namespace java com.twitter.interaction\_graph.thriftjava

#@namespace scala com.twitter.interaction\_graph.thriftscala

#@namespace strato com.twitter.interaction\_graph

// These could be either a Vertex or an edge feature name

// when you add a new feature, update VertexFeatureCombiner.java and EdgeFeatureCombiner.java.

enum FeatureName {

num\_retweets = 1

num\_favorites = 2

num\_mentions = 3

num\_direct\_messages = 4

num\_tweet\_clicks = 5

num\_link\_clicks = 6

num\_profile\_views = 7

num\_follows = 8

num\_unfollows = 9

num\_mutual\_follows = 10

address\_book\_email = 11

address\_book\_phone = 12

address\_book\_in\_both = 13

address\_book\_mutual\_edge\_email = 14

address\_book\_mutual\_edge\_phone = 15

address\_book\_mutual\_edge\_in\_both = 16

total\_dwell\_time = 17

num\_inspected\_statuses = 18

num\_photo\_tags = 19

num\_blocks = 20

num\_mutes = 21

num\_report\_as\_abuses = 22

num\_report\_as\_spams = 23

num\_tweet\_quotes = 24

num\_push\_opens = 25

num\_ntab\_clicks = 26,

num\_rt\_favories = 27,

num\_rt\_replies = 28,

num\_rt\_tweet\_quotes = 29,

num\_rt\_retweets = 30,

num\_rt\_mentions = 31,

num\_rt\_tweet\_clicks = 32,

num\_rt\_link\_clicks = 33

num\_shares = 34,

num\_email\_click = 35,

num\_email\_open = 36,

num\_ntab\_dislike\_7\_days = 37,

num\_push\_dismiss = 38,

num\_push\_report\_tweet\_click = 39,

num\_push\_report\_user\_click = 40,

num\_replies = 41,

// vertex features after 128

num\_create\_tweets = 129,

}

// do remember to update the tests in InteractionGraphAggregationJobTest when adding new features but not updating agg\_all

struct TimeSeriesStatistics {

1: required double mean;

// For computing variance online: http://en.wikipedia.org/wiki/Algorithms\_for\_calculating\_variance#On-line\_algorithm

2: required double m2\_for\_variance;

3: required double ewma; // Exponentially weighted moving average: ewma\_t = \alpha x\_t + (1-\alpha) ewma\_{t-1}

4: required i32 num\_elapsed\_days; // Total number of days since we started counting this feature

5: required i32 num\_non\_zero\_days; // Number of days when the interaction was non-zero (used to compute mean/variance)

6: optional i32 num\_days\_since\_last; // Number of days since the latest interaction happen

}(persisted="true", hasPersonalData = 'false')

struct VertexFeature {

1: required FeatureName name;

2: required bool outgoing; // direction e.g. true is num\_retweets\_by\_user, and false is num\_retweets\_for\_user

3: required TimeSeriesStatistics tss;

}(persisted="true", hasPersonalData = 'false')

struct Vertex {

1: required i64 user\_id(personalDataType = 'UserId');

2: optional double weight;

3: list<VertexFeature> features;

}(persisted="true", hasPersonalData = 'true')

/\*

\* These features are for an edge (a->b). Examples:

\* (i) follow is whether a follows b

\* (ii) num\_retweets is number of b's tweets retweet by a

\*/

struct EdgeFeature {

1: required FeatureName name;

2: required TimeSeriesStatistics tss;

}(persisted="true", hasPersonalData = 'false')

struct Edge {

1: required i64 source\_id(personalDataType = 'UserId');

2: required i64 destination\_id(personalDataType = 'UserId');

3: optional double weight;

4: list<EdgeFeature> features;

}(persisted="true", hasPersonalData = 'true')

// these structs below are used by our ml pipeline

struct EdgeLabel {

1: required i64 source\_id(personalDataType = 'UserId');

2: required i64 destination\_id(personalDataType = 'UserId');

3: required set<FeatureName> labels(personalDataType = 'AggregateImpressionEngagementData');

}(persisted="true", hasPersonalData = 'true')