Horton – Online Query Execution Engine for Large Graphs

Mohamed Sarwat*, Sameh Elnikety, Yuxiong He, Gabriel Kliot
*University of Minnesota & Microsoft Research

**Horton**
- Manage and query large graphs online.

**System Design**
- Use a declarative query language.
- Graph is main-memory resident.
- Graph is partitioned among several servers.

**Data Model**
- A node has id, categorical type, and attributes.
- An edge has direction, categorical type, and attributes.

**Query Language**
- Regular language reachability.
- Query is sequence of node and edge predicates.

**Examples**
- Alice’s photos
  - Photo, tags, Alice
    - **Node:** type=photo, **edge:** type=tags, **node:** type=person, name=Alice
    - Result: matching paths
- Alice’s org chart
  - Alice, (manages, person)*

**Execution Engine**
- Query is transformed into a finite state machine.
- Graphs is traversed in BFS manner constrained by the finite state machine.

---

### Declarative query:

<table>
<thead>
<tr>
<th>Declare</th>
<th>Navigational program:</th>
</tr>
</thead>
</table>
| ✅ Photo, tags, Alice | ```
Foreach(n1 in graph.Nodes.SelectByType(photo))
{
    foreach(n2 in n1.GetNeighboursByEdgeType(tag))
    {
        if(node2 == Alice)
        {
            return path(node1, tag, node2)
        }
    }
} ``` |