Dynamic Graph Query Support for SDN Management

Ramy Raghavendra
IBM TJ Watson Research Center
Roadmap

SDN scenario 1: Cloud provisioning

Management/Analytics primitives
Current Cloud Offerings

- Limited control of the network
  - Requires integration of third-party solutions
  - Limits the opportunity to migrate production applications

Examples of Missing Features

- No ability to create VLANs in the cloud
- No facility to manage bandwidth or QoS
- Limited ability to craft network segments
- No intelligence for dynamically structured networks

Network as a Service Model

- Provide rich set of services
  - isolation, custom addressing, service differentiation etc.
- Interact with variety of network devices
  - support multiple cloud platforms, device management protocols
- Introduces a network controller
  - network-aware VM placement, QoS support, real-time monitoring, diagnostics, management, security etc.

Cloud Networking-as-a-Service

• **Cloud controller**
  • Provides base IaaS service for managing VM instances and images
  • Self-service provisioning UI
  • Connects VMs via host virtual switches

• **Network controller**
  • Provides VM placement directives to cloud controller
  • Generates virtual network between VMs
  • Configures physical and virtual switches
Prototype

• Cloud Controller: OpenNebula 1.4
  • Modified to accept user-specified network policies
  • Modified to accept placement decisions from Network Controller

• Network Controller: NOX and OpenFlow-enabled switches
  • Network controller implemented as a C++ NOX application (~2500 LOC)
  • HP Procurve 5400 switches w/ OpenFlow 1.0 firmware
Roadmap

SDN scenario 1: Cloud provisioning

Management/Analytics primitives
Graph Queries for Network Management

- Support for efficient queries on data structure containing time-varying graph, e.g., shortest path between two nodes
- Utilized in various network management operations

Network graphs can represent:
- physical network elements such as routers, switches, and servers
- virtual elements such as VMs and virtual switches
- logical elements such as people, processes, web pages, etc.
- and links between them

Algorithm support:
- shortest path
- spanning tree
- min flow
- …
Cloud DCN Graphs Are Dynamic

Cloud DCNs may experience frequent changes
- Deployment of new VMs, removal of old ones
- Migration of VMs
- Layer 2/3 network reconfiguration

Graph updates
- edge/vertex insertion & deletion
- edge weight change (e.g. congestion level)

Customer: this is what my workloads look like

High-level VM & network description

Monitoring, notifications

High-level network description & commands

Cloud controller

Network controller

OpenFlow
NetGraph Software Architecture

*NetGraph: Dynamic Graph Query Primitives for SDN-based Cloud Network Management, HotSDN, Sigcomm 2012*
System G Tools v1.0 Architecture

Visualization
- Huge Network Visualization
- Network Propagation
- I2 3D Network Visualization
- Geo Network Visualization
- Graphical Model Visualization

Analytics
- Communities
- Graph Search
- Network Info Flow
- Bayesian Networks
- Centralities
- Graph Query
- Shortest Paths
- Latent Net Inference
- Ego Net Features
- Graph Matching
- Graph Sampling
- Markov Networks

Middleware
- Graph Processing Interface
- Shared Memory Run Time Library
- Distr. Memory RT Library
- Graphs RDMA
- MPI
- Cluster (BladeCenter, BlueGene)
- Graphs FPGA/HMC
- Infosphere Streams (ISS)
- BigInsights
- Pthreads
- Hadoop
- PERCS Coh. Clus.

Database
- Graph Data Interface
- GBase (update, scan, operators, indexing)
- Native Store
- Netezza
- DB2 RDF
- TinkerPop Compliant DBs
- HBase
- HDFS
- DB2
- DB2 RDF
- Netezza
- TinkerPop Compliant DBs

Hardware
- System G Assets
- Open Source
- IBM Product
- IBM Hardware
Graph Workload Types

- **Type 1: Computations on graph structures / topologies**
  - Example → converting Bayesian network into junction tree, graph traversal (BFS/DFS), etc.
  - Characteristics → Poor locality, irregular memory access, limited numeric operations

- **Type 2: Computations on graphs with rich properties**
  - Example → Belief propagation: diffuse information through a graph using statistical models
  - Characteristics
    - Locality and memory access pattern depend on vertex models
    - Typically a lot of numeric operations
    - Hybrid workload

- **Type 3: Computations on dynamic graphs**
  - Example → streaming graph clustering, incremental k-core, etc.
  - Characteristics
    - Poor locality, irregular memory access
    - Operations to update a model (e.g., cluster, sub-graph)
    - Hybrid workload
System G Analytics Overview

System G is a comprehensive set of graph analytics libraries for IBM Big Data

- **Analytics target at big graphs**
  - Create HBase coprocessors to provide scalable graph analytics by distributing data and computation in a balanced way based on graph topology. They outperform MapReduce-based approaches by 2.8 times.
  - Exploit RDMA and other hardware-based optimization for CPU intensive analytics to achieve high CPU utilization, low IO and good speed up

- **Analytics target at dynamic graphs**
  - A major differentiator compared to competitors like GraphLab (dynamic graph clustering coefficients analytics on System G is up to 21 times faster than GraphLab)
  - Include incremental K core, evolution aware clustering, and streaming graph clustering coefficients
  - Exploit IBM InfoSphere Stream framework
<table>
<thead>
<tr>
<th>Analytics</th>
<th>Exact or Approx.</th>
<th>Graph size and dynamics</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-core</td>
<td>Exact</td>
<td>Tested against &gt; 200M edge graph</td>
</tr>
<tr>
<td>K-neighborhood</td>
<td>Exact</td>
<td>Scale free</td>
</tr>
<tr>
<td>K shortest path</td>
<td>Exact</td>
<td>Scale free</td>
</tr>
<tr>
<td>Connected component</td>
<td>Exact</td>
<td>Tested against &gt; 70M edge graph</td>
</tr>
<tr>
<td>Pagerank</td>
<td>Exact</td>
<td>Tested against &gt; 70M edge graph</td>
</tr>
<tr>
<td>Graph search &amp; recommendation</td>
<td>Exact</td>
<td>Scale free</td>
</tr>
<tr>
<td>Probabilistic Inference in Bayesian Networks</td>
<td>Approx.</td>
<td>Constrained by physical memory</td>
</tr>
<tr>
<td>XRIME scalable community detection</td>
<td>Approx.</td>
<td>Tested against &gt; 5M edge graph</td>
</tr>
<tr>
<td>Snazzy community detection</td>
<td>Exact</td>
<td>Tested against &gt; 1M edge graph</td>
</tr>
<tr>
<td>Dynamic subgraph matching</td>
<td>Exact</td>
<td>Tested against &gt; 150K edge graph. Change up to 30% within 10 sec</td>
</tr>
<tr>
<td>Incremental streaming K-core</td>
<td>Exact</td>
<td>Tested against &gt; 16M edge graph</td>
</tr>
<tr>
<td>Streaming graph clustering</td>
<td>Approx.</td>
<td>Tested against 400K updates/sec</td>
</tr>
<tr>
<td>Streaming graph clustering coefficient</td>
<td>Exact</td>
<td>Tested against &gt; 1.8B edge graph, up to 24M updates/sec</td>
</tr>
</tbody>
</table>

* Cloud Service Placement via Subgraph Matching, ICDE 2014
System G – Graph Data Interface API

• Existing Implementations:
  • GBase – HBase implementation of Graph
  • Native Store – C++ Graph Store, in memory and on disc
  • DB2RDF – Tuned for SPARQL + RDF
  • Non-IBM Graph DBs:
    • TinkerGraph – in memory graph
    • Neo4J
    • Titan (on HBase, BerkeleyDB, Cassandra)
  • More
Sample queries

• What is the average hourly RTT at a router?
• When RTT is high, are the k hop links experiencing high RTT?
• What is the current shortest path between two network endpoints?
  • Is the direct link more expensive than the overlay link?
Ongoing Work

• **Graph + Time series Queries**
  • Graph store coupled with (distributed) time series support
  • What was the shortest path at peak hour yesterday?
  • How do routes vary on a daily basis?

• **Graph + Location Queries**
  • RCA based on client locations, or service locations
Thank You!

Ramya Raghavendra
IBM Research
rraghav@us.ibm.com