Energy-Efficient Task-Mapping for Data-Driven Sensor Network Macroprogramming Using Constraint Programming

Farshid Hassani Bijarbooneh Pierre Flener
Edith Ngai Justin Pearson

ASTRA and CoRe Research Groups
Department of Information Technology
Uppsala University, Sweden
http://www.it.uu.se/research/group/astra
http://www.it.uu.se/research/group/core

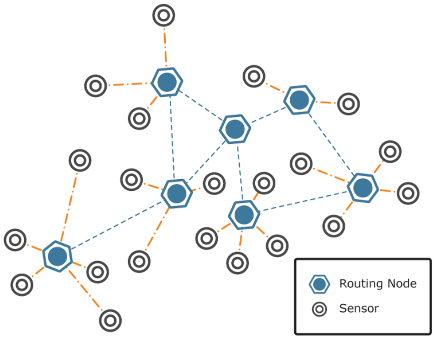
ICS'11, the 12th INFORMS Computing Society Conference
Monterey, California, USA, January 9 – 11, 2011

Outline

Outline

Wireless Sensor Networks

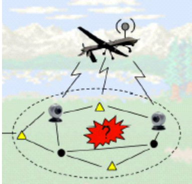
- A node has:
  - A sensor or an actuator.
  - A radio transmitter.
  - A processor.
  - A power source.

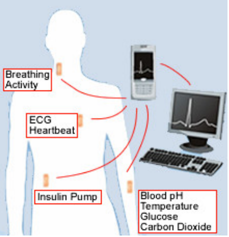


- A task implements applications of the network: The tasks are to be mapped to the nodes.
- A node fires its tasks at a fixed rate per round.
- The network repeats the same behaviour in all rounds.

Wireless Sensor Networks: Applications

- Military and security
- Environment and agriculture monitoring
- Industrial sensing and monitoring
- Health monitoring
- Home automation
- Automotive





UPPSALA UNIVERSITET

Task Mapping Problem

Models

Experiments

Conclusion

Task Types

- Sensing task:
  Calls a sensor to collect data at each round.
  Example: A task sensing the temperature of a room.

- Operative task:
  Operates on data collected by sensing tasks.
  Example: A task computing an average temperature.

- Actuator task:
  Performs an action to affect the environment.
  Example: A task turning on a heater in a room.

UPPSALA UNIVERSITET

Task Mapping Problem

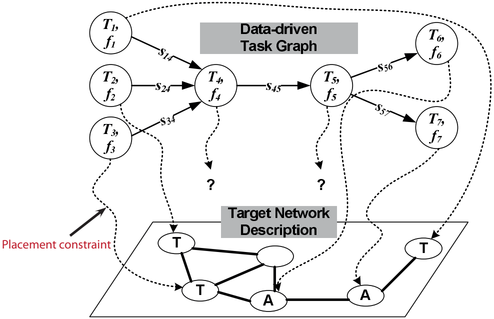
Models

Experiments

Conclusion

Challenges

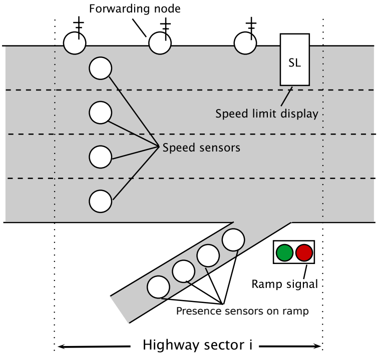
- Programming tasks for sensors is very time consuming.
- Data-driven macroprogramming:
  Create a task graph based on the flow of data, subject to placement and energy constraints:

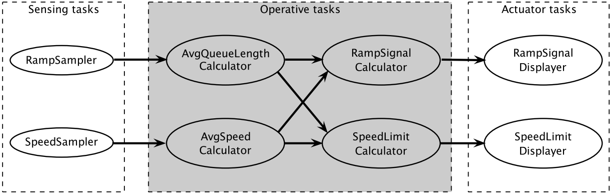


Application: Highway Traffic Management

Reduce the congestion of vehicles on a highway:

- Control speed limits.
- Control highway access.





Goal and Motivation

- Take a published IP model of the problem, and solve it using constraint programming (CP).
- Network communication is the most costly process:
  - The number of tasks running on a node.
  - The task firing rate.
  - The cost of routing a message is paid by all nodes, not just by the end nodes.
- Objective: Minimise the maximum fraction of initial energy spent by any node during one round. That is: Maximise the time to reconfiguration = the time when energy drops below some fraction of the initial energy for some node.

Outline

1. Task Mapping Problem
2. Models
3. Experiments
4. Conclusion

A Mathematical Model

- Constants:
  - N = set of wireless sensor network nodes
  - T = set of tasks
  - A = set of arcs in the task graph (T, A)
  - $f[t]$ = firing rate of task t
  - $s[t, t']$ = size of data sent from task t to task t'
  - $e[n, n', n'']$ = routing energy spent by node n for one unit of data sent from node n' to node n'' via node n
  - $e^0[n]$ = initial energy of node n

A Mathematical Model (continued)

- Decision variables and energy constraints:
  - $node[t] \in N$ = the node assigned to task $t \in T$.
  - $energy[n]$ = the energy spent by node n in one round:

$$energy[n] = \sum_{(t',t'') \in A} f[t'] \cdot s[t',t''] \cdot e[n, node[t'], node[t'']]$$

  Observe the decision variables $node[t']$ and $node[t'']$ among the indices to the given $e[\ldots]$ matrix.

- Objective function, to be minimised:
  The maximum fraction of initial energy spent by any node during one round:

$$\max_{n \in N} \frac{1}{e^0[n]} \cdot energy[n]$$

UPPSALA UNIVERSITET

Task Mapping Problem

Models

Experiments

Conclusion

Integer Programming (IP) Model

- Let $x[t, n] = 1$ iff task t is mapped to node n. The energy constraints temporarily become quadratic:

$$energy[n] = \sum_{(t',t'') \in A} \sum_{n' \in N} \sum_{n'' \in N} f[t'] \cdot s[t', t''] \cdot e[n, n', n''] \cdot x[t', n'] \cdot x[t'', n'']$$

- So let $y[t', n', t'', n''] = x[t', n'] \cdot x[t'', n'']$ and add the following channelling constraints:

$$y[t', n', t'', n''] \leq x[t', n']$$
$$y[t', n', t'', n''] \leq x[t'', n'']$$
$$y[t', n', t'', n''] \geq x[t', n'] + x[t'', n''] - 1$$

- The energy constraints are now linear:

$$energy[n] = \sum_{(t',t'') \in A} \sum_{n' \in N} \sum_{n'' \in N} f[t'] \cdot s[t', t''] \cdot e[n, n', n''] \cdot y[t', n', t'', n'']$$

- We have thus added $|T|^2 \cdot |N|^2 + |T| \cdot |N|$ decision variables, as well as $3 \cdot |T|^2 \cdot |N|^2 + |T|$ constraints.

UPPSALA UNIVERSITET

Task Mapping Problem

Models

Experiments

Conclusion

Constraint Programming (CP) Model

- The energy constraints can be directly modelled as in the mathematical model:

$$energy[n] = \sum_{(t',t'') \in A} f[t'] \cdot s[t',t''] \cdot e[n, node[t'], node[t'']]$$

This is implemented via the *element* constraint (1988), which allows indexing a matrix by decision variables.

- We have added only $|A| \cdot |N|$ constraints and variables.

- Branching only on the *node* decision variables.

Outline

Platform

- CP solver: *Gecode* (version 3.4.0, open-source)
- IP solvers:
  - *Gurobi* (version 3.0.1, commercial)
  - *SCIP* (version 1.2.0)
  - *lp_solve* (version 5.5)
- Operating system: Mac OS X 10.6.3 (64-bit)
- CPU: Intel Core 2 Duo 2.53GHz, 3MB cache
- Memory: 4GB

UPPSALA UNIVERSITET

Task Mapping Problem

Models

Experiments

Conclusion

Results: Highway Traffic Management

Highway	*Gecode*			*Gurobi*		
$\langle n, t\rangle$	time	time_{opt}	cost	time	time_{opt}	cost
$\langle 7, 9\rangle$	0.001	**0.010**	20	< 1	0.03	20
$\langle 13, 18\rangle$	0.009	**0.024**	60	< 1	0.42	60
$\langle 19, 27\rangle$	0.022	**0.034**	100	< 1	8.12	100
$\langle 25, 36\rangle$	0.049	**0.060**	100	< 1	10.81	100
$\langle 32, 45\rangle$	0.091	**0.109**	100	< 1	7.48	100
$\langle 38, 54\rangle$	0.166	**0.222**	100	< 1	11.07	100
$\langle 44, 63\rangle$	0.264	**0.300**	100	< 1	45.50	100
$\langle 63, 90\rangle$	0.985	**1.048**	100	98	153.97	100
$\langle 74, 36\rangle$	0.549	> 600.000	300	38	> 600.00	300
$\langle 75, 108\rangle$	1.888	**2.007**	100	142	428.90	100
$\langle 88, 126\rangle$	3.350	**3.499**	100	1	117.80	100
$\langle 113, 162\rangle$	8.427	**8.756**	100	2	96.73	100
$\langle 124, 60\rangle$	3.155	**286.338**	300	165	> 600.00	300
$\langle 125, 180\rangle$	12.545	**12.956**	100	3	329.46	100
$\langle 138, 198\rangle$	17.598	**18.282**	100	421	546.33	100
$\langle 150, 216\rangle$	24.205	**25.033**	100	3	> 600.00	100

Outline

UPPSALA UNIVERSITET

Summary and Conclusion

- More efficient to use constraint programming (CP).
- CP model is at least competitive to published IP model.
- CP model captures the mathematical model directly, using CP technology of 1988.
- Similar performance of CP models (over published IP models) for quadratic assignment problems has been reported by Laurent Michel and Pascal Van Hentenryck at CPAIOR'08 + '09 and at CP'09 + '10.
- The whole setup and the constraints are different from task mapping in classical distributed systems:
  - Not just the end points pay the cost of routing.
  - There are energy constraints.

Future Work

- Implement heuristics and search procedure taking the structure of the task graph into account.
- Investigate impact of task computation costs.
- Challenge: Is there a better IP model?
- Solve the problem using stochastic local search.

UPPSALA UNIVERSITET

Acknowledgements

- This research is sponsored by the Swedish Foundation for Strategic Research (SSF) under research grant RIT08-0065 for the project *ProFuN: A Programming Platform for Future Wireless Sensor Networks*.

- We thank Animesh Pathak of INRIA Paris Rocquencourt (France) for useful discussions and the datasets.