

GoFFish: A Sub-Graph Centric Framework for Large-Scale Graph Analytics

Guide:

Prof. Dinkar Sitaram
Department of Computer Science and Engineering
PES Institute of Technology

Ms. Prafullata
Department of Computer Science and Engineering
PES Institute of Technology

Mentor:

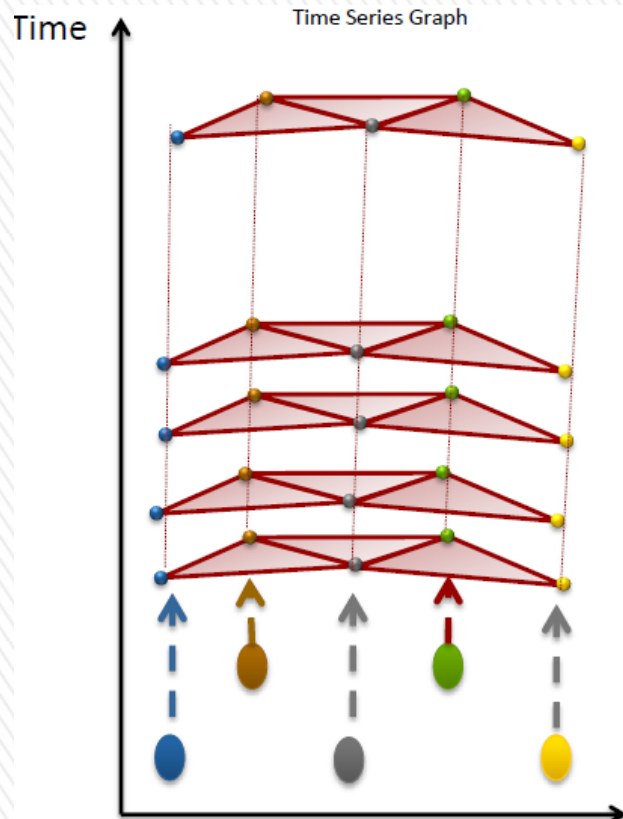
Prof. Yogesh Simmhan
Department of SERC
Indian Institute of Science

Team Members:

Bhavani B	1PI11IS027
Anushree PK	1PI11IS017
Mithilesh K G	1PI11IS059

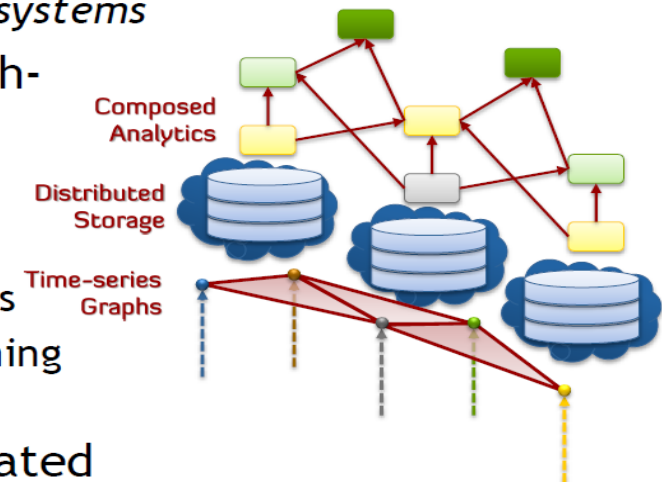


Analytics on Time series Graphs and the GoFFISH Software Platform



- Time series graph \mathcal{P} is a series of graph instances \hat{G} , over time.
- A Graph instance \hat{g} in \hat{G} , is a ordered pair $\hat{g} = (G, \mathfrak{t})$ where $G=(V,E)$ is a ordered pair comprising set V of vertices and or nodes together with set of E edges ; \mathfrak{t} denotes a time associated with a given Graph instance.

- Platform to *store*, *compose* & *execute* analytics on *time-series graph* datasets
 - At scale, on distributed systems
- **GoFS**: Distributed Graph-oriented File System
- **Gopher**
 - Compose sub-graph centric complex analytics
 - Executed on **Floe** streaming dataflow engine
- Data & Compute collocated



Problem Definition

Using the GoFFish framework:

- ▶ To store the real time graphs as – time series and subgraph-centric formats
- ▶ To compare the efficiencies of both based on the input query

Time Series Format

Timestamp	Graph
t0	graph_instance_0
.	.
.	.
.	.
<u>tn</u>	<u>graph_instance n</u>

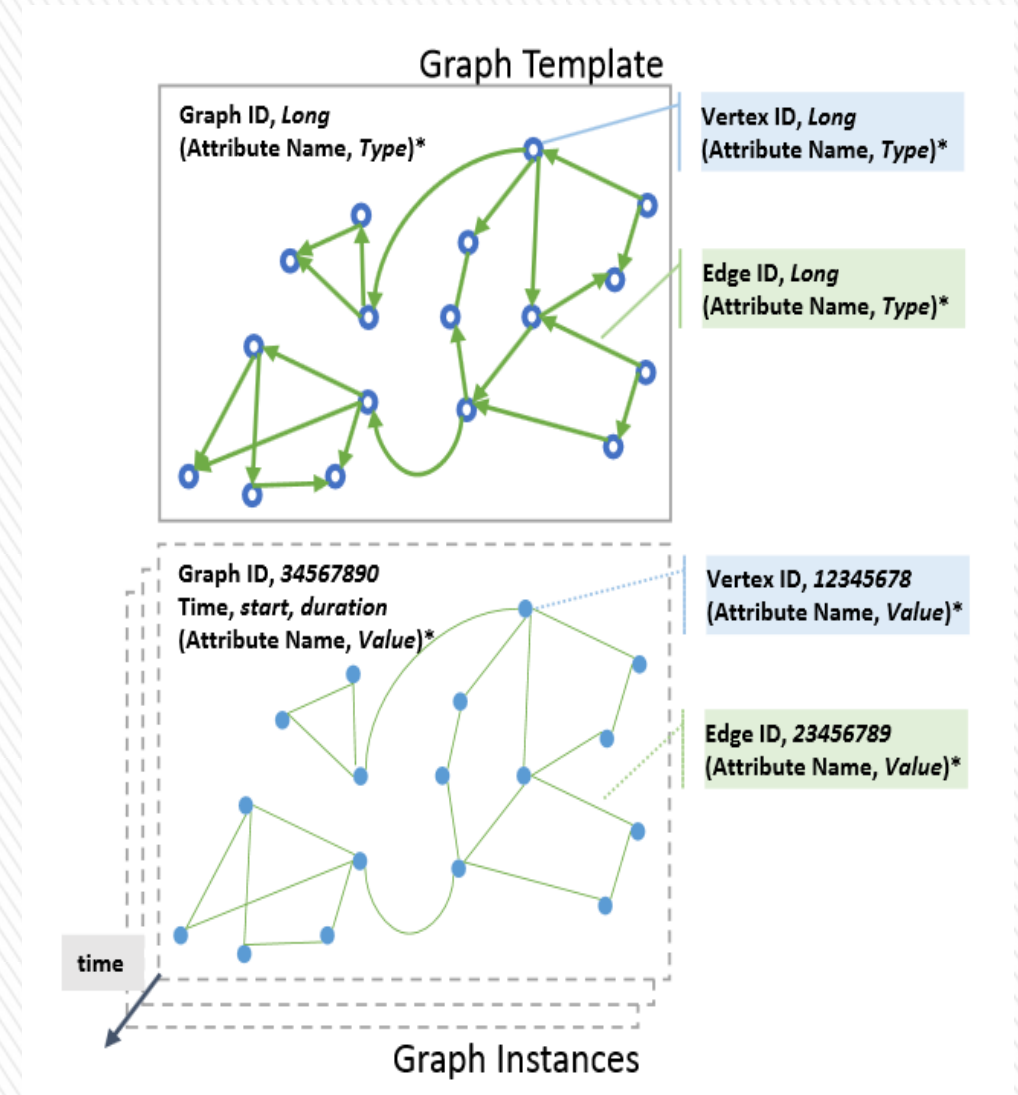
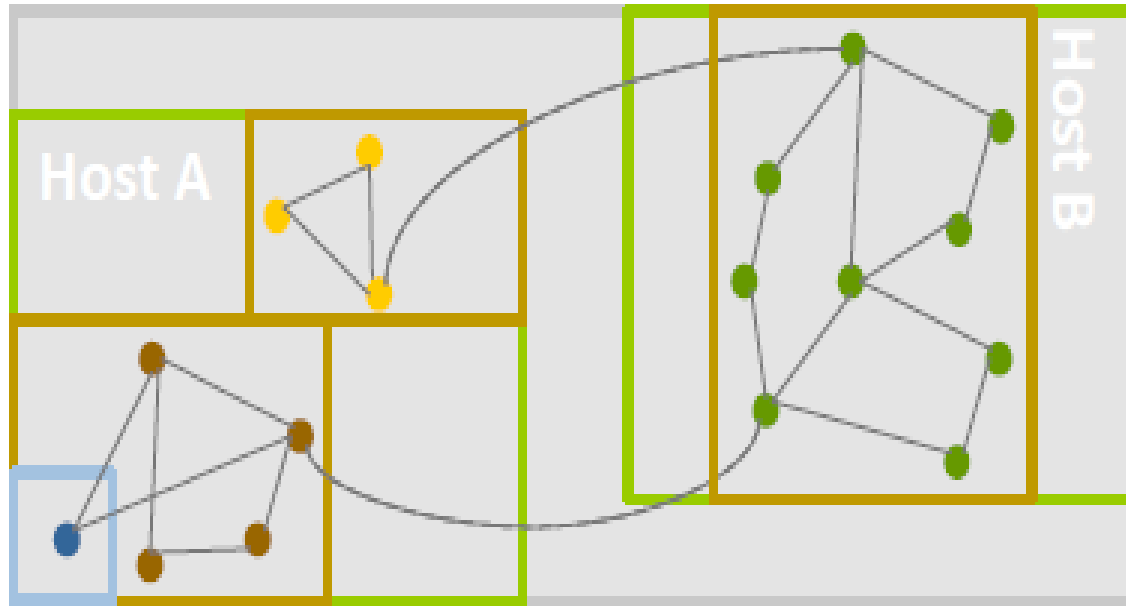
Subgraph Centric Format

Subgraph	t0	t1	t2	.	.	tn
Subgraph_1	val0	val1	val2	.	.	<u>valn</u>
.
.
.
<u>Subgraph_n</u>



Our Approach

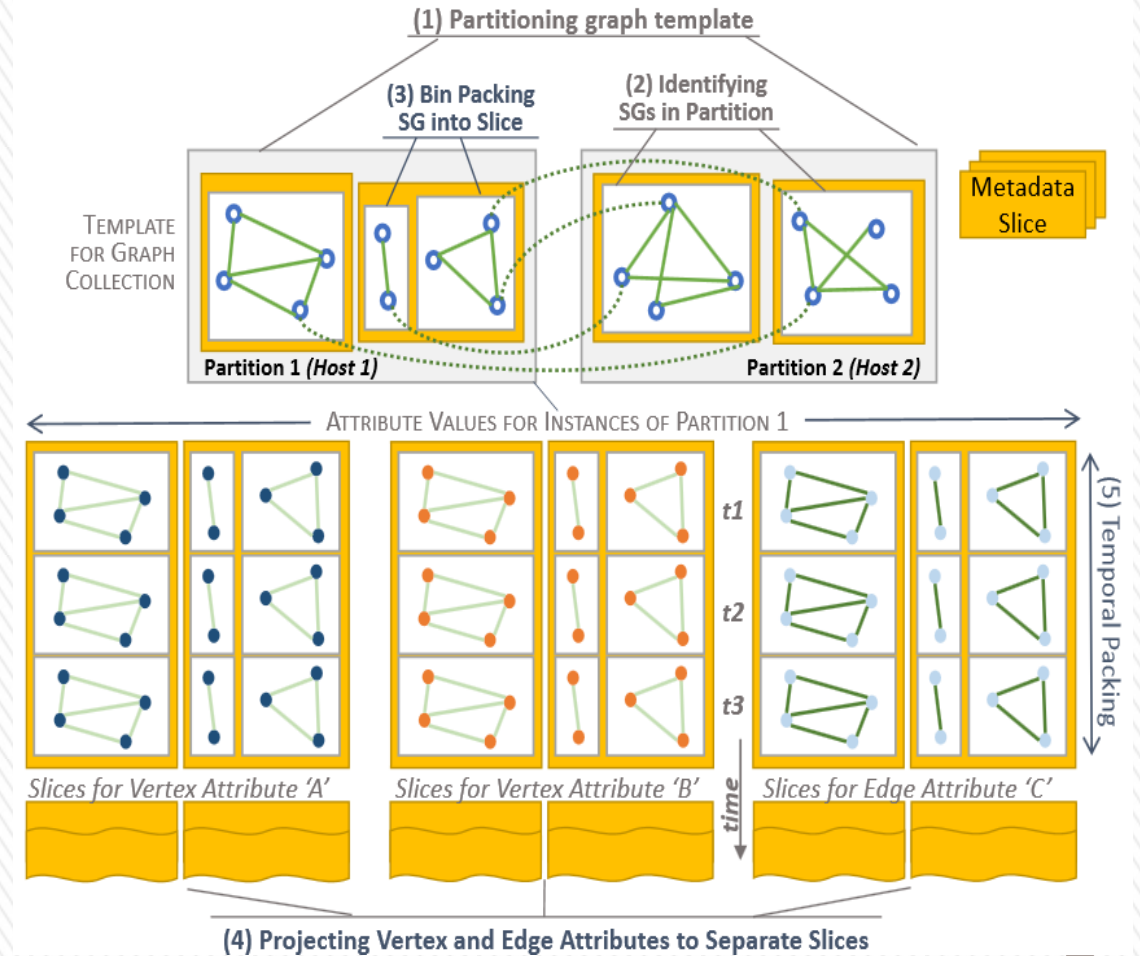
- Single Node Installation
- Load a graph and submit a Gopher job
- Compute time taken to run the Gopher job
- Print the result



Temporal and Subgraph bin packing

Temporal instance packing - packing nearby instances together within a single slice. Thus, an attribute slice storing a subgraph instance values will contain adjacent instances, and the slice will contain instances that span a time duration.

Subgraph bin packing-partitioning large graphs results in partitions with hundreds of subgraphs with highly variable vertex and edge counts, causing imbalances in slice read times across subgraphs and also imbalances in execution. y having a fixed number of slices (bins) and packing multiple subgraphs into a slice (bin) to balance the number of vertices/edges/vertices+edges in a bin, we limit the slice size and count



Set `serializer:instancegroupingsize` to ALL and set `serializer:numsubgraphbins` to a large value

Our Approach

RESULTS:

Algorithm : Vertex Count

Graph Datasets Source: Stanford Network Analysis Platform (SNAP)

- Set the parameters:
serializer:instancegroupingsize to ALL
serializer:numsubgraphbins to 10000
- Compute time taken

Graph	Graph Size(Nodes)	No. of Instances	No. of Partitions	Time Diff(without params set) in ns	Time Diff(with params set) in ns
P2P -1	6,301	4	1	171580	147348
P2P-1	6,301	4	2	117204	140786
P2P-2	62,586	4	1	250666	269528
P2P-2	62,586	4	2	269528	230438
P2P-2	62,586	200	1	188698	184639
P2P-2	62,586	200	2	165393	138883
Citation	34,546	4	4	236474	163621



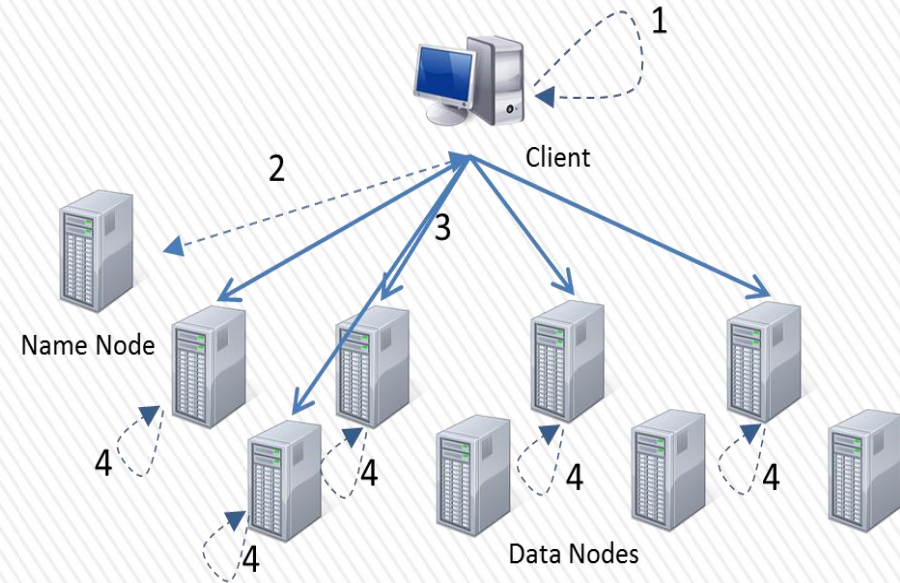
GoFFISH Cluster

- ▶ One of the machine in GoFFish cluster acts as head node, other nodes are referred to as worker nodes
- ▶ All machines in GoFFish cluster should be able to do password less ssh among each other
- ▶ Deploy GoFFish
- ▶ Load sample graph in GoFS
- ▶ Run sample Gopher job

Our Cluster:-

10.10.1.58 - Head node + Client + Worker node

10.10.1.59 & 10.10.1.60 - Worker nodes



Approach

Dataset :- Road network graph

Vehicle route tracking using traffic cams

- Time-series graph of sync camera snapshots

- Sensors are vertices

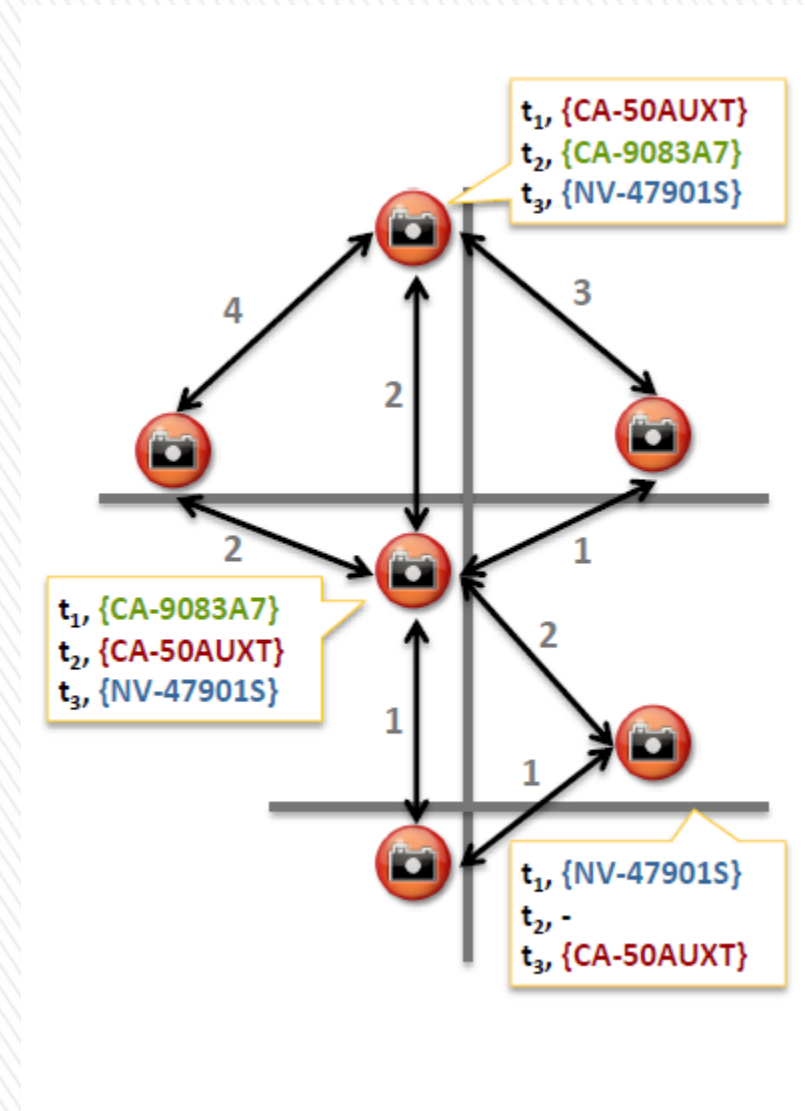
- Edges* are road connectivity w/ distance weight

Graph instance is image metadata every N sec

- License plate, vehicle color, direction, speed

Urban

Algorithms :- Vert-count, Connected components



GoFFISH Cluster : Loggers

There are three Log files .

The Subgraph log contains information about the lifecycle of the app.

The Partition log contains the time for each of the steps as well the total times for Compute , IO and both.

The Container log is a combination of the other two.

Dataset Information: CA Road Network

Number of instances : 4

Number of Partitions : 3

Algorithm : Connected Components

Dataset statistics

Nodes	1965206
Edges	2766607



Results obtained from Logger

Before setting the slicing pointers

```
PERF,1,,0,3,1430287604262,2015-04-29 11:36:44.262+0530,ALL_SG_COMPUTE_TASKS,0
PERF,1,0,3,1430287604262,2015-04-29 11:36:44.262+0530,PERF.PART.SS_WALL_TIME,1430287604256,1430287604262,6
STATE,1,0,4,1430287604397,2015-04-29 11:36:44.397+0530,STATE.PART.START_SS
STATE,1,0,4,1430287604397,2015-04-29 11:36:44.397+0530,STATE.PART.PART_VOTE_HALT
STATE,1,0,4,1430287604402,2015-04-29 11:36:44.402+0530,STATE.PART.END_SS
PERF,1,,0,4,1430287604402,2015-04-29 11:36:44.402+0530,ALL_SG_COMPUTE_TASKS,0
PERF,1,0,4,1430287604402,2015-04-29 11:36:44.402+0530,PERF.PART.SS_WALL_TIME,1430287604397,1430287604402,5
STATE,1,0,4,1430287604402,2015-04-29 11:36:44.402+0530,STATE.PART.HALT
PERF,1,0,4,1430287604403,2015-04-29 11:36:44.403+0530,PERF.PART.TOTAL_WALL_TIME,1430287600063,1430287604403,4399
PERF,1,,0,4,1430287604403,2015-04-29 11:36:44.403+0530,PERF.PART.TOTAL_SG_COMPUTE_TASK_DURATION,601
PERF,1,,0,4,1430287604403,2015-04-29 11:36:44.403+0530,PERF.PART.TOTAL_SG_COMPUTE_SEQ_DURATION,0
PERF,1,,0,4,1430287604404,2015-04-29 11:36:44.404+0530,PERF.PART.TOTAL_SG_MSG_SEQ_DURATION,0
STATE,1,0,0,1430287604404,2015-04-29 11:36:44.404+0530,STATE.PART.STOP
```

After setting the slicing pointers

```
PERF,1,,0,3,1430288257812,2015-04-29 11:47:37.812+0530,PERF.PART.SG_MSG_SEQ_DURATION,0
STATE,1,0,4,1430288257929,2015-04-29 11:47:37.929+0530,STATE.PART.START_SS
STATE,1,0,4,1430288257929,2015-04-29 11:47:37.929+0530,STATE.PART.PART_VOTE_HALT
STATE,1,0,4,1430288257934,2015-04-29 11:47:37.934+0530,STATE.PART.END_SS
PERF,1,,0,4,1430288257934,2015-04-29 11:47:37.934+0530,ALL_SG_COMPUTE_TASKS,0
PERF,1,0,4,1430288257934,2015-04-29 11:47:37.934+0530,PERF.PART.SS_WALL_TIME,1430288257929,1430288257934,5
STATE,1,0,4,1430288257935,2015-04-29 11:47:37.935+0530,STATE.PART.HALT
PERF,1,0,4,1430288257936,2015-04-29 11:47:37.936+0530,PERF.PART.TOTAL_WALL_TIME,1430288253517,1430288257936,4419
PERF,1,,0,4,1430288257936,2015-04-29 11:47:37.936+0530,PERF.PART.TOTAL_SG_COMPUTE_TASK_DURATION,628
PERF,1,,0,4,1430288257936,2015-04-29 11:47:37.936+0530,PERF.PART.TOTAL_SG_COMPUTE_SEQ_DURATION,0
PERF,1,,0,4,1430288257936,2015-04-29 11:47:37.936+0530,PERF.PART.TOTAL_SG_MSG_SEQ_DURATION,0
STATE,1,0,0,1430288257936,2015-04-29 11:47:37.936+0530,STATE.PART.STOP
```



Screenshots

```
bhavani@top-code: ~/goffish/deployment/goffish_home/gofs-2.0/bin
bhavani@top-code:~$ ssh localhost
Welcome to Ubuntu 14.04.2 LTS (GNU/Linux 3.13.0-49-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Tue Apr 28 00:08:11 2015 from localhost
bhavani@top-code:~$ cd goffish/deployment/goffish_home/gofs-2.0/bin/
bhavani@top-code:~/goffish/deployment/goffish_home/gofs-2.0/bin$ ./StartNamenode.sh
Loaded name node information from file.
Apr 28, 2015 12:18:06 AM com.sun.jersey.api.core.PackagesResourceConfig init
INFO: Scanning for root resource and provider classes in the packages:
    edu.usc.goffish.gofs.namenode.resources
Apr 28, 2015 12:18:06 AM com.sun.jersey.api.core.ScanningResourceConfig logClasses
INFO: Root resource classes found:
    class edu.usc.goffish.gofs.namenode.resources.SerializerResource
    class edu.usc.goffish.gofs.namenode.resources.AvailableResource
    class edu.usc.goffish.gofs.namenode.resources.DataNodesResource
    class edu.usc.goffish.gofs.namenode.resources.DirectoryResource
Apr 28, 2015 12:18:06 AM com.sun.jersey.api.core.ScanningResourceConfig init
INFO: No provider classes found.
Apr 28, 2015 12:18:06 AM com.sun.jersey.server.impl.application.WebApplicationImpl _initiate
INFO: Initiating Jersey application, version 'Jersey: 1.17.1 02/28/2013 12:47 PM'
Apr 28, 2015 12:18:07 AM com.sun.jersey.spi.inject.Errors processErrorMessages
WARNING: The following warnings have been detected with resource and/or provider classes:
    WARNING: A sub-resource method, public javax.ws.rs.core.Response edu.usc.goffish.gofs.namenode.resources.DirectoryResource.getGraphs(), with U
    RI template, "", is treated as a resource method
Apr 28, 2015 12:18:07 AM org.glassfish.grizzly.http.server.NetworkListener start
INFO: Started listener bound to [localhost:9998]
Apr 28, 2015 12:18:07 AM org.glassfish.grizzly.http.server.HttpServer start
INFO: [HttpServer] Started.
```

```
bhavani@top-code: ~/goffish/deployment/goffish_home/gofs-2.0/bin
*****
METIS 5.0 Copyright 1998-13, Regents of the University of Minnesota
(HEAD: , Built on: Dec 20 2014, 19:02:52)
size of idx_t: 32bits, real_t: 32bits, idx_t *: 64bits

Graph Information -----
Name: /tmp/gofs_metis441336343067469649/input.metis, #Vertices: 34546, #Edges: 420877, #Parts: 4

Options -----
ptype=kway, objtype=cut, ctype=shem, rtype=greedy, iptype=metisrb
dbgvlvl=0, ufactor=1.030, no2hop=NO, minconn=NO, contig=NO, nooutput=NO
seed=-1, niter=10, ncuts=1

Direct k-way Partitioning -----
- Edgecut: 39730, communication volume: 21552.

- Balance:
    constraint #0: 1.015 out of 0.000

- Most overweight partition:
    pid: 3, actual: 8763, desired: 8636, ratio: 1.01.

- Subdomain connectivity: max: 3, min: 3, avg: 3.00

- The original graph had 61 connected components and the resulting
  partitioning after removing the cut edges has 65 components.
Timing Information -----
I/O:                0.033 sec
Partitioning:        0.085 sec  (METIS time)
Reporting:           0.024 sec

Memory Information -----
Max memory used:     18.875 MB
*****
loading metis output... [51ms]
partitioning finished [694ms]
building partitions...
partitioning template... [449ms writing] [1108ms]
collating partitions...
```

Screenshots

The image displays two screenshots from a Linux desktop environment. The left screenshot shows a terminal window with the following output:

```
bhavani@top-code: ~/goffish/deployment/gopher-client-2.0/bin
partition 4 mapped to file://localhost/home/bhavani/goffish/deployment/goffish_home/gofs-data/gofs/
writing partition... [796 KB]
moving partition 4 to file://localhost/home/bhavani/goffish/deployment/goffish_home/gofs-data/gofs/slices...
executing: "scp -B -q -r -p /tmp/gofs_scpdist319202962346174584/5dc7fdcf-9202-4561-a34c-441bdb27781a.slc /tmp/gofs_scpdist319202962346174584/1d6d39f2-1ece-4417-b8cb-e7596c1a3a77.slc localhost:/home/bhavani/goffish/deployment/goffish_home/gofs-data/gofs/slices"
distribution finished [240ms]
**total subgraphs: 64
finished [total 8103ms]
bhavani@top-code:~/goffish/deployment/goffish_home/gofs-2.0/bin$ cd ../../
bhavani@top-code:~/goffish/deployment/goffish_home$ cd ..
bhavani@top-code:~/goffish/deployment$ cd gopher-client-2.0/bin/
bhavani@top-code:~/goffish/deployment/gopher-client-2.0/bin$ ./setup-gopher.sh
coping /home/bhavani/goffish/deployment/samples/gopher-jars/vert-count-2.0.jar to bhavani@localhost:~/
vert-count-2.0.jar
Moving the jar vert-count-2.0.jar to Gopher server location
Starting manager @ localhost
Starting coordinator @ localhost
waiting...
startin containers ...
Starting data node container @localhost
ssh localhost 'cd '/home/bhavani/goffish/deployment/goffish_home'/gopher-server-2.0/bin;./container.sh localhost 45001 fbgraph file://localhost/home/bhavani/goffish/deployment/goffish_home/gofs-data/gofs >container.out 2>&1 &'
Using HOME DIR /home/bhavani/goffish/deployment/gopher-client-2.0
Using java version:
java version "1.7.0_79"
OpenJDK Runtime Environment (IcedTea 2.5.5) (7u79-2.5.5-0ubuntu0.14.04.2)
OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
log4j:WARN No appenders could be found for logger (org.apache.commons.configuration.ConfigurationUtils).
log4j:WARN Please initialize the log4j system properly.
bhavani@top-code:~/goffish/deployment/gopher-client-2.0/bin$ ./RunGopherClient.sh
Using HOME DIR /home/bhavani/goffish/deployment/gopher-client-2.0
Using java version:
java version "1.7.0_79"
OpenJDK Runtime Environment (IcedTea 2.5.5) (7u79-2.5.5-0ubuntu0.14.04.2)
OpenJDK 64-Bit Server VM (build 24.79-b02, mixed mode)
log4j:WARN No appenders could be found for logger (org.apache.commons.configuration.ConfigurationUtils).
log4j:WARN Please initialize the log4j system properly.
Sending Init message : vert-count-2.0.jar,edu.usc.pgroupp.goffish.gopher.sample.VertCounter,1,fbgraph,http://localhost:9998
```

The right screenshot shows a text editor window (gedit) with the following content:

```
RunGoFSDeploy.sh x vert-count.txt x vert-count.txt x
1 Start Time :83388928631006
2 End Time :83388928794627
3 Difference :163621
4 Total Vertex Count :34546
```

The bottom status bar of the terminal window shows: Plain Text ▾ Tab Width: 8 ▾ Ln 3, Col 19 INS

Challenges

- Single Node Installation
 - The pom links to some packages were missing
 - we manually ran every step in the script with new links to the packages.
- Understanding bin packing schemes and how it affect our storage format
- Identifying the slicing pointers from the GoFFish code base.
- Cluster Installation
 - Java shared libraries (libjli.so) were initially not found



Literature Survey

- ▶GoFFish: A Sub-Graph Centric Framework for Large-Scale Graph Analytics - Indian Institute of Science, Bangalore 560012 India, University of Southern California, Los Angeles CA 90089 USA, November 26, 2013
- ▶Scalable Analytics over Distributed Time-series Graphs using GoFFish - Indian Institute of Science, Bangalore 560012 India, University of Southern California, Los Angeles CA 90089 USA, June 23, 2014
- ▶Chronos: A Graph Engine for Temporal Graph Analysis - Tsinghua University, University of Science and Technology of China, Microsoft Research



Thank you

