



(19)中華民國智慧財產局

(12)發明說明書公告本

(11)證書號數：TW I701596 B

(45)公告日：中華民國 109 (2020) 年 08 月 11 日

(21)申請案號：104142275

(22)申請日：中華民國 104 (2015) 年 12 月 16 日

(51)Int. Cl. : **G06F9/455 (2006.01)****G06F15/16 (2006.01)**

(30)優先權：2015/05/06 中國大陸

201510226829.8

(71)申請人：香港商阿里巴巴集團服務有限公司 (香港地區) ALIBABA GROUP SERVICES
LIMITED (HK)

香港

(72)發明人：李楠 (CN)

(74)代理人：林志剛

(56)參考文獻：

CN 101425021A

CN 102567075B

US 20110173251A1

US 20120239825A1

US 20150120928A1

審查人員：蘇齊賢

申請專利範圍項數：15 項 圖式數：8 共 43 頁

(54)名稱

一種虛擬主機及隔離方法、資源存取請求處理方法及裝置

(57)摘要

本發明實施方式公開了一種虛擬主機隔離方法。該方法包括：確定虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄，將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄；在所述虛擬主機的根目錄下創建公共資源利用目錄，建立所述公共資源利用目錄與網路伺服器根目錄之下的公共資源分享目錄之間的映射關係，以形成虛擬主機用於運行程序的容器，透過所述容器實現虛擬主機之間的隔離。本發明實施方式還公開了虛擬主機以及資源存取請求處理方法及裝置。本發明實施方式可以實現虛擬主機的安全隔離。

指定代表圖：

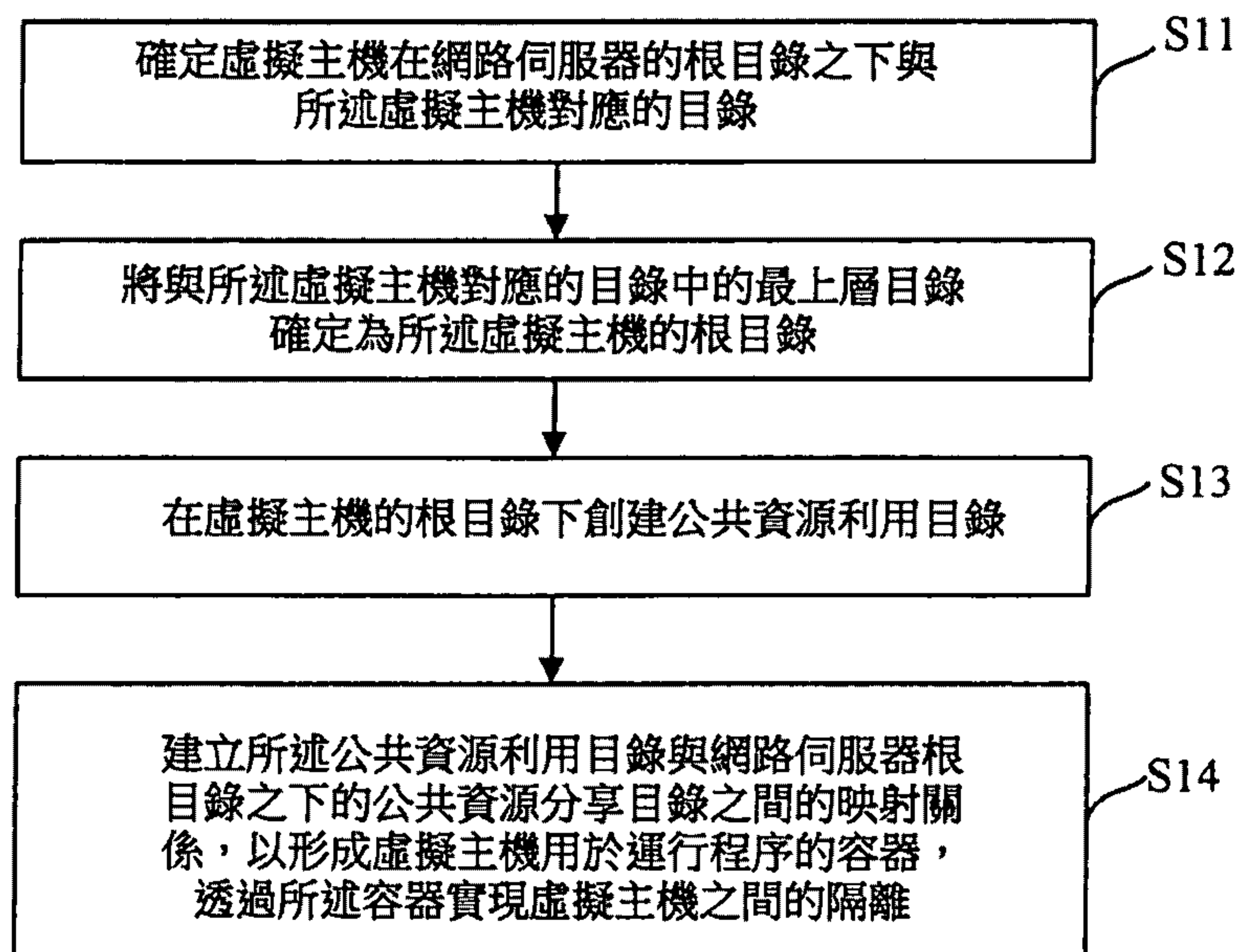


圖 1

I701596

發明摘要

※申請案號：104142275

※申請日：104 年 12 月 16 日

※IPC 分類：

【發明名稱】(中文/英文)

一種虛擬主機及隔離方法、資源存取請求處理方法及裝置

【中文】

本發明實施方式公開了一種虛擬主機隔離方法。該方法包括：確定虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄，將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄；在所述虛擬主機的根目錄下創建公共資源利用目錄，建立所述公共資源利用目錄與網路伺服器根目錄之下的公共資源分享目錄之間的映射關係，以形成虛擬主機用於運行程序的容器，透過所述容器實現虛擬主機之間的隔離。本發明實施方式還公開了虛擬主機以及資源存取請求處理方法及裝置。本發明實施方式可以實現虛擬主機的安全隔離。

【英文】

【代表圖】

【本案指定代表圖】：第(1)圖。

【本代表圖之符號簡單說明】：無

【本案若有化學式時，請揭示最能顯示發明特徵的化學式】：無

發明專利說明書

(本說明書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

一種虛擬主機及隔離方法、資源存取請求處理方法及裝置

【技術領域】

本發明係關於網路伺服器技術領域，尤其係關於一種虛擬主機隔離方法、虛擬主機、資源存取請求處理方法及其對應裝置。

【先前技術】

為了充分利用網路伺服器的運算處理、儲存空間等資源，一個網路伺服器可以透過一定的技術同時為多個資源使用者提供相對獨立的服務，這種情況下，相當於把一個網路伺服器劃分成了多個“子網路伺服器”，常見的虛擬主機即是這種資源利用方式的一種實現。位於網路伺服器上的各個虛擬主機具有相對獨立的系統資源，比如 IP 位址、磁碟空間、記憶體、CPU 使用時間等，從而可以對外提供完整的 WEB 伺服器功能。在虛擬主機以外的物件看來，每一台虛擬主機和一台獨立的伺服器沒有什麼特別不同。

網路伺服器透過其自身的各個虛擬主機向資源使用者提供 WEB 服務，但是，由於多個虛擬主機位於同一伺服

器之上，原有的防火牆等網管設備已不復存在，虛擬主機之間的邊界不甚明確，相互之間存在干擾的可能，從而導致出現安全問題。比如，網路伺服器上的某個虛擬主機出現駭客入侵，不僅影響到該虛擬主機自身的安全，還可能對該虛擬主機以外的其他虛擬主機造成安全威脅。為了使網路伺服器上的虛擬主機相互獨立，需要對虛擬主機進行隔離。在現有技術中，一種隔離方式是對虛擬主機的資源使用者的存取權限透過專門配置進行限制，只有在預先配置範圍內的資源才允許存取。然而，對存取權限的配置通常透過某種特定語言（比如，PHP 語言）來實現，這導致對虛擬主機的隔離停留在語言級別，不使用該配置語言的使用者透過其他語言可以容易繞過許可權配置，從而導致虛擬主機之間沒有實現真正隔離，仍然存在安全風險。

【發明內容】

為了解決現有技術中的技術問題，本發明實施方式提供一種網路伺服器的虛擬主機隔離方法、虛擬主機以及資源存取請求處理方法及其裝置，以實現虛擬主機之間的安全隔離，改善或提高資源存取請求處理過程的安全性能。

本發明實施方式提供了一種網路伺服器的虛擬主機隔離方法，包括：

確定虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄，將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄；

在所述虛擬主機的根目錄下創建公共資源利用目錄，建立所述公共資源利用目錄與網路伺服器根目錄之下的公共資源分享目錄之間的映射關係，以形成虛擬主機用於運行程序的容器，透過所述容器實現虛擬主機之間的隔離。

較佳地，所述將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄包括：

調用根目錄改變函數將與所述虛擬主機對應的目錄中的最上層目錄修改為所述虛擬主機的根目錄。

較佳地，所述建立所述公共資源利用目錄與網路伺服器的根目錄之下的公共資源分享目錄之間的映射關係包括：

調用目錄掛載函數將網路伺服器的根目錄之下的公共資源分享目錄以唯讀掛載的方式掛載到所述虛擬主機的公共資源利用目錄。

較佳地，所述方法還包括：使利用網路伺服器和/或虛擬主機資源的一個或多個程序在虛擬主機的容器內運行，在一個虛擬主機的容器內運行的程序與其他虛擬主機的容器內運行的程序具有不同的程序命名空間，在虛擬主機的容器內運行的程序在相同程序命名空間內具有不同的程序標識。

較佳地，所述使利用網路伺服器和/或虛擬主機資源的一個或多個程序在虛擬主機的容器內運行包括：

將待在虛擬主機內運行的程序的原始根目錄修改為虛擬主機的根目錄，以及將待在虛擬主機內運行的程序的程

序標識修改為按照虛擬主機的程序命名空間命名的程序標識，以使程序在虛擬主機的容器內運行。

較佳地，虛擬主機的容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照當前虛擬主機的程序命名空間命名程序標識，所述使利用網路伺服器 and /或虛擬主機資源的一個或多個程序在虛擬主機的容器內運行包括：

繼承所述容器管理器程序，以產生在虛擬主機的容器內運行的利用網路伺服器和 /或虛擬主機資源的一個或多個程序。

本發明實施方式還提供了一種網路伺服器的虛擬主機。該虛擬主機包括用於運行程序的容器，所述容器的根目錄為所述虛擬主機的根目錄，所述虛擬主機的根目錄為所述虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄中的最上層目錄，在所述虛擬主機的根目錄下具有公共資源利用目錄，所述公共資源利用目錄與網路伺服器的根目錄之下的公共資源分享目錄之間具有映射關係。

較佳地，在所述容器內能運行利用網路伺服器和 /或虛擬主機資源的一個或多個程序，在一個虛擬主機的容器內運行的程序與其他虛擬主機的容器內運行的程序具有不同的程序命名空間，在虛擬主機的容器內運行的程序在相同程序命名空間內具有不同的程序標識。

較佳地，所述容器內具有容器管理器程序，所述容器

管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照當前虛擬主機的程序命名空間命名程序標識。

本發明實施方式還提供了一種資源存取請求處理方法，所述方法應用於包括上述虛擬主機的網路伺服器，該方法包括：

接收網路伺服器轉發的存取動態資源的資源存取請求，所述存取動態資源的資源存取請求包括待存取的動態資源所在的虛擬主機的標識；

在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結果返回給網路伺服器，以便網路伺服器返回給發送所述資源存取請求的請求端。

較佳地，所述在虛擬主機的容器內處理所述存取動態資源的資源存取請求包括：

將第一腳本解譯器程序的原始根目錄修改為所述虛擬主機的根目錄，以及將所述第一腳本解譯器程序的程序標識修改為按照所述虛擬主機的程序命名空間命名的程序標識，以使第一腳本解譯器程序在所述虛擬主機的容器內運行；

透過在虛擬主機的容器內運行的第一腳本解譯器程序處理所述存取動態資源的資源存取請求。

較佳地，所述虛擬主機的容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照當前虛擬主機的程序命名空間命名程序標識，則所述在虛擬主機的容器內處理所述存取動

態資源的資源存取請求包括：

在接收到網路伺服器轉發的存取動態資源的資源存取請求後，繼承所述容器管理器程序，以產生第二腳本解譯器程序；

透過在虛擬主機的容器內運行的第二腳本解譯器程序處理所述存取動態資源的資源存取請求。

較佳地，在接收到網路伺服器轉發的存取動態資源的資源存取請求之前，向網路伺服器發送所述容器管理器程序的註冊請求，所述註冊請求包括所述容器管理器程序的程序標識以及虛擬主機的標識；

接收網路伺服器在建立所述容器管理器程序的程序標識與所述虛擬主機的標識之間的對應關係後發送的註冊成功消息，以便在接收到所述註冊成功消息後，接收網路伺服器根據所述對應關係轉發給所述容器管理器程序的存取動態資源的資源存取請求。

較佳地，所述資源存取請求為存取靜態資源的資源存取請求時，網路伺服器讀取所述靜態資源，以便返回給發送所述資源存取請求的請求端。

本發明實施方式還提供了一種資源存取請求處理裝置，所述裝置應用於包括上述虛擬主機的網路伺服器，該裝置包括存取請求接收單元和存取請求處理單元，其中：

所述存取請求接收單元，用於接收網路伺服器轉發的存取動態資源的資源存取請求，所述存取動態資源的資源存取請求包括待存取的動態資源所在的虛擬主機的標識；

所述存取請求處理單元，用於在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結果返回給所述網路伺服器，以便網路伺服器返回給發送所述資源存取請求的請求端。

較佳地，所述存取請求處理單元包括第一腳本解譯器程序創建子單元和第一存取請求處理子單元，其中：

所述第一腳本解譯器程序創建子單元，用於將第一腳本解譯器的原始根目錄修改為所述虛擬主機的根目錄，以及將所述第一腳本解譯器程序的程序標識修改為按照所述虛擬主機的程序命名空間命名的程序標識，以使第一腳本解譯器程序在所述虛擬主機的容器內運行；

所述第一存取請求處理子單元，用於透過在虛擬主機的容器內運行的第一腳本解譯器程序處理所述存取動態資源的資源存取請求。

較佳地，所述虛擬主機的容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照當前虛擬主機的程序命名空間命名程序標識，所述存取請求處理單元包括第二腳本解譯器程序創建子單元和第二存取請求處理子單元，其中：

所述第二腳本解譯器程序創建子單元，用於在接收到網路伺服器轉發的存取動態資源的資源存取請求後，繼承所述容器管理器程序，以產生第二腳本解譯器程序；

所述第二存取請求處理子單元，用於透過在虛擬主機的容器內運行的第二腳本解譯器程序處理所述存取動態資

源的資源存取請求。

較佳地，所述裝置包括註冊單元，用於在接收到網路伺服器轉發的存取動態資源的資源存取請求之前，向網路伺服器發送所述容器管理器程序的註冊請求，所述註冊請求包括所述容器管理器程序的程序標識以及虛擬主機的標識；所述存取請求接收單元，還具體用於接收網路伺服器在建立所述容器管理器程序的程序標識與所述虛擬主機的標識之間的對應關係後發送的註冊成功消息，以便在接收到所述註冊成功消息後，接收網路伺服器根據所述對應關係轉發給所述容器管理器程序的存取動態資源的資源存取請求。

本發明實施方式提供的虛擬主機隔離方法、虛擬主機建立（或擁有）屬於虛擬主機自身的目錄體系，在虛擬主機內建立（或擁有）用於運行程序的容器，與現有技術相比，透過該方式使利用網路伺服器資源和/或虛擬主機資源的程序的根目錄為虛擬主機的根目錄，且運行在容器內，具有較大的獨立性，不會和其他虛擬主機的程序發生相互干擾，從而確保了虛擬主機之間的安全隔離，有利於改善或增強利用所述虛擬主機進行資源存取請求處理的安全性。

【圖式簡單說明】

此處所說明的附圖用來提供對本發明的進一步理解，構成本發明的一部分，本發明的示意性實施例及其說明用

於解釋本發明，並不構成對本發明的不當限定。在附圖中：

圖 1 為本發明的網路伺服器的虛擬主機隔離方法的實施例的流程示意圖；

圖 2 (a) 為 Linux 系統的目錄結構示意圖；

圖 2 (b) 為 Linux 系統中包含虛擬主機的目錄結構示意圖；

圖 3 為本發明虛擬主機隔離方法的形象化實施過程示意圖；

圖 4 為本發明的資源存取請求處理方法的一個實施例流程圖；

圖 5 為本發明的資源存取請求處理方法的又一個實施例流程圖；

圖 6 (a) 為網路伺服器與腳本解譯器程序的通信過程示意圖；

圖 6 (b) 為網路伺服器與 CPM 程序的通信過程示意圖；

圖 7 (a) 為本發明的資源存取請求處理方法再一個實施例流程圖；

圖 7 (b) 為本發明的資源存取請求處理過程的信令交互圖；

圖 8 為本發明的資源存取請求處理裝置的實施例的結構框圖。

【實施方式】

為使本發明的目的、技術方案和優點更加清楚，下面將結合本發明的具體實施例及相應的附圖對本發明技術方案進行清楚、完整地描述。顯然，所描述的實施例僅是本發明一部分實施例，而不是全部的實施例。基於本發明中的實施例，本領域普通技術人員在沒有做出創造性勞動前提下所獲得的所有其他實施例，都屬於本發明保護的範圍。

如前所述，一台網路伺服器上可以存在多個虛擬主機，這些虛擬主機就如同一間大房屋（類似於網路伺服器）中的各個出租房間（類似於虛擬主機）。通常情況下，住在各個出租房間內的住戶（類似於虛擬主機的使用者）互不瞭解，無法判斷自己出租房間的相鄰房間的住戶是否存在惡意或者存在惡意的可能。如果該相鄰出租房間被竊賊破門而入，那麼極有可能從該相鄰出租房間的窗戶（類似於網路伺服器/虛擬主機上的系統漏洞等）鑽入到自己的出租房間，從而對自己的出租房間造成威脅。由此可見，位於同一個網路伺服器上的多個虛擬主機之間應當進行有效隔離，使它們相互之間的影響減少到最小，甚至消除相互影響。為此，本發明提供了一種虛擬主機的隔離方法的實施例。參見圖 1，該圖示出了對網路伺服器上的虛擬主機進行安全隔離的方法的流程，該流程包括：

步驟 S11：確定虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄；

網路伺服器具有自身的目錄結構體系。在其目錄結構體系中，最上面的一層目錄為網路伺服器的根目錄，在根目錄之下存在適應各種實際需要的子目錄，子目錄之下又可能存在孫目錄，孫目錄之下又可能存在孫孫目錄等，同時，在一個層級上，也可能存在多個目錄，這些由橫向、縱向的各級、各層目錄形成的目錄結構體系如同“樹形”一般，因此，網路伺服器的目錄結構體系又被稱為目錄樹。以網路伺服器的系統為 Linux 系統為例，其目錄結構體系如圖 2（a）所示。在該圖中，“/”表示網路伺服器的根目錄，在根目錄“/”的下一層級存在多個“子目錄”，如圖中所示的“bin/”、“etc/”、“lib/”、“usr/”、“home/”等。在這種目錄結構體系中，由於網路伺服器具有多個虛擬主機，每個虛擬主機在網路伺服器的目錄結構體系中具有與其自身對應的目錄。參見圖 2（b）所示，在網路伺服器的目錄結構體系中，根目錄“/”下有“home/”目錄，在“home/”目錄下有“alice/”目錄及其以下的目錄（包括“etc/”、“dev/”、“lib/”等目錄）。該“alice/”目錄及其以下的目錄對應“alice”使用者的虛擬主機目錄，當然在“home/”目錄下，還可以具有其他虛擬主機的使用者對應的虛擬主機目錄。在本發明實施方式中，確定與所述虛擬主機對應的目錄便是確定虛擬主機在網路伺服器的目錄體系結構中屬於虛擬主機的目錄，如上例中的“alice/”目錄以及“alice/”目錄之下的“etc/”、“dev/”、“lib/”等目錄。這些目錄同樣形成一個與虛擬主機對應的目錄結構體系。

步驟 S12：將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄；

透過前述步驟確定了與虛擬主機對應的目錄後，將該目錄中的最上層目錄確定為虛擬主機的根目錄，如上例中，將“alice/”目錄確定為與“alice”虛擬主機對應的根目錄。在現有技術中，“alice”虛擬主機的根目錄為整個網路伺服器的根目錄，即“/”。為實現這種根目錄的改變，可以使用多種具體方式。比如，在本發明的一個實施例中，可以透過調用根目錄改變函數將虛擬主機對應的目錄中的最上層目錄修改為虛擬主機的根目錄，從而將虛擬主機的根目錄由原來網路伺服器的根目錄改變為與該虛擬主機對應的根目錄。以 Linux 系統為例，根目錄改變函數可以是 chroot 系統函數，其調用方式是“chroot（‘目標根目錄’）”，在上例中目標根目錄為“/home/alice”，將虛擬主機目錄系統的根目錄確定後，在該根目錄之外的檔將無法“看見”該根目錄之下的內容。如果網路伺服器上具有多個虛擬主機，那麼每個虛擬主機都可以透過上述方式獲得自己專屬的“根目錄”，而不用再共用網路伺服器的根目錄。

步驟 S13：在所述虛擬主機的根目錄下創建公共資源利用目錄；

透過前述步驟確定虛擬主機的根目錄後，便可以在該虛擬主機的根目錄下創建各種子目錄，以便某些程序使用這些目錄或者目錄中的資源。在這些子目錄中，可以包括公共資源利用目錄，該目錄的創建可以使虛擬主機的程序

利用某些公共資源，比如，網路伺服器的作業系統中的系統設定檔、系統提供的工具、共用庫檔等。這裡的“公共資源”具有“公共屬性”，不專屬於某個虛擬主機，而可能由兩個或多個虛擬主機共同使用，甚至網路伺服器下的所有虛擬主機均會使用。

步驟 S14：建立所述公共資源利用目錄與網路伺服器根目錄之下的公共資源分享目錄之間的映射關係，以形成虛擬主機用於運程序的容器，透過所述容器實現虛擬主機之間的隔離。

在虛擬主機的根目錄下創建了公共資源利用目錄後，還需要給這些目錄賦予“內容”，但是，由於公共資源是多個虛擬主機要共用的，如果每一個虛擬主機下的公共資源利用目錄均直接將公共資源複製到（或者自身重新建立）自身根目錄之下，那麼將導致網路伺服器的整個目錄結構體系非常臃腫，為了避免這種情況的出現，本實施例採取的方式是在虛擬主機的公共資源利用目錄與網路伺服器根目錄之下的公共資源分享目錄之間建立映射關係，這樣在某些程序要使用虛擬主機的公共資源利用目錄時，便可透過這種映射關係“連結”到網路伺服器根目錄之下的公共資源分享目錄。在實際應用過程中，可以有多種創建這種映射關係的方式，比如，在 Linux 作業系統中，可以透過調用目錄掛載函數（比如，mount 系統調用的 bind 功能）將一個目錄掛載到另一個目錄下，從而實現兩類目錄之間映射關係的建立。

本實施例透過前述步驟為虛擬主機建立了獨立的目錄結構體系，使其根目錄與其他虛擬主機的根目錄各自隔離，而不在共同擁有網路伺服器的根目錄，而且，對公共資源的利用也僅僅直接映射到網路伺服器的根目錄下的公共資源分享目錄，而不會與其他虛擬主機的根目錄下的公共資源利用目錄發生關聯，從而將虛擬主機的存取權限限制在一個固定的目錄下，這種方式使各個虛擬主機之間形成了一個較為獨立的容器，透過該容器將不同的虛擬主機之間進行了有效隔離，達到一個虛擬主機下的程序無法存取另一個虛擬主機下的資源的目的。

下面結合圖 3 進行詳細說明。圖 3 示出了本發明實施方式針對 Linux 系統的一種形象化實施過程。在網路伺服器的目錄結構體系中，對於“alice”這個虛擬主機而言，其原本的目錄結構體系是“alice/etc/、alice/dev/、alice/lib/、alice/htdocs/”等，這些目錄均以網路伺服器的根目錄“/”為根目錄，但是，透過根目錄改變函數 chroot（“home/alice”）可以將 alice 虛擬主機的根目錄改變為“/home/alice/”，相當於將“/home/alice/”整體看成虛擬主機原來所視為的根目錄“/”，即如圖 3 中實線箭頭所示，透過這個新的根目錄相當於在虛擬主機上建立了與 alice 虛擬主機對應的一個容器（alice's container），該容器與其他虛擬主機是相互隔離的。此外，在虛擬主機的根目錄之下，還可以建立“etc/、dev/、lib/、htdocs/”等目錄，這些目錄可以按照性質劃分為兩類：一類是在虛擬機器內部使

用且不依賴於虛擬主機之外的其他目錄的目錄，在本例中即專屬於“alice”虛擬主機的目錄；一類是在虛擬主機內部使用但使用過程必須依賴於虛擬主機之外的其他目錄的目錄。對於第二類目錄，必須建立與當前虛擬主機以外的目錄與虛擬主機相應目錄之間的映射關係，這樣，在需要利用到虛擬主機下的這些目錄時，便可透過映射關係“連結”到當前虛擬主機以外的其他目錄。虛擬主機上的公共資源利用目錄即屬於第二類目錄。虛擬主機上的公共資源利用目錄在使用過程中需要借助於網路伺服器提供的公共資源分享目錄，為此需要建立當前虛擬主機的公共資源利用目錄與網路伺服器的公共資源分享目錄之間的映射關係，即如圖 3 中的虛箭頭所示。

本發明實施例提供的虛擬主機隔離方法，使虛擬主機建立（或擁有）屬於虛擬主機自身的目錄體系，在虛擬主機內建立（或擁有）用於運行程序的容器，與現有技術相比，透過該方式使利用網路伺服器資源和/或虛擬主機資源的程序的根目錄為虛擬主機的根目錄，且運行在容器內，具有較大的獨立性，不會和其他虛擬主機的程序發生相互干擾，從而確保了虛擬主機之間的安全隔離。

在上述實施例中，透過對虛擬主機根目錄的變化以及公共資源利用目錄的建立，形成了與虛擬主機對應的容器（該容器從虛擬主機的使用者的維度進行創建，把不同虛擬主機的程序和資源配置在不同的容器內），使不同虛擬主機之間透過容器實現了隔離。在實際應用過程中，還可

以透過程序命名空間（PID Namespace）進行更深層的隔離（比如，程序級別的隔離）。具體地，可以使在虛擬主機容器內運行的全部程序具有統一（相同）的程序命名空間，但在該命名空間內每個程序具有不同的程序 ID（程序標識），可以透過程序 ID 來實現檢索，而在虛擬主機與虛擬主機之間程序命名空間則各不相同，這樣，不同虛擬主機內運行的程序便可以相互隔離，從而不會出現一個虛擬主機的程序可以殺掉、調試另外一個虛擬主機內的程序。需要注意的是，雖然在不同容器內的程序命名體系不同，但並不意味著不同容器的程序在功能、結構、作用等方面一定存在差異，也就是說，可能存在這樣一種情形，一個程序可以加入多個容器，從而具有多個獨立的 PID Namespace。

與虛擬主機的目錄結構體系相對應的，容器的程序命令體系也可能形成一個命名空間樹，不同虛擬主機容器內的程序可以位於 PID Namespace 樹中同一級的不同的 PID Namespace 中，而網路伺服器的程序命名體系對應地可以是所有虛擬主機的 PID Namespace 的父 Namespace，這樣網路伺服器可以看見所有虛擬主機下容器內的程序，但某個虛擬主機則無法看到別的虛擬主機或網路伺服器上的程序。

在上述這種虛擬主機的隔離方法之下，形成了與虛擬主機對應的容器，該容器內可以用於運行各種利用網路伺服器和/或虛擬主機資源的程序。要使程序在虛擬主機的

容器內運行可以透過多種方式實現，具體採用哪種方式取決於程序自身的情況。比如，如果某個程序先前（或者現有技術中）是在容器外（或者沒有容器的情形下）運行的，則可以將該程序放入到容器內運行，具體地，將待在虛擬主機內運行的程序的原始根目錄修改為虛擬主機的根目錄，以及將待在虛擬主機內運行的程序的程序標識修改為按照虛擬主機的程序命名空間命名的程序標識。透過對程序的原始根目錄和程序標識進行修改，實現了該程序對容器的適應，從而達到了將程序放入虛擬主機容器內運行的目的。

還比如，在某些情況下，虛擬主機的容器內具有容器管理器程序，該容器管理器程序的根目錄為虛擬主機的根目錄，容器管理器程序也按照虛擬主機的程序命名空間命名程序標識（這些工作可以在容器管理器程序創建時進行完畢），這時，要使利用網路伺服器和/或虛擬主機資源的一個或多個程序在虛擬主機的容器內運行可以採取如下方式：繼承所述容器管理器程序，以產生在虛擬主機的容器內運行的利用網路伺服器和/或虛擬主機資源的一個或多個程序。透過對容器管理器程序進行繼承，容器管理器程序的根目錄以及程序命名空間也將繼承在產生的新程序之中，從而實現了該程序對容器的適應，達到在虛擬主機容器內運行的目的。

透過上述虛擬主機的隔離方法的實施例的描述可以看出，實際上已經形成了一種獨特的虛擬主機，該虛擬主機

包括用於運行程序的容器，包括這種容器的虛擬主機至少具有如下兩個特點，一是該容器的根目錄為虛擬主機的根目錄，所述虛擬主機的根目錄為所述虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄中的最上層目錄，二是在所述虛擬主機的根目錄下具有公共資源利用目錄，所述公共資源利用目錄與網路伺服器的根目錄之下的公共資源分享目錄之間具有映射關係。在這種虛擬主機中，虛擬主機使用者需要的系統資源在容器中，利用資源的程序也在虛擬主機容器內運行，不在當前虛擬主機容器中的資源對於當前虛擬主機而言，不可見，這一方面使得在當前虛擬主機容器內運行的程序僅能存取到該虛擬主機容器內的資源，而不能存取到其他虛擬主機中的資源，另一方面在其他虛擬主機中運行的程序也無法存取到該虛擬主機的程序，從而真正實現了不同虛擬主機之間的隔離。

在上述這種虛擬主機中，虛擬主機容器內可以運行一個或多個程序，這些程序可以為一種“平等”地位關係，也可以為“不平等”的地位關係。在“平等”的情形下，如果同時在容器內運行有多個程序，就容器內的各個程序而言，這些程序在相同程序命名空間內具有不同的程序標識，就當前容器與其他容器內的程序而言，各個程序具有不同的程序命名空間。在“不平等”的情形下，可以設置一個容器管理器程序（CPM，Container Process Manager），由該程序來管理容器內的各個程序。比如，將 CPM 程序作為祖先程序，在容器內運行的其他程序都根據該程序產生，

從中繼承包括程序根目錄和程序命名空間在內的屬性（該過程可以成為“fork”過程），透過這種 fork 過程可以保證容器內運行的所有程序都在容器內。透過這種方式產生的多個程序之間地位不平臺，CPM 程序處於管理地位，其他程序為 CPM 程序的子程序。

上述內容詳細描述了網路伺服器上的不同虛擬主機之間的安全隔離方法以及由此形成的獨特的虛擬主機。在上述這種虛擬主機上，可以實現對來自客戶端的資源存取請求的處理。通常而言，在網路伺服器上，存在兩類不同性質的資源：一是靜態資源，比如，HTML 檔、CSS 檔、JS 檔、圖片檔等，這些檔資源可以由網路伺服器直接讀取，並返回給資源存取請求的發起端（比如，客戶端）；二是動態資源（又稱為腳本），這類資源需要由網路伺服器上的腳本解譯器進行解釋運行，經過解釋運行後產生的結果作為輸出內容透過網路伺服器返回給客戶端。這兩類資源的性質不同，給網路伺服器帶來的安全性程度存在差異。對於靜態資源而言，由於不需要透過網路伺服器上的某個程式（程序）進行解釋運行，不進行諸如運算、加工、處理等工作，而是直接由網路伺服器回饋給客戶端，因此，相對比較安全，即使靜態資源中已帶有病毒、木馬等危險程式，因網路伺服器本身不對該資源進行執行操作，將不會對網路伺服器造成安全威脅。但是，對於動態資源而言，則存在較大差別，其需要執行解釋運行操作，在執行過程中可能導致系統的配置許可權被改變、動態資源上攜

帶的病毒程式被引發感染到網路伺服器（的虛擬主機），如果對此不加以控制，將存在極大的安全風險。

為此，本發明還提供了一種資源存取請求處理方法的實施例，該處理方法應用於包括前述所述的虛擬主機的網路伺服器。參見圖 4，該圖示出了本發明的資源存取請求處理方法的一種實施例的流程圖，該流程包括：

步驟 S41：接收網路伺服器轉發的存取動態資源的資源存取請求，所述資源存取請求包括待存取的動態資源所在的虛擬主機的標識；

步驟 S42：在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結果返回給所述網路伺服器，以便網路伺服器返回給發送所述資源存取請求的請求端。

在該實施例中，由於是將動態資源存取請求的處理安排在虛擬主機的容器內進行處理，而虛擬主機相互之間透過容器進行了隔離，這樣便保證了虛擬主機的安全。在上述資源存取請求的處理實施例中主要是從網路伺服器的某個虛擬主機的角度對其過程進行描述，實際上，整個處理流程還可以站在網路伺服器的角度進行描述，從而形成又一個實施例（參見圖 5）：

首先是網路伺服器接收到來自客戶端的資源存取請求，該資源存取請求中包括待存取的動態資源所在的虛擬主機的標識，然後，由網路伺服器對該存取請求的類型進行判斷，如果該資源存取請求為動態資源請求，則根據動態資源存取請求中的虛擬主機的標識資訊轉發到對應的虛

擬主機上。該虛擬主機接收到該動態資源存取請求後，在容器內運行腳本解譯器，利用該腳本解譯器進行解釋運行。然後將解釋運行的結果回饋給網路伺服器，由網路伺服器將結果回饋給發送資源存取請求的客戶端。

在前述資源存取請求實施例中，提到在虛擬主機的容器內處理存取動態資源的資源存取請求。在實際應用過程中，可能存在多種資源存取請求的處理方式，不同的處理方式與處理資源存取請求的程序有關。下面示例性地給出兩種實施方式，本領域技術人員可以在此基礎上得到其他的實施方式，這些實施方式均在本發明的保護範圍之內。

示例性實施方式之一：改造現有腳本解譯器的方式。動態資源存取請求需要腳本解譯器的解釋執行作用，才能獲得相應的回應結果。參見圖 6（a），在這種方式中，現有的腳本解譯器在虛擬主機中啟動，但是，該腳本解譯器並不當然在虛擬主機創建的容器內運行，為了將該腳本解譯器放入虛擬主機的容器內運行，需要將腳本解譯器的原始根目錄修改為虛擬主機的根目錄，以及將腳本解譯器程序的程序標識修改為按照虛擬主機的程序命名空間命名的程序標識，從而使腳本解譯器程序在虛擬主機的容器內運行，這樣，便可以利用該改造後的腳本解譯器程序處理動態資源存取請求。這種方式由於將腳本解譯器放入到了虛擬主機的容器內，而容器具有較高的安全性，從而保證了動態資源存取的安全性。

示例性實施方式之二：產生新腳本解譯器的方式。在

上一種示例性實施方式中，透過改變腳本解譯器的根目錄和程序命名空間實現了動態資源存取請求的安全處理。但是，這種方式需要修改腳本解譯器的原始程式碼，侵入性較強，耦合性較高。而且，對於不同的動態資源存取請求，還需要維護多個腳本解譯器版本。為了降低系統耦合度，本發明提供了另外一種方式，參見圖 6（b），該方式在虛擬主機的容器內具有容器管理器程序，該容器管理器程序的根目錄為虛擬主機的根目錄，容器管理器程序的程序標識按照虛擬主機的程序命名空間命名，這種情況下，在接收到網路伺服器轉發的存取動態資源的資源存取請求後，繼承容器管理器程序，以產生一個腳本解譯器（為了語言敘述上的方便，該新產生的腳本解譯器可以稱為第二腳本解譯器，前述示例性方式中稱為第一腳本解譯器），從而透過該新的腳本解譯器進行動態資源存取請求的處理。

在上述第二種示例性實施方式中提及存在一個 CPM 程序（容器管理器程序）。通常而言，該程序可以透過虛擬主機作為“橋樑”與網路伺服器（網路伺服器程序）實現通信，即網路伺服器接收到客戶端的資源存取請求後，由於該請求中具有待存取資源在哪個虛擬主機的標識（對於動態資源而言，如果存取的資源是靜態資源，資源存取請求中為該資源所在位置的位址資訊），從而可以將存取請求傳遞到虛擬主機上，再由虛擬主機將該請求交給該虛擬主機的容器管理器程序，進而啟動容器管理器程序進行新

腳本解譯器的產生步驟，以及隨後利用產生的腳本解譯器對資源存取請求的處理步驟，從而實現動態資源的處理。

上述透過虛擬主機的橋樑作用實現了網路伺服器與 CPM 程序之間的通信，但是，為了減少通信的中間環節以及增強安全性，在本發明的另一個實施例中，可以在網路伺服器與 CPM 程序之間建立直接的通信關係，從而可以在網路伺服器接收到資源存取請求後，如果該資源存取請求是對動態資源的存取請求，那麼可以由網路伺服器直接將需要腳本解譯器運行器進行的工作委託給 CPM 程序。基於此，需要建立網路伺服器與 CPM 之間的通信協議。在本發明中，可以按照如下方式進行建立：

CPM 程序向網路伺服器發送容器管理器程序的註冊請求，該註冊請求包括所述容器管理器程序的程序標識以及虛擬主機的標識；

網路伺服器接收到註冊請求後，建立所述容器管理器程序的程序標識與所述虛擬主機的標識之間的對應關係，比如，記錄 CPM 程序的程序標識和虛擬主機的標識，然後向 CPM 程序發送註冊成功的消息。

經過上述註冊過程，如果網路伺服器接收到資源存取請求後，經判斷，該資源存取請求為動態資源存取請求，則根據所述對應關係直接將該請求轉發給相應的 CPM 程序，CPM 接收到該請求後，根據 CPM 程序創建腳本解譯器程序，然後運行腳本，再將運行結果返回給網路伺服器，在網路伺服器收到 CPM 返回結果後，回應客戶端的

請求。上述資源存取請求的處理過程以及 CPM 程序的註冊過程可以透過圖 7 反映出來，其中：圖 7 (a) 為本發明的資源存取請求處理方法的再一個實施例的流程示意圖，圖 7 (b) 為本發明的資源存取請求處理過程的信令交互圖。在該圖中，如果資源存取請求為對靜態資源的存取，則可以如圖所示，由網路伺服器直接讀取靜態檔的內容，然後將其作為結果回應請求端。

上述內容詳細敘述了本發明基於隔離的虛擬主機進行資源存取請求方法實施例，與此對應地，本發明還提供了資源存取請求處理的裝置實施例。參見圖 8，該圖示出了本發明的資源存取請求處理裝置的結構框圖。該處理裝置應用於包括前述所述的虛擬主機的網路伺服器，該裝置包括存取請求接收單元 U81 和存取請求處理單元 U82，其中：

存取請求接收單元 U81，用於接收網路伺服器轉發的存取動態資源的資源存取請求，所述資源存取請求包括待存取的動態資源所在的虛擬主機的標識；

存取請求處理單元 U82，用於在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結果返回給所述網路伺服器，以便網路伺服器返回給發送所述存取請求的請求端。

上述資源存取請求處理裝置的工作過程是：存取請求接收單元 U81 接收網路伺服器轉發的存取動態資源的資源存取請求，然後，由存取請求處理單元 U82 在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結

果返回給所述網路伺服器，以便網路伺服器將其返回給發送所述資源存取請求的請求端。該處理裝置將對動態資源的存取請求在虛擬主機創建的容器內進行處理，而虛擬主機之間透過容器實現了隔離，從而改善或提高了資源存取請求處理過程的安全性能。

儘管上述裝置實施例已能較好地實現本發明的發明目的，但是，基於各種實際應用需要，可以對上述裝置實施例進行改進，從而進一步優化上述裝置實施例的技術效果。對上述裝置實施例的改進可以採取多種方式，比如，對於存取請求處理單元，可以包括第一腳本解譯器程序創建子單元和第一存取請求處理子單元 U82，其中：所述第一腳本解譯器程序創建子單元，用於將第一腳本解譯器的原始根目錄修改為所述虛擬主機的根目錄，以及將所述第一腳本解譯器程序的程序標識修改為按照所述虛擬主機的程序命名空間命名的程序標識，以使第一腳本解譯器程序在所述虛擬主機的容器內運行；所述第一存取請求處理子單元，用於透過在虛擬主機的容器內運行的第一腳本解譯器程序處理所述存取動態資源的資源存取請求。這種方式將原有的腳本解譯器進行了改造，從而適應了虛擬主機的容器，能夠在容器內處理存取請求，保證了安全性。

除了上述這種對存取請求的處理外，在實際應用過程中，對於存取請求處理單元 U82，在某些情況下，比如在虛擬主機的容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄、程序標識按照當前虛

擬主機的程序命名空間命名程序標識的情況下，存取請求處理單元 U82 可以包括第二腳本解譯器程序創建子單元 U821 和第二存取請求處理子單元 U822，其中：

第二腳本解譯器程序創建子單元 U821，用於在接收到網路伺服器轉發的存取動態資源的資源存取請求後，繼承所述容器管理器程序，以產生第二腳本解譯器程序；

第二存取請求處理子單元 U822，用於透過在虛擬主機的容器內運行的第二腳本解譯器程序處理所述存取動態資源的資源存取請求。

上述這種存取請求處理單元的結構相對於對現有的腳本解釋器具有較多好處，比如，不需要對腳本解譯器的原始程式碼本身進行修改，侵入性較弱、耦合性較低。在這種存取請求處理單元結構的情況下，上述裝置實施例還可以包括註冊單元，用於在接收到網路伺服器轉發的存取動態資源的資源存取請求之前，向網路伺服器發送所述容器管理器程序的註冊請求，所述註冊請求包括所述容器管理器程序的程序標識以及虛擬主機的標識；所述存取請求接收單元 U82，還具體用於接收網路伺服器在建立所述容器管理器程序的程序標識與所述虛擬主機的標識之間的對應關係後發送的註冊成功消息，以便在接收到所述註冊成功消息後，接收網路伺服器根據所述對應關係轉發給所述容器管理器程序的存取動態資源的資源存取請求。

本領域內的技術人員應明白，本發明的實施例可提供為方法、裝置或電腦程式產品。因此，本發明可採用完全

硬體實施例、完全軟體實施例、或結合軟體和硬體方面的實施例的形式。而且，本發明可採用在一個或多個其中包含有電腦可用程式碼的電腦可用儲存介質（包括但不限於磁碟記憶體、CD-ROM、光學記憶體等）上實施的電腦程式產品的形式。

本發明是參照根據本發明實施例的方法、設備、裝置和電腦程式產品的流程圖和／或方框圖來描述的。應理解可由電腦程式指令實現流程圖和／或方框圖中的每一流程和／或方框、以及流程圖和／或方框圖中的流程和／或方框的結合。可提供這些電腦程式指令到通用電腦、專用電腦、嵌入式處理機或其他可程式設計資料處理設備的處理器以產生一個機器，使得透過電腦或其他可程式設計資料處理設備的處理器執行的指令產生用於實現在流程圖一個流程或多個流程和／或方框圖一個方框或多個方框中指定的功能的裝置。

這些電腦程式指令也可儲存在能引導電腦或其他可程式設計資料處理設備以特定方式工作的電腦可讀記憶體中，使得儲存在該電腦可讀記憶體中的指令產生包括指令裝置的製造品，該指令裝置實現在流程圖一個流程或多個流程和／或方框圖一個方框或多個方框中指定的功能。

這些電腦程式指令也可裝載到電腦或其他可程式設計資料處理設備上，使得在電腦或其他可程式設計設備上執行一系列操作步驟以產生電腦實現的處理，從而在電腦或其他可程式設計設備上執行的指令提供用於實現在流程圖

一個流程或多個流程和／或方框圖一個方框或多個方框中指定的功能的步驟。

在一個典型的配置中，計算設備包括一個或多個處理器（CPU）、輸入/輸出介面、網路介面和記憶體。記憶體可能包括電腦可讀介質中的非永久性記憶體，隨機存取記憶體（RAM）和/或非易失性記憶體等形式，如唯讀記憶體（ROM）或快閃記憶體（flash RAM）。記憶體是電腦可讀介質的示例。

電腦可讀介質包括永久性和非永久性、可移動和非可移動媒體可以由任何方法或技術來實現資訊儲存。資訊可以是電腦可讀指令、資料結構、程式的模組或其他資料。電腦的儲存介質的例子包括，但不限於相變記憶體（PRAM）、靜態隨機存取記憶體（SRAM）、動態隨機存取記憶體（DRAM）、其他類型的隨機存取記憶體（RAM）、唯讀記憶體（ROM）、電可擦除可程式設計唯讀記憶體（EEPROM）、快閃記憶體或其他記憶體技術、唯讀光碟唯讀記憶體（CD-ROM）、數位多功能光碟（DVD）或其他光學儲存、磁盒式磁帶，磁帶磁磁片儲存或其他磁性存放裝置或任何其他非傳輸介質，可用於儲存可以被計算設備存取的資訊。按照本文中的界定，電腦可讀介質不包括暫存電腦可讀媒體（transitory media），如調變的資料信號和載波。

還需要說明的是，術語“包括”、“包含”或者其任何其他變體意在涵蓋非排他性的包含，從而使得包括一系列要

素的過程、方法、商品或者設備不僅包括那些要素，而且還包括沒有明確列出的其他要素，或者是還包括為這種過程、方法、商品或者設備所固有的要素。在沒有更多限制的情況下，由語句“包括一個……”限定的要素，並不排除在包括所述要素的過程、方法、商品或者設備中還存在另外的相同要素。

本領域技術人員應明白，本發明的實施例可提供為方法、系統或電腦程式產品。因此，本發明可採用完全硬體實施例、完全軟體實施例或結合軟體和硬體方面的實施例的形式。而且，本發明可採用在一個或多個其中包含有電腦可用程式碼的電腦可用儲存介質（包括但不限於磁碟記憶體、CD-ROM、光學記憶體等）上實施的電腦程式產品的形式。

以上所述僅為本發明的實施例而已，並不用於限制本發明。對於本領域技術人員來說，本發明可以有各種更改和變化。凡在本發明的精神和原理之內所作的任何修改、等同替換、改進等，均應包含在本發明的申請專利範圍的範圍之內。

【符號說明】

U81：存取請求接收單元

U82：存取請求處理單元

U821：第二腳本解譯器程序創建子單元

U822：第二存取請求處理子單元

申請專利範圍

1.一種網路伺服器的虛擬主機隔離方法，包括：

確定存在於單一所述網路伺服器中的虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄，將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄；

在所述虛擬主機的根目錄下創建公共資源利用目錄，建立所述公共資源利用目錄與網路伺服器根目錄之下的公共資源分享目錄之間的映射關係，以形成虛擬主機用於運行程序的容器，透過所述容器實現虛擬主機之間的隔離。

2.根據申請專利範圍第 1 項所述的方法，其中，所述將與所述虛擬主機對應的目錄中的最上層目錄確定為所述虛擬主機的根目錄包括：

調用根目錄改變函數將與所述虛擬主機對應的目錄中的最上層目錄修改為所述虛擬主機的根目錄。

3.根據申請專利範圍第 1 項所述的方法，其中，所述建立所述公共資源利用目錄與網路伺服器的根目錄之下的公共資源分享目錄之間的映射關係包括：

調用目錄掛載函數將網路伺服器的根目錄之下的公共資源分享目錄以唯讀掛載的方式掛載到所述虛擬主機的公共資源利用目錄。

4.根據申請專利範圍第 1 至 3 項中任何一項所述的方法，其中，所述方法還包括：使利用網路伺服器和/或虛擬主機資源的一個或多個程序在虛擬主機的容器內運行，

在一個虛擬主機的容器內運行的程序與其他虛擬主機的容器內運行的程序具有不同的程序命名空間，在虛擬主機的容器內運行的程序在相同程序命名空間內具有不同的程序標識。

5.根據申請專利範圍第 4 項所述的方法，其中，所述使利用網路伺服器和/或虛擬主機資源的一個或多個程序在虛擬主機的容器內運行包括：

將待在虛擬主機內運行的程序的原始根目錄修改為虛擬主機的根目錄，以及將待在虛擬主機內運行的程序的程序標識修改為按照虛擬主機的程序命名空間命名的程序標識，以使程序在虛擬主機的容器內運行。

6.一種網路伺服器的虛擬主機，存在於單一所述網路伺服器中的所述虛擬主機包括用於運行程序的容器，所述容器的根目錄為所述虛擬主機的根目錄，所述虛擬主機的根目錄為所述虛擬主機在網路伺服器的根目錄之下與所述虛擬主機對應的目錄中的最上層目錄，在所述虛擬主機的根目錄下具有公共資源利用目錄，所述公共資源利用目錄與網路伺服器的根目錄之下的公共資源分享目錄之間具有映射關係。

7.根據申請專利範圍第 6 項所述的虛擬主機，其中，在所述容器內能運行利用網路伺服器和/或虛擬主機資源的一個或多個程序，在一個虛擬主機的容器內運行的程序與其他虛擬主機的容器內運行的程序具有不同的程序命名空間，在虛擬主機的容器內運行的程序在相同程序命名空

間內具有不同的程序標識。

8.根據申請專利範圍第 6 或 7 項所述的虛擬主機，其中，所述容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照當前虛擬主機的程序命名空間命名程序標識。

9.一種資源存取請求處理方法，包括：

接收網路伺服器轉發的存取動態資源的資源存取請求，所述資源存取請求包括待存取的動態資源所在的虛擬主機的標識；

在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結果返回給所述網路伺服器，以便網路伺服器返回給發送所述資源存取請求的請求端；

其中，所述虛擬主機存在於單一所述網路伺服器。

10.根據申請專利範圍第 9 項所述的方法，其中，所述在虛擬主機的容器內處理所述存取動態資源的資源存取請求包括：

將第一腳本解譯器程序的原始根目錄修改為所述虛擬主機的根目錄，以及將所述第一腳本解譯器程序的程序標識修改為按照所述虛擬主機的程序命名空間命名的程序標識，以使第一腳本解譯器程序在所述虛擬主機的容器內運行；

透過在虛擬主機的容器內運行的第一腳本解譯器程序處理所述存取動態資源的資源存取請求。

11.根據申請專利範圍第 9 項所述的方法，其中，所

述虛擬主機的容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照虛擬主機的程序命名空間命名程序標識，則所述在虛擬主機的容器內處理所述存取動態資源的資源存取請求包括：

在接收到網路伺服器轉發的存取動態資源的資源存取請求後，繼承所述容器管理器程序，以產生第二腳本解譯器程序；

透過在虛擬主機的容器內運行的第二腳本解譯器程序處理所述存取動態資源的資源存取請求。

12.根據申請專利範圍第 11 項所述的方法，其中，

在接收到網路伺服器轉發的存取動態資源的資源存取請求之前，向網路伺服器發送所述容器管理器程序的註冊請求，所述註冊請求包括所述容器管理器程序的程序標識以及虛擬主機的標識；

接收網路伺服器在建立所述容器管理器程序的程序標識與所述虛擬主機的標識之間的對應關係後發送的註冊成功消息，以便在接收到所述註冊成功消息後，接收網路伺服器根據所述對應關係轉發給所述容器管理器程序的存取動態資源的資源存取請求。

13.一種資源存取請求處理裝置，包括存取請求接收單元和存取請求處理單元，其中：

所述存取請求接收單元，用於接收網路伺服器轉發的存取動態資源的資源存取請求，所述存取動態資源的資源

存取請求包括待存取的動態資源所在的虛擬主機的標識；

所述存取請求處理單元，用於在虛擬主機的容器內處理所述存取動態資源的資源存取請求，將處理結果返回給所述網路伺服器，以便網路伺服器將其返回給發送所述資源存取請求的請求端；

其中，所述虛擬主機存在於單一所述網路伺服器。

14.根據申請專利範圍第 13 項所述的裝置，其中，所述存取請求處理單元包括第一腳本解譯器程序創建子單元和第一存取請求處理子單元，其中：

所述第一腳本解譯器程序創建子單元，用於將第一腳本解譯器的原始根目錄修改為所述虛擬主機的根目錄，以及將所述第一腳本解譯器程序的程序標識修改為按照所述虛擬主機的程序命名空間命名的程序標識，以使第一腳本解譯器程序在所述虛擬主機的容器內運行；

所述第一存取請求處理子單元，用於透過在虛擬主機的容器內運行的第一腳本解譯器程序處理所述存取動態資源的資源存取請求。

15.根據申請專利範圍第 13 項所述的裝置，其中，所述虛擬主機的容器內具有容器管理器程序，所述容器管理器程序的根目錄為虛擬主機的根目錄，所述容器管理器程序按照當前虛擬主機的程序命名空間命名程序標識，所述存取請求處理單元包括第二腳本解譯器程序創建子單元和第二存取請求處理子單元，其中：

所述第二腳本解譯器程序創建子單元，用於在接收到

網路伺服器轉發的存取動態資源的資源存取請求後，繼承所述容器管理器程序，以產生第二腳本解譯器程序；

所述第二存取請求處理子單元，用於透過在虛擬主機的容器內運行的第二腳本解譯器程序處理所述存取動態資源的資源存取請求。

圖式

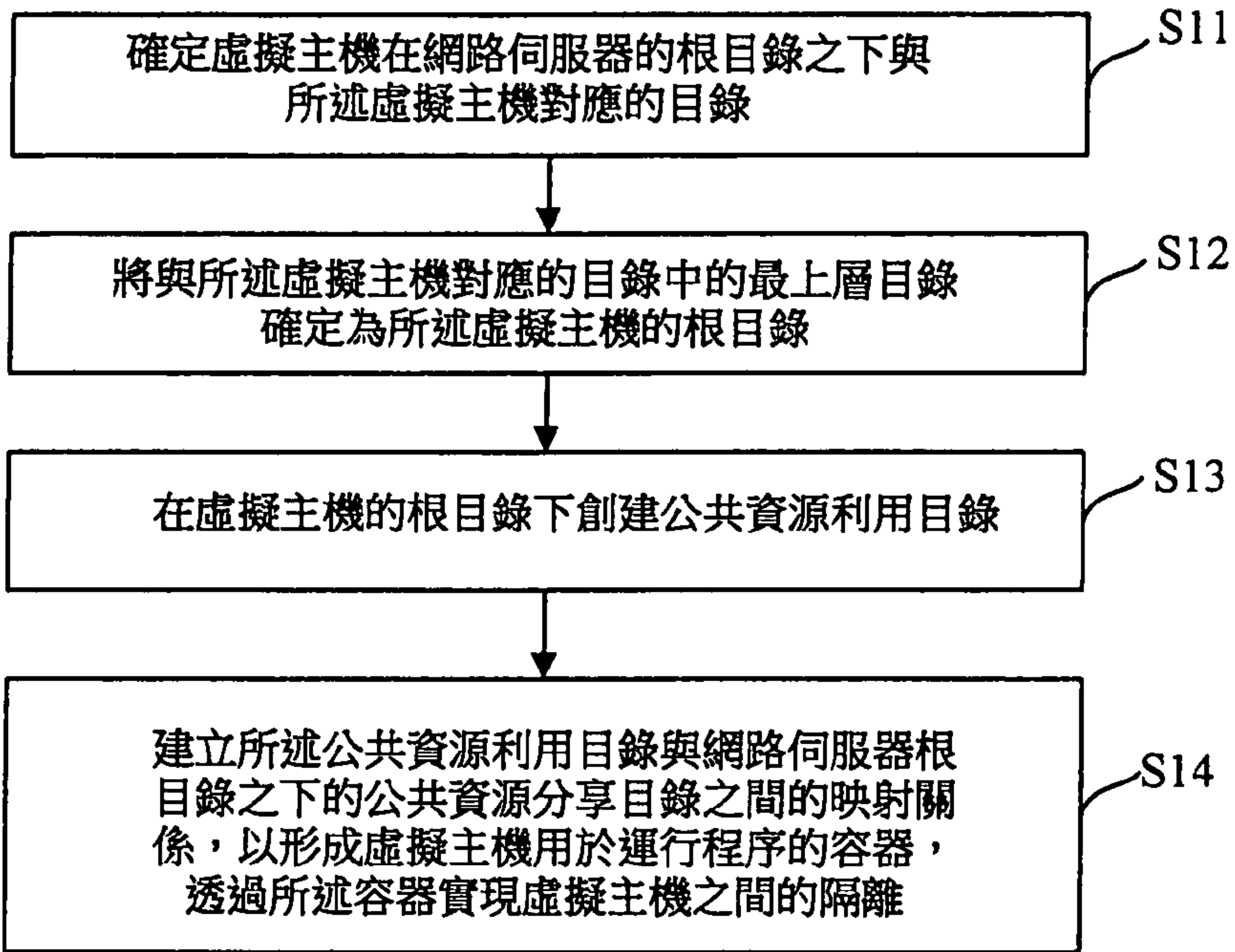


圖 1

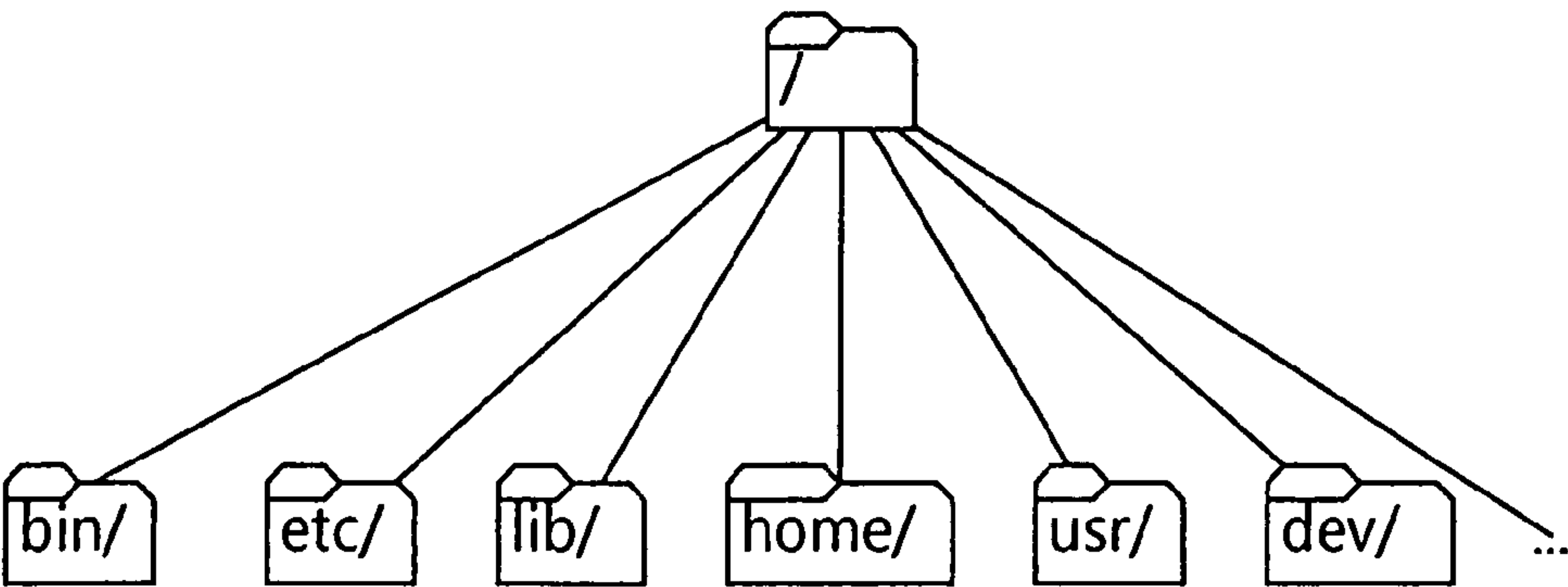


圖 2(a)

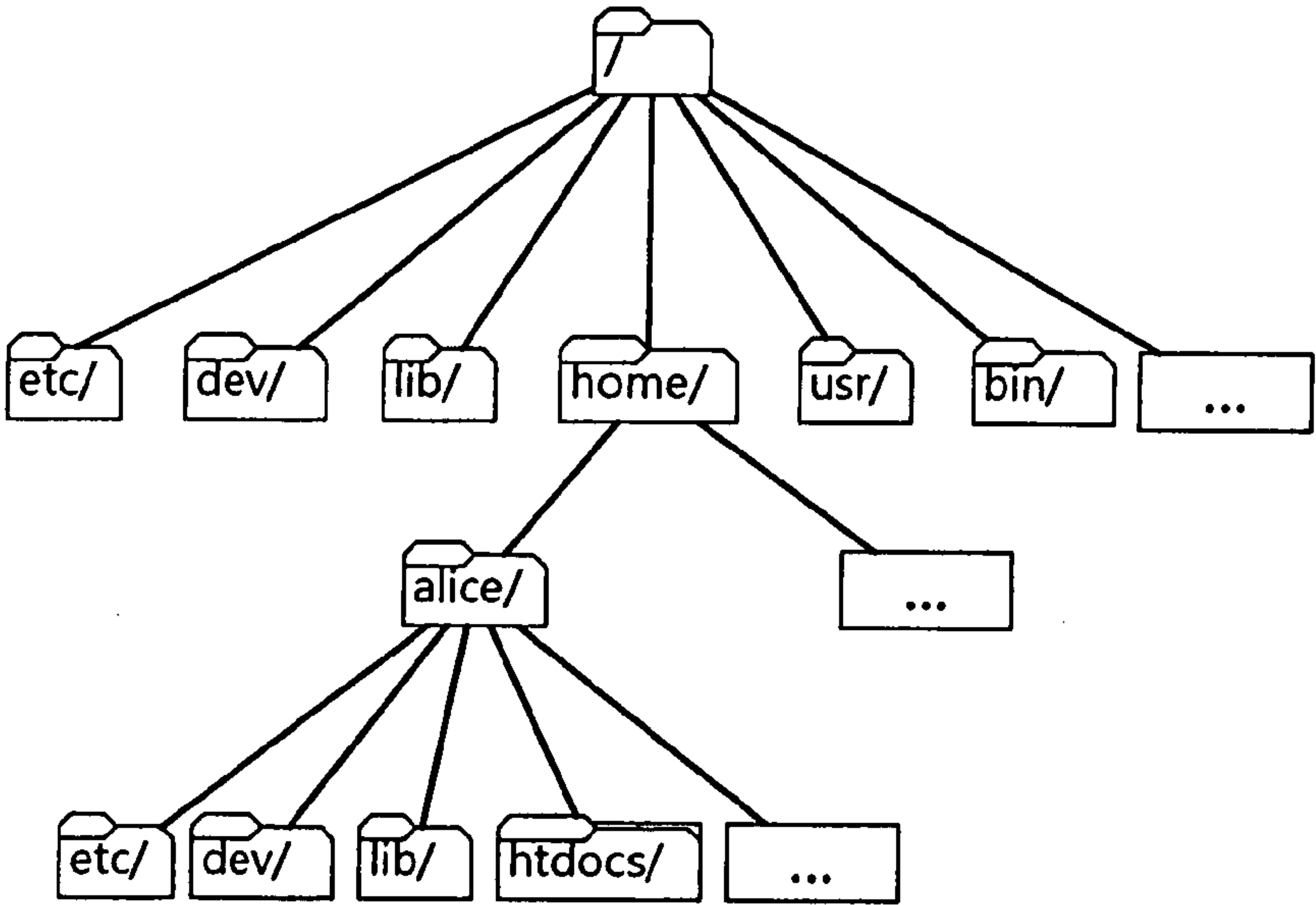


圖 2(b)

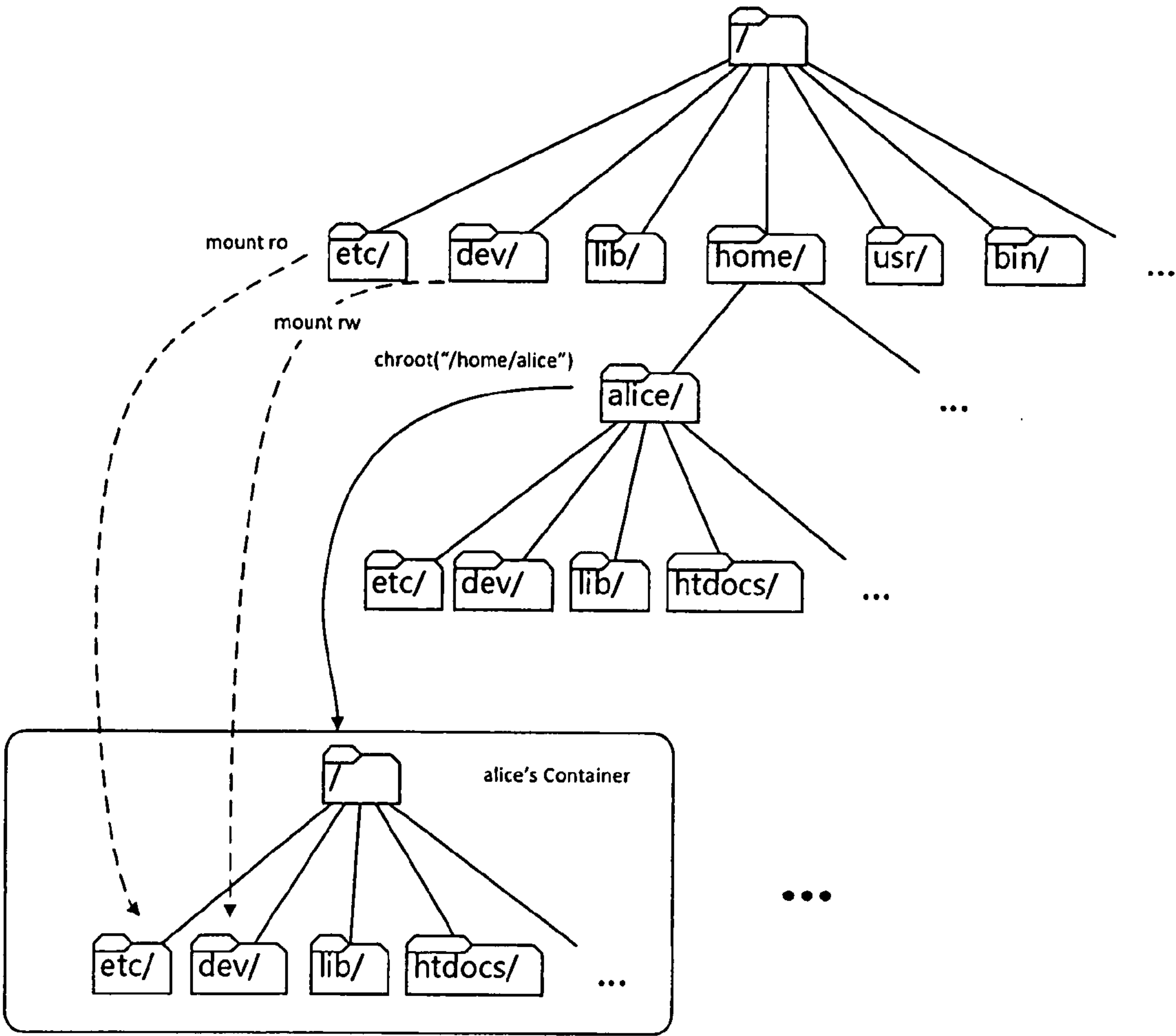


圖 3

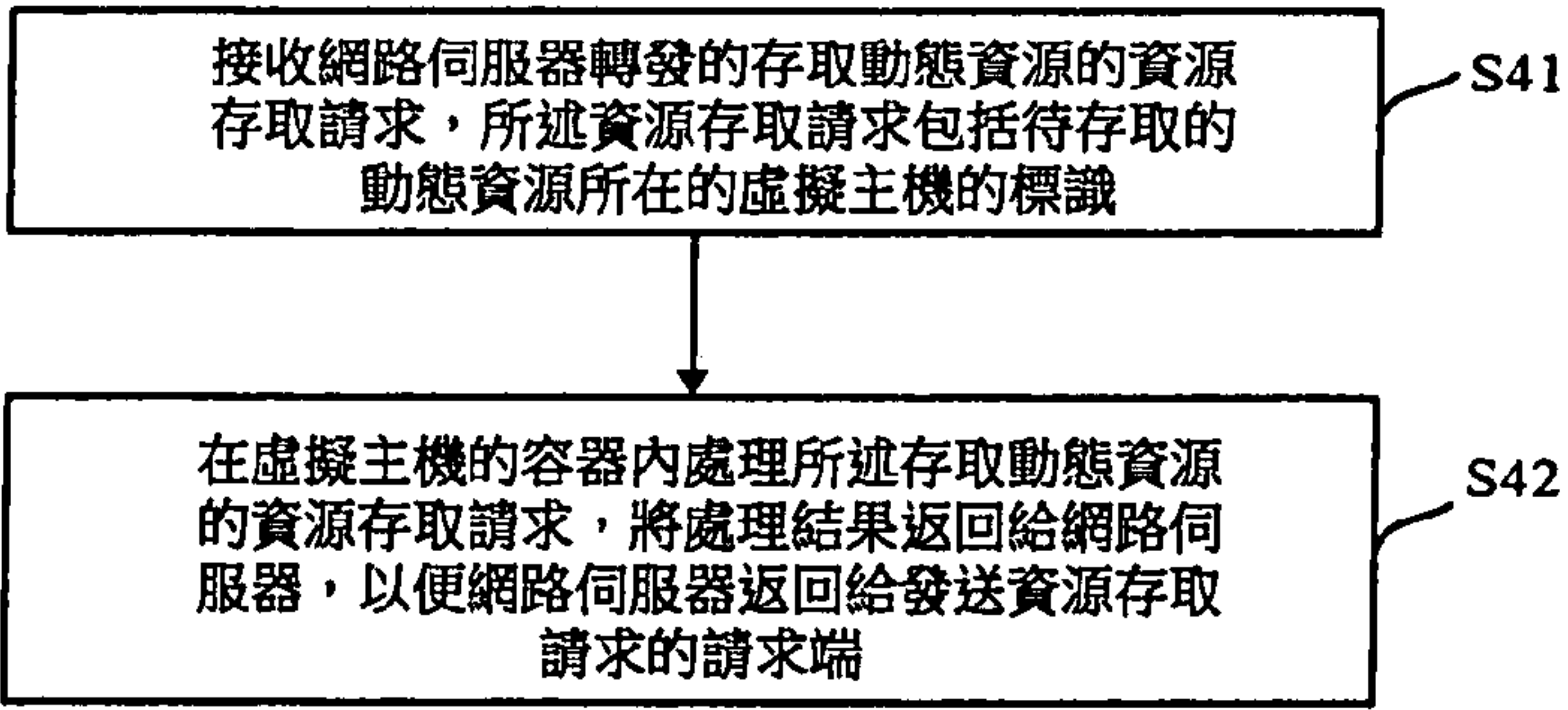


圖 4

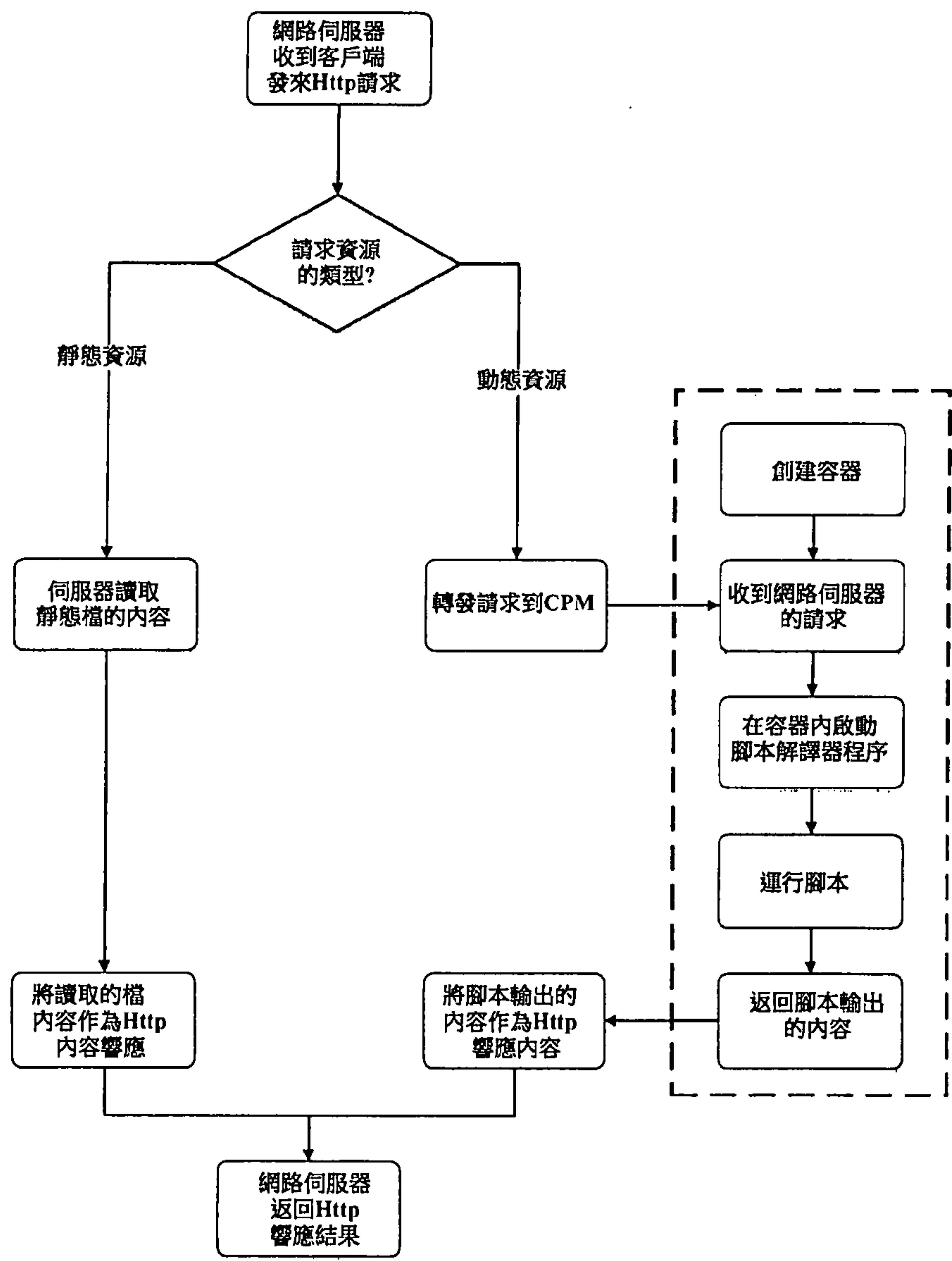


圖 5

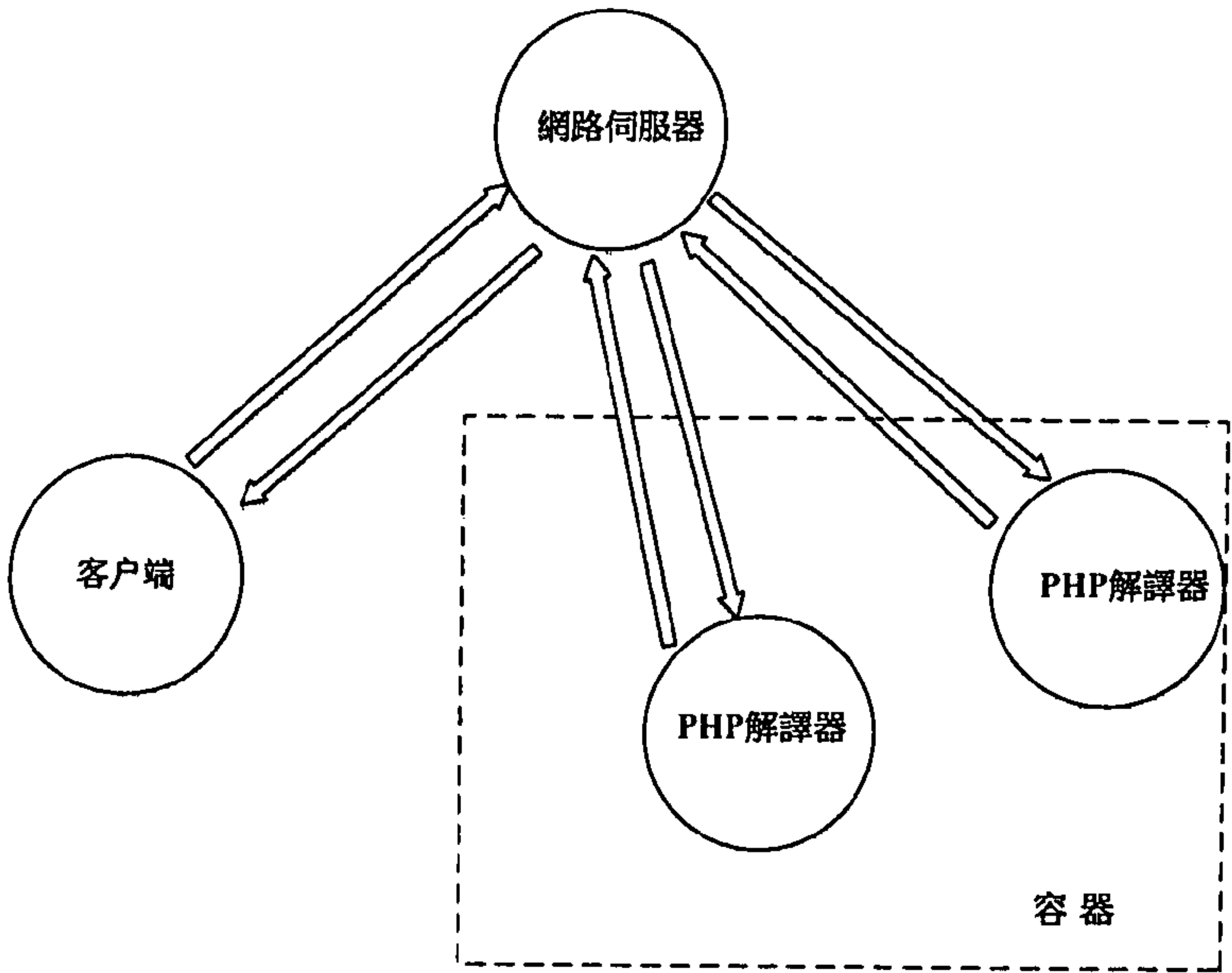


圖 6(a)

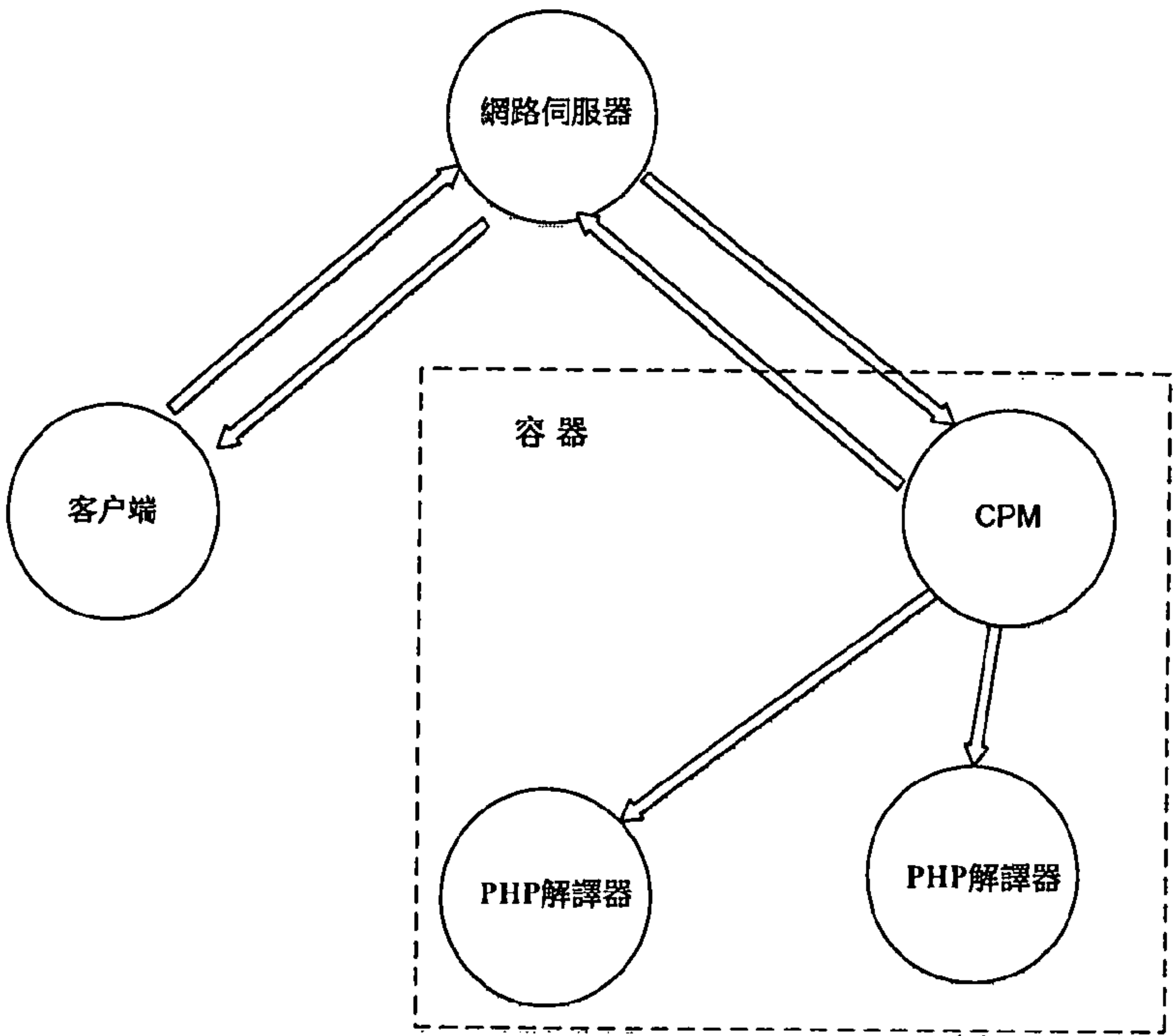


圖 6(b)

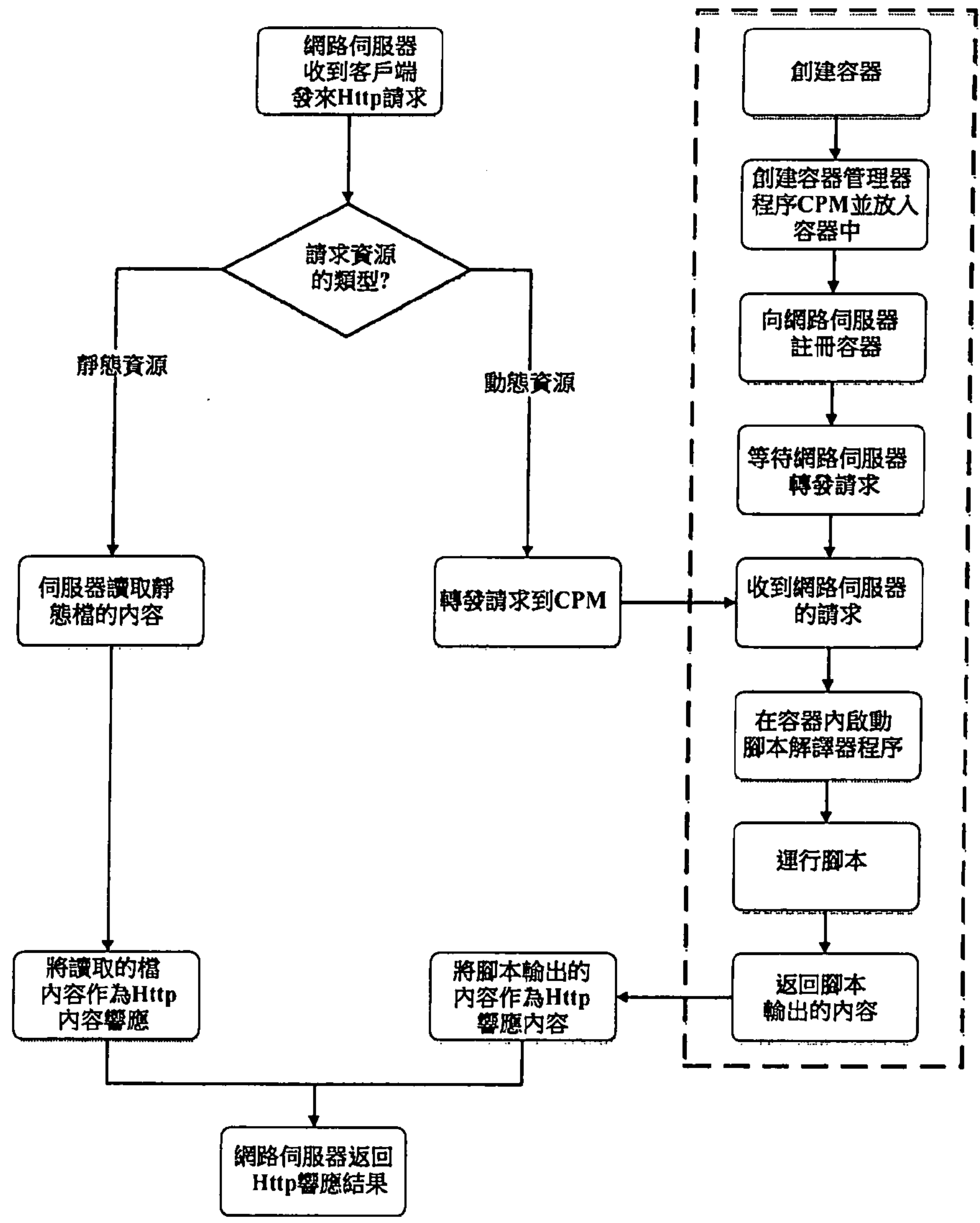


圖 7(a)

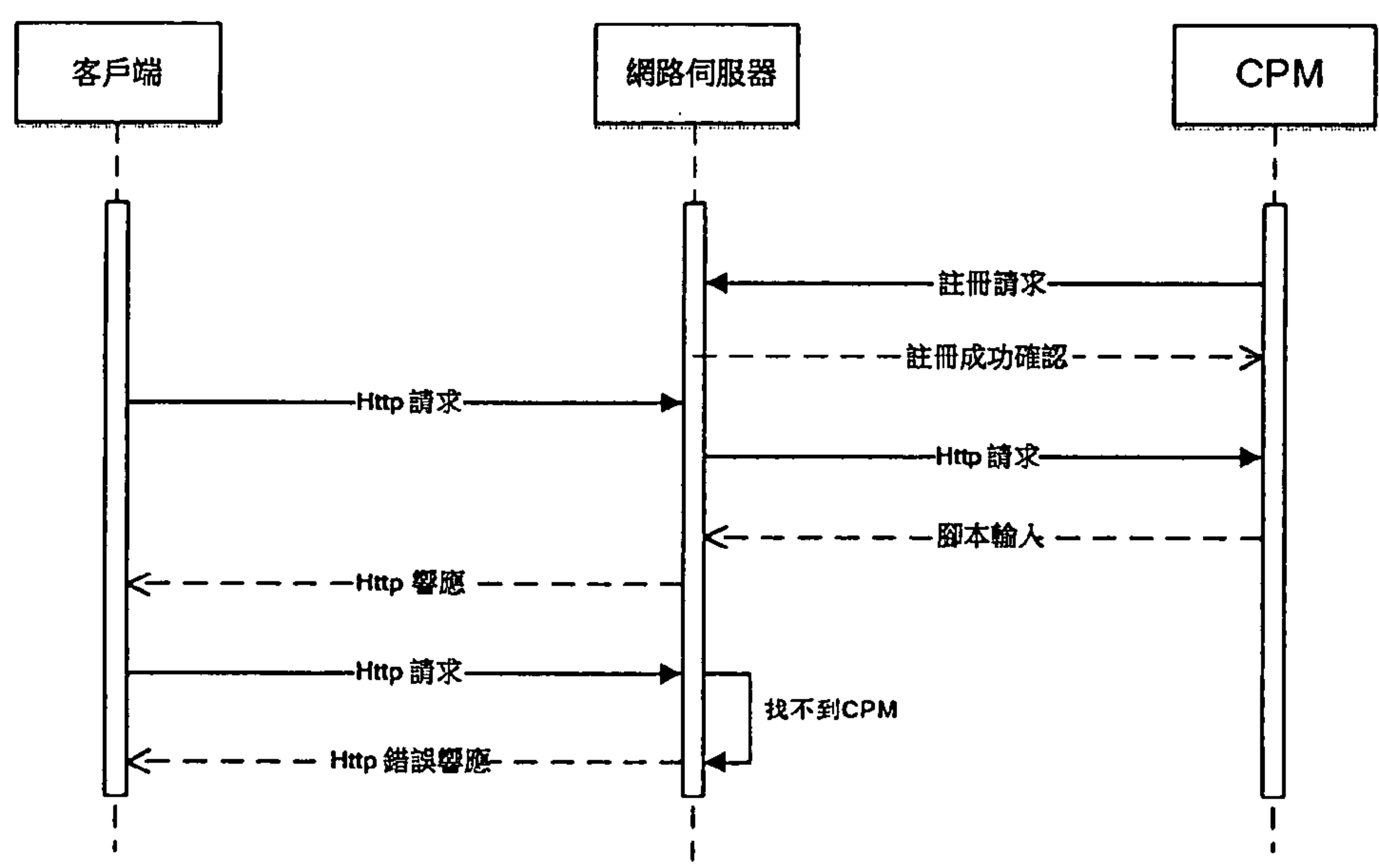


圖 7(b)

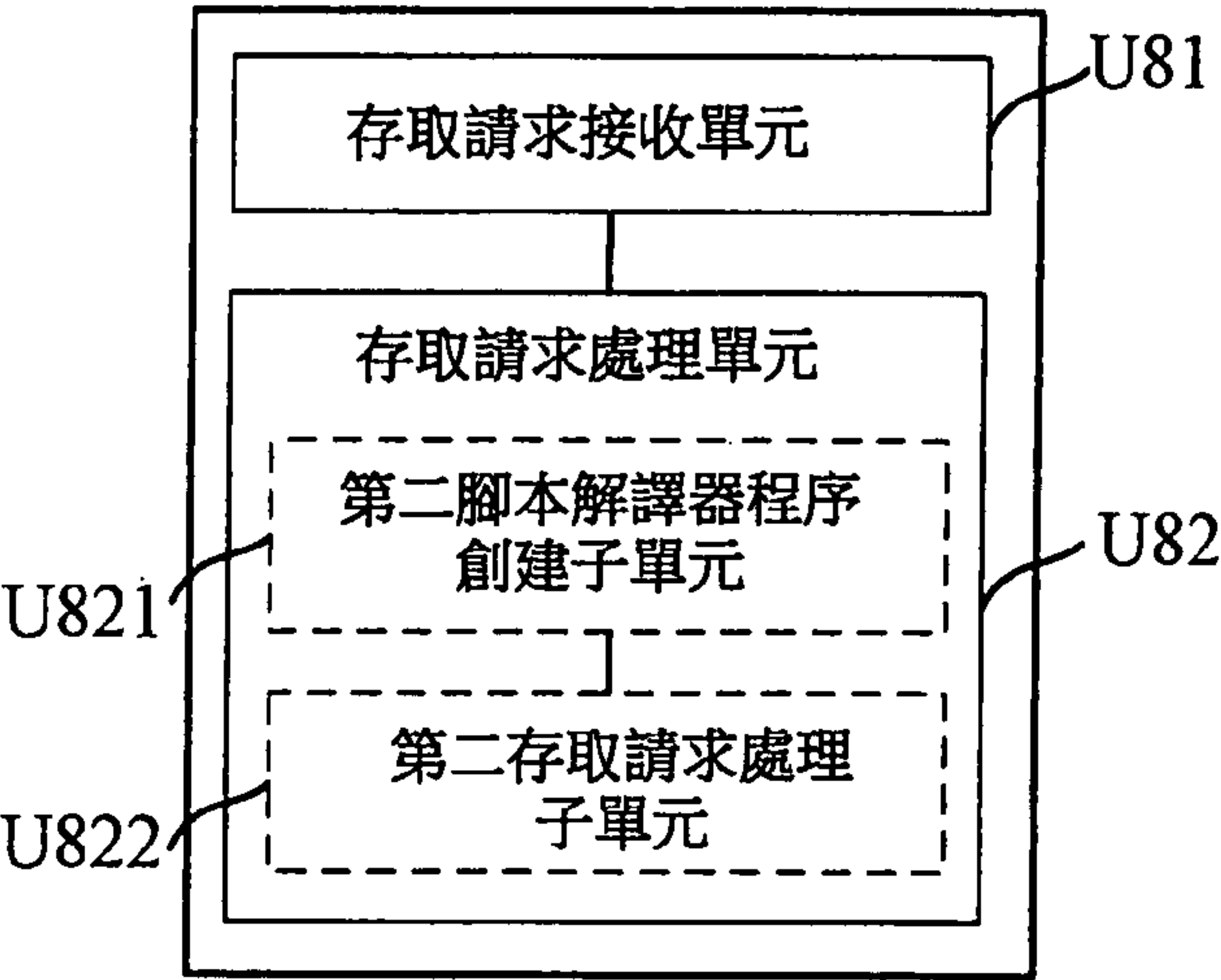


圖 8