Parallel Processing of Graphs

Bin Shao

Microsoft Research Asia (Beijing, China)

This talk is about graph processing from a pragmatic point of view …

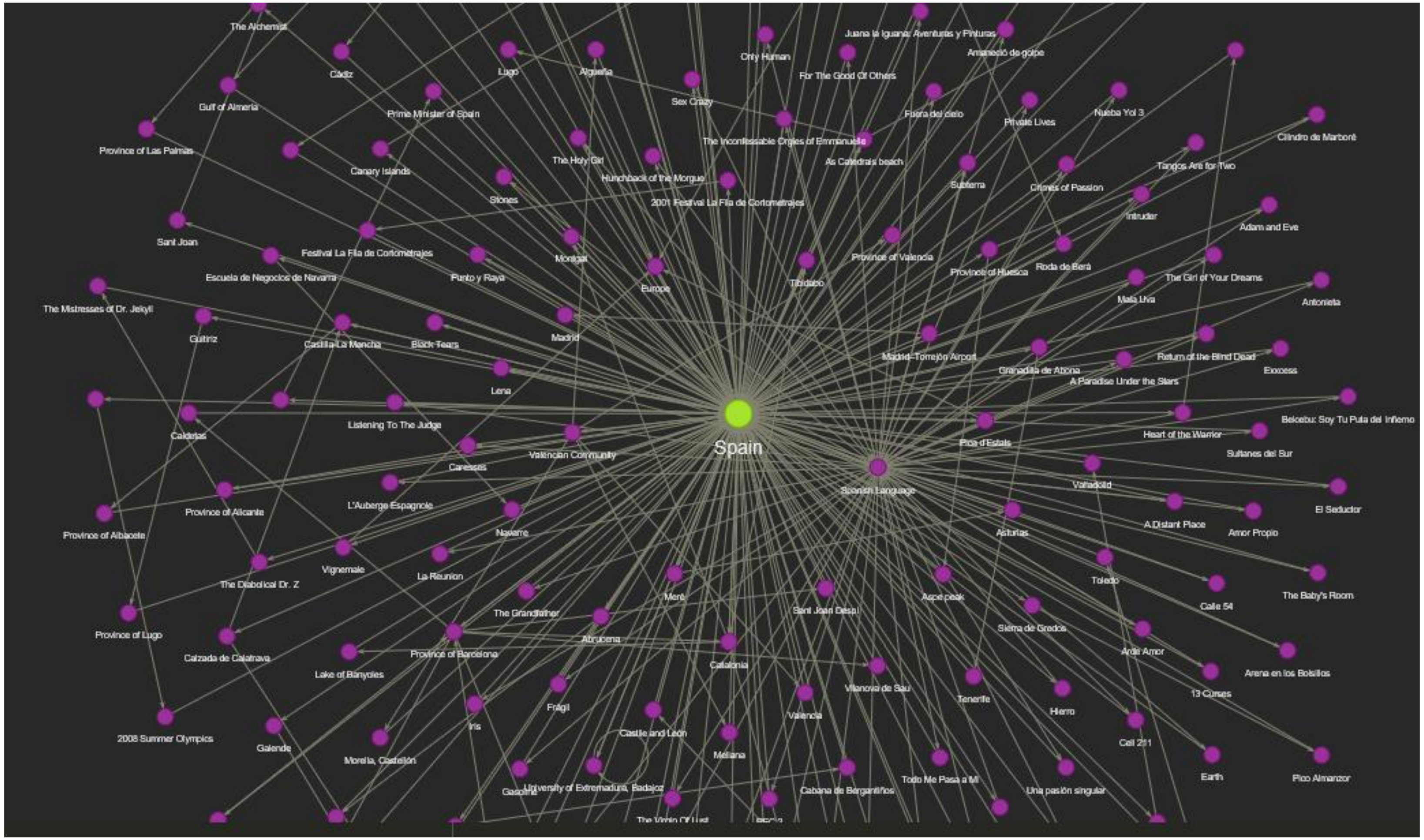
Appetizer

Entities related to Spain

```
KnowledgeGraph
    .StartFrom(519480106787667)
    .VisitNode(_ =>_.continue_if(_.dice(0.1)))
    .VisitNode(_ =>_.continue_if(_.dice(0.1)))
    .VisitNode(_ =>_.return_if(_.has_cell_id(519480106787667)))
```

A graph query example

Entities related to Spain

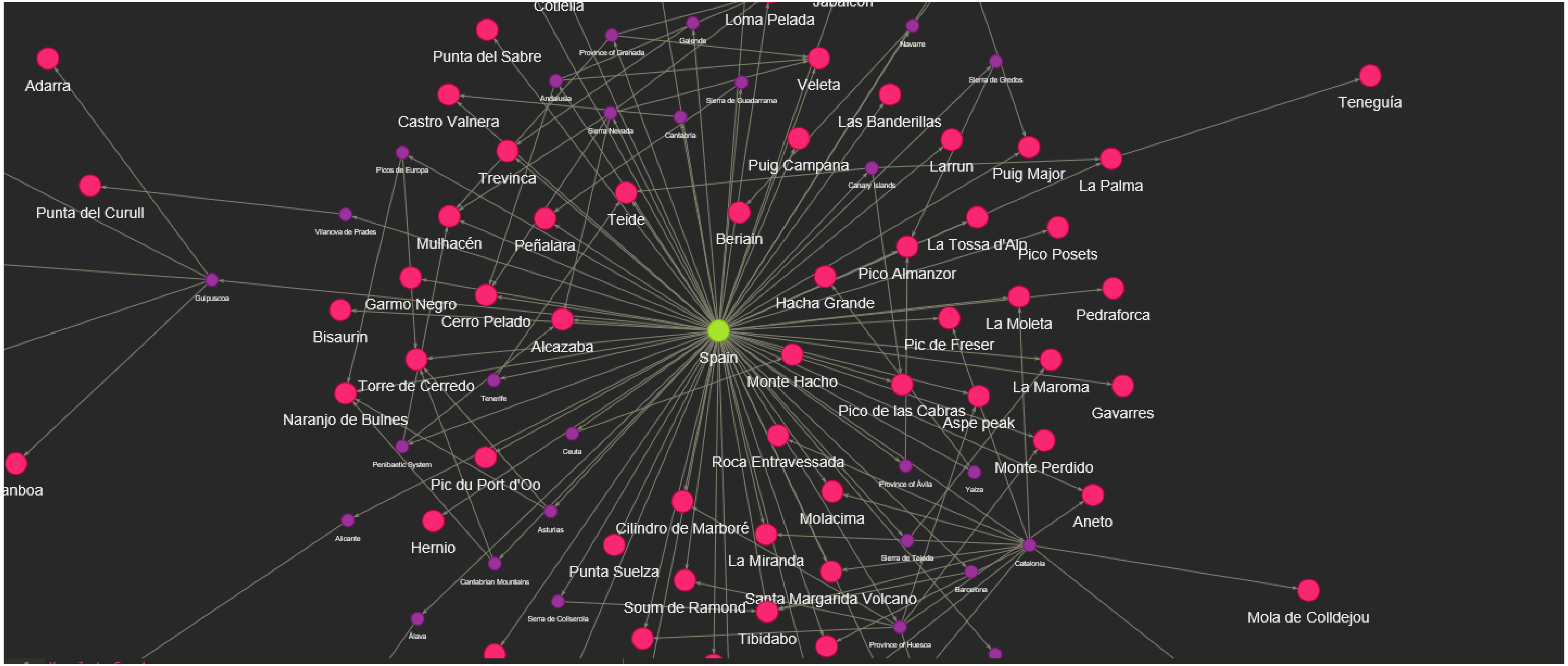


Mountains in Spain

```
KnowledgeGraph
     .StartFrom("Spain")
     .FollowEdge("location_location_contains")
     .VisitNode(_ => Action.Continue & _.return_if(_.type("geography_mountain")))
     .FollowEdge("location_location_contains")
     .VisitNode(_ => _.return_if(_.type("geography_mountain")))
```

A graph query example

Mountains in Spain



Outline

- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- Offline analytics
- Online query processing
- Graph generation
- Case study
- Advance topics

Outline

- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- *Offline analytics*
- Online query processing
- Graph generation
- Case study
- Advance topics

Graph processing scenarios

Online query processing + Offline analytics

A real-life relation search scenario

A News Headline

Tom Cruise Admits **Katie Holmes** Divorced Him To Protect **Suri** From Scientology

1. **Tom Cruise** – people.person.marriage – (marriage) – time.event.person – **Katie Holmes**
2. **Tom Cruise** – people.person.children – (**Suri Cruise**) – people.person.parent – **Katie Holmes**
3. **Tom Cruise** – film.actor.film – (Bambi Verleihung 2007) – film.filmactor – **Katie Holmes**
4. . . .

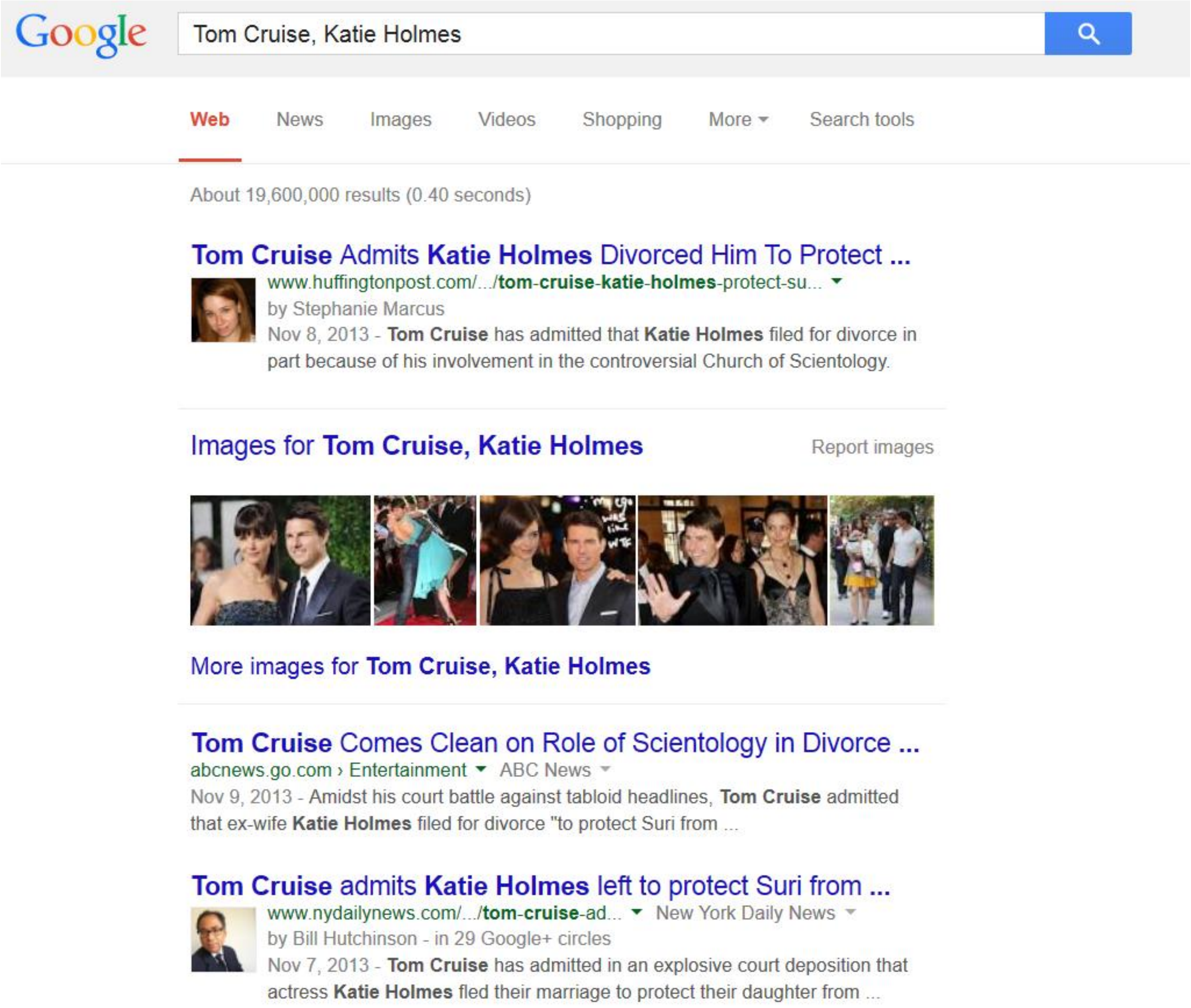
Relation search in knowledge graph

Entity A $\cdots \rightsquigarrow$ Entity B

Multi-hop Relation Search

- Discover the **hidden relations** between entities
- Enable more than what entity indexes can support

Search results of Google



Search results of Bing

bing | Tom Cruise, Katie Holmes

MS Beta | 4,340,000 RESULTS | Any time

News about Tom Cruise, Katie Holmes

bing.com/news

KATIE HOLMES DATING JAMIE FOXX RUMORS CONTINUE AS THE ACTRESS' EX-HUSBAND, **TOM CRUISE** WAS REPORTED TO HAVE FINALLY MOVED ON

Travelers Today · 3 days ago

Katie Holmes dating rumors again sparked as her ex-husband **Tom Cruise** was reportedly dating other woman and that…

Is **Tom Cruise** Dating Laura Prepon - **Katie Holmes** Ex Lands Scientologist Girlfriend?

The National Ledger · 10 days ago

Katie Holmes Celebrates Suri **Cruise**'s 8th Birthday

WebProNews · 3 days ago

Images of Tom Cruise, Katie Holmes

bing.com/images

Katie Holmes Celebrates Suri Cruise's 8th Birthday ...

www.webpronews.com/katie-holmes-celebrates-suri-**cruises**-8th...

Katie Holmes helped daughter Suri **Cruise** celebrate her 8th birthday in style. She treated her daughter, along with a few guests, to dinner at Nobu Next …

Tom Cruise: Katie Holmes Divorce Was A Surprise (UPDATE)

www.huffingtonpost.com/2013/04/09/**tom-cruise-katie-holmes**-divorce...

Apr 09, 2013 · **Tom Cruise** says **Katie Holmes** divorce was a surprise. Here, the former couple is pictured at the "Mission Impossible: Ghost Protocol" premiere in Dec. 2011.

See results for

Katie Holmes

American Actress

Kate Noelle "Katie" Holmes is an American actress and model who first achieved fame for her role as Joey Pot…

Tom Cruise

Film Actor

Tom Cruise, is an American film actor and producer. He has been nominated for three Academy Awards and h…

Related searches

- Tom Cruise Katie Holmes **Married**
- Tom Cruise Katie Holmes **Gossip**
- Tom Cruise Katie Holmes **Photos**
- Tom Cruise Katie Holmes **Baby**
- Tom Cruise Katie Holmes **Unusual Marriage**
- Katie Holmes Tom Cruise **Split**
- Tom Cruise Katie Holmes **Suri Custody Settlement**
- **Leah Remini Problems Started** Tom Cruise **Wedding**

Relation search in knowledge graph

Satori

Add Search

Tom Cruise, Mimi Rogers, Nicole Kidman, Katie Holmes

Results View

94 Results (103 ms)

Results
o--film.actor.film-->(Eyes Wide Shut)--film.film.actor-->(Nicole Kidman)
o--film.actor.film-->(National Movie Awards)--film.film.actor-->(Katie Holmes)
o--film.actor.film-->(InStyle: Celebrity Weddings)--film.film.actor-->(Katie Holmes)
o--people.person.marriage-->(marriage)--time.event.person-->(Katie Holmes)
o--people.person.marriage-->(marriage)--time.event.person-->(Nicole Kidman)
o--film.actor.film-->(War of the Worlds: UK Premiere Special)--film.film.actor-->(Katie Holmes)
o--film.producer.film-->(The Others)--award.nominated_work.nomination-->(nomination)--award.nomination.nominee--(Nicole Kidman)
o--people.person.children-->(Connor Cruise)--people.person.siblings-->(Isabella Jane Cruise)--people.person.parent--(Nicole Kidman)
o--film.producer.film-->(The Others)--award.nominated_work.nomination-->(nomination)--award.nomination.nominee--(Nicole Kidman)
o--film.actor.performance-->(performance)--film.performance.film-->(Eyes Wide Shut)--film.film.actor--(Nicole Kidman)

Prev Page Next Page

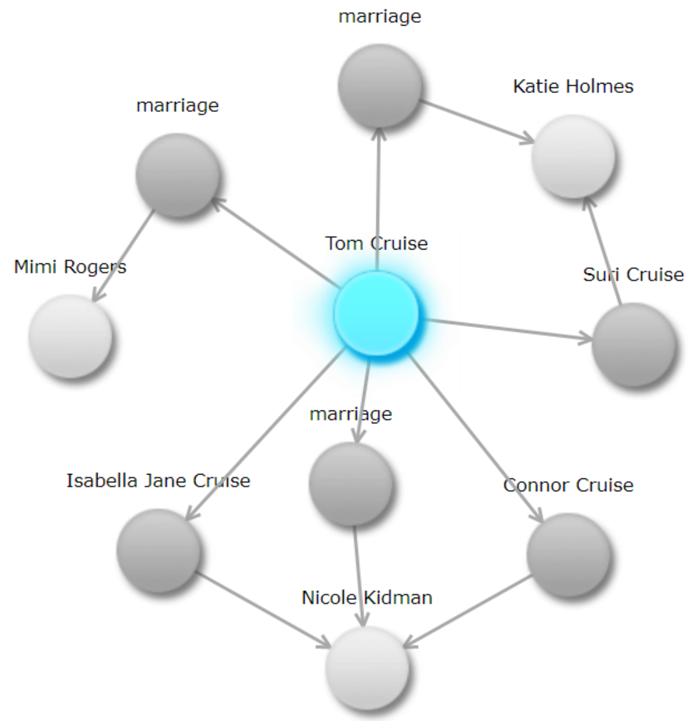
Relation search in knowledge graph

Satori

Add Search

Tom Cruise, Mimi Rogers, Nicole Kidman, Katie Holmes

Results View



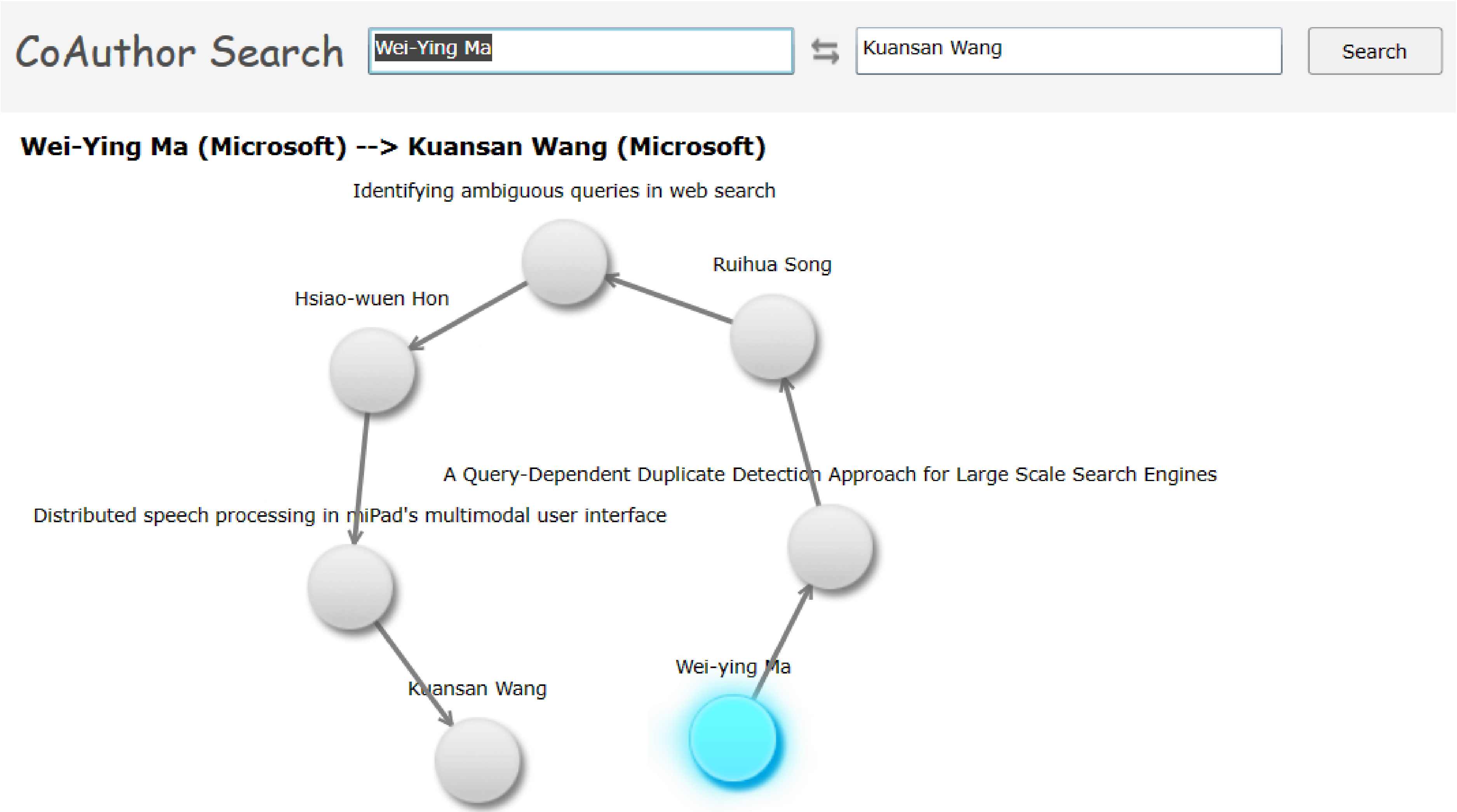
Tom Cruise

Tom Cruise (born Thomas Cruise Mapother IV; July 3, 1962), is an American film actor and producer. He has been nominated for three Academy Awards and has won three Golden Globe Awards. He started his career at age 19 in the 1981 film Endless Love. After portraying supporting roles in Taps (1981) and The Outsiders (1983), his first leading role was in Risky Business, released in August 1983. Cruise became a full-fledged movie...

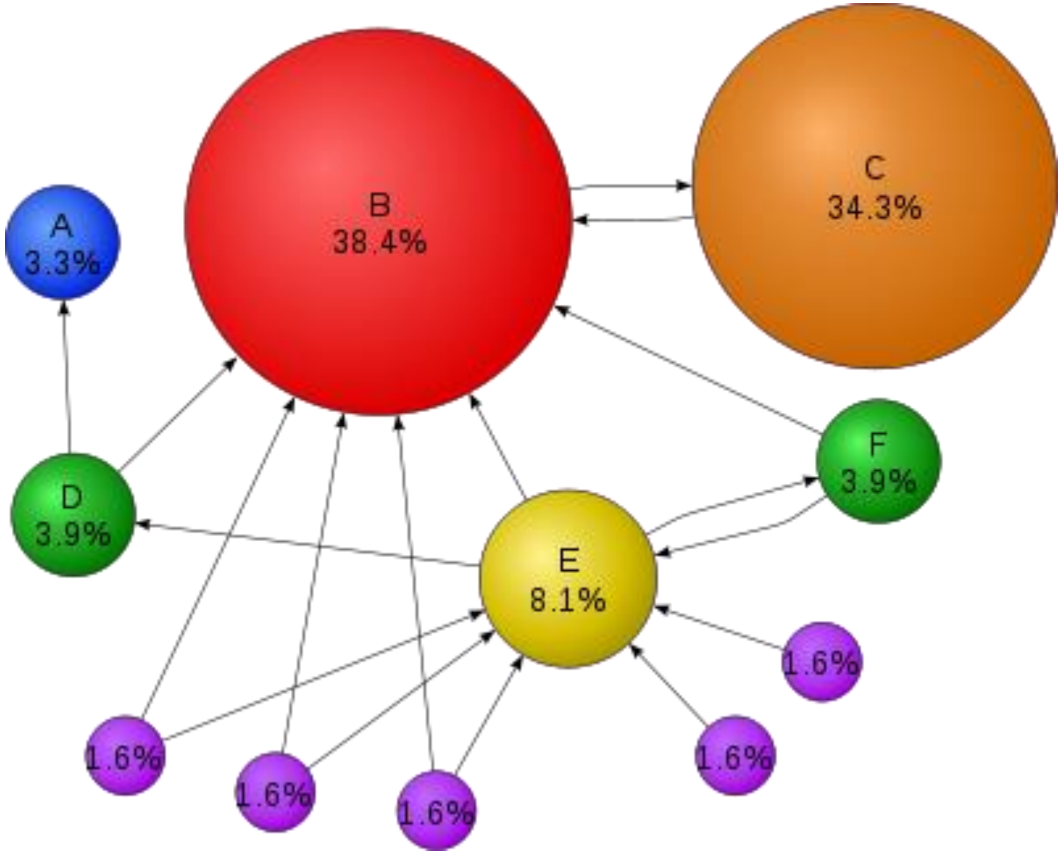
Types

award.nominee, award.winner, film.actor, film.director, film.producer, film.story_contributor ...

Academic graph



Offline analytics example: PageRank

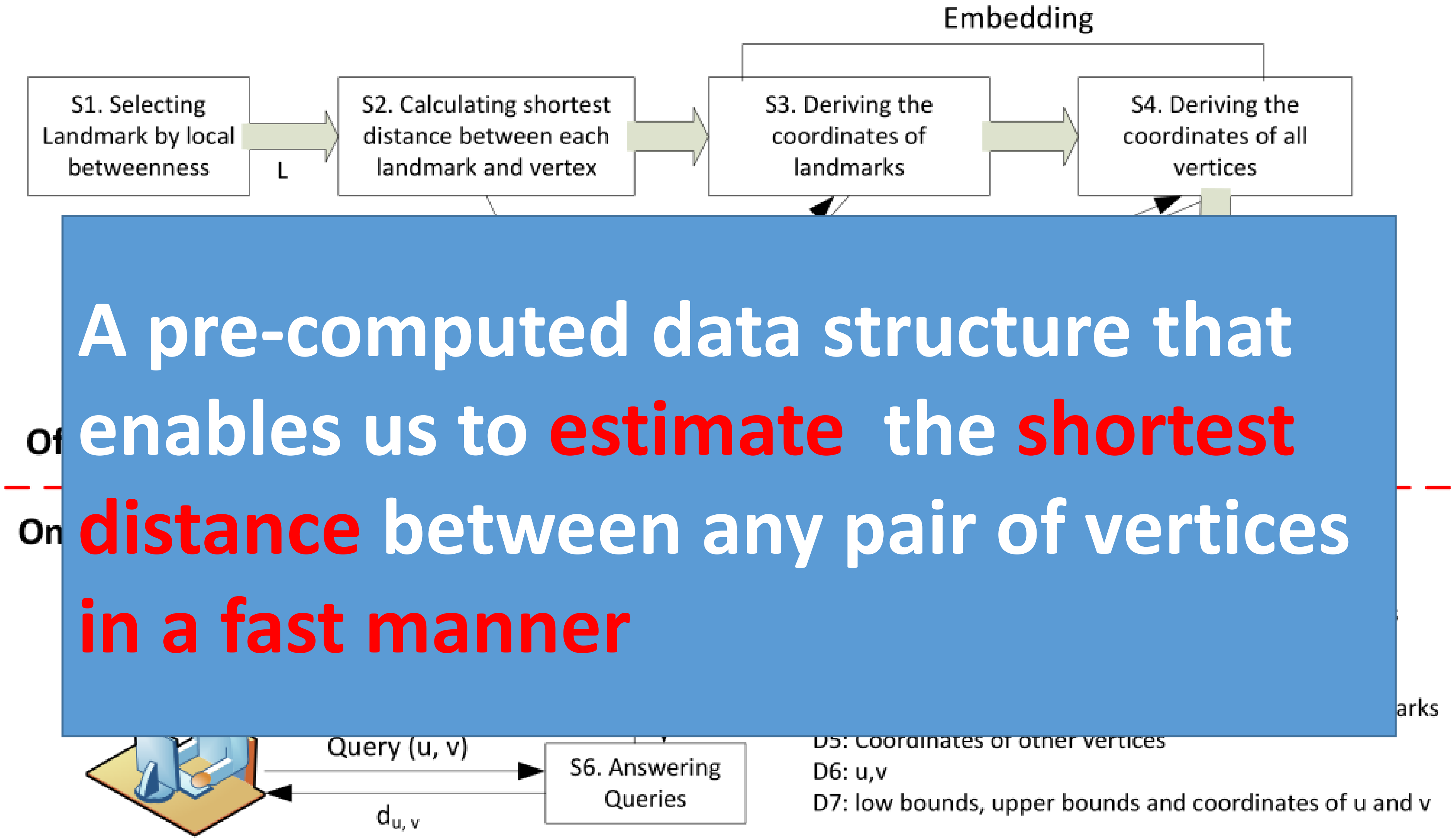


An important algorithm behind Google, Bing, …

https://en.wikipedia.org/wiki/PageRank

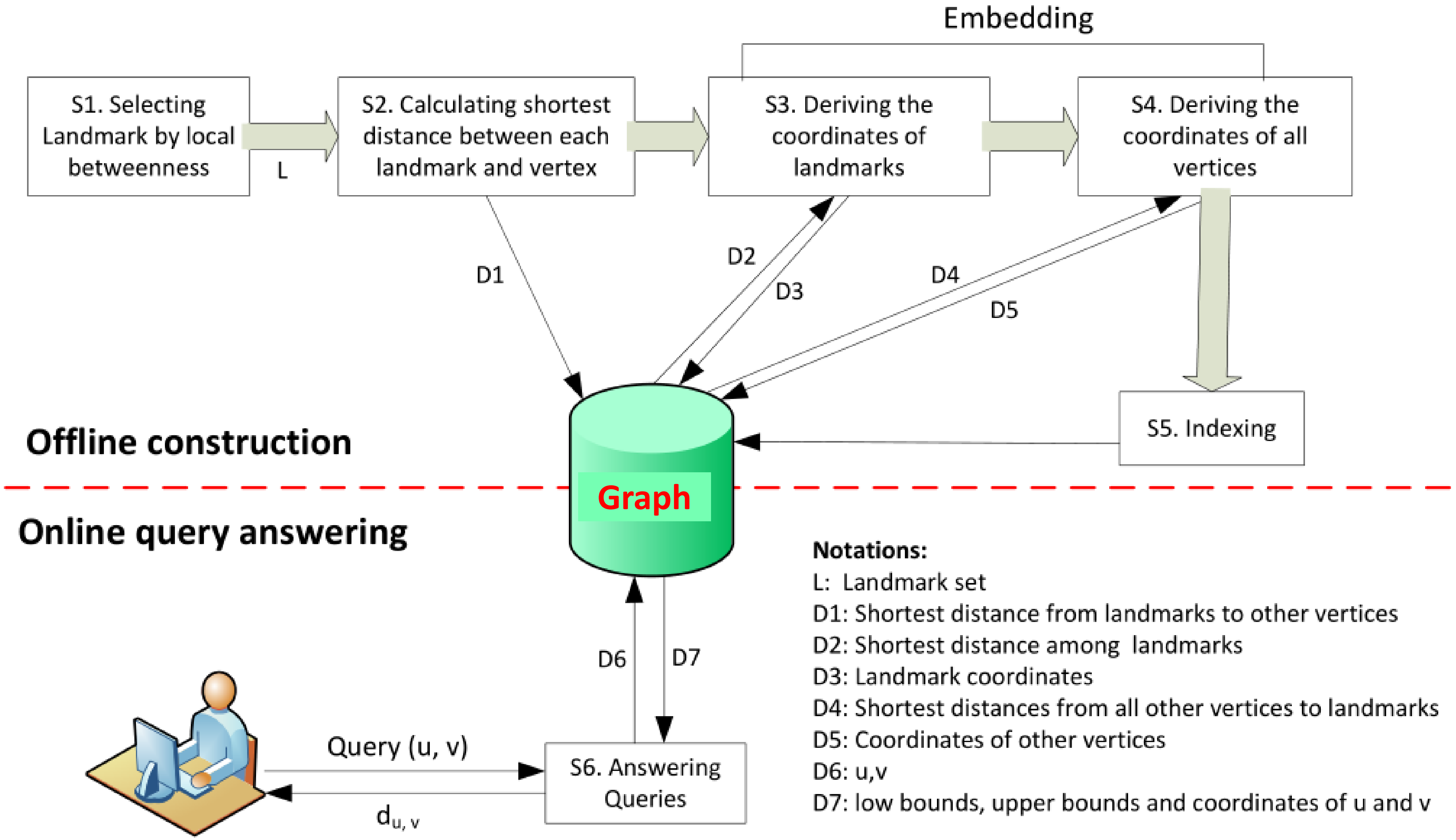
Query processing + offline analytics

Architecture of distance oracle [Qi et al. vldb 2014]



A pre-computed data structure that enables us to **estimate** the **shortest distance** between any pair of vertices **in a fast manner**

Architecture of distance oracle [Qi et al. vldb 2014]



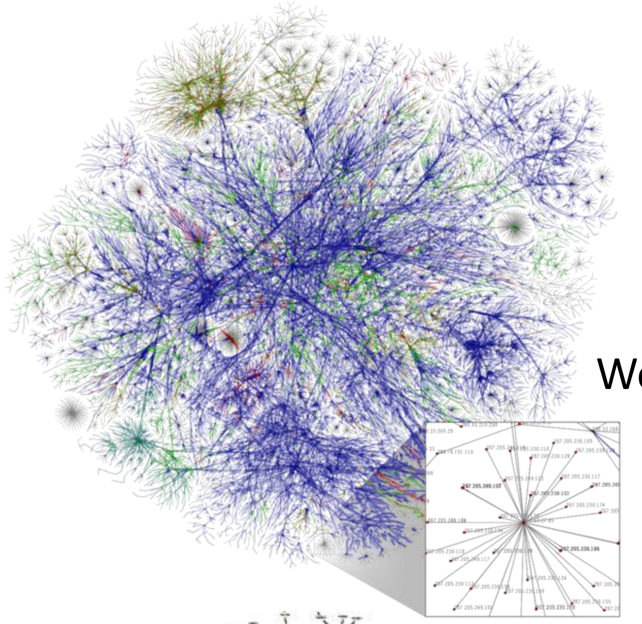
Notations:
L: Landmark set
D1: Shortest distance from landmarks to other vertices
D2: Shortest distance among landmarks
D3: Landmark coordinates
D4: Shortest distances from all other vertices to landmarks
D5: Coordinates of other vertices
D6: u,v
D7: low bounds, upper bounds and coordinates of u and v

Outline

- Graph processing scenarios
- **Challenges of large graph processing**
- General design principles
- *Offline analytics*
- Online query processing
- Graph generation
- Case study
- Advance topics

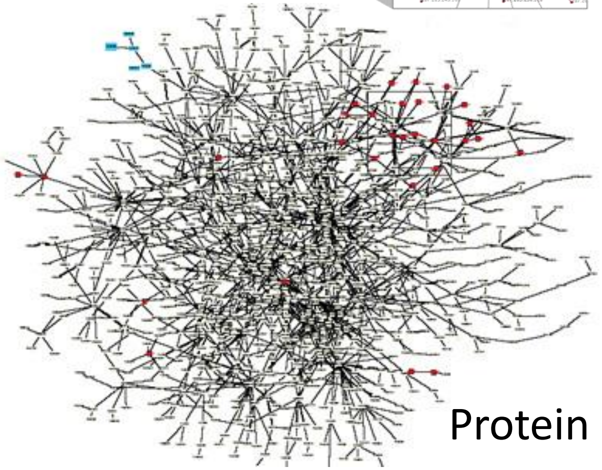
Challenge I: diversity of graphs

Schema Graph



Web Graph

Social Network



Protein Interaction Network

Do we need to design algorithms for each type of graphs?

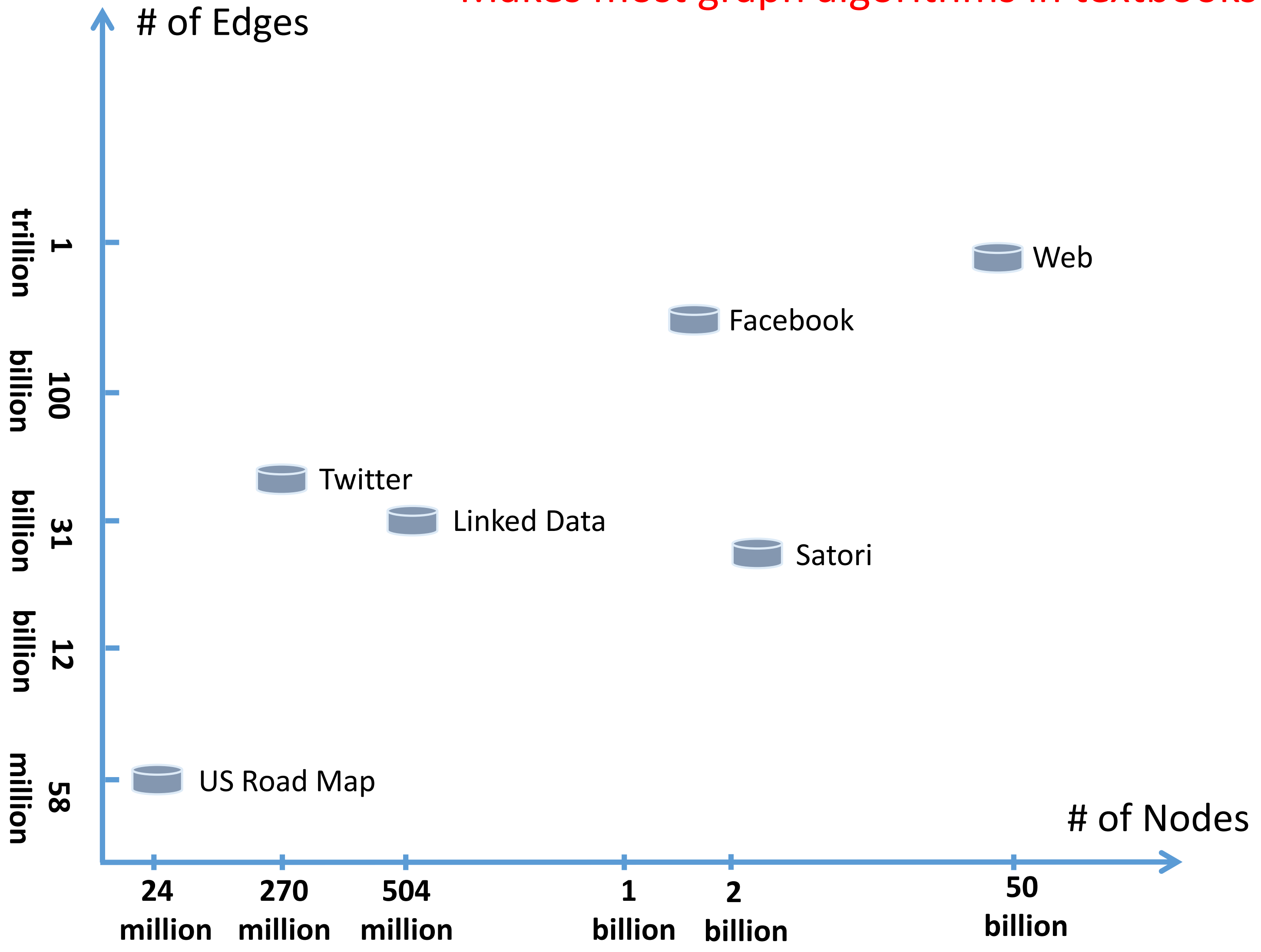
Challenge II: diversity of computations

- Online query processing
  - Shortest path query
  - Subgraph matching query
  - SPARQL query
  - …
- Offline graph analytics
  - PageRank
  - Community detection
  - …
- Other graph operations
  - Graph generation, visualization, interactive exploration, etc.

Do we need to implement systems for each graph operation?

Challenge III: the scale of graphs

Makes most graph algorithms in textbooks ineffective!



Existing systems

- Mature data processing systems
  - RDBMS
  - Map Reduce Systems
- Systems specialized for certain graph operations:
  - PageRank, FlockDB
- General-purpose graph processing systems
  - Neo4j, Trinity, Horton, HyperGraphDB, TinkerGraph, InfiniteGraph, Cayley, Titan, PEGASUS, Pregel, Giraph, GraphLab, GraphChi, GraphX …

Representative graph processing systems

	Property graphs	Online query	Data sharding	In-memory storage	Atomicity & Transaction
★ Neo4j	Yes	Yes	No	No	Yes
★ Trinity	Yes	Yes	Yes	Yes	Atomicity
★ Horton	Yes	Yes	Yes	Yes	No
★ HyperGraphDB	No	Yes	No	No	Yes
★ FlockDB	No	Yes	Yes	No	Yes
★ TinkerGraph	Yes	Yes	No	Yes	No
★ InfiniteGraph	Yes	Yes	Yes	No	Yes
★ Cayley	Yes	Yes	SB	SB	Yes
★ Titan	Yes	Yes	SB	SB	Yes
★ MapReduce	No	No	Yes	No	No
★ PEGASUS	No	No	Yes	No	No
★ Pregel	No	No	Yes	No	No
★ Giraph	No	No	Yes	No	No
★ GraphLab	No	No	Yes	No	No
★ GraphChi	No	No	No	No	No
★ GraphX	No	No	Yes	No	No

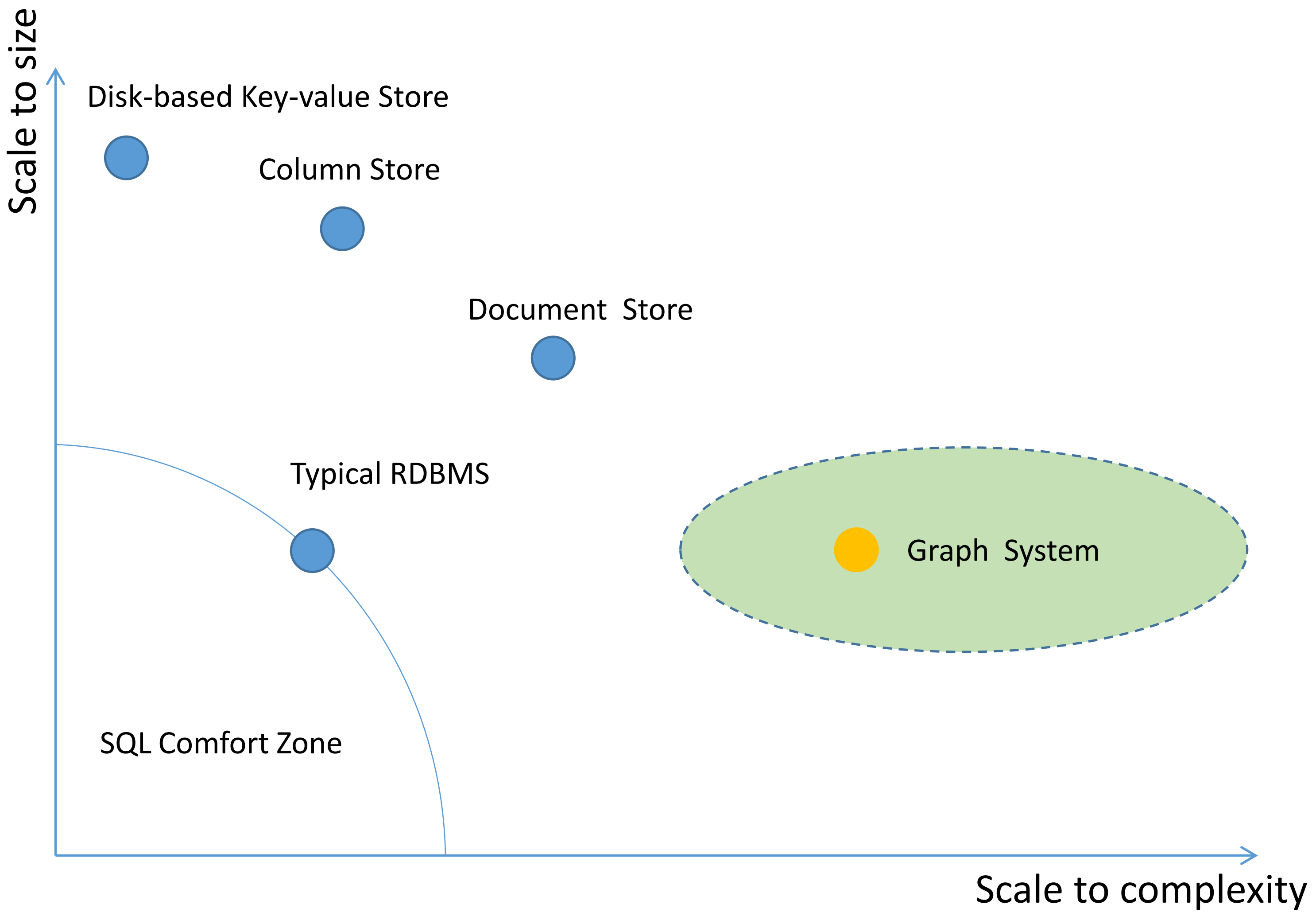
Outline

- Graph processing scenarios
- Challenges of large graph processing
- **General design principles**
- *Offline analytics*
- Online query processing
- Graph generation
- Case study
- Advance topics

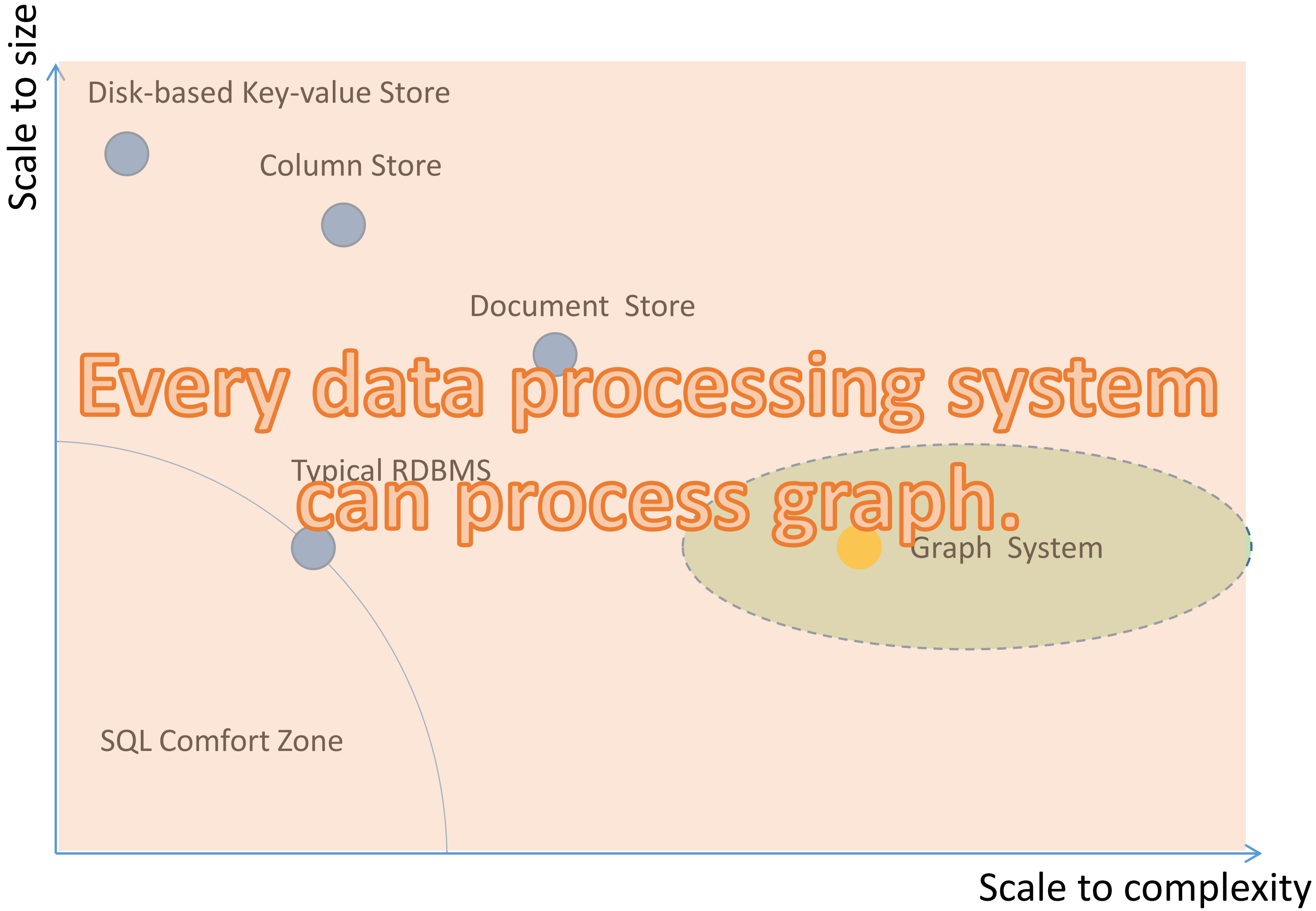
The needs ultimately determine the design

- The first important rule: there is no one-size-fits-all system

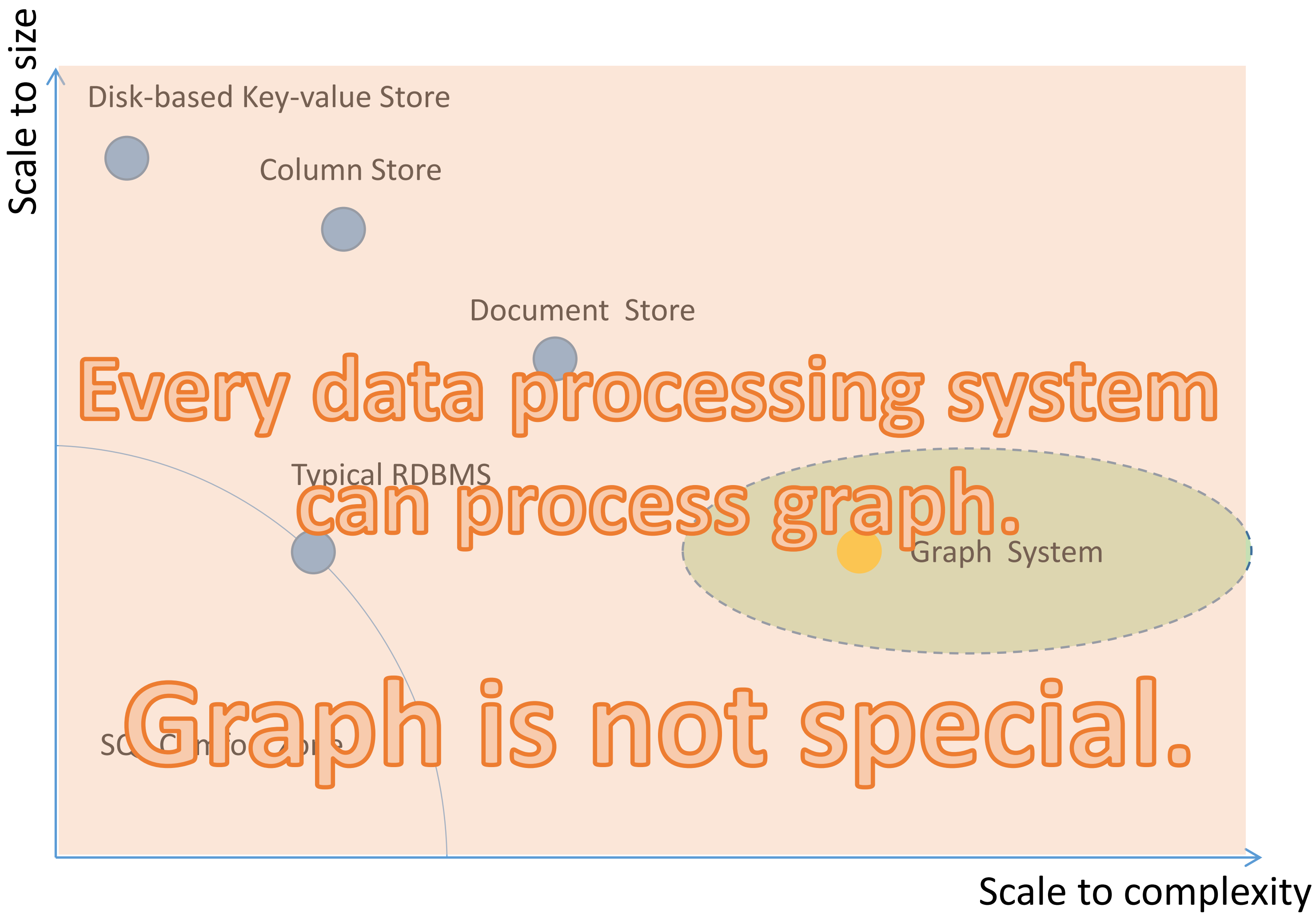
First rule: no one-size-fits-all system



First rule: no one-size-fits-all system

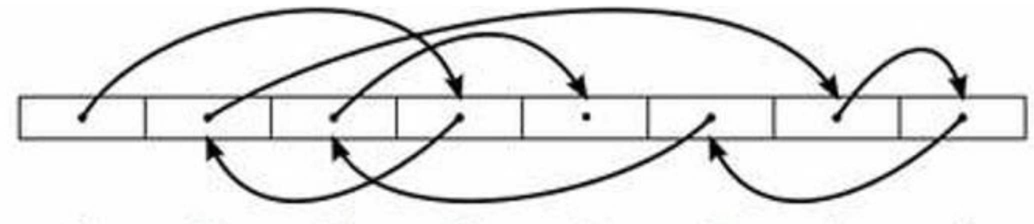


First rule: no one-size-fits-all system



Characteristics of parallel graph processing

- Random access (Poor Locality)
  - For a node, its adjacent nodes cannot be accessed without “jumping” no matter how you represent a graph
  - Not cache-friendly, data reuse is hard

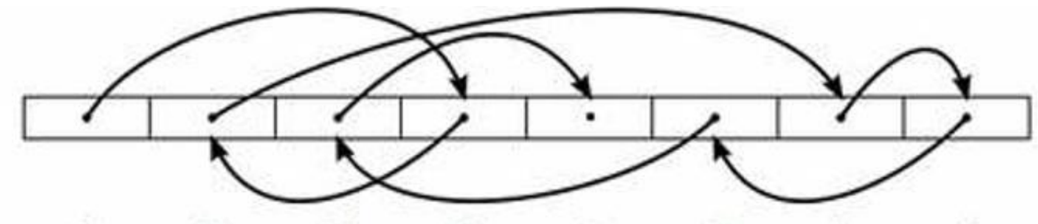


- Data is hard to partition
  - Difficult to extract parallelism by partitioning data
  - Hard to get an efficient “Divide and Conquer” solution
- Data driven
  - the structure of computations is not known a priori
- High data access to computation ratio

Reference: Challenges in parallel graph processing

Characteristics of parallel graph processing

- Random access (Poor Locality)
  - For a node, its adjacent nodes cannot be accessed without "jumping" no matter how you represent a graph
  - Not cache-friendly, data reuse is hard



- Data is hard to partition

In this sense, graph is "special".

- Data driven
  - the structure of computations is not known a priori
- High data access to computation ratio

Reference: Challenges in parallel graph processing

Design choices

- **First important rule: there is no one-size-fits-all system**
- Does this system support online queries, offline analytics, or both?
- Is the system optimized for response time, throughput, or both?
- Does the system scale, “out” or “up”?
- Does the system need transaction support?

Online queries vs. offline analytics

- Online query processing is usually optimized for response time
- Offline analytics is usually optimized for throughput
- Compared with offline analytics, online queries are usually harder to optimize
  - Online queries are sensitive to latency
  - The data access patterns of a graph query are difficult to predict

Online queries vs. offline analytics

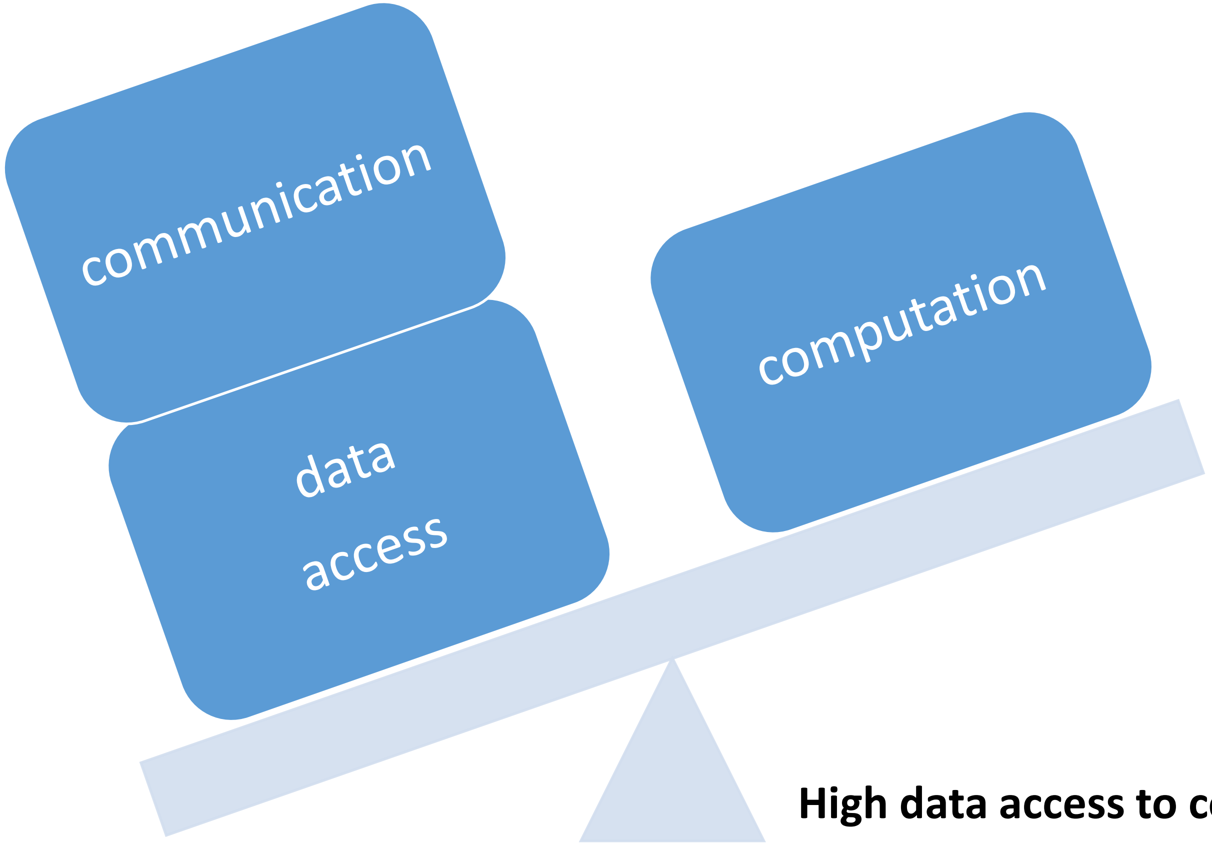
Online Query Processing

- ★ Random access
- ★ Data is hard to partition
- ★ Data driven
- ★ High data access to computation ratio

Offline Analytics

- ★ Data is hard to partition
- ★ High data access to computation ratio

Query response time:
data access + communication + computation



High data access to computation ratio

System design choice

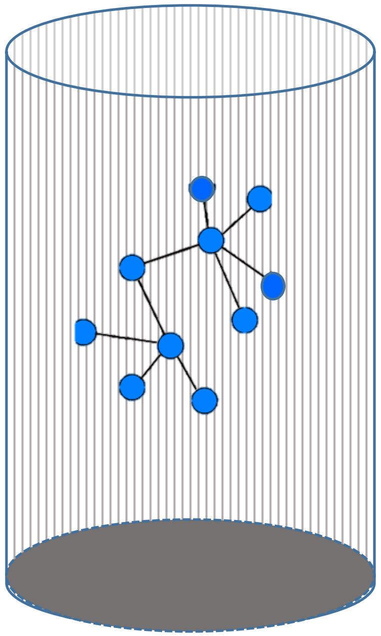
- Main storage (storage backend)
- Index
- Communication paradigm: two-sided vs. one-sided
- Scale out or scale up
- ACID Transactions or not

System design choice

- **Main storage (storage backend)**
- Index
- Communication paradigm: two-sided vs. one-sided
- Scale out or scale up
- ACID Transactions or not

Graph may be in the jail of storage

- We can use may existing data management systems to process graph
- Many existing systems are mature, but not for graph
  - RDBMS, MapReduce
  - The commonest graph operation “traversal” incurs excessive amount of joins



Graph in the Jail of the storage

Traverse graph using joins in RDBMS

ID	name	....
1	N1	…
2	N2	…
3	N3	…
4	N4	…
5	N5	…
6	N6	…
…	…	…

Node Table: N

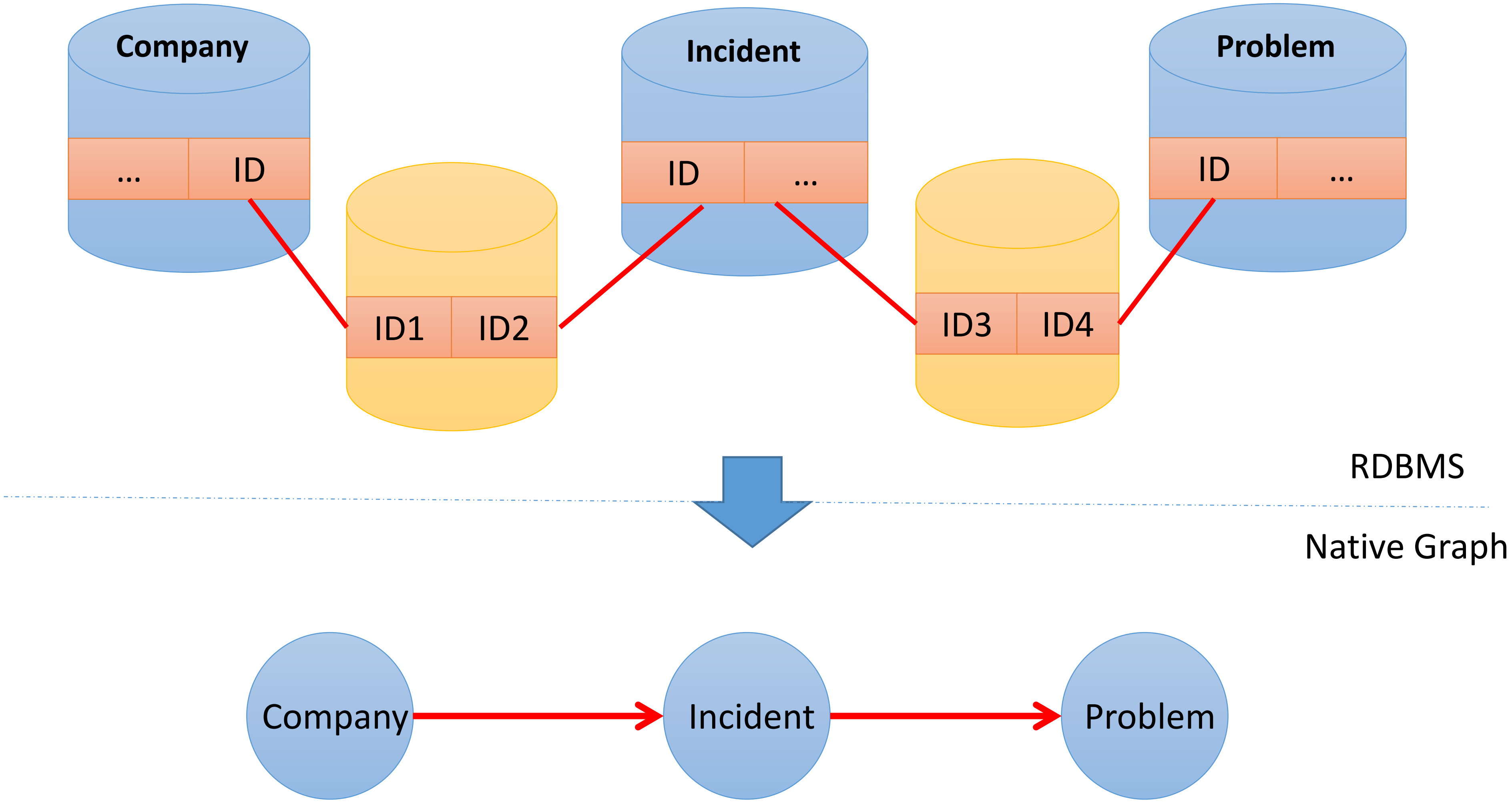
src	dst
1	3
2	4
2	1
4	3
1	5
1	6
…	…

Edge Table: E

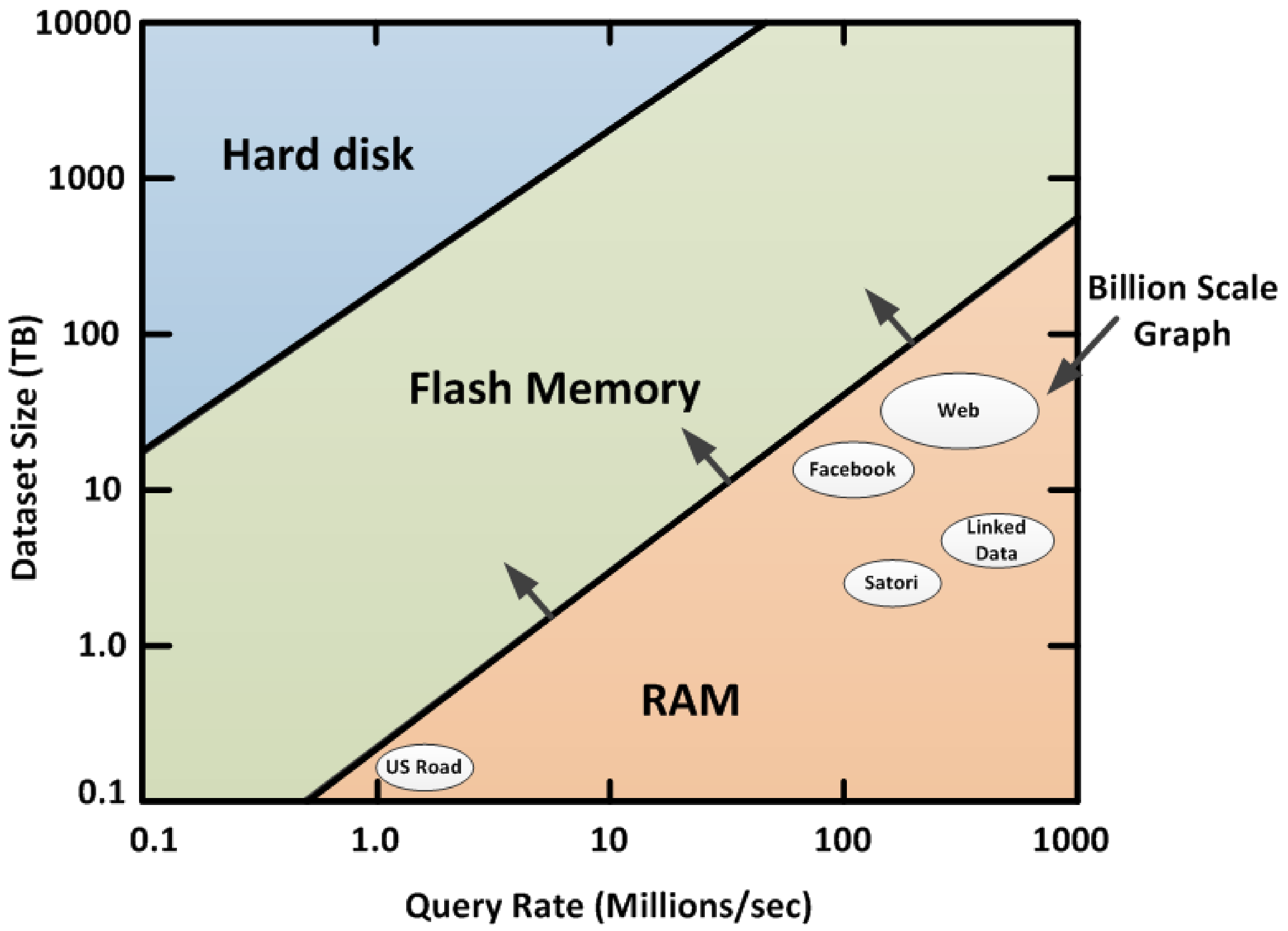
Get neighbors of N1

```
SELECT*
FROM N
LEFT JOIN  E ON N.ID = E.dst
WHERE E.src = 1;
```

Multi-way Join vs. graph traversal



Total cost of ownership



Reproduced from Anderson's SOSP 2009 paper

Trend in cost of RAM

	Today	In 5-10 years
# servers	1000	1000
GB/server	64GB	1024GB
Total capacity	64TB	1PB
Total server cost	\$4M	\$4M
\$/GB	\$60	\$4

Adapted from: John Ousterhout, RAMCloud, 2010

System design choice

- Main storage (storage backend)
- **Index**
- Communication paradigm: two-sided vs. one-sided
- Scale out or scale up
- ACID Transactions or not

Index

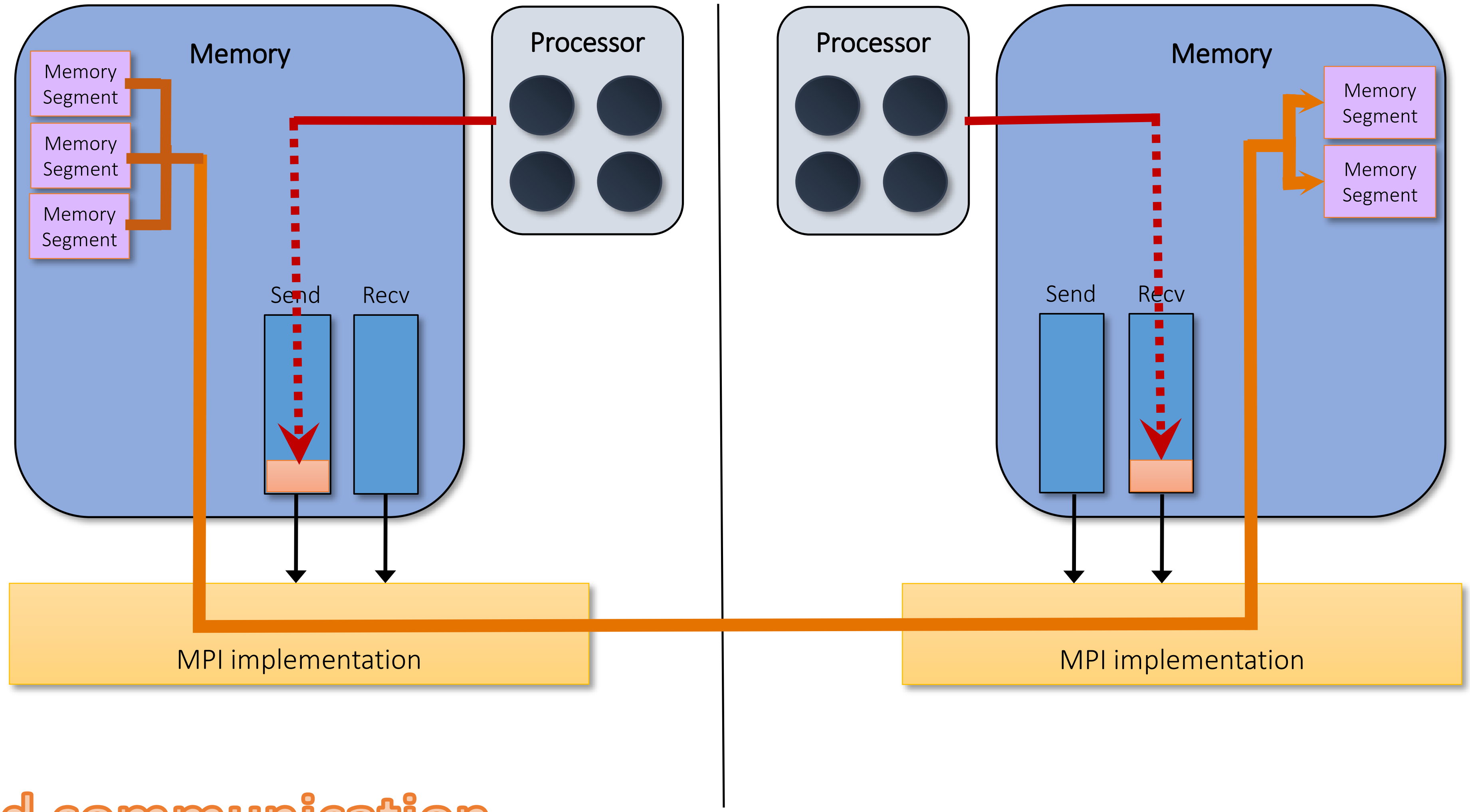
- It is costly to index graph structures, use it wisely.

- We will get back to this later …

System design choice

- Main storage (storage backend)
- Index
- **Communication paradigm: two-sided vs. one-sided**
- Scale out or scale up
- ACID Transactions or not

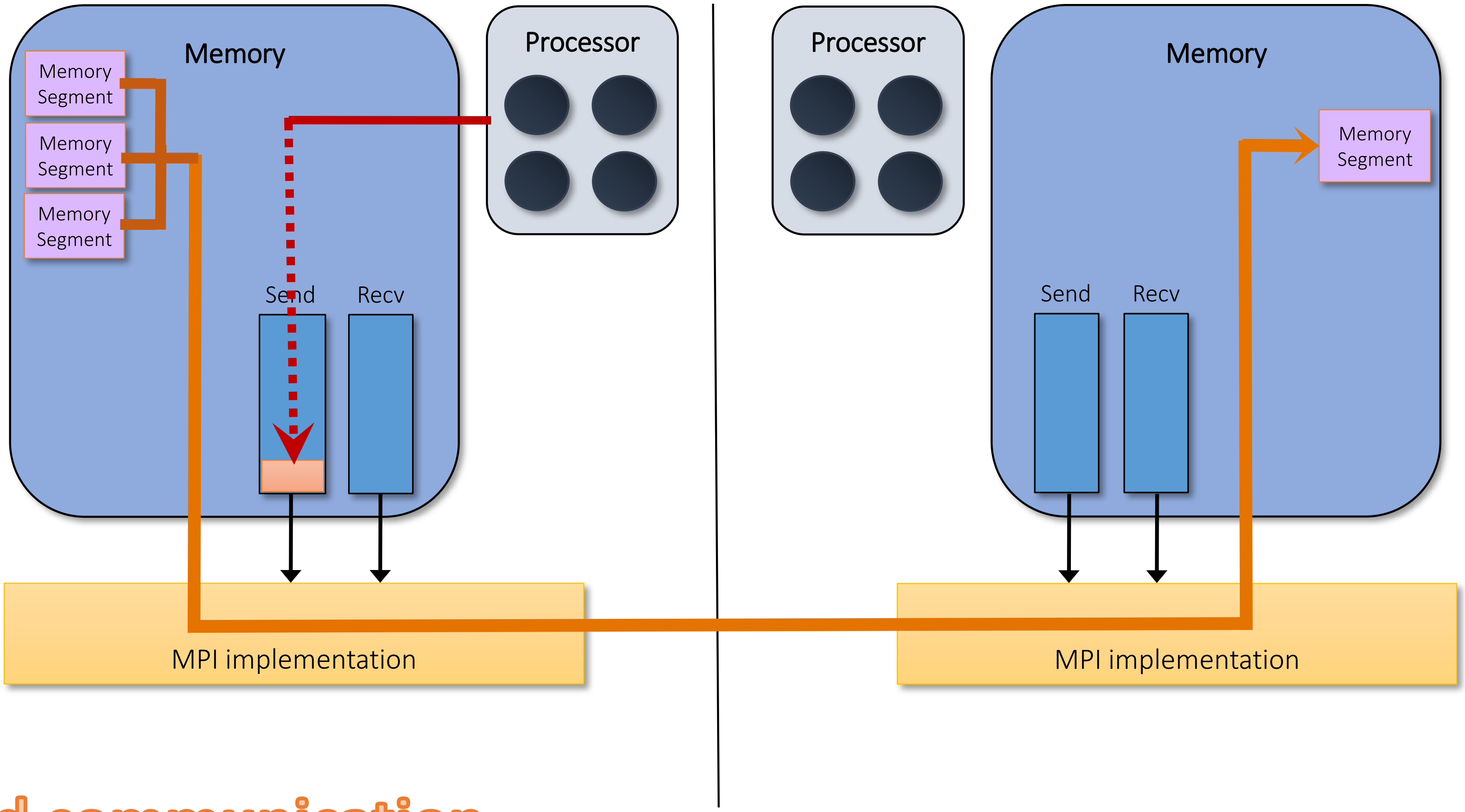
Communication paradigm: two-sided vs. one-sided



Two-sided communication

Adapted from: Advanced parallel programming with MPI (Balaji et al)

Communication paradigm: two-sided vs. one-sided



One-sided communication

Adapted from: Advanced parallel programming with MPI (Balaji et al)

System design choice

- Main storage (storage backend)
- Index
- Communication paradigm: two-sided vs. one-sided
- **Scale out or scale up**
- ACID Transactions or not

Design choice: scale-up vs. scale-out

- Supercomputer model
  - Programming model simple and efficient
    - shared memory address space
  - Expensive and not common
  - Hardware is your ultimate limit
- Distributed cluster model
  - Programming model is complex
    - Message passing and synchronization is more complex
  - Relatively cheaper and can make use of commodity pc
  - More flexible to meet various needs

Scale “OUT”, not “UP”

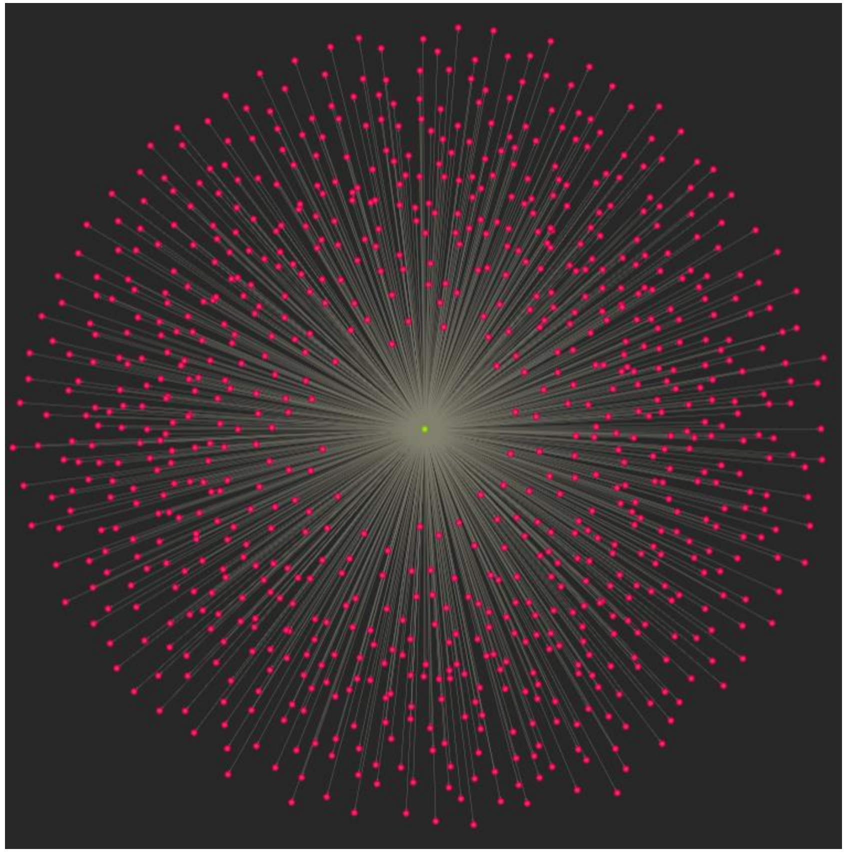
System design choice

- Main storage (storage backend)
- Index
- Communication paradigm: two-sided vs. one-sided
- Scale out or scale up
- **ACID Transactions or not**

Think twice before diving into transactions

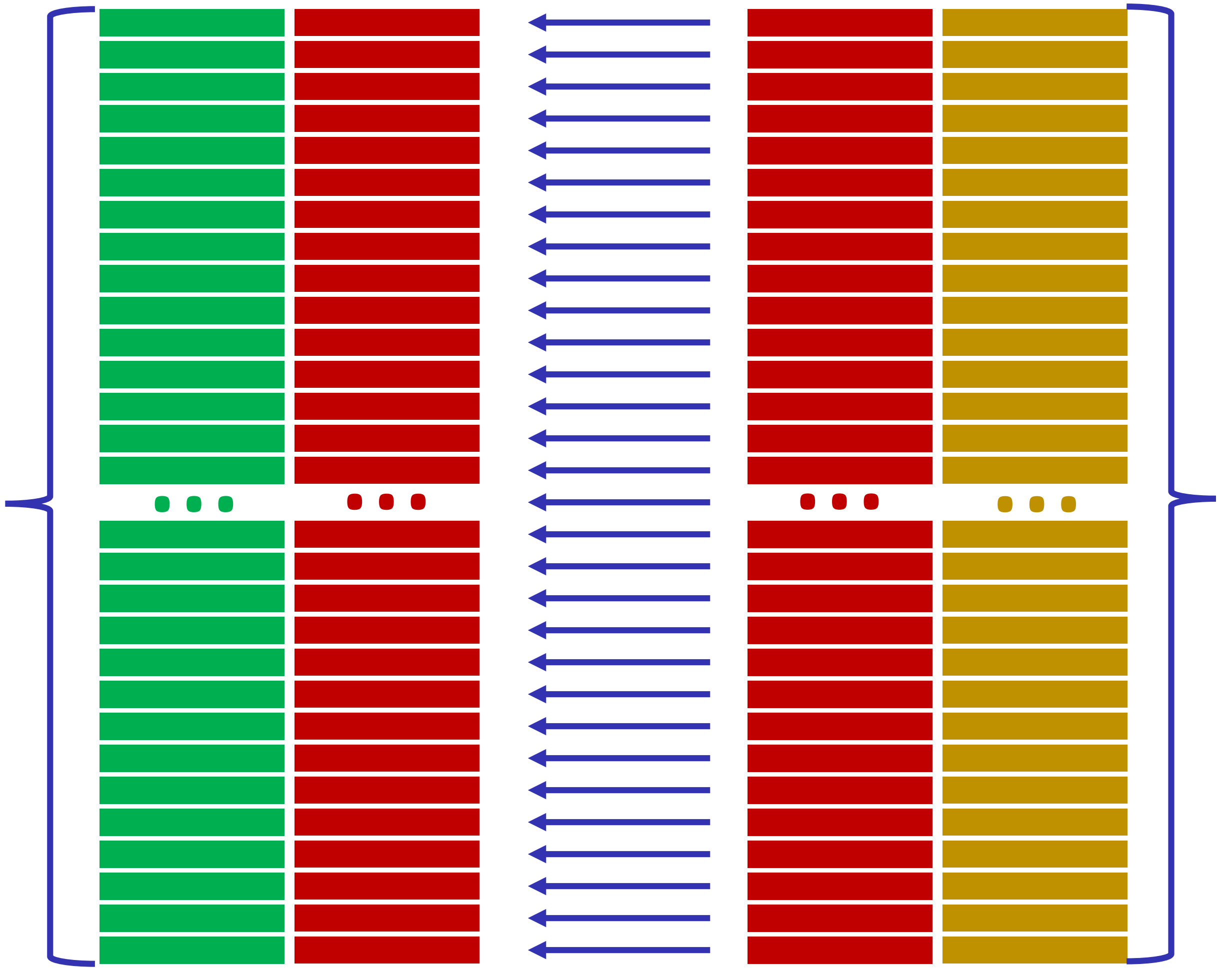
- Pros
  - Strong data consistency guarantee
- Cons
  - The hell of referential integrity
  - The disaster of cascading rollback
  - Multi-round network communications per commit for distributed transactions

The hell of referential integrity

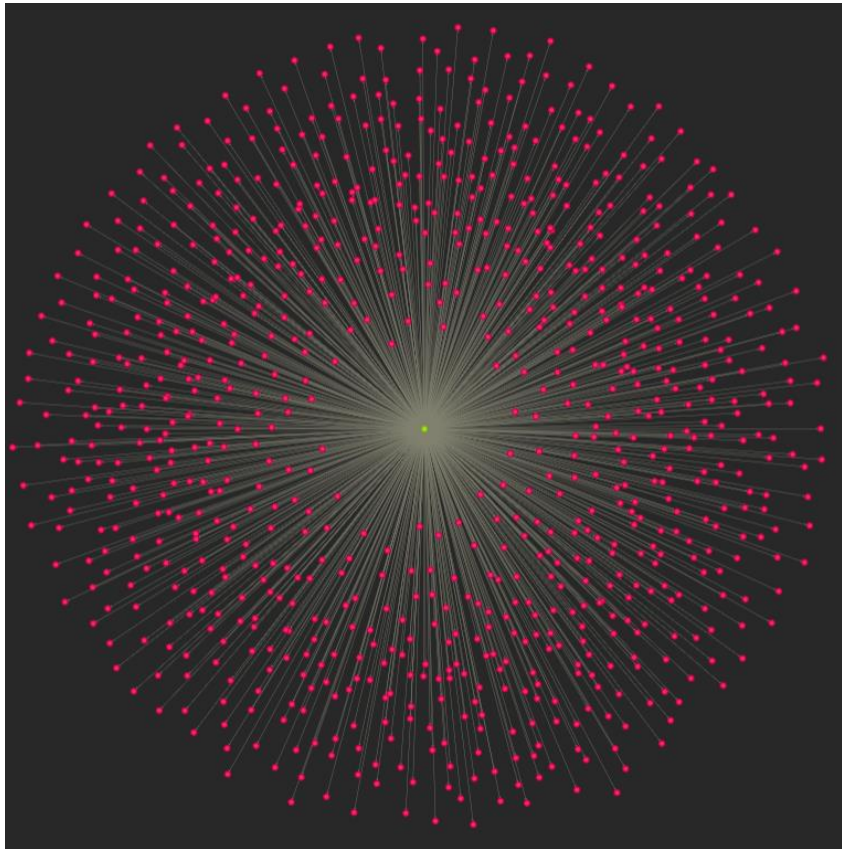


Lady Gaga in Freebase

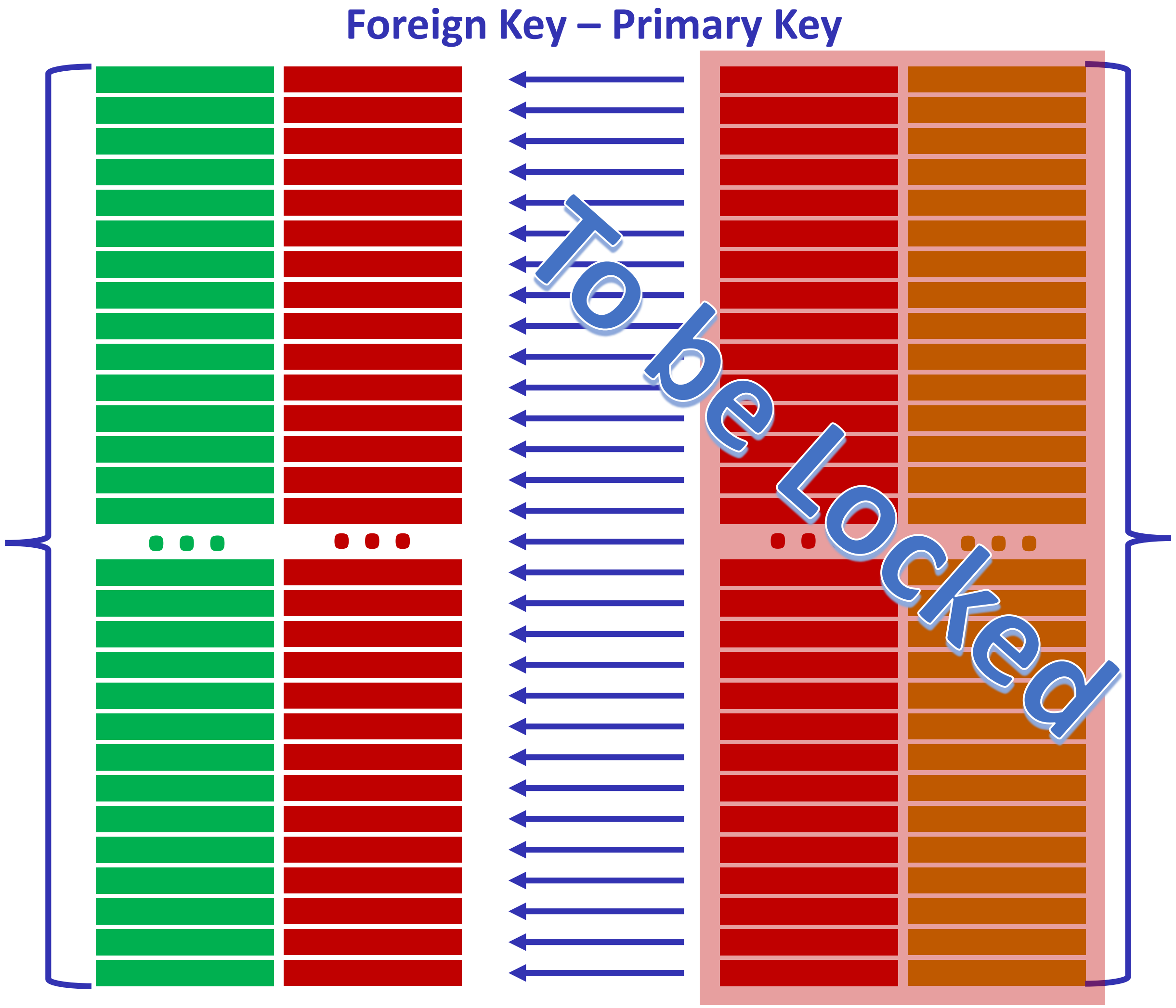
Lady Gaga



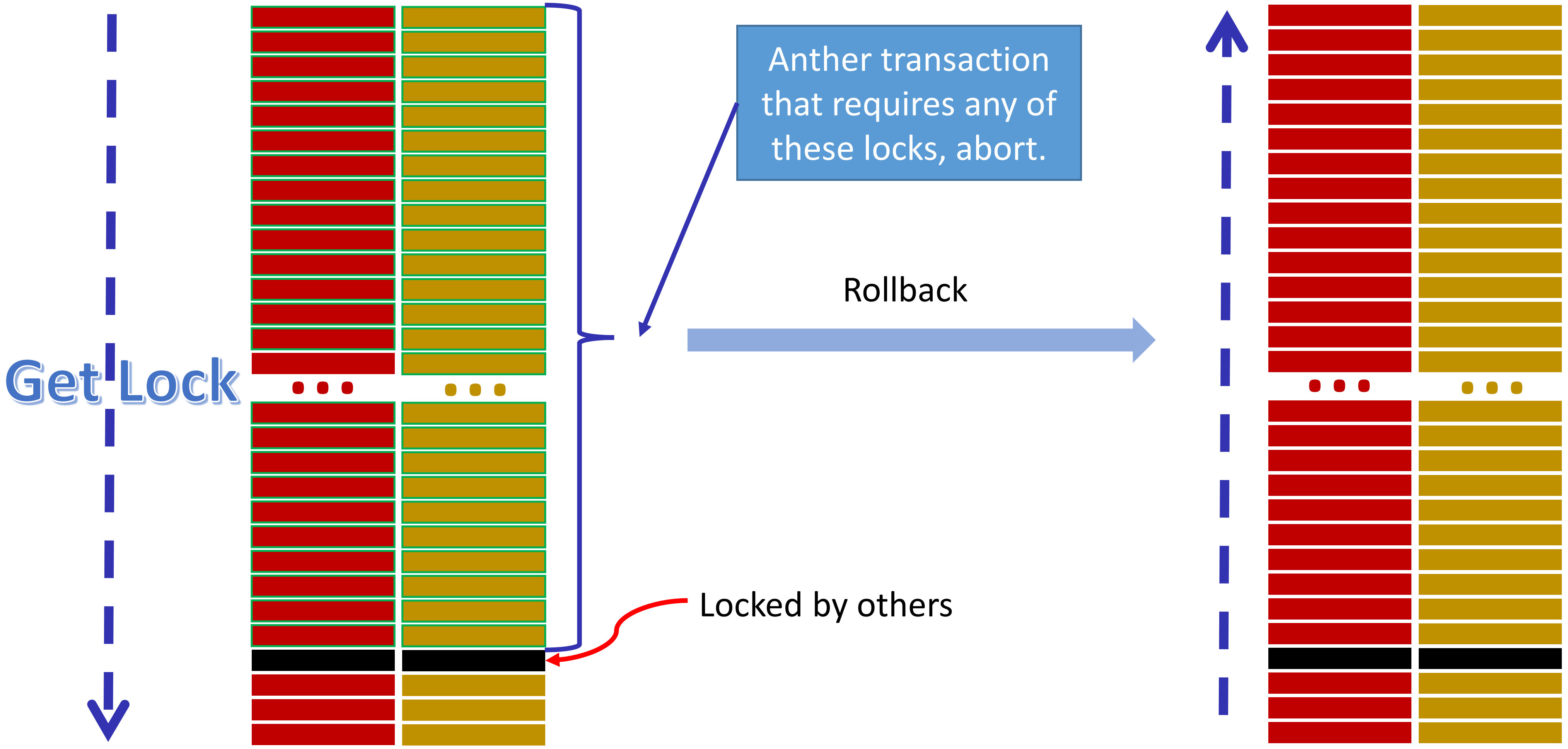
The hell of referential integrity



Lady Gaga



The disaster of cascading rollback



Outline

- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- ***Offline analytics***
- Online query processing
- Graph generation
- Case study
- Advance topics

MapReduce

MapReduce

- High latency, yet high throughput general purpose data processing platform
- Optimized for offline analytics on large data partitioned on hundreds of machines

Processing graph using MapReduce

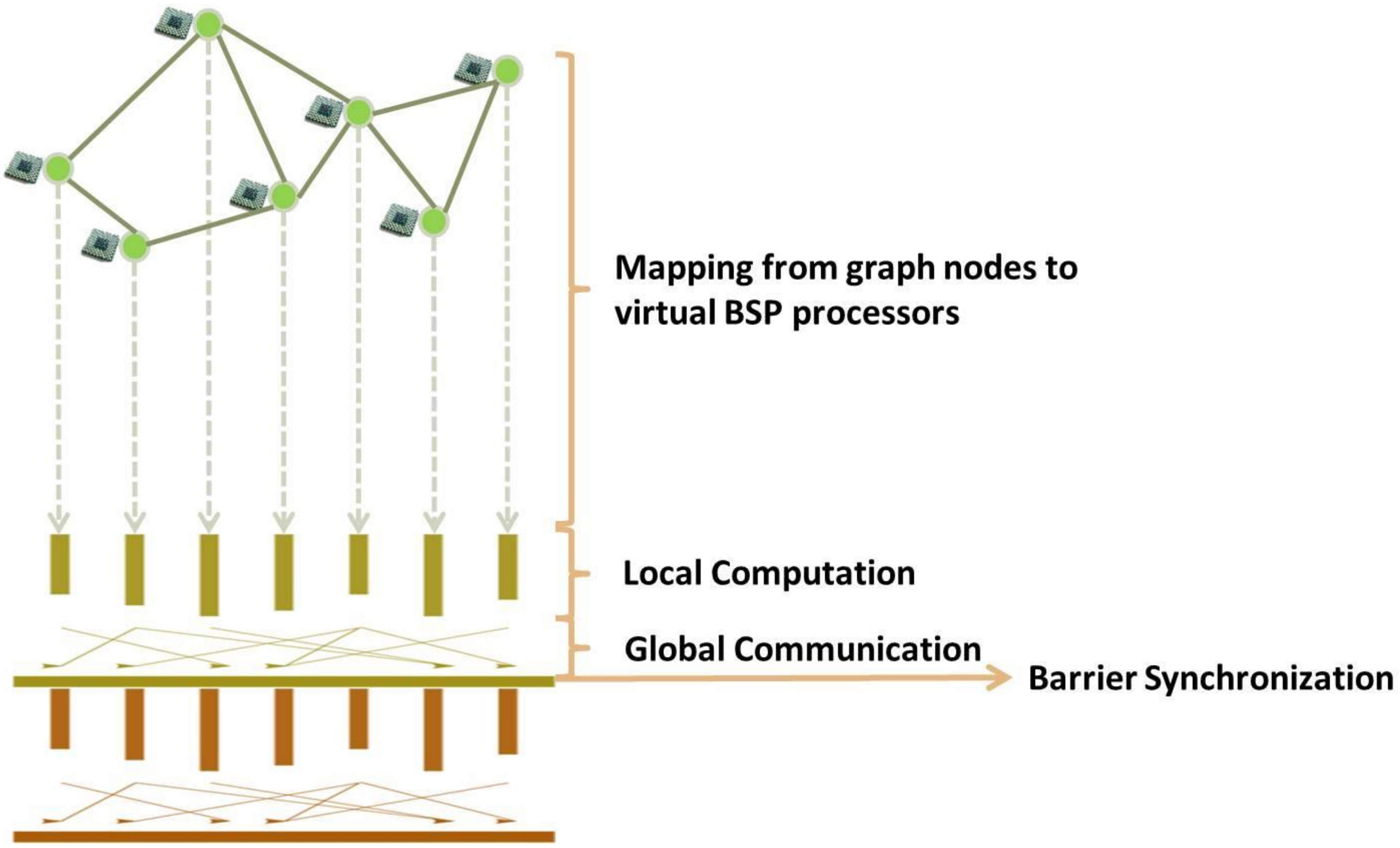
- No online query support
- The data model of map reduce cannot describe graph natively
  - Graph algorithms cannot be expressed intuitively
- Inefficiency for graph processing
  - Intermediate results of each iteration need to be materialized
  - Entire graph structure need to be sent over network iteration after iteration, this incurs huge unnecessary data movement

MapReduce

- De facto of distributed large data processing
- Great scalability: supports extremely large data, but unfortunately not for graph

Vertex-centric graph computation

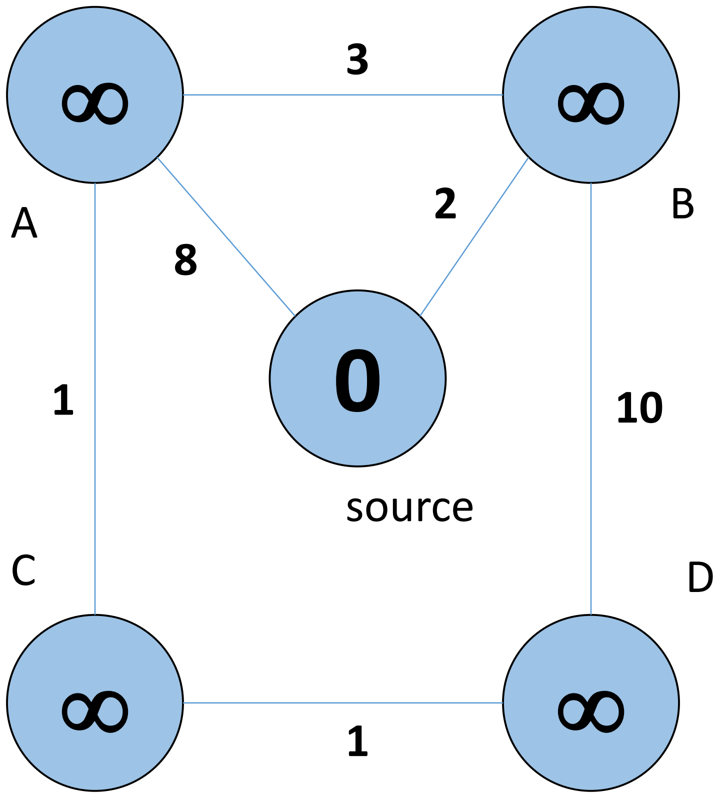
Basic idea: think like a vertex!



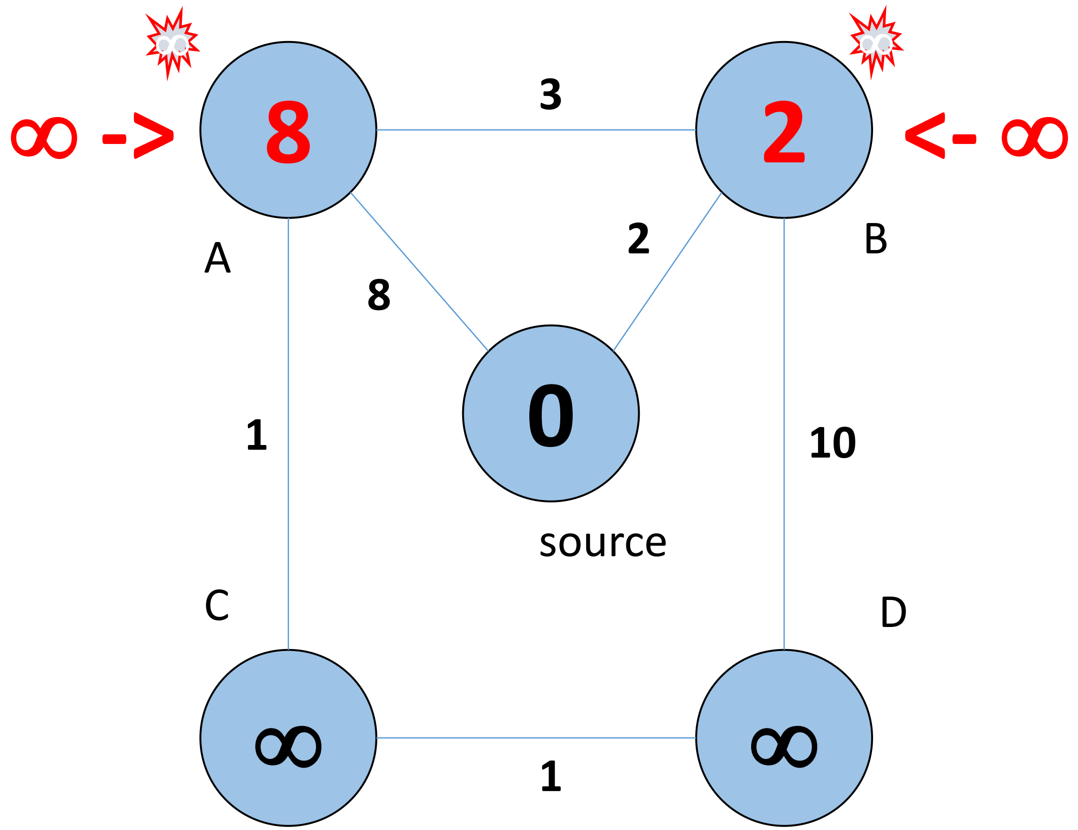
Computation model

- Graph computation is modeled as many supersteps
- Each vertex reads messages sent in previous superstep
- Each vertex performs computation in parallel
- Each vertex can send messages to other vertices in the end of an iteration

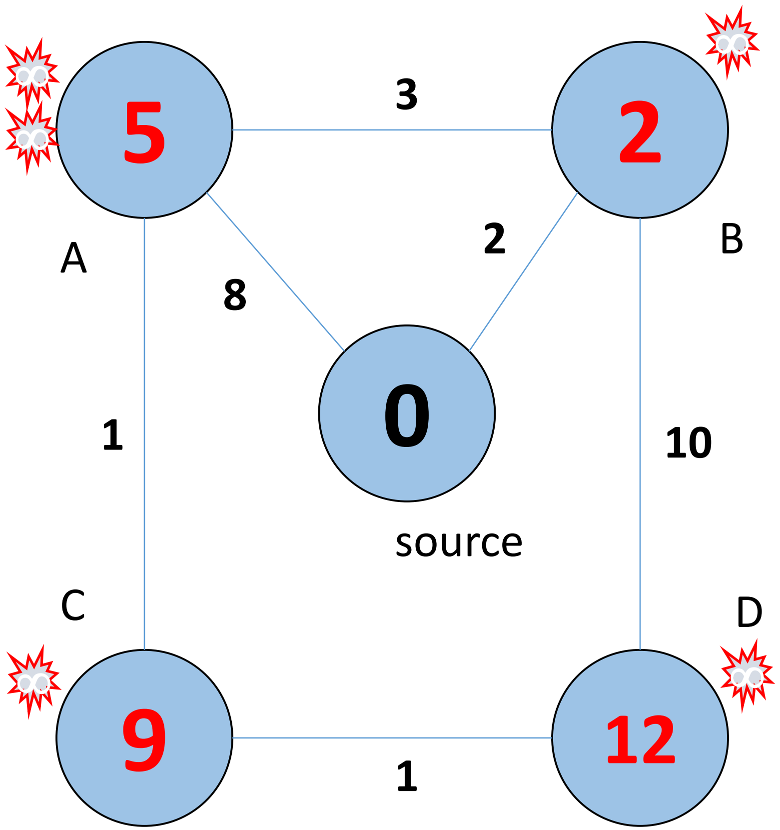
Example: SSSP in Pregel



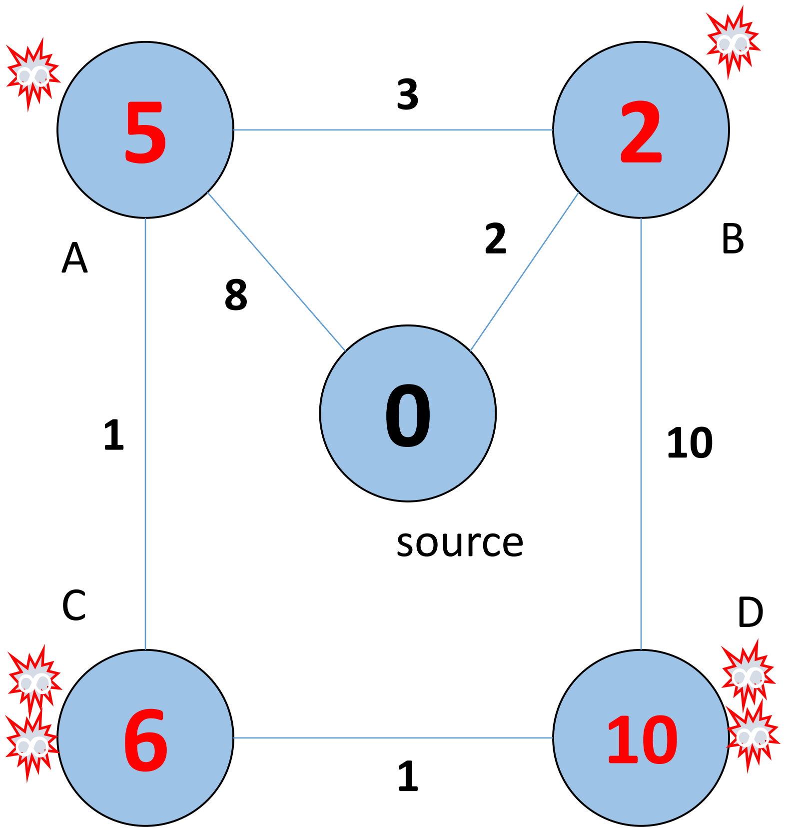
Example: SSSP in Pregel



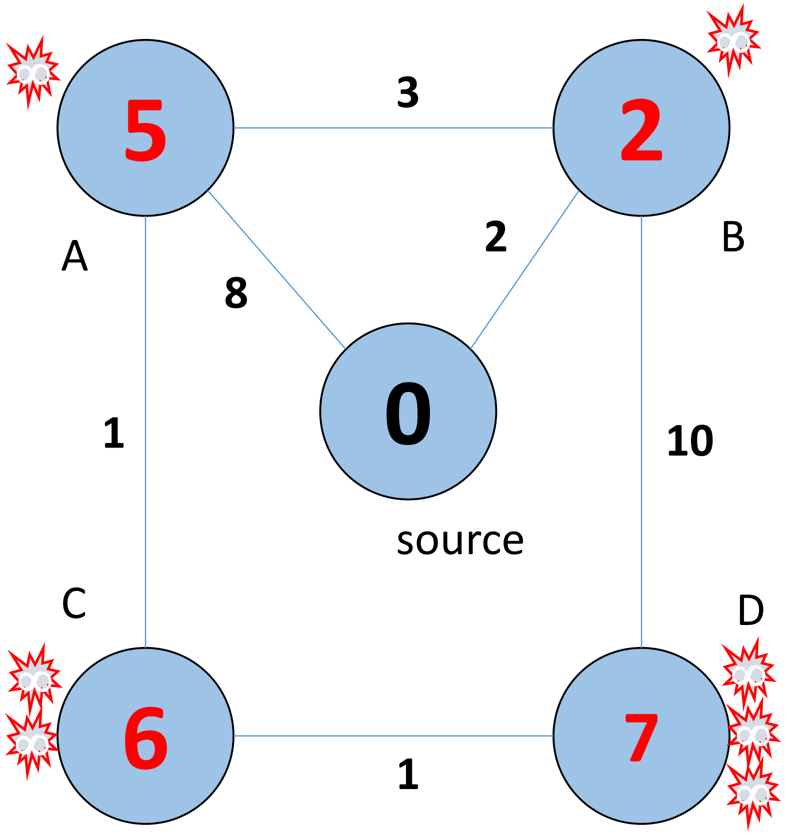
Example: SSSP in Pregel



Example: SSSP in Pregel



Example: SSSP in Pregel

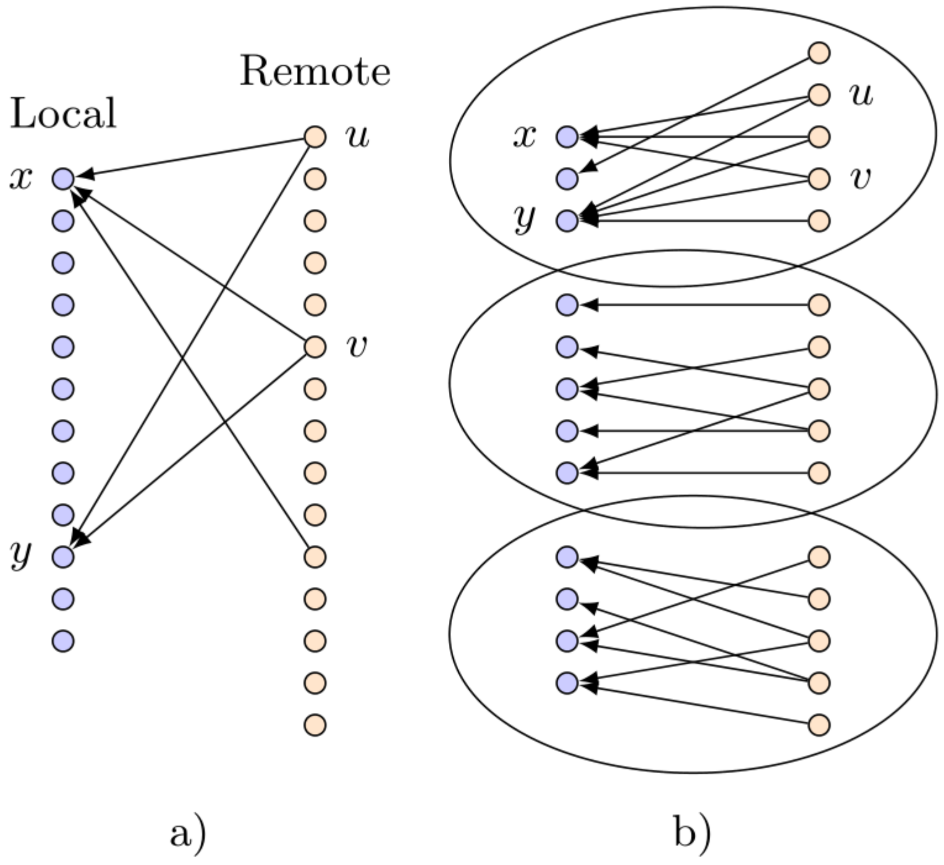


Pregel vs. MapReduce

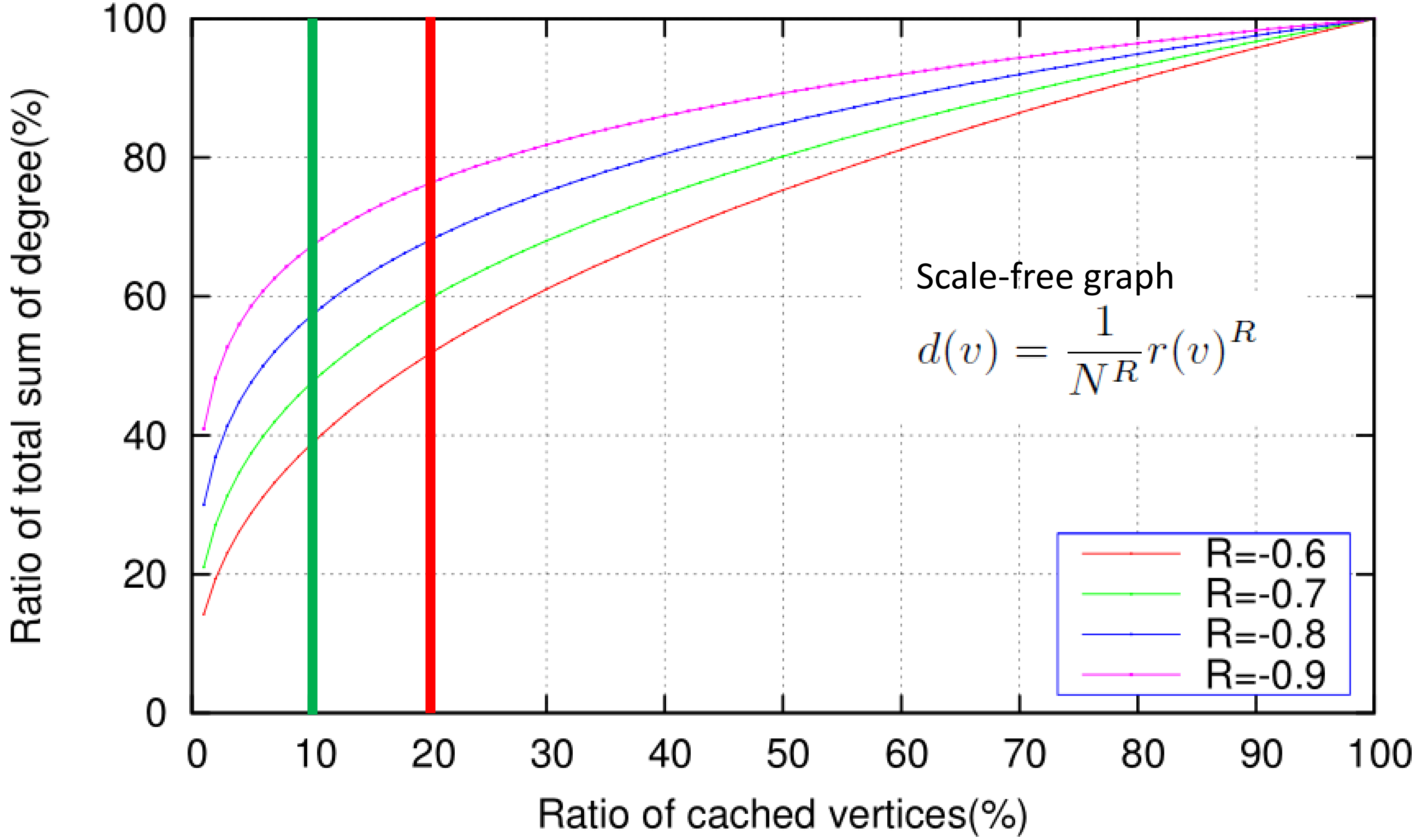
- Exploits fine-grained parallelism at node level
- Pregel doesn't move graph partitions over network, only messages among nodes are passed at the end of each iteration
- Many graph algorithms cannot be expressed using vertex-centric computation model intuitively and elegantly

Communication optimization

Bipartite view of a graph on a local machine



Message cache ("80/20" rule in real graphs)



Outline

- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- Offline analytics
- **Online query processing**
- Case study
- Graph generation
- Other graph representations and their applications

Online query processing

- Where latencies come from and asynchronous fan-out search

- Index-free query processing

Online query processing

- Where latencies come from and asynchronous fan-out search

- Index-free query processing

People search challenge in Facebook graph

- Among adult Facebook users, the average number of friends is 338.

338

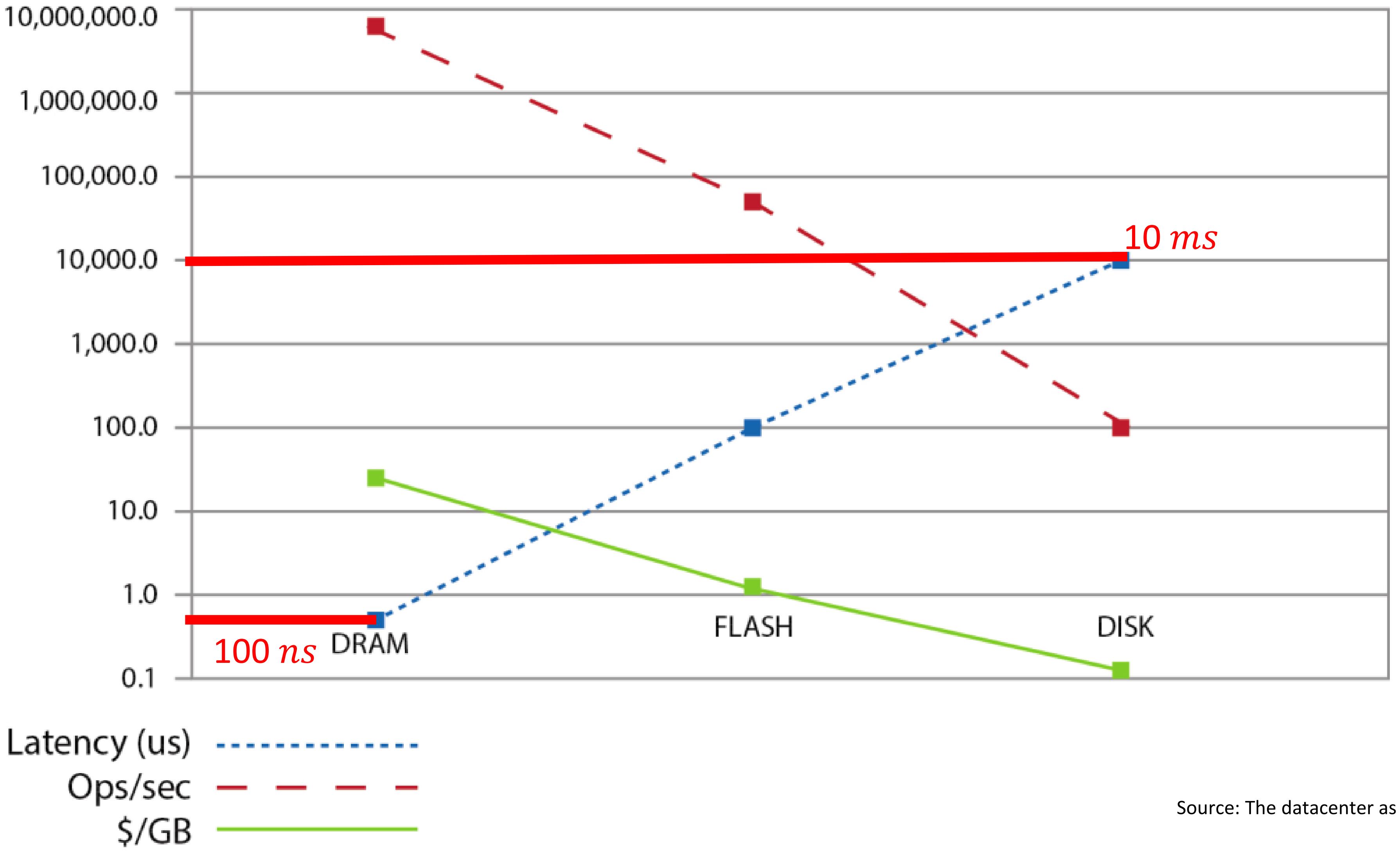
+338 x 338

+338 x 338 x 338

=38,729,054

Can we search a person in one's 3-hop neighborhood within 500 ms?

Latency, Bandwidth, and Capacity



Source: The datacenter as a computer (book)

Disk-based approach

338
+338 x 338
+338 x 338 x 338
=38,729,054 → 387,290,540 ms
= 4.5 days

each disk seek + read: > 10 ms

RAM-based approach

- DRAM latency: 100 ns

10 million reads/writes per second

1 million node-level read/write per second

38,729,054 nodes to access, it takes at least 38 seconds.

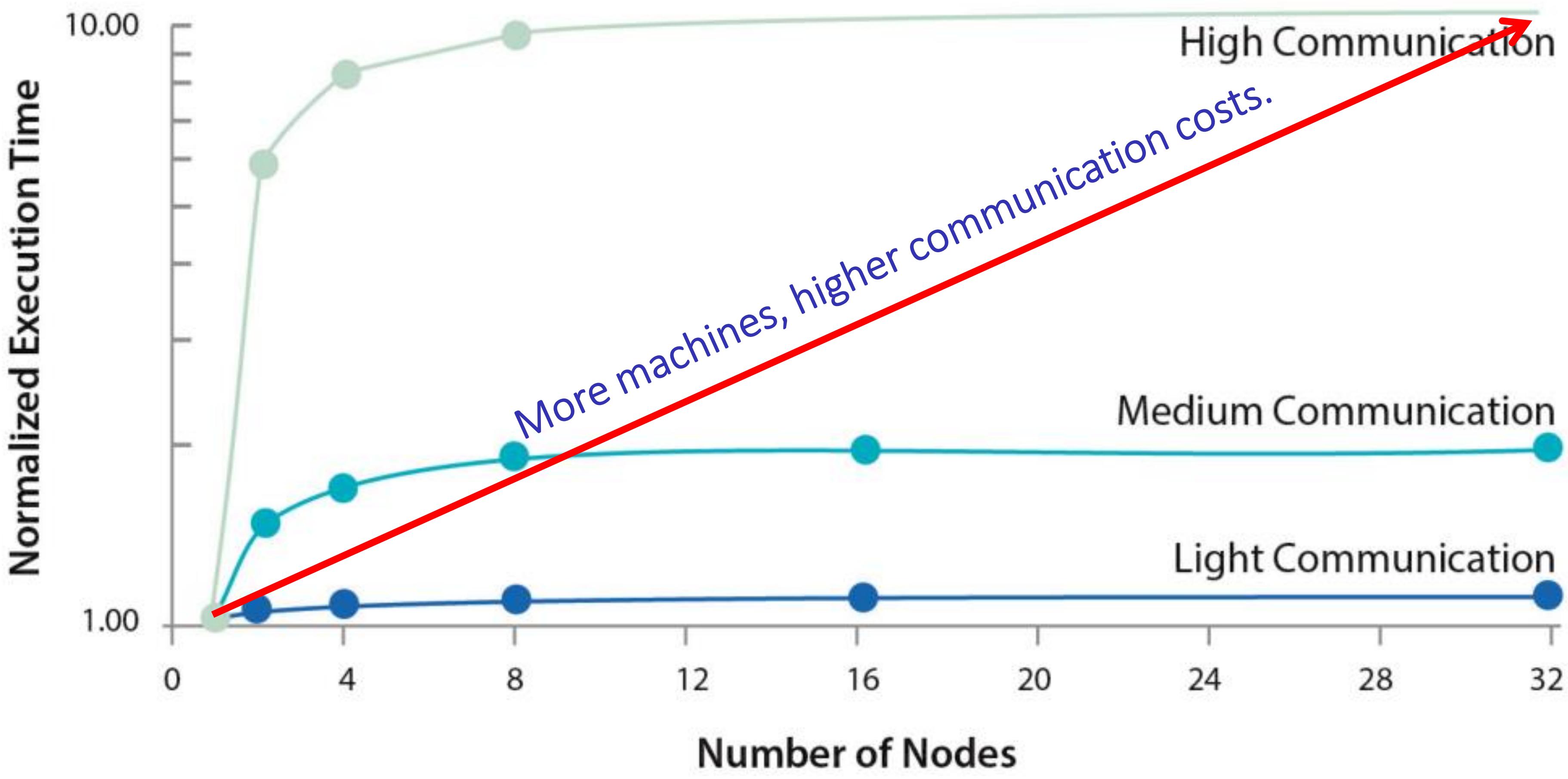
Hmm, no parallel data accesses yet!

How much time can we reduce with parallel data accesses?

Where do latencies come from?

- Algorithms ← Single-threaded, multi-threaded, synchronous, asynchronous
- Communications ← Multi-cores, Multi-nodes
- Storage ← DRAM, Flash, or Disk

Move computation, instead of data!



Source: The datacenter as a computer (book)

If you care about latency, do not use the shared-memory model in a distributed setting.

Lessons learned so far (how to reduce latencies)

- RAM (Hardware sometimes does matter a lot)
  - The stupid buy faster computers, smart ones write better programs?

- Avoid moving data

- ……

Lessons learned so far (how to reduce latencies)

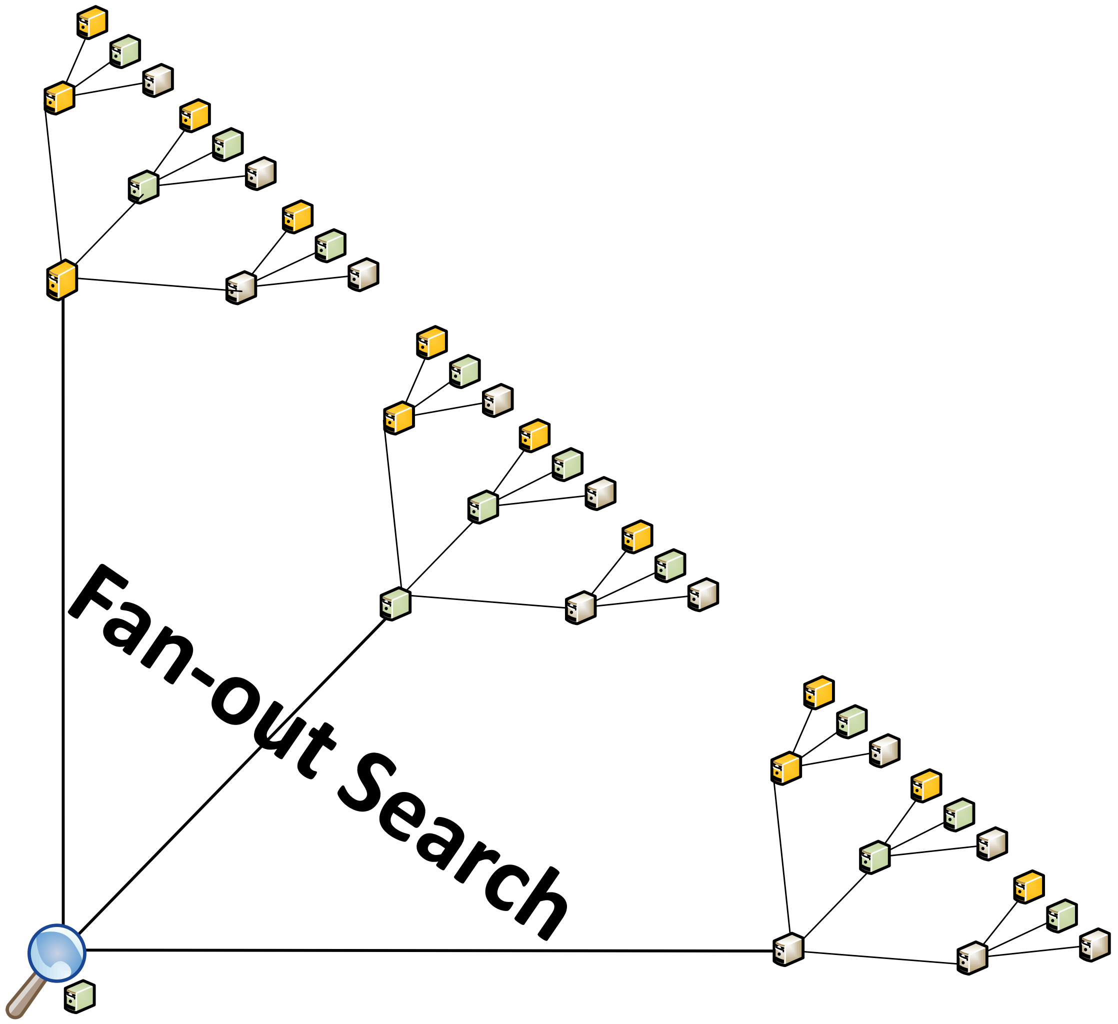
- RAM (Hardware sometimes does matter a lot)
  - The stupid buy faster computers, smart ones write better programs?

- Avoid moving data

- Avoid unnecessary synchronizations

Makes programming harder

Asynchronous fan-out search



Hop	Msg #	Node # per machine
1	n	$\frac{d}{n}$
2	n^2	$\frac{d^2}{n}$
3	n^3	$\frac{d^3}{n}$

n is the server count
d is the average degree

MORGAN & CLAYPOOL PUBLISHERS

The Datacenter as a Computer

An Introduction to the Design of Warehouse-Scale Machines

Second Edition

Luiz André Barroso
Jimmy Clidaras
Urs Hölzle

SYNTHESIS LECTURES ON COMPUTER ARCHITECTURE

Mark D. Hill, *Series Editor*

Online query processing

- Where latencies come from and fan-out search
- Index-free query processing

Online query example: subgraph matching

Procedure:

1. Break a graph into basic units (edges, paths, frequent subgraphs, ...)
2. Build index for every possible basic unit
3. Decompose a query into multiple basic unit queries, and join the results.

Query Index Examples

Algorithms	Index Size	Index Time	Update Cost
Ullmann [Ullmann76], VF2 [CordellaFSV04]	-	-	-
RDF-3X [NeumannW10]	$\mathbf{O}(m)$	$\mathbf{O}(m)$	$\mathbf{O}(d)$
BitMat [AtreCZH10]	$\mathbf{O}(m)$	$\mathbf{O}(m)$	$\mathbf{O}(m)$
Subdue [HolderCD94]	-	**Exponential**	$\mathbf{O}(m)$
SpiderMine [ZhuQLYHY11]	-	**Exponential**	$\mathbf{O}(m)$
R-Join [ChengYDYW08]	$\mathbf{O}(nm^{1/2})$	$\mathbf{O}(n^4)$	$\mathbf{O}(n)$
Distance-Join [ZouCO09]	$\mathbf{O}(nm^{1/2})$	$\mathbf{O}(n^4)$	$\mathbf{O}(n)$
GraphQL [HeS08]	$O(m + nd^r)$	$O(m + nd^r)$	$\mathbf{O}(d^r)$
Zhao [ZhaoH10]	$\mathbf{O}(nd^r)$	$\mathbf{O}(nd^r)$	$\mathbf{O}(d^L)$
GADDI [ZhangLY09]	$O(nd^L)$	$O(nd^L)$	$\mathbf{O}(d^L)$

Index-based subgraph matching [Sun VLDB 2012]

Query Index Examples

Algorithms	Index Size for Facebook	Index Time for Facebook	Query Time on Facebook (s)
Ullmann [Ullmann76], VF2 [CordellaFSV04]	-	-	$>$1000
RDF-3X [NeumannW10]	1T	$>$20 days	$>$48
BitMat [AtreCZH10]	2.4T	$>$20 days	$>$269
Subdue [HolderCD94]	-	$>$ 67 years	-
SpiderMine [ZhuQLYHY11]	-	$>$ 3 years	-
R-Join [ChengYDYW08]	$>$175T	$> 10^{15}$ years	$>$200
Distance-Join [ZouCO09]	$>$175T	$> 10^{15}$ years	$>$4000
GraphQL [HeS08]	$>$13T(r=2)	$>$ 600 years	$>$2000
Zhao [ZhaoH10]	$>$12T(r=2)	$>$ 600 years	$>$600
GADDI [ZhangLY09]	$> 2 \times 10^5$T (L=4)	$> 4 \times 10^5$ years	$>$400

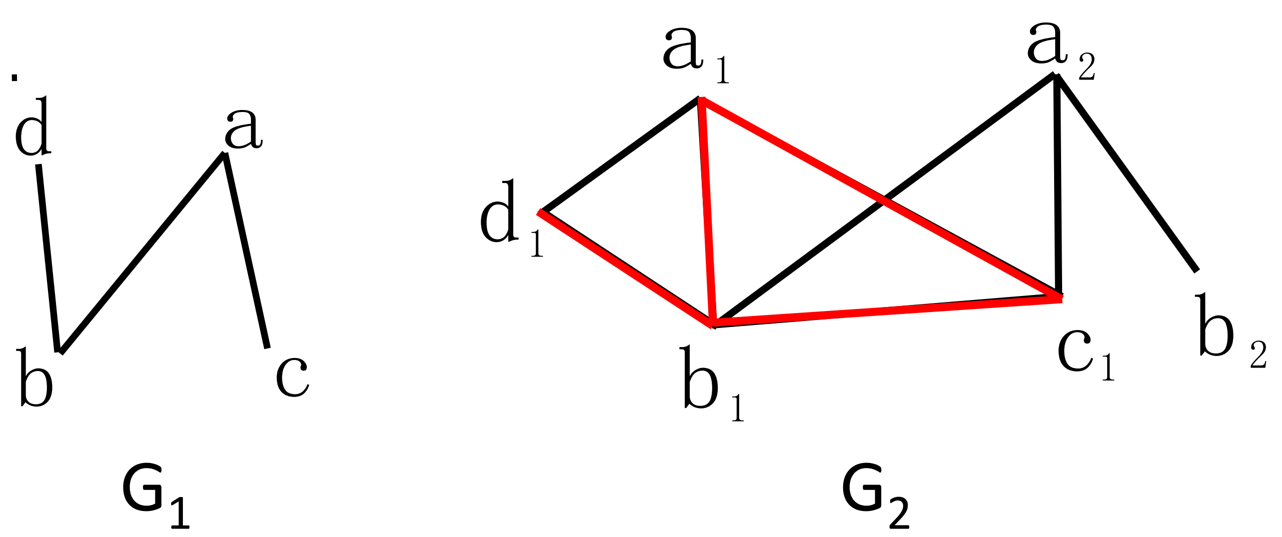
Index-based subgraph matching [Sun VLDB 2012]

Case study: distributed subgraph matching

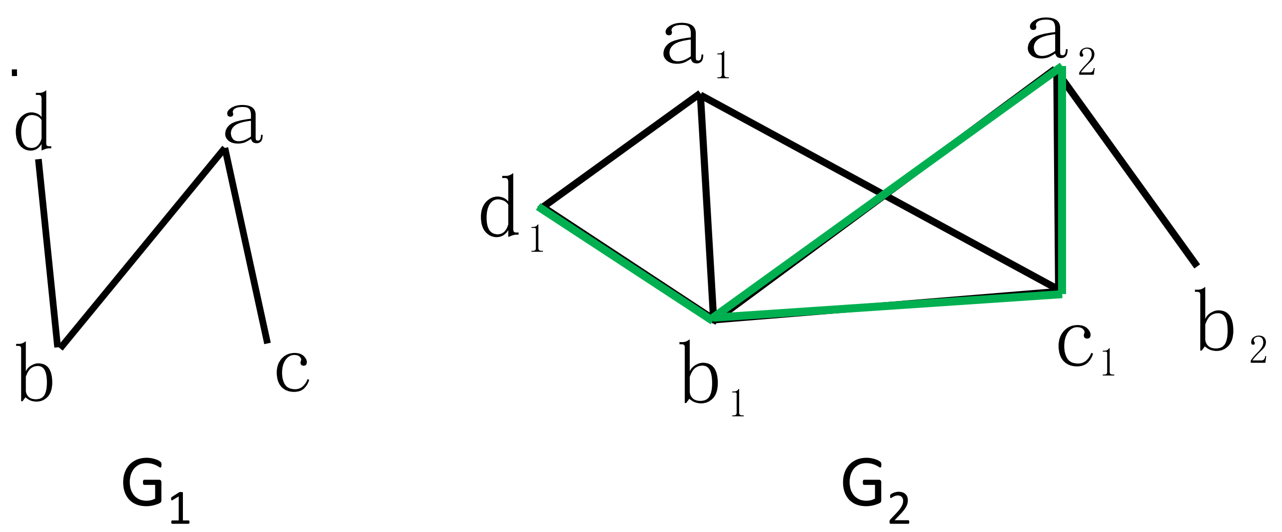
Procedure:

1. Break a query into basic units
2. **Match the basic units in parallel on the fly**
3. Join the results

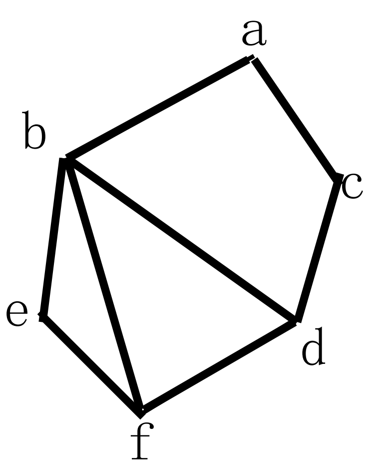
Subgraph matching

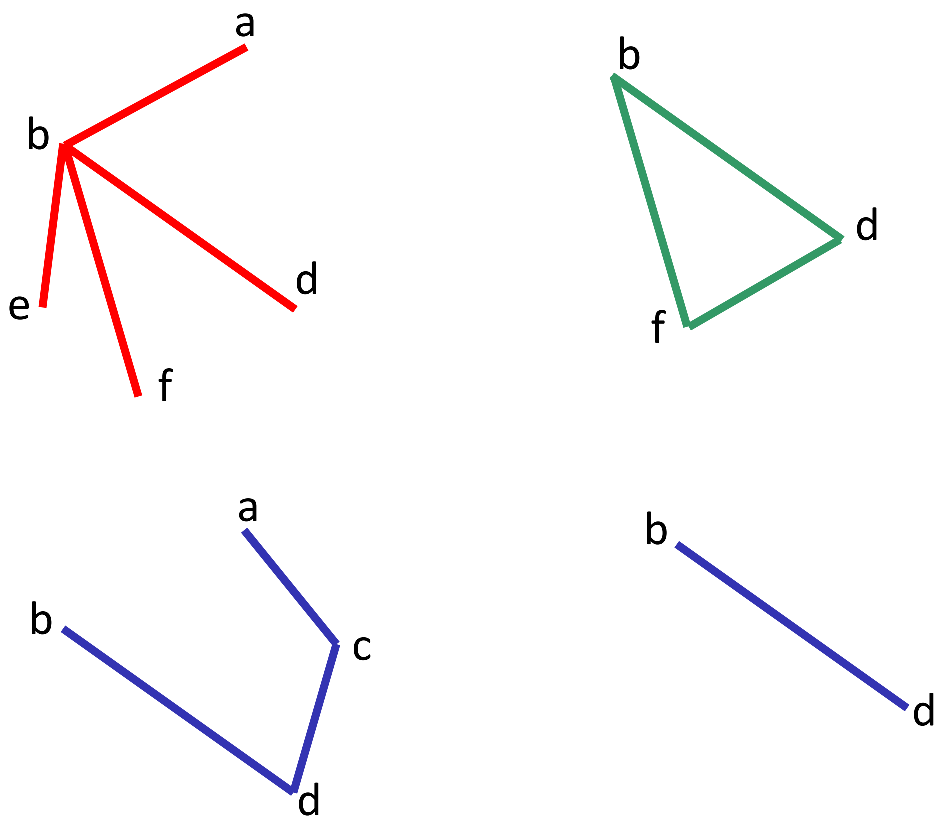


Subgraph matching



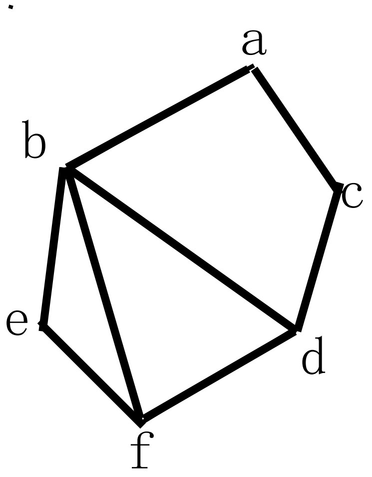
Basic unit for distributed subgraph matching

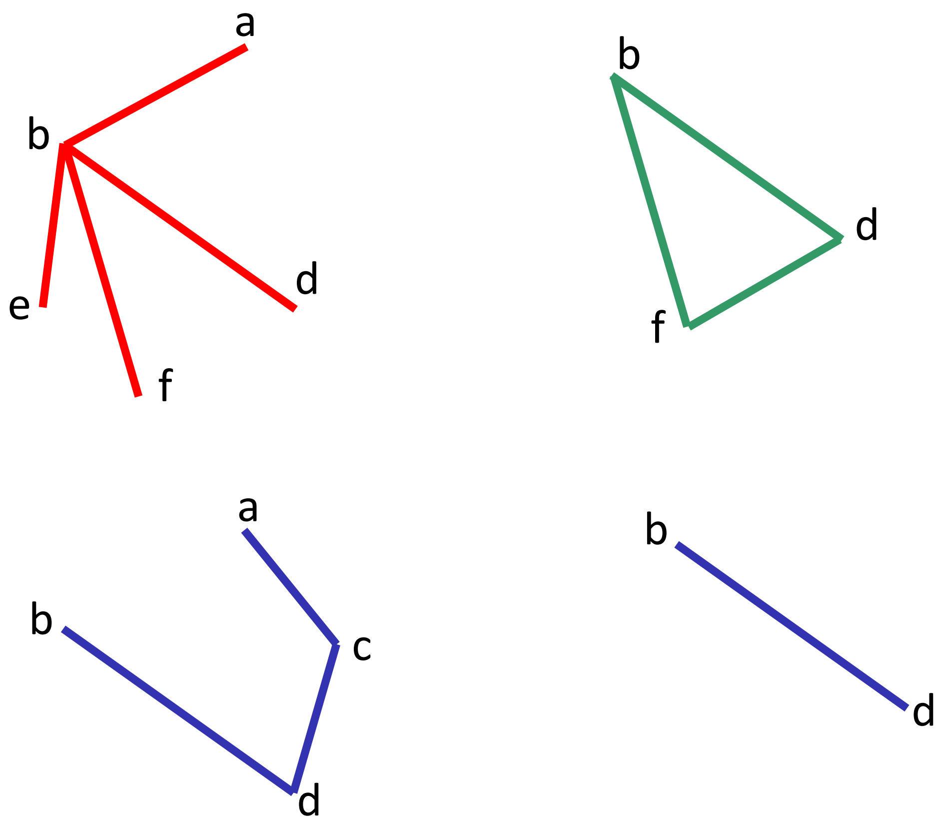




As a basic unit, which one is the best?

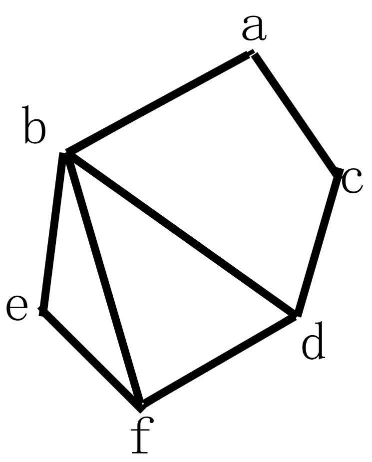
Basic unit for distributed subgraph matching

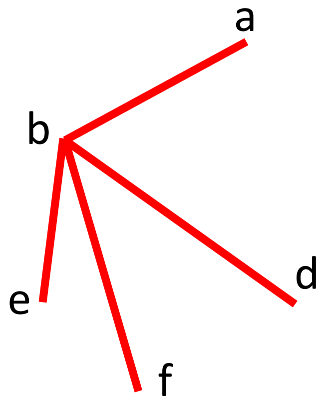




As a basic unit, which one is the best?

Basic unit for distributed subgraph matching



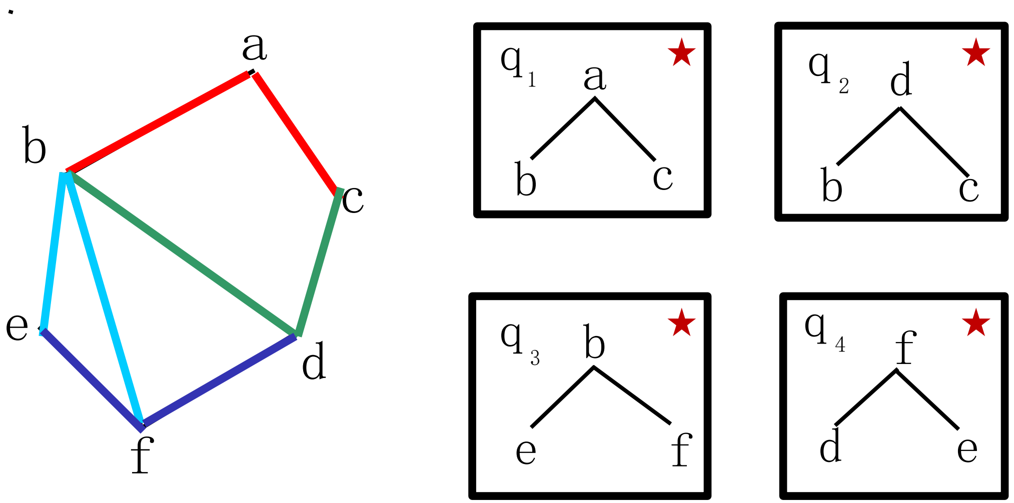


Twig

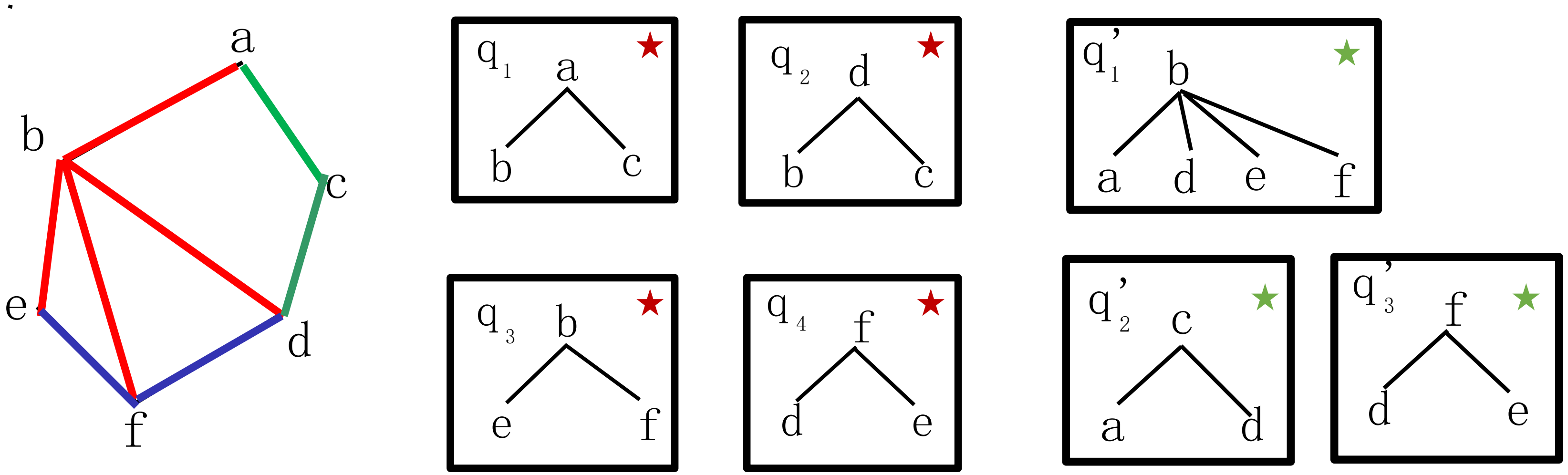
- Easy to decompose
- Height is always one
  - It at most needs to cross the network once

As a basic unit, which one is the best?

Query decomposition



Query decomposition



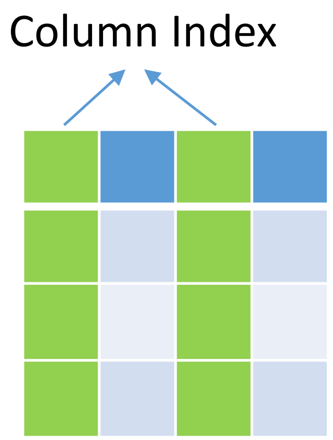
Query optimization problems

- How to choose a good query decomposition
- How to choose a good execution order
- How to choose a good join order

Outline

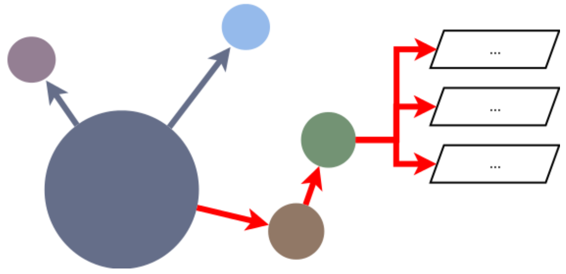
- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- Offline analytics
- Online query processing
- **Case study**
- Graph generation
- Other graph representations and their applications

How to Serve the Knowledge?



Triplets/RDF

- Table + column indexes
- Free text search
- Native graph exploration



Representative Knowledge Serving Systems

System	Query Language	Known Scalability	Distributed
sw-Store	SPARQL	55M	V
RDFJoin	SPARQL	44M	
RDFKB	-	44M	
BitMat	SPARQL-like	47M	
RDF-3x	SPARQL	51M	V
Virtuoso	SPARQL	1,068M	V
Trinity	LIKQ, TQL, SPARQL	24.6B	V

Reference: A survey of RDF storage approaches (David C. FAYE, et al.)

Problem and Our Goal

Problem

- KG is a massive entity network
  - The most valuable part is its rich relationships, but
  - Currently mainly used via entity indexes
  - Cannot answer queries requiring accesses of 2+ hop relations

Goal

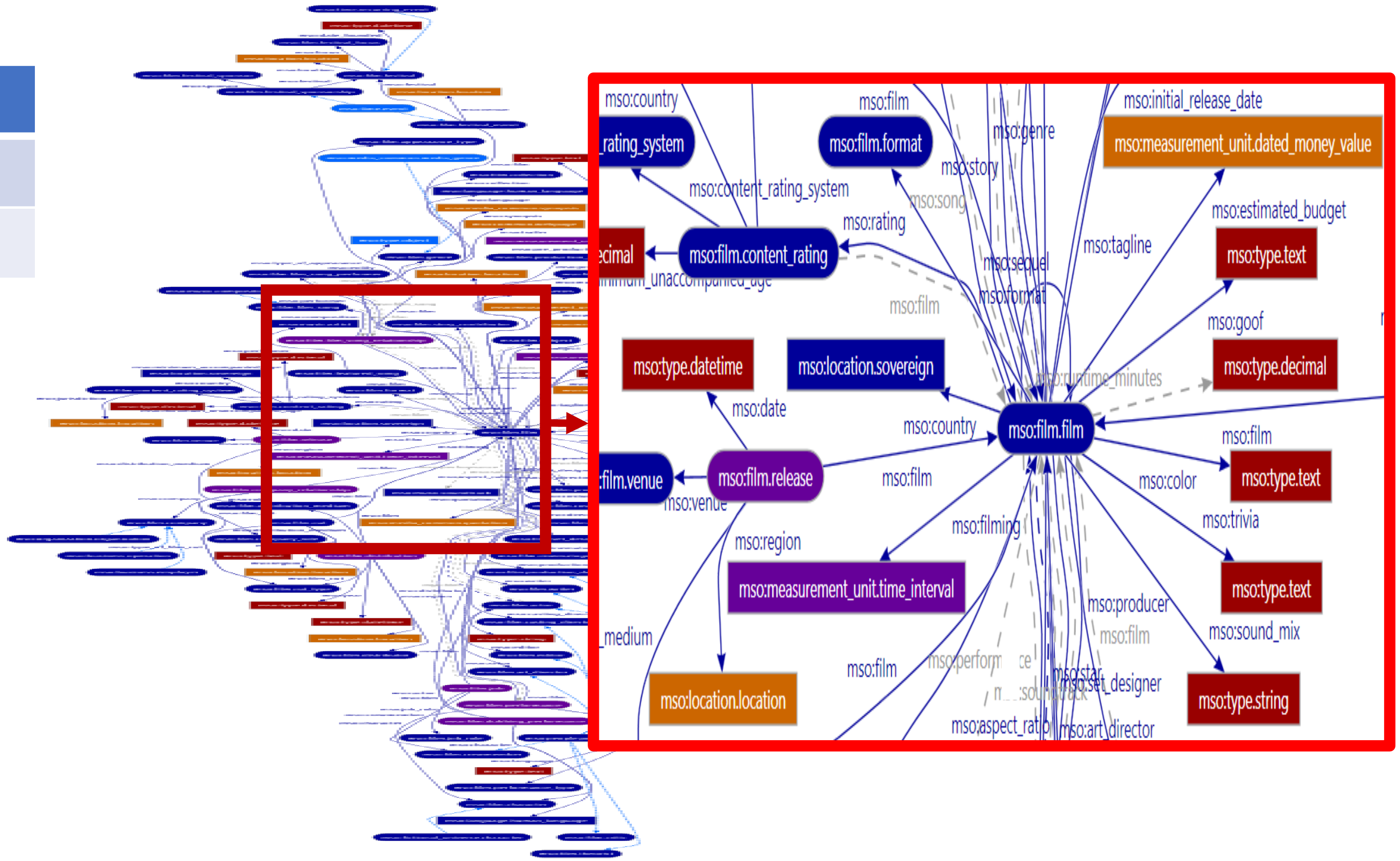
This project is to **provide advanced real-time knowledge graph serving operators**

- Serve the full-scale KG
- Make KG accessible in real time
- Provide advanced graph operators

Challenges of Serving KG

Data size	
Raw RDF data	5T+
Triple Facts	25B+

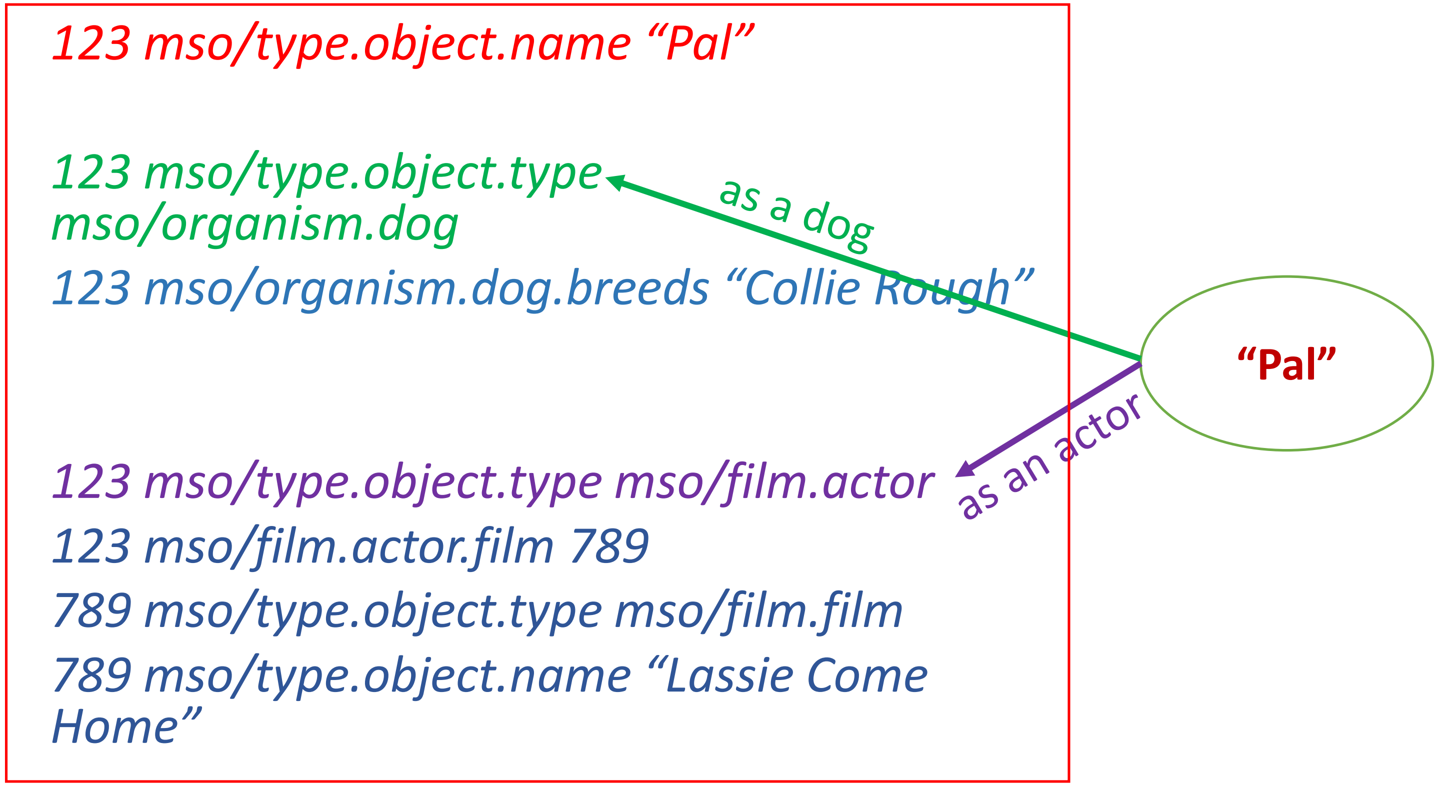
- Complex data schema
  - Rich relations



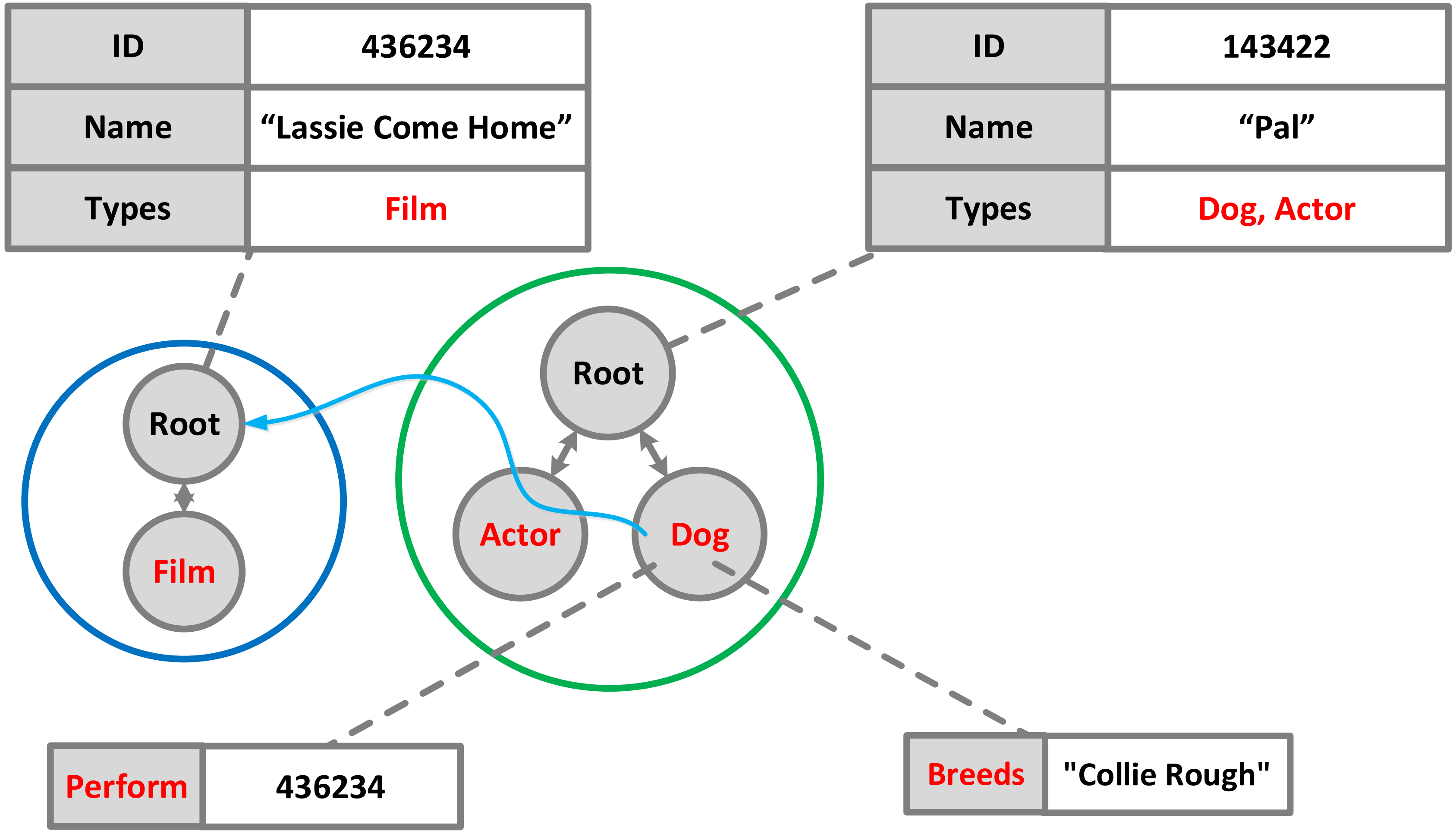
Challenges of Serving KG

Data size	
Raw RDF data	5T+
Triple Facts	25B+

- Complex data schema
  - Rich relations
  - Multi-typed entities

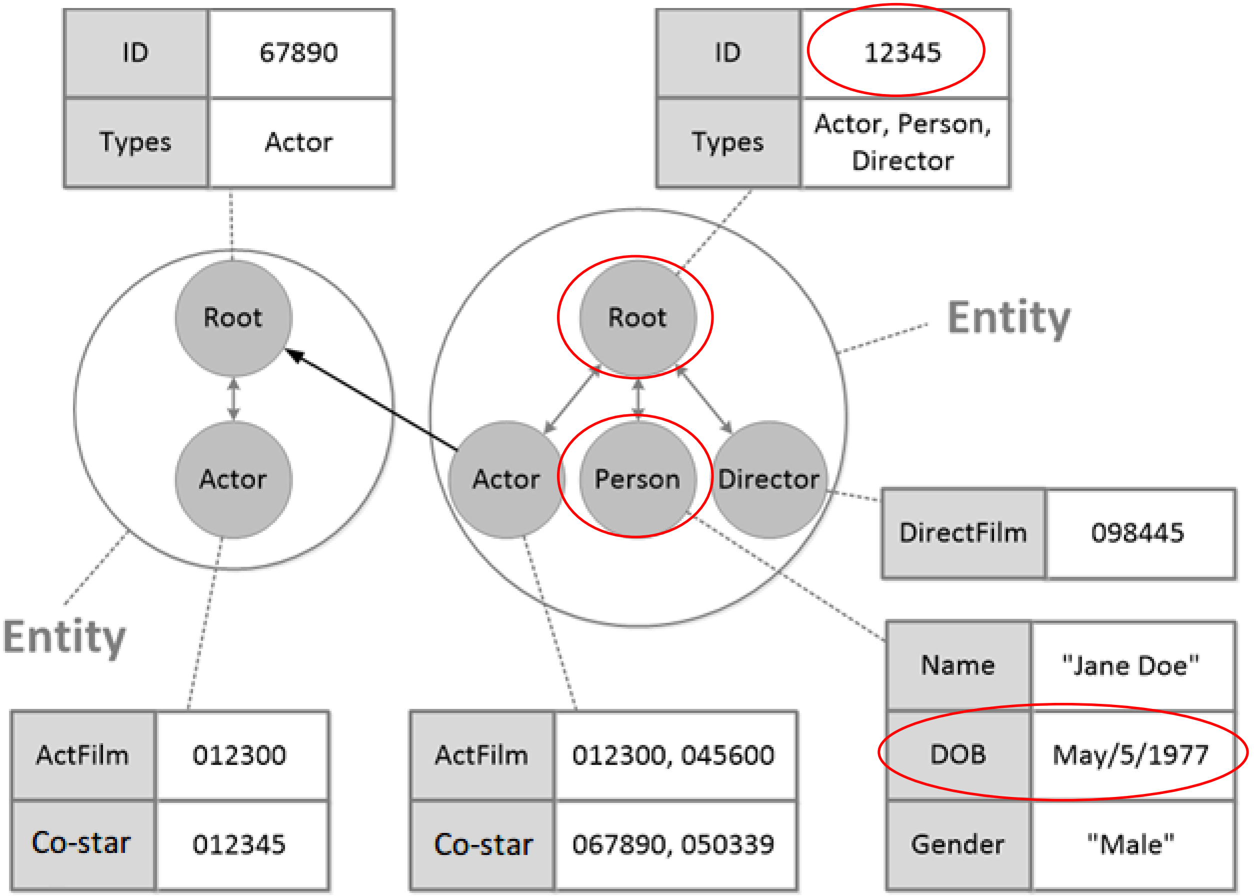


A Strongly Typed System for RDF



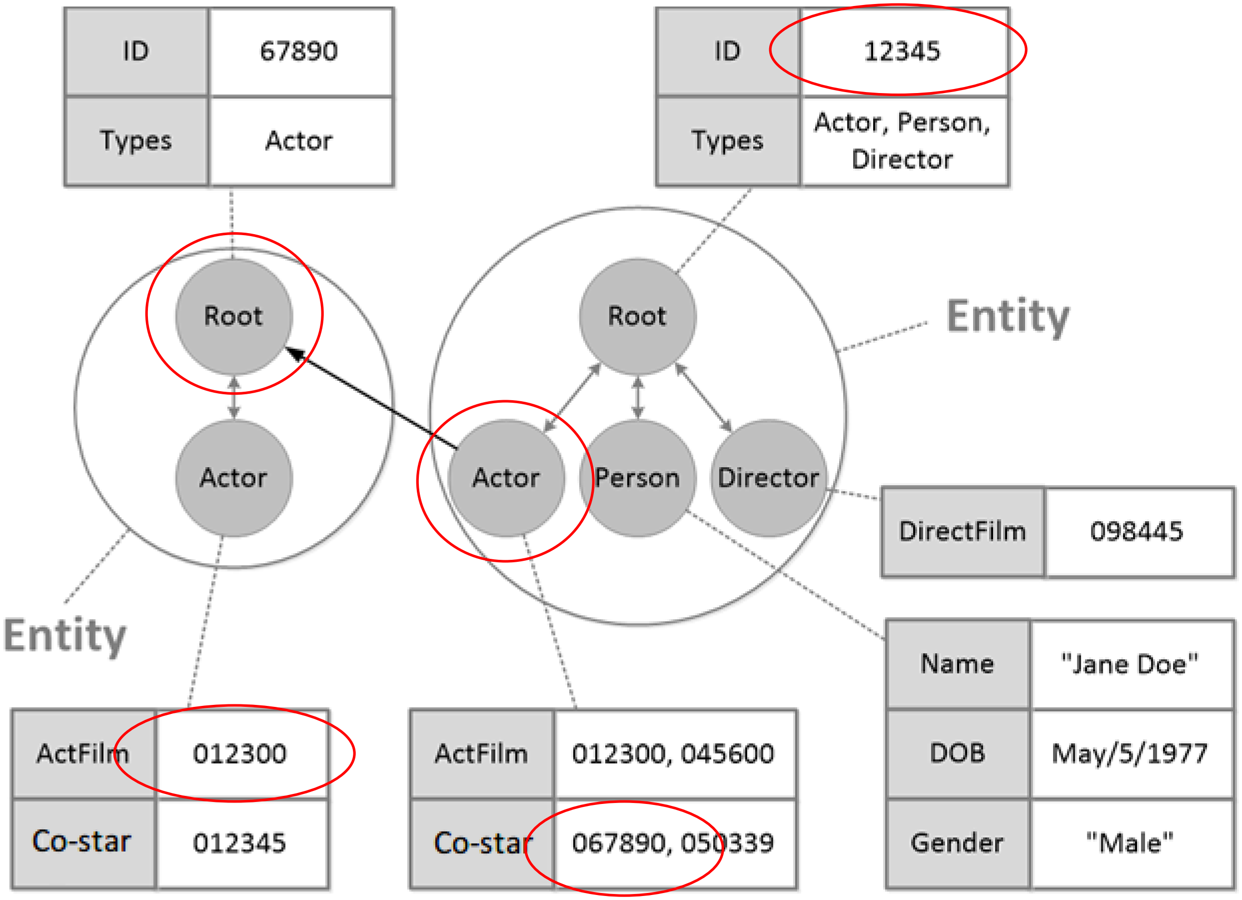
Models Multi-Typed Entities in a Strongly Typed Manner

Strongly-typed data accesses



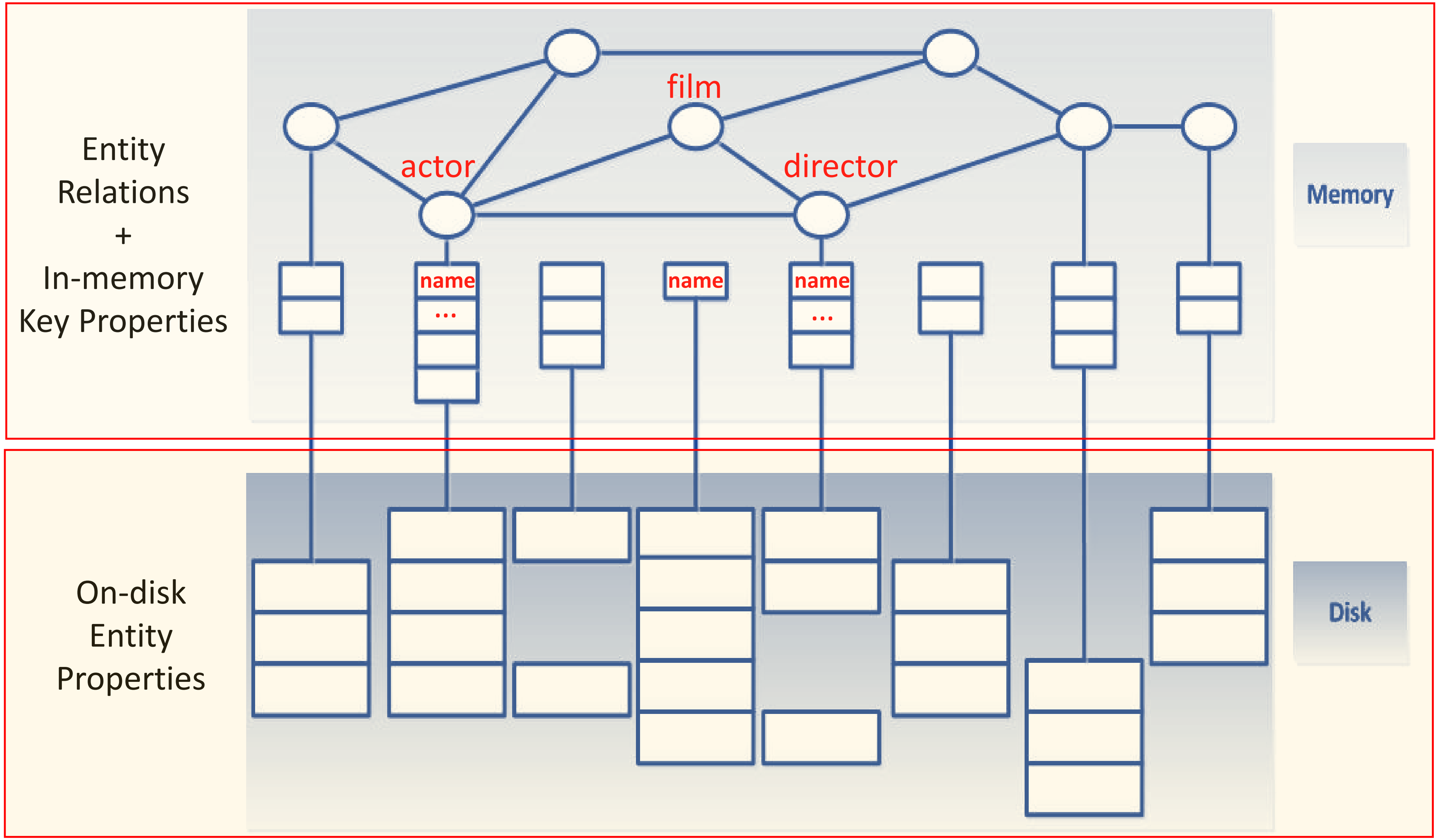
Get the DOB of entity 12345

Strongly-typed data accesses

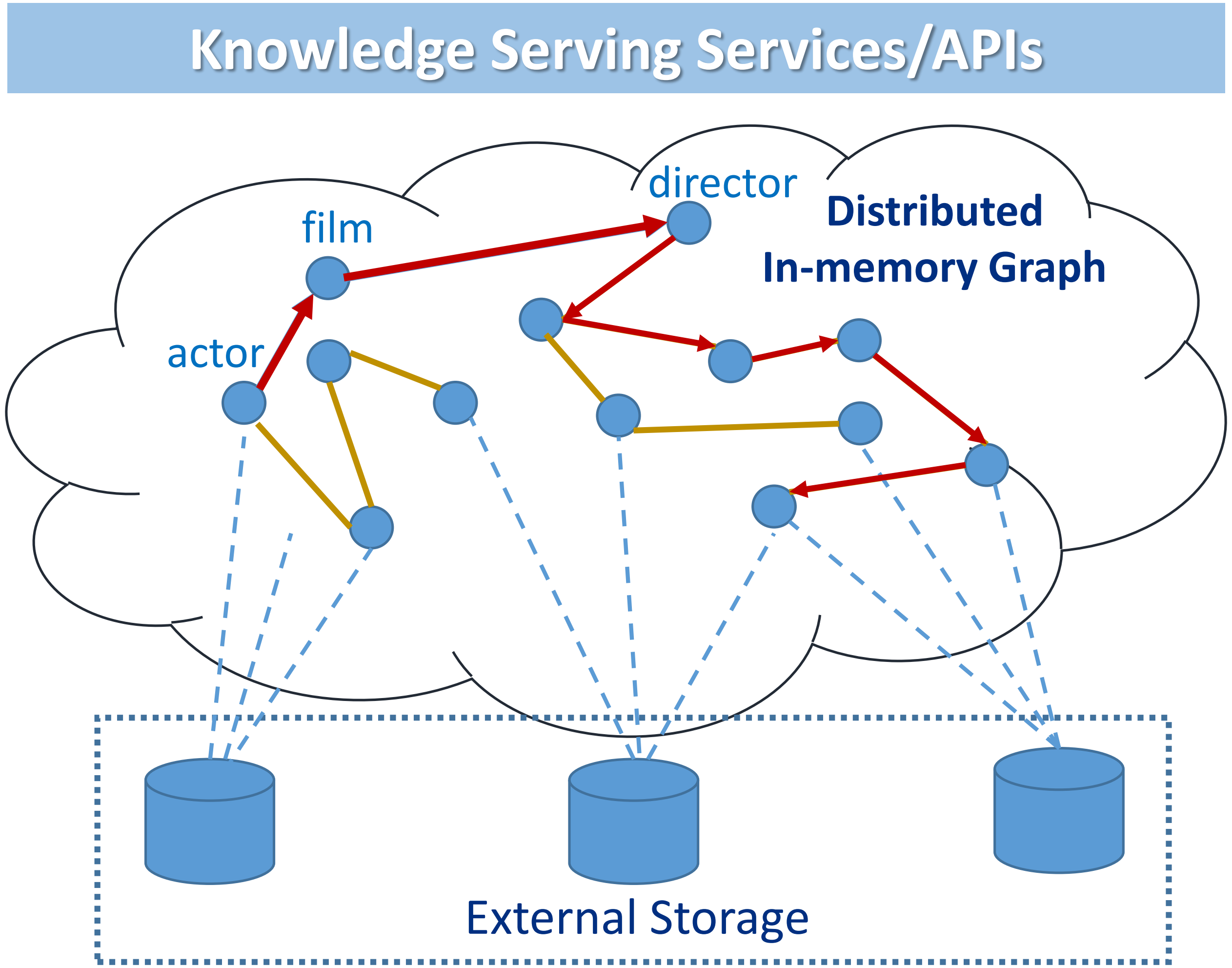


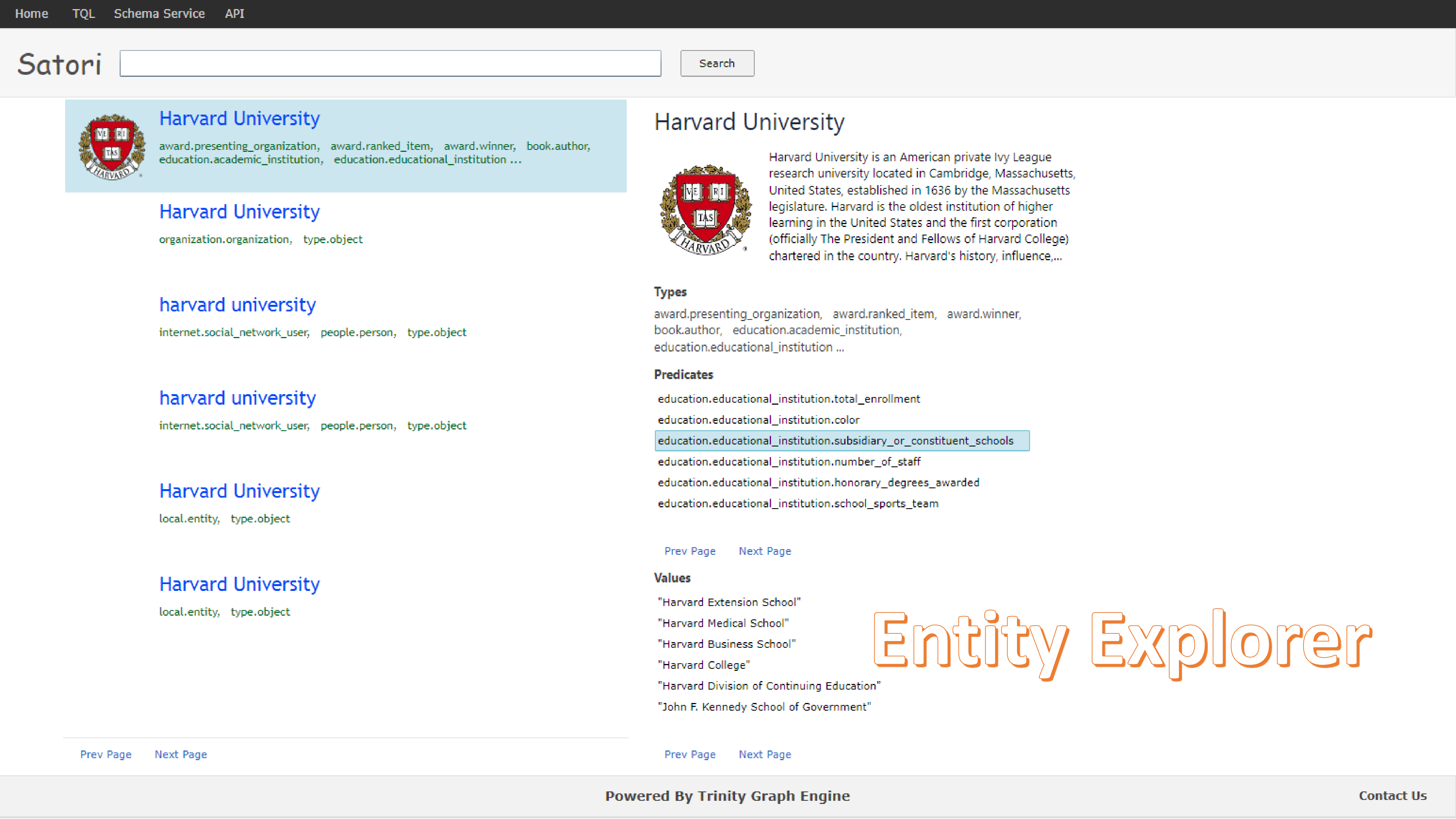
Get the films of actors co-starring with entity 12345

Storage Architecture



Query KG via Graph Exploration




Home
TQL
Schema Service
API
Satori
Search
Harvard University
award.presenting_organization, award.ranked_item, award.winner, book.author, education.academic_institution, education.educational_institution ...
Harvard University
organization.organization, type.object
harvard university
internet.social_network_user, people.person, type.object
harvard university
internet.social_network_user, people.person, type.object
Harvard University
local.entity, type.object
Harvard University
local.entity, type.object
Prev Page
Next Page
Harvard University
Harvard University is an American private Ivy League research university located in Cambridge, Massachusetts, United States, established in 1636 by the Massachusetts legislature. Harvard is the oldest institution of higher learning in the United States and the first corporation (officially The President and Fellows of Harvard College) chartered in the country. Harvard's history, influence,...
Types
award.presenting_organization, award.ranked_item, award.winner, book.author, education.academic_institution, education.educational_institution ...
Predicates
education.educational_institution.total_enrollment
education.educational_institution.color
education.educational_institution.subsidiary_or_constituent_schools
education.educational_institution.number_of_staff
education.educational_institution.honorary_degrees_awarded
education.educational_institution.school_sports_team
Prev Page
Next Page
Values
"Harvard Extension School"
"Harvard Medical School"
"Harvard Business School"
"Harvard College"
"Harvard Division of Continuing Education"
"John F. Kennedy School of Government"
Prev Page
Next Page
Entity Explorer
Powered By Trinity Graph Engine
Contact Us

Schema Graph

Meta Graph of Satori

Schema Type:

mso/people.person Go

Schema Path:

mso/people.person → mso/book.author Go

Fields:

.bust_measurement	mso/type.decimal
.date_of_birth	mso/type.datetime
.eye_color	mso/type.text
.first_name	mso/type.string
.hair_color	mso/type.text
.height	mso/type.decimal
.hips_measurement	mso/type.decimal
.last_name	mso/type.string
.waist_measurement	mso/type.decimal
.weight	mso/type.decimal

Links:

.business_employment_tenure	mso/business.employment_tenure
.children	mso/people.person
.city_of_birth	mso/location.location

mso/people.person	.quotation	mso/media_common.quotation
mso/media_common.quotation	.character	mso/fictional_universe.character
mso/fictional_universe.character	.appears_in_these_fictional_universes	mso/fictional_universe.universe
mso/fictional_universe.universe	.literary_series	mso/book.literary_series
mso/book.literary_series	.author	mso/book.author

Schema Graph Services

Satori Knowledge Graph Access API

API Names	Availability	Description
GetEntityIdByName	Available	Gets a list of Trinity entity Ids by the specified entity name.
GetPredicatesByEntityId	Available	Gets a list of predicates for the entity with the specified Trinity entity Id.
GetValuesByEntityPredicate	Available	Gets the values of the specified predicates for the specified entity.
GetSubjectsByPredicateObject	Available	Gets the subjects for the given object and a predicate.
GetEntityIdBySatoriId	Available	Gets the corresponding Trinity entity Id for the specified Satori Guid.
GetSatoriIdByEntityId	Available	Gets the corresponding Satori Guid for the specified Trinity entity Id.
GetRankedEntityIdByName	Available	Gets a list of Trinity entity Ids by the specified entity name sorted by their static rank.
GetScoredValuesByEntityPredicate	Available	Gets the values of the specified predicates for the specified entity, sorted by confidence score.
GetSortScoredValuesByEntityPredicate	Available	Gets the values of the specified predicates for the specified entity, sorted by the column index (1 for...
GetEntityDescription	Available	Gets the description of the specified entityid.

Testing: GetScoredValuesByEntityPredicate

Please input test parameters below:

EntityId: 24604518639751

Predicate: mso/film.actor.film

Submit

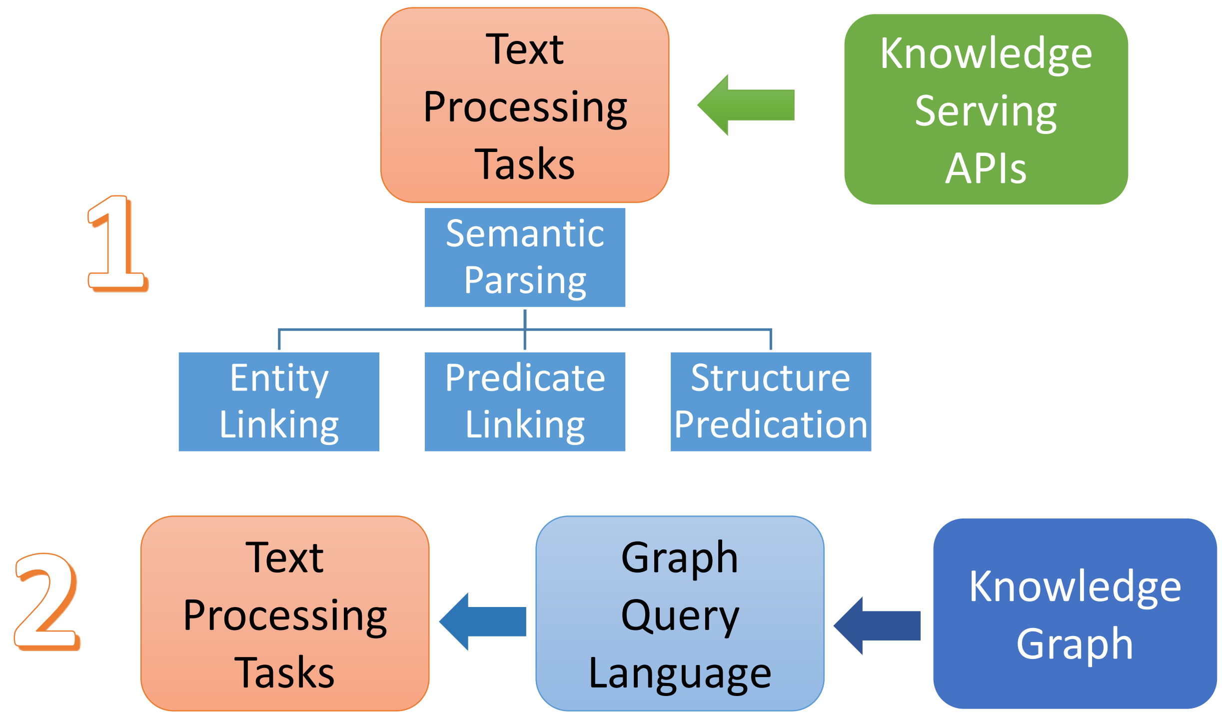
PredicateValue	ConfidenceScore	OverallScore
2987469205879	0.71	1.311128
116281907553515	0.71	1.409593
265920831012309	0.71	1.416611
58184534540412	0.71	1.710736
237856925167352	0.71	1.228339
57628320423344	0.71	1.272192

Page 1 of 1

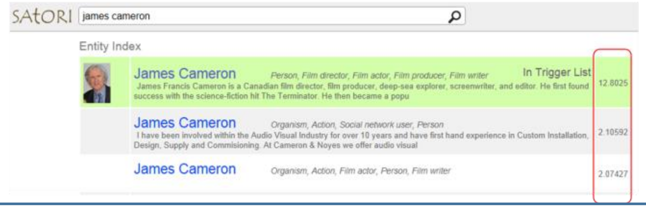
Knowledge Serving APIs

Knowledge Serving for Text Processing

APIs for Knowledge Access
List<string[]> GetEntityIdByName (string entityName)
List<string> GetPredicatesByEntityId (string entityId)
List<string[]> GetValuesByEntityPredicate (string entityId, string predicate)
……



- List<string[]> GetEntityIdByName (string entityName)
  - Input
    - An entity name
  - Output
    - A list of <entity id, entity ranking score> pairs



- List<string> GetPredicatesByEntityId (string entityId)
  - Input
    - An entity id
  - Output
    - A set of predicates

- List<string[]> GetValuesByEntityPredicate (string entityId, string predicate)
  - Input
    - An entity id and a predicate
  - Output
    - A set of <entity id, confidence score> pairs

Entity Disambiguation/Type Resolving

Who are the advisees of Michael Jordan?

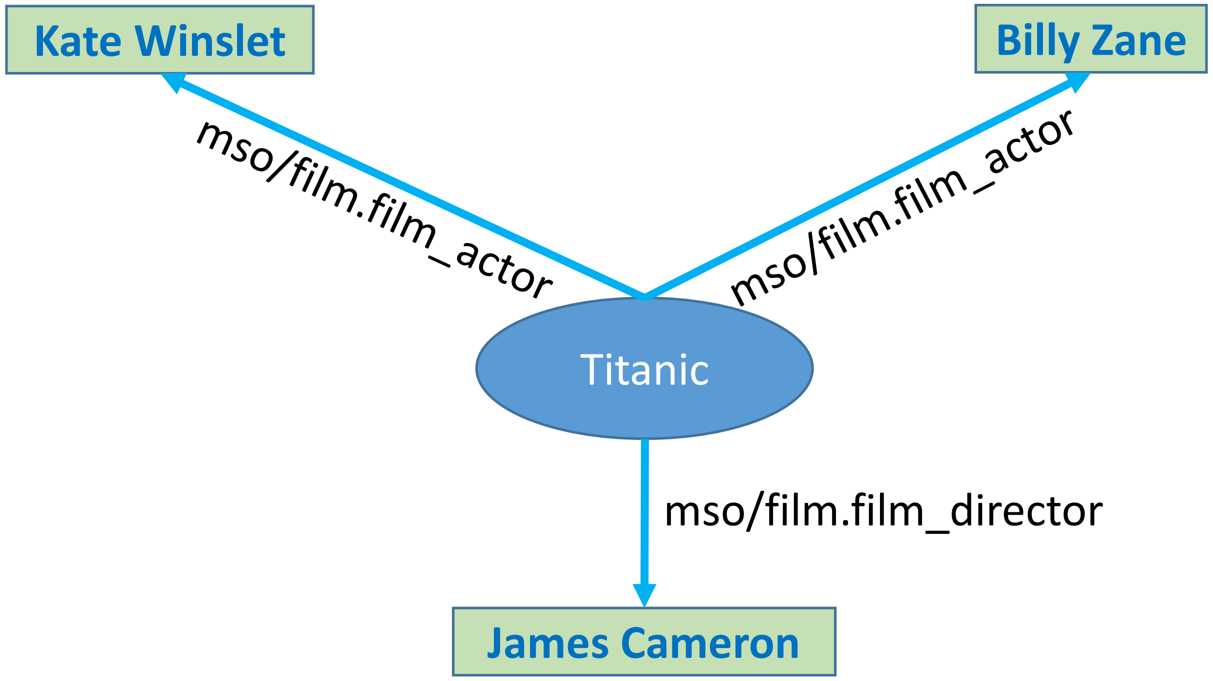
Which Michael?

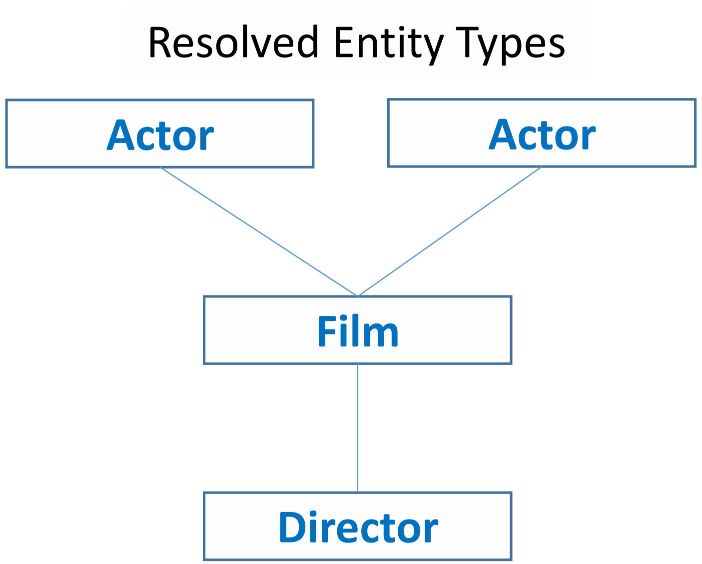
- Michael Jordan (footballer) (born 1986)
- Michael-Hakim Jordan (basketball player) (born 1977)
- Michael Jordan (Irish politician)
- Michael I. Jordan (Professor) (born 1957)
- ....

8234993200123	mso/education.academic.advisees	"Andrew Ng"
8234993200123	mso/type.object.name	“Michael Jordan"
8234993200123	mso/people.person.profession	“Professor"

Discover Linking Entities

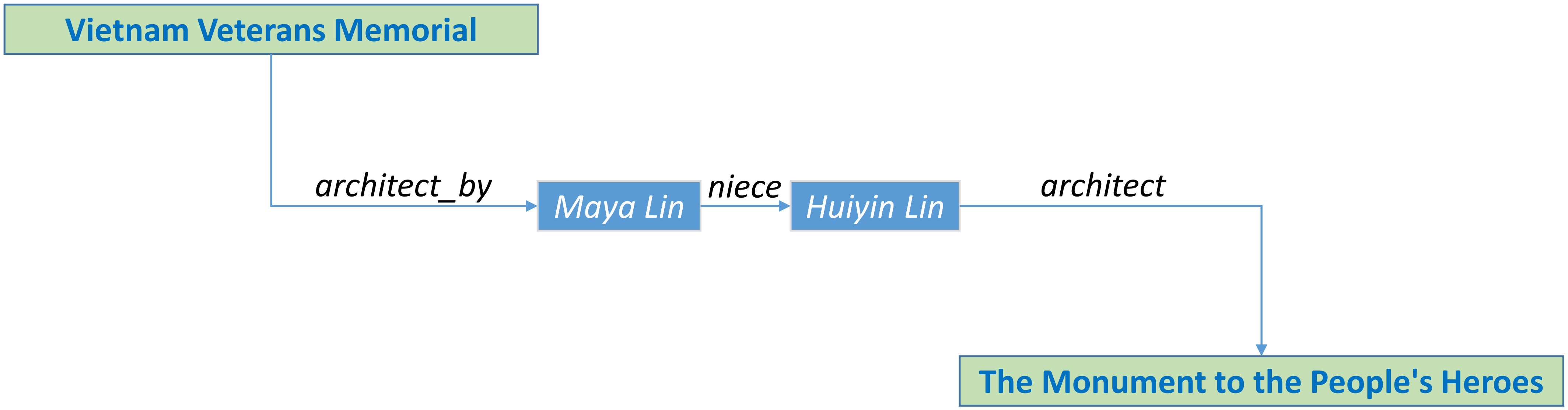
Given three entities “*Kate Winslet*”, “*Billy Zane*”, and “*James Cameron*”

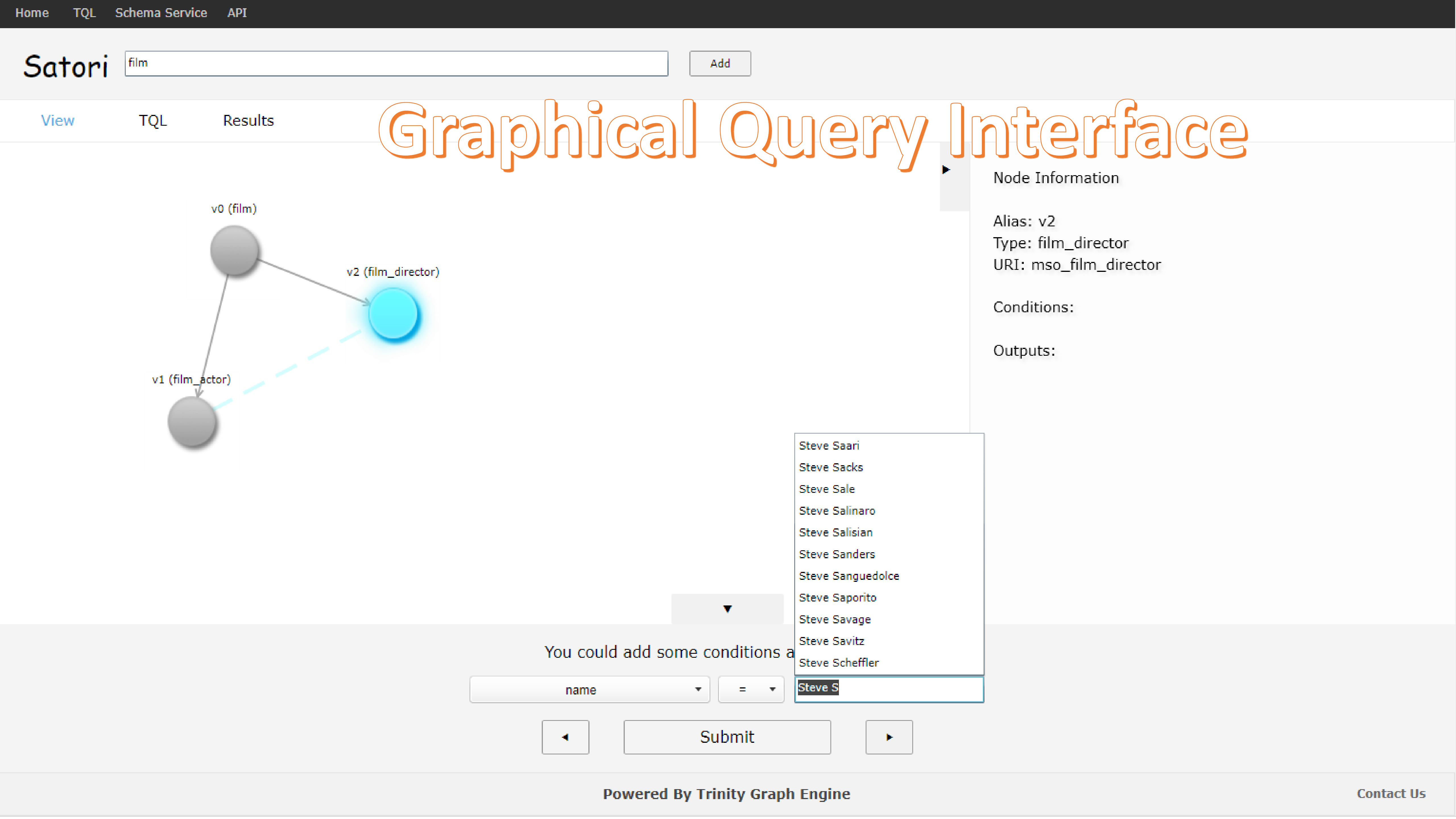




Discover Linking Relations

Given two entities *“Vietnam Veterans Memorial” and “The Monument to the People's Heroes”*




Home
TQL
Schema Service
API
Satori
film
Add
View
TQL
Results
Graphical Query Interface
Node Information
v0 (film)
v2 (film_director)
v1 (film_actor)
Alias: v2
Type: film_director
URI: mso_film_director
Conditions:
Outputs:
Steve Saari
Steve Sacks
Steve Sale
Steve Salinaro
Steve Salisian
Steve Sanders
Steve Sanguedolce
Steve Saporito
Steve Savage
Steve Savitz
Steve Scheffler
You could add some conditions a
name
=
Steve S
Submit
Powered By Trinity Graph Engine
Contact Us

Multi-hop Relation Search

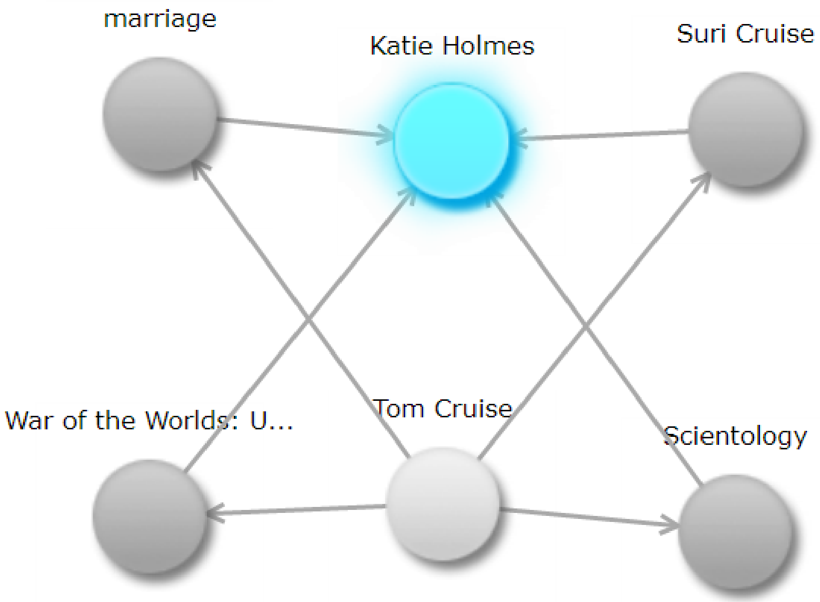
Home | Schema | API | Relation Search | People Relation Search

Satori

Add | Search

Tom Cruise, Katie Holmes

Results | View



Katie Holmes

Kate Noelle "Katie" Holmes (born December 18, 1978) is an American actress and model who first achieved fame for her role as Joey Potter on The WB television teen drama Dawson's Creek from 1998 to 2003....

Types

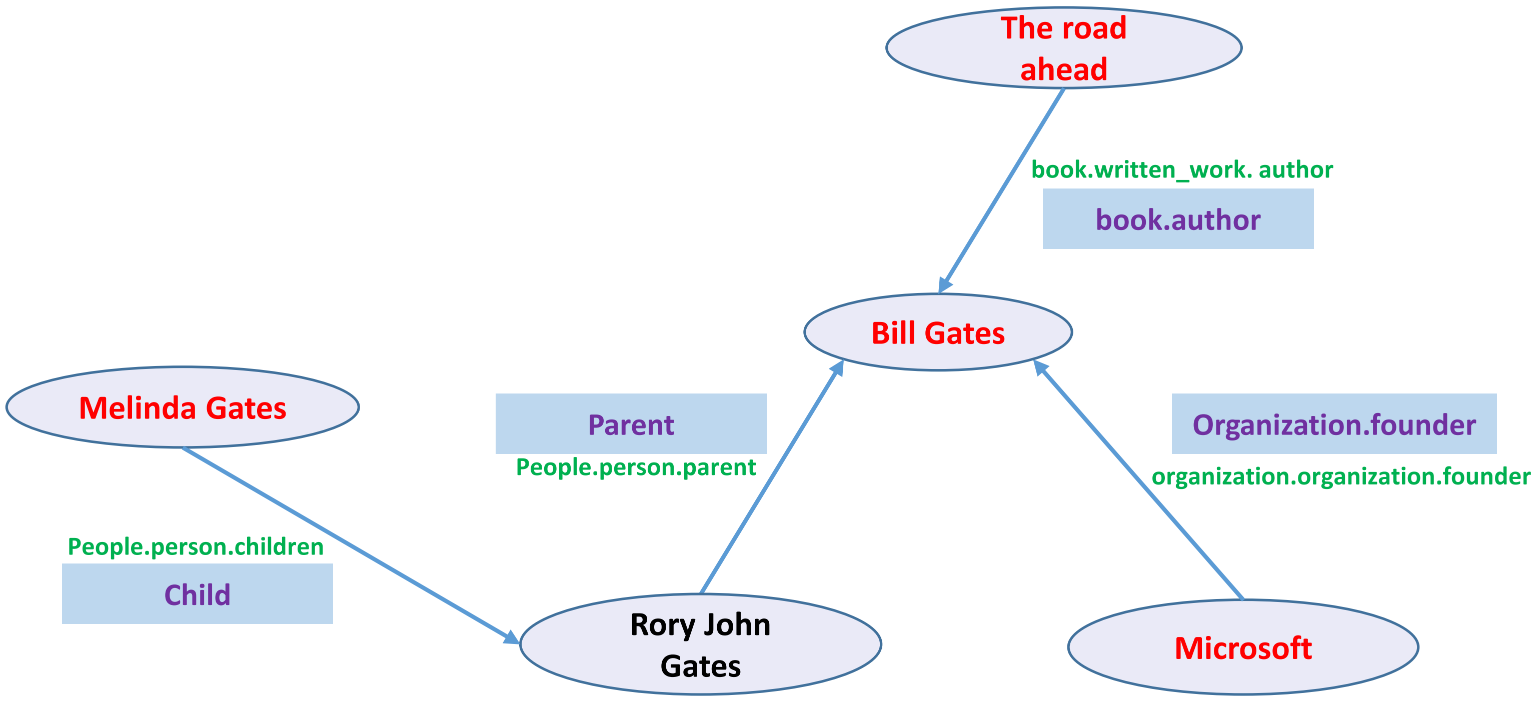
award.nominee, award.ranked_item, award.winner, film.actor, film.writer, medicine.notable_person_with_medical_condition ...

Demo: http://graph007

Powered By Trinity Graph Engine

Contact Us

Keyword Search



(Bill Gates, Melinda Gates, Microsoft, The road ahead)

Relation Search Demo

Satori | Add | Search

Tom Cruise, Mimi Rogers, Nicole Kidman, Katie Holmes

Results | View

94 Results (103 ms)

Results
o--film.actor.film-->(Eyes Wide Shut)--film.film.actor-->(Nicole Kidman)
o--film.actor.film-->(National Movie Awards)--film.film.actor-->(Katie Holmes)
o--film.actor.film-->(InStyle: Celebrity Weddings)--film.film.actor-->(Katie Holmes)
o--people.person.marriage-->(marriage)--time.event.person-->(Katie Holmes)
o--people.person.marriage-->(marriage)--time.event.person-->(Nicole Kidman)
o--film.actor.film-->(War of the Worlds: UK Premiere Special)--film.film.actor-->(Katie Holmes)
o--film.producer.film-->(The Others)--award.nominated_work.nomination-->(nomination)--award.nomination.nominee--(Nicole Kidman)
o--people.person.children-->(Connor Cruise)--people.person.siblings-->(Isabella Jane Cruise)--people.person.parent--(Nicole Kidman)
o--film.producer.film-->(The Others)--award.nominated_work.nomination-->(nomination)--award.nomination.nominee--(Nicole Kidman)
o--film.actor.performance-->(performance)--film.performance.film-->(Eyes Wide Shut)--film.film.actor--(Nicole Kidman)

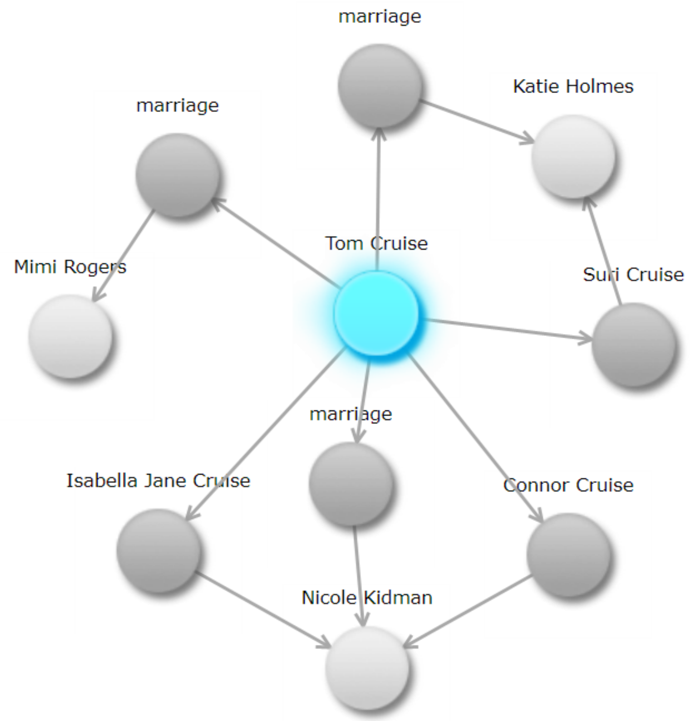
Prev Page   Next Page

Relation Search Demo

Satori | Add | Search

Tom Cruise, Mimi Rogers, Nicole Kidman, Katie Holmes

Results View



Tom Cruise

Tom Cruise (born Thomas Cruise Mapother IV; July 3, 1962), is an American film actor and producer. He has been nominated for three Academy Awards and has won three Golden Globe Awards. He started his career at age 19 in the 1981 film Endless Love. After portraying supporting roles in Taps (1981) and The Outsiders (1983), his first leading role was in Risky Business, released in August 1983. Cruise became a full-fledged movie...

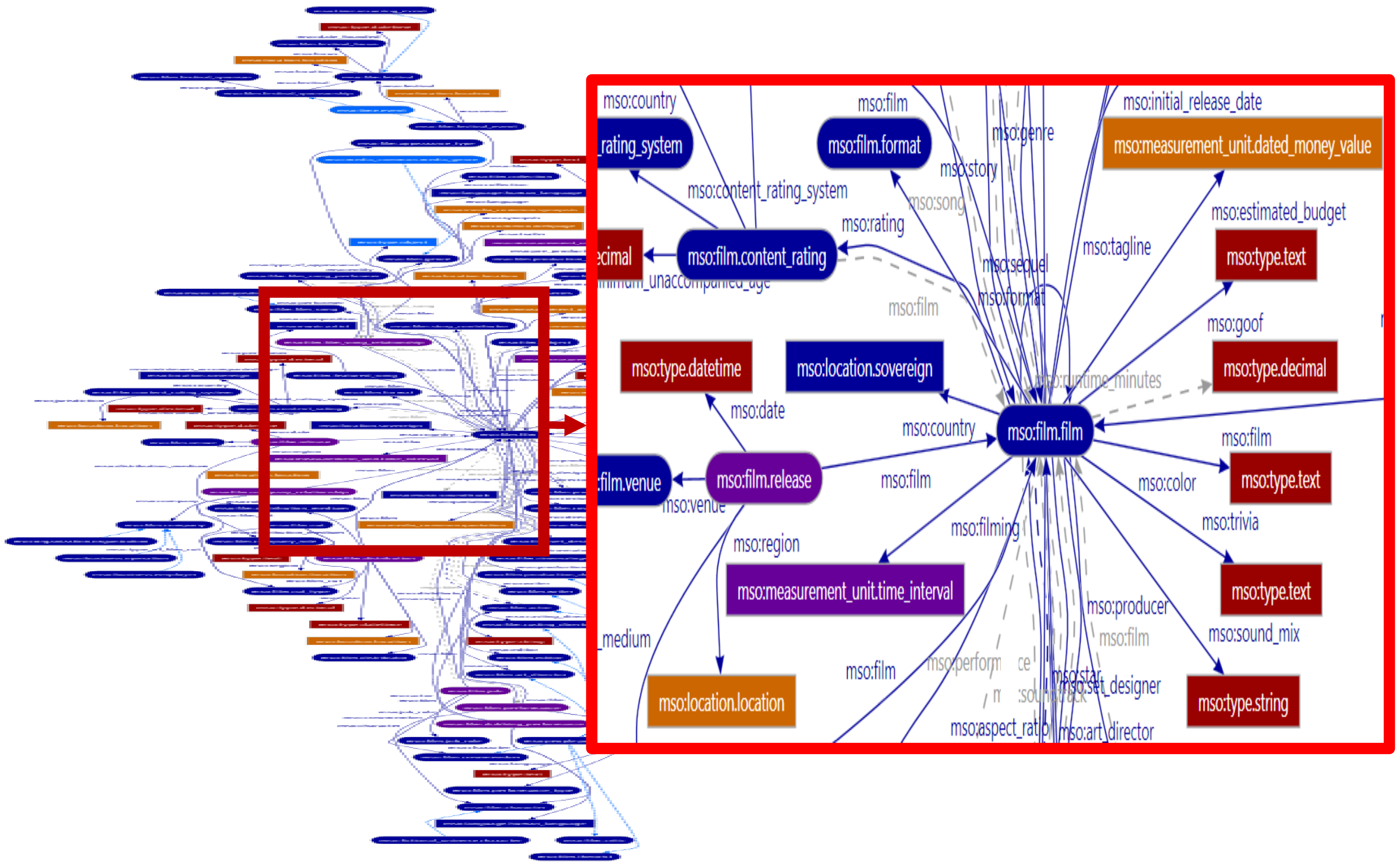
Types

award.nominee, award.winner, film.actor, film.director, film.producer, film.story_contributor ...

How can we make it fast enough

- Big data
  - emm, we have a large variety of tools available

- But, how do we handle “big schema” …


mso:country
rating_system
mso:film
mso:film.format
mso:genre
mso:initial_release_date
mso:measurement_unit.dated_money_value
mso:story
mso:content_rating_system
mso:song
mso:rating
mso:estimated_budget
mso:tagline
ecimal
mso:film.content_rating
mso:type.text
mso:sequel
mso:format
mso:film
mso:goof
mso:type.datetime
mso:location.sovereign
mso:runtime_minutes
mso:type.decimal
mso:date
mso:country
mso:film.film
mso:film
film.venue
mso:film.release
mso:film
mso:color
mso:type.text
mso:venue
mso:trivia
mso:region
mso:filming
mso:measurement_unit.time_interval
mso:type.text
mso:producer
mso:film
medium
mso:sound_mix
mso:film
mso:set_designer
mso:location.location
mso:type.string
mso:art_director

Big Schema

How can we make it fast enough

- Big data
  - emm, we have a large variety of tools available

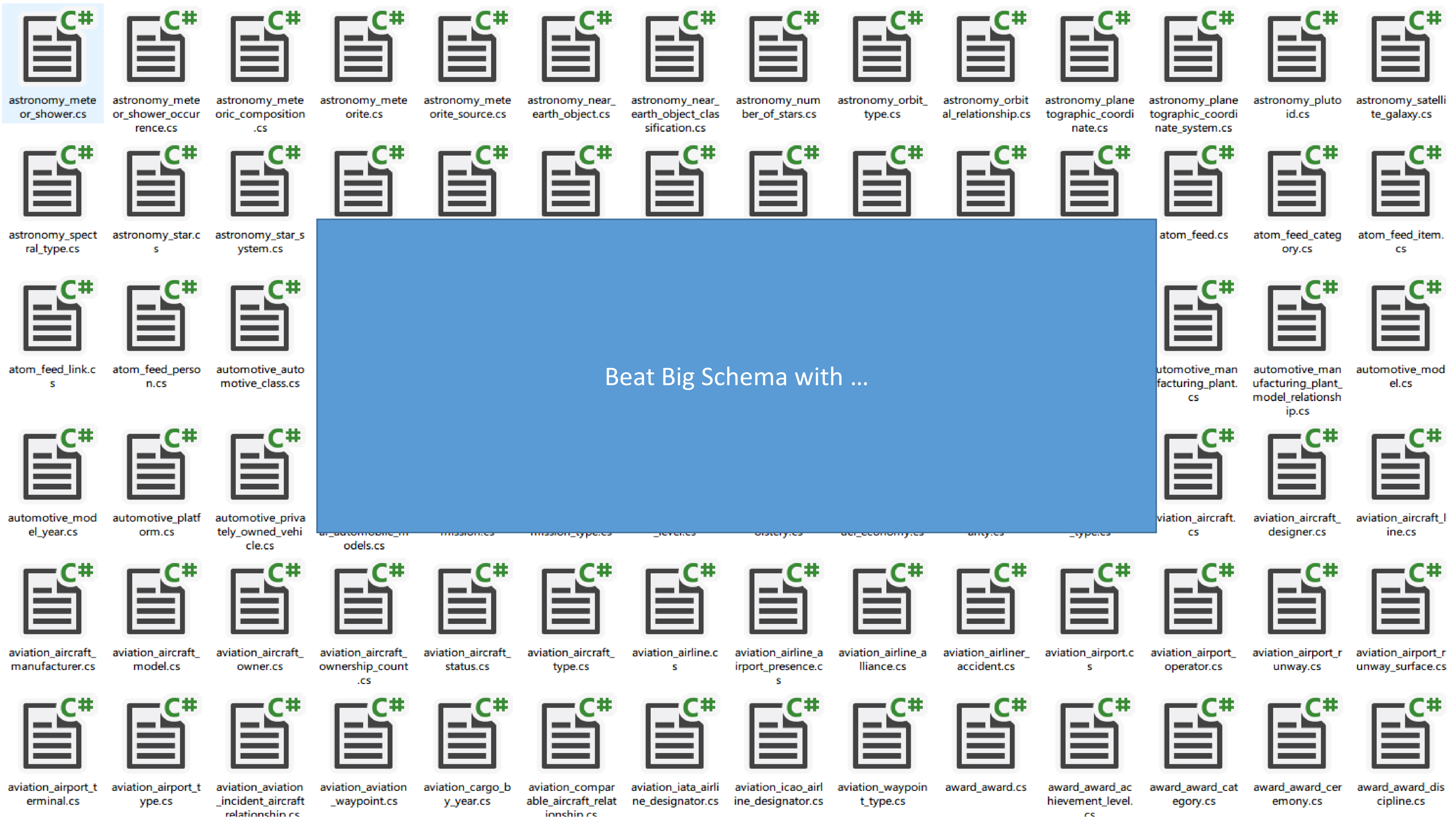
- But, how do we handle "big schema" …

  If we treat everything as texts and build indexes for these piles of words

⬇

- Inefficient data processing (weakly-typed system)
- Limited search functionality we can provide

Beat Big Schema with …


Beat Big Schema with …
astronomy_meteor_shower.cs
astronomy_meteor_shower_occurrence.cs
astronomy_meteoric_composition.cs
astronomy_meteorite.cs
astronomy_meteorite_source.cs
astronomy_near_earth_object.cs
astronomy_near_earth_object_classification.cs
astronomy_number_of_stars.cs
astronomy_orbit_type.cs
astronomy_orbital_relationship.cs
astronomy_planetographic_coordinate.cs
astronomy_planetographic_coordinate_system.cs
astronomy_plutoid.cs
astronomy_satellite_galaxy.cs
astronomy_spectral_type.cs
astronomy_star.cs
astronomy_star_system.cs
atom_feed.cs
atom_feed_category.cs
atom_feed_item.cs
atom_feed_link.cs
atom_feed_person.cs
automotive_automotive_class.cs
utomotive_manfacturing_plant.cs
automotive_manufacturing_plant_model_relationship.cs
automotive_model.cs
automotive_model_year.cs
automotive_platform.cs
automotive_privately_owned_vehicle.cs
odels.cs
viation_aircraft.cs
aviation_aircraft_designer.cs
aviation_aircraft_line.cs
aviation_aircraft_manufacturer.cs
aviation_aircraft_model.cs
aviation_aircraft_owner.cs
aviation_aircraft_ownership_count.cs
aviation_aircraft_status.cs
aviation_aircraft_type.cs
aviation_airline.cs
aviation_airline_airport_presence.cs
aviation_airline_alliance.cs
aviation_airliner_accident.cs
aviation_airport.cs
aviation_airport_operator.cs
aviation_airport_runway.cs
aviation_airport_runway_surface.cs
aviation_airport_terminal.cs
aviation_airport_type.cs
aviation_aviation_incident_aircraft_relationship.cs
aviation_aviation_waypoint.cs
aviation_cargo_by_year.cs
aviation_comparable_aircraft_relationship.cs
aviation_iata_airline_designator.cs
aviation_icao_airline_designator.cs
aviation_waypoint_type.cs
award_award.cs
award_award_achievement_level.cs
award_award_category.cs
award_award_ceremony.cs
award_award_discipline.cs

Beat Big Schema with …

Big Code!

Freebase Graph:

- Generated lines of code for Freebase: 8,868,163
- Bytes of code: 446,747,058

What is the huge amount of code for?

- Provides extremely fine-grained data access methods best matching the data

= Efficiency

Outline

- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- Offline analytics
- Online query processing
- Case study
- **Graph generation**
- Other graph representations and their applications

What is a graph generator & why do we need one

- A graph generator generates can graphs with user-specified size and properties

- We need to generate large graphs for experiments

- Large graph generation takes a long time

PGBL graph generator

http://www.boost.org/doc/libs/1_59_0/libs/graph_parallel/doc/html/rmat_generator.html

When the existing ones cannot meet our needs, we may want to write one.

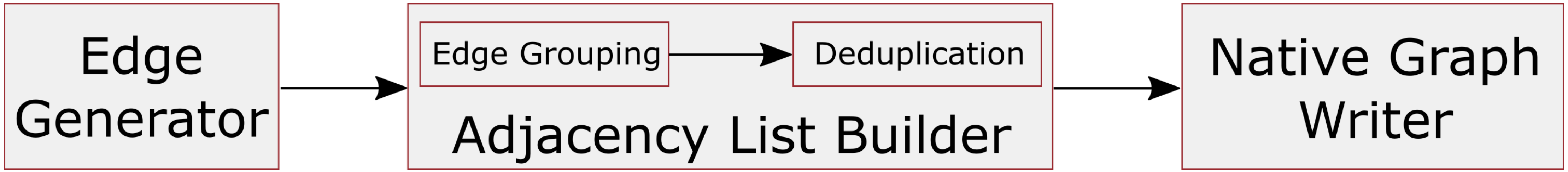
What is a good graph generator

- It can generate a graph with certain properties
- It can generate a large graph fast
- It is as resource economical as possible
- It can generate graphs in native graph formats

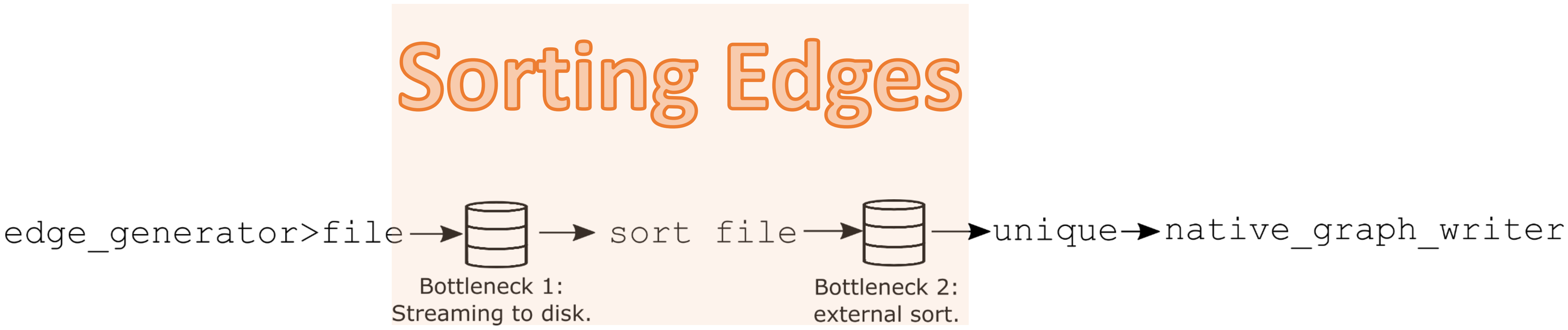
Graph representation: adjacency list vs. matrix

- In most cases, adjacency list is preferable to the matrix representation
    - Matrix does not support dynamic node insertion and deletion
    - The space overhead is high when we are generating a sparse graph

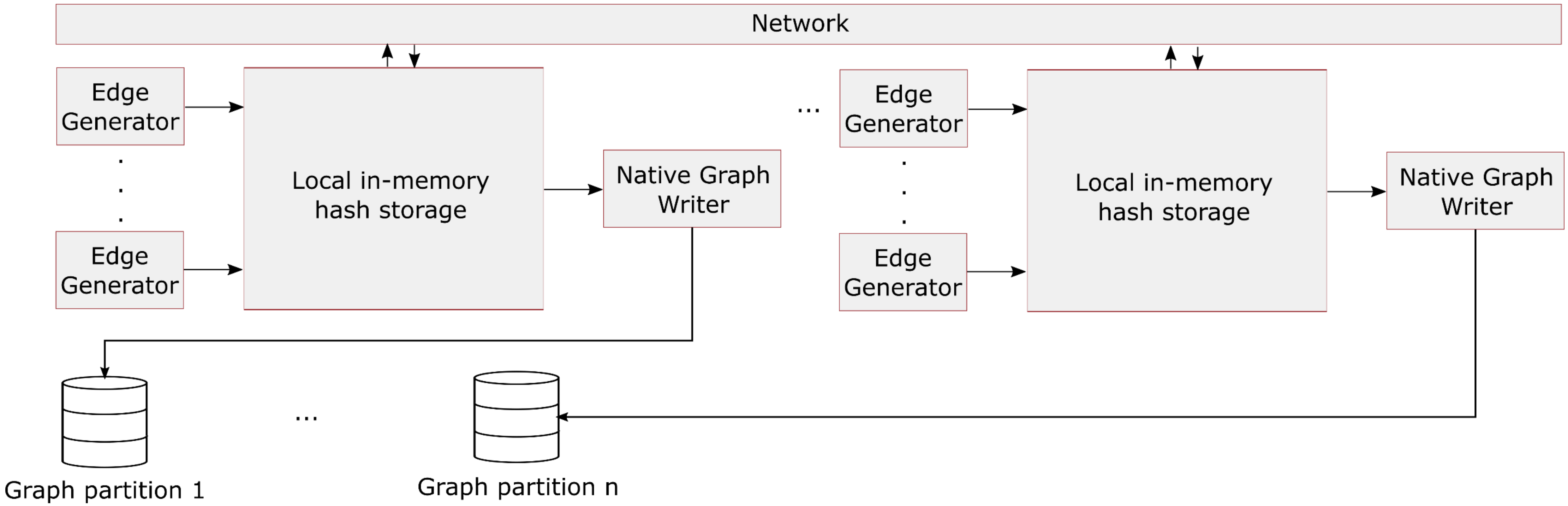
A common graph generation pipeline



Bottlenecks of the pipeline

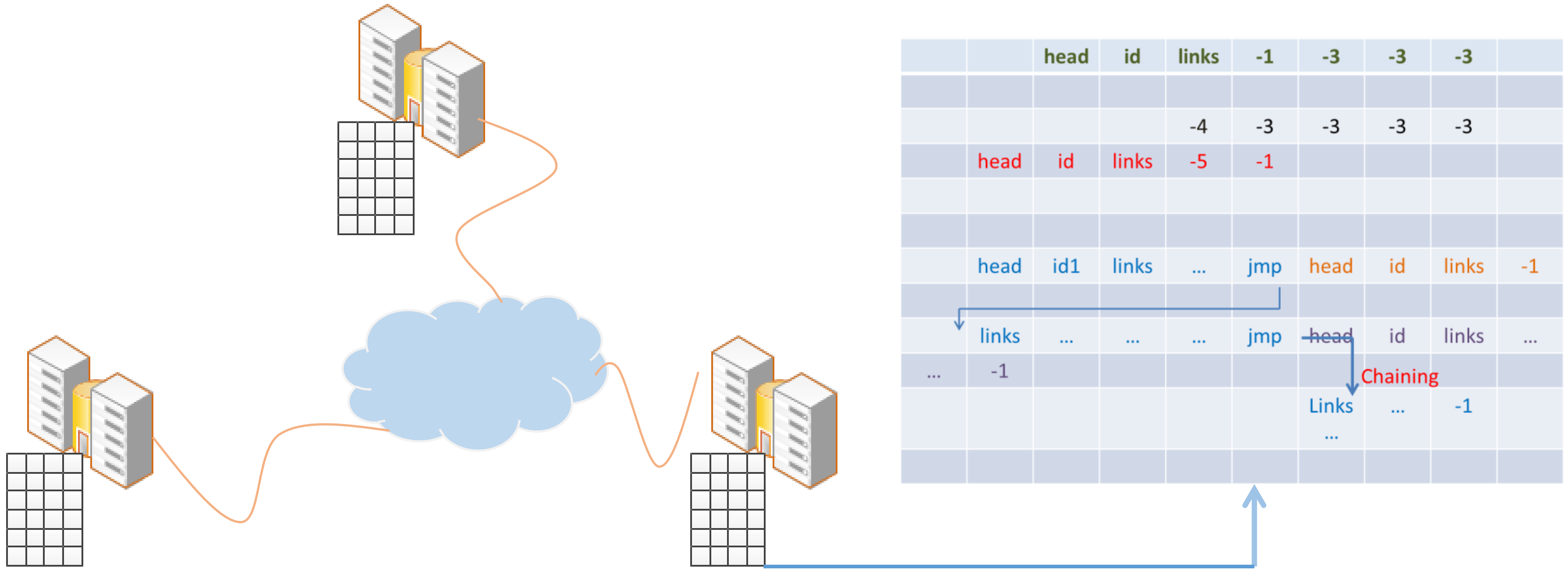


Avoid sorting by using an in-memory hash storage



Distributed graph generation

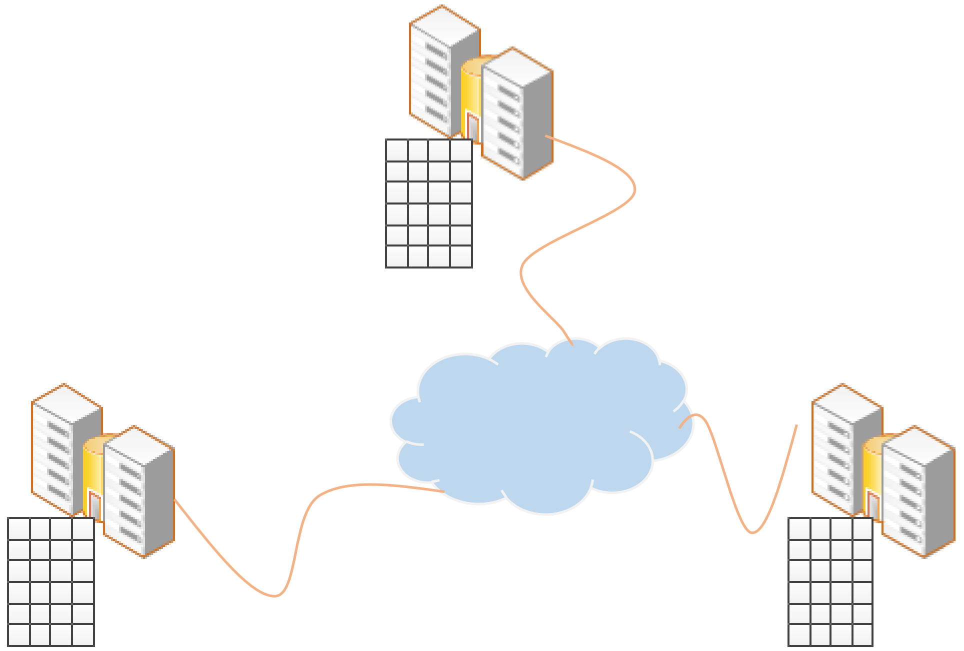
- Step 1: Preparing a distributed hash storage



In-memory Hash Storage

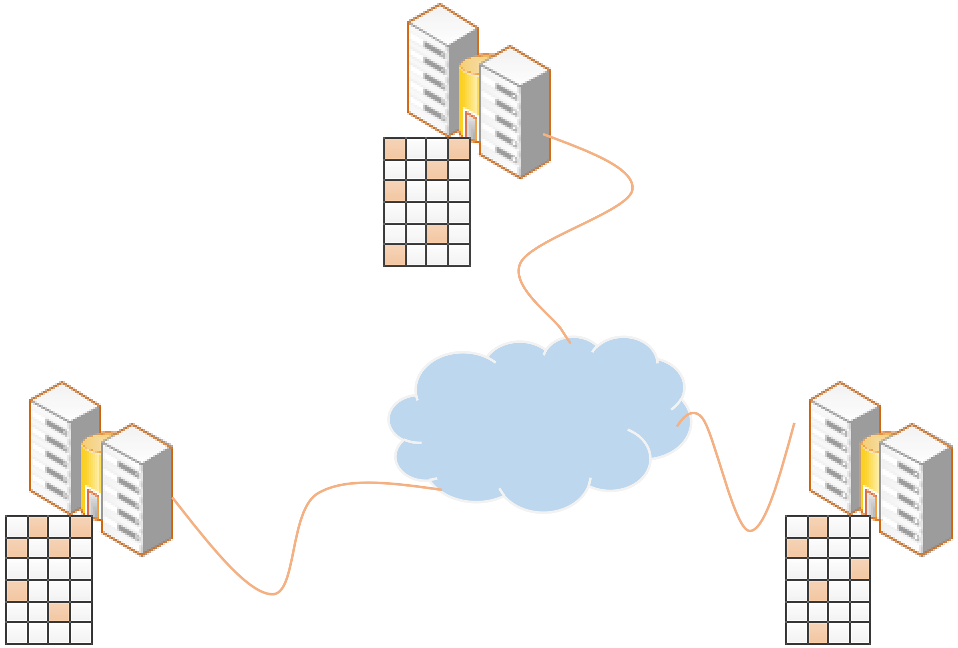
Distributed graph generation

- Step 2: Generating the graph in parallel



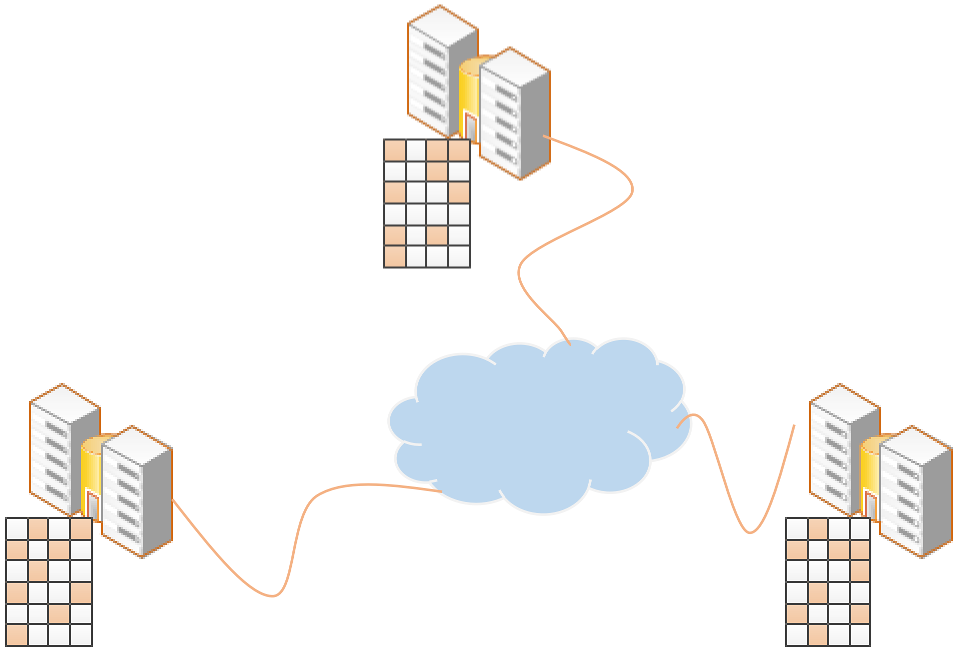
Distributed graph generation

- Step 2: Generating the graph in parallel



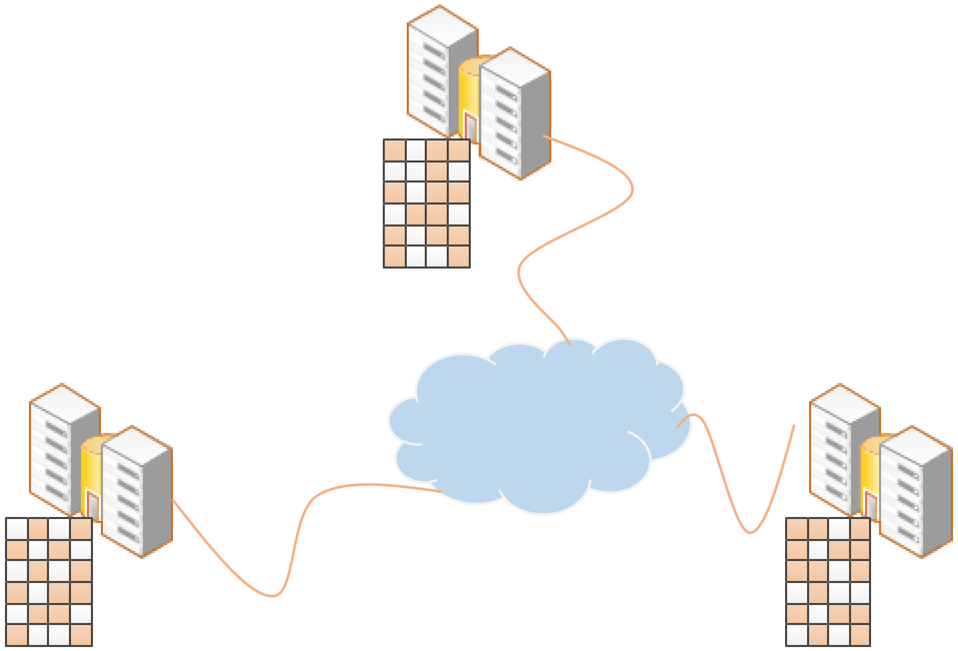
Distributed graph generation

- Step 2: Generating the graph in parallel



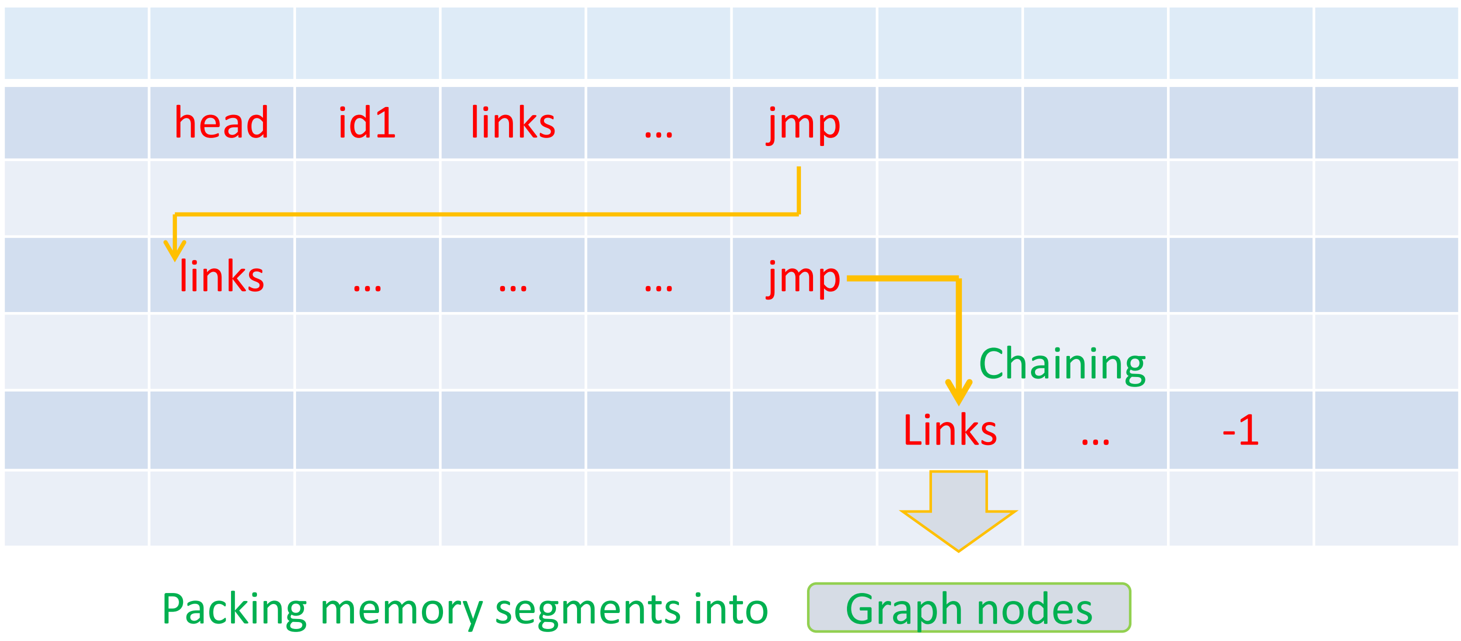
Distributed graph generation

- Step 2: Generating the graph in parallel



Distributed graph generation

- Step 3: Write the generated graph to disk



Hash Storage

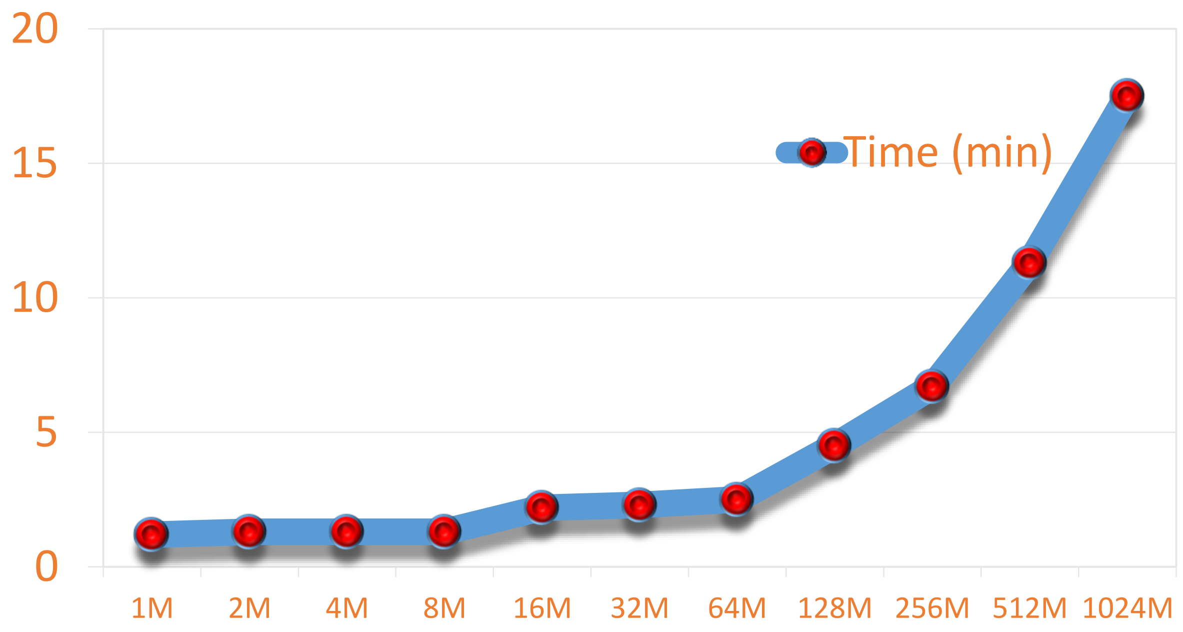
		head	id	links	-1				
	head	id	links	…	-1				
	head	id1	links	…	jmp	head	id	links	-1
	links	…	…	…	jmp	head	id	links	…
…	-1					Chaining			
						Links…	…	-1	

or

In-memory key-value store

An example

http://www.graphengine.io/docs/manual/DemoApps/GraphGenerator.html



Outline

- Graph processing scenarios
- Challenges of large graph processing
- General design principles
- Offline analytics
- Online query processing
- Case study
- Graph generation
- **Other graph representations and their applications**

Matrix arithmetic

Representative System: Pegasus

- Open source large graph mining system
  - Implemented on Hadoop

- Convert graph mining operations into iterative matrix-vector multiplication

- Pegasus uses a n by n matrix M and a vector v of size n to represent graph data

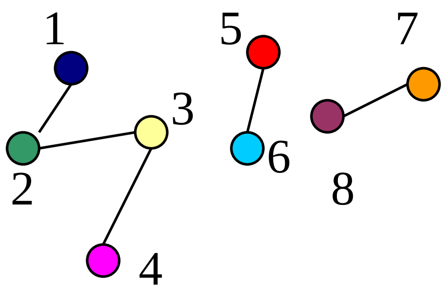
Generalized Iterated Matrix-Vector Multiplication

$$M \times v = v' \text{, where } v'_i \boxed{=} \boxed{\sum_{j=1}^{n}} \boxed{m_{i,j} \times v_j}$$

- Three primitive graph mining operations
  - $combine2(m_{i,j}, v_j)$: multiply $m_{i,j}$ and v_j
  - $combineAll_i(x_1, ..., x_n)$: sum n all the multiplication results from $combine2$
  - $assign(v_i, v_{new})$: decide how to update v_i with v_{new}

- Graph mining problems are solved by customizing the three operations

Example: Connected Components

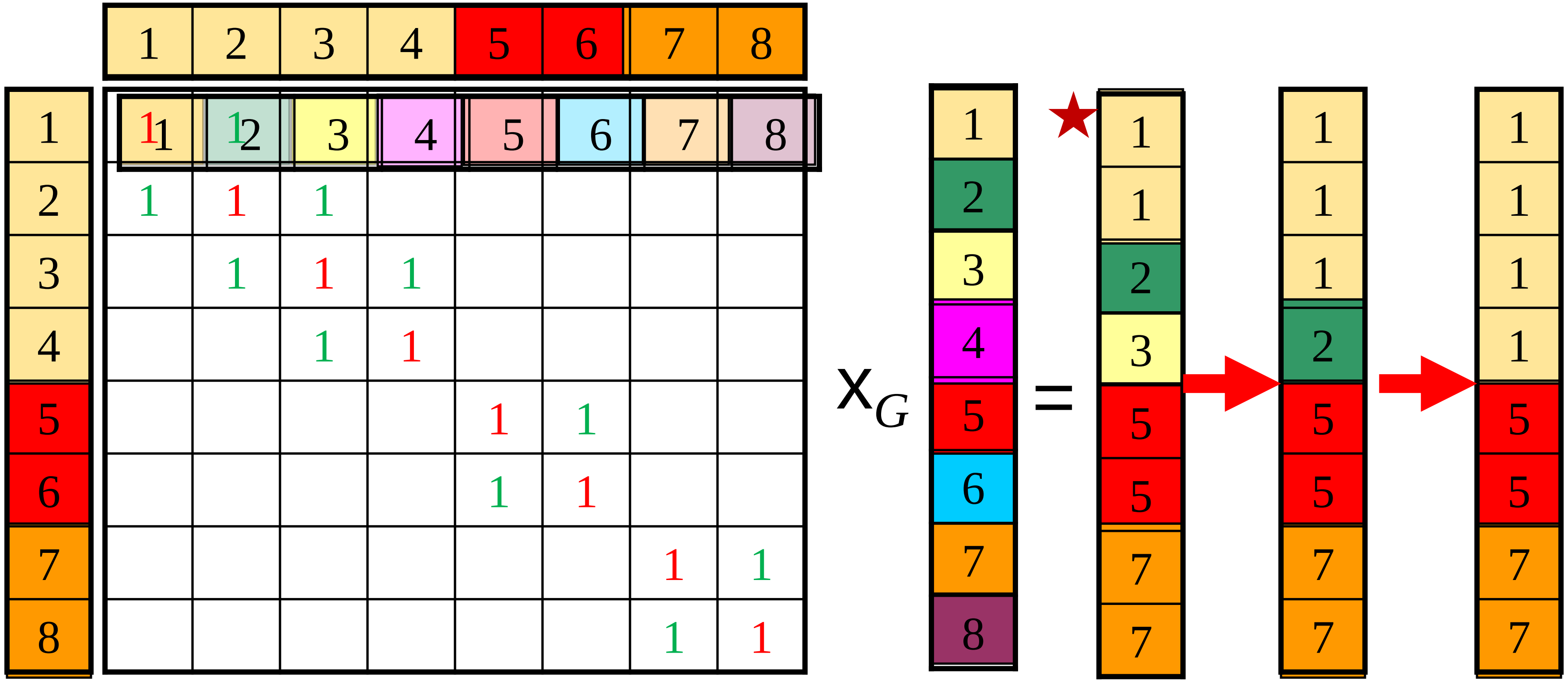


G1 G5 G7

	1	2	3	4	5	6	7	8
1		1						
2	1		1					
3		1		1				
4			1					
5						1		
6					1			
7								1
8							1	

Source: Pegasus, Kevin Andryc, 2011

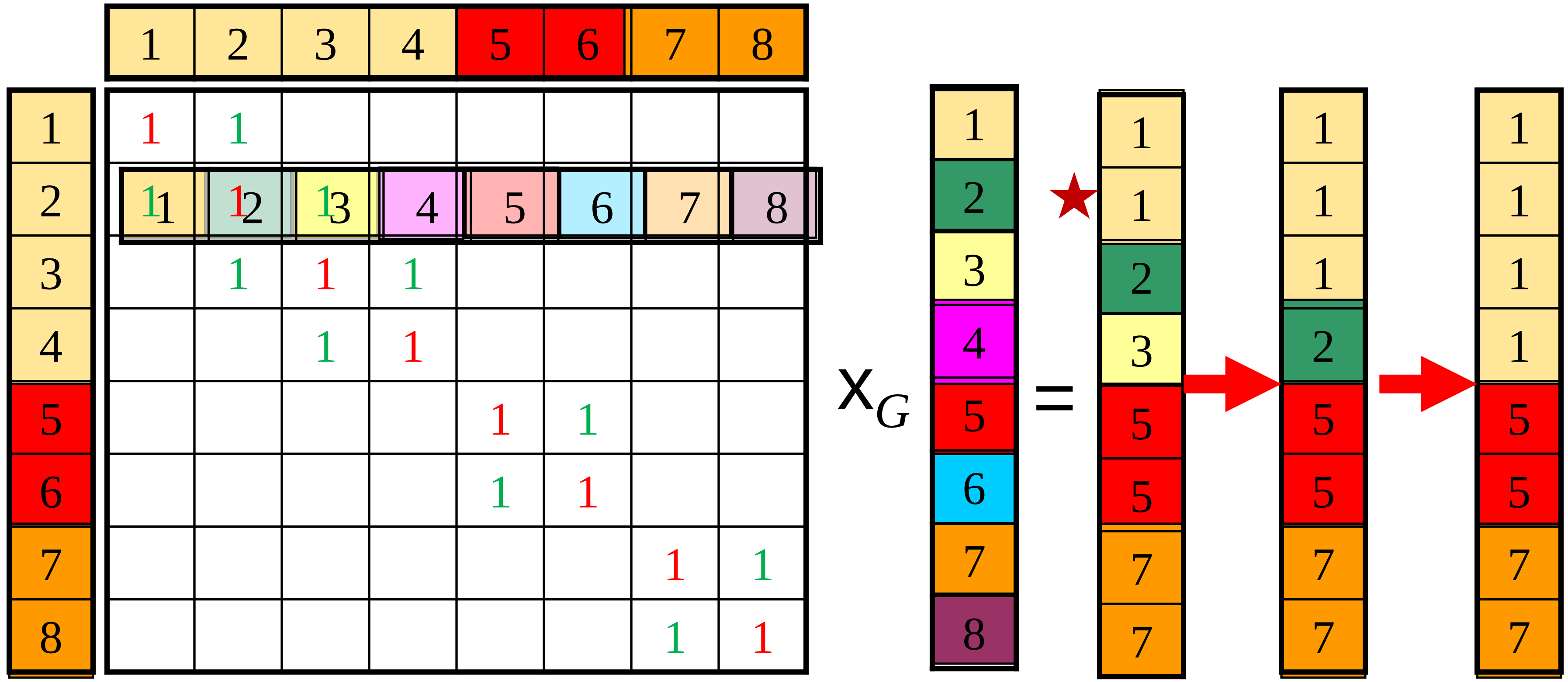
Example: Connected Components



$$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$$
$$\texttt{combineAll}_i(x_1, ..., x_n) = \mathrm{MIN}\{x_j \mid j = 1..n\}$$
$$\texttt{assign}(v_i, v_{new}) = \mathrm{MIN}(v_i, v_{new}).$$

Adapted from: Pegasus, Kevin Andryc, 2011

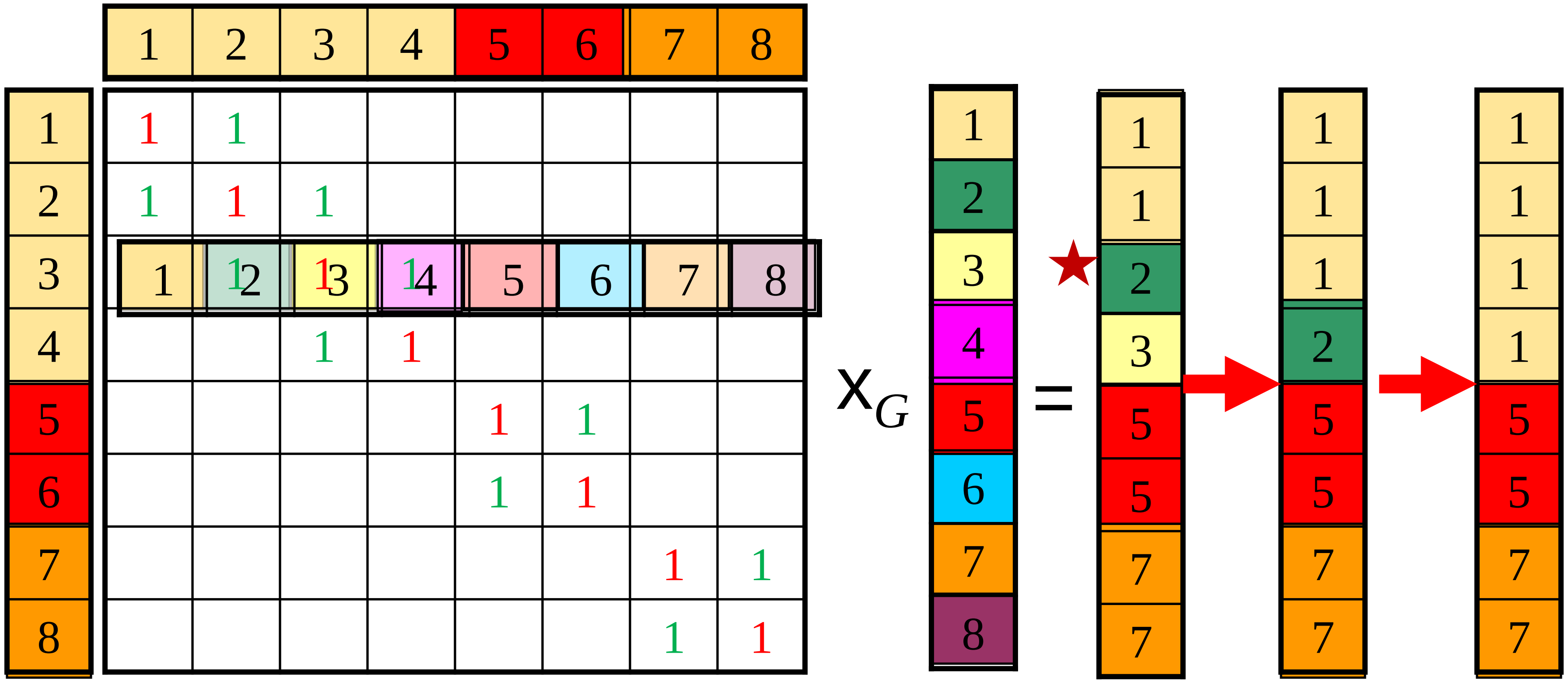
Example: Connected Components



$$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$$
$$\texttt{combineAll}_i(x_1, ..., x_n) = \text{MIN}\{x_j \mid j = 1..n\}$$
$$\texttt{assign}(v_i, v_{new}) = \text{MIN}(v_i, v_{new}).$$

Adapted from: Pegasus, Kevin Andryc, 2011

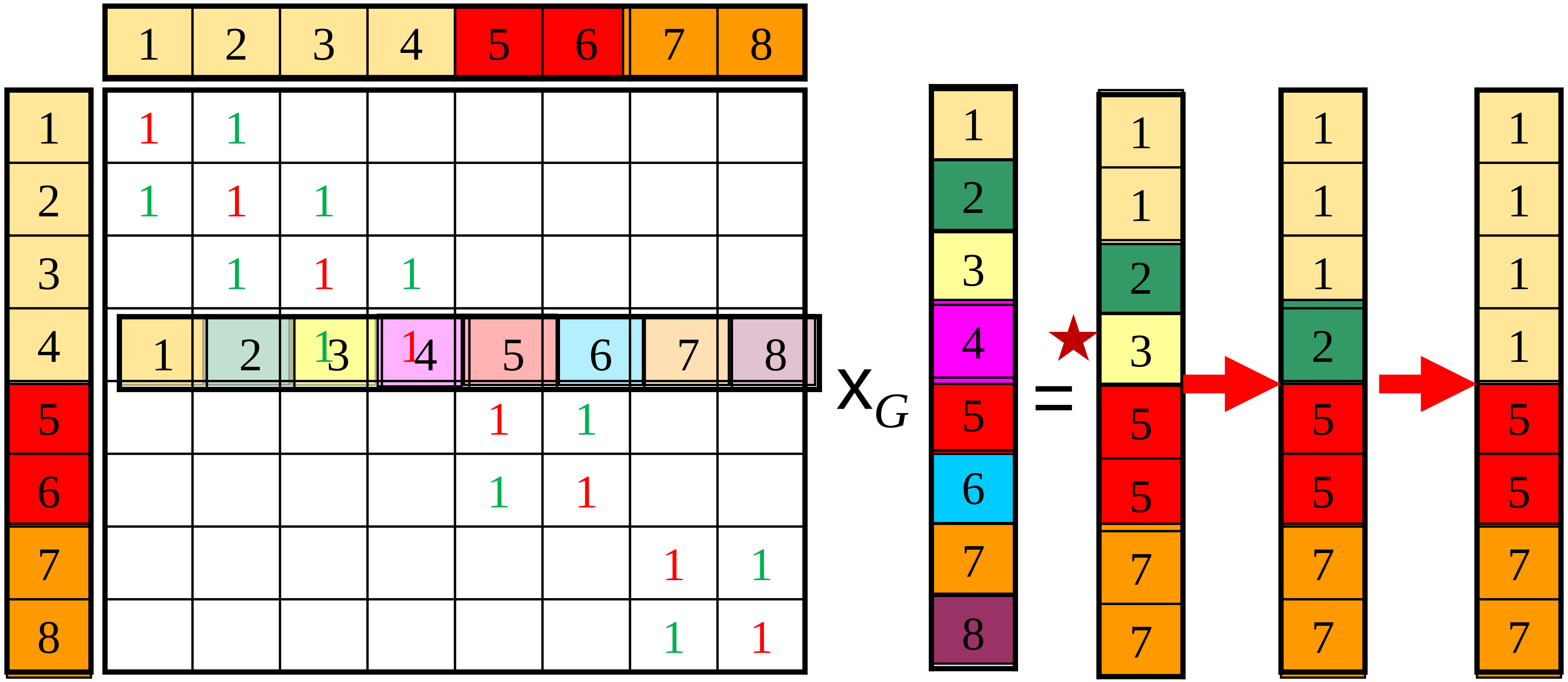
Example: Connected Components



$$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$$
$$\texttt{combineAll}_i(x_1, ..., x_n) = \text{MIN}\{x_j \mid j = 1..n\}$$
$$\texttt{assign}(v_i, v_{new}) = \text{MIN}(v_i, v_{new}).$$

Adapted from: Pegasus, Kevin Andryc, 2011

Example: Connected Components



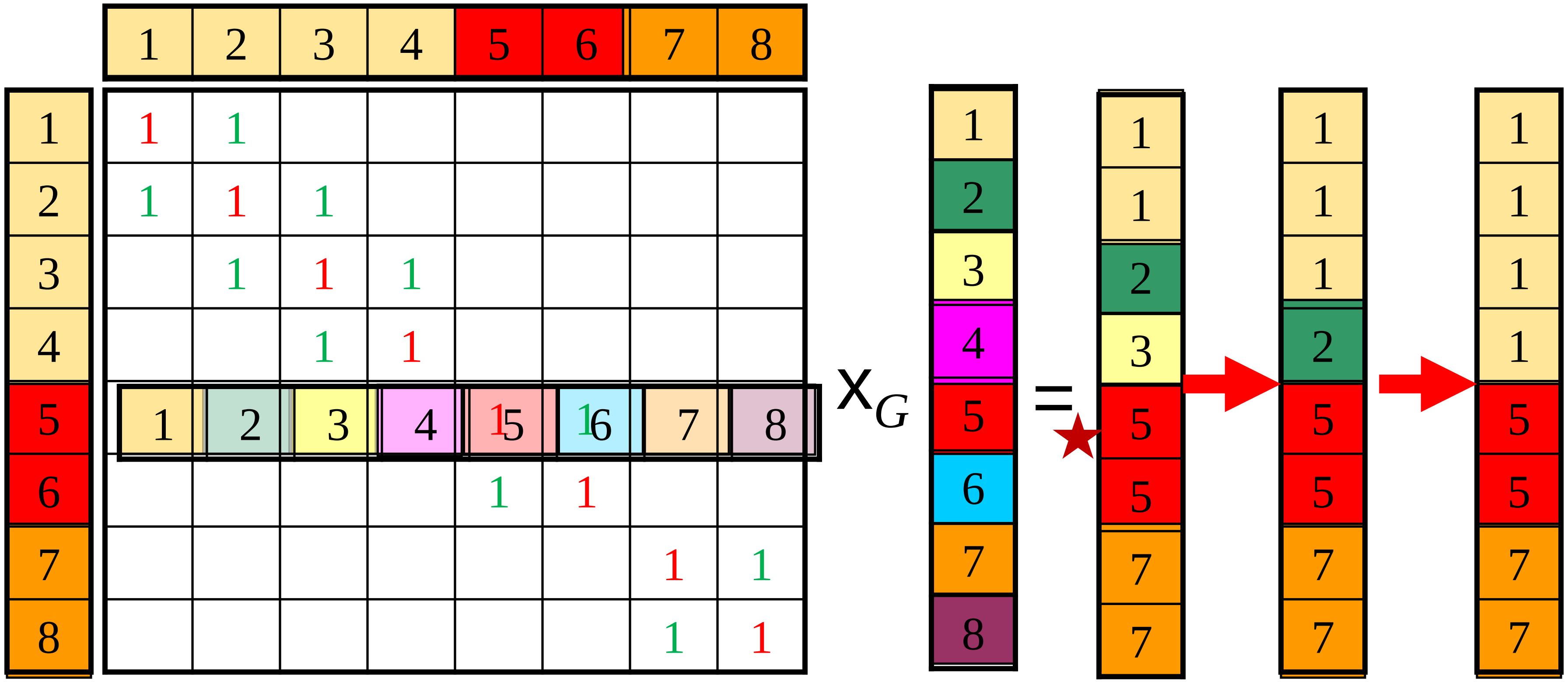
$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$

$\texttt{combineAll}_i(x_1, ..., x_n) = \text{MIN}\{x_j \mid j = 1..n\}$

$\texttt{assign}(v_i, v_{new}) = \text{MIN}(v_i, v_{new}).$

Adapted from: Pegasus, Kevin Andryc, 2011

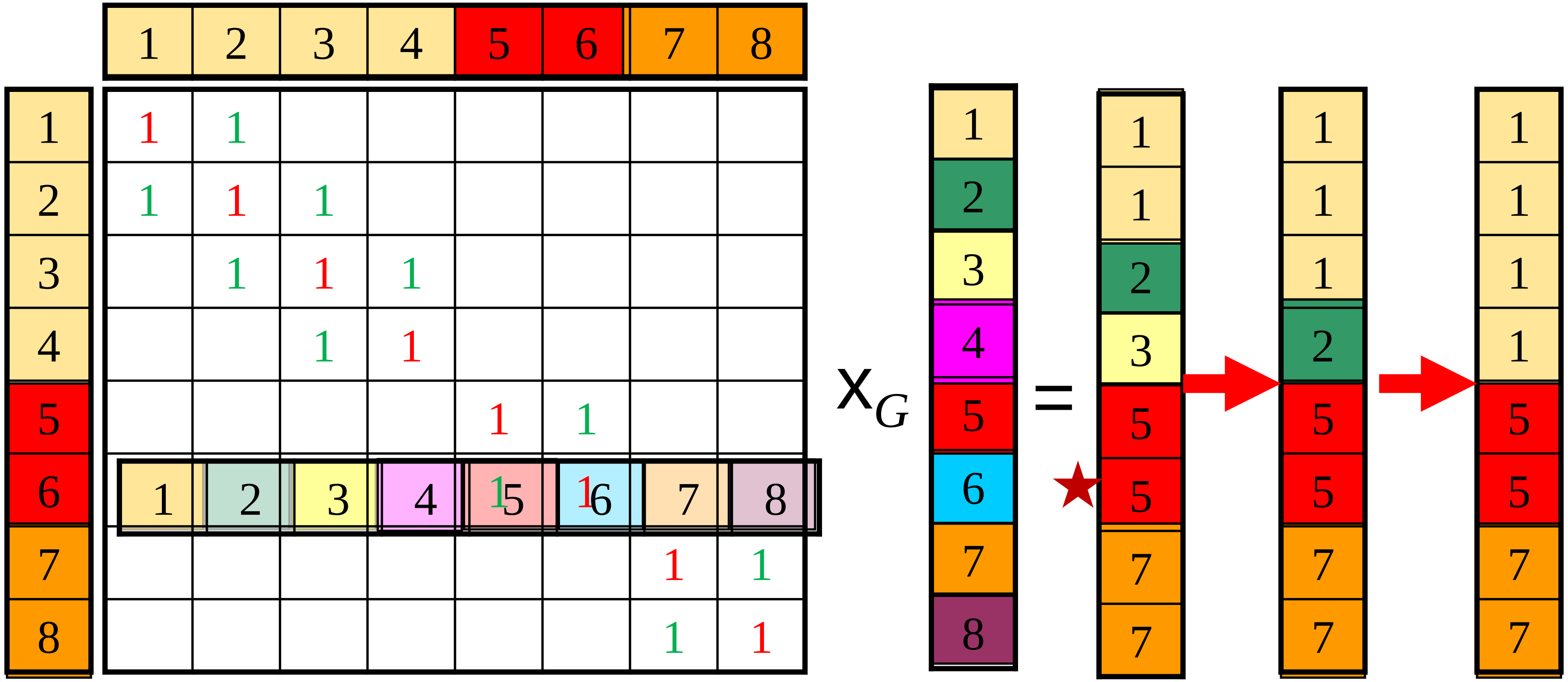
Example: Connected Components



$$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$$
$$\texttt{combineAll}_i(x_1, ..., x_n) = \text{MIN}\{x_j \mid j = 1..n\}$$
$$\texttt{assign}(v_i, v_{new}) = \text{MIN}(v_i, v_{new}).$$

Adapted from: Pegasus, Kevin Andryc, 2011

Example: Connected Components



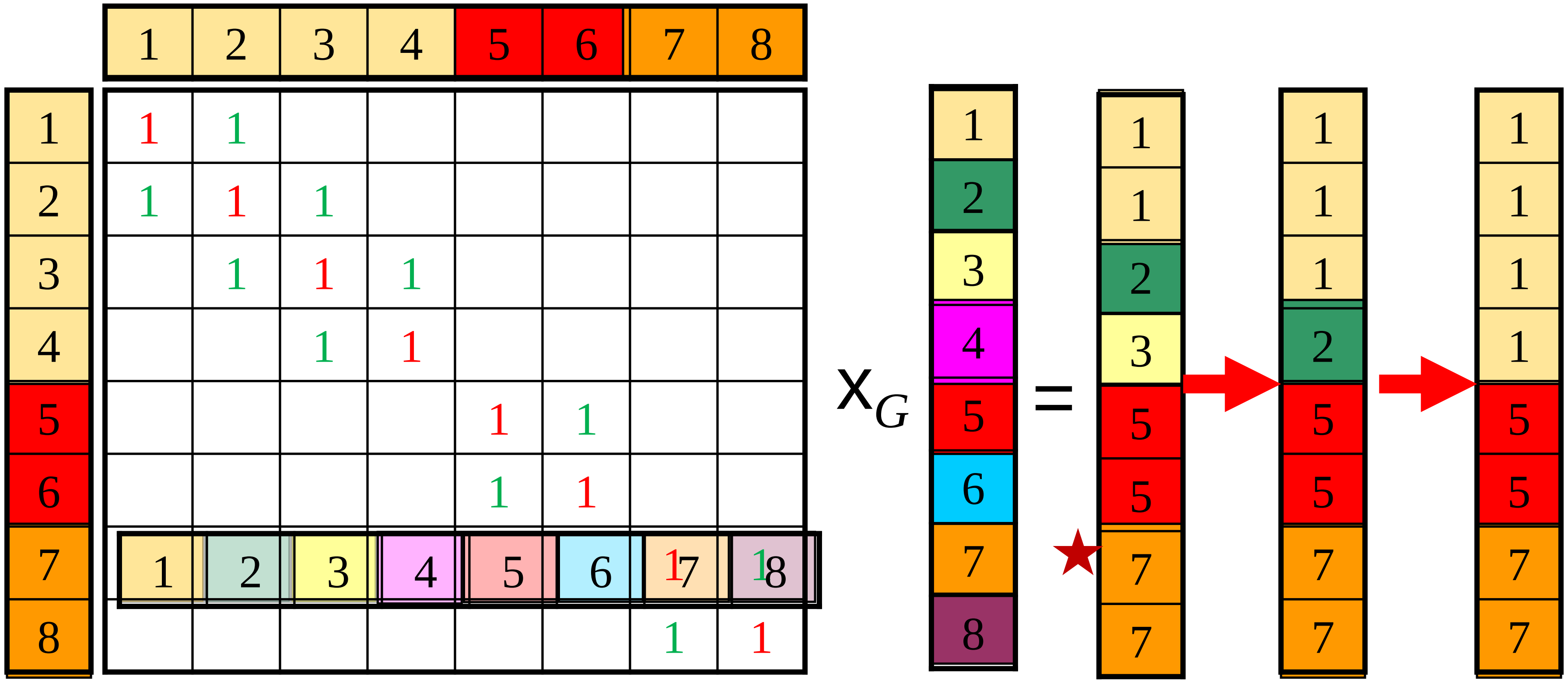
$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$

$\texttt{combineAll}_i(x_1, ..., x_n) = \text{MIN}\{x_j \mid j = 1..n\}$

$\texttt{assign}(v_i, v_{new}) = \text{MIN}(v_i, v_{new}).$

Adapted from: Pegasus, Kevin Andryc, 2011

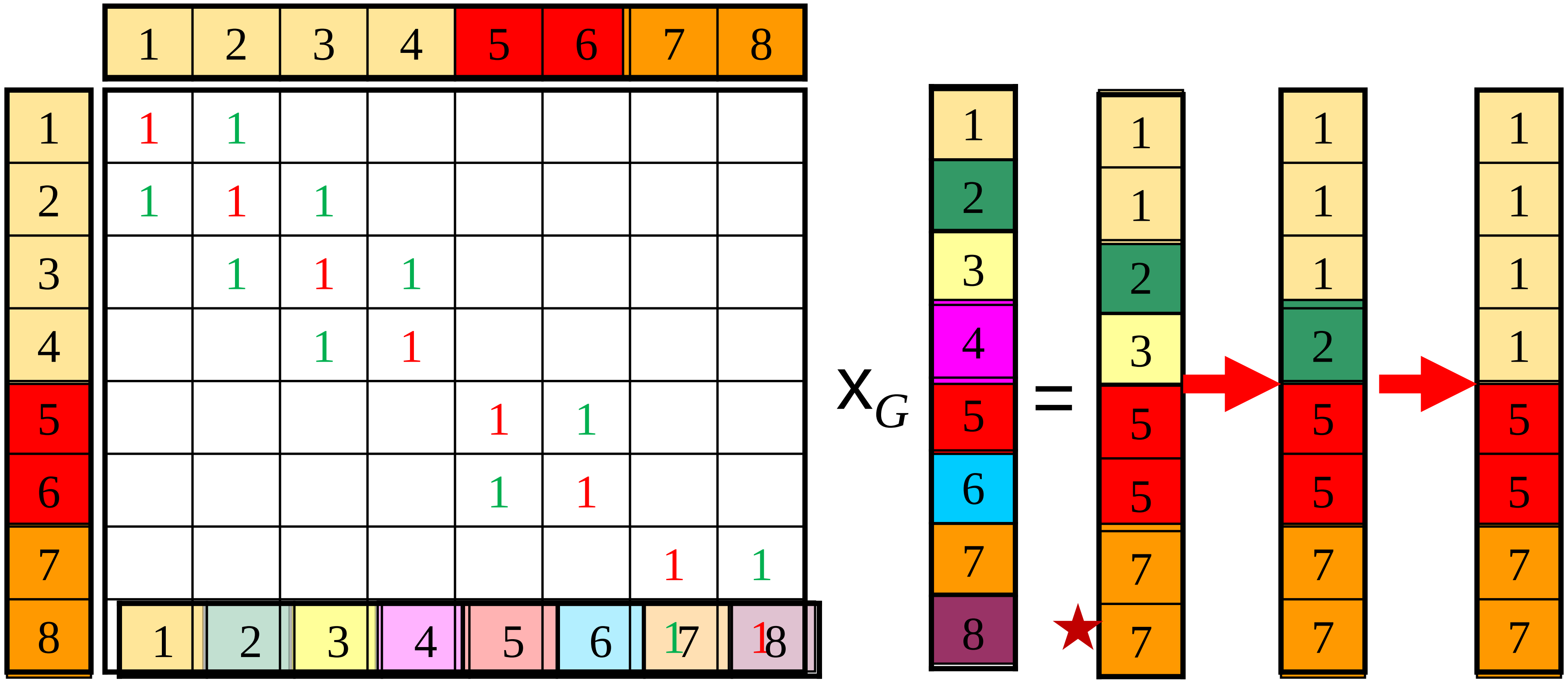
Example: Connected Components



$$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$$
$$\texttt{combineAll}_i(x_1, ..., x_n) = \mathrm{MIN}\{x_j \mid j = 1..n\}$$
$$\texttt{assign}(v_i, v_{new}) = \mathrm{MIN}(v_i, v_{new}).$$

Adapted from: Pegasus, Kevin Andryc, 2011

Example: Connected Components



$\texttt{combine2}(m_{i,j}, v_j) = m_{i,j} \times v_j.$

$\texttt{combineAll}_i(x_1, ..., x_n) = \text{MIN}\{x_j \mid j = 1..n\}$

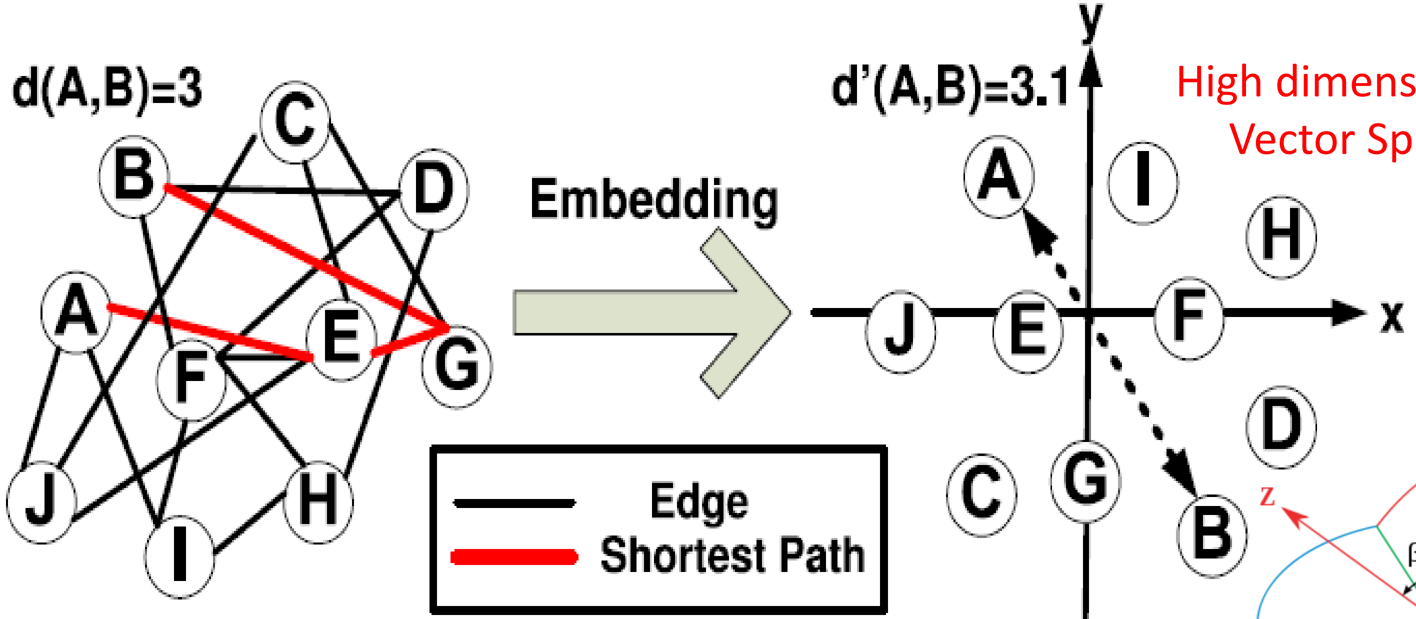
$\texttt{assign}(v_i, v_{new}) = \text{MIN}(v_i, v_{new}).$

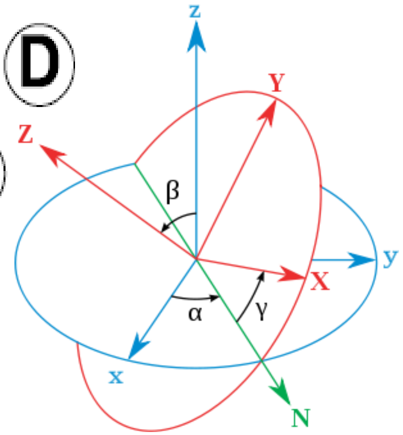
Adapted from: Pegasus, Kevin Andryc, 2011

Graph embedding

Graph Embedding

- Embed a graph into a geometric space so that distances in the space preserve the shortest distances in the graph





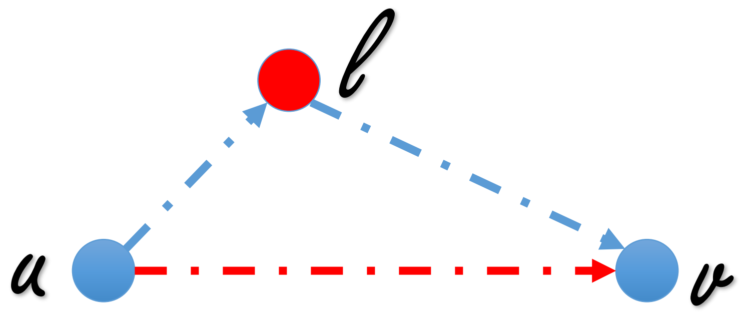
Reference: [zhao 2010, zhao2011]

Application: Distance Oracle

- Choose a small number of landmarks (~100)
  - Heuristics: Degree , betweenness, …
- Calculate the distance from each landmark to all other vertices using *BFS starting from each landmark*
- Calculate the embedding of landmarks using the *downhill simplex method* according to the distances between landmarks
- Calculate the embedding of other vertices using the *downhill simplex method* according to the distances from these vertices to landmarks

Distance oracle in a nutshell

- Step 1: Using sketch to give the lower and upper bound of the shortest distance between two vertices



$$|d(u,l) - d(l,v)| \leq d(u,v) \leq d(u,l) + d(l,v)$$

Triangle Inequality

$$l(u,v) \leq d(u,v) \leq r(u,v)$$

Distance oracle in a nutshell

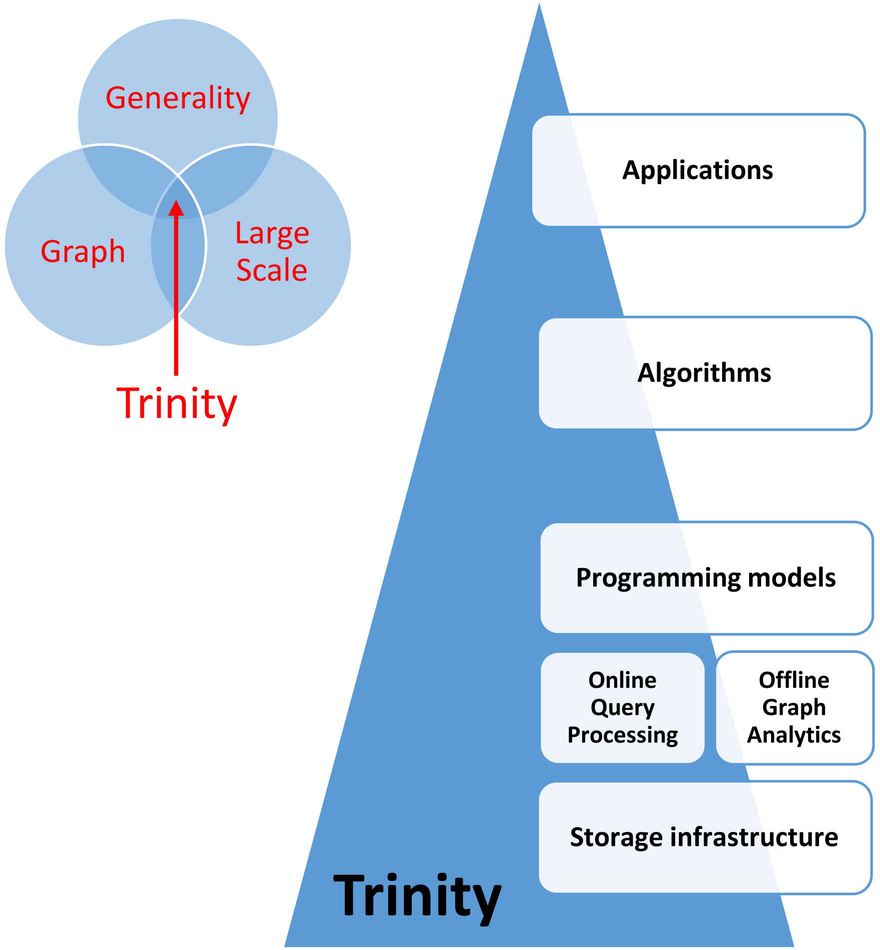
- Step 2: Refining results using graph embedding

$$d(u,v) = \begin{cases} \bar{d}(u,v) & \text{if } l(u,v) \leq \bar{d}_{u,v} \leq r(u,v); \\ l(u,v) & \text{if } \bar{d}_{u,v} < l(u,v); \\ r(u,v) & \text{if } \bar{d}_{u,v} > r(u,v); \end{cases}$$

$\bar{d}(u,v)$ is the coordinate distance in the embedding space

A Brief Introduction to Trinity Graph Engine

Trinity Research Roadmap



Real-time knowledge serving on knowledge graph, academic search, etc

Subgraph matching, Trinity.RDF, distance oracle, graph partitioning, reachability …
[VLDB 2012, 2013, 2014], ICDE 2014

Trinity Graph Engine:
[Sigmod 2012, 2013]

Trinity Memory Cloud

Design Philosophy

Not a one-size-fits-all graph system, but a graph engine

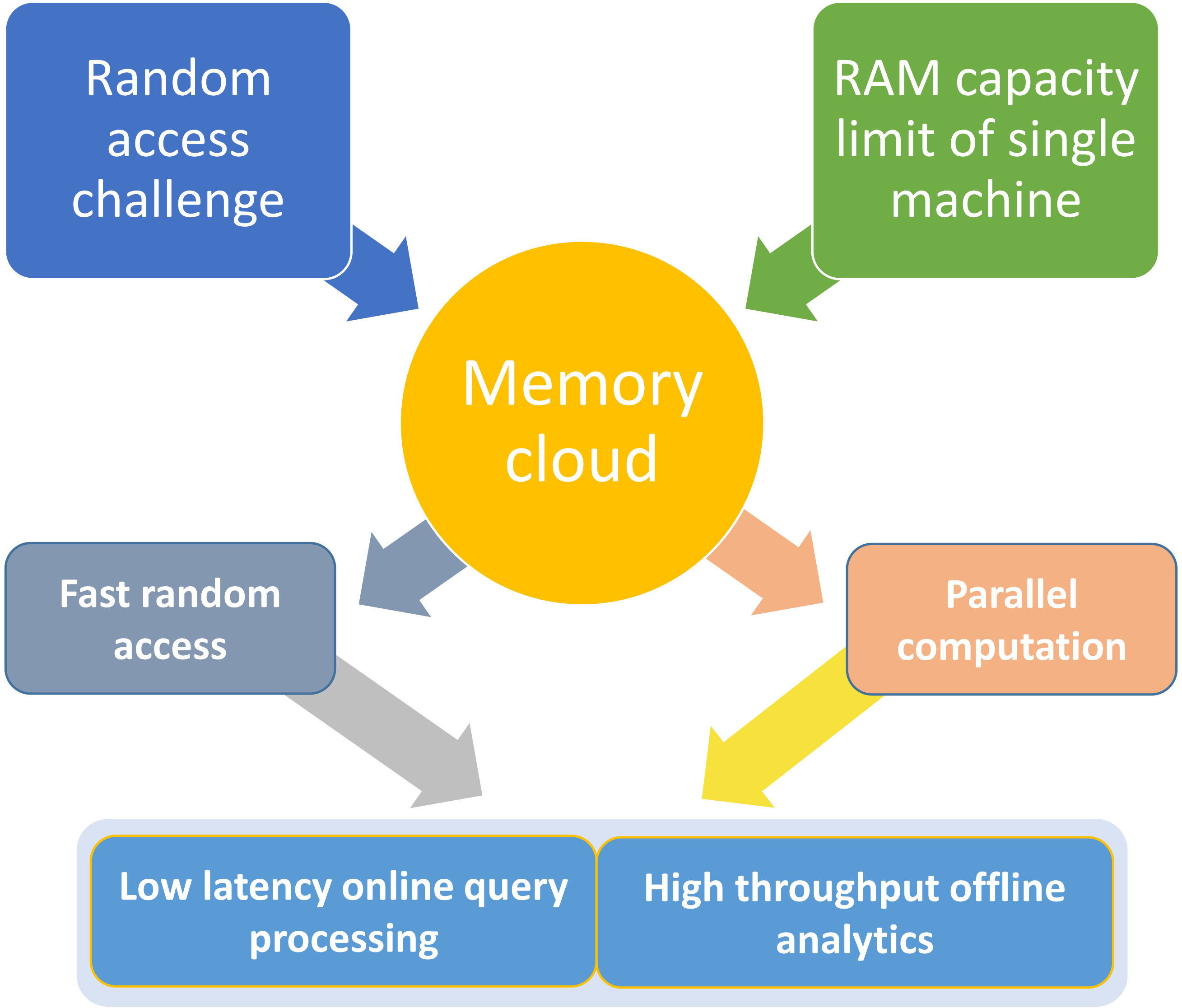
Flexible data and computation modeling capability

⬇

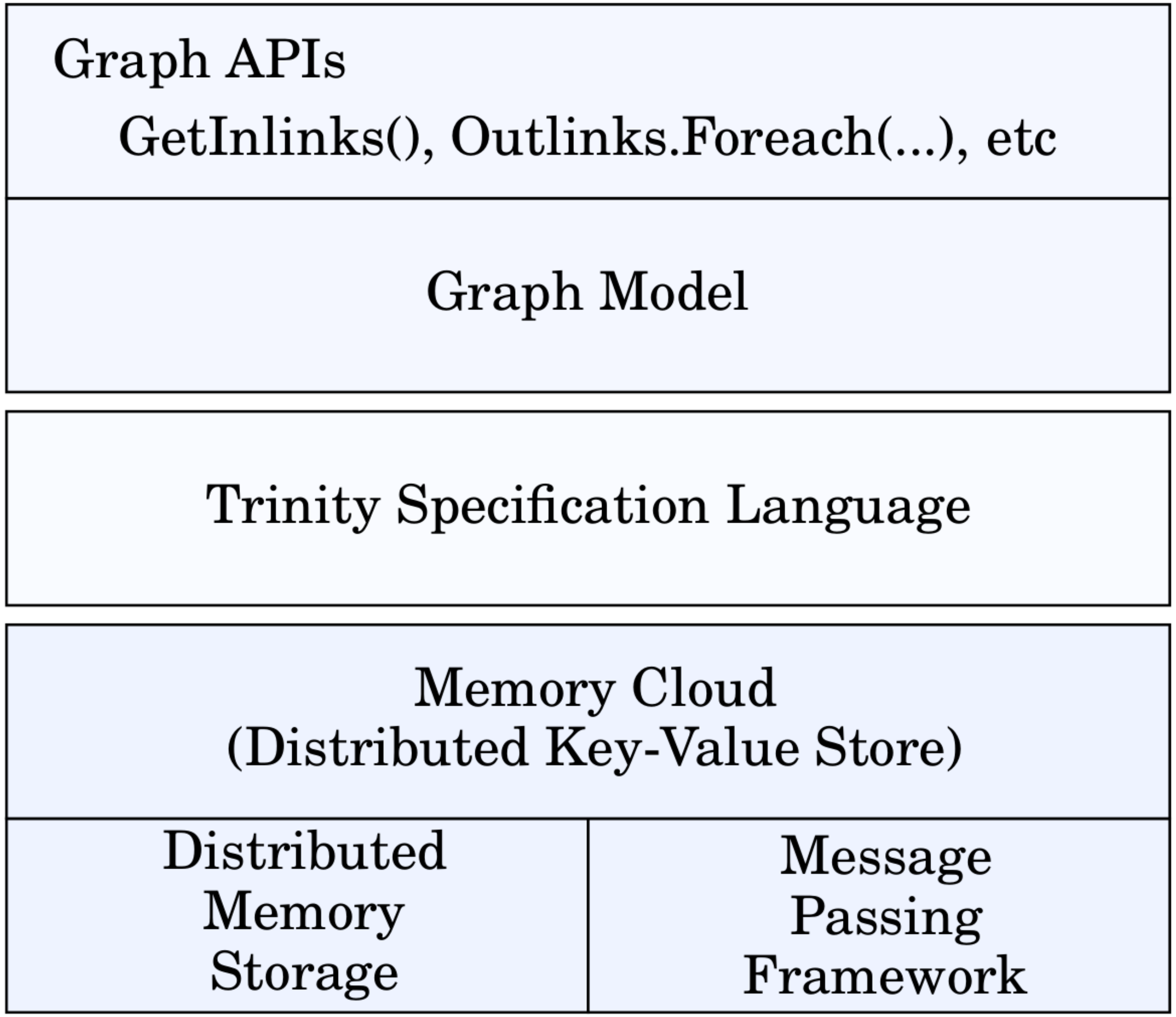
Trinity can morph into
a large variety of graph processing systems

Trinity **= Graph Modeling Tools +**
Distributed In-memory Data Store +
Declarative Programming Model

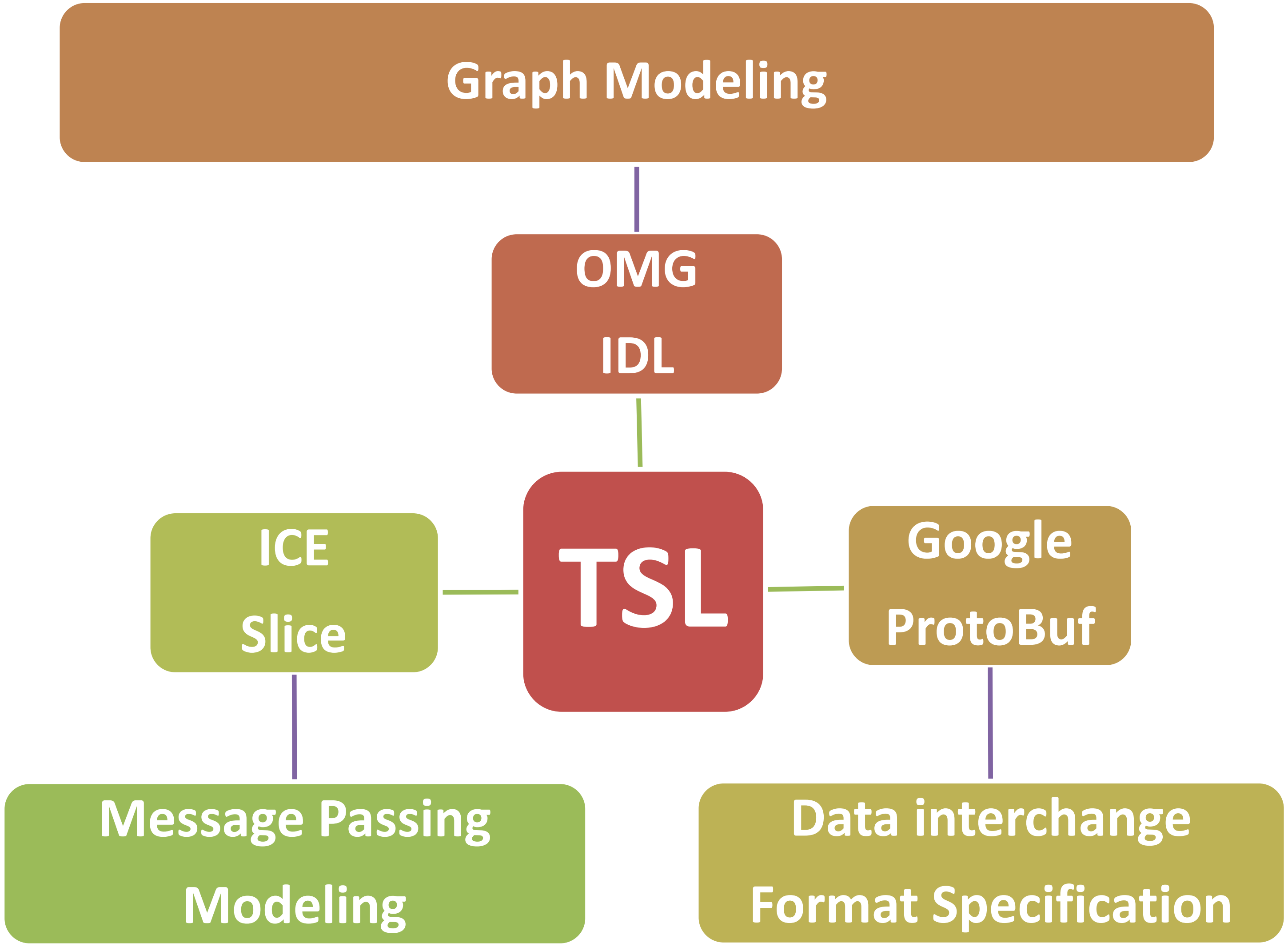
Design Rationale of Memory Cloud



System Stack



Trinity Specification Language



Why TSL?

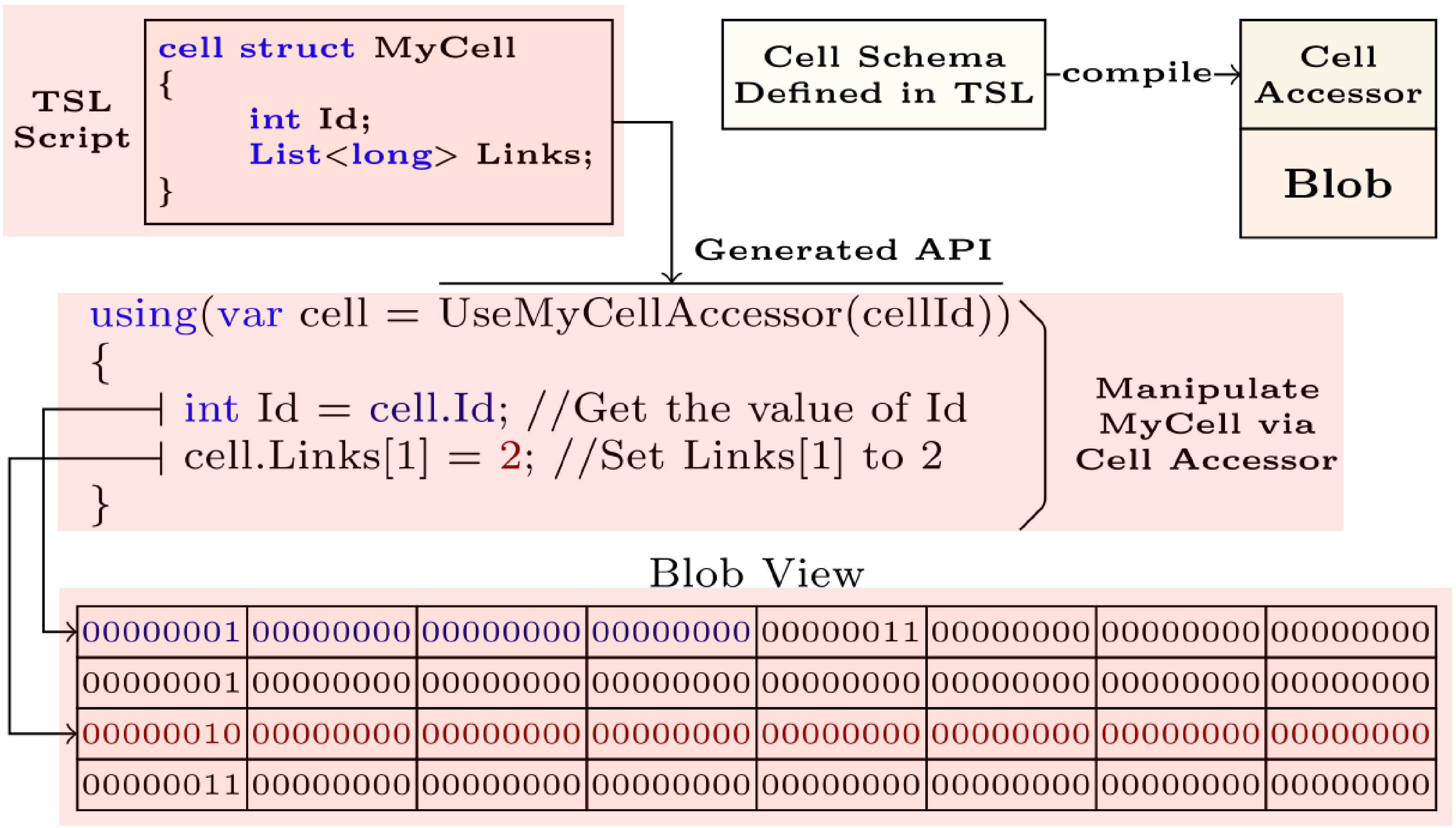
- TSL allows users to define graph schemata, and communication protocols through declarative interfaces.

- TSL makes Trinity memory cloud beyond a key-value store
  - Users are allowed to freely define the data schema
  - TSL makes message passing programming ever so easy

Modeling a Movie and Actor Graph

```
[CellType: NodeCell]
cell struct Movie
{
    string Name;
    [EdgeType: SimpleEdge, ReferencedCell: Actor]
    List<long> Actors;
}
[CellType: NodeCell]
cell struct Actor
{
    string Name;
    [EdgeType: SimpleEdge, ReferencedCell: Movie]
    List<long> Movies;
}
```

TSL-enabled Cell Accessor: Efficient and User-friendly



Modeling Message Passing

```
struct MyMessage
{
    string Text;
}
protocol Echo
{
    Type: Syn;
    Request: MyMessage;
    Response: MyMessage;
}
```

Trinity-enabled Graph Computation Paradigms

- **Vertex-centric graph analytics**
  - Prosperous since Pregel, e.g. Giraph, GraphChi
- **Approximate graph computation based on local sampling**
  - Enabled by randomly partitioned in-memory graph
  - Fast approximate computation with minimum communication costs
  - Application: distance oracle [VLDB 2014]
- **Index-free real-time online query processing**
  - Enabled by fast in-memory distributed graph exploration
  - Examples, subgraph match (vldb 2012) and Trinity.RDF (vldb 2013)

Graph Engine

SERVING BIG GRAPH IN REAL-TIME

http://www.graphengine.io/

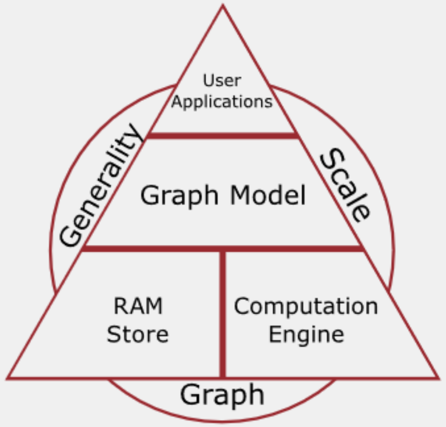
Graph Engine

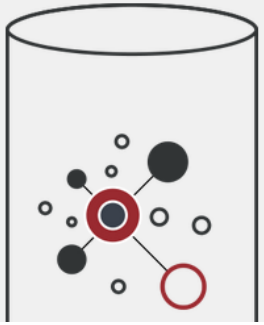
= RAM Store + Computation Engine + Graph Model

Graph Engine (GE) is a highly modularized graph processing system, underpinned by a strongly-typed RAM store and a general computation engine.

The distributed RAM store provides a globally addressable high-performance key-value store over a cluster of machines. Through the RAM store, GE enables the fast random data access power over a large distributed data set.

The capability of fast data exploration and distributed parallel computing makes GE a natural large graph processing platform. GE supports both low-latency online query processing and high-throughput offline analytics on billion-node large graphs.





Strongly-typed RAM Store

Schema Matters

Schema does matter when we need to process data efficiently. Strongly-typed data modeling is crucial for compact data storage, fast data access, and clear data semantics.

One Byte Counts

GE is good at managing billions of run-time objects of varied sizes. One byte counts as the number of objects goes large. GE provides fast memory allocation and efficient memory reallocation with

Thanks!

http://www.graphengine.io/