Trinity Graph Engine

Bin Shao, Yatao Li, Wei-Ying Ma

Microsoft Research Asia
Why do we need a graph system?
Existing Systems

• Mature data processing systems
  – RDBMS
  – Map Reduce Systems, e.g. cosmos

• Systems specialized for certain graph operations:
  – PageRank
Graph Data is “Special” ...

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

- Unstructured nature of graph
  - Difficult to extract parallelism by partitioning data
  - Hard to get an efficient “Divide and Conquer” solution
Graph in the Jail of Storage

• RDBMS/cosmos, mature but not for graphs

• The commonest graph operation “traversal” incurs excessive amount of table joins
Challenge I: Diversity of Graphs

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

- Satori Schema Graph
- Social Network
- Internet Web Graph
- Protein Interaction Network
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!

<table>
<thead>
<tr>
<th># of Nodes</th>
<th># of Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>24 million</td>
<td>1 billion</td>
</tr>
<tr>
<td>504 million</td>
<td>1.4 billion</td>
</tr>
<tr>
<td>1 billion</td>
<td>1 trillion</td>
</tr>
<tr>
<td>1.2 billion</td>
<td>31 billion</td>
</tr>
<tr>
<td>1.4 billion</td>
<td>140 billion</td>
</tr>
<tr>
<td>1 trillion</td>
<td>1 trillion</td>
</tr>
</tbody>
</table>

Linked Data

Satori

Facebook

Web

US Road Map
Roadmap of the Graph Engine project

Develop Apps

Applications

Design Algorithms

Algorithms

Trinity Graph Engine

Build System

Data Modeling and Programming Interfaces

Distributed in-memory Computation Engine

RAM Store

Message Passing

Real-time graph serving on Satori, Freebase, enterprise graph etc.

Subgraph matching, SPARQL, distance oracle, graph partitioning, reachability ...

[VLDB 2012, 2013, 2014], ICDE 2014

Trinity Graph Engine:
[Sigmod 2012, 2013]  

Research

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

- Random access challenge
- Fast random access
- RAM capacity limit of single machine
- Parallel computation
- Low latency online query processing
- High throughput offline analytics
System Stack

- Graph APIs
  - GetInlinks(), Outlinks.Foreach(...), etc
- Graph Model
- Trinity Specification Language
- Memory Cloud
  - (Distributed Key-Value Store)
- Distributed Memory Storage
- Message Passing Framework
One Byte Counts
(Trinity vs. PBGL)
Trinity Specification Language

Graph Modeling

OMG IDL

ICE Slice

TSL

Google ProtoBuf

Message Passing Modeling

Data interchange Format Specification
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

cell struct Movie {
    string Name;
    [GraphEdge]
    List<CellId> Actors;
}

cell struct Actor {
    string Name;
    [GraphEdge]
    List<CellId> Movies;
}
TSL-enabled Cell Accessor: Efficient and User-friendly

```csharp
using(var cell = UseMyCellAccessor(cellId))
{
    int Id = cell.Id; //Get the value of Id
    cell.Links[1] = 2; //Set Links[1] to 2
}
```

Blob View:

```
00000001 00000000 00000000 00000000 00000011 00000000 00000000 00000000
00000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000010 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000011 00000000 00000000 00000000 00000000 00000000 00000000 00000000
```
Modeling Message Passing

```c
struct MyMessage
{
    string Text;
}
protocol Echo
{
    Type: Syn;
    Request: MyMessage;
    Response: MyMessage;
}
```
TSL-Powered Message Passing
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)
## Query Index Examples

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

Index-based subgraph matching [Sun VLDB 2012]
## Query Index Examples

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>Index Size for Facebook</th>
<th>Index Time for Facebook</th>
<th>Query Time on Facebook (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ullmann [Ullmann76], VF2 [CordellaFSV04]</td>
<td>-</td>
<td>-</td>
<td>&gt;1000</td>
</tr>
<tr>
<td>RDF-3X [NeumannW10]</td>
<td>1T</td>
<td>&gt;20 days</td>
<td>&gt;48</td>
</tr>
<tr>
<td>BitMat [AtreCZH10]</td>
<td>2.4T</td>
<td>&gt;20 days</td>
<td>&gt;269</td>
</tr>
<tr>
<td>Subdue [HolderCD94]</td>
<td>-</td>
<td>&gt; 67 years</td>
<td>-</td>
</tr>
<tr>
<td>SpiderMine [ZhuQLYHY11]</td>
<td>-</td>
<td>&gt; 3 years</td>
<td>-</td>
</tr>
<tr>
<td>R-Join [ChengYDW08]</td>
<td>&gt;175T</td>
<td>&gt; 10^{15} years</td>
<td>&gt;200</td>
</tr>
<tr>
<td>Distance-Join [ZouCO09]</td>
<td>&gt;175T</td>
<td>&gt; 10^{15} years</td>
<td>&gt;4000</td>
</tr>
<tr>
<td>GraphQL [HeS08]</td>
<td>&gt;13T(r=2)</td>
<td>&gt; 600 years</td>
<td>&gt;2000</td>
</tr>
<tr>
<td>Zhao [ZhaoH10]</td>
<td>&gt;12T(r=2)</td>
<td>&gt; 600 years</td>
<td>&gt;600</td>
</tr>
<tr>
<td>GADDI [ZhangLY09]</td>
<td>&gt; 2 \times 10^5 T (L=4)</td>
<td>&gt; 4 \times 10^5 years</td>
<td>&gt;400</td>
</tr>
</tbody>
</table>

Index-based subgraph matching [Sun VLDB 2012]
Index-free Query Processing
Trinity Applications
Source Code Graph (Visual Studio)
Source Code Graph (Visual Studio)
FROM a in { "author.FirstName" = 'Leslie Lamport' } MATCH a --> b (Paper.Author, Organization) --> c (Paper) SELECT a.firstName, c.title

<table>
<thead>
<tr>
<th>a.FirstName</th>
<th>c.Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>Leslie Lamport</td>
<td>Composition: A Way to Make Proofs Harder</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>A Formal Basis for the Specification of Concurrent Systems</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>The Operators of T1AC</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>The Synchronization of Independent Processes</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>Corrigendum: &quot;A New Approach to Proving the Correctness of Multiprocess Programs&quot;</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>Comment on Bell's quadratic quotient method for hash coded searching</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>SIFT: Design and analysis of a fault-tolerant computer for aircraft control</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>Latex: a document preparation system</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>Constructing digital signatures from a one-way-function</td>
</tr>
<tr>
<td>Leslie Lamport</td>
<td>Specifying</td>
</tr>
</tbody>
</table>

1 2 3 4 5 6 7 8 9 10 ->
Knowledge Graph
Satori Knowledge Graph
Powered by Trinity
Challenges of Serving Satori

- Complex data schema
  - Rich relations
Satori: An ever-growing knowledge repository

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw RDF data</td>
<td>4T+</td>
</tr>
<tr>
<td>Entities</td>
<td>2.4B+</td>
</tr>
<tr>
<td>Triple Facts</td>
<td>20B+</td>
</tr>
</tbody>
</table>

**Challenges of Serving Satori**

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

```
123 mso/type.object.name "Pal"

123 mso/type.object.type mso/organism.dog
123 mso/organism.dog.breeds "Collie Rough"

123 mso/type.object.type mso/film.actor
123 mso/film.actor.film 789
789 mso/type.object.type mso/film.film
789 mso/type.object.name "Lassie Come Home"
```
Challenges of Serving Satori

- Complex data schema
  - Rich relations
  - Multi-typed entities
- Distributed in-memory knowledge graph

<table>
<thead>
<tr>
<th>Satori: An ever-growing knowledge repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw RDF data</td>
</tr>
<tr>
<td>Entities</td>
</tr>
<tr>
<td>Triple Facts</td>
</tr>
</tbody>
</table>
Challenges of Serving Satori

- Complex data schema
  - Rich relations
  - Multi-typed entities
Trinity’s Storage Architecture for Satori

Entity
Relations
+
In-memory
Entity Properties

On-disk
Entity Properties

Memory
Disk

Entity Properties

film
actor
director

name

name

name

…
Demo
SATORI

Satori Service Portal
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...
<table>
<thead>
<tr>
<th>API Names</th>
<th>Availability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetEntityIdByName</td>
<td>Available</td>
<td>Gets a list of Trinity entity ids by the specified entity name.</td>
</tr>
<tr>
<td>GetPredicatesByEntityId</td>
<td>Available</td>
<td>Gets a list of predicates for the entity with the specified Trinity entity Id.</td>
</tr>
<tr>
<td>GetValuesByEntityPredicate</td>
<td>Available</td>
<td>Gets the values of the specified predicates for the specified entity.</td>
</tr>
<tr>
<td>GetSubjectByPredicateObject</td>
<td>Available</td>
<td>Gets the subjects for the given object and a predicate.</td>
</tr>
<tr>
<td>GetEntityIdBySatoriId</td>
<td>Available</td>
<td>Gets the corresponding Trinity entity Id for the specified Satori Guid.</td>
</tr>
<tr>
<td>GetSatoriIdByEntityId</td>
<td>Available</td>
<td>Gets the corresponding Satori Guid for the specified Trinity entity Id.</td>
</tr>
<tr>
<td>GetRankedEntityIdByName</td>
<td>Available</td>
<td>Gets a list of Trinity entity ids by the specified entity name sorted by their static rank.</td>
</tr>
<tr>
<td>GetScoredValuesByEntityPredicate</td>
<td>Available</td>
<td>Gets the values of the specified predicates for the specified entity, sorted by confidence score.</td>
</tr>
<tr>
<td>GetSortScoredValuesByEntityPredicate</td>
<td>Available</td>
<td>Gets the values of the specified predicates for the specified entity, sorted by the column index (1 for...</td>
</tr>
<tr>
<td>GetEntityDescription</td>
<td>Available</td>
<td>Gets the description of the specified entity Id.</td>
</tr>
</tbody>
</table>
Graphical Query Interface

Node Information
Alias: v2
Type: film_director
URI: ms0 Film_jump

Conditions:

Outputs:

You could add some conditions as:

name = Steve_S
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...
Thanks!

http://www.graphengine.io/