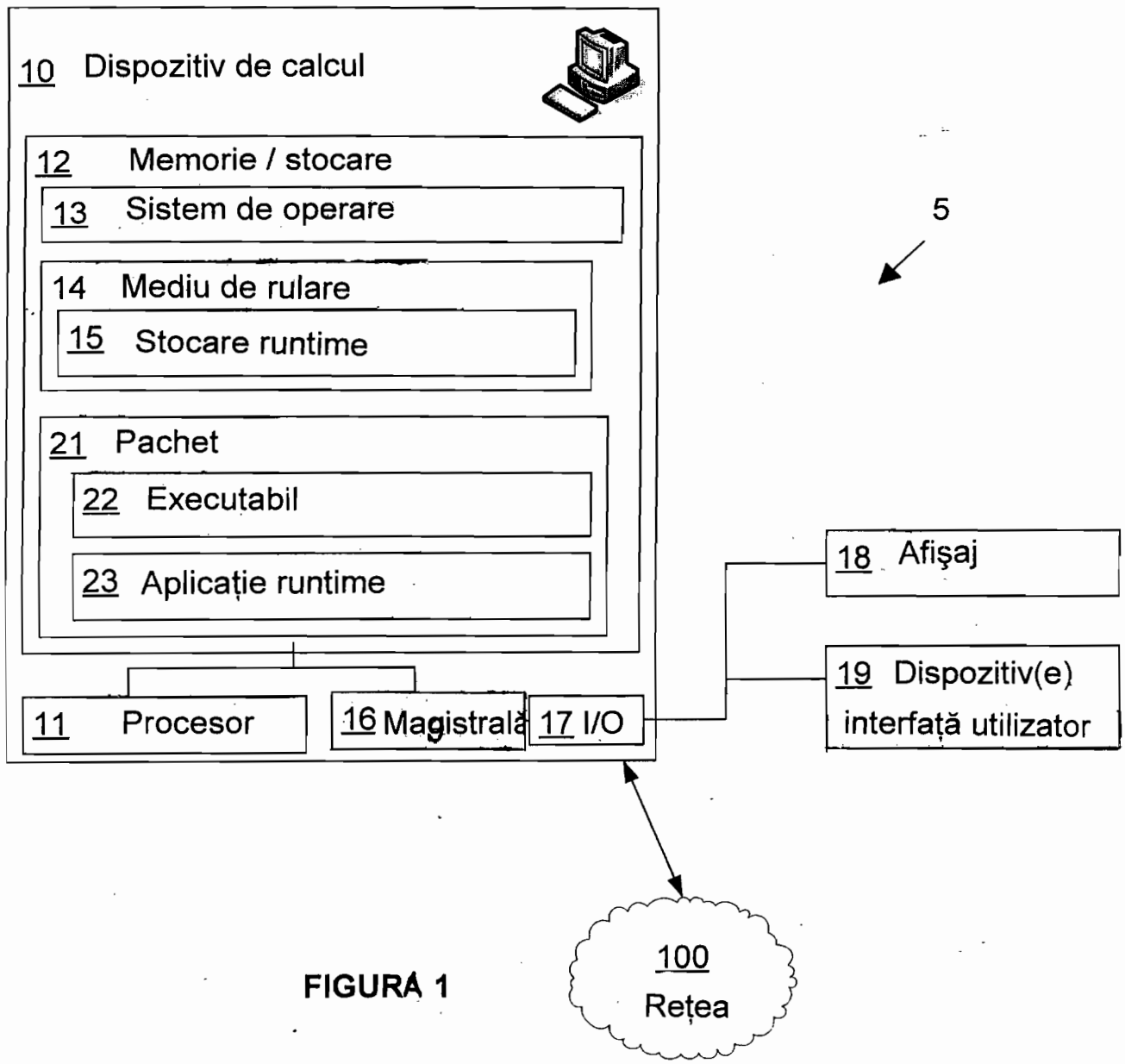
10 Dispozitiv de calcul
12 Memorie / stocare
13 Sistem de operare
14 Mediu de rulare
15 Stocare runtime
21 Pachet
22 Executabil
23 Aplicație runtime
11 Procesor
16 Magistrală
17 I/O
18 Afișaj
19 Dispozitiv(e) interfață utilizator
5
100
Rețea

FIGURA 1

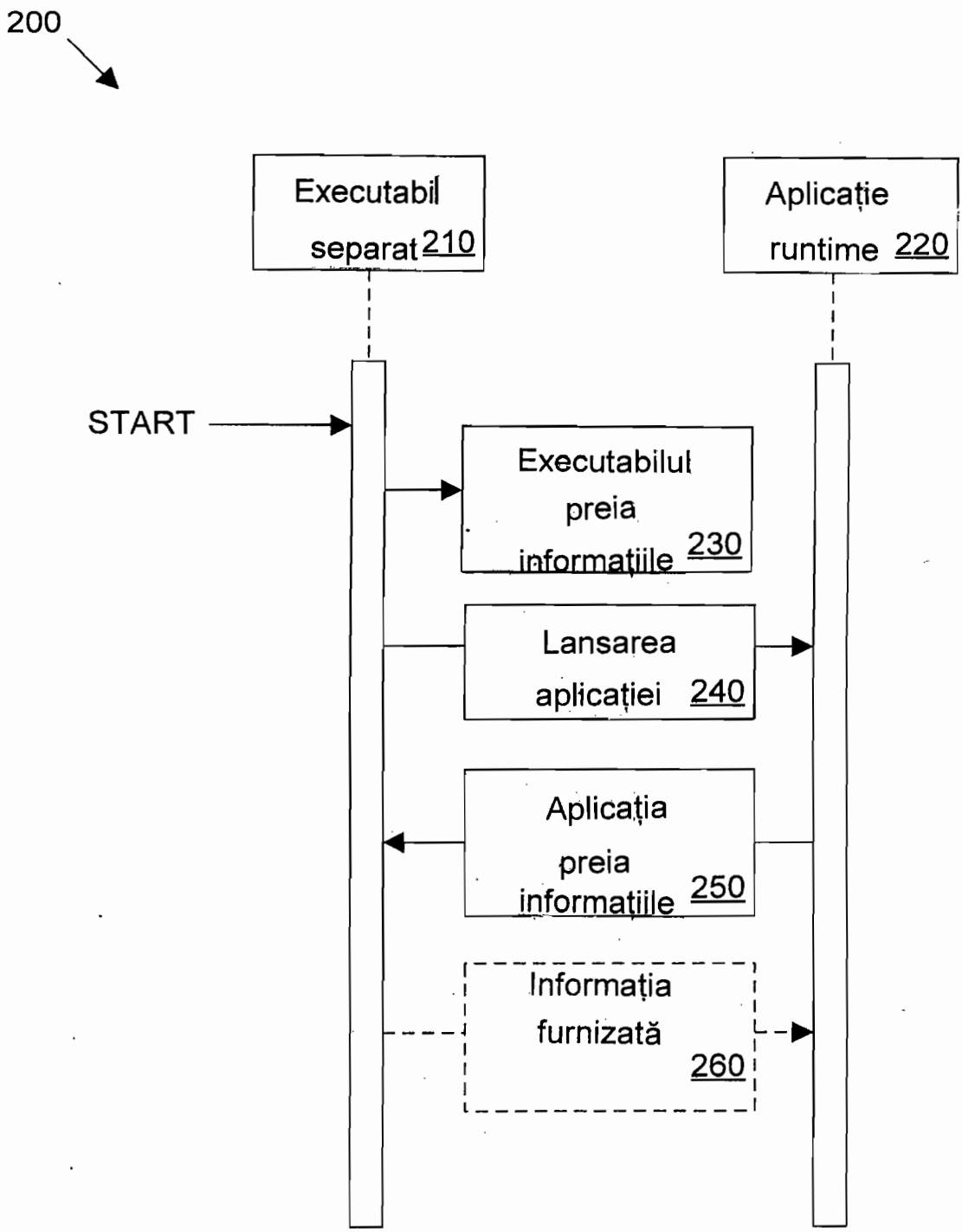
200
Executabil separat 210
Aplicație runtime 220
START
Executabilul preia informațiile 230
Lansarea aplicației 240
Aplicația preia informațiile 250
Informația furnizată 260

FIGURA 2

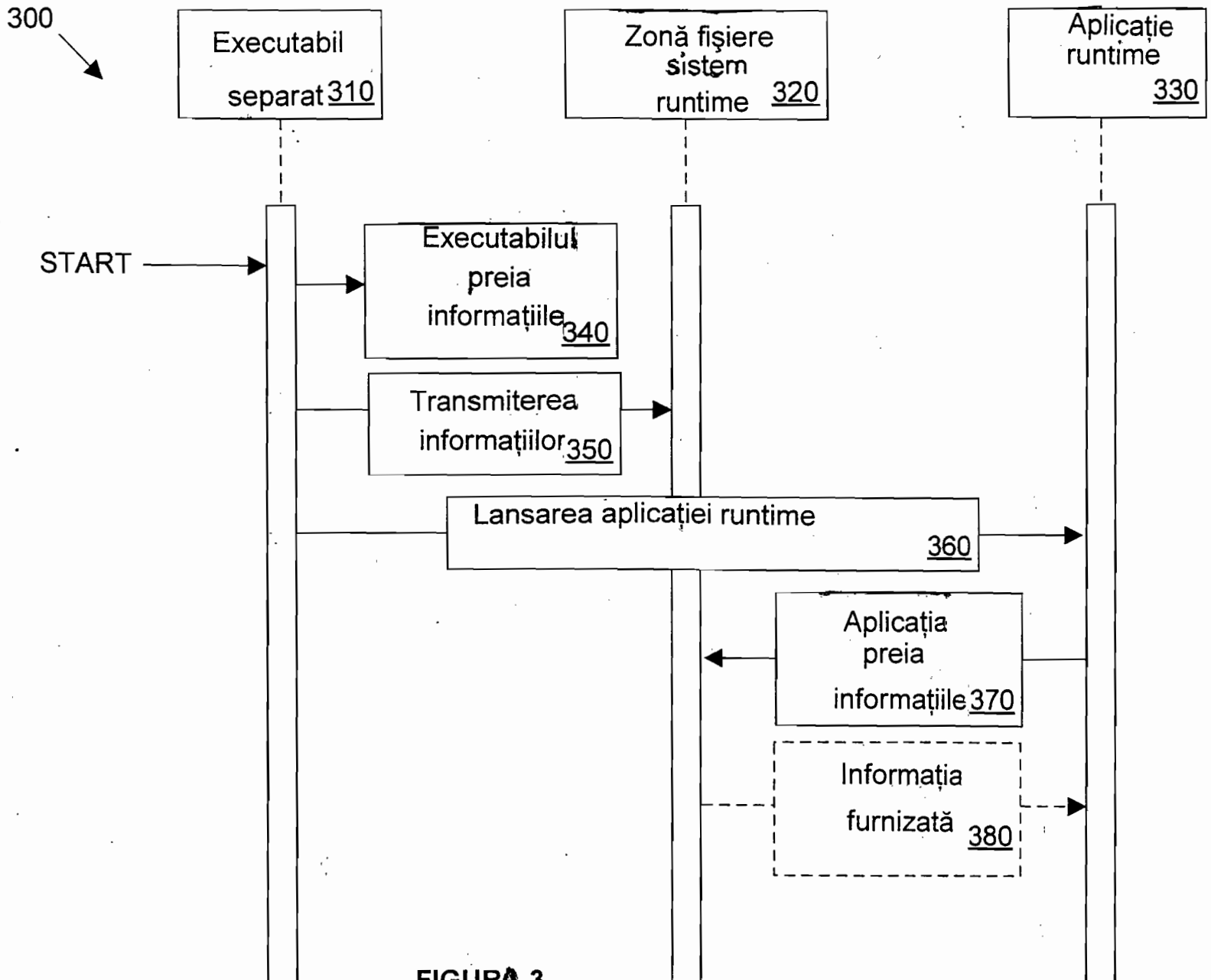

FIGURA 3