InfoNetOLAPer: Integrating InfoNetWarehouse and InfoNetCube with InfoNetOLAP

Chuan Li[1,2], Philip S. Yu[2], Lei Zhao[3], Yan Xie[2], and Wangqun Lin[2]
[1]Sichuan University, China [2]University of Illinois at Chicago, USA [3]Univ. of Science and Tech. of China
lcharles@scu.edu.cn {lcharles,psyu,yxie,wl}@uic.edu leizhao8@mail.ustc.edu.cn

ABSTRACT

To support efficient graph OLAP operations on information networks, we propose two significant intermediate infrastructures: InfoNetWarehouse and InfoNetCube. InfoNetWarehouse is designed with novelty to be the warehouse model for information networks, which provides topic-oriented, integrated, and multi-dimensional organizational solutions for Information networks. InfoNetCube is our proposed datacube implemention that serves the OLAP of information networks. We further integrate the two infrastructures with InfoNetOLAP module into a prototype called InfoNetOLAPer, which has the following noteworthy features: (1) The basic InfoNetWarehouse schema is well implemented based on SQL Server 2000, (2) InfoNetCube improves the efficiency of InfoNetOLAP by the pre-computation of InfoNetLattice, and (3) InfoNetOLAPer supports efficient I-OLAP and T-OLAP operations.

1. INTRODUCTION

Graph OLAP operations provide multi-dimensional and multi-level view of Information Networks (InfoNetworks) [1] and thus have received growing research interests [2, 3]. With the continuous accumulation and increasing prevalence of Information Networks, OLAP and mining of InfoNetworks have become one of the new research frontiers.

Co-author Network is a typical example of InfoNetwork based on bibliographical datasets including authors, publications, etc. The main focus of Co-author Network is the co-authorship between different authors as shown in Figure 1.

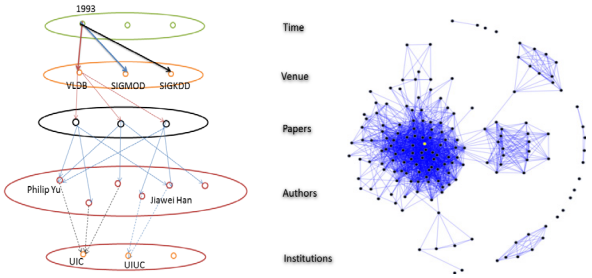


Figure 1. Bibliographical Dataset and Co-author Network.

Proceedings of the VLDB Endowment, Vol. 4, No. 12

Two different types of InfoNetworks OLAP (InfoNetOLAP) operations are introduced in [1]: Informational OLAP (I-OLAP) and Topological OLAP (T-OLAP). T-OLAP changes the topological structure of InfoNetwork by merging nodes, whereas I-OLAP is like overlaying multiple pieces of InfoNetworks without changing the topological structure. The informational dimensions of Co-author Network include [1]:

- *Venue*: *conference* → *area* → *all*,
- *Time*: *year* → *decade* → *all*;

And the topological dimension includes attribute:

- *Background*: *person* →*institution*→ *all*

By performing InfoNetOLAP operations such as roll-up, drill-down and slice/dice, we can observe the evolution of co-authorship from different multi-dimensional and multi-level views as shown in Figure 11.

Despite the research attentions on InfoNetOLAP framework [2], and efficient topological InfoNetOLAP algorithm design [3], a much more fundamental issue concerning the design of the organization infrastructure of InfoNetworks has not been addressed. Not only InfoNetOLAP, but also most other InfoNetwork analyses have long suffered from the current rather inefficient ways to explore data storage of InfoNetwork information.

Firstly, current organizational structures tend to be too simplistic with most InfoNetwork data stored just in XML or TXT files. In addition, different institutions or researchers may have different designs for the coverage, format, measure, or presentation of the data. For example, there are different versions of DBLP datasets for different usages [4]. Therefore, in order to conduct InfoNetOLAP, designers must dig into these files to select the appropriate features and representations, write programs to extract them into task-relevant dataset, and build InfoNetworks manually. This is a time and resource consuming process as shown in Figure 2.

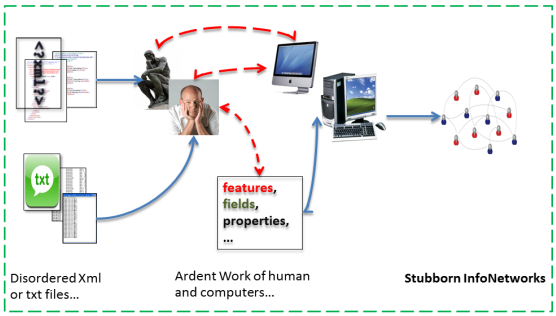


Figure 2. Manual InfoNetwork Construction Process.

Secondly, current techniques do not provide even the basic application flexibility. Take the Bibliographic Information

Networks as example [5], when users need to change their focus from one topic like Co-authorship analysis to another, say, Co-keywordship analysis, a different extraction program is required to generate a new InfoNetwork (for Co-keywordship analysis), and at most cases, the related InfoNetOLAP programs need to be rewritten too because of their tight dependency on the source data.

Thirdly, the efficient maintenance of the intermediate InfoNetOLAP results has not been studied. Therefore, for each request, a seperate round of data extraction, transformation and InfoNetOLAP needs to be carried out. Moreover, for the high level cuboids, a complete round of base cuboid calculation is inevitable, which is an extremely time consuming bottleneck.

Limitations of Existing InfoNetwork Data Organization. In summary, the following data organization features are urgently needed, but are not supported by the existing solutions.

(1) **Integration**. There needs to be a central uniform infrastructure that integrates all heterogeneous InfoNetwork data files to be analyzed. The infrastructure should be carefully designed to generalize its schemas, unify its measures, and standardize its specifications in precisions and presentation of all fields. Integration leads to generalization.

(2) **Topic-Orientation**. InfoNetworks should be organized in a topic-oriented manner, where the topics are modeled as the center with all related information organized around it. In addition, all the interesting topics should be taken into consideration in the modeling for possible future use.

(3) **Materialization of Intermediate InfoNetOLAP results.** The storage of intermediate results eliminates redundant cuboids calculation and hence guarantees the potential OLAP efficiency.

Proposed Solutions: Integrating InfoNetOLAP with InfoNetWarehouse and InfoNetCube.

To a large extent, the difficulties arise from the giant leap from original data directly to InfoNetOLAP tasks, where we lack a bridge mediating the two stages. The retrospect of evolutionary process of the information system may help us understand where the problem is and where we are as shown in Figure 3.

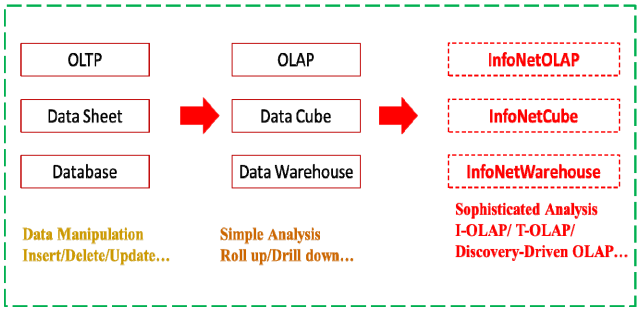


Figure 3. Evolution of Information Systems to InfoNetwork.

Main Contributions In this paper, we propose for the first time two new infrastructures: InfoNetWarehouse and InfoNetCube to support graph OLAP of information networks. We organize and maintain source data in an InfoNetWarehouse, calculate InfoNetCube, and on top of these, we develop our efficient prototype InfoNetOLAPer.

2. INFONETOLAPER ARCHITECTURE

Figure 4 illustrates the system architecture of our demo system InfoNetOLAPer, which is composed of 4 layers. **InfoNetWarehouse Layer** is designed to be responsible for the implementation of conceptual modeling with well-structured storage and maintenance of both topic-oriented measure and dimensional information. **InfoNetCube Layer** works with InfoNetWarehouse layer to calculate the base cuboid and stores it in a compound structure (composed by an orthogonal list for non-empty cuboid cells, a compressed adjacency matrix for cell network, and corresponding external bitmap files for external materialization). **InfoNetOLAP Layer** is efficiently implemented with both I-OLAP and T-OLAP algorithms. **3-D Presentation Layer** utilizes Java 3D and our specially designed 3D visualization algorithms to display InfoNetwork, where users can perform operations like move, rotate, zoom in and out in 3D space.

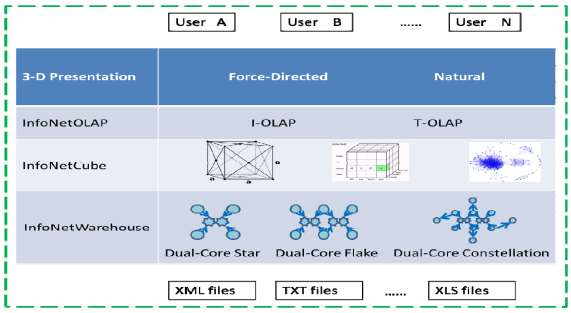


Figure 4. System Architecture of InfoNetOLAPer.

3. TECHNICAL FEATURES

3.1 InfoNetWarehouse

The main objective of InfoNetWarehouse modeling is to organize dimensions and graph measures of InfoNetwork data in a way that it is easy to perform multi-dimensional and multi-level analysis. We propose, for the first time, three typical types InfoNetWarehouse schemas: Dual-Core Star schema, Dual-Core Flake schema, and Dual-Core Constellation schema. Let us take the Co-author Network as a scenario.

Dual-Core Star Schema. Figure 5 shows the Dual-Core Star schema for the Co-author Networks, with *Time* table and *Venue* table to be informational dimension tables (IDT), and *Background* table to be a topological dimension table (TDT). *Papers* table and *Co-authors* tables form the 2 core facts tables, referred to as the frame fact table (FFT) and clique fact table (CFT), respectively. The base cuboid can then be generated by performing simple two-step SQL statements shown below.

Section A : Calculating **BASECUBOID_FRAME**

```
SELECT TIME.YEAR, VENUE.CONFERENCE, PAPERS.PAPER_ID
FROM TIME, VENUE, PAPERS
INTO BASECUBOID_FRAME
WHERE TIME.TID=PAPAERS.TID AND VENUE.VID=PAPERS.VID
```

BASECUBOID_FRAME is the frame of the base cuboid, (just like the cube in Figure 9), where *TID* and *VID* determine the location of the cell to be considered. Since there is a link between every two authors of a paper, every non-empty cell in the base cuboid includes a clique (just like the triangle in Figure 10 (b),

where the nodes represent authors and links represent co-authorships).

Section B: Calculating **BASECUBOID_CLIQUE**

```
SELECT CO-AUTHORS.PAPER_ID, BACKGROUND.PERSON
FROM BASECUBOID_FRAME, CO-AUTHORS, BACKGROUND
INTO BASECUBOID_CLIQUE
WHERE CO-AUTHORS.BID=BACKGOUND.BID AND
      BASECUBOID_FRAME.PAPER_ID=CO-AUTHORS.PAPER_ID
```

BASECUBOID_CLIQUE includes cliques of all the base cuboid cells. *Paper_ID* references to *Co-authors* table and corresponds to a set of authors who then form a clique in a cell. **BASECUBOID_FRAME** and **BASECUBOID_CLIQUE** work together via *Paper_ID*. The level of *BID* here decides the granularity of the graph nodes in each cell.

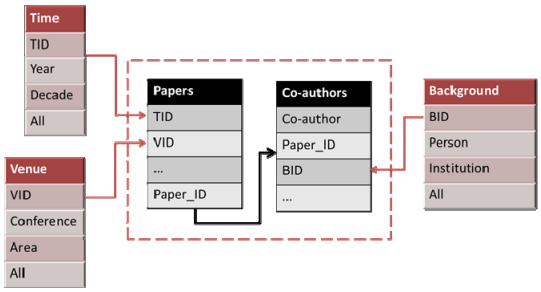


Figure 5. Dual-Core Star Schema for Co-author Network.

To extend the model so that it fits various application contexts, we further propose the generalized structure of Dual-Core Schema as shown in Figure 6, where IDT*1* to IDT*n* represent informational dimension tables, TD*1* to TD*m* represent topological dimension tables, and feature *Co-Interests* works with other facts table features to represent the particular OLAP interest of the InfoNetwork topics (like the co-author frequencies, etc.).

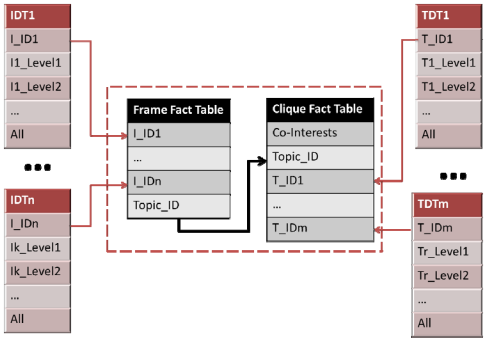


Figure 6. General Dual-Core Star Schema.

Dual-Core Flake Schema. As the InfoNetworks data accumulate, we usually split the dimension tables vertically to reduce the redundant storage and achieve better maintainability. By doing so, we can get Dual-Core Flake schema as shown in Figure 7. Both IDT and TDT can extend their high level features into standalone affiliated dimension tables, e.g., E_IDT*k* for IDT*k* and E_TDT*r* for TDT*r* in Figure 7, where *k* and *r* are integers. Experiences show such split is quite suitable for dimension with more than 6 hierarchical levels.

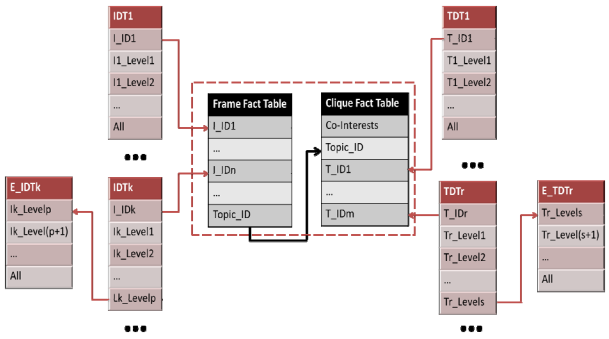


Figure 7. Dual-Core Flake Schema.

Dual-Core Constellation Schema. When there are more than one topic involved in the InfoNetworks dataset, we can use Dual-Core Constellation schema as shown in Figure 8, where Sharing Dimension Tables (short for SDT) are shared between facts table pairs of different topics. When users change the OLAP focus from one topic to another, no extra data preprocessing effort is needed because the InfoNetworks data in both topics are well organized.

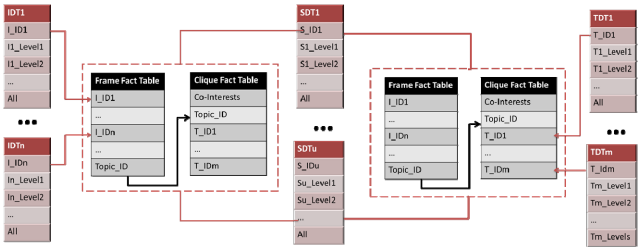


Figure 8. Dual-Core Constellation Schema.

3.2 InfoNetCube

InfoNetCube is different from traditional data cube in that each cell of every cuboid no longer stores the simple numeric measure but a network (graph) as shown in Figure 9.

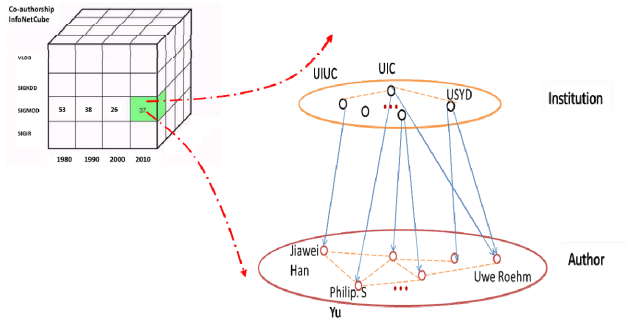
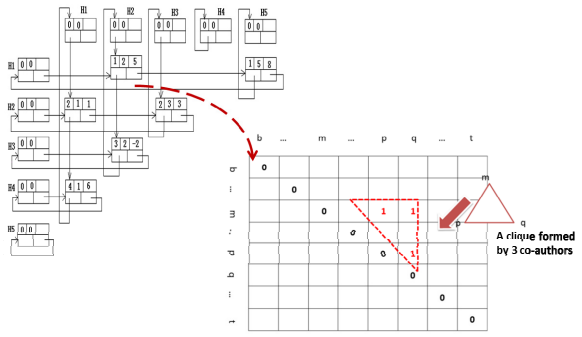


Figure 9. InfoNetCuboid of the Co-author Networks.

Suppose there are two levels, *Author* and *Institution*, in the topological dimension of Co-author Network. The cell network is composed of upper and lower parts. The dotted line denotes the co-authorships between different authors or institutions. And the solid lines connecting two parts represent the roll up paths along with the topological dimension. Like traditional Data Cube, InfoNetCube constitutes a lattice. InfoNetLattice of Co-author Networks is composed of *Author* section and *Institution* section.

3.3 InfoNetOLAP

We utilize the orthogonal list in Figure 10(a) to store the non-empty cells of the base cuboid, where each cell actually contains a clique representing the co-authorship in an adjacency matrix as shown in Figure 10(b) (The 3-person m, p, q triangle on co-authorship is stored in the adjacency matrix as shown). All the cells in other cuboids of higher InfoNetLattice positions are then calculated by aggregating these graphs from the base cuboid.



(a) Orthogonal List (b) Adjacency Matrix

Figure 10. Physical Design of InfoNetCube Cell.

With these two structures, both I-OLAP and T-OLAP algorithms are easy to design and implement. I-OLAP operations can be simply done by accumulating the overlapped adjacency matrixes of the corresponding cuboids (we use roll-up operation here for the purpose of illustration because all other cuboids are obtained through step-by-step rolling up the base cuboid), while T-OLAP operations only involve combining the columns and rows of the adjacent matrixes. Due to space limitation, we omit the implementation details here.

4. SYSTEM DEMONSTRATION

InfoNetOLAPer was implemented with Java 2 sdk 1.6, Microsoft SQL Server 2000 Personal Edition. The demo takes the scenario of Co-author Network for its popularity. InfoNetOLAPer can perform any InforNetwork OLAP operations like roll-up, drill-down, slice, dice, etc. By Java 3D techniques, we can move InfoNetwork, rotate it, zoom it in or out in 3D space. Different from traditional OLAP systems, InfoNetOLAPer is capable of performing graph OLAP operations on information networks.

Performance Discussion We have conducted extensive experiments on various synthetic datasets. Experiments show that at most times on the graph datasets of 6,000 ~ 8,000 vertices, the roll-up and drill-down operations together with 3-D graphics calculations can be done in less than 3 seconds.

4.1 InfoNetOLAPer Operations

The audience will be able to manipulate and see through all the InfoNetWork OLAPing operations in the demo.

Roll-Up and Drill-Down. In Figure 11 we first roll up the topological dimension from *person* to *institution* and then drill down along the informational dimension time from *all* to *decade*. Since InfoNetworks of different decades cannot be seen altogether, we can choose to show them one by one. For example, we may want to look at the *1990s* network here. And after that, we drill down the venue dimension from *All* to *DB*.

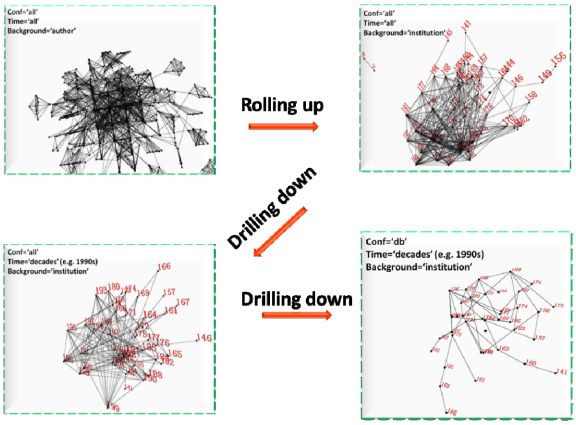


Figure 11. InfoNetwork Evolution via InfoNetOLAP.

Slicing and Dicing. By setting the interesting dimensions values or value ranges, slicing and dicing can be done directly by retrieving a subset of roll-up or drill-down results. Suppose we want to see the situation of hardware conference in year 2000 by authors. We can get the result in Figure 12.

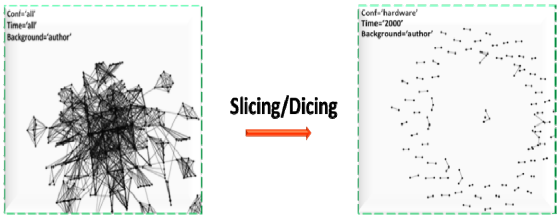


Figure 12. Slicing or Dicing.

5. Acknowledgements

This work is supported in part by NSF through grants IIS 0905215, DBI-0960443, OISE-0968341, OIA-0963278 and Doctoral Site Foundation from Ministry of Education of China through grant No. 20090181120064.

6. REFERENCES

[1] C. Chen, X. Yan, F. Zhu, J. Han, and P. S. Yu, "Graph OLAP: Towards Online Analytical Processing on Graphs", ICDM'08, 103-112, Pisa, Italy, Dec. 2008.

[2] C. Chen, X. Yan, F. Zhu, J. Han, and P. S. Yu, "Graph OLAP: A Multi-Dimensional Framework for Graph Data Analysis", KAIS, Volume 21, Issue 1, 41-63, 2009.

[3] Q. Qu, F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Li, "Efficient Topological OLAP on Information Networks", DASFAA'11, 389-403, Hong Kong, Apr. 2011.

[4] University at Trier http://dblp.uni-trier.de/xml/,

KDL http://kdl.cs.umass.edu/data/dblp/dblp-info.html,

TNAG JIE http://arnetminer.org/citation#b540

[5] Y. Sun, T. Wu, H. Cheng, J. Han, X. Yin, and P. Zhao, "BibNetMiner: Mining Bibliographic Information Networks", SIGMOD'08, 1341-1344, Vancouver, Canada, June 2008.