

# Support for Concept Hierarchies in DHTs

Athanasia Asiki,  
Katerina Doka,  
Dimitrios Tsoumakos  
and Nectarios Koziris

School of Electrical and Computer Engineering  
National Technical University of Athens

Computing Systems Laboratory  
Email: {nasia, katerina, dtsouma, nkoziris}@cslab.ece.ntua.gr

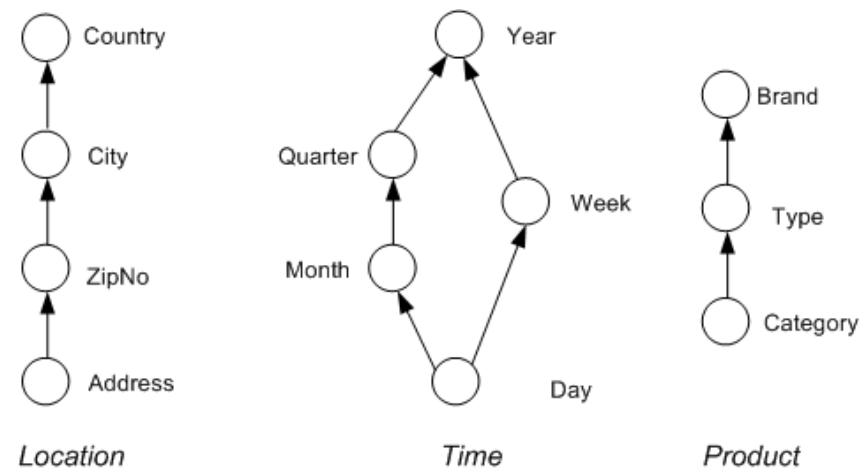
# Motivation

---

- ▶ Efficient, on-line processing of data
  - ▶ Over multiple dimensions
  - ▶ Organized in **hierarchies**
  
- ▶ Assist in the reuse and efficient manipulation
  
- ▶ Observed in many applications
  - ▶ Computer networks (e.g., router data)
  - ▶ Business (e.g., sales data)
  - ▶ Data warehouses

# Concept Hierarchies

- ▶ A sequence of mappings from more general to lower-level concepts
- ▶ Allow the structuring of information into categories
- ▶ Allow queries in multiple levels of granularity



# Our Goals

---

- ▶ Efficient storage of hierarchical data in DHTs

<i>Location</i>				<i>Measure</i>
GR	Athens	16674	Prom. st.	1500€
IT	Rome	00100	Camp. st.	1990€

- ▶ Query data on different levels of granularity (online!)
  - ▶ Roll-up towards more generalized and summarized levels
  - ▶ Drill-down towards more detailed levels
- ▶ Maintain hierarchy-specific information
  - ▶ *Which country is Athens part of?*
  - ▶ *Which addresses correspond to zip-code 15341?*

# Overview

---

- ▶ Data insertion, while maintaining hierarchy-specific information
- ▶ Data Lookup
  - ▶ With DHT lookups for values of the *pivot levels*
  - ▶ With soft-state indices
  - ▶ With flooding
- ▶ Adaptive reindexing on a per-node basis according to the incoming queries

# Data Insertion

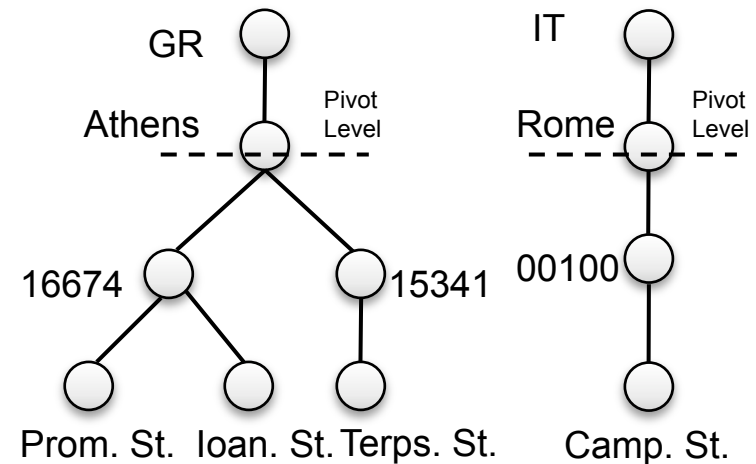
- ▶ A level of the hierarchy is selected as *pivot level*
- ▶ Keys are the hashed values of the *pivot levels*
- ▶ Tuples are stored in *tree structures*
- ▶ Statistics are maintained for each tree

Fact Table

*Pivot Level*

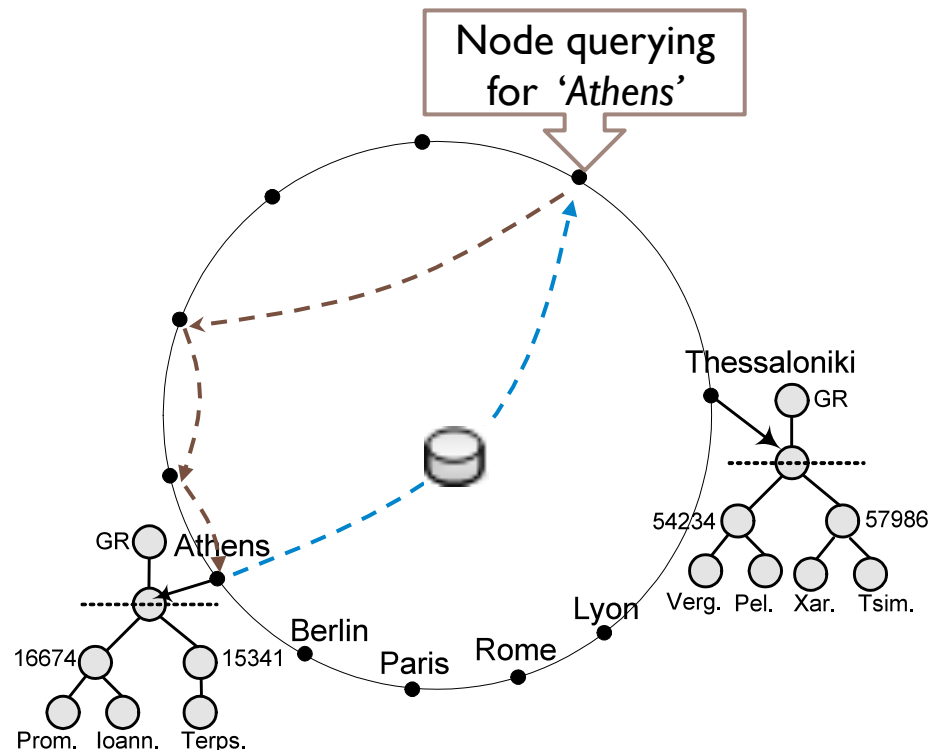
↓

GR	Athens	16674	Prom. st.	1500
GR	Athens	16674	loan. st.	9900
GR	Athens	15341	Terps. st.	2450
IT	Rome	00100	Camp. st.	1990



# Data Lookup (1)

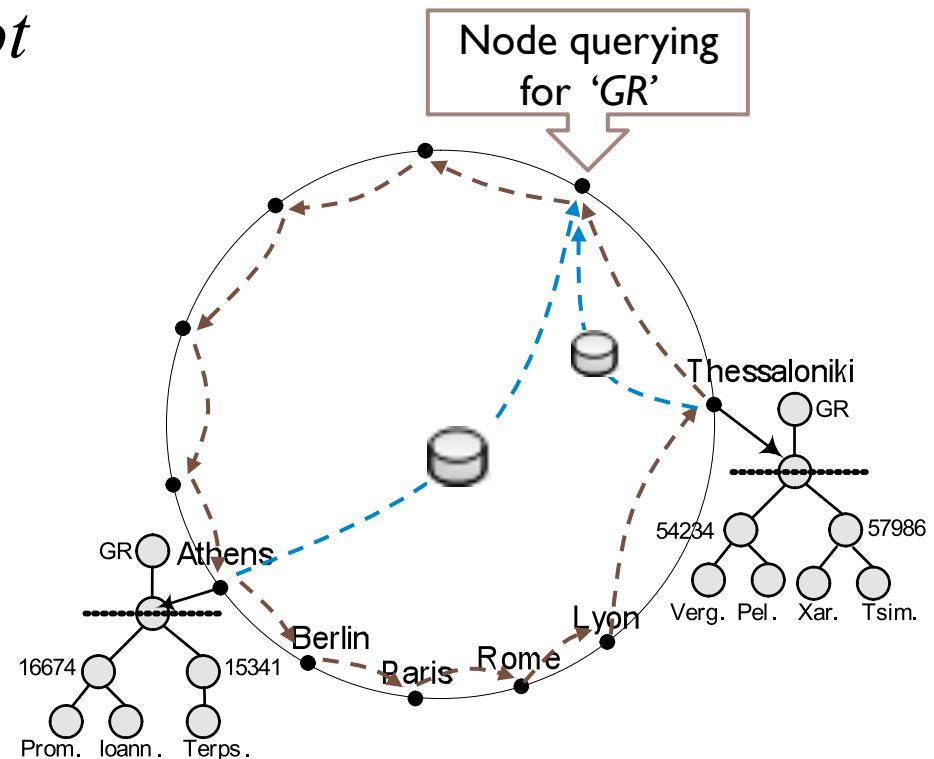
- ▶ Queries concerning values of the *pivot level*:
  - ▶ Answered by the DHT lookup mechanism within  $O(\log N)$  steps



## Data Lookup (2)

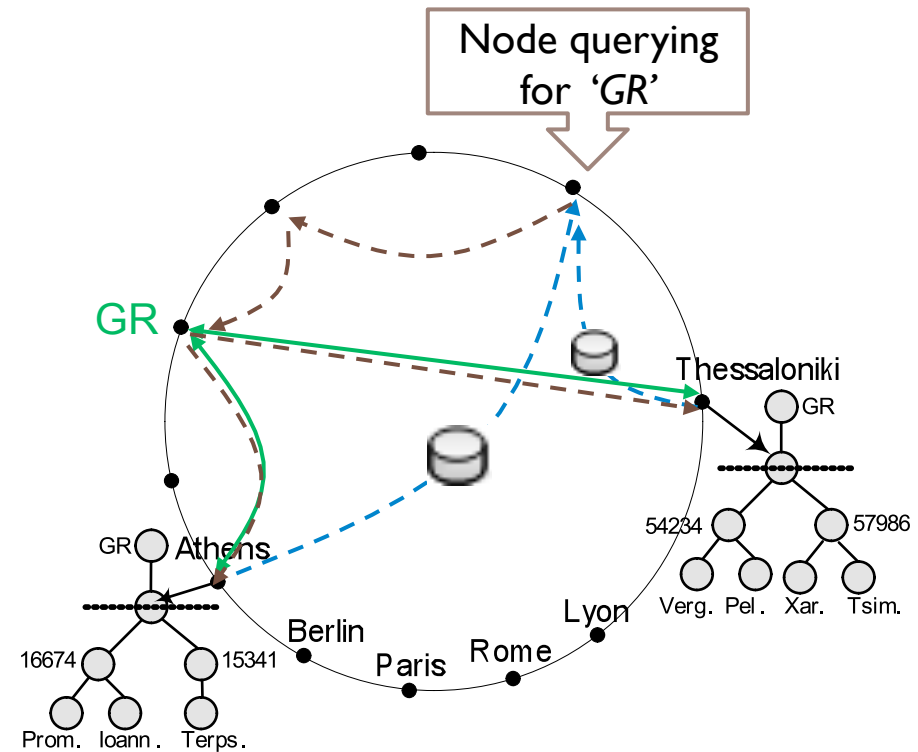
► Queries for values of other levels than the *pivot* level :

- Flooding of the query
- If the node stores tuples with the queried value:
  - Checks statics of the corresponding tree
  - Informs the query initiator if roll-up or drill-down is needed



# Soft-state Indices

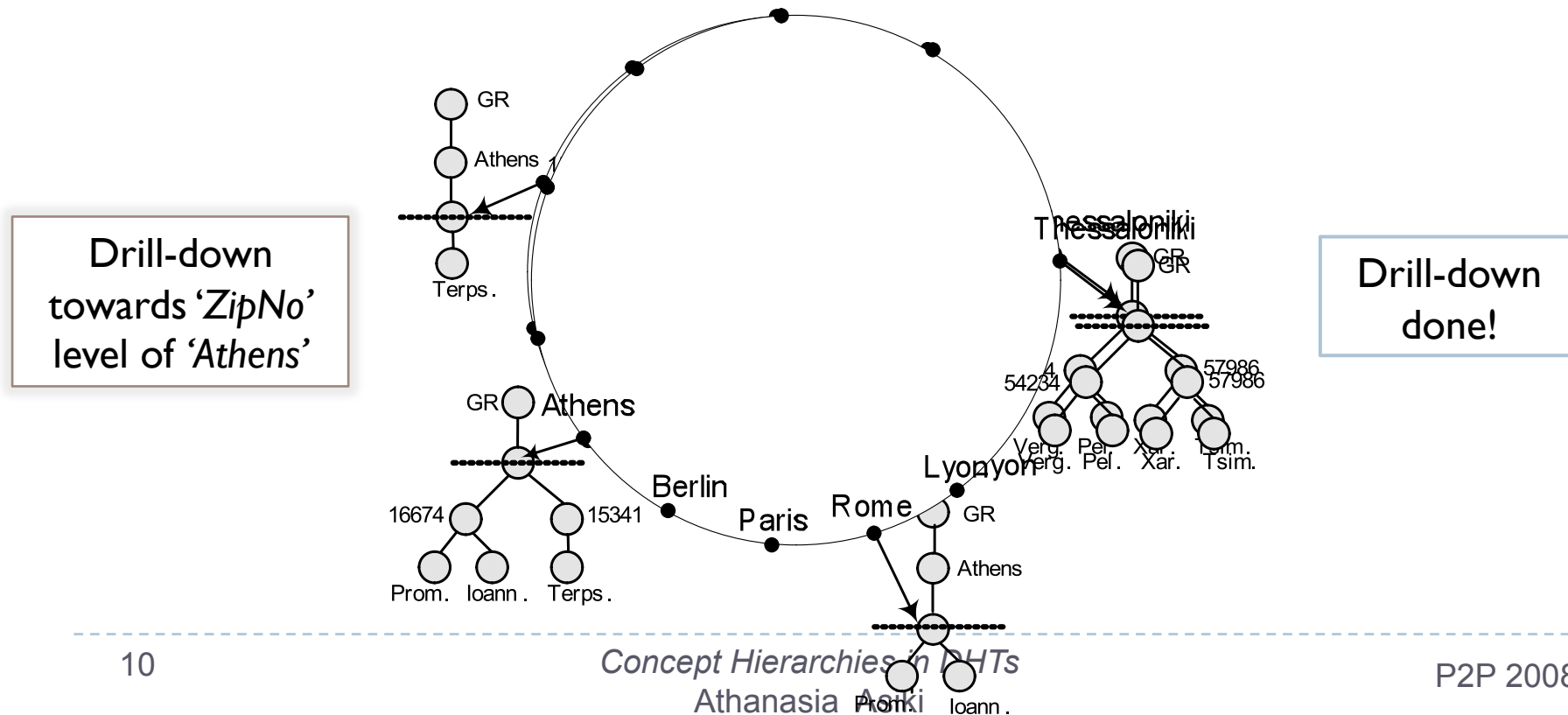
- ▶ After flooding:
  - ▶ Roll-up or drill-down is performed
  - ▶ *OR* a soft-state index is created
  
- ▶ Lookup of indexed data
  - ▶ Lookup ends up in the node storing the index
  - ▶ Index is renewed
  - ▶ Query is forwarded only to nodes with actual data



# Adaptive Reindexing (1)

## ► Drill-down

- Only one node stores values lower than the *pivot level*
- If the total number for  $l_\alpha$  is *threshold%* or more of the total number of queries for this tree



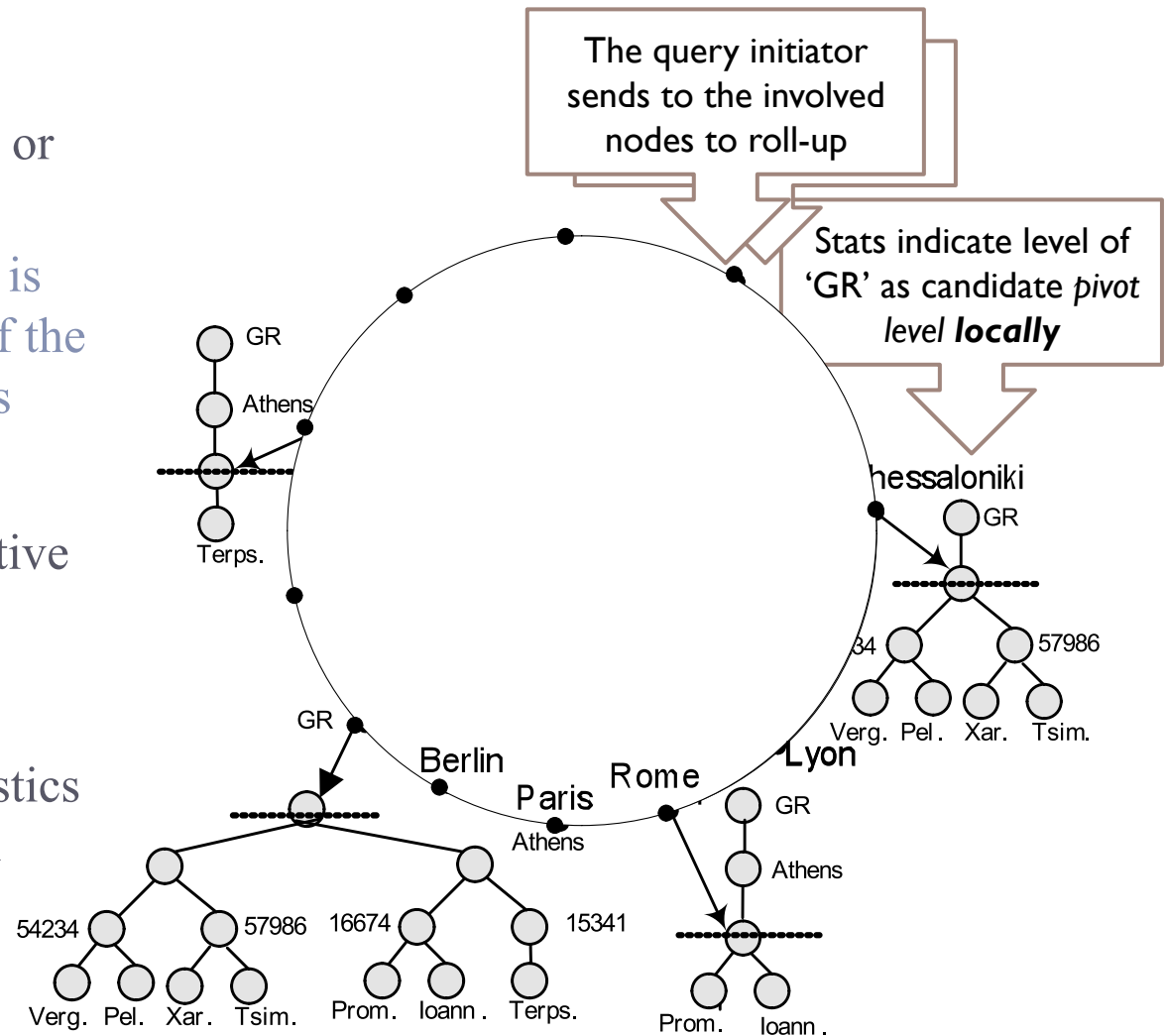
# Adaptive Reindexing (2)

## ► Roll-up

- Positive to roll-up locally or roll-up:

the total number for  $I_\alpha$  is *threshold%* or more of the total number of queries

- The initiator of a flooded query is informed of positive to roll-up nodes
- Roll-up is decided by the query initiator from statistics of all trees with the asked value



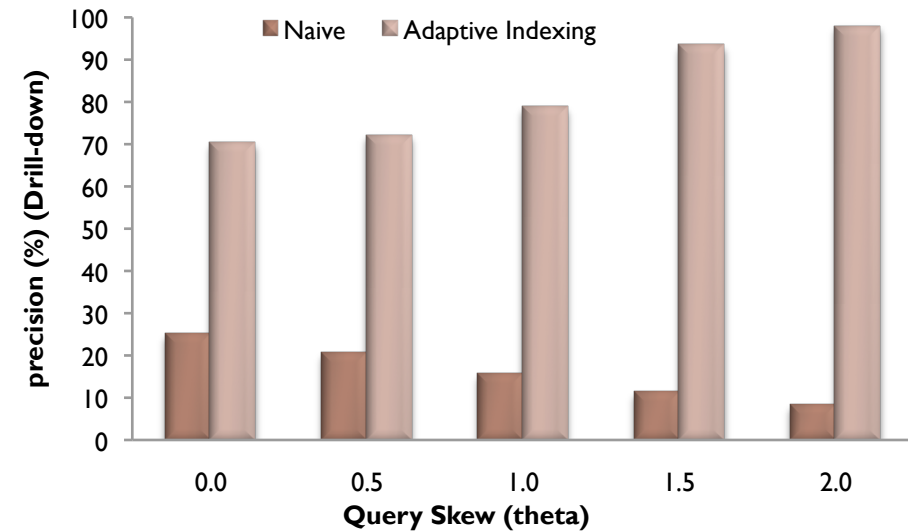
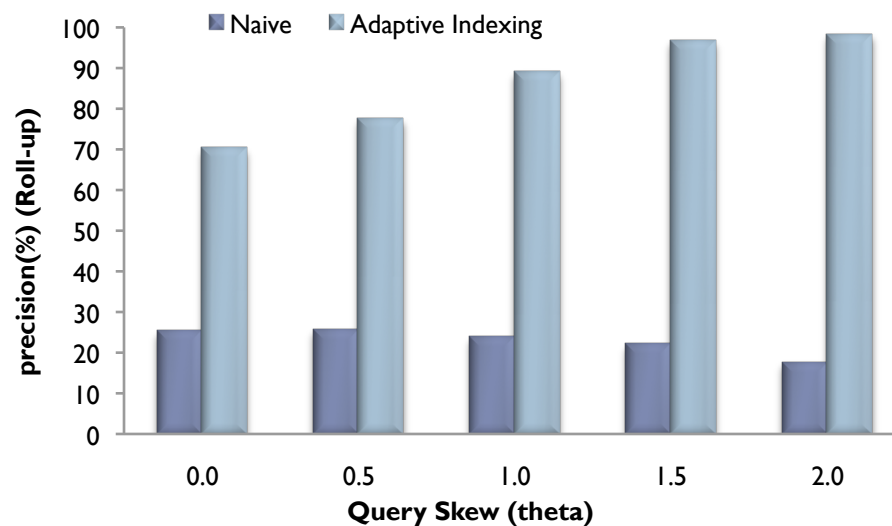
# Initial Experimental Results (1)

---

- ▶ 512 in a modified FreePastry DHT
- ▶ *Synthetically generated data:*
  - ▶ 5000 tuples
  - ▶ 4-level concept hierarchy with one numerical fact
  - ▶ Values distribution
    - ▶  $|l_0|=100|$  (Country level)
    - ▶  $|l_1|=500|$  (City level)  $\Rightarrow$  *Pivot level* during initial insertion
    - ▶  $|l_2|=1000|$  (ZipNo level)
    - ▶  $|l_3|=5000|$  (Address level)
  - ▶ Uniform distribution of values per level
  - ▶ Each distinct value of level  $l_i$  has a constant number of children in  $l_{i+1}$
- ▶ *Queries*
  - ▶ *Zipfian distribution*
- ▶ *Measurements*
  - ▶ **precision (%)**: *Ratio of queries answered without flooding*

## Initial Experimental Results (2)

- ▶ Better performance for more skewed workloads
- ▶ over 70% precision for uniform workload
- ▶ Number of roll-ups and drill-downs adapts to the query skew



# Questions

---

