

# G\*The Dynamic Graph Database

## Technical Overview



Alan G. Labouseur, Ph.D.

Marist College  
3NF Consulting, Inc.

[alan.labouseur@Marist.edu](mailto:alan.labouseur@Marist.edu)  
[alan@3NFconsulting.com](mailto:alan@3NFconsulting.com)



## Summary

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

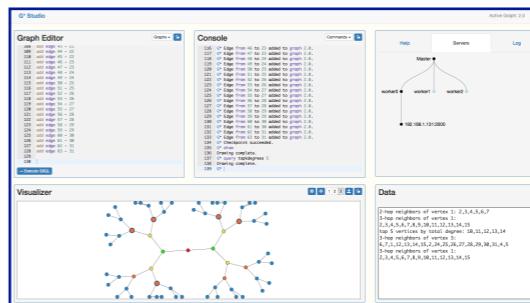
- graph distribution
  - multi-core scale up
  - multi-server scale out
- deduplicated disk storage for very large graphs
- in-memory compact indexing
- shared computation
- easy use of sophisticated parallel graph-theoretic queries
- integrates with Relational databases and other stores



National Science Foundation  
WHERE DISCOVERIES BEGIN

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

G\* Studio

Active Graph: 2.0

### Graph Editor

```
108 aaa edge 45 - 21
109 add edge 44 - 22
110 add edge 45 - 22
111 add edge 46 - 23
112 add edge 47 - 23
113 add edge 48 - 24
114 add edge 49 - 24
115 add edge 50 - 25
116 add edge 51 - 25
117 add edge 52 - 26
118 add edge 53 - 26
119 add edge 54 - 27
120 add edge 55 - 27
121 add edge 56 - 28
122 add edge 57 - 28
123 add edge 58 - 29
124 add edge 59 - 29
125 add edge 60 - 30
126 add edge 61 - 30
127 add edge 62 - 31
128 add edge 63 - 31
129
130
131
132
133
134 G* Checkpoint succeeded.
135 G* draw
136 Drawing complete.
137 G* query topkdegrees 5
138 Drawing complete.
139 G*
```

→ Execute GSQL

### Console

```
116 G* Edge from 46 to 23 added to graph 2.0.
117 G* Edge from 47 to 23 added to graph 2.0.
118 G* Edge from 48 to 24 added to graph 2.0.
119 G* Edge from 49 to 24 added to graph 2.0.
120 G* Edge from 50 to 25 added to graph 2.0.
121 G* Edge from 51 to 25 added to graph 2.0.
122 G* Edge from 52 to 26 added to graph 2.0.
123 G* Edge from 53 to 26 added to graph 2.0.
124 G* Edge from 54 to 27 added to graph 2.0.
125 G* Edge from 55 to 27 added to graph 2.0.
126 G* Edge from 56 to 28 added to graph 2.0.
127 G* Edge from 57 to 28 added to graph 2.0.
128 G* Edge from 58 to 29 added to graph 2.0.
129 G* Edge from 59 to 29 added to graph 2.0.
130 G* Edge from 60 to 30 added to graph 2.0.
131 G* Edge from 61 to 30 added to graph 2.0.
132 G* Edge from 62 to 31 added to graph 2.0.
133 G* Edge from 63 to 31 added to graph 2.0.
134 G* Checkpoint succeeded.
135 G* draw
136 Drawing complete.
137 G* query topkdegrees 5
138 Drawing complete.
139 G*
```

### Servers

Help      Servers      Log

```
graph TD; Master --- worker0; Master --- worker1; Master --- worker2;
```

worker0  
192.168.1.131:2000

### Visualizer

### Data

```
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
```

# G\* studio

## 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](#)

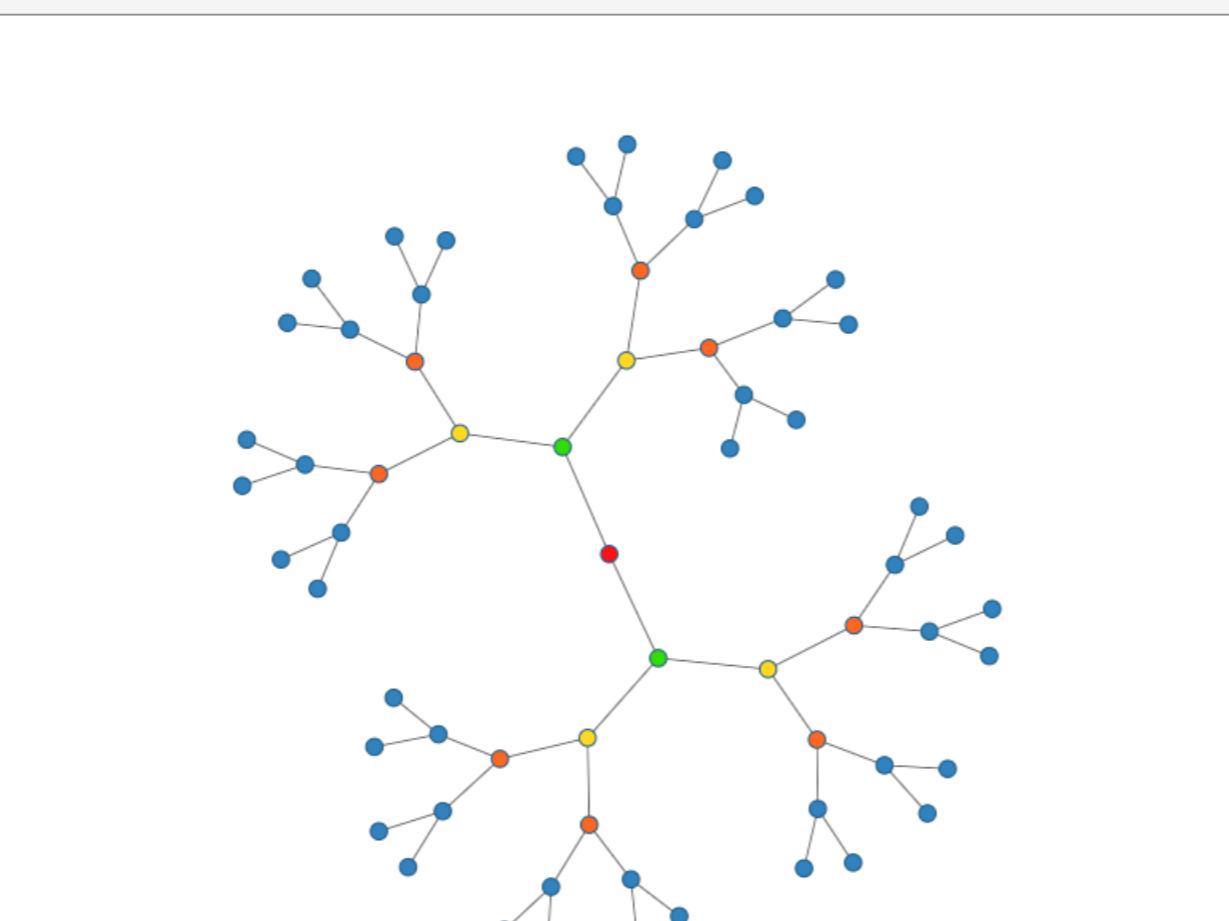
[Top-k vertices by degree](#)

[Top-k vertices with the largest change in degree over consecutive graph snapshot pairs](#)

# G\* studio

## Visualizer and Data

### Visualizer



The Visualizer panel displays a complex graph structure with numerous nodes and edges. Some nodes are highlighted in different colors (blue, orange, yellow, green), while others are standard blue dots. The graph appears to be a large, interconnected cluster with several smaller sub-clusters.

### 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>[,...])**

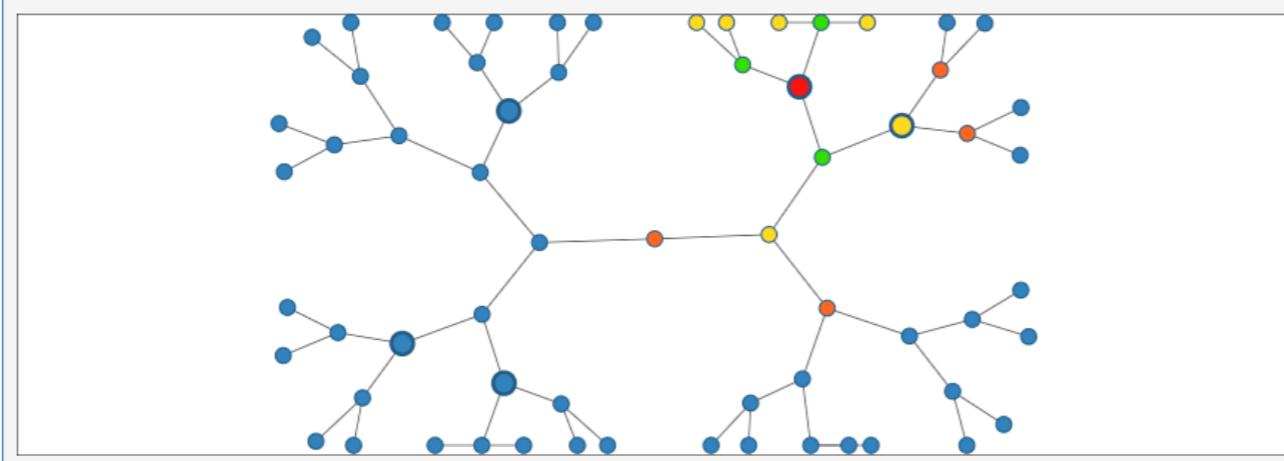
### Data

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

# G\* studio

## Top-k Query

**Visualizer**



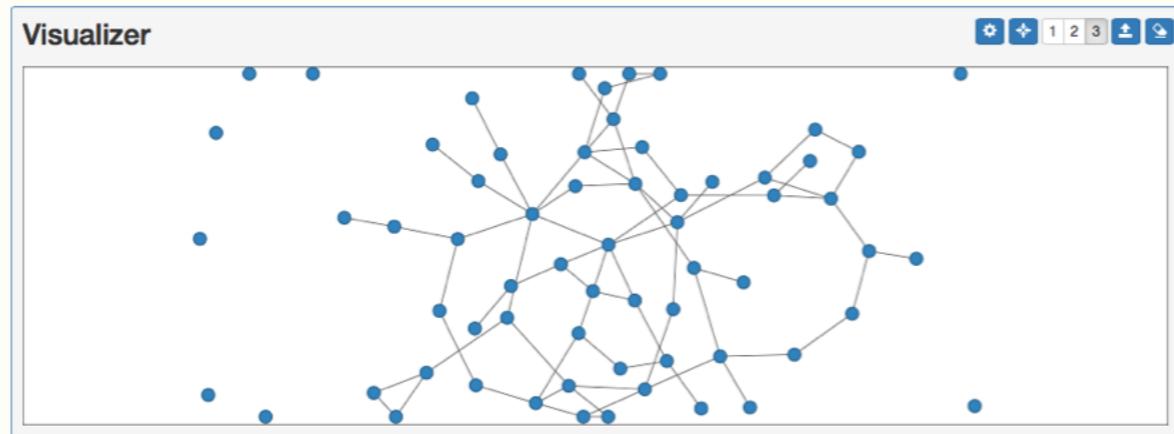
**Data**

```
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
```

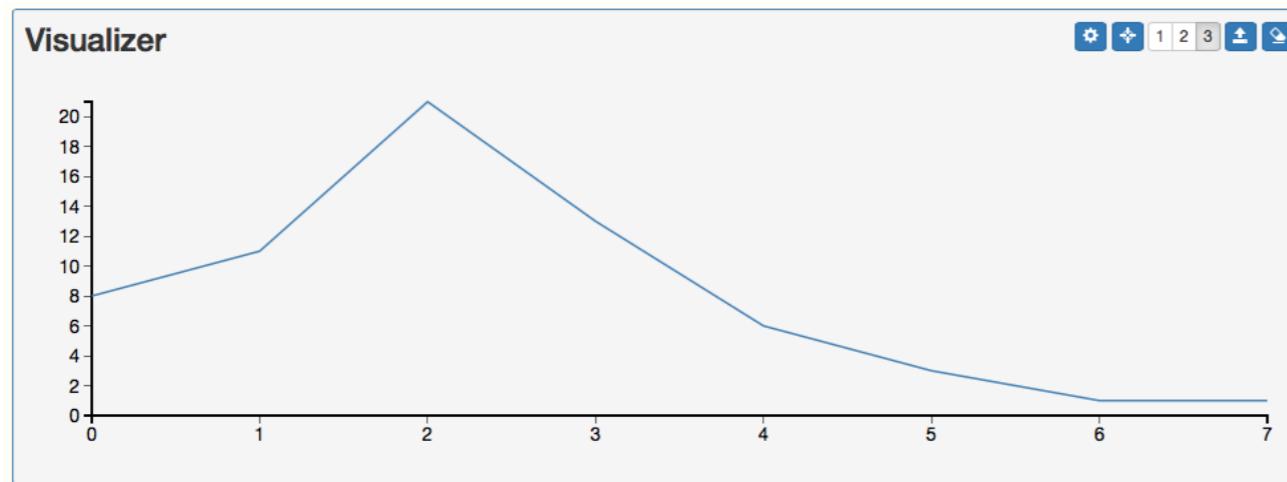


## Degree Distribution Query

Erdős-Rényi random graph



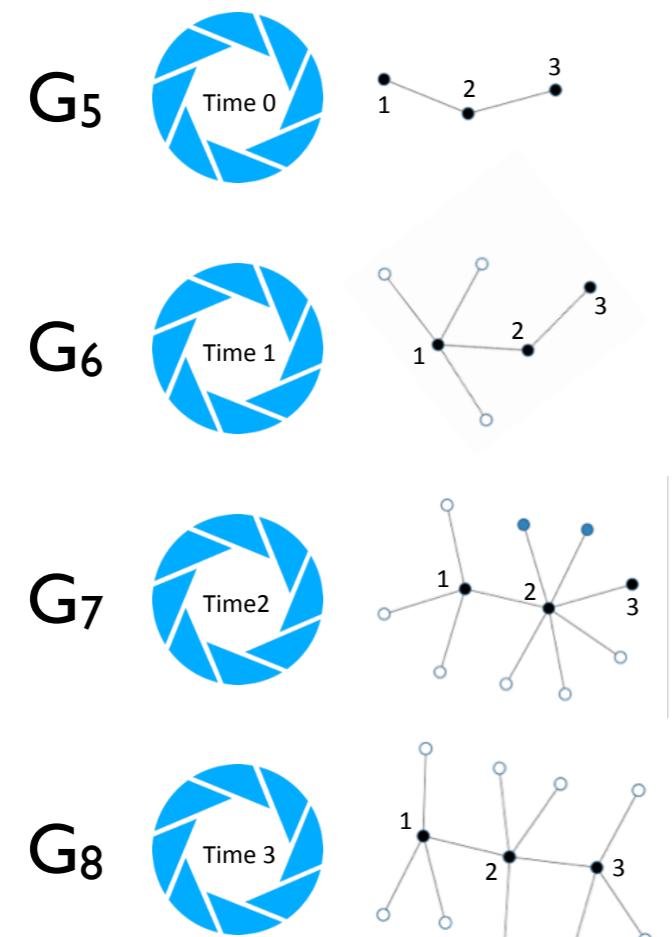
Degree Distribution



Data

```
degree distribution:  
total_degree , count  
0 , 8  
1 , 11  
2 , 21  
3 , 13  
4 , 6  
5 , 3  
6 , 1  
7 , 1
```

## Rising/Falling Stars Query

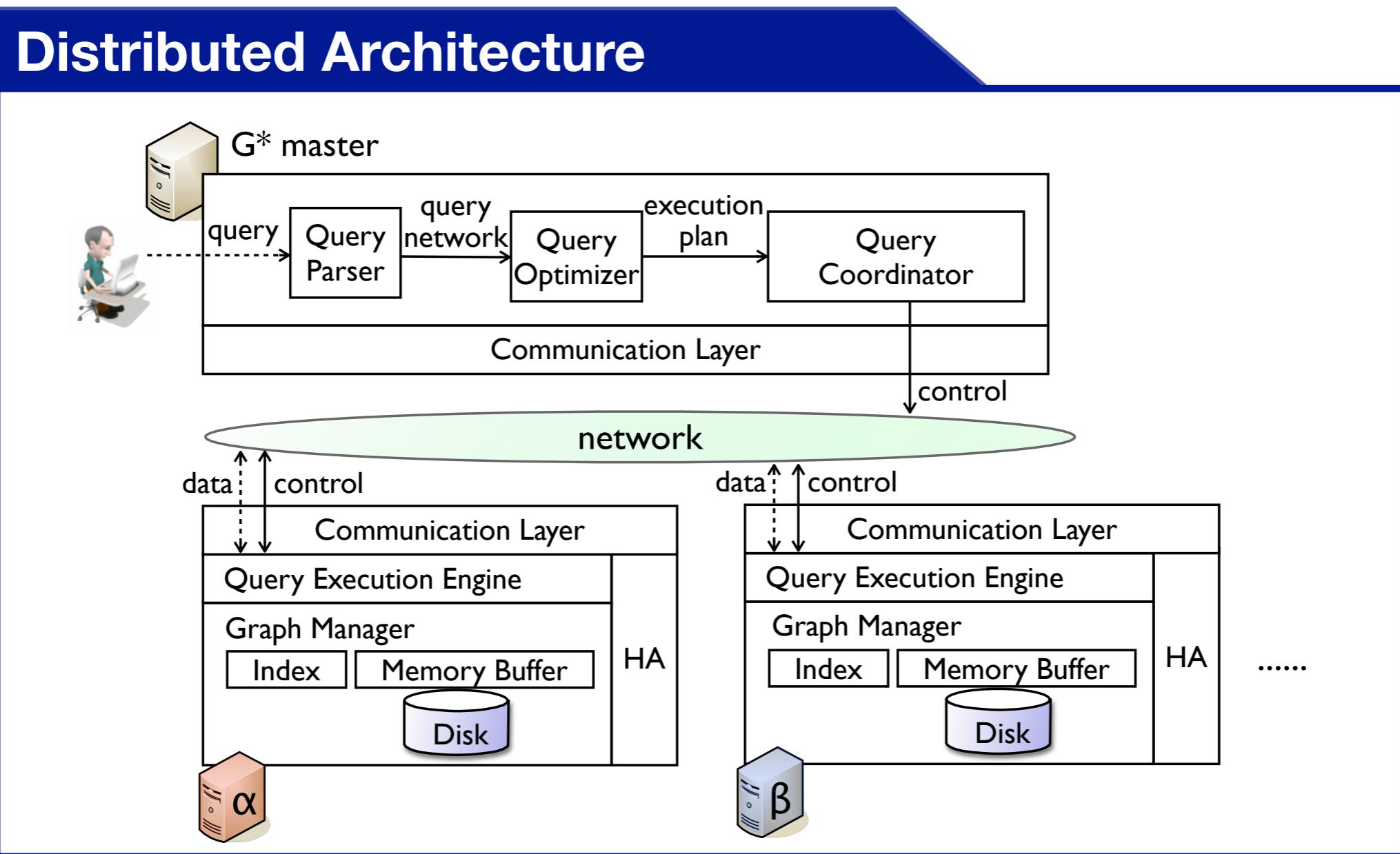


### Data

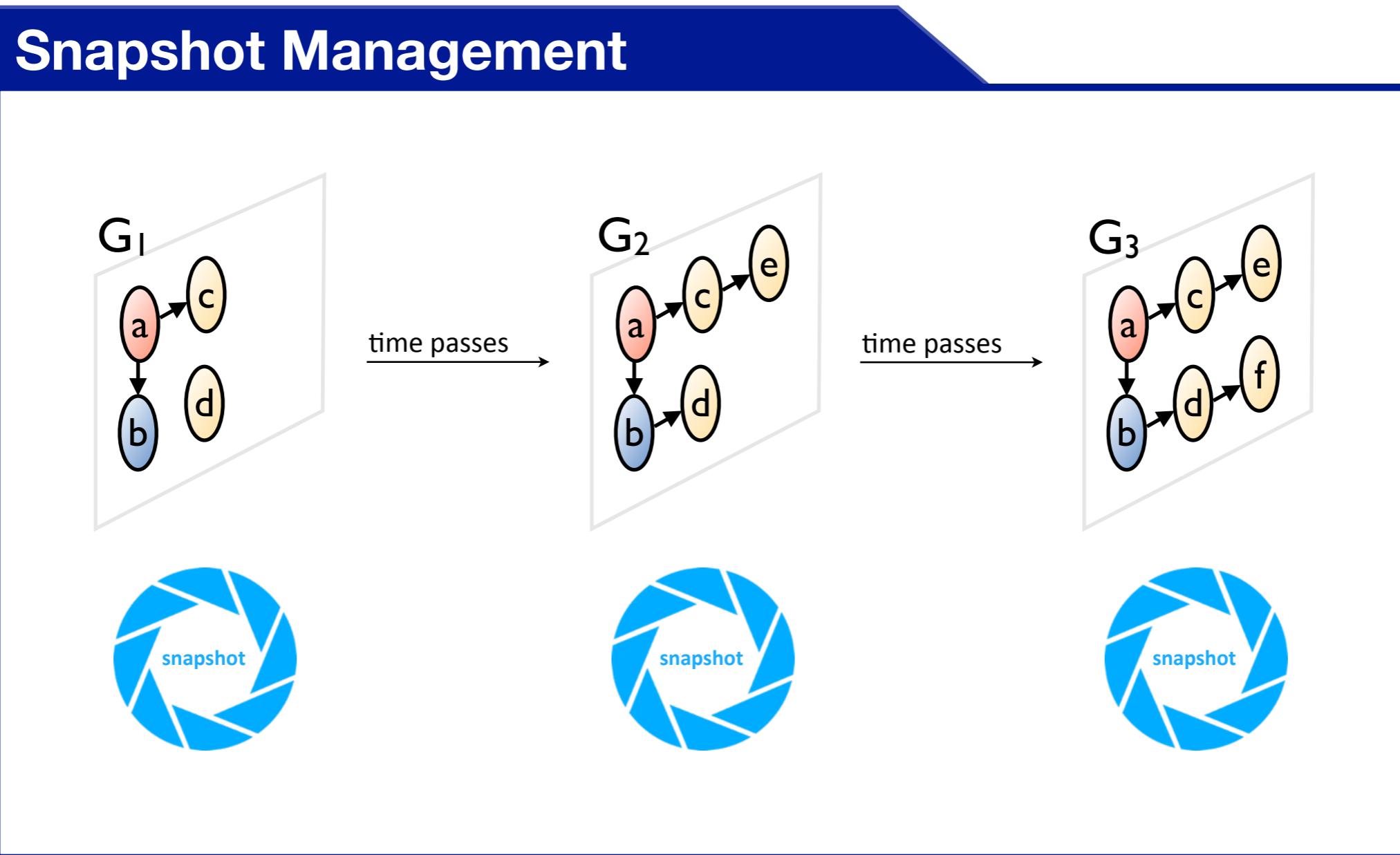
top 20 vertices with the largest change in degree over consecutive graph snapshot pairs from 6 to 8:  
snapshotPairs , vertexID , change

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

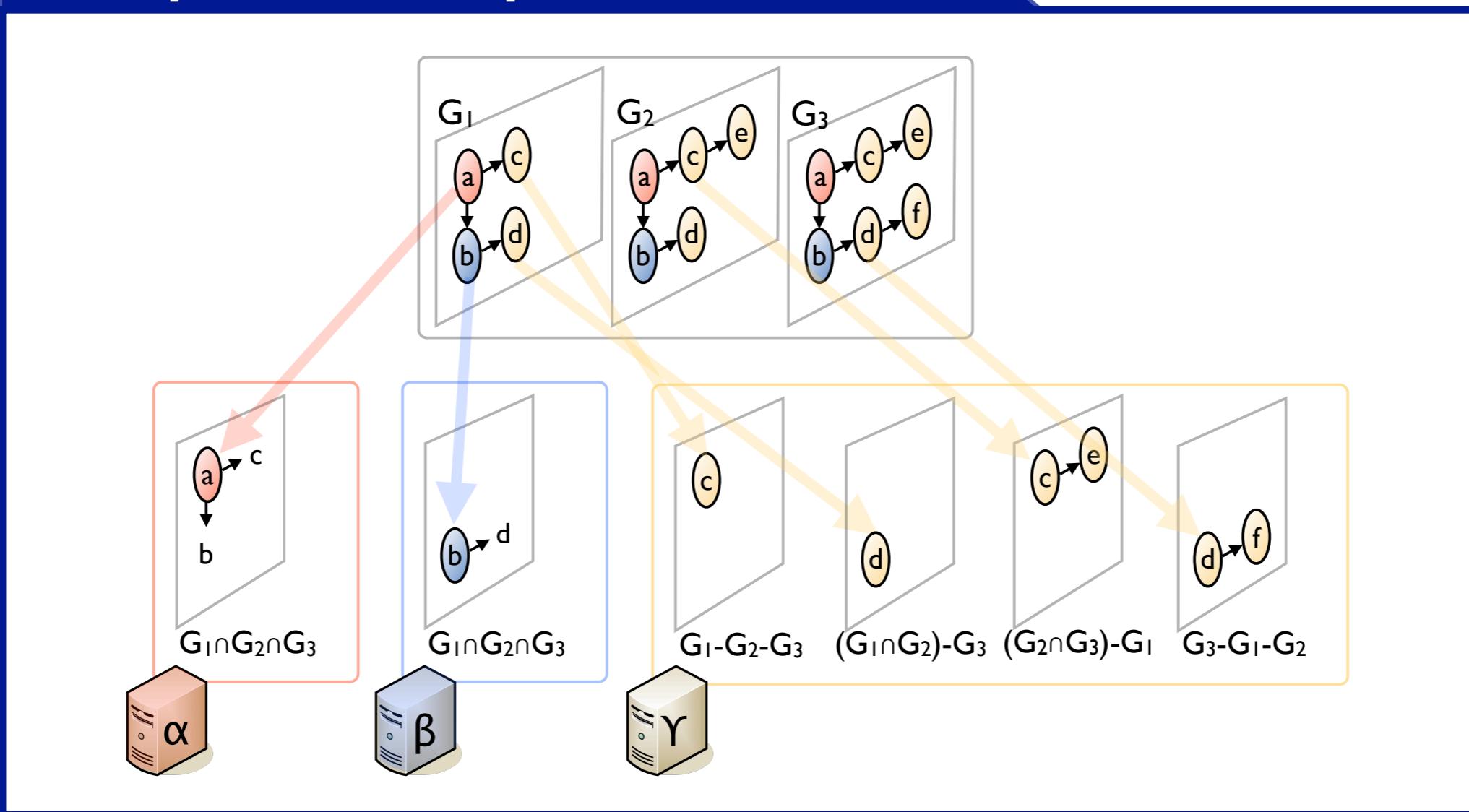


# G\*The Dynamic Graph Database



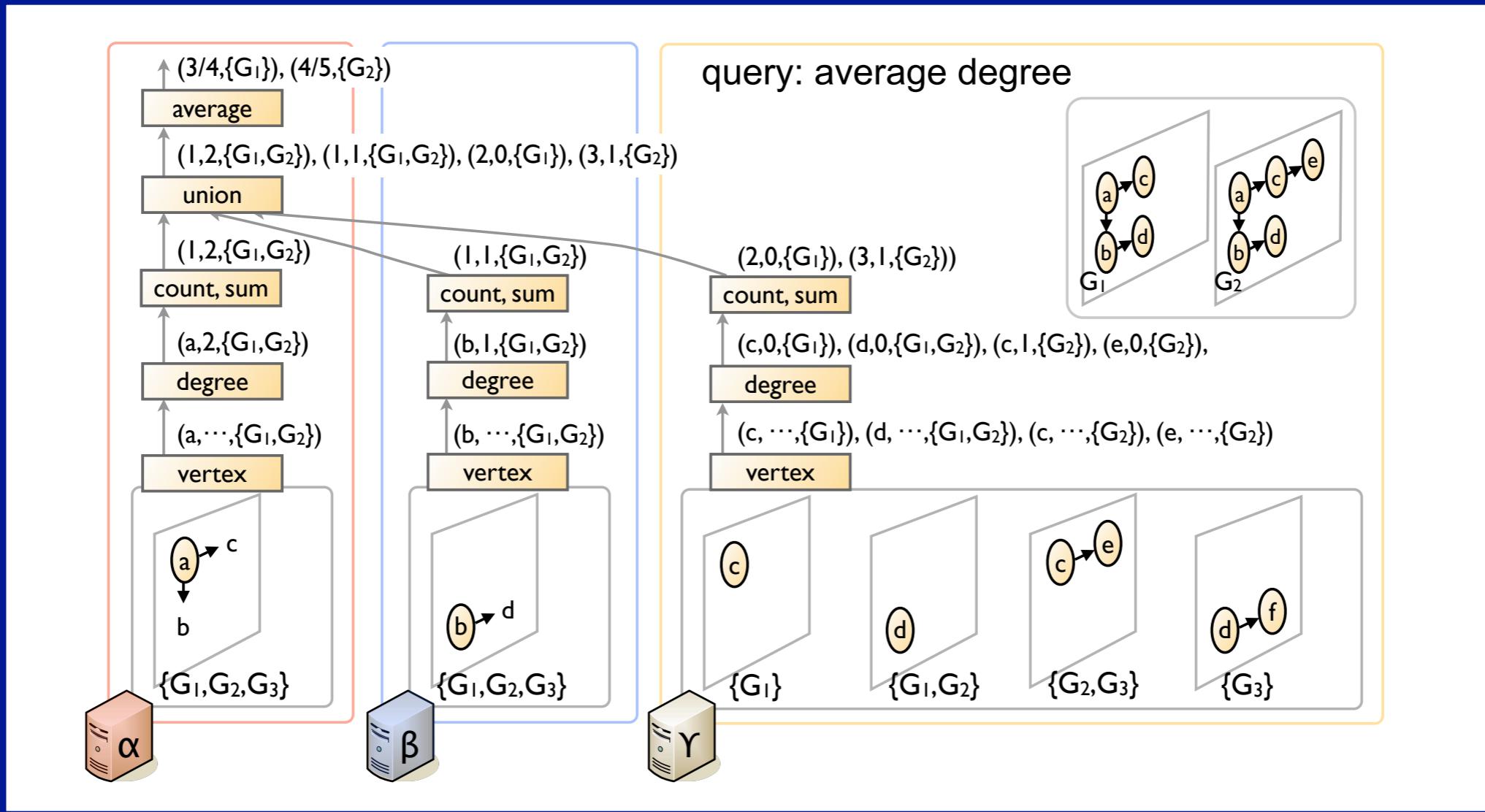
# G\* The Dynamic Graph Database

## Deduplicated Graph Distribution



# G\* The Dynamic Graph Database

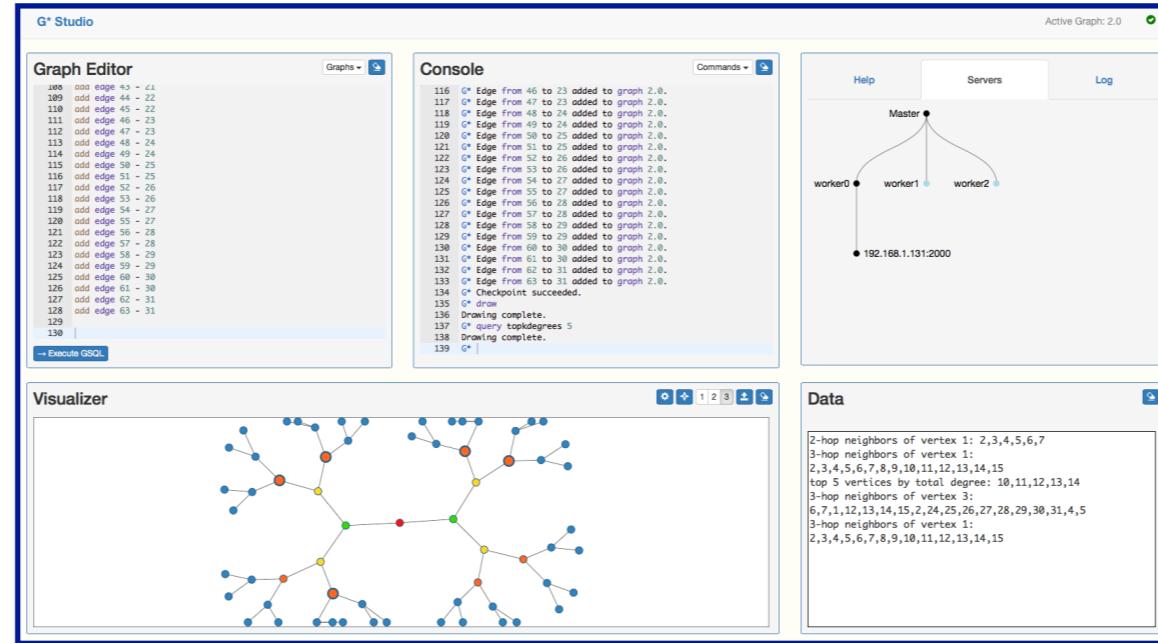
## Parallel Graph Query Execution





## Implementation Overview

browser

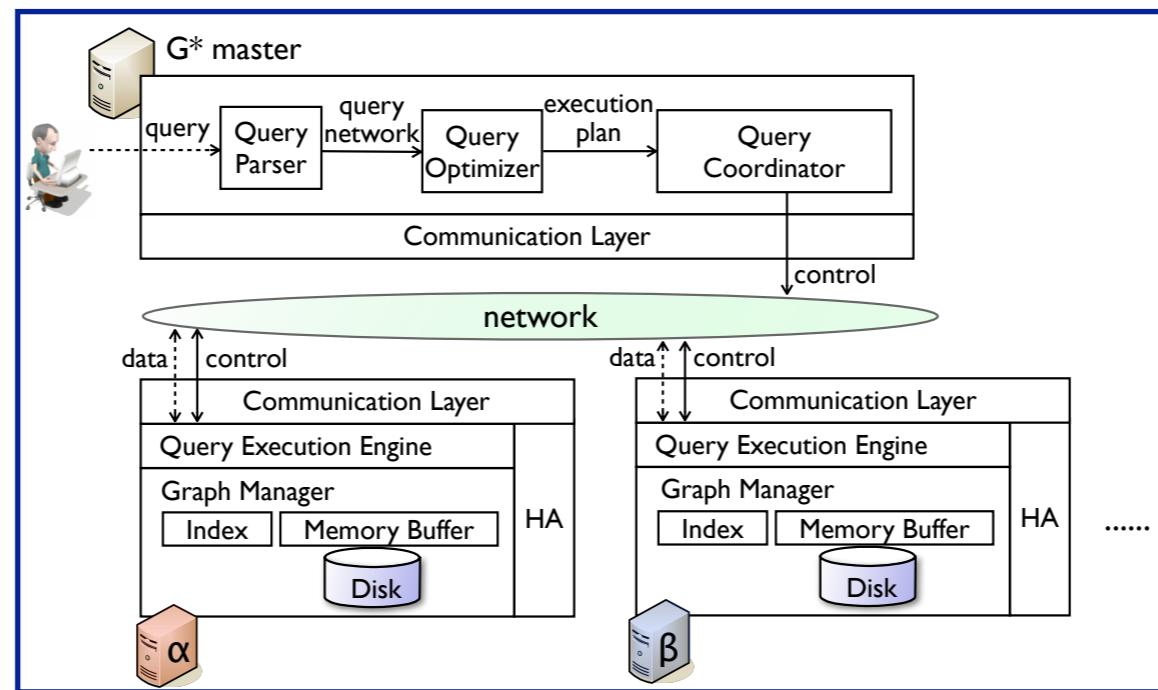


API

REST

JSON

servers



JavaScript  
Bootstrap  
jQuery  
D3

Java  
NanoHTTPD

Java  
custom RMI, BSP, and messaging



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

Welcome / > graphdb\_alan.ipynb

File Edit View Insert Cell Kernel Help

Cell Toolbar: None Code Default: Python 3

## G\* Playpen

A simple exploratory notebook that demonstrates the use of the [G\\* API on AWS](#) within a Jupyter based interactive notebook.

In [4]: `import pandas as pd`

Try it

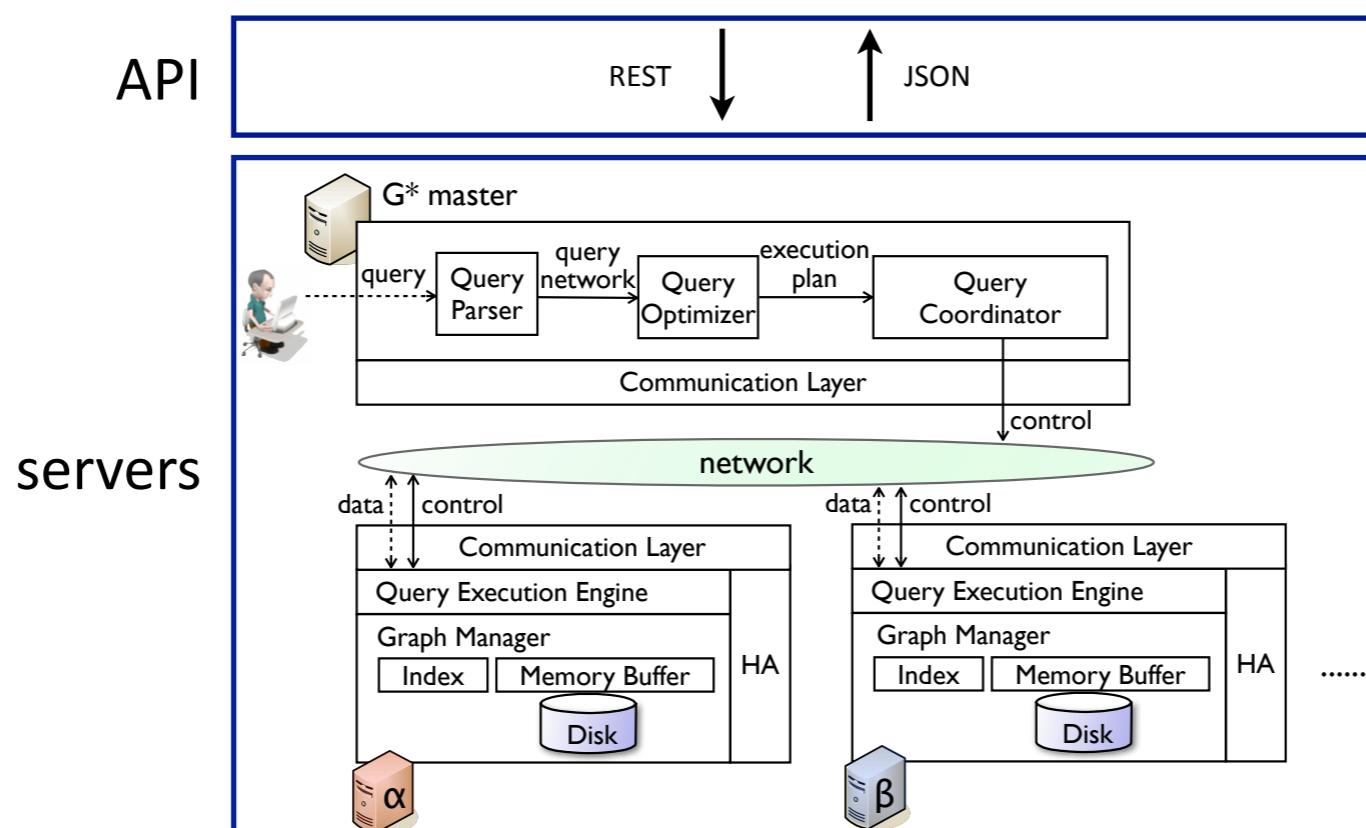
Query a list of all graphs from the database.

In [5]:

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

Out[5]:

	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





## and IBM's Knowledge Anyhow Workbench

The screenshot shows a Jupyter notebook interface with the following details:

- Title:** Welcome / graphdb\_alan.ipynb
- Toolbar:** File, Edit, View, Insert, Cell, Kernel, Help, Cell Toolbar: None, Code Default: Python
- Section Header:** **G\* Playpen**
- Description:** A simple exploratory notebook that demonstrates the use of the [G\\* API on AWS](#) within a Jupyter based interactive notebook.
- 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`
- Out[5]:** (A DataFrame showing the following data)

	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





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]

```
+-- 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/degreeistribution/[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
```

- 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 a edge

### API Testers

GET	/	<input type="text"/>
POST	/	<input type="text"/>
<input type="button" value="Submit"/>		

REST ↓ ↑ JSON



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

BIG\* Data Studio

Active Graph: 11.0

**Graph Editor**

```
139  add edge pat.54 - dr.14
140  add edge pat.47 - dr.12
141  add edge pat.45 - dr.16
142  add edge pat.29 - dr.18
143  add edge pat.29 - dr.23
144  add edge pat.32 - dr.23
145  add edge pat.36 - dr.25
146  add edge pat.39 - dr.27
147  add edge pat.58 - dr.30
148  add edge pat.0 - dr.33
149  add edge pat.46 - dr.33
150  add edge pat.47 - dr.33
151  add edge pat.49 - dr.33
152  add edge pat.62 - dr.33
153  add edge pat.56 - dr.38
154  add edge pat.48 - dr.39
155  add edge pat.50 - dr.39
156  add edge pat.64 - dr.39
157  add edge pat.52 - dr.42
158  add edge pat.55 - dr.47
159  add edge pat.64 - dr.49
160
161
```

→ Execute GSQL

**Console**

```
438 G* Edge from pat.47 to dr.33 added to graph 11.0.
439 G* Edge from pat.49 to dr.33 added to graph 11.0.
440 G* Edge from pat.62 to dr.33 added to graph 11.0.
441 G* Edge from pat.56 to dr.38 added to graph 11.0.
442 G* Edge from pat.48 to dr.39 added to graph 11.0.
443 G* Edge from pat.50 to dr.39 added to graph 11.0.
444 G* Edge from pat.64 to dr.39 added to graph 11.0.
445 G* Edge from pat.52 to dr.42 added to graph 11.0.
446 G* Edge from pat.55 to dr.47 added to graph 11.0.
447 G* Edge from pat.64 to dr.49 added to graph 11.0.
448 G* Checkpoint succeeded.
449 G* draw
450 Drawing complete.
451 G*
```

**SQL Editor**

```
1
2 -- Examples: Doctors/Patients Sample
3 SELECT 'doctor' as doctor,
4       d.did,
5       'is treating ' AS is_treating,
6       p.pid,
7       'for feeling ' AS for_feeling,
8       t.symptom
9 FROM testDoctors d,
10    testPatients p,
11    testTreatments t
12 WHERE t.did = d.did
13   AND t.pid = p.pid
14 ORDER BY did ASC, pid ASC
15
16
```

→ Execute SQL      74 rows returned.

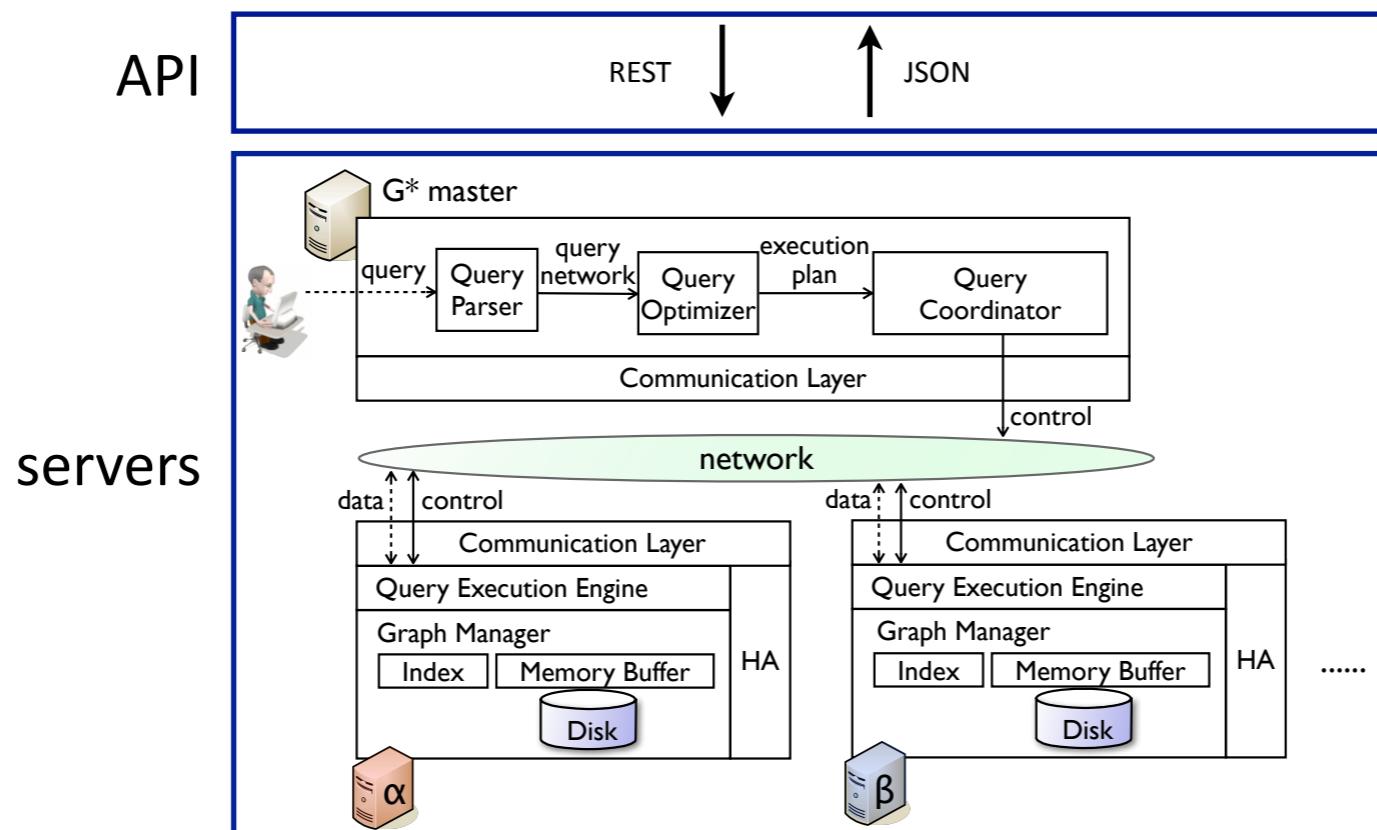
**Data**

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



# G\*

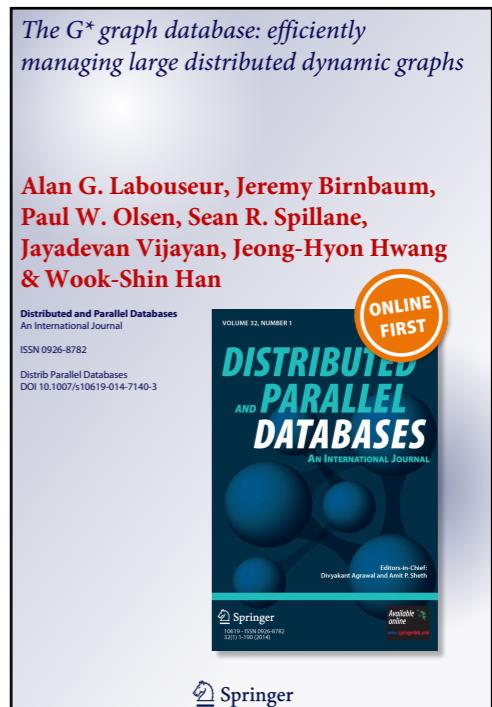
More integrations coming soon.





## 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



National Science Foundation  
WHERE DISCOVERIES BEGIN



WWW 2014  
The 23rd International World Wide Web Conference  
April 7-11, 2014 coex





and



# Contact

**Alan G. Labouseur, Ph.D.**

*phone*    1-845-440-1102

*web*      [www.Labouseur.com](http://www.Labouseur.com)

[www.3NFconsulting.com](http://www.3NFconsulting.com)

[www.marist.edu/compscimath/faculty.html](http://www.marist.edu/compscimath/faculty.html)

*e-mail*    [alan@Labouseur.com](mailto:alan@Labouseur.com)

[alan@3NFconsulting.com](mailto:alan@3NFconsulting.com)

[alan.labouseur@Marist.edu](mailto:alan.labouseur@Marist.edu)