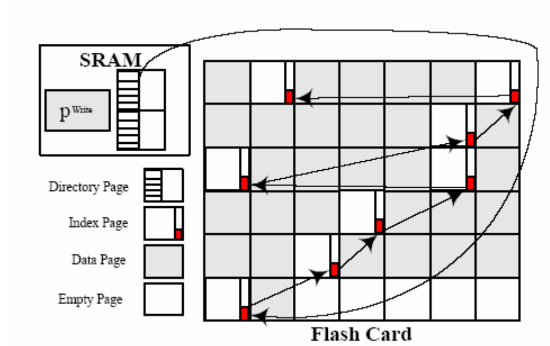
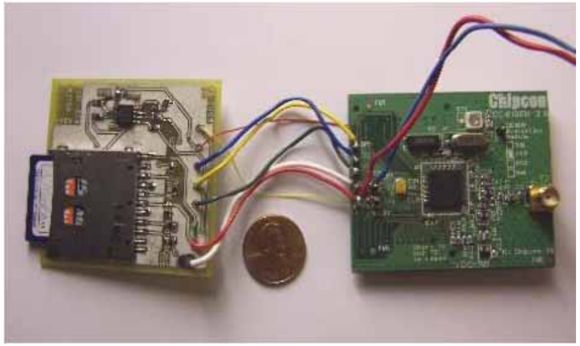
“Data Storage In Wireless Sensor Databases”

Demetris Zeinalipour
Department of Computer Science
University of Cyprus*



eNEXT WG1 Workshop on Sensor and Ad-hoc Networks

University of Cyprus - UCY, Nicosia, 13-14 March 2006

* Presented work was conducted at the University of California – Riverside.

http://www2.cs.ucy.ac.cy/~dzeina/

Presentation Goals

- To present a new **perspective** on **data management** and **query processing** related issues in sensor networks.

- This is an **overview talk** of various individual aspects that are important in this context.

- It does not focus on networking related technologies, but rather on how to organize the information generated by sensors in an energy-efficient manner.

Sensor Networks & The Silicon Era

- Applications:
  - Environmental and habitant monitoring
  - Seismic and Structural monitoring, ….
- Result:
  - Non-Intrusive/Non-Disruptive technology that enables the human to **monitor** and **understand** the physical world.

Wildlife Tracking: GPS Collars

Environmental Monitoring

Structural Monitoring

The typical SensorNet Framework

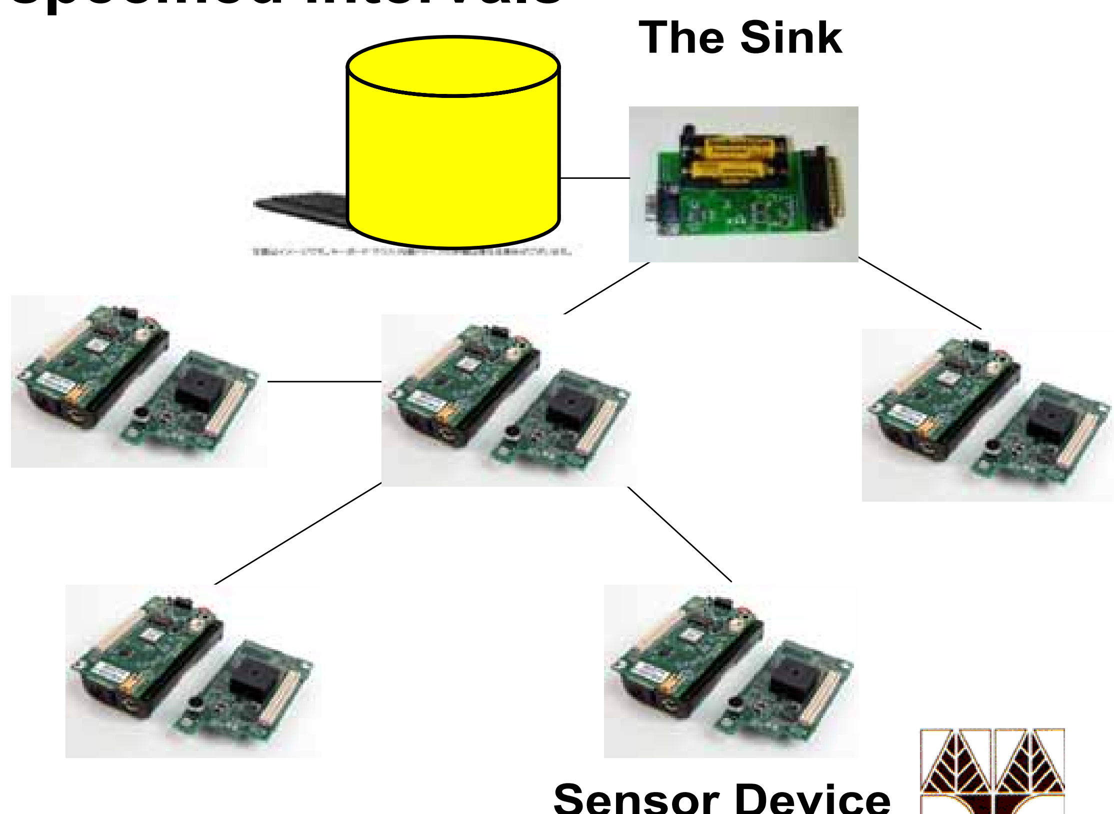
Sense and Send Paradigm

Sensors acquire environmental parameters and transmit these to the sink at pre-specified intervals

A Database that collects readings from many Sensors

Centralized:

- **Storage, Indexing**
- **Query Processing**
- **Triggers, etc..**



The typical SensorNet Framework

Data Acquisition

TinyDB (SIGMOD'03) and **Cougar** (CIDR'03) Frameworks:

- Provide a declarative SQL-like approach for accessing data.
- Are suitable for continuous queries.
- Push aggregation in the network (**TAG** - OSDI'02) but keep much of the processing at the sink.

SELECT $\{AGG(expr), attrs\}$
FROM $\{table\}$
WHERE $\{selectPreds\}$
GROUP BY $attrs$
HAVING $\{havePreds\}$

EPOCH DURATION i

But Many applications do not require the query to be evaluated continuously…

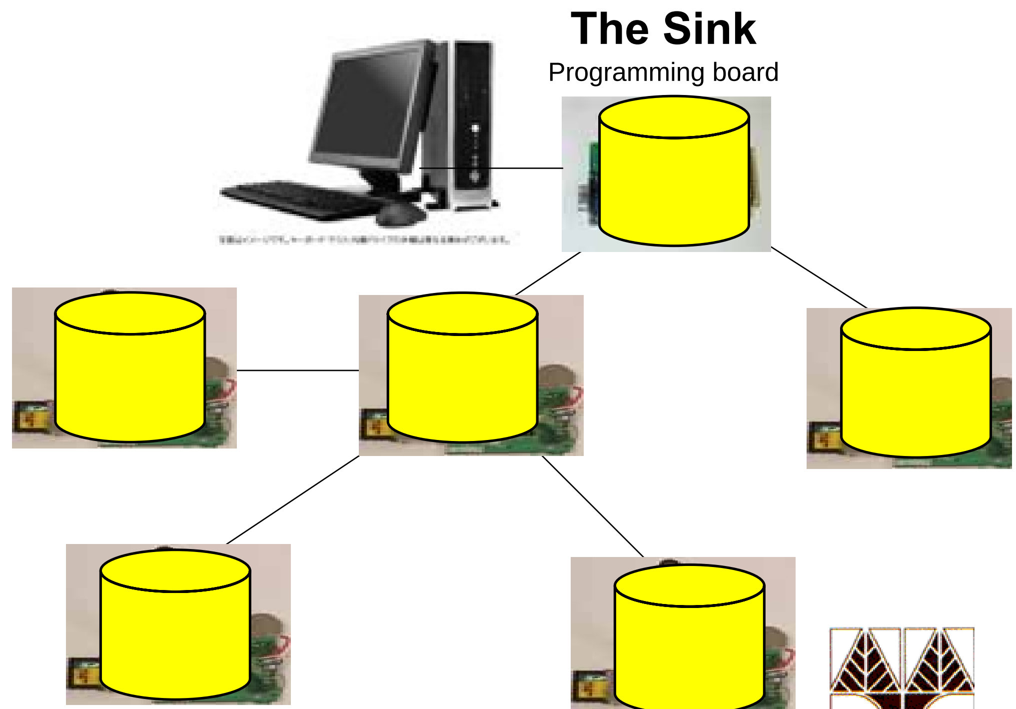
Our Model: In-Situ Data Storage

1. Sensors acquire readings from their surrounding environment.
2. The data remains In-situ (at the generating site) in a sliding window fashion.
3. *When Users want to search/retrieve some information they perform on-demand queries.*

A network of Sensor Databases

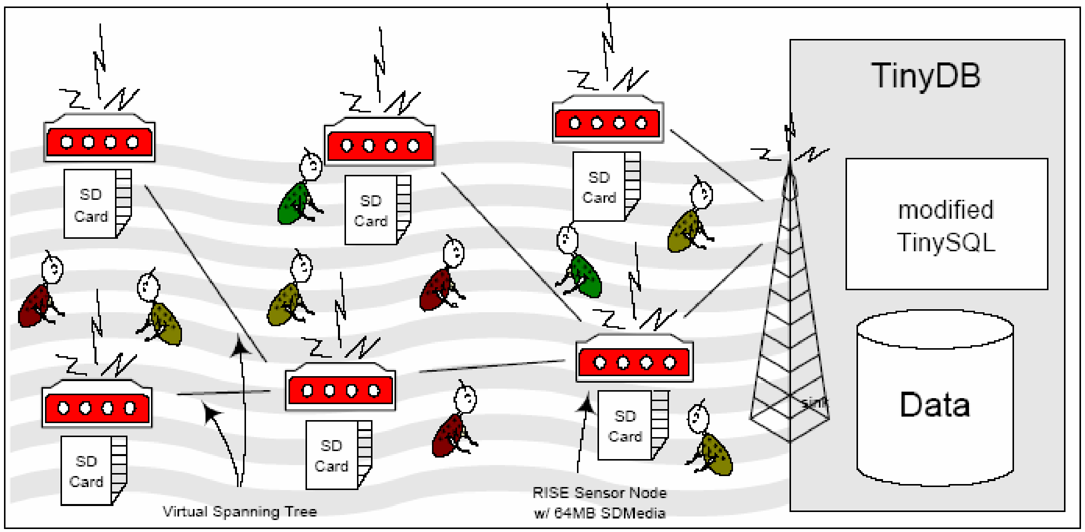
- Distributed Storage
- Distributed Query Processing

Objective: To minimize the utilization of the radio



Our Motivation

- The **Bio-complexity** and **James Reserve** projects at **UC-Riverside**, where biologists want to utilize non-intrusive, not necessarily online, technologies to monitor **CO_2 levels in the soil**, rather that in laboratory recreations.
- Scientists do not need answers to their queries at all times.
- However a query execution has to adhere to the distinct characteristics of a Wireless Sensor Environment (minimize communication, local processing and aggregation, etc).



Challenges of the In-Situ Model

- **How to efficiently store information locally**

  Solution: We build the RISE Sensor that features an external flash memory – Giga-scale storage)

  [IEEE/ACM IPSN’05, IEEE SECON’05, ACM Senmetrics’05]

- **How to efficiently access a Giga-Scale storage medium of a Sensor Device?**

  Solution: We build the MicroHash Index Structure

  [IEEE NetDB (ICDE’05), USENIX FAST’05]

- **How to find the most important events without pulling together all distributed relations?**

  Solution: We build the Threshold Join Algorithm

  [IEEE DMSN’05 (VLDB’05)]

Talk Outline

1. **The RISE Hardware Platform.**

2. Indexing on Flash Memory of a Sensor Device.

3. Distributed top-k Query Processing.

4. Conclusions and Future Work.

A) The RISE Hardware Platform

The ***RISE (RIverside SEnsor)*** has been built as the prototype sensor platform demonstrating the In-Situ Data Storage Paradigm

- High performance, low power, state of the art platform
- Built around the **Chipcon CC1010** (System on Chip)
- Incorporates **TinyOS v1.1** (with nesC v1.2alpha1)
- Gigabyte scale - High capacity flash data storage (SD-Card)
- Multitude of sensors (Temperature, Carbon dioxide, Humidity, etc)
- Integrated radio transceiver - Compatible with MICA for interoperability and investigation into the nature of heterogeneous networks.

A) The RISE Hardware Platform

The RISE Sensor Specs

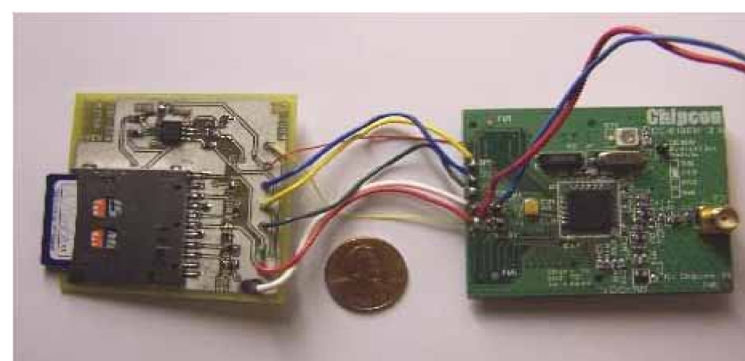
Characteristic	Capability
MCU	
Processor	24 MHz 8051 core
On-Chip Flash Memory	32 KB
Current (On,Idle,Off) at 14 MHz	14.8 mA, 8.2 mA, 0.2 μA
Radio (RF Transceiver)	
Communication Rate	76.8 kbits/s
Communication Range	250m at 868/915 MHz
Current (Receive,Send at 10dBm)	11.9 mA, 26.6 mA
SD Card & SPI Bus	
SPI bus rate	Up to 3 Mbps
Data page size	512 bytes
Data block size	16 KBytes
Current (Read,Write,Delete)	1.17mA, 37 mA, 57μA
Time (Read,Write,Delete) (512B)	6.25ms, 6.25ms, 2.26ms

Table 1. Characteristics of the RISE platform.

IPSN'05

The RISE Storage board (RISE v2) [IEEE SECON'05, SenMetrics'05]

1. Data in RISE v1 is stored on the external SDMedia (NAND flash).
2. NAND flash is not suitable for accessing data at a byte granularity.
3. RISE Storage Board features NOR flash (efficient byte-level granularity) and the SDMedia Card.
4. It complements RISE v1



Talk Outline

1. The RISE Hardware Platform.

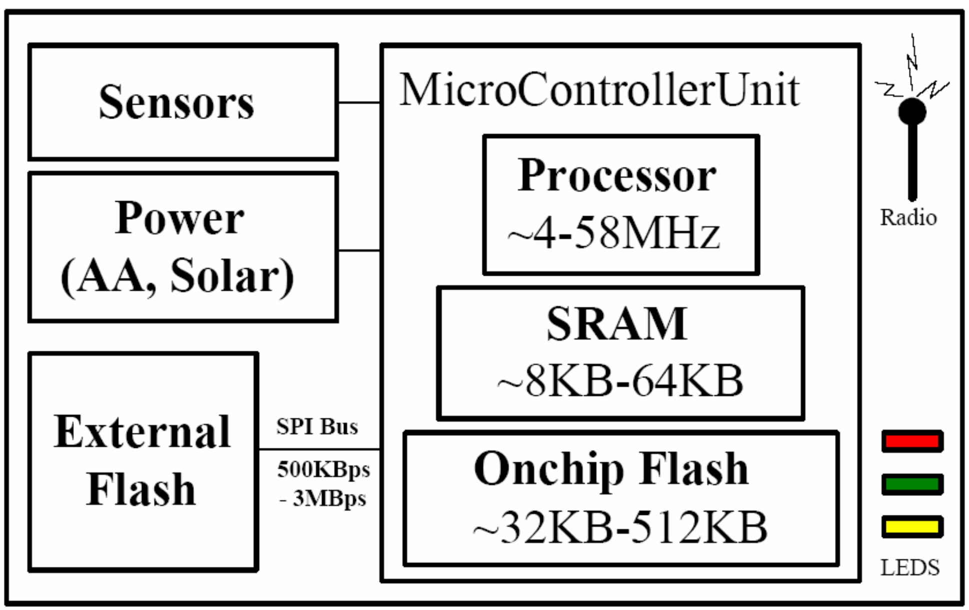
2. **Indexing on Flash Memory of a Sensor Device.**

3. Distributed top-k Query Processing.

4. Conclusions and Future Work.

B) Indexing on Sensor Devices

- **Task:** *“Find from local storage all records that satisfy some query predicate”* (e.g. temp=95F)
- The most prevalent volatile medium for a Sensor Devices is **Flash Memory**.



<u>Flash (NAND) Advantages</u>

- Simple Cell Architecture
- Economical Reproduction
- Shock Resistant
- Fast Read Times
- Power Efficient

B) Indexing on Sensor Devices

Why is Flash so different from other Storage Mediums (disks, ram, etc)?

(1) **Read-Constraint:** Reading data stored on flash memory can be performed at granularity ranging from a single byte to a whole block (typically 8KB-64KB).

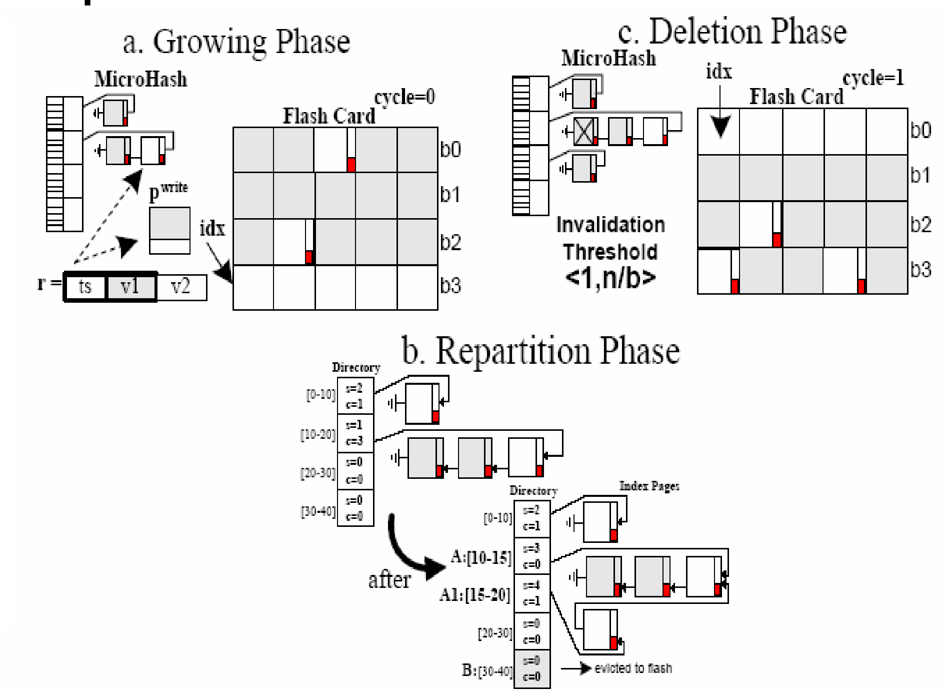
(2) **Delete-Constraint:** Deleting data stored on flash memory can only be performed at a block granularity (i.e. 8KB-64KB).

(3) **Write-Constraint:** Writing data can only be performed at a page granularity (typically 256B-512B), after the respective page (and its respective 8KB-64KB block) has been deleted.

(4) **Wear-Constraint:** Each page can only be written a limited number of times (typically 10,000-100,000).

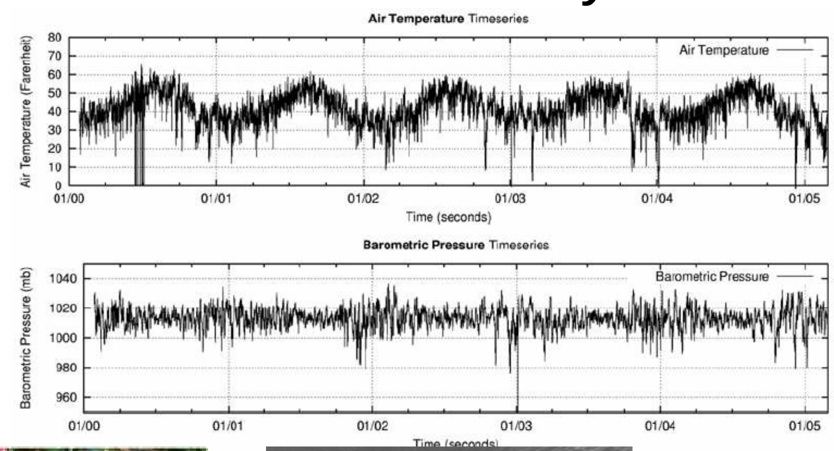
B) Indexing on Sensor Devices

- There is no related work on **Local Indexes for Sensor Device Databases** (most research focuses on Magnetic Disk and Main Memory Databases)
- We developed the MicroHash Index **[FAST'05]** which is an efficient structure to this problem.
- We also developed efficient **Search algorithms** that locate information stored on flash.
- **Main Idea:** Minimize expensive random access deletions



B) Indexing on Sensor Devices

- We have implemented all these algorithms in nesC, the programming language of TinyOS.
- Extensive trace-driven simulations using 5-year long temperature/humidity datasets from the University of Washington.
- We also used datasets from the Great Duck Island Study in Maine (UC-Berkeley)



B) Indexing on Sensor Devices

- Finding a record **by a value (e.g. temp=95F)** can be performed in constant time.
- Finding a record **by timestamp (e.g. 14/3/06 10:30:00)** can be performed in 3-6 page reads.

Great Duck Island Study

Index On Attribute	Overhead Ratio $\Phi(\%)$	Energy Index (mJ)	ScaleSearch Avg Page Read
Light	26.47	4,134	4.45
Temperature	27.14	4,172	5.45
Thermopile	24.08	4,005	6.29
Thermistor	14.43	3,554	5.10
Humidity	7.604	3,292	2.97
Voltage	20.27	3,771	4.21

Note: Storing records without index 3042mJ

Talk Outline

1. The RISE Hardware Platform.
2. Indexing on Flash Memory of a Sensor Device.
3. **Distributed top-k Query Processing.**
4. Conclusions and Future Work.

3) Distributed Top-K Query Processing

Motivating Example (Problem Formulation)

- Assume that we have **n=5** sensor each of maintains locally a sliding window of **m=5** readings. (See table)
- ***TOP-1 Query:*** *"On which timestamp did we have the highest temperature across all sensors?"*
- ***Note: Score(o_i) can only be calculated if we combine the readings from all 5 sensor.***

Local score

Timestamp

s1	s2	s3	s4	s5	TOP-5
o3,.99	o1,.91	o1,.92	o3,.74	o3,.67	o3,4.05/5=.81
o1,.66	o3,.90	o3,.75	o1,.56	o4,.67	o1,3.63/5=.73
o0,.63	o0,.61	o4,.70	o2,.56	o1,.58	o4,2.07/5=.41
o2,.48	o4,.07	o2,.16	o0,.28	o2,.54	o0,1.88/5=.32
o4,.44	o2,.01	o0,.01	o4,.19	o0,.35	o2,1.75/5=.29

Complete Score

m

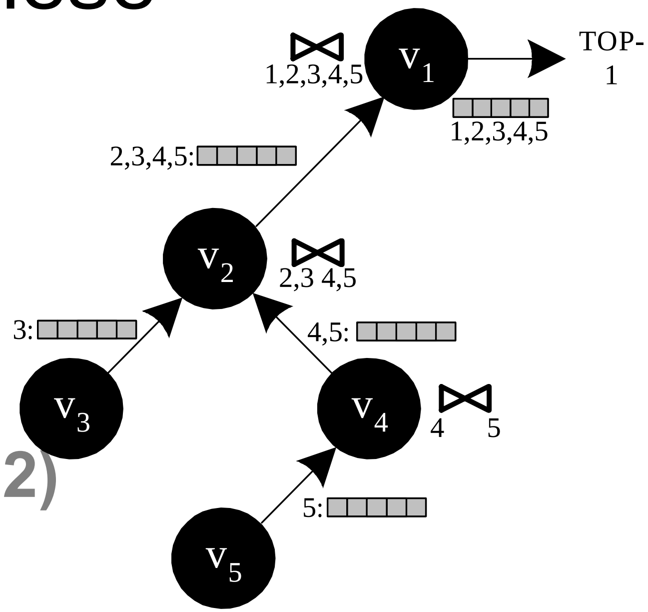
n

$list_{s1}$

Current Approach: TAG

- Aggregate the lists before these are forwarded to the parent

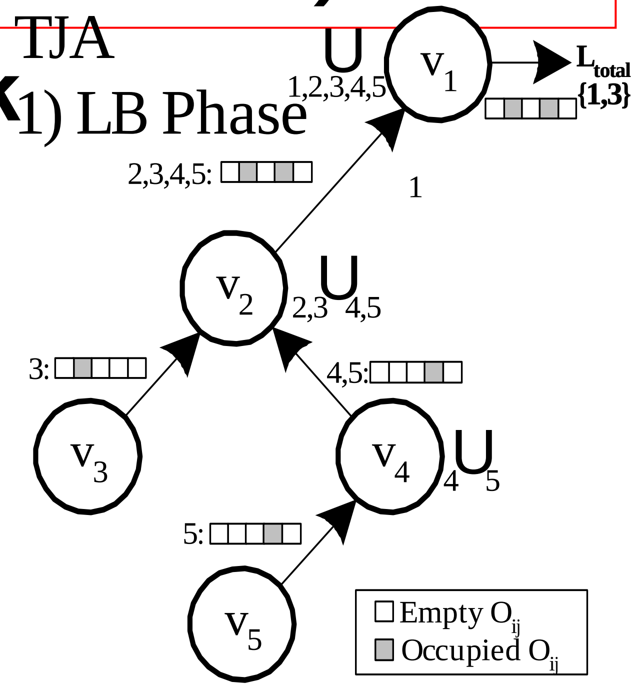
- This is essentially the TAG approach (Madden et al. OSDI '02)



- **Advantage:** Only (n-1) messages

- **Drawback:** Sending everything!

TJA Step 1 (LB Phase)



- Each node sends its **top-k** results to its parent.
- Each intermediate node performs a **union** of all received lists (denoted as **τ**):

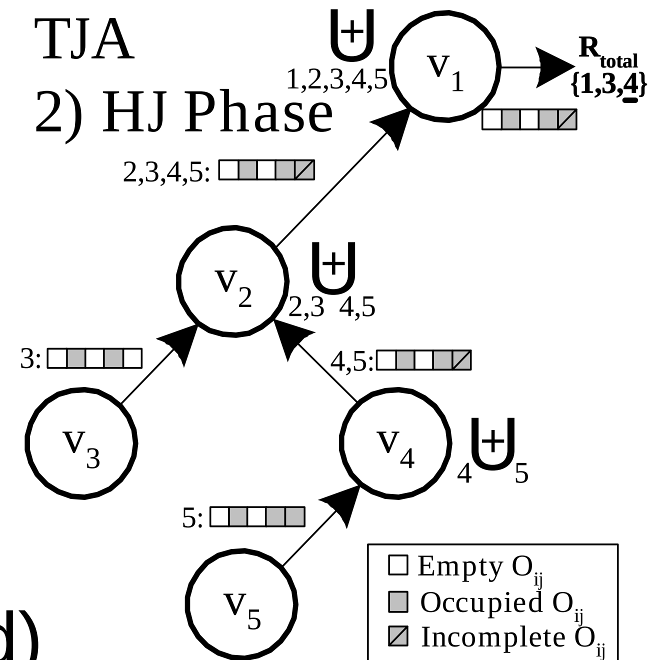
Query: TOP-1

v1	v2	v3	v4	v5
o3, 99	o1, 91	o1, 92	o3, 74	o3, 67
o1, 66	o3, 90	o3, 75	o1, 56	o4, 67
o0, 63	o0, 61	o4, 70	o2, 56	o1, 58
o2, 48	o4, 07	o2, 16	o0, 28	o2, 54
o4, 44	o2, 01	o0, 01	o4, 19	o0, 35

LB
{o3, o1}

TJA Step 1 (HJ Phase)

- Disseminate **τ** to all nodes
- Each node sends back everything with score above all objectIDs in **τ**.
- Before sending the objects, each node tags as incomplete scores that could not be computed exactly (upper bound)



v1	v2	v3	v4	v5
o3, 99	o1, 91	o1, 92	o3, 74	o3, 67
o1, 66	o3, 90	o3, 75	o1, 56	o4, 67
o0, 63	o0, 61	o4, 70	o2, 56	o1, 58
o2, 48	o4, 07	o2, 16	o0, 28	o2, 54
o4, 44	o2, 01	o0, 01	o4,19	o0, 35

HJ	
o3, 405	Complete
o1, 363	Complete
o4',354	Incomplete

TJA Step 1 (CL Phase)

Have we found K objects with a <u>complete</u> score?

Yes: The answer has been found!

No: Find the *complete score* for each incomplete object (all in a single batch phase)

- **CL ensures correctness!**
- **This phase is rarely required in practice.**

v1	v2	v3	v4	v5
o3, 99	o1, 91	o1, 92	o3, 74	o3, 67
o1, 66	o3, 90	o3, 75	o1, 56	o4, 67
o0, 63	o0, 61	o4, 70	o2, 56	o1, 58
o2, 48	o4, 07	o2, 16	o0, 28	o2, 54
o4, 44	o2, 01	o0, 01	o4, 19	o0, 35

TOP-5
o3,405
o1, 363
o4, 207
o0, 188
o2, 175

Talk Outline

1. The RISE Hardware Platform.
2. Indexing on Flash Memory of a Sensor Device.
3. Distributed top-k Query Processing.
4. **Conclusions and Future Work.**

Conclusions and Future Work

- In-Situ Data Storage is a new approach for **data management** in Sensor Networks

- We want to incorporate the presented ideas in a unified **In-Situ Storage and Retrieval Management System**, similar to TinyDB, but distributed.

Related Publications

- ## Indexing on Flash Memory
  - **MicroHash Index:**

    D. Zeinalipour-Yazti, S. Lin, V. Kalogeraki, D. Gunopulos, W. Najjar **"MicroHash: An Efficient Index Structure for Flash-Based Sensor Devices"**, 4th USENIX Conference on File and Storage Technologies (**FAST'2005**), San Francisco, CA, 2005.

  - **Indexing Spatiotemporal Records MicroGF – Online Compression Algorithms:**

    S. Lin, **D. Zeinalipour-Yazti,** V. Kalogeraki, D. Gunopulos, W. Najjar **"Efficient Indexing Data Structures for Flash-Based Sensor Devices"**, under review.

- ## TOP-K Query Processing & In-Situ Data Storage
  - **D. Zeinalipour-Yazti,** Z. Vagena, D. Gunopulos, V. Kalogeraki, V. Tsotras, M. Vlachos, N. Koudas, D. Srivastava "The Threshold Join Algorithm for Top-k Queries in Distributed Sensor Networks", Proceedings of the 2nd international workshop on Data management for sensor networks DMSN (VLDB'2005), Trondheim, Norway, 2005.
  - **D. Zeinalipour-Yazti,** S. Neema, D. Gunopulos, V. Kalogeraki and W. Najjar, "Data Acquision in Sensor Networks with Large Memories", IEEE Intl. Workshop on Networking Meets Databases NetDB (ICDE'2005), Tokyo, Japan, 2005.
  - **D. Zeinalipour-Yazti**, V. Kalogeraki, D. Gunopulos, A. Mitra, A. Banerjee and W. Najjar "*Towards In-Situ Data Storage in Sensor Databases*", 10th Panhellenic Conference on Informatics (**PCI'2005**) Volos, Greece, 2005.

Related Publications (cont.)

- **RISE Hardware platform**
  - A. Banerjee, A. Mitra, W. Najjar, **D. Zeinalipour-Yazti**, V. Kalogeraki and D. Gunopulos **"*RISE Co-S : High Performance Sensor Storage and Co-Processing Architecture*",** Second Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, (**SECON'2005**), Santa Clara, California, USA, to appear in 2005.
  - A. Mitra, A. Banerjee, W. Najjar, **D. Zeinalipour-Yazti**, V. Kalogeraki and D. Gunopulos, **"*High-Performance Low Power Sensor Platforms Featuring Gigabyte Scale Storage*",** IEEE/ACM 3rd International Workshop on Measurement, Modelling, and Performance Analysis of Wireless Sensor Networks SenMetrics'2005, (**collocated w/ MobiQuitous'2005**), San Diego, CA, to appear in 2005.
  - S. Neema, A. Mitra, A. Banerjee, W. Najjar, **D. Zeinalipour-Yazti**, D. Gunopulos, V. Kalogeraki, **"*NODES: A Novel System Design for Embedded Sensor Networks*",** IEEE Intl. Conference on Information Processing in Sensor Networks (**IPSN'2005**) (Demo), Los Angeles, CA, to appear in 2005.

"Data Storage In Wireless Sensor Databases"

Demetris Zeinalipour

Department of Computer Science

University of Cyprus*

Thank You!

eNEXT WG1 Workshop on Sensor and Ad-hoc Networks

University of Cyprus - UCY, Nicosia, 13-14 March 2006

*** Presented work was conducted at the University of California – Riverside.**

http://www2.cs.ucy.ac.cy/~dzeina/