201935343

(19)中華民國智慧財產局

(12)發明說明書公開本 (11)公開編號：TW 201935343 A

(43)公開日： 中華民國 108 (2019) 年 09 月 01 日

(21)申請案號：107146608 (22)申請日： 中華民國 107 (2018) 年 12 月 22 日

(51)Int. Cl.： *G06Q10/06 (2012.01)* *G06F15/163 (2006.01)*

(30)優先權：2018/02/13 中國大陸 201810148300.2

(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：瞿紅衛 (CN)

(74)代理人：林志剛

申請實體審查：有 申請專利範圍項數：21 項 圖式數：5 共 32 頁

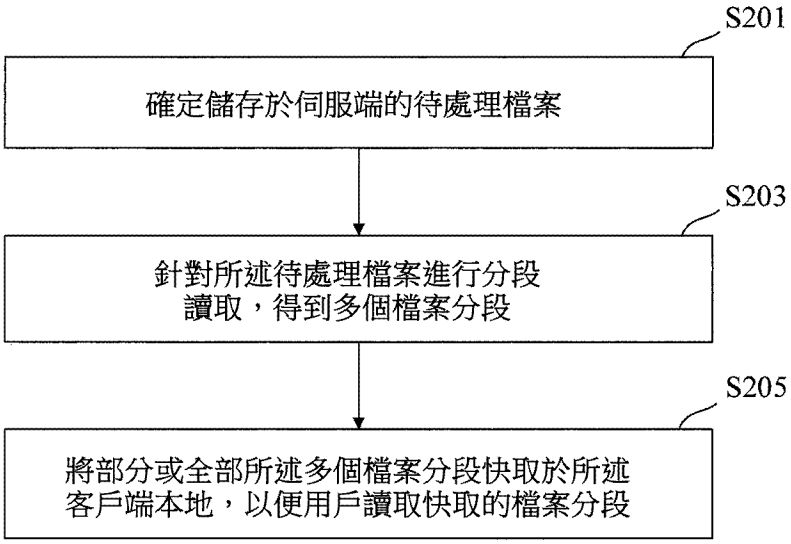
(54)名稱

檔案處理方法、裝置及設備

(57)摘要

本說明書實施例揭露了一種檔案處理方法、裝置及設備。對於使用客戶端讀取儲存在伺服端中檔案的場景而言，在確定了儲存於伺服端的待處理檔案之後，便可以透過客戶端從伺服端中分段讀取相應的待處理檔案，從而得到多個檔案分段，並可將分段讀取得到的快取至客戶端本地。顯然，正是採用分段讀取的方式，能夠在同一時間內並行式地讀取到同一待處理檔案的不同內容。

指定代表圖：



【圖 2】

【發明說明書】

【中文發明名稱】

檔案處理方法、裝置及設備

【技術領域】

本案涉及電腦技術領域，尤其涉及一種檔案處理方法、裝置及設備。

【先前技術】

目前，用戶在使用諸如：物件儲存服務(Object Storage Service，OSS)等雲端儲存業務的場景下，能夠使用本地客戶端從雲端獲取相應的檔案，從而可在客戶端本地讀取該檔案。

對於上述場景，在現有技術中，客戶端與伺服端之間通常採用inputstream輸入流的方式傳輸資料，具體地，針對伺服端中儲存的檔案，透過inputstream輸入流將該檔案中一定位元組的資料快取至客戶端本地，以便用戶進行讀取。

客戶端和伺服端之間採用inputstream輸入流的方式，往往需要二者之間長期保持連接狀態，並且，採用inputstream輸入流通常是逐個位元組進行傳輸。

基於現有技術，需要一種更為有效的檔案處理方式。

【發明內容】

本說明書實施例提供一種檔案處理方法、裝置及設備，用以提供一種更為有效的從伺服端獲取檔案的方案。

本說明書實施例採用下述技術方案：

本說明書實施例提供一種檔案處理方法，包括：

客戶端確定儲存於伺服端的待處理檔案；

針對所述待處理檔案進行分段讀取，得到多個檔案分段；

將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

對應地，本說明書實施例還提供一種檔案處理裝置，包括：

檔案確定模組，確定儲存於伺服端的待處理檔案；

分段讀取模組，針對所述待處理檔案進行分段讀取，得到多個檔案分段；

快取模組，將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

對應地，本說明書實施例還提供一種檔案處理設備，包括：處理器、記憶體，其中：

所述記憶體，儲存檔案處理程式；

所述處理器，調用記憶體中儲存的檔案處理序，並執行：

確定儲存於伺服端的待處理檔案；

針對所述待處理檔案進行分段讀取，得到多個檔案分段；

將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

本說明書實施例採用的上述至少一個技術方案能夠達到以下有益效果：

對於使用客戶端讀取儲存在伺服端中檔案的場景而言，在確定了儲存於伺服端的待處理檔案之後，便可以透過客戶端從伺服端中分段讀取相應的待處理檔案，從而得到多個檔案分段，並可將分段讀取得到的快取至客戶端本地。顯然，正是採用分段讀取的方式，能夠在同一時間內並行式地讀取到同一待處理檔案的不同內容，相較於inputstream輸入流等傳輸方式，分段式的讀取方式能夠在短時間內獲得足量的檔案資料並快取在客戶端本地，傳輸效率有效提升。

【圖式簡單說明】

此處所說明的圖式用來提供對本案的進一步理解，構成本案的一部分，本案的示意性實施例及其說明用於解釋本案，並不構成對本案的不當限定。在圖式中：

圖1為本說明書實施例提供的檔案處理方法所基於的架構示意圖；

圖2為本說明書實施例提供的檔案處理過程示意圖；

圖3為本說明書實施例提供的有序分段讀取的原理示意圖；

圖4為本說明書實施例提供的無序分段讀取的原理示

意圖；

圖5為本說明書實施例提供的檔案處理裝置結構示意圖。

【實施方式】

為使本案的目的、技術方案和優點更加清楚，下面將結合本案具體實施例及相應的圖式對本案技術方案進行清楚、完整地描述。顯然，所描述的實施例僅是本案一部分實施例，而不是全部的實施例。基於本案中的實施例，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，都屬於本案保護的範圍。

正如前述，對於雲端儲存業務場景而言，用戶所使用的客戶端與雲端伺服端之間可透過inputstream輸入流的方式進行資料的傳輸。但inputstream輸入流的方式通常是針對某一待傳輸檔案，進行逐個位元組的傳輸，顯然，在對業務執行時間要求較短的業務場景下，該方式的傳輸效率存在瓶頸。此外，採用inputstream輸入流的方式要求客戶端和伺服端之間一直保持連接狀態，通常情況下，伺服器由於自身的連接檢測機制，會將一定時長內未進行資料傳輸的連接斷開，所以，對於業務執行時間較長的業務場景，採用inputstream輸入流的方式可能會導致連接斷開，從而需要重連，增加了連接成本。

故為了盡可能的減少或避免上述情況，本說明書實施例中提供一種檔案處理方法，透過對檔案執行分段讀取操

作，調用多個執行緒一次性讀取多個檔案分段並快取至客戶端本地，以便用戶讀取快取的檔案，提升對檔案的讀取效率。

同時，為便於理解，在本說明書的一個或多個實施例中，儲存在雲端伺服端中的待處理檔案可以是諸如：可執行檔案(後綴為.exe)、壓縮檔案(後綴為.rar)、系統檔案(後綴為.dll)、資料檔案(後綴為.csv、.xls等)以及各類圖像、音視訊檔案等等，這裡不再一一列舉。而如果採用前述的OSS雲端儲存服務，則待處理檔案還可以儲存為統一的檔案格式，即，OSS檔案。當然，這裡並不應構成對本案的限定。

本說明書實施例中涉及針對待處理的檔案進行分段處理(File Split)，可將經過分段處理後所得到的部分檔案稱為：檔案分段。其中，對待處理檔案進行分段式地讀取、下載等處理方式，也可理解為本說明書實施例中所述的分段處理。在後續實施例中，將客戶端從雲端伺服端分段式地獲取檔案，稱為分段讀取，這裡並不應理解為對本案的限定。

在本說明書實施例中所述的檔案處理方法可以採用如圖1所示的架構，該架構中至少可包括：客戶端以及雲端伺服端(以下可簡稱為：伺服端)。

具體而言，所述客戶端，可認為是由雲端儲存業務提供方所提供的應用程式(Application，APP)，該客戶端運行在用戶所使用的終端上。此外，該客戶端可在其運行的

終端上創建多個分段檔案讀取執行緒，該分段檔案讀取執行緒用於建立與伺服端之間的連接，並針對指定的待處理檔案將分段後的檔案分段快取在終端記憶體中的容器內，以便用戶透過該客戶端進行讀取。

所述的伺服端，可認為是用於提供雲端儲存業務的業務伺服器，當然，業務伺服器進一步可以採用單一伺服器、集群式伺服器或分散式伺服器等架構，這裡並不作具體限定。

基於上述如圖1所述的架構，以下將詳細說明本說明書實施例中所提供的技術方案。

本說明書實施例中提供一種檔案處理方法，如圖2所述，具體包括以下步驟：

步驟S201：客戶端確定儲存於伺服端的待處理檔案。

如前所述，待處理檔案儲存在伺服端中，若用戶需要讀取待處理檔案，則需要透過客戶端選擇並確定所需讀取的待處理檔案。

當然，在本說明書的一種可行實施例中，用戶透過客戶端可以瀏覽伺服端中所儲存的檔案，在此基礎上，用戶可透過客戶端選定某一/某些檔案。對於客戶端而言，將記錄用戶所選擇的檔案標識(如：檔案名、檔案序號、儲存路徑等)，以此來確定相應的待處理檔案。後續也可根據檔案標識，讀取儲存在伺服端中相應的檔案。

對於客戶端而言，用戶所發出的選擇操作，可看作是一種檔案讀取指令，除此之外，還可以透過相應的服務/

程式按照設定的規則，自動向客戶端發出檔案讀取指令，所以，在本說明書的一種可行實施例中，客戶端確定儲存於伺服端的待處理檔案，就可以包括：客戶端接收檔案讀取指令，根據所述檔案讀取指令確定儲存於伺服端的待處理檔案。

步驟S203：針對所述待處理檔案進行分段讀取，得到多個檔案分段。

一般來說，在本說明書中部分實施例所涉及到的業務場景中，儲存在伺服端中的檔案通常是較大的檔案，如：幾GB~幾百GB，甚至可達TB級別。如此一來，若要針對上述檔案進行讀取，使用現有的inputstream輸入流將會導致傳輸效率較低。

因此，在本說明書實施例中，可以針對儲存在伺服端的待處理檔案進行分段式的讀取，而分段讀取可以是並行的，也就是說，可以針對同一個待處理檔案並行地讀取該檔案中的不同內容，得到多個檔案分段。

可以理解的是，每一個檔案分段中所包含的檔案內容僅是待處理檔案的一部分，在一些業務場景中，分段讀取得到的多個檔案分段，包含了待處理檔案的全部內容。而在待處理檔案過大的情況下，分段讀取後得到的多個檔案分段，也可能僅是待處理檔案的一部分。

作為本說明書實施例中的一種可行方式，分段讀取得到的檔案分段的大小也可以進行預先設置，如，設置分段大小為100MB，也即，在此情況下分段讀取得到的每一檔

案分段的大小為100MB。當然，具體可根據實際應用的需要進行設置，這裡並不構成對本案的限定。

此外還需要說明的是，分段讀取可認為是客戶端與伺服端建立相應的網路連接，並透過網路連接針對伺服端中的待處理檔案進行的分段讀取，該網路連接可以採用超文本傳輸協議(Hyper Text Transfer Protocol，HTTP)、安全超文本傳輸協議(Hyper Text Transfer Protocol over Secure Socket，HTTPS)等實現，這裡並不作具體限定。

一般來說，客戶端可以在其運行的終端操作系統中創建或調用相應的執行緒，實現上述分段讀取操作，也即，針對所述待處理檔案進行分段讀取的過程具體可包括：客戶端調用或創建設定數量的分段讀取執行緒，透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

步驟S205：將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

在本說明書實施例中，對於多個檔案分段，可以快取在客戶端所運行的終端記憶體中。這樣一來，用戶便可以透過客戶端快速便捷地讀取快取在記憶體中的檔案分段。

當能，在實際應用中，受記憶體中儲存容量的限制，針對讀取到的多個檔案分段，也可能只有部分檔案分段優先快取在記憶體中，後續隨著記憶體中快取的資料被清除或記憶體空間的釋放，剩餘的部分檔案分段可以隨之快取至記憶體中。當然，這裡並不構成對本案的限定。

透過上述步驟，對於使用客戶端讀取儲存在伺服端中

檔案的場景而言，在確定了儲存於伺服端的待處理檔案之後，便可以透過客戶端從伺服端中分段讀取相應的待處理檔案，從而得到多個檔案分段，並可將分段讀取得到的快取至客戶端本地。顯然，正是採用分段讀取的方式，能夠在同一時間內並行式地讀取到同一待處理檔案的不同內容，相較於inputstream輸入流等傳輸方式，分段式的讀取方式能夠在短時間內獲得足量的檔案資料並快取在客戶端本地，傳輸效率有效提升。

對於前述如圖2所示的方法，通常而言，其執行主體可以是客戶端，在部分應用場景中，也可能是客戶端所運行於的終端操作系統，具體而言，如果客戶端具有足夠的權限，則透過該客戶端可以創建並調用相應的執行緒，完成上述步驟，而如果客戶端並未獲得足夠的權限，則可由客戶端請求終端操作系統中的相應功能/服務執行上述步驟。並且，在執行上述步驟的過程中，執行主體可以發生變化，例如：可由客戶端執行上述步驟S201，再由終端操作系統執行上述步驟S203及S205。當然，具體將根據實際應用的需要進行確定，這裡並不構成對本案的限定。

針對上述方法需要說明的是，在實際應用場景中，對待處理檔案進行分段讀取和快取的過程有不同的處理方式，下面將進行詳細說明：

方式一、有序分段讀取

有序分段讀取通常要求讀取得到的多個檔案分段是順

序排列的，當然，快取在記憶體中時，也同樣按照相應的順序進行快取。

具體而言，如圖3所示，為執行有序分段讀取時的原理示意圖。在圖3中，假設需要針對儲存於伺服端中的待處理檔案A進行有序分段讀取，那麼，可執行如下步驟：

步驟S1：客戶端根據檔案讀取指令，確定需要讀取的待處理檔案A。

在本示例中，檔案讀取指令可以由用戶發出，換言之，客戶端可以向用戶提供相應的檔案選擇介面，透過該介面，用戶可瀏覽到儲存在伺服端中的相應檔案。並且可以在該選擇介面中透過諸如控制項、選項等方式發出檔案讀取指令。同樣，如前所述，客戶端根據檔案讀取指令，可以獲知相應的檔案標識，並可進一步根據該檔案標識來確定待處理檔案A。該過程不再過多贅述。

步驟S3：創建或調用設定數量的分段讀取執行緒，透過所述分段讀取執行緒針對所述待處理檔案A執行分段讀取。

客戶端可以創建或調用一定數量的分段讀取執行緒，每一分段讀取執行緒相互獨立，並且，每一分段讀取執行緒能夠自行建立與伺服端建立網路連接。作為本說明書實施例中的一種可行方式，分段讀取執行緒可以基於檔案讀取指令中所攜帶的檔案標識，從伺服端分段讀取待處理檔案A。

這裡需要說明的是，在圖3中，由若干個分段讀取執

行緒將待處理檔案A劃分為7個有序的檔案分段(應注意的是，圖3中的劃分式樣僅是為了表達對待處理檔案A的分段讀取，並非將儲存的待處理檔案A拆分為7個部分)。

步驟S5：將分段處理得到的多個有序檔案分段儲存在客戶端本地的記憶體容器中。

在本示例中，記憶體容器可採用Map容器，該容器適用於有序儲存。在一種方式下，記憶體中設置一個Map容器，其容器大小(容量)可以與記憶體大小(記憶體的儲存空間)相一致，也可以小於記憶體大小。

在另一種方式下，記憶體中可以設置多個Map容器，多個Map容器的總大小可以與記憶體大小相一致。Map容器的數量可以與分段讀取執行緒的數量一致，也可以不一致，具體可根據實際應用的需要進行設置，這裡便不再過多贅述。

同時，在快取時，針對不同的檔案分段生成相應的分段標識，即，分段key，從而形成一種鍵-值對的結構：

分段key-檔案分段

這裡的分段key能夠起到序號的作用，從而有序地儲存在Map容器中。

步驟S7：針對儲存在記憶體容器中的有序檔案分段進行順序讀取。

在本示例中，用戶可以透過客戶端針對檔案進行順序讀取。

對於上述過程而言，客戶端通常可基於記憶體容器的使用程度控制執行緒的排程或創建，具體來說，容器大小達到上限不會將執行緒排程去讀取待處理檔案A，沒有達到上限會自動讀取後面的分段資料，用戶讀完一個分段的資料後將容器中資料刪除。

方式二、無序分段讀取

區別於前述的有序分段讀取，無序分段讀取並不要求多個檔案分段是否有序，多個分段讀取執行緒可以讀取待處理檔案中任意部分(應理解，無論是採用有序分段讀取或是無序分段讀取，得到的每一檔案分段所對應的檔案內容應該是不同的)

如圖4所示，為執行無序分段讀取時的原理示意圖。結合圖4，與前述有序分段讀取過程的相同或相似部分不再過多贅述，區別在於：

圖4中的記憶體儲存結構，可以採用有界塊佇列，透過有界塊佇列，可以控制分段讀取的速度，以防止檔案分段在記憶體中堆積。

進一步地，在本說明書實施例中，可以採用分段計數器計算有界塊佇列中所包含的檔案分段的數量，並以此來判斷該有界塊佇列中的檔案分段是否被讀取完畢。

當用戶透過客戶端讀取有界塊佇列中的檔案分段時，將按照先進先出的機制進行讀取，同時，用戶讀取完一個檔案分段後，則客戶端可在有界塊佇列中將該檔案分段刪

除，自動調用前述的分段讀取執行緒獲取下一檔案分段，並快取至有界塊佇列中。

從上述的內容可見，本方案可將較大的檔案進行分段式讀取，得到多個檔案分段。用戶對快取的檔案分段的讀取過程和從伺服端獲取檔案分段的過程異步化，具體而言，在用戶讀取快取的檔案分段之前，客戶端可以同時獲得多個檔案分段，供用戶讀取，且當用戶讀取後，可以再獲取新的檔案分段。用戶透過客戶端讀取檔案分段的操作是純記憶體操作，並非遠端獲取，在提升檔案傳輸效率的同時，能夠減緩或避免客戶端與伺服端之間連接超時導致檔案讀取中斷的情況。

以上為本說明書實施例提供的檔案處理方法，基於同樣的思路，本說明書實施例還提供相應的檔案處理裝置。

具體而言，本說明書實施例中所提供的檔案處理裝置如圖5所示，所述裝置包括：

檔案確定模組501，確定儲存於伺服端的待處理檔案；

分段讀取模組502，針對所述待處理檔案進行分段讀取，得到多個檔案分段；

快取模組503，將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

進一步地，檔案確定模組501，接收檔案讀取指令，根據所述檔案讀取指令確定儲存於伺服端的待處理檔案。

進一步地，分段讀取模組502，調用或創建設定數量

的分段讀取執行緒，透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

進一步地，分段讀取模組502，按照設定的檔案分段大小，透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

進一步地，分段讀取模組502，透過所述分段讀取執行緒，針對所述待處理檔案進行有序分段讀取，得到多個有序檔案分段。

進一步地，快取模組503，將部分或全部的所述多個有序檔案分段，按照順序快取於記憶體的Map容器中。

其中，所述Map容器的數量為一個或多個，且Map容器的容量不大於所述內容的容量。

進一步地，快取模組503，根據所述Map容器的容量，將部分或全部的所述多個有序檔案分段按照順序快取於記憶體的Map容器中。

進一步地，分段讀取模組502，透過所述分段讀取執行緒，針對所述待處理檔案進行無序分段讀取，得到多個無序檔案分段。

進一步地，快取模組503，將部分或全部的所述多個無序檔案分段，快取至記憶體的有界塊佇列中。

進一步地，所述裝置還包括：資料清除模組504，當用戶讀取快取的檔案分段後，刪除快取的該檔案分段。

基於圖5所示的裝置，在實際應用中可由實體的設備(如：終端設備)所實現，具體而言，該設備包括：處理

器、記憶體，其中，

所述記憶體，儲存檔案處理程式；

所述處理器，調用記憶體中儲存的檔案處理程式，並執行：

確定儲存於伺服端的待處理檔案；

針對所述待處理檔案進行分段讀取，得到多個檔案分段；

將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

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，對於裝置、設備和媒體類實施例而言，由於其基本相似於方法實施例，所以描述的比較簡單，相關之處參見方法實施例的部分說明即可，這裡就不再一一贅述。

至此，已經對本主題的特定實施例進行了描述。其它實施例在所附申請專利範圍的範疇內。在一些情況下，在申請專利範圍中記載的動作可以按照不同的順序來執行並且仍然可以實現期望的結果。另外，在圖式中描繪的過程不一定要求示出的特定順序或者連續順序，以實現期望的結果。在某些實施方式中，多任務處理和平行處理可以是有利的。

在20世紀90年代，對於一個技術的改進可以很明顯地區分是硬體上的改進(例如，對二極體、電晶體、開關等

電路結構的改進)還是軟體上的改進(對於方法流程的改進)。然而，隨著技術的發展，當今的很多方法流程的改進已經可以視為硬體電路結構的直接改進。設計人員幾乎都透過將改進的方法流程程式化到硬體電路中來得到相應的硬體電路結構。因此，不能說一個方法流程的改進就不能用硬體實體模組來實現。例如，可程式化邏輯裝置(Programmable Logic Device, PLD)(例如現場可程式化閘陣列(Field Programmable Gate Array，FPGA))就是這樣一種積體電路，其邏輯功能由用戶對裝置程式化來確定。由設計人員自行程式化來把一個數位系統“整合”在一片PLD上，而不需要請晶片製造廠商來設計和製作專用的積體電路晶片。而且，如今，取代手工地製作積體電路晶片，這種程式化也多半改用“邏輯編譯器(logic compiler)”軟體來實現，它與程式開發撰寫時所用的軟體編譯器相類似，而要編譯之前的原始碼也得用特定的程式化語言來撰寫，此稱之為硬體描述語言(Hardware Description Language，HDL)，而HDL也並非僅有一種，而是有許多種，如ABEL(Advanced Boolean Expression Language)、AHDL(Altera Hardware Description Language)、Confluence、CUPL(Cornell University Programming Language)、HDCal、JHDL(Java Hardware Description Language)、Lava、Lola、MyHDL、PALASM、RHDL(Ruby Hardware Description Language)等，目前最普遍使用的是VHDL(Very-High-Speed Integrated Circuit Hardware Description

Language)與Verilog。本領域技術人員也應該清楚，只需要將方法流程用上述幾種硬體描述語言稍作邏輯程式化並程式化到積體電路中，就可以很容易得到實現該邏輯方法流程的硬體電路。

控制器可以按任何適當的方式實現，例如，控制器可以採取例如微處理器或處理器以及儲存可由該(微)處理器執行的電腦可讀程式碼(例如軟體或韌體)的電腦可讀媒體、邏輯閘、開關、專用積體電路(Application Specific Integrated Circuit，ASIC)、可程式化邏輯控制器和嵌入微控制器的形式，控制器的例子包括但不限於以下微控制器：ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20 以及Silicone Labs C8051F320，記憶體控制器還可以被實現為記憶體的控制邏輯的一部分。本領域技術人員也知道，除了以純電腦可讀程式碼方式實現控制器以外，完全可以透過將方法步驟進行邏輯程式化來使得控制器以邏輯閘、開關、專用積體電路、可程式化邏輯控制器和嵌入微控制器等的形式來實現相同功能。因此這種控制器可以被認為是一種硬體部件，而對其內包括的用於實現各種功能的裝置也可以視為硬體部件內的結構。或者甚至，可以將用於實現各種功能的裝置視為既可以是實現方法的軟體模組又可以是硬體部件內的結構。

上述實施例闡明的系統、裝置、模組或單元，具體可以由電腦晶片或實體實現，或者由具有某種功能的產品來實現。一種典型的實現設備為電腦。具體的，電腦例如可

以為個人電腦、膝上型電腦、蜂巢式電話、相機電話、智慧型電話、個人數位助理、媒體播放器、導航設備、電子郵件設備、遊戲控制台、平板電腦、可穿戴設備或者這些設備中的任何設備的組合。

為了描述的方便，描述以上裝置時以功能分為各種單元分別描述。當然，在實施本案時可以把各單元的功能在同一個或多個軟體和/或硬體中實現。

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、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媒體(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(系統)、和電腦程式產品的流程圖和/或方塊圖來描述的。應理解可由電腦程式指令實現流程圖和/或方塊圖中的每一流程和/或方塊、以及流程圖和/或方塊圖中的流程和/或方塊的結合。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化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化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/或方塊圖一個方塊或多個方塊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化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/或方塊圖一個方塊或多個方塊中指定的功能。

這些電腦程式指令也可裝載到電腦或其他可程式化資料處理設備上，使得在電腦或其他可程式化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化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/或方塊圖一個方塊或多個方塊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媒體中的非永久性記憶體，隨機存取記憶體(RAM)和/或非揮發性記憶體等形式，如唯讀記憶體(ROM)或快閃記憶體(flash RAM)。記憶體是電腦可讀媒體的示例。

電腦可讀媒體包括永久性和非永久性、可移除式和非可移除式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。電腦的儲存媒體的例子包括，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)、電可抹除可程式化唯讀記憶體(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，磁帶磁碟儲存或其他磁性儲存設備或任何其他非傳輸媒體，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，電腦可讀媒體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，本案的實施例可提供為方法、系統或電腦程式產品。因此，本案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，本案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媒體(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。

本案可以在由電腦執行的電腦可執行指令的一般上下文中描述，例如程式模組。一般地，程式模組包括執行特定事務或實現特定抽象資料類型的例程、程式、對象、組

件、資料結構等等。也可以在分散式計算環境中實踐本案，在這些分散式計算環境中，由透過通信網路而被連接的遠端處理設備來執行事務。在分散式計算環境中，程式模組可以位於包括儲存設備在內的本地和遠端電腦儲存媒體中。

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，對於系統實施例而言，由於其基本相似於方法實施例，所以描述的比較簡單，相關之處參見方法實施例的部分說明即可。

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

【符號說明】

S201, S203, S205：步驟

S1, S3, S5, S7：步驟

501：檔案確定模組

502：分段讀取模組

503：快取模組

504：資料清除模組

201935343

【發明摘要】

【中文發明名稱】

檔案處理方法、裝置及設備

【中文】

本說明書實施例揭露了一種檔案處理方法、裝置及設備。對於使用客戶端讀取儲存在伺服端中檔案的場景而言，在確定了儲存於伺服端的待處理檔案之後，便可以透過客戶端從伺服端中分段讀取相應的待處理檔案，從而得到多個檔案分段，並可將分段讀取得到的快取至客戶端本地。顯然，正是採用分段讀取的方式，能夠在同一時間內並行式地讀取到同一待處理檔案的不同內容。

【指定代表圖】第(２)圖。
【代表圖之符號簡單說明】無

【特徵化學式】無

【發明申請專利範圍】

【第1項】

一種檔案處理方法，包括：

客戶端確定儲存於伺服端的待處理檔案；

針對所述待處理檔案進行分段讀取，得到多個檔案分段；

將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

【第2項】

如申請專利範圍第1項所述的方法，其中該客戶端確定儲存於伺服端的待處理檔案，具體包括：

所述客戶端接收檔案讀取指令；

根據所述檔案讀取指令確定儲存於伺服端的待處理檔案。

【第3項】

如申請專利範圍第1項所述的方法，其中針對所述待處理檔案進行分段讀取，具體包括：

所述客戶端調用或創建設定數量的分段讀取執行緒；

透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

【第4項】

如申請專利範圍第3項所述的方法，其中透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取，具體包括：

按照設定的檔案分段大小，透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

【第5項】

如申請專利範圍第3項所述的方法，其中透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取，具體包括：

透過所述分段讀取執行緒，針對所述待處理檔案進行有序分段讀取，得到多個有序檔案分段。

【第6項】

如申請專利範圍第5項所述的方法，其中將部分或全部所述多個檔案分段快取於所述客戶端本地，具體包括：

將部分或全部的所述多個有序檔案分段，按照順序快取於記憶體的Map容器中；

其中，所述Map容器的數量為一個或多個，且Map容器的容量不大於所述內容的容量。

【第7項】

如申請專利範圍第6項所述的方法，其中將部分或全部的所述多個有序檔案分段，按照順序快取於記憶體的Map容器中，具體包括：

根據所述Map容器的容量，將部分或全部的所述多個有序檔案分段按照順序快取於記憶體的Map容器中。

【第8項】

如申請專利範圍第3項所述的方法，其中透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取，具體包

括：

透過所述分段讀取執行緒，針對所述待處理檔案進行無序分段讀取，得到多個無序檔案分段。

【第9項】

如申請專利範圍第8項所述的方法，其中將部分或全部所述多個檔案分段快取於所述客戶端本地，具體包括：

將部分或全部的所述多個無序檔案分段，快取至記憶體的有界塊佇列中。

【第10項】

如申請專利範圍第1項所述的方法，所述方法還包括：

當用戶讀取快取的檔案分段後，刪除快取的該檔案分段。

【第11項】

一種檔案處理裝置，包括：

檔案確定模組，確定儲存於伺服端的待處理檔案；

分段讀取模組，針對所述待處理檔案進行分段讀取，得到多個檔案分段；

快取模組，將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。

【第12項】

如申請專利範圍第11項所述的裝置，其中所述檔案確定模組，接收檔案讀取指令，根據所述檔案讀取指令確定儲存於伺服端的待處理檔案。

【第13項】

如申請專利範圍第11項所述的裝置，其中所述分段讀取模組，調用或創建設定數量的分段讀取執行緒，透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

【第14項】

如申請專利範圍第13項所述的裝置，其中所述分段讀取模組，按照設定的檔案分段大小，透過所述分段讀取執行緒針對所述待處理檔案進行分段讀取。

【第15項】

如申請專利範圍第13項所述的裝置，其中所述分段讀取模組，透過所述分段讀取執行緒，針對所述待處理檔案進行有序分段讀取，得到多個有序檔案分段。

【第16項】

如申請專利範圍第15項所述的裝置，其中所述快取模組，將部分或全部的所述多個有序檔案分段，按照順序快取於記憶體的Map容器中；

其中，所述Map容器的數量為一個或多個，且Map容器的容量不大於所述內容的容量。

【第17項】

如申請專利範圍第16項所述的裝置，其中所述快取模組，根據所述Map容器的容量，將部分或全部的所述多個有序檔案分段按照順序快取於記憶體的Map容器中。

【第18項】

如申請專利範圍第13項所述的裝置，其中所述分段讀

取模組，透過所述分段讀取執行緒，針對所述待處理檔案進行無序分段讀取，得到多個無序檔案分段。

【第19項】

如申請專利範圍第18項所述的裝置，其中所述快取模組，將部分或全部的所述多個無序檔案分段，快取至記憶體的有界塊佇列中。

【第20項】

如申請專利範圍第11項所述的裝置，所述裝置還包括：資料清除模組，當用戶讀取快取的檔案分段後，刪除快取的該檔案分段。

【第21項】

一種檔案處理設備，包括：處理器、記憶體，其中：

所述記憶體，儲存檔案處理程式；

所述處理器，調用記憶體中儲存的檔案處理程式，並執行：

確定儲存於伺服端的待處理檔案；

針對所述待處理檔案進行分段讀取，得到多個檔案分段；

將部分或全部所述多個檔案分段快取於所述客戶端本地，以便用戶讀取快取的檔案分段。