

發明專利說明書

(本申請書格式、順序，請勿任意更動)

【發明名稱】(中文/英文)

第三方應用的執行方法及系統

【技術領域】

本發明係關於第三方應用開發技術領域，特別是關於第三方應用的執行方法及系統。

【先前技術】

平台型網際網路應用（例如，某電子商務交易平台）為了給使用者（例如，電子商務交易平台中的賣方使用者）提供更細分、更垂直的服務，一般需要引入第三方開發者來完成。例如，第三方開發者可以對點擊量、跨店鋪點擊、訂單流轉量、甚至相關即時通信工具中的聊天記錄等資訊，進行收集和分析，最終提供給賣方用戶直觀的建議。也就是說，對於某網際網路應用的用戶而言，在該網際網路應用的網頁中瀏覽到的一些資料分析結果等資訊，通常是由第三方伺服器提供的。

現有技術中，網際網路應用需要通過開放平台開放相關資料給第三方開發者，第三方開發者將獲取到的資料存放入自有的儲存系統，以使用這些資料來完成服務的開發以及後續的運行。

但是，這種實現方式至少具有以下缺點：首先，資料

被第三方開發者取走後，存在較大的資料洩露、濫用和不當使用等風險，因為無法對被取走的資料進行安全控制。其次，正是由於這樣的安全缺陷，使得網際網路應用一般不能開放高價值（隱私、安全、競爭等風險）的資料給第三方開發者，這就使得第三方開發者能夠實現的功能也很有限。另外，第三方開發者自身的軟硬體處理能力比較有限，無法實現大資料量的計算處理。

例如，某第三方開發者想要給某電子商務交易平台中的賣方使用者提供商品補貨預測服務，則現有技術中，第三方開發者就需要從開放平台將所需的資料取走，放入第三方開發者的環境中進行處理和計算。而補貨預測需要對商品的銷量進行預測，但由於資料安全的問題，開放平台不能直接提供預測模型需要的資料。例如行業商品銷量（涉及到行業資料隱私）、同類商品成交的人群屬性（涉及到行業和消費者資料隱私）等等。而對於賣方用戶的補貨管道、補貨策略等，這些也是比較核心的競爭力，直接提供給第三方開發者也存在一定信任問題。另外，預測模型需要考慮行業使用者行為、商品交易行為等，資料量很大，第三方開發者可能不具備必要的軟硬體開發環境等。總之，以上各種因素都可能導致該第三方伺服器不能很好地為交易平台中的使用者提供相關服務。

【發明內容】

本發明提供了第三方應用的執行方法及系統，解決了

資料安全風險和開放高價值資料的矛盾。

本發明提供了如下方案：

一種第三方應用的執行方法，包括：

接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給第三方開發者，以供第三方開發者基於所述元資料描述資訊進行第三方應用的開發；

接收第三方開發者提交的第三方應用的程式資訊，並將所述第三方應用部署到資料容器環境中，其中，所述資料容器環境中還保存有第三方應用在執行過程中所需的具體資料；

在所述資料容器環境中執行所述第三方應用。

一種第三方應用的執行系統，包括：

元資料描述資訊發送單元，用於接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給第三方開發者，以供第三方開發者基於所述元資料描述資訊進行第三方應用的開發；

應用部署單元，用於接收第三方開發者提交的第三方應用的程式資訊，並將所述第三方應用部署到資料容器環境中，其中，所述資料容器環境中還保存有第三方應用在執行過程中所需的具體資料；

應用執行單元，用於在所述資料容器環境中執行所述第三方應用。

根據本發明提供的具體實施例，本發明公開了以下技

術效果：

通過本發明實施例，為第三方開發者提供了資料容器系統，將計算所需的資料按類別等存放在系統內部的資料容器中，而不是發送給第三方開發者，第三方開發者在開發完成第三方應用的計算邏輯或演算法邏輯之後，將第三方應用也部署到該資料容器中，這樣資料以及對資料的操作過程相當於都是在資料容器內部完成的，第三方開發者不用看到具體的資料內容，只要提供應用的計算邏輯或者演算法就好，從而解決了資料安全風險和開放高價值資料的矛盾。

當然，實施本發明的任一產品並不一定需要同時達到以上所述的所有優點。

【圖式簡單說明】

為了更清楚地說明本發明實施例或現有技術中的技術方案，下面將對實施例中所需要使用的附圖作簡單地介紹，顯而易見地，下面描述中的附圖僅僅是本發明的一些實施例，對於本領域普通技術人員來講，在不付出創造性勞動的前提下，還可以根據這些附圖獲得其他的附圖。

圖 1 是本發明實施例提供的方法的流程圖；

圖 2 是本發明實施例提供的系統的示意圖。

【實施方式】

下面將結合本發明實施例中的附圖，對本發明實施例

中的技術方案進行清楚、完整地描述，顯然，所描述的實施例僅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所獲得的所有其他實施例，都屬於本發明保護的範圍。

在本發明實施例中，為了保證資料的安全性，並且使得第三方開發者能夠提供更豐富的服務，提供了資料容器系統，將計算所需的資料按類別等存放在系統內部的資料容器中，而不是發送給第三方開發者，第三方開發者在開發完成第三方應用的計算邏輯或演算法邏輯之後，將第三方應用也部署到該資料容器中，這樣資料以及對資料的操作過程相當於都是在資料容器內部完成的，第三方開發者不用看到具體的資料內容，只要提供應用的計算邏輯或者演算法就好，從而解決了資料安全風險和開放高價值資料的矛盾。下面對具體的實現過程進行詳細地介紹。

首先需要說明的是，由於並不會直接將資料內容提供給第三方開發者，因此，在整個實現過程中，應用平台在第三方伺服器進行服務開發的過程也要提供相應的支援。開發完成之後，應用平台還需要進行後續的部署等方面的支持。參見圖 1，本發明實施例提供的第三方應用的執行方法可以包括以下步驟：

S101：接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給第三方開發者，以供第三方開發者基於所述元資料描述資訊進行第三方應用的開發；

為第三方開發者提供用於後台開發的介面，第三方開發者可以在其中編寫代碼、調試等等。例如，一般可以直接以網頁的形式，也即第三方開發者直接登錄應用平台的系統，就可以進入後台開發頁面進行具體的開發操作。

其中，由於資料容器中的資料眾多，而第三方開發者一般只會提供某一方面的服務，因此，一般只會使用到其中的一部分資料，因此，可以預先對資料容器中的資料劃分成多個類別，每個類別的資料可以對應一個資料主題。例如，在某電子商務交易平台中，資料主題可能包括發貨、收貨、訂單管理、客服等主題。第三方開發者進入開發平台後，可以首先將各種可選的資料主題顯示給第三方開發者，第三方伺服器可以根據自己需要開發的服務，選擇對應的資料主題。例如，某第三方開發者想要開發訂單管理方面的服務，則可以選擇訂單管理這一資料主題，等等。

在第三方開發者選擇了某一資料主題之後，可以將該資料主題中可能會使用到的元資料描述資訊提供給第三方開發者，以供第三方開發者進行應用的開發。需要說明的是，開放平台中的資料一般以資料表（關係型數據庫中的表）的格式進行保存，因此，具體在向第三方開發者提供元資料描述資訊時，可以包括資料表的名稱，以及資料表中的欄位資訊。也就是說，可以使第三方開發者知曉資料表中有哪些欄位，例如，對於某保存有使用者資訊的資料表 A 而言，其中的欄位包括年齡、性別等欄位，此時，就

可以將該資料表 A 的名稱，以及年齡、性別等欄位名稱提供給第三方開發者。這樣，第三方開發者就可以基於這些欄位名稱，結合資料表的基本操作方法，來開發第三方應用中具體的邏輯。其中，對資料表的基本操作可以是支援多種程式設計語言，例如，JAVA、SQL 等用於存取資料以及查詢、更新和管理關聯式資料庫系統的語言，來實現對資料表中各欄位資料的操作。

也就是說，在本發明實施例中，不需要將資料表中的具體資料提供給第三方開發者，而只需要將資料表中有哪些欄位等資訊提供給第三方開發者。當然，在實際應用中，為了使得第三方開發者在開發出具體的應用之後，對應用的計算邏輯等進行驗證操作，開放平台可以為第三方開發者提供一些樣本資料。這種樣本資料可以是一些示例性的資料，例如，可以是將使用者幾年前的行為資料拼接在一起組合而成。由於已經是幾年前的資料，並且是由一些資料片段組合而成的，因此，一般不會存在資訊洩漏等問題。

對於一些大型的第三方應用而言，需要的代碼量往往非常大，編寫代碼的過程比較複雜，且容易出錯。在本發明實施例中，為了簡化開發過程，允許第三方開發者將一個應用拆分成多個子應用，在開發環境中分別開發各個子應用的代碼，最後再由開放平台根據各個子應用之間的依賴關係等，將各個子應用關聯起來，共同實現對應的第三方應用。

另外，還可以將第三方應用劃分為不同的類型，分別為各種類型的應用提供對應的開發介面，並且分別提供各種類型的開發入口。如果第三方開發者想要開發某類型的應用，則從該類型的開發入口進入，開放平台在通過該開發入口接收到用戶的請求後，就可以顯示出對應的開發介面，第三方開發者就可以在該介面中進行編寫代碼、調試等工作。例如，可以將第三方應用分為計算類應用以及演算法類應用。同時，開放平台可以根據具體進入的介面，將對應開發的第三方應用所屬的類型進行標識，以便後續在進行調度時，發送給相匹配的資料容器。

需要說明的是，第三方開發者開發的同一個第三方應用可能由多個子應用組成，各子應用可以是不同類型的應用，並且，相互之間也可以進行資料傳遞。

另外需要說明的是，在實際應用中，在接收到某第三方開發者發送的資料申請請求時，還可以對這種資料申請進行審核，例如，包括對第三方開發者的資質等進行的審核。再者，如果涉及到使用者資料，還可以建立使用者授權。也即，詢問用戶是否允許第三方開發者使用其資料，如果使用者允許，再向第三方開發者返回相應的元資料描述資訊、樣本資料等。否則，如果沒有獲取到用戶授權，則不會向第三方開發者返回相應的資訊。

S102：接收第三方開發者提交的第三方應用的程式資訊，並將所述第三方應用部署到資料容器環境中，其中，所述資料容器環境中還保存有第三方應用在執行過程中所

需的具體資料；

第三方開發者在完成了某第三方應用的開發之後，就可以將其提交到開發平台，相應的，開發平台就可以將第三方應用的邏輯部署到開放平台內部的資料容器環境中。由於該資料容器環境中保存有第三方應用在執行過程中所需的具體資料，因此，就可以在該資料容器環境中執行第三方應用。這樣，具體的資料處理的過程都是在資料容器內部完成的，不需要將待處理的資料公開給第三方開發者。

在實際應用中，在接收到第三方開發者提交的第三方應用之後，還可以對應用進行安全審核。例如，判斷是否存在敏感性資料，是否存在對資料的敏感操作等，進而，如果存在對敏感性資料的操作，或者存在敏感操作，則可以在資料輸出時進行攔截，進一步提高資料的安全性。具體在進行安全審核時可以有多種方式，例如，在一種簡單的方式下，可以直接利用關鍵字等進行過濾；或者，判斷第三方應用中是否存在關於某些特殊商品物件的交易資訊；或者，對於可以量化的資料，還可以判斷第三方應用使用的某行業資料的資料量所占總數據量的百分比，如果該百分比達到某數值，則可能會由此推導出行業的其他資訊，導致該行業資料的洩漏，因此認為存在安全性問題。再或者，還可以在第三方應用中包含一些使用者 ID 等資訊的情況下，判斷是否能夠反推獲知用戶的真實身份等敏感資訊，如果是，則也認為存在安全性問題，等等。總

之，在進行安全性審核時，可以預先設置多個判斷條件，對於某個第三方應用而言，可以利用這些判斷條件進行判斷，如果命中其中某個或者某幾個條件，則認為存在安全性問題。

如前文所述，不同的第三方應用之間可能存在一定的依賴關係。例如，某應用需要依賴于另一應用的執行結果，則該應用需要在另一應用執行完成結束之後執行。另外，還有一些第三方應用需要在特定的應用場景下執行，等等。因此，在向資料容器提交部署之前，可以根據依賴關係、應用場景等資訊對提交上來的各個第三方應用進行調度，將當前需要執行的第三方應用提交到資料容器。

向資料容器提交需要執行的應用的同時，還可以提交一些控制命令，例如，該應用的優先順序、資源配置情況（例如，分配多少的 CPU、多少的記憶體給該應用）等，以便資料容器按照這些控制命令來執行。其中，關於資源配置情況，可以預估資料的輸入輸出量，並根據一定的演算法計算出需要佔用的資源，當然實際應用中，還可以根據第三方開發者訂購的資源數量等資訊來綜合確定。

S103：在所述資料容器環境中執行所述第三方應用。

在將第三方應用部署到資料容器之後，就可以在資料容器中執行第三方應用。具體在執行時，可以根據第三方應用任務的性質，進行同步或者非同步的處理。相應的，資料容器可以分為「分散式處理容器」或「同步處理容器」。其中，前者相當於是將任務分配到多個容器分別執

行，最後再進行匯總，一般用於處理對時間要求不高的任務；後者就是接收到執行任務後馬上進行處理，一般用於處理對時間要求比較高的任務。因此，前述任務的性質就可以是指對時間要求的高低。其中，關於任務的性質可以由第三方開發者在開發時進行指定。

在第三方應用執行完成之後，就可以將執行結果提供給用戶使用。其中，具體在向用戶提供執行結果時，可以有多種方式，其中一種方式是：可以由第三方開發者將執行結果取走，然後在其介面中展現執行結果。為了使得第三方開發者能夠獲取到執行結果，可以如下進行：對於第三方開發者開發完成的第三方應用代碼，可以根據輸出資料的標識，封裝成介面（一般是一個 URL），資料容器完成執行之後，可以記錄執行結果與該 URL 的對應關係。第三方開發者請求執行結果時，可以根據該 URL 返回執行結果。具體實現時，可以有多種方式。例如，其中一種方式可以是，向各個第三方開發者提供統一的獲取執行結果的入口，也即，每個第三方開發者在每次請求執行結果時，可向同一個 URL 發送請求，但是，同時要攜帶上第三方開發者的 ID 等標識資訊，以及此次請求的輸出資料的標識，開發平台在接收到該請求之後，就可以根據攜帶的 ID 以及輸出資料的標識確定出所需的執行結果實際對應的 URL，然後向第三方開發者返回執行結果即可。或者，在另一種實現方式下，也可以直接將執行結果實際對應的 URL 提供給第三方開發者，這樣，第三方開發者就

可以根據實際對應的 URL 請求獲取對應的執行結果。

另外，在實際應用中，還可能存在以下情況：對於在資料容器中執行得到的執行結果仍然存在一些敏感性資料，此時，如果直接由第三方開發者將執行結果取走，則仍然有可能造成對使用者資料的洩漏等危險。例如，某第三方應用用於篩選出一些使用者進行某種資訊的推廣，則該第三方應用的執行結果中可能會包含使用者的聯繫方式等資訊，如果將這些資訊提供給第三方開發者顯然會存在風險。因此，對於這種情況，在具體實現時，還可以要求第三方開發者將其前台介面展現邏輯也部署到資料容器中，這樣，在獲取到執行結果之後，就可以直接根據資料容器中的前台介面展現邏輯，將執行結果直接提供給用戶。

為了便於理解，下面通過一個實際的例子進行詳細的介紹。

具體實現時，可以包括以下步驟：

步驟 1：第三方開發者登錄到開放平台的後台；此時，介面中可以展現出各個可選擇的資料主題；

步驟 2：第三方開發者申請某資料主題；

步驟 3：開放平台對第三方開發者的申請進行審核，主要用於確定該第三方開發者是否具有申請該資料主題的資質；

步驟 4：判斷資料申請審核是否通過；如果未通過，則返回失敗等通知消息；如果通過，則進入步驟 5；

步驟 5：第三方開發者申請樣本資料；

步驟 6：開放平台向使用者發送授權審核消息；

步驟 7：判斷授權審核是否通過，如果未通過，則返回失敗等通知消息；如果通過，則進入步驟 8；

步驟 8：第三方開發者進入開發介面進行第三方應用的開發；

步驟 9：開發完畢之後，將第三方應用提交到開放平台進行審核；

步驟 10：開放平台對第三方應用進行資料安全審核；

步驟 11：判斷資料安全審核是否通過，如果未通過，則可以返回審核失敗等通知消息，如果通過，則繼續進入步驟 12；

步驟 12：第三方開發者將第三方應用提交到開放平台進行部署；

步驟 13：開放平台將第三方應用部署到資料容器進行執行；

步驟 14：將執行結果提供給用戶進行使用。

為了更好的理解本發明實施例的技術方案，下面將開發平台劃分為多個模組，分別介紹各個模組的功能以及模組之間的關聯關係。具體的，可以包括以下模組：

登錄驗證模組：第三方開發者使用預先註冊的帳戶資訊登錄到開放平台後，開放平台可以對第三方開發者的身份進行認證，完成身份認證後，將授予第三方開發者權

杖，第三方開發者可以帶上該權杖進入系統；

授權驗證模組：開放平台還可以對第三方開發者擁有的資料許可權和操作許可權進行認證，完成授權認證後，將授予第三方開發者對應的資料許可權和應用開發許可權；

資料應用開發模組：集成資料應用開發環境，為第三方開發者提供開發介面，可以在其中編寫代碼、調試等等。具體實現時，資料應用開發模組可以由兩部分組成，分別為通用計算類資料應用開發子模組和演算法類資料應用開發子模組，第三方開發者可以根據待開發的副程式所屬的類別選擇進入對應的子模組。在開發介面中，可以提供進入兩個子模組的入口，進入之後，兩個子模組具有各自的開發介面。如果某第三方開發者想要開發計算類的副程式，則可以從通用計算類資料應用開發子模組的入口進入，進而打開通用計算類資料應用開發子模組對應的開發介面；類似的，第三方開發者想要開發演算法類的副程式，則可以從通用演算法類資料應用開發子模組的入口進入，進而打開演算法類資料應用開發子模組對應的開發介面。

其中，通用計算類資料應用開發模組的主要功能可以包括：

a) 應用專案管理：包括對應用項目的增刪改查、分享、授權、協作等；

b) 代碼開發環境介面：

- i .代碼開發編輯器；
 - ii .代碼高亮、語法檢查等效率輔助工具；
 - iii .元資料管理：元資料詳細描述（包括表標識、欄位名等）、資料依賴、交互管理；
- c) 代碼沙箱環境管理介面：
 - i .代碼模擬運行（可以在開發完代碼後，進行測試，對計算邏輯進行驗證）
 - ii .資料集創建
- d) 代碼管理：代碼版本管理、目錄管理。當代碼開發完成之後，開放平台可以將代碼保存到代碼版本管理中，後續如果需要對代碼進行修改，則可以生成新的版本。在保存一份代碼時，開放平台可以將其封裝成對應的套裝程式，對於同一應用中的不同副程式對應的套裝程式而言，還可以標識各套裝程式的應用場景需求、啟動條件（例如，需要在某時間點啟動，或者需要在某指定程式運行完畢之後啟動，等等），以及各套裝程式之間的依賴關係等資訊。其中，關於套裝程式的啟動條件可以由第三方開發者指定的，而套裝程式之間的依賴關係，可以由第三方開發者指定，也可以由開放平台自動獲取。例如，開放平台通過對各套裝程式的代碼進行分析得知，套裝程式 1 的邏輯是對表 a 進行處理生成表 b、c，而套裝程式 2 的邏輯是對表 b 處理生成表 d，因此，可以判斷出套裝程式 2 依賴套裝程式 1，進而就可以自動標識該依賴關係。
- e) 應用管理：應用發佈、應用下架、應用狀態查

詢。

f) 費用管理：交費、續費、結算、清算、餘額管理。

演算法類資料應用開發子模組的主要功能可以包括：

a) 演算法模型管理：增刪改查、分享、授權、協作。

b) 基礎模型庫。

c) 資料特徵分析環境：統計分佈驗證、特徵提取、驗證。

d) 模型訓練管理：模型訓練、結果驗證。

沙箱資料容器環境模組，提供代碼、資料集驗證功能的計算環境，包括：

a) 代碼運行環境：

i .分散式代碼運行環境（Map/Reduce 程式、SQL 程式）

ii .演算法運行環境

b) 資料環境：提供代碼運行的資料支撐環境，必要的資料庫，資料集，資料集創建工具等，這些資料可以是開放平台提供的樣本資料，例如，將使用者幾年前的產生的資料拼接在一起等等，生成具有相同的資料結構，但並不真是的資料，這種資料提供給第三方開發者，用於對其開發的代碼進行驗證。

任務工作流調度模組：負責將第三方開發者提交的應用根據任務依賴關係、應用場景需求執行調度，並將需要

執行的應用提交給資料容器環境（資料容器通過預置的介面監聽任務工作流調度模組的提交請求，因此，當任務工作流調度模組發現有需要執行的應用時，就可以通過資料容器的介面進行發送）。其中，如果開發模組包括計算類資料應用開發子模組以及演算法類資料應用開發子模組，則在調度時，可以根據開發過程中標識出的計算類應用或者演算法類應用，將需要執行的應用發送到「通用計算容器環境」或者「演算法容器環境」執行。另外，任務工作流調度模組在向資料容器提交需要執行的應用的同時，還可以提交一些控制命令，以便資料容器按照這些控制命令來執行。

通用計算容器環境：

a) 負責接收任務工作流調度模組提交的通用計算任務，將計算任務根據性質不同提交給對應的「分散式處理容器」或「同步處理容器」。

b) 對任務進行監控、跟蹤、資料獲取

c) 和「資料查詢中間層」打通，完成資料查詢功能

演算法容器環境：

a) 負責接收任務工作流調度模組提交的演算法任務，將演算法任務根據性質不同提交給對應的：「分散式處理容器」或「同步處理容器」。

b) 「基礎演算法包」子模組。第三方開發者在演算法類資料應用開發子模組中開發時，可以根據自己的需求編寫演算法代碼，或者，開放平台也可以為第三方開發者

提供一些基礎演算法，這樣，第三方開發者只要在基礎演算法的基礎上進行選擇需要的特徵、調整演算法的參數等操作即可，以此提高開發效率。相應的，在演算法容器環境中，還可以提供基礎演算法可執行調用模組，其中，基礎演算法可以包括：

- i .分類演算法
- ii .聚類演算法
- iii .回歸演算法

資料安全審核引擎：當第三方開發者完成代碼的開發之後，在部署到資料容器之前，還可以進行安全審核。

與本發明實施例提供的第三方應用的執行方法相對應，本發明實施例還提供了一種第三方應用的執行系統，參見圖 2，該系統可以包括：

元資料描述資訊發送單元 201，用於接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給第三方開發者，以供第三方開發者基於所述元資料描述資訊進行第三方應用的開發；

應用部署單元 202，用於接收第三方開發者提交的第三方應用的程式資訊，並將所述第三方應用部署到資料容器環境中，其中，所述資料容器環境中還保存有第三方應用在執行過程中所需的具體資料；

應用執行單元 203，用於在所述資料容器環境中執行所述第三方應用。

具體實現時，該系統還可以包括：

調度單元，用於根據任務依賴關係和/或應用場景需求對第三方應用進行調度；

所述應用部署單元 202 具體用於：

根據調度結果，將當前需要執行的第三方應用部署到資料容器環境中。

另外，該系統還可以包括：

安全審核單元，用於接收到第三方開發者提交的第三方應用後，對第三方應用對資料的操作進行安全審核。

為了便於第三方進行開發，該系統還可以包括：

樣本資料提供單元，用於接收到第三方開發者發送的資料申請請求後，向第三方開發者提供樣本資料，以供第三方開發者對開發出的第三方應用的邏輯進行驗證。

其中，不同的應用類型對應不同的開發入口，所述系統還可以包括：

介面提供單元，用於從某類型的開發入口接收到用戶的開發請求時，為用戶提供該類型應用的開發介面。

不同的應用類型對應不同的資料容器環境，所述系統還可以包括：

標識單元，用於從某類型的開發入口接收到用戶的開發請求後，將對應開發的第三方應用所屬的類型進行標識；

所述應用部署單元 202 具體可以用於：

根據所述第三方應用所屬的類型，將所述第三方應用部署到該類型對應的資料容器環境中。

所述應用類型包括計算類或演算法類。

具體的，應用執行單元 203 具體可以用於：

根據第三方應用需執行的任務對時間的要求，在所述資料容器環境中對所述第三方應用進行分散式或者同步處理。

另外，該系統還可以包括：

封裝單元，用於所述第三方應用執行完成後，根據輸出資料的標識封裝成介面，以便第三方開發者根據所述介面獲取第三方應用的執行結果。

還可以包括：

前台介面邏輯接收單元，用於接收第三方開發者提交的第三方應用的前台介面邏輯；

介面邏輯部署單元，用於將所述前台介面邏輯部署到資料容器中，以便在所述第三方應用執行完成後，在前台介面中顯示執行結果。

總之，通過本發明實施例，為第三方開發者提供了資料容器系統，將計算所需的資料按類別等存放在系統內部的資料容器中，而不是發送給第三方開發者，第三方開發者在開發完成第三方應用的計算邏輯或演算法邏輯之後，將第三方應用也部署到該資料容器中，這樣資料以及對資料的操作過程相當於都是在資料容器內部完成的，第三方開發者不用看到具體的資料內容，只要提供應用的計算邏輯或者演算法就好，從而解決了資料安全風險和開放高價值資料的矛盾。

通過以上的實施方式的描述可知，本領域的技術人員可以清楚地瞭解到本發明可借助軟體加必需的通用硬體平台的方式來實現。基於這樣的理解，本發明的技術方案本質上或者說對現有技術做出貢獻的部分可以以軟體產品的形式體現出來，該電腦軟體產品可以儲存在儲存介質中，如 ROM/RAM、磁碟、光碟等，包括若干指令用以使得一台電腦設備（可以是個人電腦，伺服器，或者網路設備等）執行本發明各個實施例或者實施例的某些部分所述的方法。

本說明書中的各個實施例均採用遞進的方式描述，各個實施例之間相同相似的部分互相參見即可，每個實施例重點說明的都是與其他實施例的不同之處。尤其，對於系統或系統實施例而言，由於其基本相似於方法實施例，所以描述得比較簡單，相關之處參見方法實施例的部分說明即可。以上所描述的系統及系統實施例僅僅是示意性的，其中所述作為分離部件說明的單元可以是或者也可以不是物理上分開的，作為單元顯示的部件可以是或者也可以不是物理單元，即可以位於一個地方，或者也可以分佈到多個網路單元上。可以根據實際的需要選擇其中的部分或者全部模組來實現本實施例方案的目的。本領域普通技術人員在不付出創造性勞動的情況下，即可以理解並實施。

以上對本發明所提供的第三方應用的執行方法及系統，進行了詳細介紹，本文中應用了具體個例對本發明的原理及實施方式進行了闡述，以上實施例的說明只是用於

幫助理解本發明的方法及其核心思想；同時，對於本領域的一般技術人員，依據本發明的思想，在具體實施方式及應用範圍上均會有改變之處。綜上所述，本說明書內容不應理解為對本發明的限制。

【符號說明】

201：元資料描述資訊發送單元

202：應用部署單元

203：應用執行單元

I650650

發明摘要

※申請案號：103132814

※申請日：103 年 09 月 23 日

※IPC 分類：

【發明名稱】(中文/英文)

第三方應用的執行方法及系統

【中文】

本發明公開了第三方應用的執行方法及系統，其中，所述方法包括：接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給第三方開發者，以供第三方開發者基於所述元資料描述資訊進行第三方應用的開發；接收第三方開發者提交的第三方應用的程式資訊，並將所述第三方應用部署到資料容器環境中，其中，所述資料容器環境中還保存有第三方應用在執行過程中所需的具體資料；在所述資料容器環境中執行所述第三方應用。通過本發明，解決了資料安全風險和開放高價值資料的矛盾。

【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

圖 式

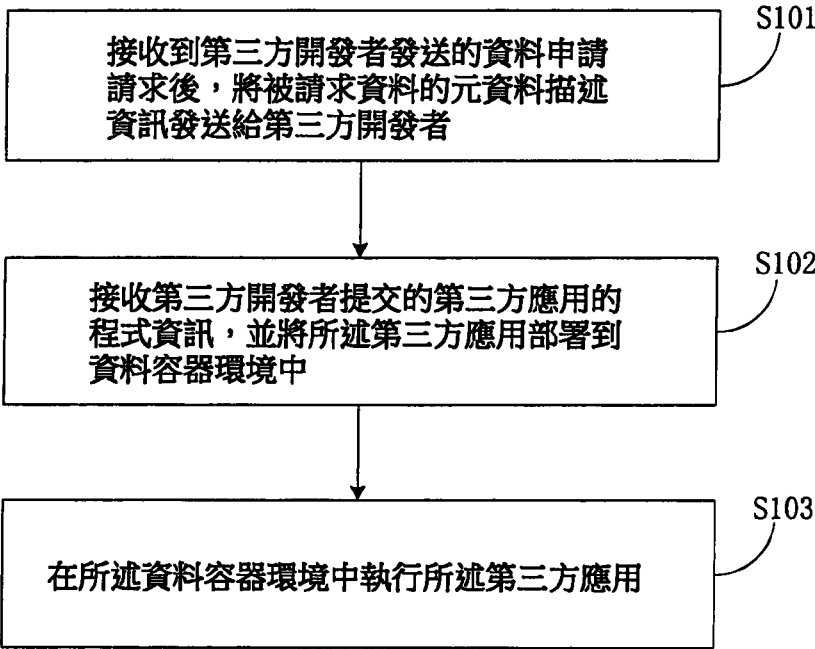


圖 1

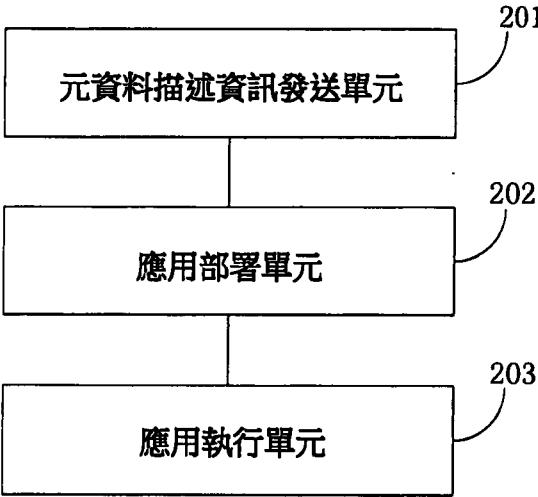


圖 2

申請專利範圍

1. 一種第三方應用的執行方法，包括：

接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給所述第三方開發者，以供所述第三方開發者基於所述元資料描述資訊進行第三方應用的開發，其中，所述被請求資料是按照類別存放在資料容器環境中；

接收所述第三方開發者提交的所述第三方應用的程式資訊，並將所述第三方應用部署到所述資料容器環境中，其中，所述資料容器環境中還保存有所述第三方應用在執行過程中所需的具體資料；

在所述資料容器環境中執行所述第三方應用。

2. 如申請專利範圍第 1 項所述的方法，其中，還包括：

根據任務依賴關係和/或應用場景需求對所述第三方應用進行調度；

所述將所述第三方應用部署到所述資料容器環境中，包括：

根據調度結果，將當前需要執行的所述第三方應用部署到所述資料容器環境中。

3. 如申請專利範圍第 1 項所述的方法，其中，還包括：

接收到所述第三方開發者提交的所述第三方應用的程式資訊後，對所述第三方應用對資料的操作進行安全審

核。

4. 如申請專利範圍第 1 項所述的方法，其中，還包括：

接收到所述第三方開發者發送的資料申請請求後，向所述第三方開發者提供樣本資料，以供所述第三方開發者對開發出的所述第三方應用的邏輯進行驗證。

5. 如申請專利範圍第 1 項所述的方法，其中，不同的應用類型對應不同的開發入口，所述方法還包括：

從某類型的開發入口接收到用戶的開發請求時，為所述用戶提供該類型應用的開發介面。

6. 如申請專利範圍第 5 項所述的方法，其中，不同的應用類型對應不同的資料容器環境，所述方法還包括：

從所述某類型的開發入口接收到所述用戶的開發請求後，將對應開發的所述第三方應用所屬的類型進行標識；

所述將所述第三方應用部署到所述資料容器環境中，包括：

根據所述第三方應用所屬的類型，將所述第三方應用部署到該類型對應的所述資料容器環境中。

7. 如申請專利範圍第 5 項或第 6 項所述的方法，所述應用類型包括計算類或演算法類。

8. 如申請專利範圍第 1 項所述的方法，其中，所述在所述資料容器環境中執行所述第三方應用，包括：

根據所述第三方應用需執行的任務對時間的要求，在所述資料容器環境中對所述第三方應用進行分散式或者同

步處理。

9. 如申請專利範圍第 1 項至第 6 項、第 8 項中任一項所述的方法，其中，還包括：

所述第三方應用執行完成後，根據輸出資料的標識封裝成介面，以便所述第三方開發者根據所述介面獲取所述第三方應用的執行結果。

10. 如申請專利範圍第 1 項至第 6 項、第 8 項中任一項所述的方法，其中，還包括：

接收所述第三方開發者提交的所述第三方應用的前台介面邏輯；

將所述前台介面邏輯部署到資料容器中，以便在所述第三方應用執行完成後，在前台介面中顯示執行結果。

11. 一種第三方應用的執行系統，包括：

元資料描述資訊發送單元，用於接收到第三方開發者發送的資料申請請求後，將被請求資料的元資料描述資訊發送給所述第三方開發者，以供所述第三方開發者基於所述元資料描述資訊進行第三方應用的開發，其中，所述被請求資料是按照類別存放在所述資料容器環境中；

應用部署單元，用於接收所述第三方開發者提交的第三方應用的程式資訊，並將所述第三方應用部署到所述資料容器環境中，其中，所述資料容器環境中還保存有第三方應用在執行過程中所需的具體資料；

應用執行單元，用於在所述資料容器環境中執行所述第三方應用。