TinyDB / GSK
Past, Present, Future

Sam Madden
Nest Retreat
6/17/03
Intro

• TinyDB
  - Query processing and macro-programming engine mote networks
    » Aggregates, in-network storage, etc.
    - “find the average light value in every room of this hotel”

• GSK
  - TinyDB based “kit”
    » Focus: simple “selection” queries
    » Suite of deployment tools, GUI, etc.
    » Server-side database which stores:
      • Configuration information
      • Queries
      • All results (raw and processed)
  - This talk: focus on mote aspects

GSK Team
Wei Hong (fearless leader)
Phil Buonadonna
David Culler
Anind Dey
David Gay
Alan Mainwaring
Rob Szewczyk
Su Ping
Joe Polastre
Outline

- Progress since last retreat
- Current implementation goals
- Current research plans
Outline

- Progress since last retreat
- Current implementation goals
- Current research plans
Progress

• Time Sync Integration
  - Some issues still remain...

• Power Management
  - Simple version, based on Service Scheduler
    » Depends on time synchronization
  - Processing
    » Sampling, sending, receiving
• **GSK Server**
  - Stable, in-use on GDI, integrated with TinyDB and GUI tools

• **Support for Mica2 / Mica2Dot**
  - Lots of new attributes
  - Long startup times
  - Calibration coefficients (on server side)

• **Isolation of routing code**
  - No longer required to use TinyDB routing
  - Also support Surge3, HSN

• **Extensible aggregate framework**

• **JUnit based test harness**
• Support for logging via MatchBox
  - Tables persist across resets
  - Up to 1/2 meg of storage
  - Higher data rates than available over-air
    » Need to push this

• Syntax:

  CREATE TABLE stuff (field1 uint16, field2 uint16)
  SELECT nodeid,light FROM sensors INTO stuff
  SELECT field1, field2 FROM stuff
Outline

- Progress since last retreat
- *Current implementation goals*
- *Current research plans*
Implementation Tasks

- Deploy
  - Lab
  - Botanical Garden
  - GDI (?)
  - SeaWorld (w/ DTN)
Implementation Tasks

• Release -- soon, after GSK testing
  - When is the next TinyOS release

• Things that need to be built:
  - Features from ACQP, TAG papers
  - Mote -> Server metadata registry
  - Higher data rates
  - Precompiled queries
  - DIM (Xin)
    » Distributed Index for Multidimensional Data
  - DTN layer
Outline

• Progress since last retreat
• Current implementation goals
• Current research plans
  – 4 of many!
Research Tasks

• Investigate relationship between TinyDB and PARC sensor tasking, tracking applications
  - Handoff based querying vs. tree-based querying
    » Queries which follow a phenomena through the network
  - IDSQ vs. Tree-based aggregation
• Idea: Task sensors in order of most valuable contribution to some aggregate:
  - Choose leader(s)
    » Suppress subordinates
    » Task subordinates, one at a time
  • Until some measure of goodness is met
    - E.g. "Mahalanobis Distance" -- Accounts for correlations in axes, tends to favor minimizing principal axis

Model location estimate as a point with 2-dimensional Gaussian uncertainty.

Preferred because it reduces error along principal axis.

Area of residuals is equal.
IDSQ vs. Tree Based Routing

• IDSQ provides a nice way to talk about “certainty” of aggregates

• IDSQ Advantages
  - Leader can stop when estimate is “good enough”
  - Picks nodes in “good” order

• IDSQ Disadvantages
  - Leader tasks nearby nodes it knows about
  - Lots of round trips
    » Costly, especially if leader horizon is > 1

• Tree based approach can be more precise
  - Uses information from all nodes

• Challenges
  - Avoid transmissions from nodes which contribute little
  - How to best construct the tree?
  - Do better than a single Gaussian as error estimate?

Joint work with Mark Paskin
Approximate Queries

- Sketches
  - Sketches provide single pass space-efficient estimates of lots of interesting aggregates
    » E.g. Second frequency moment in $\log(n)$ space
      • Estimate size of self-join
      • Useful in query optimization
  - Generally randomized aggregates with good probabilistic bounds
    » Philosophically in-line with sensor networks
Multi-round Queries

• Previously, we “reset” state after each aggregate
  - What about multi-pass aggregates
    » E.g. wavelets, more efficient medians
  - Or algorithms that depend on the state of the previous aggregate
    » E.g. “sensors whose temperature more than 10% above the average”
DTN & TinyDB

• Is this more than “store and forward” networking?

• DB level policies for dropping results?
  - See “ACQP”
    » Value based prioritization of results
    » What does “admission control” mean for a streaming, continuous process?
Conclusions

• TinyDB, GSK in-deployment
  - Use it, please!

• Increasing complexity
  - In-network queries (query handoff)
  - Queries with confidences
  - Multi-round stuff
  - Interaction with a variety of networking layers

• Lots of fun problems to chase...
  - Ranging from mathy to systemsy, and everything in between!