Table2Graph: 使用 Map-Reduce 從關聯表構建可擴展圖 - 論文解析

Abstract（摘要）

翻譯： 識別實體在不同數據集（或數據庫）內部和之間的相關性和關係在許多領域中具有重要意義。目前最廣泛實踐的數據倉庫整合方法被發現不足以實現這一目標。相反，我們探索了一種替代解決方案，將多個不同的數據源轉化為單一異構圖模型，這樣通過檢查圖中的連結，可以加速跨不同數據源的實體匹配。然而，我們發現，雖然基於圖的模型為此目的提供了出色的功能，但從關聯源數據庫構建這種模型非常耗時，主要依賴特定的專有腳本。這促使我們開發了一個可重新配置和可重用的圖構建工具，設計為大規模工作。在本文中，我們介紹了 Table2Graph，這是一個基於 Hadoop 上 Map-Reduce 框架的圖構建工具。我們還討論了將 Table2Graph 應用於整合不同醫療保健數據庫的結果。

解析與重點：

- **問題背景**：在不同數據集之間識別實體關係的重要性及傳統數據倉庫方法的局限性
- **提出的解決方案**：Table2Graph 工具，將多個數據源轉換為單一異構圖模型
- **技術基礎**：基於 Hadoop 的 Map-Reduce 框架
- **主要優勢**：
    1. 可重新配置性：方便應對不同數據源
    2. 可擴展性：設計用於處理大規模數據
    3. 效率：加速跨數據源的實體匹配
- **實際應用**：整合醫療保健數據庫

I. INTRODUCTION（介紹）

翻譯： 數據整合涉及將不同來源和結構的數據整合並將其合併成統一模型。傳統上，這需要設計一個通用的提取、轉換和加載(ETL)過程，當分別應用於每個不同的數據源時，會產生符合數據倉庫中統一數據庫模型的數據。ETL 過程反覆執行，以將源數據的變化整合到統一模型中。

數據倉庫解決方案具有許多固有的缺點。一旦建立了統一模型的模式，現實中很難修改設計，限制了分析只能在模型的語義結構內進行。此外，源數據庫模式的輕微變化可能需要重新實現 ETL 過程，因為它與源模式緊密耦合。此外，隨著每個數據源數據量的爆炸性增長，ETL 過程需要更頻繁地執行，這被發現是不可擴展的。

識別不同數據源（或數據庫）內部或之間實體之間的相關性被發現在許多領域具有重要的戰略意義。一個例子是 2010 年《患者保護與平價醫療法案》(PPACA)的第 6402(a)條。該法規旨在通過識別跨多個計劃的潛在不良行為者來保護聯邦和州醫療保健計劃的完整性。為了實現這一目標，第 6402(a)條要求整合數據庫至少包括來自 Medicare、Medicaid、州兒童健康保險計劃（CHIP）和社會保障的數據進行匹配。

與美國醫療保險和醫療補助服務中心（CMS）合作，橡樹嶺國家實驗室（ORNL）探索了解決第 6402(a)條實施數據整合挑戰的創新方法和手段。考慮到前述數據倉庫解決方案的問題以及主要需求，即匹配，我們決定開發一個靈活模式的模型，在 ETL 過程中也不會丟失任何源數據信息。特別是，我們選擇了一個異構圖基模型，通過語義連接（而非合併）整合多個數據源，從源特徵（表示為節點和屬性）和關係（表示為邊）的角度，使得行為者之間的關係表示為圖中特徵的連接。與傳統關係模型不同，基於圖的模型直接嵌入其自然形式的連接，因此可以非常高效地處理關係相關查詢，否則需要在規範化關係模型中進行多次深層連接操作。

然而，即使對於構建基於圖的整合模型，從規範化關係模型構建圖模型的 ETL 過程仍然不是一項簡單的任務。首先，整合圖模型可能包括數十億個節點和邊。數據源可能包括數十億行和數百列，跨越多個關係表。因此，應該開發一個可擴展的 ETL 解決方案，以便源數據的快速增加能夠以微不足道的延遲整合到整合模型中。其次，整合模型可能會改變其設計，或源數據庫不定期更新其模式。因此，應該開發一個可重新配置的 ETL 過程，無需太多修改即可重複使用。在探索這些需求的各種選項時，我們發現最常見的做法是編寫特定的內部腳本，沒有可用的系統化工具，即使是針對單一數據源。因此，我們設計了一個可擴展且可重新配置的 ETL 工具，名為 Table2Graph。通過這個工具，數據實踐者可以指定源數據庫中數據元素與圖模型之間的必要映射，然後將自動轉換為用於實例化圖的 MapReduce 代碼。這樣的用戶配置存儲在 XML 文件中，任何未來的更改都可以直接應用於該文件。

本文的其餘部分安排如下。第二節簡要概述 NoSQL 數據庫和圖轉換過程。第三節詳細描述 Table2Graph。第四節報告了對 Table2Graph 性能的實證研究。最後，第五節總結了本文，討論了未來的方向。

解析與重點：

- **數據整合問題**：
    1. 傳統 ETL 流程的局限性：難以修改設計、模式變化需要重新實現、擴展性差
    2. 在不同數據源間識別關聯的挑戰
- **實際應用背景**：
    1. 患者保護與平價醫療法案(PPACA)第 6402(a)條要求
    2. 需要整合 Medicare、Medicaid 等多個醫療數據系統進行匹配
- **解決方案設計思路**：
    1. 採用異構圖模型代替傳統關係型數據庫模型
    2. 通過語義連接而非合併來整合多數據源
    3. 圖模型直接嵌入連接，提高查詢效率
- **Table2Graph 工具特點**：
    1. 可擴展性：處理數十億節點和邊
    2. 可重新配置：通過 XML 配置文件實現映射
    3. 基於 MapReduce：自動生成代碼用於實例化圖
- **解決的關鍵挑戰**：
    1. 處理大規模數據
    2. 應對數據源模式變化
    3. 替代特定內部腳本的系統化工具

II. RELATED WORK（相關工作）

翻譯：

在本節中，我們描述了選擇圖數據庫的決定。特別是，我們簡要概述了新興的數據庫技術，並比較它們在我們任務方面的優缺點。還討論了與 Table2Graph 相關的工作。

A. NoSQL 數據庫技術

對於我們數據整合的目的 - 匹配醫療服務提供者，如上一節簡要說明的那樣，關係數據模型被發現主要是由於其在執行關係相關查詢方面的不適當性、不靈活的模式和可擴展性問題而不適合。在尋找替代數據模型時，我們調查了各種新興的 NoSQL 數據庫，如列導向存儲、文檔導向數據庫和圖數據庫。

列導向數據庫通過利用列中重複元素可以被總結的觀察來幫助加速計算。此外，這些數據庫提供了高級數據壓縮的機會。然而，建立在關係模型上，它在適應和更改數據模式方面不太靈活。此外，關係搜索仍然依賴於傳統的多表間連接操作來提取數據集之間的間接連接。這種約束增加了列存儲架構在存儲和探索可能隔離幾個表的間接關係時的複雜性。

與關係數據庫的嚴格模式相反，文檔導向數據庫圍繞文檔的較少限制概念操作，可視為靈活模式。因此，這種模型適合創建非剛性數據模型。強調存儲可靠性和規模，一些文檔導向數據庫也建立在 Hadoop 上的 MapReduce 上。我們發現文檔導向數據庫是整合不同數據源的良好候選，因為它是靈活和可擴展的。然而，匹配實體的概念與關係模型中的相同。

圖數據庫設計用於支持圖形查詢，如計算節點之間的最短路徑或檢測圖中的社區。它不強制嚴格的模式要求，而是以無模式方式存儲數據和關係。它在建模和識別數據集之間的關聯方面本質上更快。由於它不需要昂貴的連接操作，可以在分佈式數據框架（如 Hadoop）上實例化，它自然地擴展。無模式特性允許通過簡單引入新節點和邊來方便地將新數據類型或實體插入模型。我們發現圖數據庫最適合我們目的的數據整合和後續分析。

B. 關聯到圖數據轉換

存在幾種將關係數據轉換為圖數據的現有方法。Fong 等人和 Fernandez 等人介紹了將關係數據轉換為半結構化 XML 文檔的方法。我們可以認為這些方法處理的是一種特定類型的圖，因為 XML 文檔是根方向連接圖。在語義網研究領域，隨著開放連接數據的出現，有許多工作和工具旨在使關係數據作為 RDF（資源描述框架）表示可用。RDF 是網絡數據交換的標准語言，基於將資源表示為

RDF 三元組的理念，每個三元組由主語、謂語和賓語組成。RDF 數據集是一個圖，其中每個三元組在數據集中可以表示為一個主語節點、一個謂語邊和一個賓語節點。R2O 是一種用於映射關係數據庫模式和在 RDFS 或 OWL 中實現的本體論的語言。D2RQ 旨在直接將非 RDF 關係數據庫作為虛擬 RDF 圖可用。Virtuoso RDF Views 是另一種聲明性語言，用於定義 SQL 數據到預先存在的 RDF 詞彙的映射，旨在將 SQL SELECT 查詢的結果集轉換為一組三元組。Triplify 將 HTTP-URI 請求的結果關係轉換為 RDF 聲明。在這些現有研究中，關係表中的每個元組都轉換為一個節點，每個外鍵-主鍵連接都轉換為相應節點之間的定向邊。由於方法的簡單性，這種方法構建的圖無法捕捉關係數據庫中暗示的所有語義；例如，這種方法不會創建任何帶屬性的邊。請注意，在現實世界應用中可能存在各種帶屬性的關係。此外，雖然 RDF 被認可為通用數據模型，但它的過於簡單性常常導致具有大量節點和邊的複雜圖結構。

最近，Virgilio 等人提出了一種將關係數據庫轉換為使用屬性圖的圖數據庫的方法。屬性圖之所以值得注意，是因為它足夠表達覆蓋許多現實世界應用，並在最先進的通用圖數據庫如 Neo4J、Titan 或 DEX 中被利用。作者專注於通過基於可連接元組聚合構建圖結構來加速處理構建圖上的查詢。我們注意到，該方法有兩個主要限制。首先，作者只考慮了屬性圖的簡化版本，其中只有節點有屬性，而邊只有表示節點之間關係的標籤。其次，它總是確定性地將一個源關係數據庫轉換為一個特定結構的圖數據庫；然而，在現實世界中，根據不同的分析或所需查詢，即使對同一源關係數據庫，我們想要創建的圖結構也可能有所不同。

在本文中，我們也像[22]那樣採用屬性圖模型，但我們允許節點和邊都擁有一組鍵值對，稱為屬性。[23]展示了一項利用 Map-Reduce 框架和 Hadoop 進行可擴展圖構建的努力。然而，在這種方法中，用戶需要在 Map 函數中編寫特定應用的解析器和例程，這對數據庫管理員來說是個負擔。使用 Table2Graph 系統，不需要用戶編程努力，它只需要用戶努力編寫源數據庫和目標數據庫之間的映射，這可以在不需要太多複雜編程或查詢語言知識的情況下完成。

解析與重點：

- **NoSQL 數據庫技術評估**：
    1. **列導向數據庫**：
        - 優點：加速計算、支持數據壓縮

        - 缺點：模式不靈活、關係搜索依賴多表連接
    2. **文檔導向數據庫**：
        - 優點：靈活模式、存儲可靠性、可擴展性
        - 缺點：實體匹配概念與關係模型相似
    3. **圖數據庫**（選擇的解決方案）：
        - 優點：支持圖查詢、無模式存儲、關聯識別快速、不需要昂貴連接操作、易於添加新數據類型
        - 結論：最適合數據整合和後續分析
- **關聯到圖數據轉換研究**：
    1. **XML 轉換方法**：將關係數據轉換為半結構化 XML 文檔
    2. **RDF 相關工作**：
        - 多種工具將關係數據轉為 RDF 表示(如 R2O、D2RQ 等)
        - 常見做法：元組轉節點，外鍵-主鍵連接轉定向邊
        - 缺點：無法捕捉所有語義，特別是帶屬性的邊；RDF 過於簡單導致複雜結構
    3. **屬性圖相關工作**：
        - Virgilio 等人的方法：使用屬性圖但限制為只有節點有屬性
        - 限制：固定轉換模式，不允許根據需求變化
    4. **Table2Graph 的差異化**：
        - 採用屬性圖但允許節點和邊都有屬性
        - 利用 Map-Reduce 框架但無需用戶編寫複雜程序
        - 通過簡單映射配置實現轉換

III. TABLE2GRAPH: SYSTEM DESIGN（系統設計）

翻譯： 本節描述了 Table2Graph 的設計。在眾多圖模型變體中，我們選擇了屬性圖模型，因為它可以通過屬性保留關於節點或邊的豐富信息集。事實上，這

個模型被許多最先進的通用圖數據庫如 Neo4J [9]、DEX [11]和 Titan [10]所採用。屬性圖是一個有向多重圖 $G = (V, E, TV, TE)$，其中 V 是有限的節點集，$E \subseteq V \times V$ 是有限的邊多重集，TV 是有限的節點類型集，TE 是有限的邊類型集。每個節點通過映射函數 $\phi V: V \rightarrow TV$ 映射到一個節點類型，每個邊通過另一個映射函數 $\phi E: E \rightarrow TE$ 映射到一個邊類型。

一個節點 $vi \in V$ 或一個邊 $ek(vi,vj) \in E$ 有一組<屬性，值>對，構成屬性。然後，屬性圖 G 的模式，一個節點和邊類型級別的模板，被定義為一個有向圖 $GS=(TV, TE)$，其中 TV 是有限的節點類型集，$TE \subseteq TV \times TV$ 是有限的邊類型集。類似於關係模型的模式，圖模型的模式定義了圖實例化所遵循的類型和屬性。

如圖 1 所示，Table2Graph 分為兩個步驟：模式映射和數據轉換。在前者中，用戶創建從源關係表模式到目標屬性圖模式的映射，而後者將源表的元組轉換為按照映射指定的圖的節點和邊。

A. 模式映射

模式映射步驟設計為允許靈活配置和編輯。通過這個能力，數據實踐者可以為一個圖模型創建映射，並通過最少的手動工作為另一個模型重用相同的映射。從數據源到目標模型的映射存儲在 XML 文檔中（參見圖 2(a)）。節點和邊的映射是分開完成的。對於節點，這涉及識別源模式的一個屬性（或一組屬性）轉換為目標節點，其他包含在目標節點的屬性，以及註釋節點的標籤。我們特別將這些屬性和標籤表示為 key、property 和 label。同樣，邊模式映射包括 key、property 和 label。此外，還包括一對 from_key 和 to_key，用於指定邊的方向，即連接節點（起始和目的地）的 key。此外，節點和邊映射都可以被限制，以限制源數據流入目標模型。

B. 數據轉換

數據轉換步驟執行兩個階段的多個 Map-Reduce 作業。在第一階段，多個 Map-Reduce 作業集體創建臨時節點和邊，我們稱之為節點和邊組件。在此階段，使用統一資源標識符（URI）唯一識別節點或邊。隨後，在第二階段，這些具有相同 URI 的節點或邊組件被合併，生成唯一的節點和邊。圖 4 顯示了數據轉換的詳細 Map-Reduce 工作流。

數據轉換中的 Map-Reduce 作業分為以下四種類型之一：

- 節點組件構造器，N

- 邊組件構造器，E
- 節點集合合併器，MN
- 邊集合合併器，ME

在詳細描述這些作業類型（或操作符）之前，讓我們定義幾個在所有操作符中常用的實用函數。首先，getURI(value1, ...)是一個哈希函數，給定任意數量的輸入，返回唯一的 URI。mergeNodeComponent(v1, v2, ...)是一個接受任意數量節點組件作為輸入，返回合併輸入組件的唯一節點的函數。這裡所有輸入節點組件 v1, v2, ...具有相同的 URI。mergeNodeComponent(·)的一個重要方面是合併所有輸入組件的屬性。如果輸入節點的相同屬性存在不同值，則根據用戶選項，這些值要麼合併為一個值（例如數組），要麼選擇一個值。同樣，mergeEdgeComponent(e1, e2, ...)返回一個唯一的邊，作為合併具有相同 URI 的邊組件的結果。

算法 1 顯示了節點組件構造器 N 的 Map-Reduce 函數。它接受五個輸入參數 R, U, P, ϕ 和 l。R 是源關係，U 是 URI 屬性集，P 是節點屬性屬性集。ϕ 是由條件元素和邏輯運算符∨（與）、∧（或）和¬（否定）組成的命題公式，l 是節點類型標籤。然後，它通過轉換關係 R 返回一組 l 節點組件。生成的節點的 URI 將基於 U 中屬性的值分配。每個節點都有屬性，換句話說，屬性在 P 中。請注意，映射配置文件中的 Mapping 元素可以解釋為算法 1 的輸入。U 從'key properties' Map 獲得，P 從'properties' Map 獲得，l 從'node_type_label' Map 獲得，ϕ 從'condition' Map 獲得。

[此處略過算法 1 的具體代碼，重點是描述了如何從關聯表中構建節點]

類似於節點組件構造器，邊組件構造器 E 接受以下參數作為輸入。R 是源關係，U 是 URI 屬性集，P 是邊屬性屬性集，ϕ 是命題公式，l 是邊類型標籤。lf rom 是起始節點的節點類型標籤，lto 是目的節點的節點類型標籤。Uf rom 和 Uto 是 URI 屬性，用於生成起始和目的節點的 URI。算法 2 顯示了邊構造器 E 如何為給定參數生成邊。邊映射文檔中的 Mapping 元素可以解釋為算法 2 的輸入。P 從'properties' Map 獲得，l 從'edge_type_label' Map 獲得，ϕ 從'condition' Map 獲得。lf rom、lto、Uf rom、Uto 分別從相應的'from_node_type_label' Map、'to_node_type_label' Map、'from_key_properties' Map 和'to_key_properties' Map 獲得。

[此處略過算法 2 的具體代碼，重點是描述了如何從關聯表中構建邊]

組件合併提供了一種從多個源表單獨構建節點和邊，然後合併構建完整節點或邊的方法。讓我們考慮圖 4(a)中的例子，其中人員信息分別存儲在 PERSON 和 ADDRESS 表中。從多個表創建一類節點或邊的簡單方法是通過關係數據庫內的連接操作，然後對大型連接表進行模式映射步驟和數據轉換步驟（圖 4(a)）。這種方法的限制在於，對於大型關係表，連接可能非常昂貴。利用 Map-Reduce 執行此類連接操作對於擴展圖構建過程很有用（圖 4(b)）。

如圖 4(b)所示，節點構造器（算法 2）和邊構造器（算法 2）的 Map-Reduce 作業可以針對每個表執行一次或多次。每次執行 Map-Reduce 作業將產生一組<key, v>對或一組<key, e>對，其中 v 是節點組件，e 是邊組件。因此，可能有多個臨時節點集或邊集作為中間結果。節點集合合併器（算法 3）MN(V)合併第一階段中產生的所有節點組件集，返回最終的節點集。邊集合合併器（算法 4）ME(E)對邊組件集做同樣的事情。

[略過算法 3 和算法 4 的具體代碼，重點是描述了如何合併節點和邊組件]

解析與重點：

- **系統設計概述**：
    1. **屬性圖模型選擇**：
        - 允許節點和邊都有屬性（鍵值對）
        - 與 Neo4J、DEX、Titan 等先進圖數據庫使用相同模型
        - 形式定義：G = (V, E, TV, TE)
    2. **Table2Graph 的兩個主要步驟**：
        - 模式映射：用戶定義源關聯表到目標圖的映射
        - 數據轉換：將元組轉為節點和邊
- **模式映射設計**：
    1. 使用 XML 文檔存儲配置
    2. 節點映射包含：key、property、label
    3. 邊映射包含：key、property、label、from_key、to_key

4. 支持限制條件以控制數據流

- **數據轉換過程**：
    1. **兩階段 Map-Reduce 工作流**：
        - 第一階段：創建臨時節點和邊組件
        - 第二階段：合併相同 URI 的組件
    2. **四種 Map-Reduce 作業類型**：
        - 節點組件構造器(N)：從關系表創建節點
        - 邊組件構造器(E)：從關系表創建邊
        - 節點集合合併器(MN)：合併節點組件
        - 邊集合合併器(ME)：合併邊組件
    3. **關鍵算法功能**：
        - getURI()：生成唯一標識符
        - mergeNodeComponent()：合併相同 URI 的節點組件
        - mergeEdgeComponent()：合併相同 URI 的邊組件
    4. **處理多源表的策略**：
        - 避免昂貴的關聯表連接操作
        - 使用 Map-Reduce 框架分別處理各表後合併

第 IV 节：实验 (EXPERIMENTS)

实验设计与目的

Table2Graph 设计的目的是能够大规模地将源关系型数据集转换为整合的图模型。为了展示其可扩展性，研究人员测量了在使用不同计算资源规模时构建整合图模型所需的时间。实验旨在证明随着使用更多资源，Table2Graph 的性能可以按预期增长。

研究团队使用亚马逊网络服务(AWS)公共云进行实验，测试了 1、2、4 和 8 个节点的集群规模。所有实验使用的是 m2.2xlarge 实例类型，每个实例具有 4 个

CPU、34.2GB 内存和 850GB 本地文件 I/O 存储。实验环境基于 Hadoop 2.4.0，并配置为能够同时运行所有独立的 MapReduce 任务，以最大化转换过程的并行性。

数据集与应用场景

研究团队将 Table2Graph 应用于整合三个数据库到一个合并的图模型中：

1. NPPES (National Plan & Provider Enumeration System) - 医疗保险数据
2. PECOS (Provider Enrollment, Chain, and Ownership System) - 医疗保险数据
3. TMSIS (Transformed Medicaid Statistical Information System) - 来自德克萨斯州的医疗补助数据

实验图 5 简化展示了从这三个数据源构建的模型结构，图中只介绍了少量节点和边以便说明。

实验结果与分析

由于 PECOS 和 TMSIS 数据集包含受保护的健康信息，不能放在亚马逊 Web 服务等公共计算设施中，且项目到期后必须丢弃这些数据集，所以实验仅使用了 NPPES 数据集。

NPPES 是 CMS(医疗保险和医疗补助服务中心)每月发布的医疗服务提供者数据集，包含服务提供者的人口统计特征，如地址、性别、分类法、其他标识符等。实验使用的是 2013 年 6 月版本，数据集大小为 5.07GB，包含 4,181,747 条记录。

图 6 展示了各个集群设置的构建时间。构建的图包含 16,543,479 个节点和 48,415,932 条边。使用 8 个工作节点时，构建速度比单个工作节点快 3.41 倍。

![图 6：NPPES 数据集的图构建时间]

这一实验证实了 Table2Graph 可以在合理的处理时间内构建大规模图，并且随着工作节点数量的增加，处理时间可以显著减少。

第 V 节：结论 (CONCLUSIONS)

图模型的广泛应用

图模型因其快速关联数据集处理能力而在广泛领域中使用，例如：

- 电力网络
- 交通路线
- 疾病爆发路径
- 科学研究间的关系

本文描述了图模型的另一个应用领域：链接不同医疗服务提供者的数据源以进行匹配。尽管有许多高效的图分析算法可用于这一任务，但在数据源和数据分析系统之间仍存在巨大差距，即将源数据转换为整合的图模型。

Table2Graph 的贡献与特点

为解决这一问题，研究团队介绍了 Table2Graph，这是一个设计用于促进将多个关系数据表转换为图模型的 ETL(提取、转换、加载)过程的工具。基于 Hadoop 上的 MapReduce 框架，Table2Graph 设计为使用商用计算资源处理大量数据。初步实证性能研究证明了其可扩展性。

Table2Graph 还具有高度可配置性和适应性，可以轻松应用于任意领域。

未来工作方向

为进一步扩展 Table2Graph 的可访问性和可用性，研究团队正在开展以下工作：

1. **开发半自动架构映射**：
    - 当前版本的 Table2Graph 要求用户手动编写 XML 文档
    - 未来工作将通过文本挖掘和数据分布分析来识别数据集之间的语义相似性
    - 这些功能将通过交互式图形用户界面(GUI)向用户提供推荐映射
    - 这将极大地帮助那些对数据库理解较少的领域专家
2. **采用内存大规模数据处理系统**：
    - 现代内存大规模数据处理系统(如 Apache Spark)正获得学术界和工业界的广泛关注

- 这些技术将大大减少 ETL 处理时间
- 研究团队正在开发 Table2Graph 的新版本，以利用这种内存分布式计算技术的优势

总结

Table2Graph 是一个创新性工具，它解决了将关系型数据转换为图模型的复杂问题。通过其可扩展的设计和灵活的配置能力，它为数据集成和图分析提供了强大的基础。实验结果表明，随着计算资源的增加，其性能显著提升，证明其在处理大规模数据时的有效性。未来的改进将进一步提高其易用性和处理效率。