PORTAL

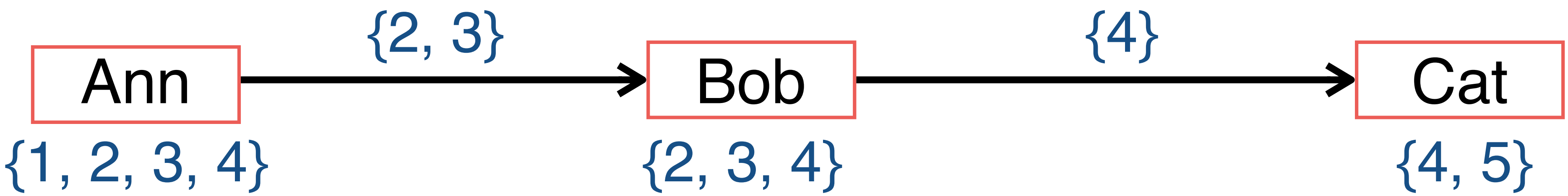
Querying Billion-Edge Evolving Property Graphs with Portal

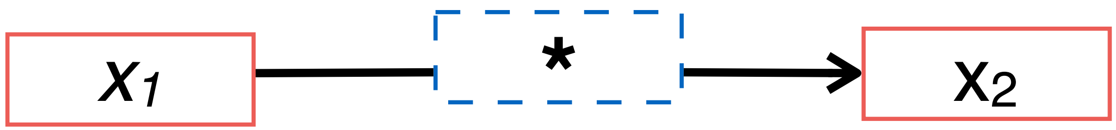
Julia Stoyanovich
New University, USA

stoyanovich@nyu.edu
@stoyanoj

portalDB.github.io

Is there a path from Ann to Cat?





snapshot reducibility

in a snapshot? (**TNGP**)

$$\begin{tikzcd}
r_1, \ldots, r_n \arrow[r, "\psi^T"] \arrow[d, "\text{timeslice at } t"'] & \psi^T(r_1, \ldots, r_n) \arrow[d, "\text{timeslice at } t"] \\
\tau_t(r_1), \ldots, \tau_t(r_n) \arrow[r, "\psi"'] & \psi\left(\tau_t(r_1), \ldots, \tau_t(r_n)\right)
\end{tikzcd}$$

Is there a path from Ann to Cat?

PORTAL

$$\underset{\{1, 2, 3, 4\}}{\boxed{\text{Ann}}} \xrightarrow{\{2, 3\}} \underset{\{2, 3, 4\}}{\boxed{\text{Bob}}} \xrightarrow{\{4\}} \underset{\{4, 5\}}{\boxed{\text{Cat}}}$$

$\boxed{x_1} \longrightarrow \boxed{*} \longrightarrow \boxed{x_2}$

in a snapshot? (**TNGP**)

$\boxed{x_1} \longrightarrow \boxed{\leq^*} \longrightarrow \boxed{x_2}$

over time? (**journey**)

snapshot reducibility

+

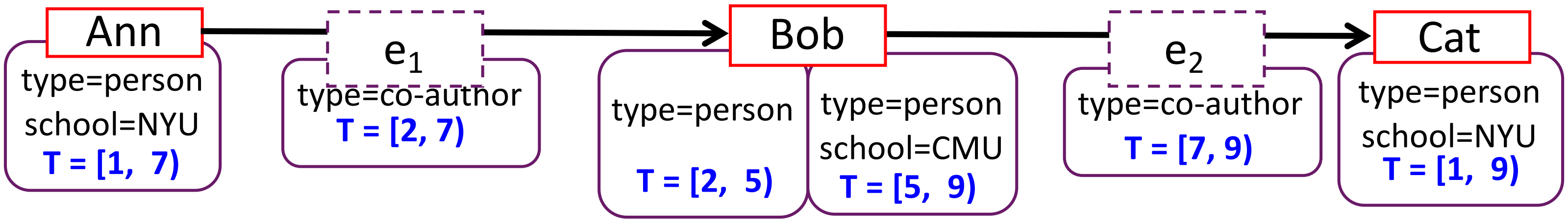
time as data

=

point-based semantics

TGraph: evolving property graph

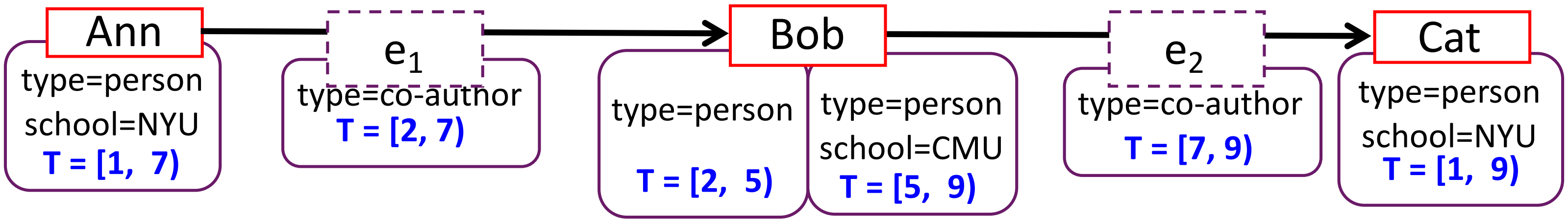
TGraph: evolving property graph

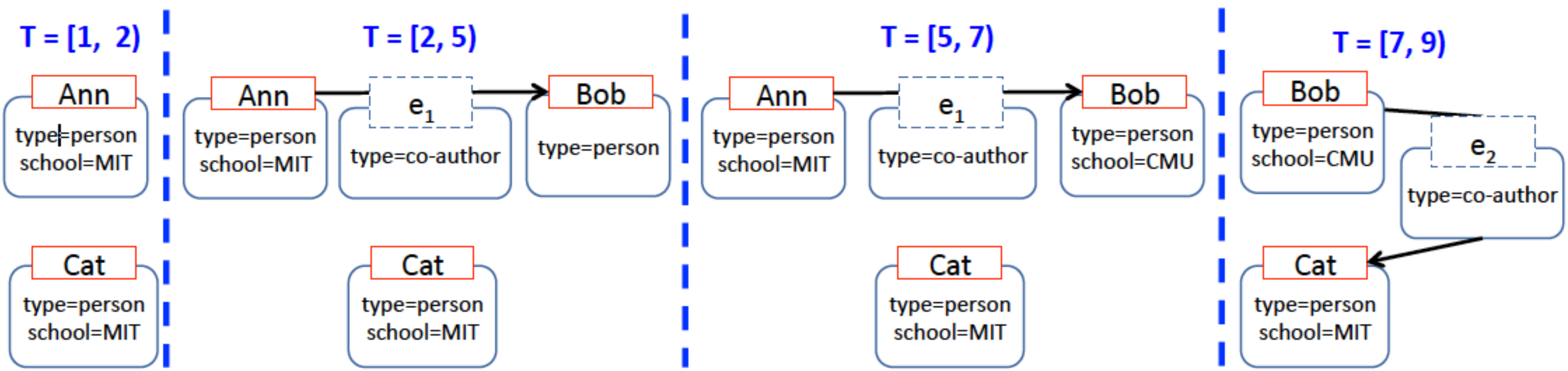


- **Definition 1.** TGraph is a six-tuple $\mathcal{G} = (V, E, L, \rho, \xi^T, \lambda^T)$, where
  - V is a finite set of *nodes* (or *vertices*), E is a finite set of *edges*, $V \cap E = \emptyset$;
  - L is a finite set of property labels;
  - $\rho : E \to (V \times V)$ is a total function that maps an edge to its source and destination nodes;
  - $\xi^T : (V \cup E) \times \Omega^T \to B$ is a total function that maps a node or an edge, and time point, to a Boolean, indicating existence of that node or edge; and
  - $\lambda^T : (V \cup E) \times L \times \Omega^T \to val$ is a partial function that maps a node or an edge, a property label, and a time point, to a property value.

[Moffitt, Stoyanovich, **DBPL 2017**]

TGraph: evolving property graph

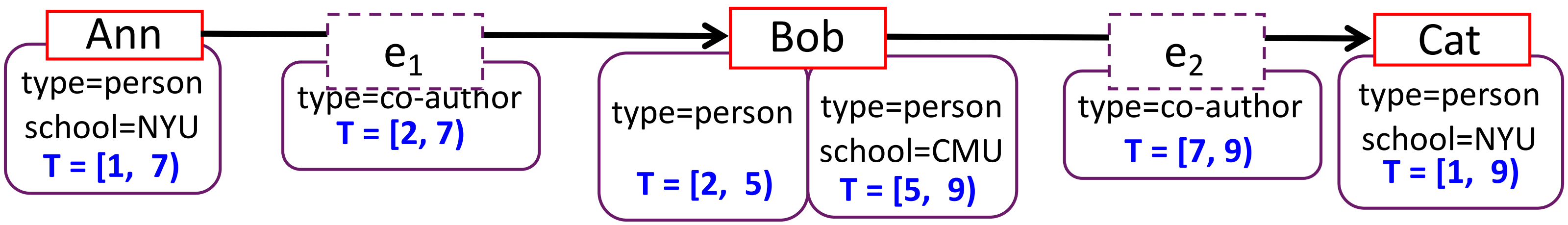




[Moffitt, Stoyanovich, **DBPL 2017**]

TGraph algebra (TGA)

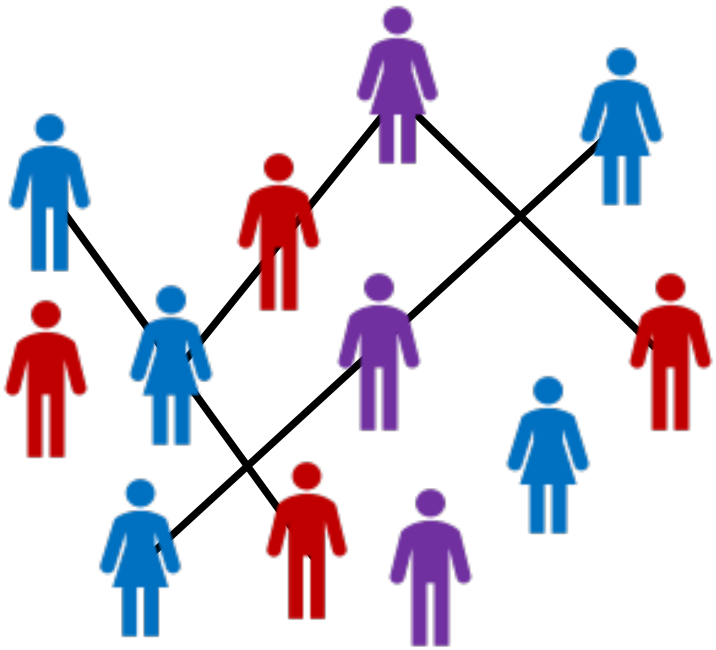
Temporal Graph Algebra (TGA)

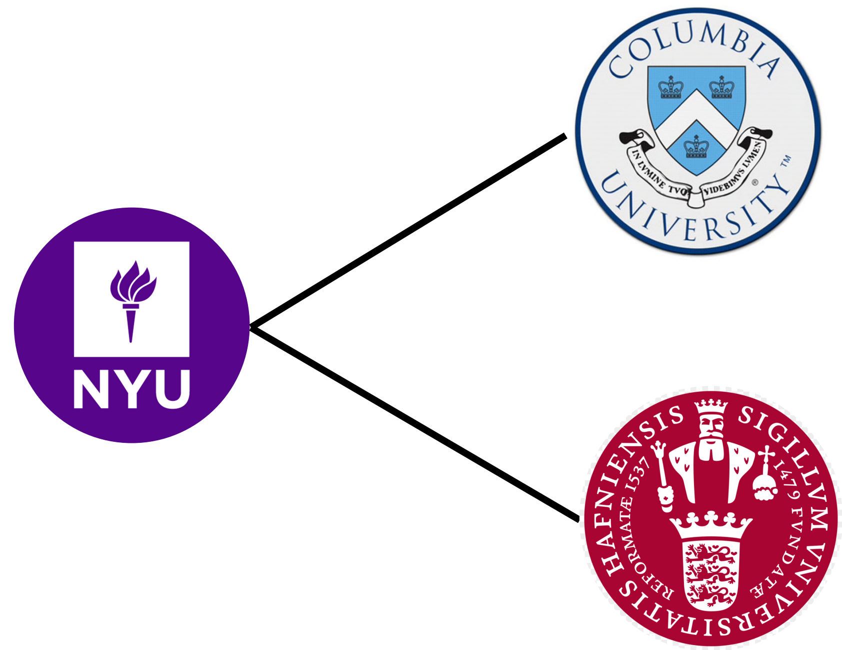


- Operators
  - temporal variants of standard graph operators: union, intersection, difference, slice, subgraph, filter, Pregel-style analytics
  - novel operator: temporal window-based zoom
- TGA is **compositional**
- Operations maintain model integrity under point-based semantics

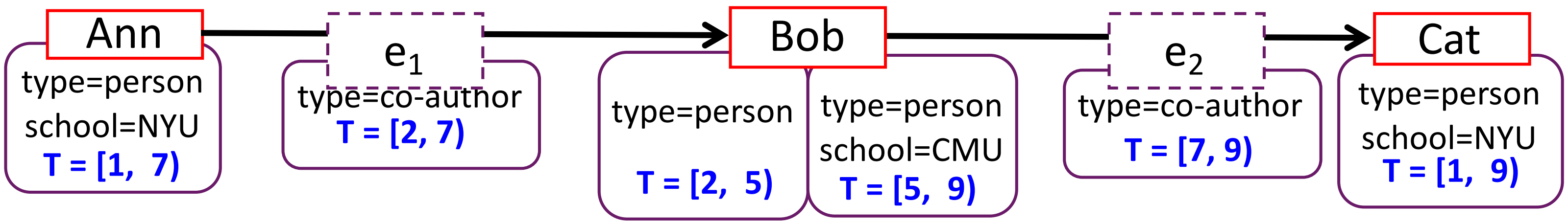
[Moffitt, Stoyanovich, **DBPL 2017**]

Which institutions collaborate?

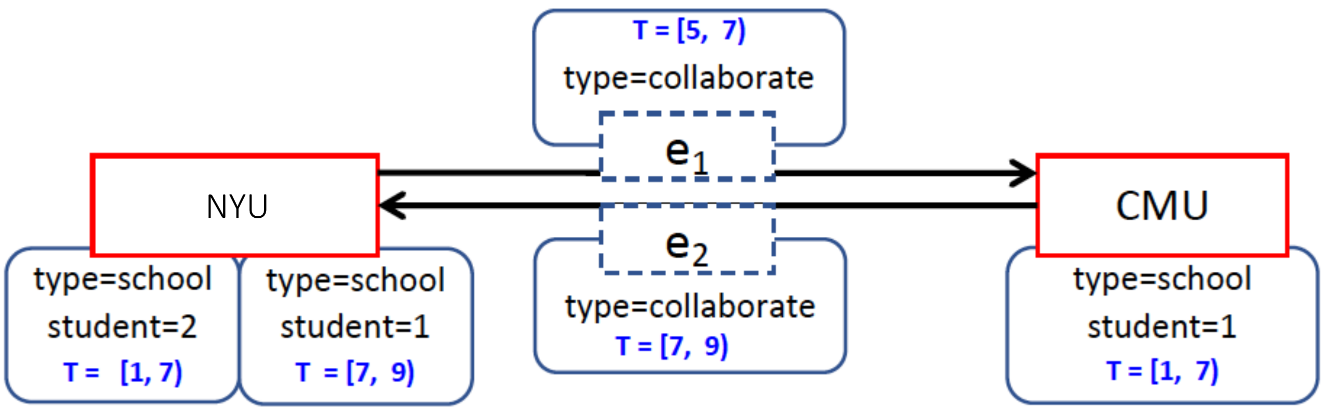




Which institutions collaborate?

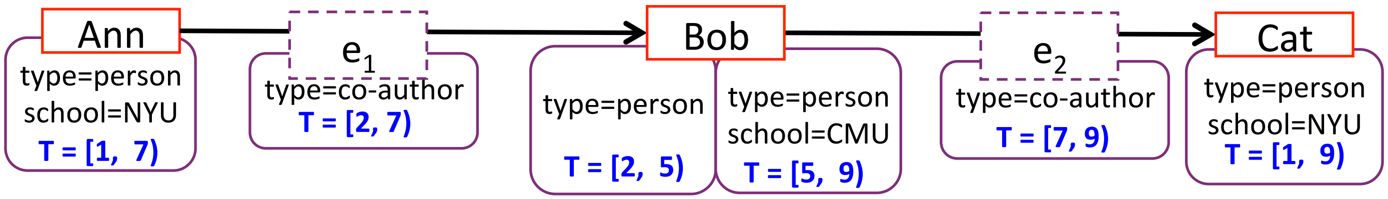


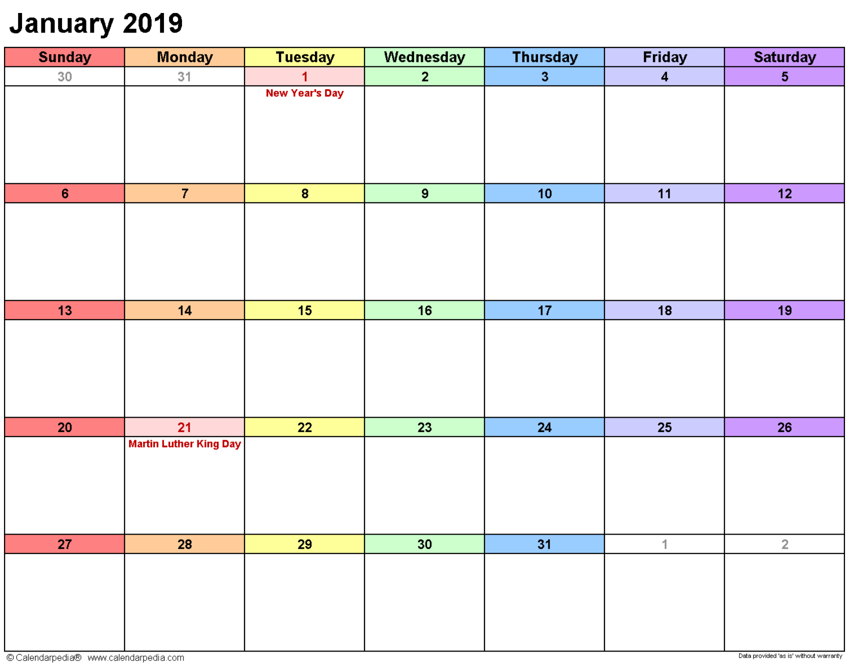
temporal attribute-based zoom

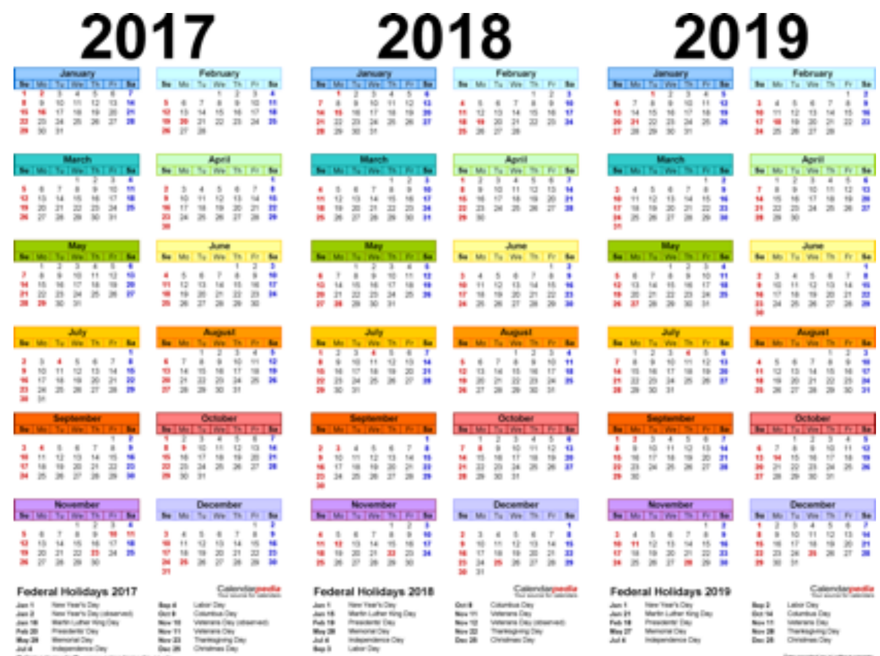


snapshot reducibility

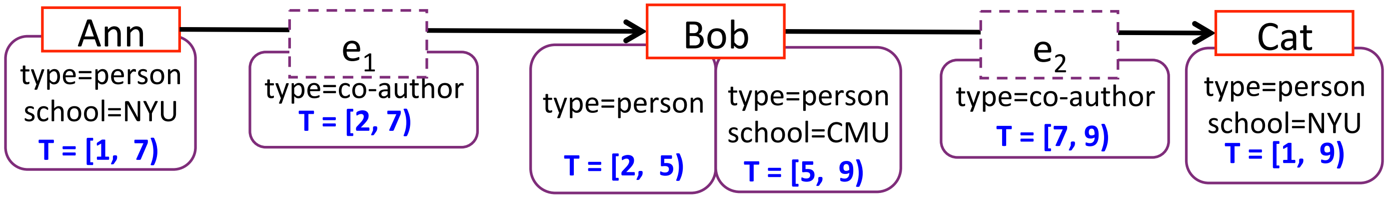
Who is Bob’s BFF?







Who is Bob's BFF?

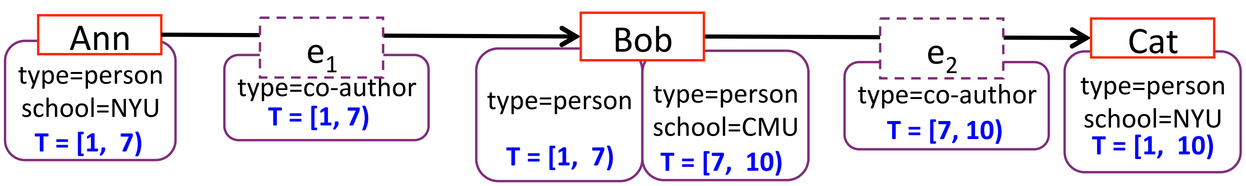


temporal window-based zoom

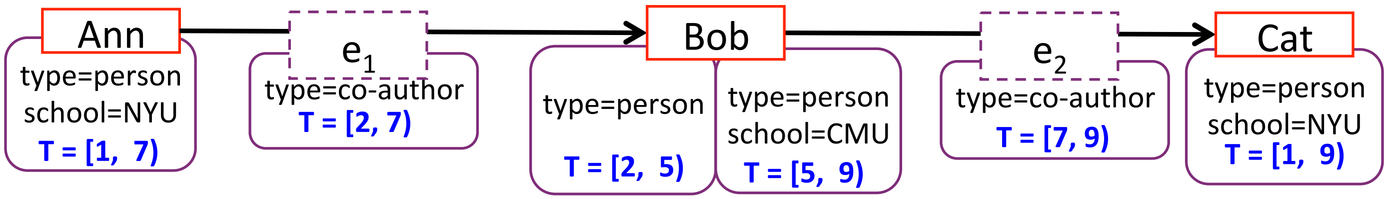
window	points	interval
Q1	1, 2, 3	[1,4)
Q2	4, 5, 6	[4,7)
Q3	7, 8, 9	[7,10)

friend in *some* snapshot during a given quarter

window width=3, nodes=**EXIST**, edges=**EXIST**, node values=**first**, edge values=any



Who is Bob's BFF?

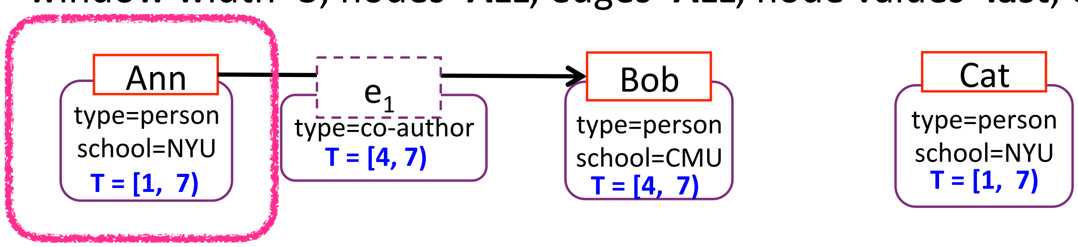


temporal window-based zoom

window	points	interval
Q1	1, 2, 3	[1,4)
Q2	4, 5, 6	[4,7)
Q3	7, 8, 9	[7,10)

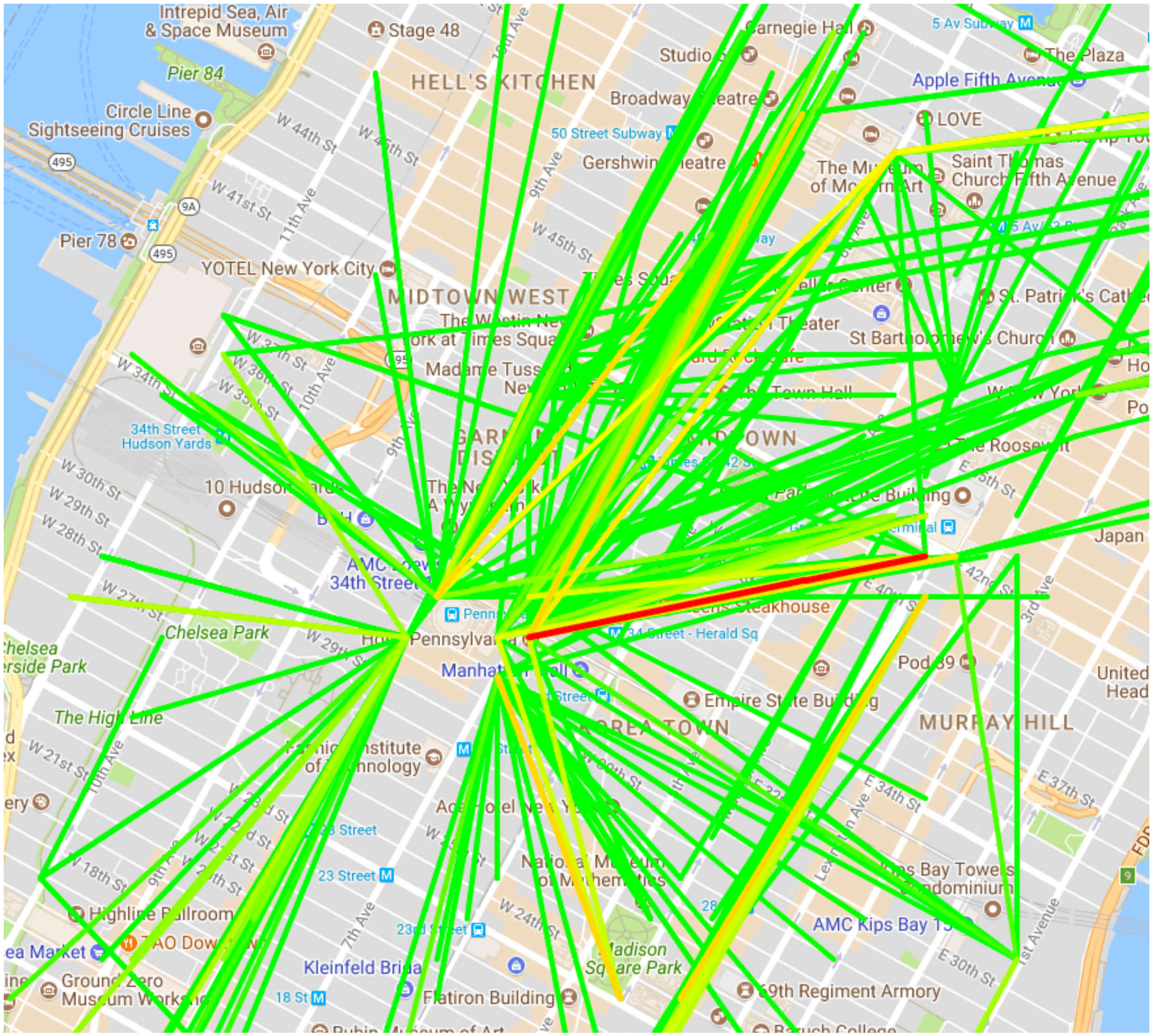
friend in *every* snapshot during a given quarter

window width=3, nodes=**ALL**, edges=**ALL**, node values=**last**, edge values=any



extended snapshot reducibility

Where is my bus?



Most frequent taxi route in NYC: Penn Station to Grand Central

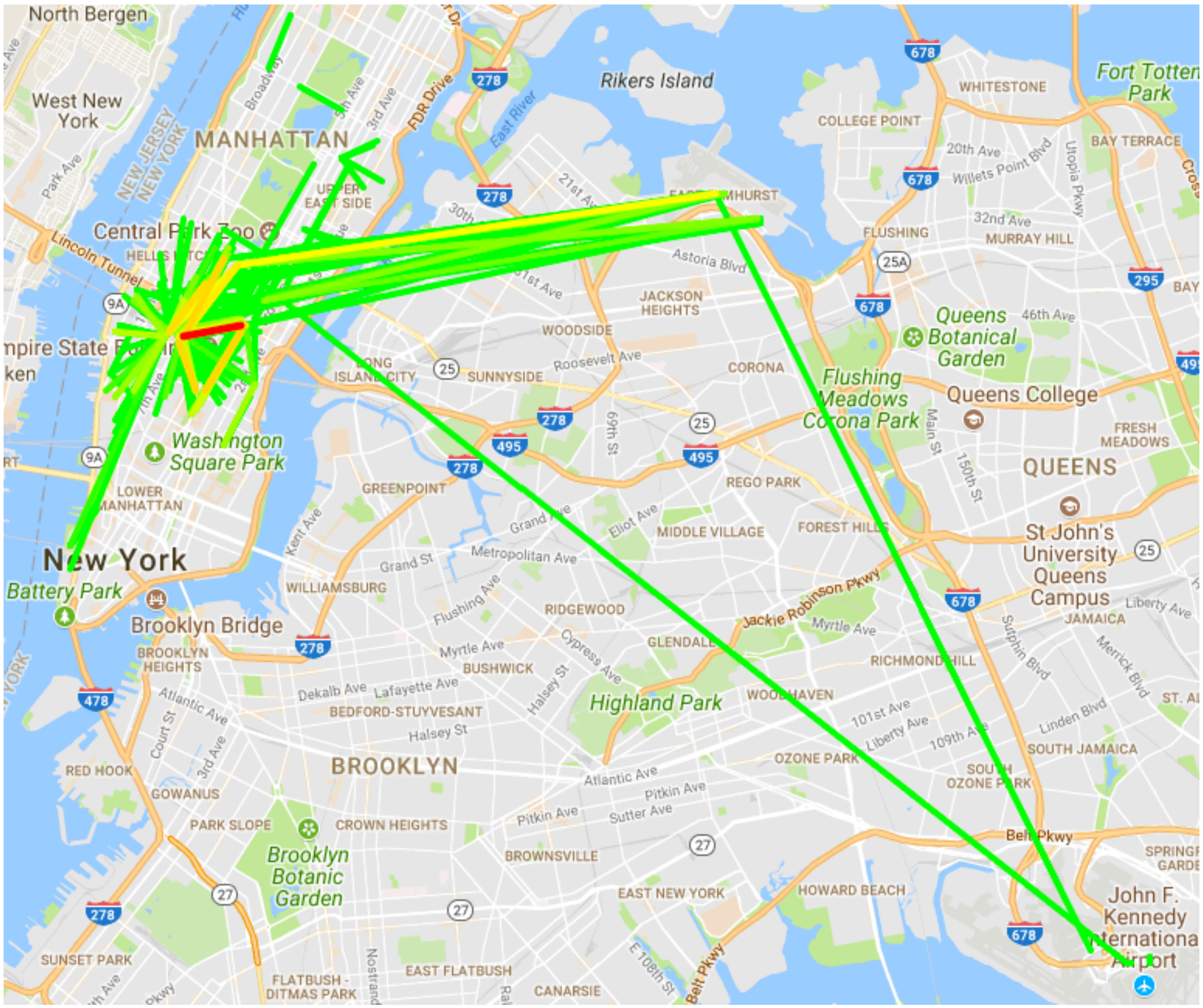
Why?

NYC TLC data

07/2015 - 06/ 2016

Where is my bus?

PORTAL



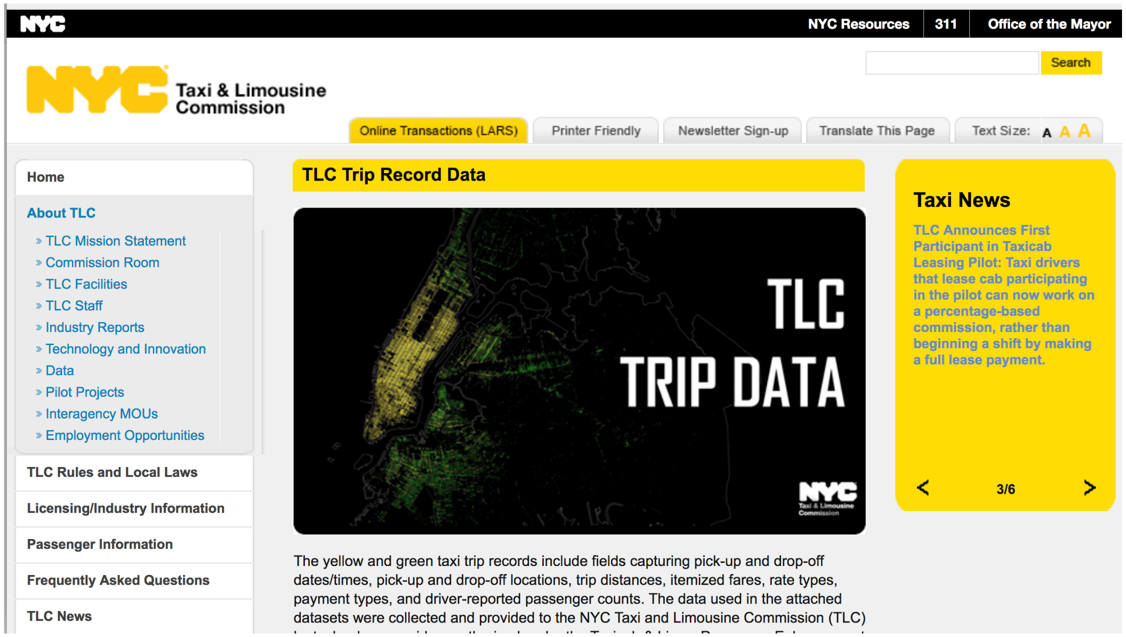
Midtown to airports: LGA is much smaller than JFK, but gets far more taxi traffic.

Why?

NYC TLC data

07/2015 - 06/ 2016

Where is my bus?



- data: pick-up / drop-off time & location, fare, passenger count
- trips represented as a TGraph
  - **nodes** represent locations, with latitude / longitude coordinates as an attribute; a node exists from the time of the first incoming or outgoing trip until the time of the last trip
  - **edges** represent trips, with duration, fare etc as attributes; an edge exists for the duration of the trip

NYC TLC data
07/2015 - 06/ 2016

Portal: implementation

GraphX

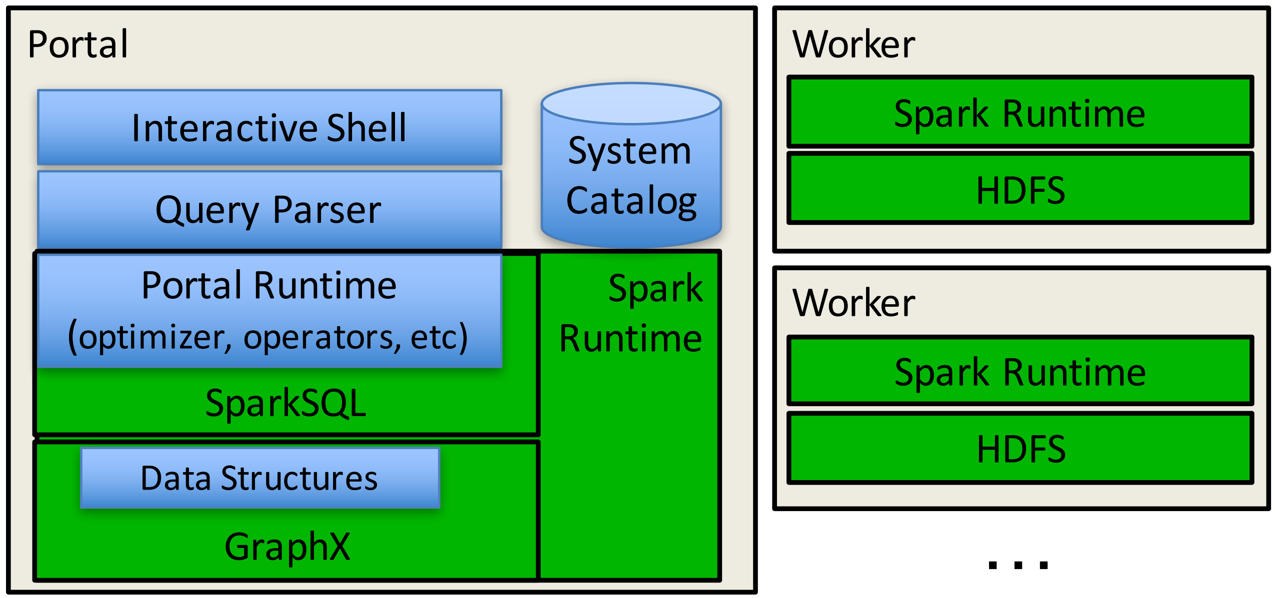
Parquet

APACHE Spark

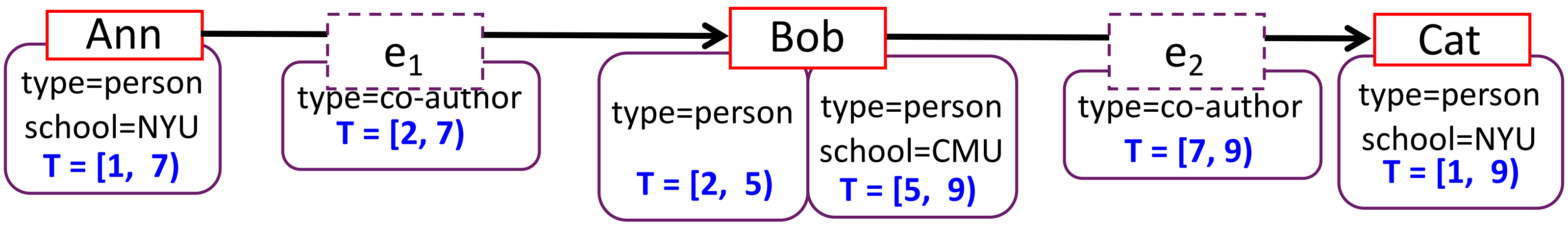
hadoop

goal: principled and systematics support for usable, scalable and extensible analysis of evolving graphs

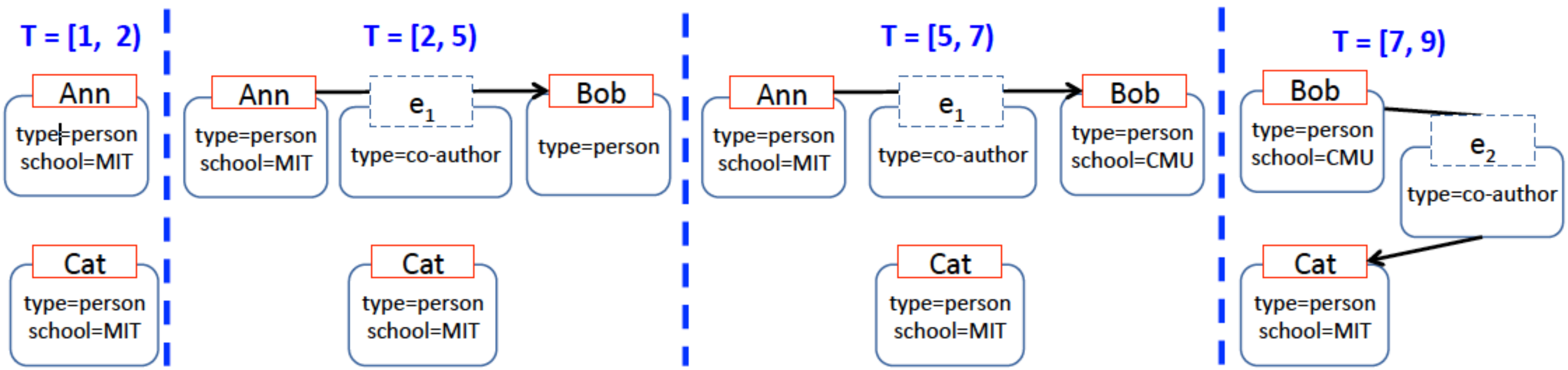
PORTAL


Portal
Interactive Shell
Query Parser
System Catalog
Portal Runtime
(optimizer, operators, etc)
SparkSQL
Spark Runtime
Data Structures
GraphX
Worker
Spark Runtime
HDFS
Worker
Spark Runtime
HDFS
...

TGraph: in-memory representations

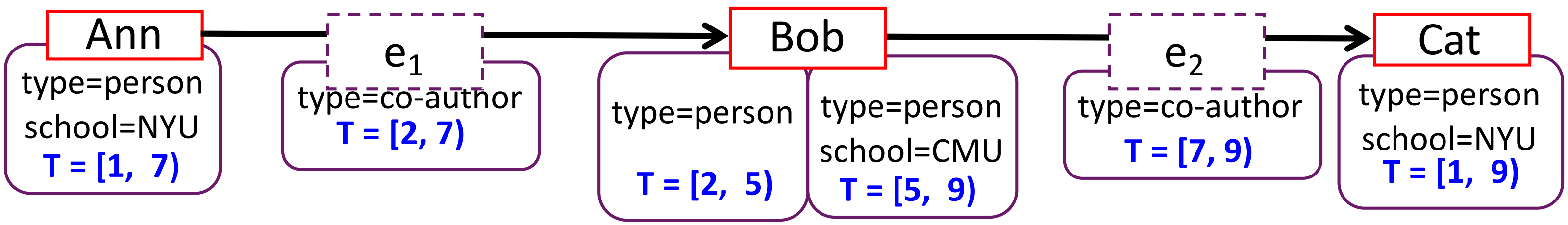


"Representative Graphs" (snapshot sequence)



[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

TGraph: in-memory representations



"Vertex Edge" (nested relational)

Vertices

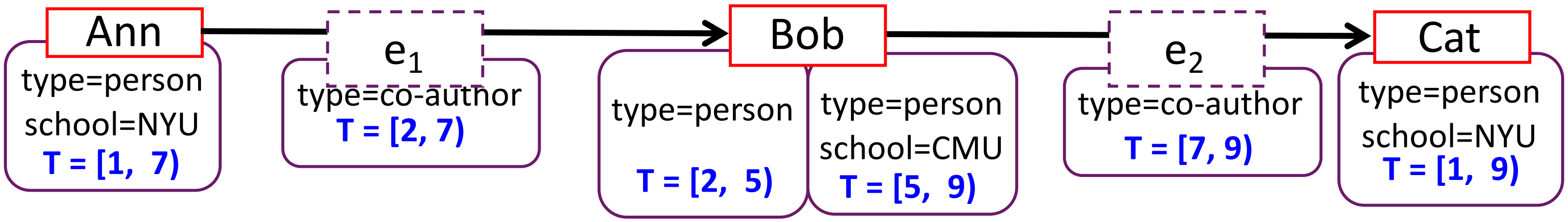
ID	Attributes	T
Ann	Type=person, School=NYU	[1,7)
Bob	Type=person	[2,5)
Bob	Type=person, School=CMU	[5,9)
Cat	Type=person, School=NYU	[1,9)

Edges

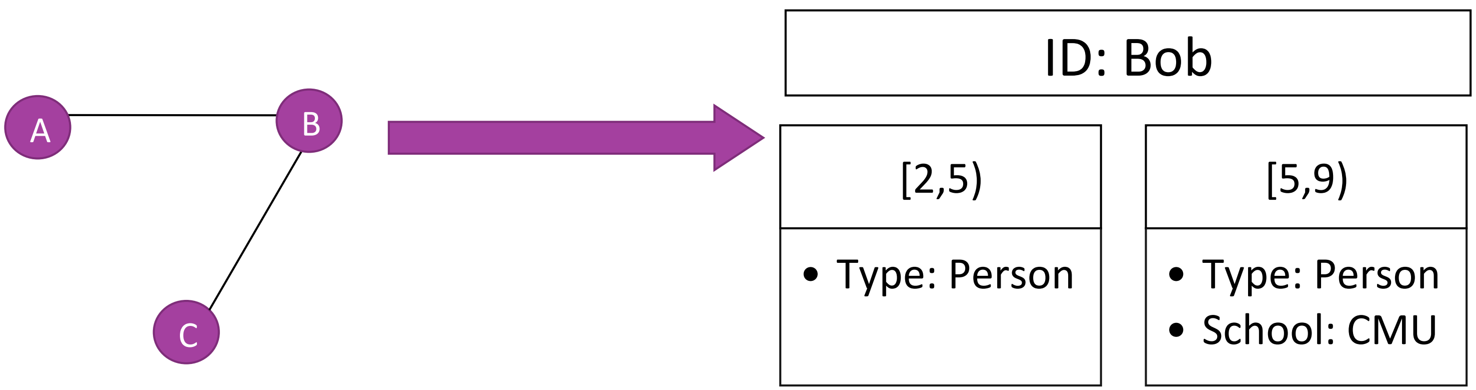
ID	V1	V2	Attributes	T
e1	Ann	Bob	Type=co-author	[2,7]
e2	Bob	Cat	Type=co-author	[7,9)

[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

TGraph: in-memory representations



"One Graph" (GraphX graph)



[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

Performance highlights

Datasets

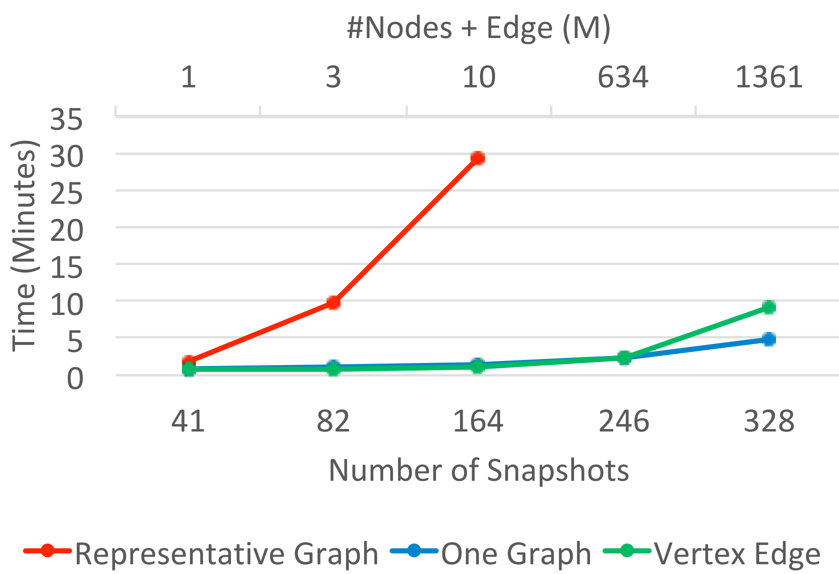
PORTAL

WikiTalk	SNB(LDBC Social Network Benchmark)	nGrams
Communication graph	Friendship graph	Word co-occurrence graph
# of nodes: 2.9 M	# of nodes: 3.3 M	# of nodes: 48 M
# of edges: 10.7M	# of edges: 202 M	# of edges: 1.32B
# of intervals: 179	# of intervals: 36	# of intervals: 328

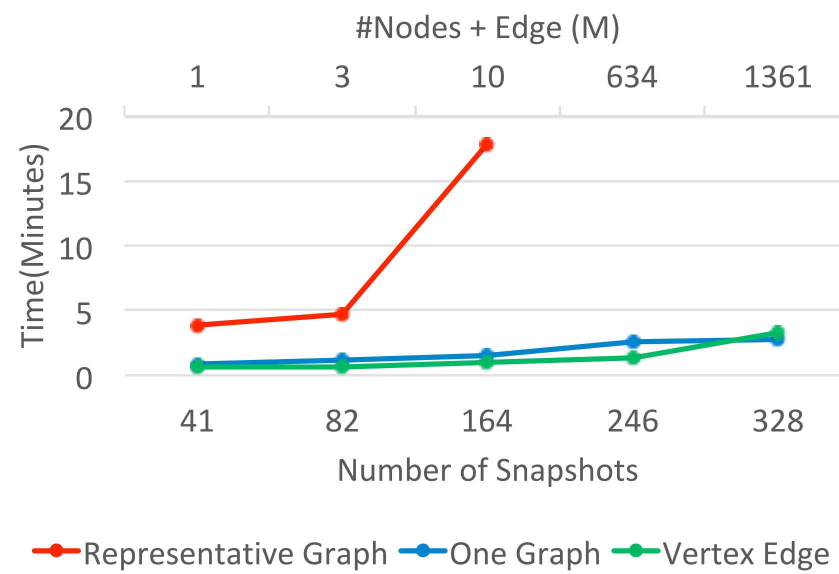
Cluster : 16-workers in-house cluster
Workers: 4 cores and 32 GB of RAM

[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

Zoom on nGrams



temporal attribute-based zoom



temporal window-based zoom

Cluster : 16-workers in-house cluster
Workers: 4 cores and 32 GB of RAM

[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

Wrapping up

Ongoing work

- **TGA / Portal**
  - Declarative language, query optimization
  - Data generation, benchmarking
  - Applications: socioeconomic studies
- **Journeys, temporal regular path queries:** semantics, complexity of evaluation, implementation

Take-aways

- TGraph: a logical model of property graphs with time
- TGA: a compositional temporal graph algebra under point semantics
- Portal: a library on top of Apache Spark, inter-operable with SparkSQL and other libraries
- Performs well on billion-edge graphs with interesting evolution patterns
- NYC Taxi use case, working on others

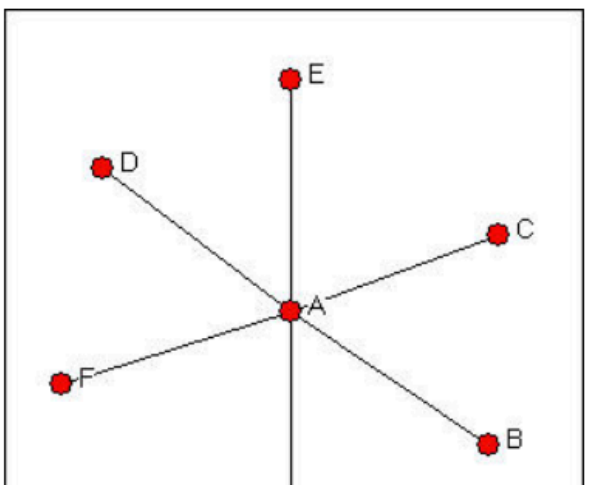
Home | Docs | Install | Publications

PORTAL

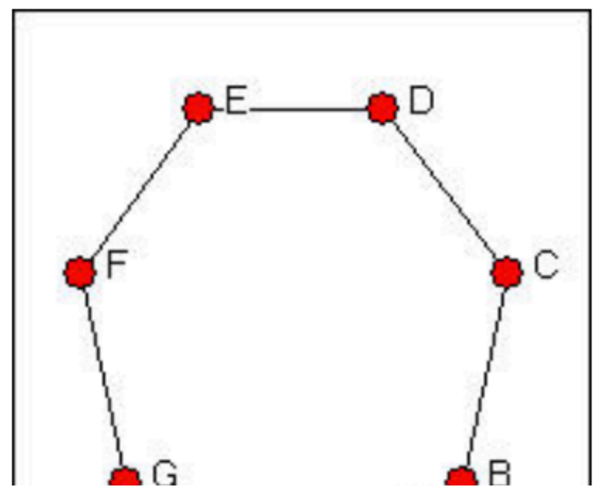
Example: Graph Centrality Over Time

In this example we compute the degree centrality of DBLP over time.
Degree centrality is a simple measure the uniformity of influence in a graph.
For more information, see the definition here.

The most centralized graph is a "star graph". A central node is connected to every other, and every outside node is connected only to the center.



The least centralized graph is a "circle graph", where all nodes have the same exact degree.



PORTAL

Vera Moffitt

Amir Aghasadeghi

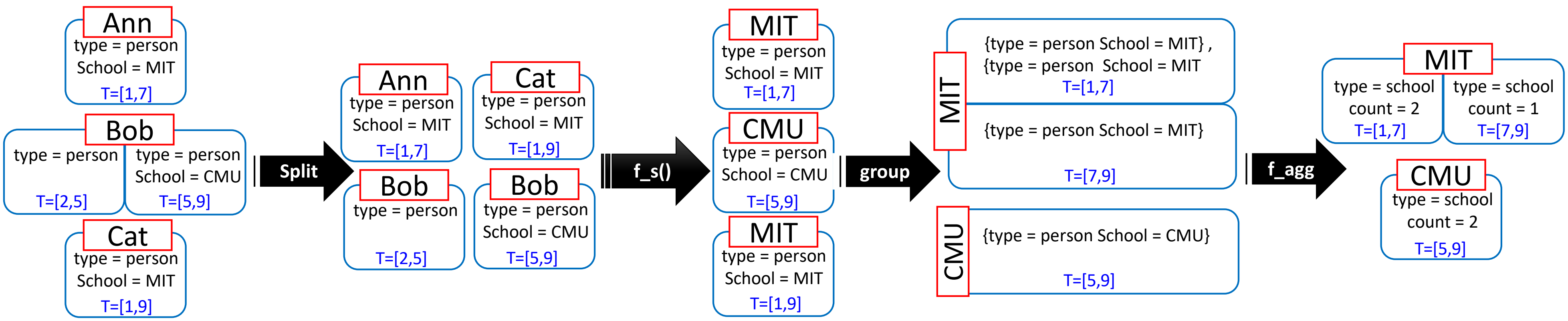
Sebastian Schelter

[CAREER] Querying evolving graphs, 03/2018-

Thank you!

Back-up

aZoom (“One Graph”)



Algorithm 3 aZoom^T over OG

Require: Skolem function $f_s : V \Rightarrow \mathbb{N}$; Aggregation function $f_{agg} : V \times V \Rightarrow V$

$V' \leftarrow V$ **.flatMap**$\{v \Rightarrow$
    $v.history$.map$\{(_, attr) \Rightarrow$
        v.copyWithIdAndAttributes$(f_s(v.vid), attr)$ $\}\}$
.groupBy$\{v \Rightarrow v.vid\}$
.reduce$\{(v_a, v_b) => f_{agg}(v_a, v_b)\}$
$E' \leftarrow E$ **.map**$\{e \Rightarrow$
    $h \leftarrow$ recompute_history(e)
    e.copyWithVidsAndHistory$(f_s(e.v1.vid)$,
        $f_s(e.v2.vid), h)\}$
return new TGraph $G(V', E')$

[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

wZoom ("Vertex Edge")

v	a	T
Bob	{ }	2,3,4
Bob	{ s=CMU }	5,6,7, 8

Align with window
1, 2, 3 ➜ [1,4)
4, 5, 6 ➜ [4,7)
7, 8, 9 ➜ [7,10)

v	a	T
Bob	{ }	[1, 4]
Bob	{ }	[4, 7)
Bob	{ s=CMU }	[4,7)
Bob	{ s=CMU }	[7, 10)

Group by
Id,
interval

v	a	T	r_v
Bob	[{ }]	[1, 4)	0.66
Bob	[{ }, { s=CMU }]	[4, 7) :	1.0
Bob	[{ s=CMU }]	[7, 10)	0.33

Filter by
r_v
Resolve:
last

v	a	T
Bob	{ s=CMU }	[4, 7)

[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]

wZoom (“Vertex Edge”)

PORTAL

Algorithm 5 wZoom^T over VE

Require: resolve functions f_v, f_e; quantifiers r_v, r_e

▷ *Computation of new intervals*
$I' \leftarrow I$.**map**{ $i \Rightarrow (i,$ computeNewInterval$(i))$ }
▷ *Vertex aggregation for new intervals*
$V' \leftarrow V$.**join**(I').**on**{ $(v, (i, n)) \Rightarrow v.n == i$ }
  .**map** { $(v, (i, newInterval)) \Rightarrow$
    v.copyWithNewInterval$(newInterval)$}
  .**groupBy**{ $v \Rightarrow (v.id, v.interval)$ }
  .**filter**{$(i, vertices) \Rightarrow$ match_threshold$(vertices, r_v)$}
  .**reduceByKey**{$((v_a), (v_b)) \Rightarrow f_v(v_a, v_b)$}
▷ *Edge aggregation for new intervals*
$E' \leftarrow E$.**join**(I').**on**{ $(e, (i, n)) \Rightarrow e.interval == n$ }
  .**map** { $(e, (i, newInterval)) \Rightarrow$
    e.copyWithNewInterval$(newInterval)$}
  .**groupBy**{ $e \Rightarrow (e.id, e.interval)$ }
  .**filter**{$(i, edges) \Rightarrow$ match_threshold$(edges, r_e)$}
  .**reduceByKey**{$((e_a), (e_b)) \Rightarrow f_e(e_a, e_b)$}
  if $r_v > r_e$ **then** ▷ *Dangling edge removal*
    $E'' \leftarrow E'$.**semijoin**(V')
     .**on**{ $(e, v) \Rightarrow e.vid1 == v.id$ and in_interval(e, v)}
    $E''' \leftarrow E''$.**semijoin**(V')
     .**on**{ $(e, v) \Rightarrow e.vid2 == v.id$ and in_interval(e, v)}
return new TGraph (V', E''')

[Aghasadeghi, Moffitt, Schelter, Stoyanovich, **EDBT 2020**]