Adaptive Spatiotemporal Node Selection in Dynamic Networks

Pradip Hari
John McCabe
Jonathan Banafato
Marcus Henry
Ulrich Kremer
*Rutgers University*

Kevin Ko
Emmanouil Koukoumidis
Margaret Martonosi
_Princeton University_

Li-Shiuan Peh
Massachusetts Institute of Technology
Motivation

• Wireless devices becoming ubiquitous
• 1.13B phones sold in 2009, 174M were smart phones
(Source: Eric Brewer, UC Berkeley, BEARS 2010)
Motivation

- Computing power can be exploited
- Physical location can be exploited
Dynamic Networks

- Spontaneous, dynamic sets of cooperating devices
- Potentially mobile and heterogeneous
- Applications are location- and time-sensitive
- Applications are accountable for resource usage
Sample Application: Amber Alert
Sample Application: Amber Alert
Sample Application: Amber Alert
Dynamic Networks: Challenges

- How to express cooperation of nodes in space and time?
- How to express resource constraints of an application?
- How to select nodes for execution to obtain best results for a limited budget?
Prior Work

- Mostly on sensor networks, not dynamic networks
- Spatial programming (non-resource-aware):
  - SpatialViews [PLDI '05]
  - Regiment [IPSN '07]
  - Pleiades [PLDI '07]
- Resource management (only node- or network-centric):
  - SORA [NSDI '05]
  - Eon [SenSys '07]
  - Pixie [SenSys '08]
Our Solution

• Support for application-centric resource accountability and budgeting
• Framework allowing expression of application’s static and dynamic spatiotemporal properties
• Node selection strategy for improving program outcome by exploiting these spatiotemporal properties
Adaptive Spatiotemporal Node Selection in Dynamic Networks

① Introduction

② Language Abstractions

③ Execution Model

④ Experimental Evaluation

⑤ Conclusions and Future Work
Adaptive Spatiotemporal Node Selection in Dynamic Networks

① Introduction

② Language Abstractions

③ Execution Model

④ Experimental Evaluation

⑤ Conclusions and Future Work
Spatiotemporal Patterns

- Program outcome is determined by spatiotemporal distribution of node visits
Spatiotemporal Patterns

(a) spatial clustering

(f) temporal averaging

spatiotemporal clustering
Sarana

- **Space-Aware, Resource-Aware Network Architecture**
- Language, compiler, and run-time infrastructure
- Parallel macroprogramming framework
- Support for spatial and temporal constraints
- Application-centric cost model / resource management
Assumptions

- Distributed computations as parallel loops
- In-network computation via loop nesting
- Cost model based on virtual currency ("credits")
- Nodes advertise services in system-wide directory
- Application executed with a specified budget
- Possibility of node failure or withdrawal from network
- Security issues will be addressed later
Language

```plaintext
spatialregion cameraSpace = CameraService @ Circle(1000);
```
Language

spatialregion cameraSpace = CameraService @ Circle(1000);

visit (100, 120) camera in cameraSpace by 300

{ }

}
Language

spatialregion cameraSpace = CameraService @ Circle(1000);

visit (100, 120) camera in cameraSpace by 300

{
    ImageBlob image = camera.takePhoto();
}

}
spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet =
    new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300

{
    ImageBlob image = camera.takePhoto();
}

spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet =
    new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300
{
    ImageBlob image = camera.takePhoto();
    foundSet.add(image);
}
spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet =
    new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300

{
    ImageBlob image = camera.takePhoto();
    foundSet.add(image);
    if (myAnalyzer.approves(image))
    {
        report SUCCESS;
    }
}
spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet =
    new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300 :
    cluster-space (SUCCESS, 50)
{
    ImageBlob image = camera.takePhoto();
    foundSet.add(image);
    if (myAnalyzer.approves(image))
    {
        report SUCCESS;
    }
}
spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet = new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300 :
    cluster-space(SUCCESS, 50), disperse-time(SUCCESS, 10)
    {
        ImageBlob image = camera.takePhoto();
        foundSet.add(image);
        if (myAnalyzer.approves(image))
            {
                report SUCCESS;
            }
    }
spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet =
    new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300 :
    cluster-space(SUCCESS, 50), disperse-time(SUCCESS, 10)
{
    ImageBlob image = camera.takePhoto();
    foundSet.add(image);
    double result = myAnalyzer.analyze(image));
    report SUCCESS = result;
}
spatialregion cameraSpace = CameraService @ Circle(1000);

collection_reduction ArrayList<ImageBlob> foundSet =
    new ArrayList<ImageBlob>();

visit (100, 120) camera in cameraSpace by 300 :
    cluster-space(SUCCESS, 50), disperse-space(FAILURE, 10)
{
    ImageBlob image = camera.takePhoto();
    foundSet.add(image);
    double result = myAnalyzer.analyze(image));
    if (result > 0) { report SUCCESS = result; }
    else { report FAILURE; }
}

Adaptive Spatiotemporal Node Selection in Dynamic Networks

① Introduction

② Language Abstractions

③ Execution Model

④ Experimental Evaluation

⑤ Conclusions and Future Work
Adaptive Spatiotemporal Node Selection in Dynamic Networks

① Introduction

② Language Abstractions

③ Execution Model

④ Experimental Evaluation

⑤ Conclusions and Future Work
Execution

- Compiler divides program into *tasks*
- Directory is queried for suitable nodes
- First-pass schedule built, tasks distributed only to necessary targets
- Application evaluates results and reports them to framework
- Feedback from application drives node selection in future passes
Sample Application: Amber Alert
Adaptive Spatiotemporal Node Selection in Dynamic Networks

① Introduction

② Language Abstractions

③ Execution Model

④ Experimental Evaluation

⑤ Conclusions and Future Work
Adaptive Spatiotemporal Node Selection in Dynamic Networks

1. Introduction
2. Language Abstractions
3. Execution Model
4. Experimental Evaluation
5. Conclusions and Future Work
Experimental Evaluation

- Physical prototype
  - 11 Nokia N810 handhelds
  - 14 Neo FreeRunner (OpenMoko) smartphones
  - 1 iBook (laptop)
- Simulation
  - 79 Linux boxes simulating up to 500 nodes
- Applications:
  - Amber Alert (spatiotemporal clustering)
  - Bird Tracking (spatial dispersal, weighted events)
  - Crowd Estimation (spatial coverage, temporal synchronization)
Amber Alert (physical experiments)

25 camera nodes, maximum 27 useful images acquirable
Budget ranges from 5% to 35% of exhaustive search cost
Amber Alert (simulation)

90 camera nodes, maximum 90 useful images acquirable
Budget ranges from 10% to 100% of exhaustive search cost
Bird Tracking (physical experiments)

24 microphone nodes, maximum 4 birds recordable
Budget ranges from 25% to 100% of exhaustive search cost
500 microphone nodes, maximum 51 birds recordable
Budget ranges from 10% to 100% of exhaustive search cost
Crowd Estimation (physical experiments)

25 camera nodes
Crowd Estimation (simulation)

300 camera nodes (100 sampled)
Adaptive Spatiotemporal Node Selection in Dynamic Networks

1. Introduction
2. Language Abstractions
3. Execution Model
4. Experimental Evaluation
5. Conclusions and Future Work
Adaptive Spatiotemporal Node Selection in Dynamic Networks

① Introduction

② Language Abstractions

③ Execution Model

④ Experimental Evaluation

⑤ Conclusions and Future Work
Conclusions

- Spatiotemporal application properties can be exploited to improve outcomes for comparable resource cost
- Implementation of this strategy is feasible on today’s handheld devices
Future Work

- Privacy/security
  - Encrypted communication
  - Identity confirmation
- Compiler optimizations
  - Improved task division (loop interchange, distribution, fusion)