Querical Data Networks

Cyrus Shahabi
University of Southern California, USA

Farnoush Banaei-Kashani
University of Southern California, USA

INTRODUCTION

Recently, a family of massive self-organizing data networks has emerged. These networks mainly serve as large-scale distributed query-processing systems. We term these networks querical data networks (QDN). A QDN is a federation of a dynamic set of peer, autonomous nodes communicating through a transient-form interconnection. Data is naturally distributed among the QDN nodes in extra-fine grain, where a few data items are dynamically created, collected, and/or stored at each node. Therefore, the network scales linearly to the size of the data set. With a dynamic data set, a dynamic and large set of nodes, and a transient-form communication infrastructure, QDNs should be considered as the new generation of distributed database systems with significantly less constraining assumptions as compared to their ancestors. Peer-to-peer networks (Daswani, Garcia-Molina, & Yang, 2003) and sensor networks (Akyildiz, Su, Sankarasubramaniam, & Cayirci, 2002; Estrin, Govindan, Heidemann, & Kumar, 1999) are well-known examples of QDNs.

QDNs can be categorized as instances of “complex systems” (Bar-Yam, 1997) and studied using the complex system theory. Complex systems are (mostly natural) systems hard (or complex) to describe information-theoretically and hard to analyze computationally. QDNs share the same characteristics with complex systems and, particularly, bear a significant similarity to a dominating subset of complex systems most properly modeled as large-scale interconnection of functionally similar (or peer) entities. The links in the model represent some kind of system-specific entity-to-entity interaction. Social networks, a network of interacting people, and cellular networks, a network of interacting cells, are two instances of such complex systems. With these systems, complex global system behavior (e.g., a social revolution in a society, or food digestion in a stomach!) is an emergent phenomenon, emerging from simple local interactions. Various fields of study, such as sociology, physics, biology, chemistry, etc., were founded to study different types of initially simple systems and have been gradually matured to analyze and describe instances of incrementally more complex systems. An interdisciplinary field of study, the complex system theory, was recently founded based on the observation that analytical and experimental concepts, tools, techniques, and models developed to study an instance of complex system at one field can be adopted, often almost unchanged, to study other complex systems in other fields of study. More importantly, the complex system theory can be considered as a unifying metatheory that explains common characteristics of complex systems. One can extend application of the complex system theory to QDNs by:

1. Adopting models and techniques from a number of impressively similar complex systems to design and analyze QDNs as an instance of engineered complex systems; and
2. Exporting the findings from the study of QDNs (which are engineered, hence, more controllable) to other complex system studies.

This article is organized in two parts. In the first part, we provide an overview, where we (1) define and characterize QDNs as a new family of data networks with common characteristics and applications, and (2) review possible database-like architectures for QDNs as query processing systems and enumerate the most important QDN design principles. In the second part of the article, as the first step toward realizing the vision of QDNs as complex distributed query-processing systems, we focus on a specific problem, namely, the problem of effective data location (or search) for efficient query processing in QDNs. We briefly explain two parallel approaches, both based on techniques/models borrowed from the complex system theory, to address this problem.

BACKGROUND

Here, we enumerate the main componental characteristics and application features of a QDN.
Componental Characteristics

A network is an interconnection of nodes via links, usually modeled as a graph. Nodes of a QDN are often massive in number and bear the following characteristics:

- **Peer functionality**: All nodes are capable of performing a restricted but similar set of tasks in interaction with their peers and the environment, although they might be heterogeneous in terms of their physical resources. For example, joining the network and forwarding search queries are among the essential peer tasks of every node in a peer-to-peer network.

- **Autonomy**: Aside from the peer tasks mentioned above, QDN nodes are autonomous in their behavior. Nodes are either self-governing or governed by out-of-control uncertainties. Therefore, to be efficacious and applicable, the QDN engineering should avoid imposing requirements to and making assumptions about the QDN nodes. For example, strict regulation of connectivity (e.g., enforcing number of connections and/or target of connections) might be an undesirable feature for a QDN design.

- **Intermittent presence**: Nodes may frequently join and leave the network based on their autonomous decision, due to failures, etc.

On the other hand, links in various QDNs stand for different forms of interaction and communication. Links may be physical or logical, and they are fairly inexpensive to rewire. Therefore, a QDN is a large-scale federation of a dynamic set of autonomous peer nodes building a transient-form interconnection. Conventional approaches developed to model and analyze traditional distributed database systems (and classical networks, as their underlying communication infrastructure) are either too weak (oversimplifying) or too complicated (overcomplicated) to be effective with large-scale and topology-transient QDNs. The complex system theory (Bar-Yam, 1997), on the other hand, provides a set of conceptual, experimental, and analytical tools to contemplate, measure, and analyze systems such as QDNs.

Application Features

A QDN is applied as a distributed source of data (a data network) with nodes that are specialized for cooperative query processing and data retrieval. The node cooperation can be as trivial as forwarding the queries or as complicated as in-network data analysis. In order to enable such an application, QDN should support the following features:

- **Data-centric naming, addressing, routing, and storage**: With a QDN, queries are declarative; i.e., query refers to the names of data items and is independent of the location of the data. The data may be replicated and located anywhere in the data network, the data holders are unknown to the querier and are only intermittently present, and the querier is interested in data itself rather than the location of the data. Therefore, naturally QDN nodes should be named and addressed by their data content rather than an identifier in a virtual name space such as the IP address space. Consequently, with data-centric naming and addressing of the QDN nodes (Heidemann et al., 2001), routing (Ratnasamy, Francis, Handley, Karp, & Shenker, 2001) and storage (Ratnasamy et al., 2003) in QDN are also based on the content. It is interesting to note that non-procedural query languages such as SQL also support declarative queries and are appropriate for querying data-centric QDNs.

- **Self-organization for efficient query processing**: QDNs should be organized for efficient query processing. A QDN can be considered as a database system with the data network as the database itself (see the next section). QDN nodes cooperate in processing the queries by retrieving, communicating, and preferably on-the-fly processing of the data distributed across the data network. To achieve efficiency in querying processing with high resource utilization and good performance (e.g., response time, query throughput, etc.), QDNs should be organized appropriately. Examples of organization are: intelligent partitioning of the query to a set of sub-queries to enable parallel processing, or collaborative maintenance of the data catalogue across the QDN nodes. However, the peer tasks of the QDN nodes should be defined such that they self-organize to the appropriate organization. In other words, organization must be a collective behavior that emerges from local interactions among nodes; otherwise, the dynamic nature and large scale of a QDN render any centralized micromanagement of a QDN unscalable and impractical.

VISION: A DATABASE QUERYING FRAMEWORK FOR QDNS

In the previous section, we defined a querical data network (QDN) as a distributed data source and a query