

G*studio

and

G*The Dynamic
Graph Database

Winter 2015

Technical Overview



Alan G. Labouseur, Ph.D.

Marist College

3NF Consulting, Inc.

alan.labouseur@Marist.edu

alan@3NFconsulting.com



Summary

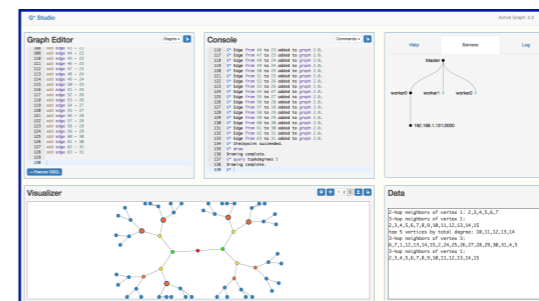
G* is a dynamic graph database system with many features.

- graph distribution
 - multi-core scale up
 - multi-server scale out
- deduplicated disk storage for very large graphs
- in-memory compact indexing
- shared computation
- easy use of sophisticated parallel graph-theoretic queries
- integrates with Relational databases and other stores



Analyzing evolving graphs enables applications in many areas.

- social media analysis
- network traffic threat assessment
- fraud detection
- marketing
- transportation
- epidemiology
- pharmacology
- ...and many other areas



G*studio

Browser Application

The screenshot displays the G* Studio browser application interface, which is divided into several functional panels:

- Graph Editor:** Located at the top left, it contains a list of commands for adding edges between vertices. The commands range from line 108 to 129, each following the format "add edge [source] - [target]". A "Execute GSQL" button is located at the bottom of this panel.
- Console:** Located at the top middle, it shows the execution log of the commands. It displays messages such as "G* Edge from [source] to [target] added to graph 2.0." for each edge addition, followed by "G* Checkpoint succeeded.", "G* draw", "Drawing complete.", and "G* query topkdegrees 5".
- Servers Panel:** Located at the top right, it shows a hierarchical tree structure. At the top is the "Master" node, which is connected to three "worker" nodes: "worker0", "worker1", and "worker2". Below the workers is a server address "192.168.1.131:2000".
- Visualizer:** Located at the bottom left, it displays a network graph visualization. The graph consists of numerous nodes (represented by colored circles) and edges connecting them. The nodes are arranged in a complex, interconnected pattern.
- Data Panel:** Located at the bottom right, it displays the results of a query. The data includes:
 - 2-hop neighbors of vertex 1: 2,3,4,5,6,7
 - 3-hop neighbors of vertex 1: 2,3,4,5,6,7,8,9,10,11,12,13,14,15
 - top 5 vertices by total degree: 10,11,12,13,14
 - 3-hop neighbors of vertex 3: 6,7,1,12,13,14,15,2,24,25,26,27,28,29,30,31,4,5
 - 3-hop neighbors of vertex 1: 2,3,4,5,6,7,8,9,10,11,12,13,14,15



Graph Editor

Graph Editor

```
130
131 -- Evolution: 4 Incremental Graphs (with cloning)
132 create graph 10.0
133 add vertex 1 with attributes (color=black)
134 add vertex 2 with attributes (color=black)
135 add vertex 3 with attributes (color=black)
136 add edge 1-2
137 add edge 2-3
138
139 clone graph 11.0 from 10.0
140 add vertex a with attributes (color=white)
141 add vertex b with attributes (color=white)
142 add vertex c with attributes (color=white)
143 add edge 1-a
144 add edge 1-b
145 add edge 1-c
146
147 clone graph 12.0 from 11.0
148 add vertex d with attributes (color=white)
149 add vertex e with attributes (color=white)
150 add vertex f with attributes (color=white)
151 add edge 2-d
152 add edge 2-e
```

→ Execute GSQL

Graphs ▾



Console

Evolution ▾

Common ▾

8-vertex Full

32-vertex Ring

32-vertex Bipartite (16 pairs)

63-vertex Tree (branch factor = 2)

64-vertex Star

64-vertex 72-edge Erdos-Renyi Random

Other ▾



Interactive Console

Console

```
5 Graph 2.0 :
6   Vertices: 63
7   Edges   : 62
8 Graph 1.0 :
9   Vertices: 4
10  Edges   : 2
11 Graph 0.0 :
12  Vertices: 2
13  Edges   : 1
14 G* create graph 4
15 New graph 4.0 was created.
16 G* add vertex Kirk
17 Vertex Kirk added to graph 4.0.
18 G* add vertex Spock
19 Vertex Spock added to graph 4.0.
20 G* add vertex McCoy
21 Vertex McCoy added to graph 4.0.
22 G* add edge Kirk-Spock
23 Edge from Kirk to Spock added to graph 4.0.
24 G* add edge Kirk-McCoy
25 Edge from Kirk to McCoy added to graph 4.0.
26 G* draw
27 Drawing complete.
28 G* |
```

Commands ▾



Information ▾

Queries ▾

Degree Distribution

Top-k vertices by degree

Top-k vertices with the
largest change in degree
over consecutive graph
snapshot pairs

Visualizer and Data

Visualizer

Help Servers Log

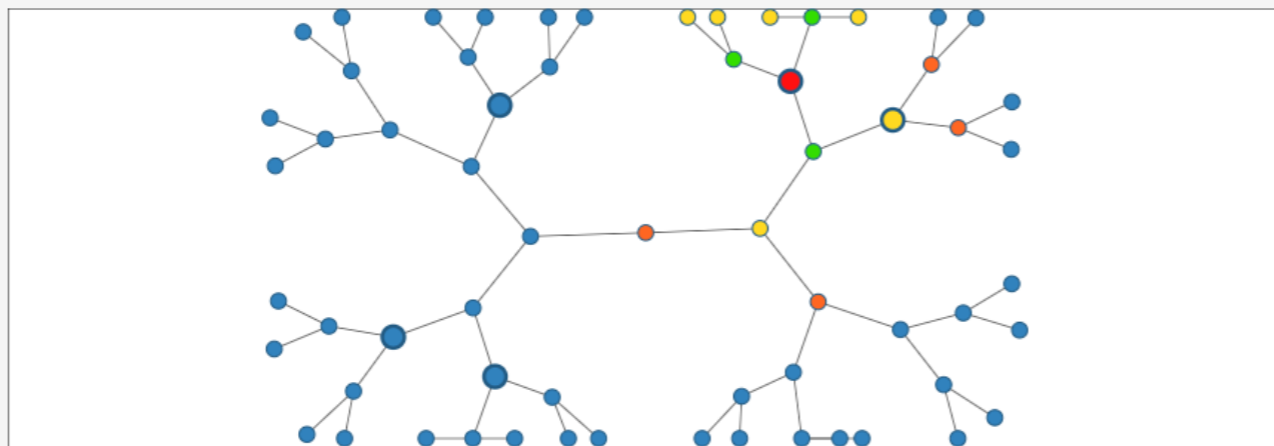
Making Graphs

- add_graph** `<graph-id>`
creates a graph with the given `<graph-id>`
- clone_graph** `<graph-id>` from `<graph-id>`
creates a new graph as a clone of an existing graph
- add_vertex** `<vertex-id>` [with `attributes(<attributeName>=<attributeValue>[,...])`
creates a vertex with id specified by `<vertex-id>` in the active graph. Can optionally add attributes, with one or more attribute pairs.
- add_edge** `<from-vertex-id>` — `<to-vertex-id>`
creates an edge from `<from-vertex-id>` to `<to-vertex-id>` in the active graph.
- update** `<vertex-id>` with `attributes(<attributeName>=<attributeValue>[.])`

Data

3-hop neighbors of vertex 1:
2,3,4,5,6,7,8,9,10,11,12,13,14,15

Visualizer

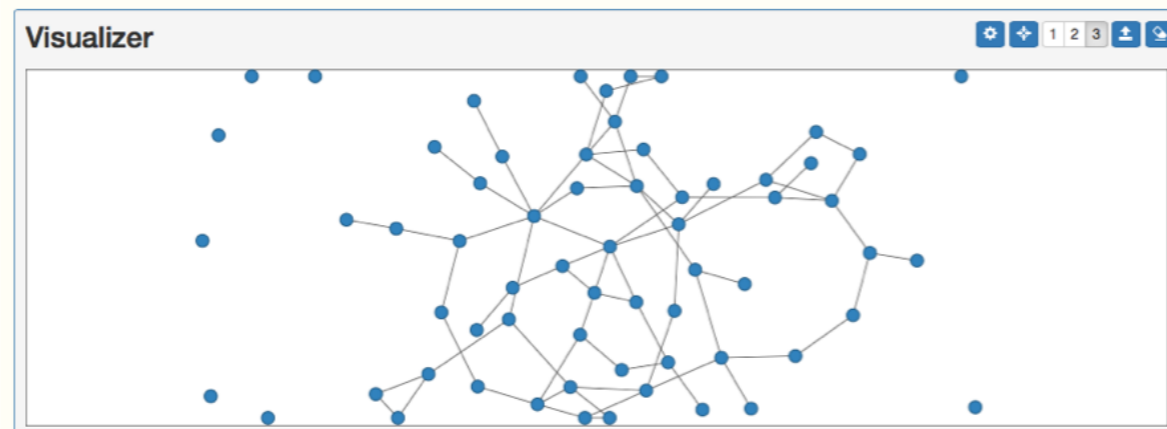


Data

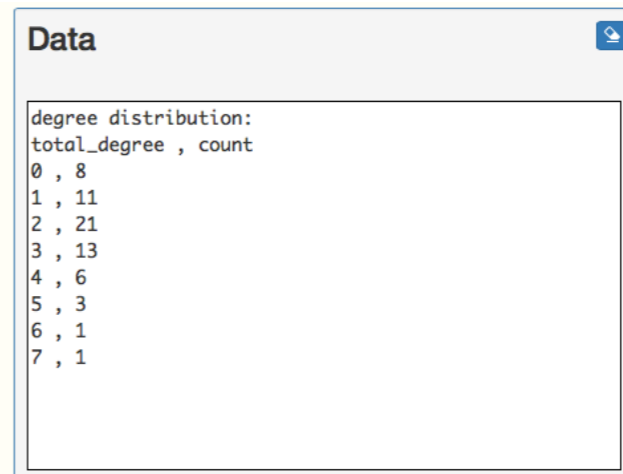
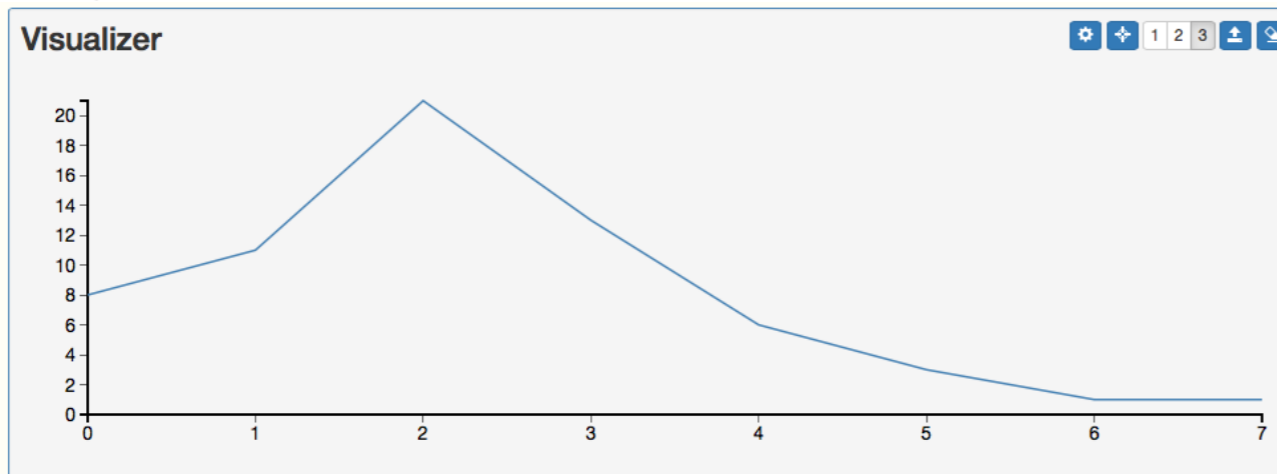
```
top 5 vertices by total degree: 10,11,12,13,14  
3-hop neighbors of vertex 1:  
2,3,4,5,6,7,8,9,10,11,12,13,14,15  
3-hop neighbors of vertex 10:  
20,21,5,40,41,42,43,2,11,4,1,22,23
```

Degree Distribution Query

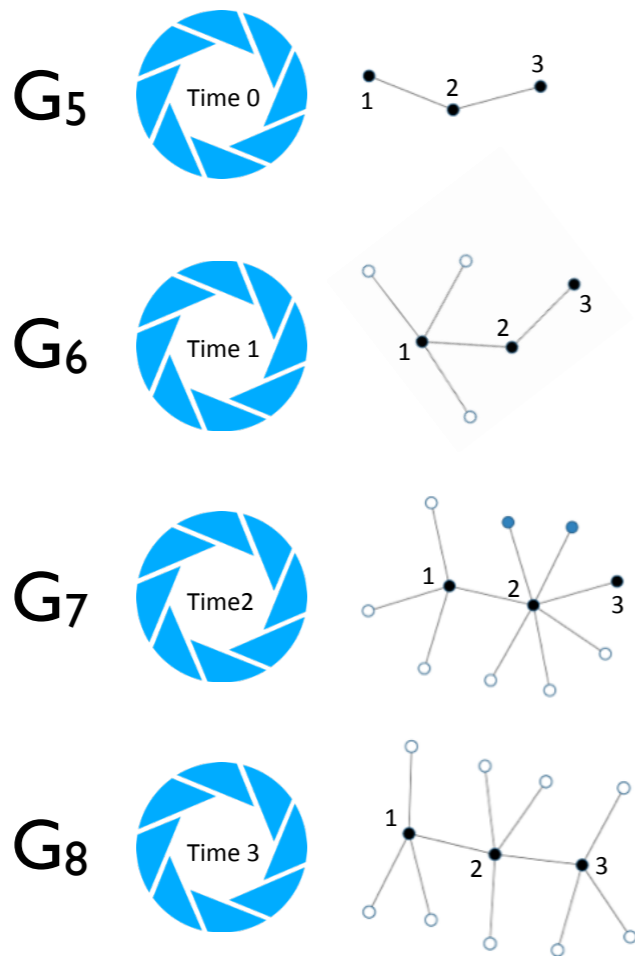
Erdős-Rényi random graph



Degree Distribution



Rising/Falling Stars Query



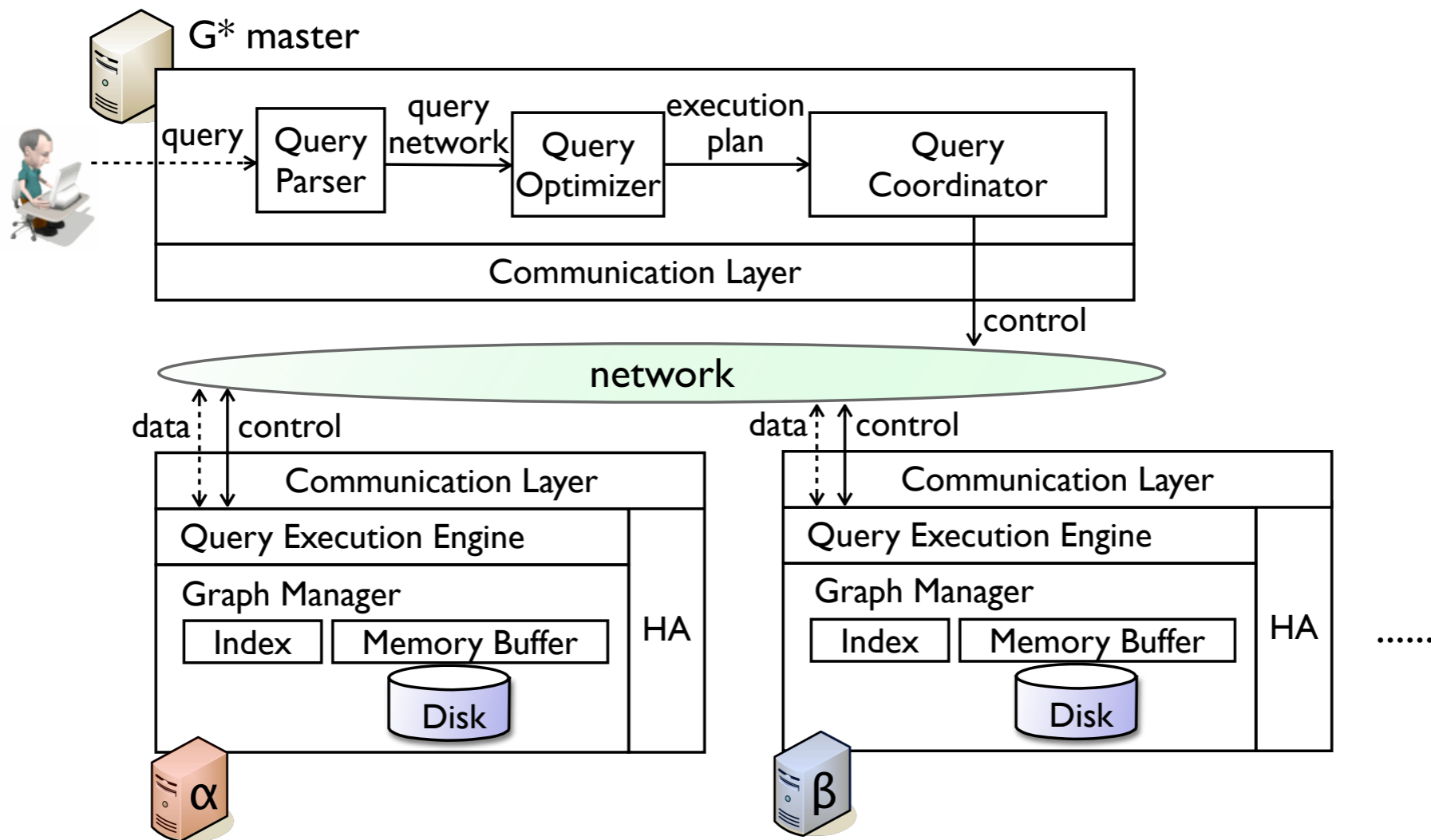
Data

top 20 vertices with the largest change in degree over consecutive graph snapshot pairs from 6 to 8:

```
shapshotPairs , vertexID , change
5->6 , 1 , +3
6->7 , 2 , +5
7->8 , 3 , +3
5->6 , 2 , 0
5->6 , 3 , 0
6->7 , 1 , 0
6->7 , 3 , 0
6->7 , a , 0
.
.
.
7->8 , 2 , -2
```

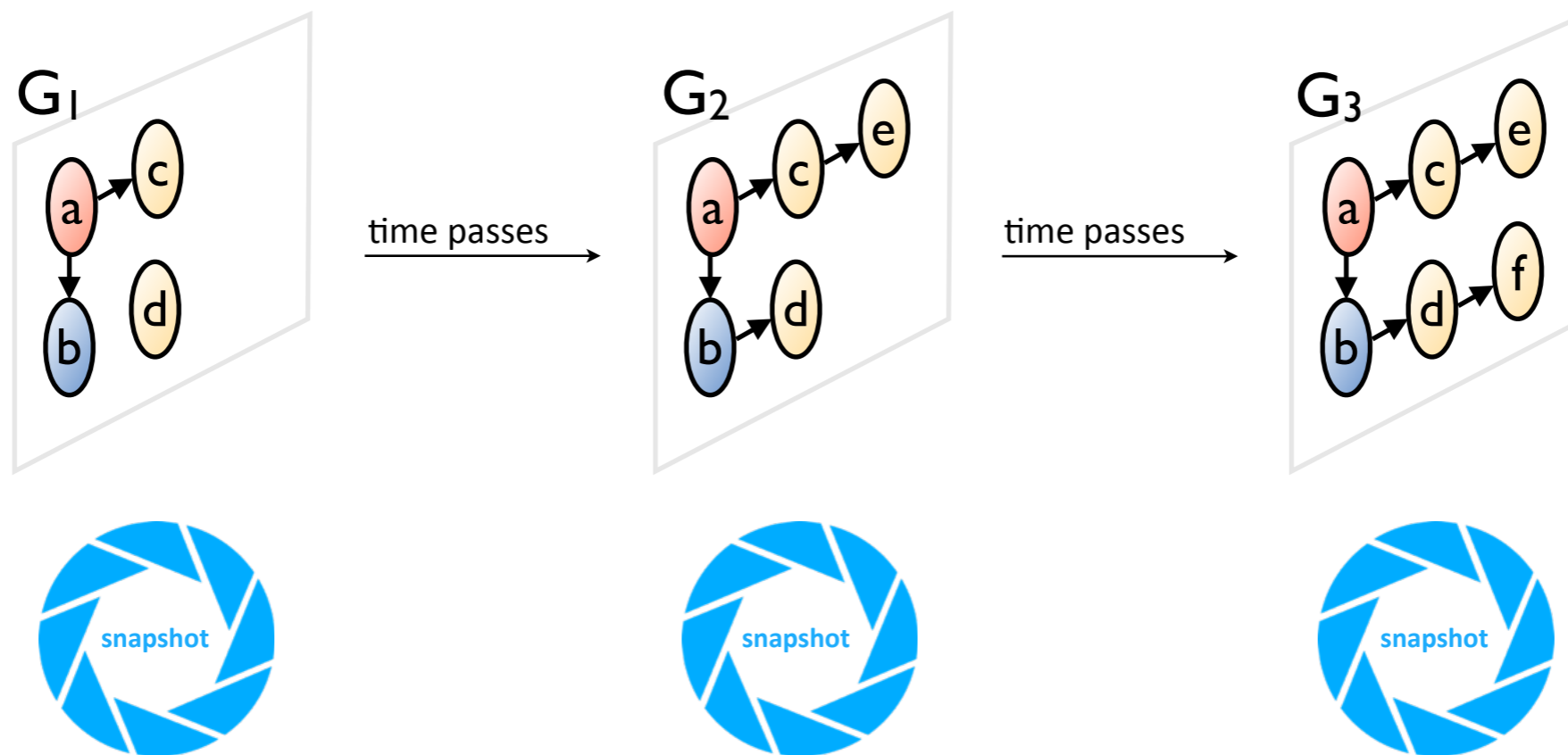
G* The Dynamic Graph Database

Distributed Architecture



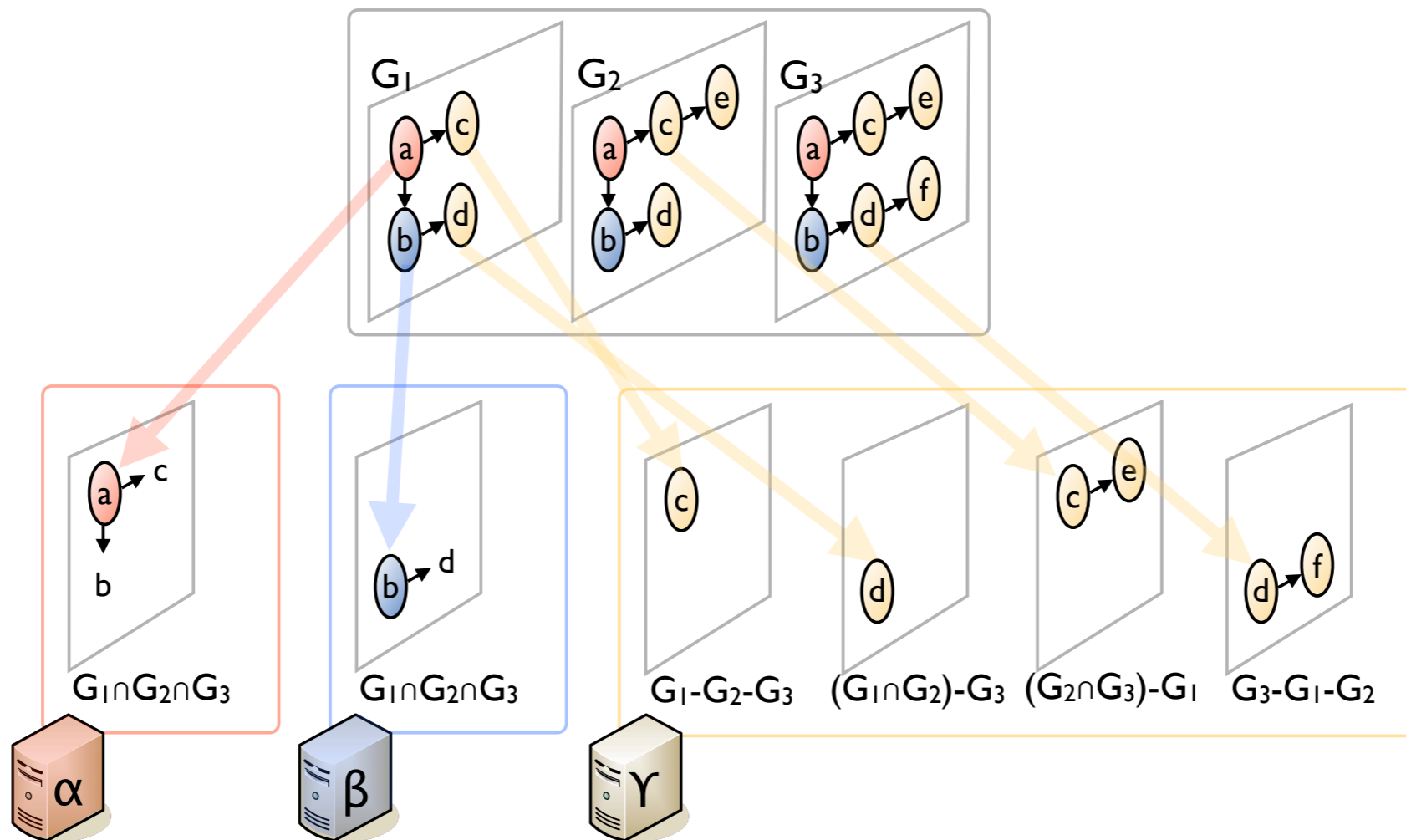
G*The Dynamic Graph Database

Snapshot Management



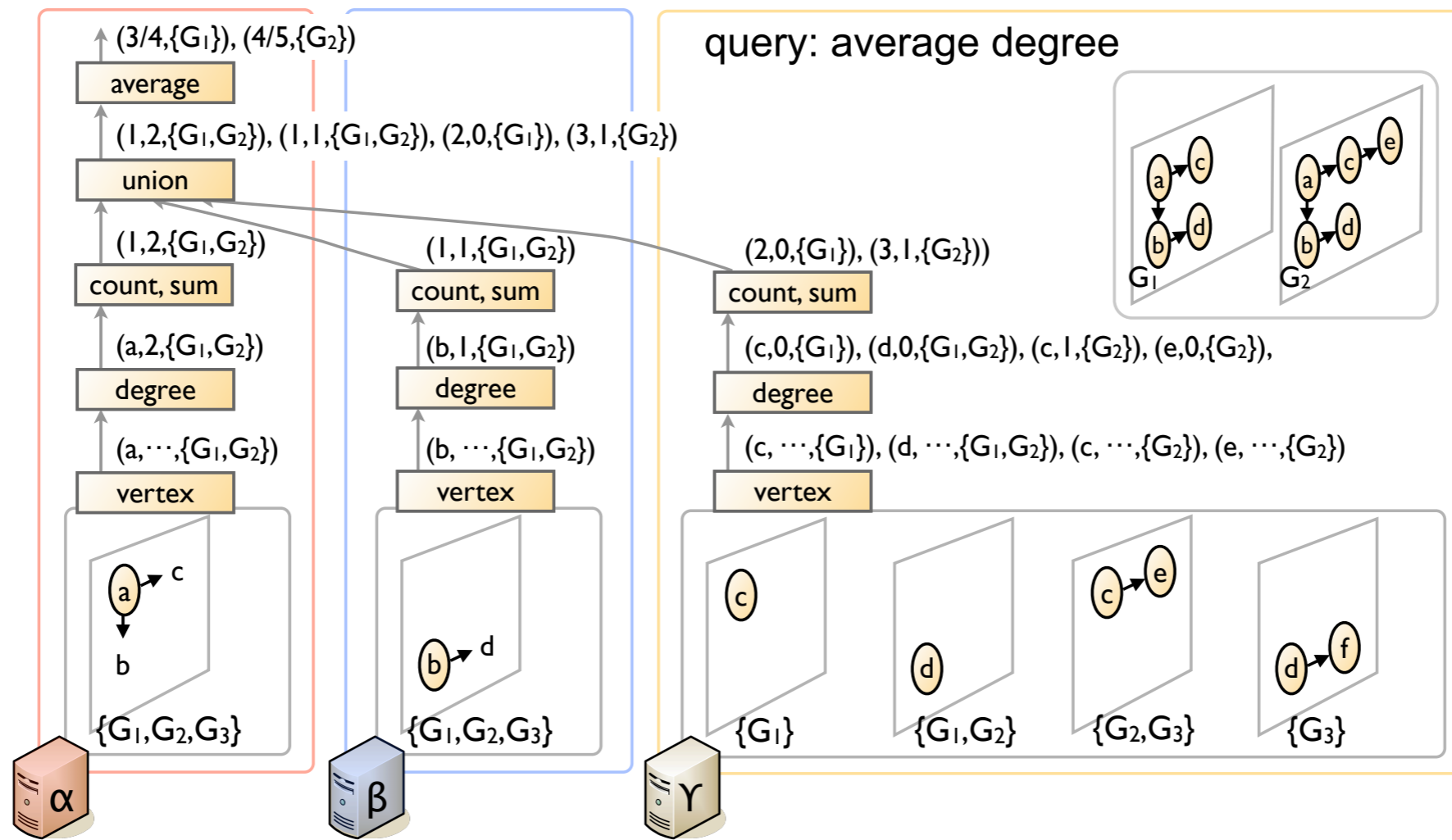
G* The Dynamic Graph Database

Deduplicated Graph Distribution



G* The Dynamic Graph Database

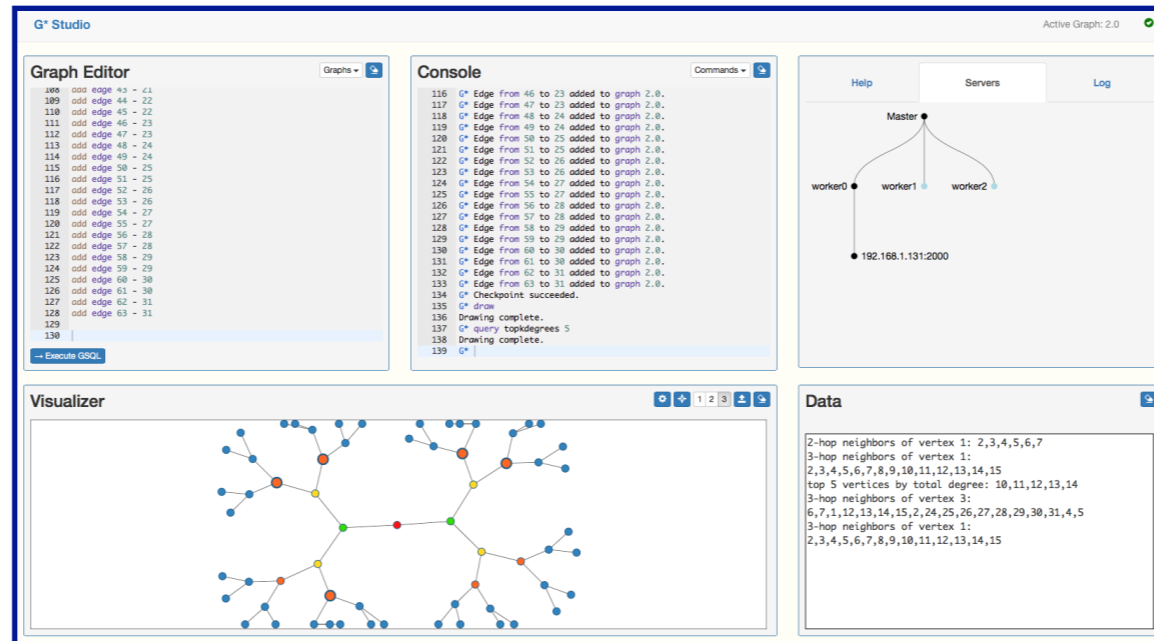
Parallel Graph Query Execution



G*

Implementation Overview

browser



JavaScript

Bootstrap
jQuery
D3

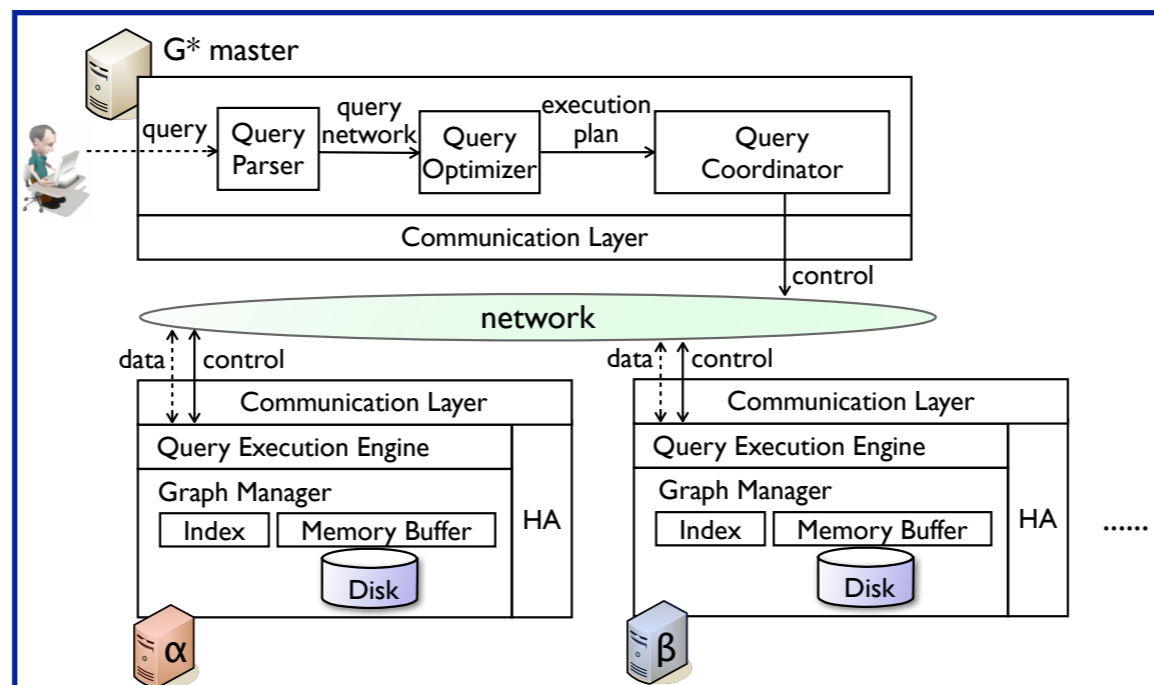
API



Java

NanoHTTPD

servers

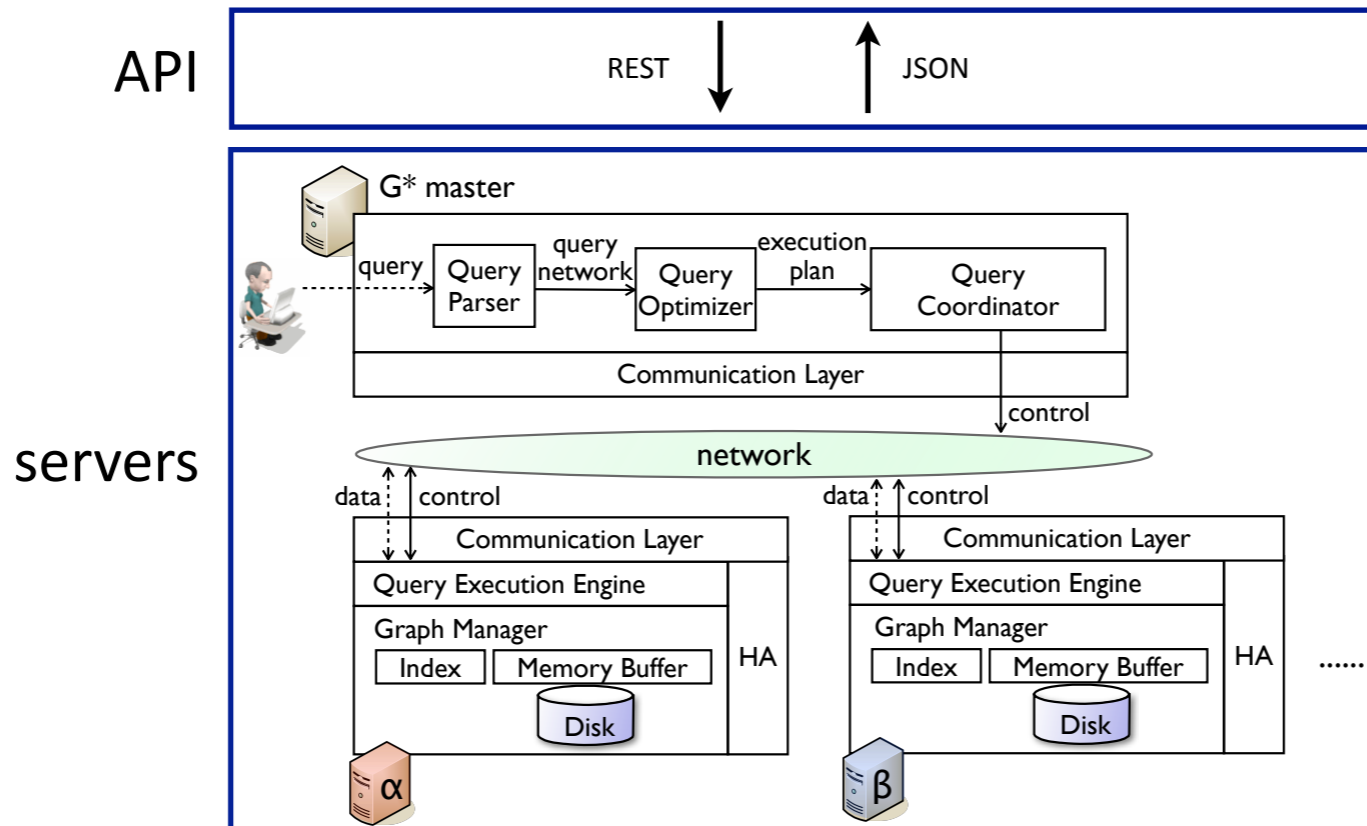
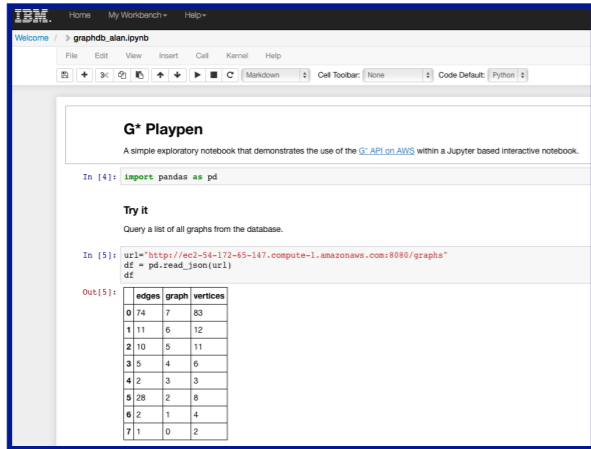


Java

custom RMI, BSP, and messaging



We can build other applications on top of the G* API.





and IBM's Knowledge Anyhow Workbench

The screenshot shows the IBM Knowledge Anyhow Workbench interface. At the top, there is a navigation bar with 'Home', 'My Workbench', and 'Help'. Below that, the current notebook is identified as 'graphdb_alan.ipynb'. A menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', and 'Help'. A toolbar contains icons for file operations and execution, along with dropdowns for 'Markdown', 'Cell Toolbar: None', and 'Code Default: Python'. The main content area features a title 'G* Playpen' and a subtitle: 'A simple exploratory notebook that demonstrates the use of the [G* API on AWS](#) within a Jupyter based interactive notebook.'

The notebook contains two code cells:

```
In [4]: import pandas as pd
```

Try it
Query a list of all graphs from the database.

```
In [5]: url="http://ec2-54-172-65-147.compute-1.amazonaws.com:8080/graphs"
df = pd.read_json(url)
df
```

The output of the second cell is a table:

	edges	graph	vertices
0	74	7	83
1	11	6	12
2	10	5	11
3	5	4	6
4	2	3	3
5	28	2	8
6	2	1	4
7	1	0	2



G* and the raw API

G* REST API Server version gs.0.63

Copyright (c) 2014-2015 Alan G. Labouseur. All Rights Reserved.

Home

Method = GET

URI / API command = /home

API help

API commands: GET [action], POST [action], PUT [action], DELETE [action]

<pre>+-- GET /ver[sion] +-- GET /time +-- GET /workers +-- GET /graphs + GET /graphs/[gid] + GET /graphs/[gid]/vertices + GET /graphs/[gid]/edges + GET /graphs/[gid]/summary +-- GET /currentgraph +-- GET /shutdown +-- GET /checkpoint +-- GET /query/gsql/degreedistribution/[gid] +-- GET /query/gsql/topkdegrees/[gid]/[k] +-- GET /query/gsql/topkdegreechangesbydelta/[start gid]/[stop gid]/[k] +-- POST /graphs/[new-gid] +-- POST /graphs/[new-gid]-[old-gid] +-- POST /graphs/[gid]/vertices/[vid] +-- POST /graphs/[gid]/edges/[vidFrom]-[vidTo] +-- POST /currentgraph/[gid] +-- POST /graphs/[gid]/vertices/[vid]/attr_type/attr_name/attr_value +-- POST /graphs/[gid]/edges/[vidFrom]-[vidTo]/attr_type/attr_name/attr_value</pre>	<ul style="list-style-type: none">- API version- current time and elapsed time since last call to time- worker data- short listing of all graphs- longer listing for graph [gid]- all vertices in graph [gid]- all edges (with directional notes) in graph [gid]- vertices and edges (with weights) in graph [gid]- show the graph context (gid of graph currently in use)- shutdown the G* workers and server. Be careful with this.- force a checkpoint.- distribution of total degrees in graph [gid]- top [k] vertices in graph [gid] by total degree- top [k] vertices with the largest change in degree over consecutive snapshot pairs - create a new (empty) graph with id [new-gid]- create a new graph with id [new-gid] as a clone of [old-gid]- create a vertex [vid] in existing graph [gid]- create an edge from vertex [vidFrom] to [vidTo] in existing graph [gid]- set the graph context (the graph currently in use) to graph [gid]- add or update an attribute for a vertex- add or update an attribute for an edge
--	--

API Testers

GET	/	<input type="text"/>
POST	/	<input type="text"/>
<input type="text"/>		





G* integrates with Relational databases.
Supports *SQL to Graph* and *Graph to SQL* transformations.

The screenshot shows the BIG* Data Studio interface with the following components:

- Graph Editor:** A list of commands for adding edges between patients and doctors.
- Console:** A log of G* commands and their execution results, such as "Edge from pat.47 to dr.33 added to graph".
- SQL Editor:** A SQL query for selecting doctors and patients based on treatment and symptoms.
- Visualizer:** A graph visualization showing nodes and edges.
- Data:** A table showing the results of a query, with columns for doctor, patient ID, treatment status, and symptoms.

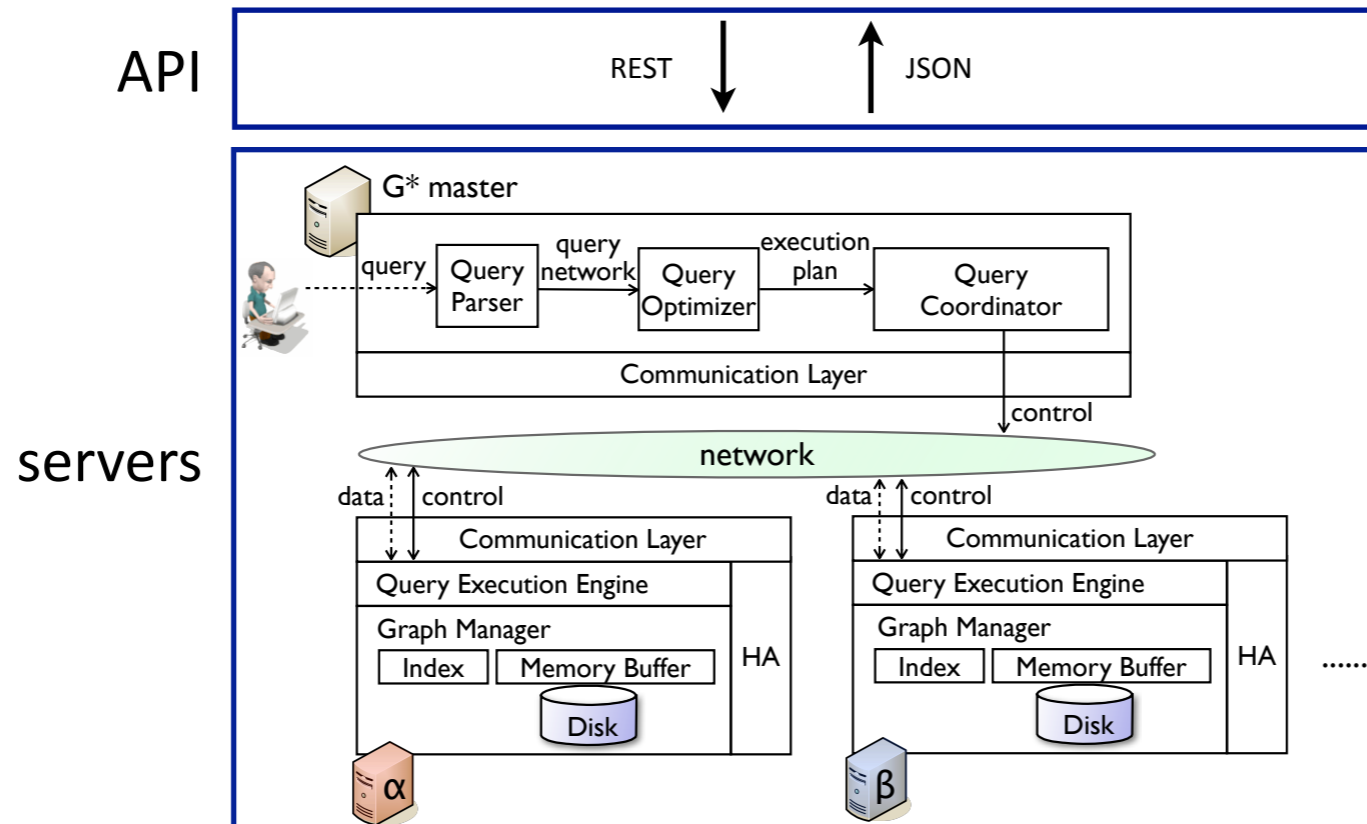
doctor	dr	is_treating	pid	for_feeling	symptom
doctor	16	is treating	45	for feeling	sick
doctor	18	is treating	29	for feeling	sick
doctor	23	is treating	29	for feeling	sick

Microsoft SQL Server
ORACLE
PostgreSQL



G*

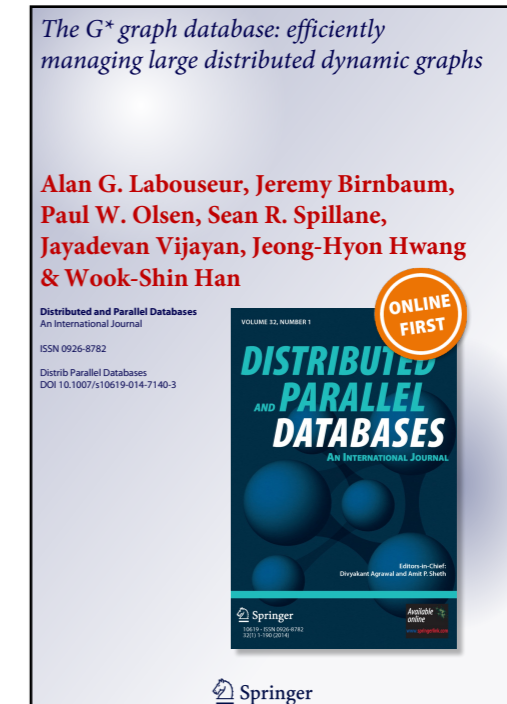
More integrations coming soon.





Selected Research Publications

- [1] Labouseur, A., Birnbaum, J., P. W. Olsen Jr., Spillane, S.R., Vijayan, J., Hwang, J.H., Han, W.S.: The g* graph database: efficiently managing large distributed dynamic graphs. *Distributed and Parallel Databases* (2014) 1–36
- [2] Spillane, S.R., Birnbaum, J., Bokser, D., Kemp, D., Labouseur, A., P. W. Olsen Jr., Vijayan, J., Hwang, J.H.: A Demonstration of the G* Graph Database System. In: *ICDE*. (2013) 1356–1359
- [3] Labouseur, A., P. W. Olsen Jr., Hwang, J.H.: Scalable and robust management of dynamic graph data. In Cormode, G., Yi, K., Deligiannakis, A., Garofalakis, M.N., eds.: *BD3@VLDB*. Volume 1018 of *CEUR Workshop Proceedings*., CEUR-WS.org (2013) 43–48
- [4] Labouseur, A., P. W. Olsen Jr., Park, K., Hwang, J.H.: A Demonstration of Query-Oriented Distribution and Replication Techniques for Dynamic Graph Data. In Chung, C.W., Broder, A.Z., Shim, K., Suel, T., eds.: *WWW (Companion Volume)*, ACM (2014) 127–130





Try it

G*, G*studio, and Big Data Studio are currently deployed for testing on 3NF Consulting's web site.

G* API <http://www.3nfconsulting.com:8080>
G*studio <http://www.3nfconsulting.com/gstarstudio/gss.html>
Big Data Studio <http://www.3nfconsulting.com/gstarstudio/index.html>



Non-commercial research use

Source code for a version of the G* database itself (excluding G*studio and API) is available for non-commercial research use. Request it at <http://www.cs.albany.edu/~gstar/request-form>

The Cloud Computing and Analytics Center at Marist College hosts G*, G*studio, and Big Data Studio in their secure cloud bunker. Contact them to discuss how their faculty and students can help transform your business through incubation, analytics, and training.

e-mail ccac@Marist.edu
web <http://ccac.Marist.edu>
phone 845-575-3058



Commercial use

G* will be available for licensing from 3NF Consulting, Inc. through a developing agreement with the The Research Foundation for The State University of New York. The G* API, G*studio, and Big Data Studio — well as graph and database consulting — are all directly available from 3NF Consulting, Inc.

e-mail alan@3NFconsulting.com
web <http://www.3NFconsulting.com>
phone 845-440-1102



3NF CONSULTING, INC.

G*studio

and

G*The Dynamic
Graph Database

Contact

Alan G. Labouseur, Ph.D.

phone 1-845-440-1102

web www.Labouseur.com
www.3NFconsulting.com
www.marist.edu/compscimath/faculty.html

e-mail alan@Labouseur.com
alan@3NFconsulting.com
alan.labouseur@Marist.edu