SARANA: A Spatially-Aware and Resource-Aware Network Architecture

Ulrich Kremer  Margaret Martonosi  Li-Shiuan Peh

Computer Science Dept.  Electrical Engineering Dept.
Rutgers University  Princeton University

A New Target Platform?
Ad-Hoc / Dynamic Networks: A New Computing Platform

Ad-hoc/dynamic networks make sharing and cooperation possible among small devices.

Related Work:
- Network of Workstations (NOW)
- Peer-to-Peer (P2P) systems

Applications for Dynamic Networks

- **Collaborative computing.**
  - Location-sensitive
  - Exploiting multiple services
  - In-network processing
  - Location resilient services

- **Sensor data collection**
  - Location-sensitive
  - QoR vs. resource consumption tradeoff
  - In-network processing
  - Streamed data
  - Time constraint

- **Augmented-reality games.**
  - Location-sensitive
  - Physical environment attached with virtual information
An AMBER Alert Emergency System

<table>
<thead>
<tr>
<th>Initialization Phase</th>
<th>Search Phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>- dynamically deploy: problem-specific image understanding software</td>
<td>- periodic exchange of information between nearby AMBER participants</td>
</tr>
<tr>
<td>- space resilient software coverage</td>
<td>- periodic exchange of information between nearby AMBER participants</td>
</tr>
<tr>
<td>- alert AMBER participants; share initial search information</td>
<td>- reporting back results to emergency center</td>
</tr>
<tr>
<td></td>
<td>- reporting back results to emergency center</td>
</tr>
<tr>
<td></td>
<td>- sharing information with nearby AMBER participants</td>
</tr>
</tbody>
</table>

How does Dynamic Networks (and Sarana) fit in?

- Isolated operation
- Sensor networks
- Fixed, high-infrastructure networks
Dynamic Networking -> Sarana

What we have:
- Mobile, often-disconnected operation
- Need to know location
- Severe energy, bandwidth constraints
- No infrastructure control
- Heterogeneous devices

What we want:
- Sense of connectivity
- Collaborative opn
- Energy-aware services
- Opportunistic
- QoR vs. Latency vs. Energy

How do we achieve this?
- Incentives for peer-2-peer operation
- Resource-aware everything: caching, scheduling, network, comm, compute, ...
- Dynamic cost model & proper abstractions

Sarana Layers

Application: Service invocations, QoR preferences, ...

Sarana Application Support Overlay

Sarana Application Support Overlay

Sarana Application Support Overlay

Sarana Runtime System:
Neighbor lists, data/code management, migration policies, optimization

Sarana Guts:
Resource accounting/attribution, location-awareness, Migration mechanism, data/device mapping

Node Guts:
Code launch / termination
Basic comms (TCP/IP as base but proxies for other devs)
I/O basics: displays, read devices,...
Past Work

- Uli Kremer
  - SpatialViews: A programming model for adhoc/dynamic networks
- Margaret Martonosi
  - ZebraNet: Mobile sensor networks for wildlife tracking
- Li-Shiuan Peh
  - Machine learning-based routing in sensor networks
  - On-chip in-network data management -> wide-area networks?

SpatialViews Project

A programming model for ad-hoc/dynamic networks

Yang Ni, Ulrich Kremer

- Dynamic integration of services
  - Deploy, discover and use services dynamically
- Location-awareness
  - Location specification at the language level
  - Enable location-aware optimizations
- Quality of Result (QoR) specification
  - Result of different qualities (number of visited nodes) with respect to space and time
- Security and Privacy
  - No language support

http://www.cs.rutgers.edu/spatialviews
import SpaceDefs.Rutgers;

public class AverageLighting {
    public static void main(String[] args) {
        spatialview v = LightSensor @ Rutgers.CampusB;
        sumreduction float s = 0;
        sumreduction int n = 0;
        visiteach x : v {
            s += x.read(); n++;
        }
        if (n>0)
            System.out.println(Float.toString(s/n));
        else
            System.out.println("No sensor found.");
    }
}

Example: Aggregating Sensor Data

- A spatial view defines a set of interesting nodes
- “Migration”-based iteration lazily discovers and visits the interesting nodes.
Each node has a separate memory space in the underlying system.
A “Shared Memory” with restrictions at the language level.
- A well-defined semantics for both serial and parallel iterations (“FORALL”).
- A single iteration step is an atomic transaction.
- **Program Variables**: Local Variables and Reduction Variables
  - “Shared” by cloned threads of a program across the network.
  - Local Variables are read-only in a nested iterator.
  - Reduction Variables are write-only with a given reduction operation.
- **Service Variables**
  - Stay on individual nodes, not accessible in a nested iterator.
  - Shared by same or different programs executing on the same node.
Each node has a separate memory space in the underlying system.

- A "Shared Memory" with restrictions at the language level.
  - A well-defined semantics for both serial and parallel iterations ("FORALL").
  - A single iteration step is an atomic transaction.

- **Program Variables**, Local Variables and Reduction Variables:
  - "Shared" by cloned threads of a program across the network.
  - Local Variables are read-only in a nested iterator.
  - Reduction Variables are write-only with a given reduction operation.

- Service Variables:
  - Stay on individual nodes, not accessible in a nested iterator.
  - Shared by same or different programs executing on the same node.

---

Compiler Generated Code for Sensor Data Aggregation Example

```java
public class AverageLighting {
    public static class SV0 extends...
    public static void main(String[] args) 
    ...}
```

- Bytecode is generated. Shown here is the equivalent source code.
- Library code is not shown.
Goal: Biologists want to track animals long-term, over long distances
- Interactions within a species?
- Interactions between species?
- Impact of human development?

Current technology is limited:
- VHF Triangulation is difficult & error-prone
- GPS trackers limit data to coarse sampling and require collar retrieval
- Overall, energy and info retrieval are key limiters
- Peer-to-peer can improve

Research Questions
- Protocols and mobility?
- Energy-efficiency?
- Software layering design?

ZebraNet vs. Other SensorNets
- All sensing nodes are mobile
- Large area: 100's-1000s sq. kilometers
- "Coarse-Grained" nodes
- GPS on-board
- Long-running and autonomous
ZebraNet Deployments in Central Kenya

January, 2004

June, 2005

ZebraNet Software Layers

Key System Novelties:

- Integrated support for networking
- System support for code update over the radio
- Low overhead event scheduling and handling
- Integrated support for data compression

CPU  Radio  GPS  FLASH  Timer  WDT
ZebraNet Accomplishments

- 6 hardware prototyping versions
- Full middleware & OS design: PHY, MAC, energy mgmt, remote software update
- 2 deployments in central Kenya
- Early fine-grained data (and first night-time data) on animal movements

Machine learning-based routing
(Wang, Martonosi, Peh)

Machine learning for predicting link connectivity in lossy wireless networks

Training Sample Collection
Training
Classification & routing instrumentation (MetricMap)

Features:
- RSSI
- sendBuf
- fwdBuf
- depth
- CLA
- pSend
- pRecv

Decision Tree learner, Rule learner, many more...

We use WEKA for training

OFFLINE training

ONLINE optimization

```c
void updateEst(Fv)
{
    if (assembly(Fv.rssi, Fv.sendBuf, Fv.fwdBuf, Fv.depth, Fv.CLA, Fv.pSend, Fv.pRecv) == "good")
    {
        recvEst = 1 * 255
    }
    else {
        recvEst = 0
    }
}
```
Data Management in Chip Multiprocessors

- **Sharing locality**
  - Want to locate the data
  - Optimal: go directly to the nearest sharer

- **Efficient invalidations**
  - Avoid unnecessary traffic (i.e. broadcast)
  - Anticipate invalidations

How? Let the on-chip network handle data management, e.g. coherence (Eisley, Peh, Shang)

- Network is not just for getting from A to B
- It can keep track of sharers too and maintain coherence
- How? With trees stored within routers
  - Tree connecting home node with sharers
  - Tree stored within routers
  - On the way to home node, when requests "bump" into trees, routers will re-route them towards sharers.
Current Status & Next Steps

- Sarana Layering details
- Space-resilient services
- Network-supported data management
- example: traffic motion pattern
- red service (object) tries to stay within blue/green target space
- service migrates if
  - forced to leave current host node due to failure or resource denial
  - host node leaves target area and some other host node is available
- service may need to be redeployed (goal: stay close to target area)
**Topic Highlight: Space Resilient Services**

- **target area**

- Example: traffic motion pattern
- Red service (object) tries to stay within blue/green target space
- Service migrates if:
  - Forced to leave current host node due to failure or resource denial
  - Host node leaves target area and some other host node is available
- Service may need to be redeployed (goal: stay close to target area)

---

**Selected topics 2: In-network code/data management**

- In-network directories keep track of data usage, location of code segments, resource usage, ...
- In-network caching, prefetching and replication of data
- In-network code scheduling and migration based on resource availability
Applications as Drivers of Infrastructure Development

Current SV Apps work

Marcin Pawelek, Shaila Musharoff, Viktor Raskin, William Fong, 2005

Four Applications.

- **Real-time bus schedule**: send queries over an ad-hoc network to get real-time bus locations to minimize waiting time.

  - Straight forward query
  - Caching and sharing query results using user-installed service
  - Bundled query among users who are close to each other

- **Evacuation coordination**: coordinate evacuation of people from a building by using their handheld devices.
Current SV Apps work

- **AR Monopoly game**: “trade” real properties in a real city, e.g., Manhattan.

- **AR Pong game**: run in real world to bounce a “virtual” ball.

Teaching & Projects: Experiences and Plans

- Spring, 2007: Get guts going.
  - PhD students + undergrad indep research

- Summer, 2007: Apps and RT system.
  - PhD students + undergrad indep research

- Fall, 2007: Research continues + prep for a for-credit course

- Spring 2008 (one year from today): Launch joint PU, Rutgers project course. ~18 people total? By application only.
Thank you