MacroLab: A Vector-based Macroprogramming Framework for Cyber-Physical Systems

Timothy W. Hnat, Tamim I. Sookoor, Pieter Hooimeijer, Westley Weimer, and Kamin Whitehouse

Department of Computer Science
University of Virginia
Centralized vs. Distributed
Centralized vs. Distributed
Centralized vs. Distributed
Centralized vs. Distributed
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Topology and Logic Matter
Solution
Solution

lightValues = lightSensor.sense()
**Solution**

lightValues = lightSensor.sense()
lightOffsets = [1, 35, ... 23]
\[ X = \text{lightOffsets} + \text{lightValues} \]

<table>
<thead>
<tr>
<th>X</th>
<th>lightOffsets</th>
<th>lightValues</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

= +

![Diagram with microcontrollers and a laptop]
Solution

lightValues = lightSensor.sense()
lightOffsets = [1,35, ... 23]
X = lightOffsets + lightValues
Solution

\[
\text{lightValues} = \text{lightSensor.sense()}
\]

\[
\text{lightOffsets} = [1, 35, \ldots, 23]
\]

\[
X = \text{lightOffsets} + \text{avg(lightValues)}
\]
Solution

lightValues = lightSensor.sense()
lighOffsets = [1, 35, ... 23]
X = lightOffsets + avg(lightValues)
Solution

\[ \text{lightValues} = \text{lightSensor.sense()} \]
\[ \text{lightOffsets} = [1, 35, \ldots, 23] \]
\[ X = \text{lightOffsets} + \text{avg(lightValues)} \]

\[ \begin{align*}
X & \quad \text{lightOffsets} & \quad \text{lightValues} \\
\end{align*} \]
Contributions
Contributions

• Easy to use programming abstraction for scientists and engineers
Contributions

• Easy to use programming abstraction for scientists and engineers

• Automatically choose best decomposition
Outline

• Programming Abstraction
• Compilation
• Evaluation
• Conclusion
Macrovector

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>94</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>61</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>
## Macrovector

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>+</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>2 3</td>
<td></td>
<td>35 8 4</td>
</tr>
<tr>
<td>2</td>
<td>3 4</td>
<td></td>
<td>61 2 3</td>
</tr>
<tr>
<td>18</td>
<td>4 5</td>
<td>+</td>
<td>10 6 2</td>
</tr>
<tr>
<td>94</td>
<td>5 6</td>
<td></td>
<td>94 1 7</td>
</tr>
<tr>
<td>10</td>
<td>6 7</td>
<td></td>
<td>2 9 3</td>
</tr>
<tr>
<td>61</td>
<td>7 8</td>
<td></td>
<td>18 9 10</td>
</tr>
</tbody>
</table>
Macrovector

<p>| | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>10</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>6</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>9</td>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ C = \begin{bmatrix} 35 & 10 & 7 \\ 2 & 12 & 7 \\ 18 & 13 & 15 \\ 94 & 6 & 13 \\ 10 & 12 & 9 \\ 61 & 9 & 11 \end{bmatrix} \]

\[ A = \begin{bmatrix} 35 & 2 & 3 \\ 2 & 3 & 4 \\ 18 & 4 & 5 \\ 94 & 5 & 6 \\ 10 & 6 & 7 \\ 61 & 7 & 8 \end{bmatrix} \]

\[ B = \begin{bmatrix} 35 & 8 & 4 \\ 2 & 9 & 3 \\ 18 & 9 & 10 \\ 94 & 1 & 7 \\ 10 & 6 & 2 \\ 61 & 2 & 3 \end{bmatrix} \]

\[ C = A + B \]
Macrovector

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>16</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>6</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>24</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>5</td>
<td>42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>54</td>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>63</td>
<td>80</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>8</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>1</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[
C = A \times B
\]
Macrovector

\[ C \overset{\text{max}}{=} B \]

\[
\begin{array}{c|cc}
35 & 8 & 4 \\
2 & 9 & 3 \\
18 & 9 & 10 \\
94 & 1 & 7 \\
10 & 6 & 2 \\
61 & 2 & 3 \\
\end{array}
\]
Macrovector

\[
\begin{array}{cccc}
35 & 8 & \text{max} & 35 \\
2 & 9 & 3 & 2 \\
18 & 10 & 10 & 94 \\
94 & 7 & 7 & 94 \\
10 & 6 & 6 & 10 \\
61 & 3 & 3 & 61 \\
\end{array}
\]

C = max B
Distributed Macrovector

\[
\begin{array}{c|c}
35 & \begin{array}{c|c}
10 & 7 \\
12 & 7 \\
13 & 15 \\
6 & 13 \\
12 & 9 \\
9 & 11 \\
\end{array} \\
2 & \begin{array}{c|c}
8 & 4 \\
9 & 3 \\
9 & 10 \\
1 & 7 \\
6 & 2 \\
2 & 3 \\
\end{array} \\
18 & \begin{array}{c|c}
2 & 3 \\
3 & 4 \\
4 & 5 \\
5 & 6 \\
6 & 7 \\
7 & 8 \\
\end{array} \\
94 & \begin{array}{c|c}
35 & \end{array} \\
10 & \begin{array}{c|c}
 & \end{array} \\
61 & \begin{array}{c|c}
 & \end{array} \\
\hline
C & A \\
\hline
A & B \\
\end{array}
\]
Distributed Macrovector

\[
\begin{array}{cccc}
10 & 7 & = & 8 & 4 & + & 2 & 3 \\
12 & 7 & = & 9 & 3 & + & 3 & 4 & 2 \\
12 & 9 & = & 6 & 2 & + & 6 & 7 & 10 \\
9 & 11 & = & 2 & 3 & + & 7 & 8 \\
13 & 15 & = & 9 & 10 & + & 4 & 5 \\
6 & 13 & = & 1 & 7 & + & 5 & 6 & 94 \\
\end{array}
\]
Centralized Macrovector

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>10</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>13</td>
<td>15</td>
<td></td>
<td></td>
</tr>
<tr>
<td>94</td>
<td>6</td>
<td>13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>12</td>
<td>9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>61</td>
<td>9</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\[ C = A + B \]
Centralized Macrovector
Centralized Macrovector
Other representations
Other representations

Neighborhood Vector
Other representations

Neighborhood Vector

Reflected Vector
Outline

• Programming Abstraction

• Compilation

• Evaluation

• Conclusion
Sample Program

light = newMacrovector(motes)
offset = newMacrovector(motes)
X = newMacrovector(motes)

offset = [1, 35, ..., 25];

every(uint16(10000))
  light = lightSensors.sense();
  X = light + offset;
end
light = newMacrovector(motes)
distributed

offset = newMacrovector(motes)
distributed

X = newMacrovector(motes)
distributed

offset = [1,35,...,25];

every(uint16(10000))
    light = lightSensors.sense();
    X = light + offset;
end
Sample Program

distributed

centralized

distributed

light = newMacrovector(motes)
offset = newMacrovector(motes)
X = newMacrovector(motes)

offset = [1,35,\ldots,25];

every(uint16(10000))
  light = lightSensors.sense();
  X = light + offset;
end
light = newMacrovector(motes)
offset = newMacrovector(motes)
X = newMacrovector(motes)

offset = [1, 35, ..., 25];

every(uint16(10000))
    light = lightSensors.sense();
    X = light + offset;
end
Compilation

Decomposer
Compilation

Macroprogram

Decomposer
Compilation

Macroprogram

Decomposer

D,D,D

D,C,D

...

C,C,C
light = newMacrovector(motes)
offset = newMacrovector(motes)
X = newMacrovector(motes)

offset = [1,35,...,25];

every(uint16(10000))
  light = lightSensors.sense();
  X = light + offset;
end
Function Resolution

distributed  light = newMacrovector(motes)
distributed  offset = newMacrovector(motes)
centralized  X = newMacrovector(motes)

offset = \([1, 35, ..., 25]\);

every(uint16(10000))
  light = lightSensors.sense();
  X = light + offset;
end

Function Library
plusddc.m
plusccc.m
plusddd.d

Addition operation
input: distributed, distributed
output: centralized
Function Resolution

distributed

light = newMacrovector(motes)

distributed

offset = newMacrovector(motes)

centralized

X = newMacrovector(motes)

offset = [1,35,...,25];

every(uint16(10000))
    light = lightSensors.sense();
    X = plusddc(light,offset);
end

Function Library

plusddc.m
plusccc.m
plusddd.m
Sample Program

light = newMacrovector(motes)
offset = newMacrovector(motes)
X = newMacrovector(motes)

offset = [1,35,...,25];

every(uint16(10000))
  light = lightSensors.sense();
  X = plusddc(light,offset);
end
Sample Program

```plaintext
light = newMacrovector(motes)
offset = newMacrovector(motes)
X = newMacrovector(motes)

offset = [1,35,...,25];

every(uint16(10000))
    light = lightSensors.sense();
    X = plusddc(light,offset);
end
```

Base Specific Implementation

Node Specific Implementation
Compilation

Macroprogram

Decomposer

C,C,C
C,D,D
...
D,D,D
Compilation

Macroprogram

Decomposer

C,C,C → C,D,D → ... → D,D,D

Cost Analysis

Topology

Embedded Matlab Compiler

C Source Code
Compilation

Macroprogram

Decomposer

C,C,C

C,D,D

...

D,D,D

Cost Analysis

Topology

Embedded Matlab Compiler

C Source Code

TinyOS RTS

nesC Compiler
Compilation

Macroprogram

Decomposer

C,C,C

C,D,D

...

D,D,D

Cost Analysis

Topology

Embedded Matlab Compiler

- getNodeID()
- Messaging
- Time Synchronization

C Source Code

TinyOS RTS

nesC Compiler
Compilation

Macroprogram

Decomposer

C,C,C

C,D,D

... 

D,D,D

Topology

Cost Analysis

Embedded Matlab Compiler

C Source Code

TinyOS RTS

nesC Compiler

Binary
Outline

• Programming Abstraction
• Compilation
• Evaluation
• Conclusion
## Lines of Code

<table>
<thead>
<tr>
<th></th>
<th>nesC/TinyOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection (Surge)</td>
<td>400</td>
</tr>
<tr>
<td>Tracking (PEG)</td>
<td>780</td>
</tr>
</tbody>
</table>
## Lines of Code

<table>
<thead>
<tr>
<th></th>
<th>nesC/TinyOS</th>
<th>MacroLab</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Collection (Surge)</td>
<td>400</td>
<td>7</td>
</tr>
<tr>
<td>Tracking (PEG)</td>
<td>780</td>
<td>19</td>
</tr>
<tr>
<td>Platform</td>
<td>ROM Size</td>
<td>RAM Size</td>
</tr>
<tr>
<td>----------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>TelosB</td>
<td>49,152</td>
<td>10,240</td>
</tr>
<tr>
<td>MICAz</td>
<td>131,072</td>
<td>4,096</td>
</tr>
<tr>
<td>Platform</td>
<td>ROM Size</td>
<td>RAM Size</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------</td>
<td>----------</td>
</tr>
<tr>
<td>TelosB</td>
<td>49,152</td>
<td>10,240</td>
</tr>
<tr>
<td>MICAz</td>
<td>131,072</td>
<td>4,096</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>Program Size</th>
<th>Heap Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurgeTelos</td>
<td>24,790</td>
<td>911</td>
</tr>
<tr>
<td>PEG</td>
<td>61,440</td>
<td>3,072</td>
</tr>
<tr>
<td></td>
<td>Platform</td>
<td>ROM Size</td>
</tr>
<tr>
<td>----------</td>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>TelosB</td>
<td></td>
<td>49,152</td>
</tr>
<tr>
<td>MICAz</td>
<td></td>
<td>131,072</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Application</th>
<th>Program Size</th>
<th>Heap Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurgeTelos</td>
<td></td>
<td>24,790</td>
<td>911</td>
</tr>
<tr>
<td>PEG</td>
<td></td>
<td>61,440</td>
<td>3,072</td>
</tr>
<tr>
<td>MacroLab_Surge</td>
<td></td>
<td>19,374</td>
<td>669</td>
</tr>
<tr>
<td>MacroLab_PEG</td>
<td></td>
<td>18,536</td>
<td>770</td>
</tr>
<tr>
<td>Platform</td>
<td>ROM Size</td>
<td>RAM Size</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td>TelosB</td>
<td>49,152</td>
<td>10,240</td>
<td></td>
</tr>
<tr>
<td>MICAz</td>
<td>131,072</td>
<td>4,096</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>Program Size</th>
<th>Heap Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurgeTelos</td>
<td>24,790</td>
<td>911</td>
</tr>
<tr>
<td>PEG</td>
<td>61,440</td>
<td>3,072</td>
</tr>
<tr>
<td>MacroLab_Surge</td>
<td>19,374</td>
<td>669</td>
</tr>
<tr>
<td>MacroLab_PEG</td>
<td>18,536</td>
<td>770</td>
</tr>
<tr>
<td>Blink</td>
<td>2,472</td>
<td>38</td>
</tr>
<tr>
<td>CountRadio</td>
<td>11,266</td>
<td>351</td>
</tr>
<tr>
<td>Oscilloscope</td>
<td>9,034</td>
<td>335</td>
</tr>
<tr>
<td>OscilloscopeRF</td>
<td>14,536</td>
<td>449</td>
</tr>
<tr>
<td>SenseToRfm</td>
<td>14,248</td>
<td>403</td>
</tr>
<tr>
<td>TOSBase</td>
<td>10,328</td>
<td>1,827</td>
</tr>
<tr>
<td>Platform</td>
<td>ROM Size</td>
<td>RAM Size</td>
</tr>
<tr>
<td>--------------</td>
<td>----------</td>
<td>----------</td>
</tr>
<tr>
<td>TelosB</td>
<td>49,152</td>
<td>10,240</td>
</tr>
<tr>
<td>MICAz</td>
<td>131,072</td>
<td>4,096</td>
</tr>
<tr>
<td>MacroLab_Surge</td>
<td>19,374</td>
<td>669</td>
</tr>
<tr>
<td>MacroLab_PEG</td>
<td>18,536</td>
<td>770</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Application</th>
<th>Program Size</th>
<th>Heap Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>SurgeTelos</td>
<td>24,790</td>
<td>911</td>
</tr>
<tr>
<td>PEG</td>
<td>61,440</td>
<td>3,072</td>
</tr>
<tr>
<td>MacroLab_Surge</td>
<td>19,374</td>
<td>669</td>
</tr>
<tr>
<td>MacroLab_PEG</td>
<td>18,536</td>
<td>770</td>
</tr>
</tbody>
</table>

| MacroLab_Surge            | 1,822        | 191       |
| MacroLab_PEG              | 1,702        | 90        |
Almost no overhead

<table>
<thead>
<tr>
<th>Application</th>
<th>Execution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surge</td>
<td>17.7 msec</td>
</tr>
<tr>
<td>MacroLab_Surge</td>
<td>18.2 msec</td>
</tr>
</tbody>
</table>
Almost no overhead

<table>
<thead>
<tr>
<th>Application</th>
<th>Execution</th>
<th>Stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>Surge</td>
<td>17.7 msec</td>
<td>120 bytes</td>
</tr>
<tr>
<td>MacroLab_Surge</td>
<td>18.2 msec</td>
<td>124 bytes</td>
</tr>
</tbody>
</table>
Estimate: 12:42pm
Arrival: 12:44pm
Bus Tracking

RTS = RunTimeSystem();
busstops = RTS.getNodes('stopnode');
buses = RTS.getNodes('bus');
estimates = Macrovector(busstops, length(buses) , 'uint16');
arrivals = Macrovector(busstops, length(buses) , 'uint16');
travelTime = Macrovector(busstops, length(busstops), length(buses) , 'uint16');
busSensors = SensorVector('BusSensor', busstops, 'uint16');
routes = uint8({[1 2 3 4], [ 5 6 7 8]}); %Example routes

every(1000)
    [busID,r] = busSensors.sense();
    busTime = RTS.getTime();
    travelTime(routes{r},routes{r},busID)[1,3] = busTime - arrivals(routes{r}, busID);
    arrivals(routes{r},busID)[1,2] = busTime;
    estimates(routes{r},busID) = travelTime(routes{r},routes{r},busID)[2,3] + busTime;
    baseDisplay(estimates(routes{r},:));
end
Bus Tracking

```matlab
RTS = RunTimeSystem();
busstops = RTS.getNodes('stopnode');
buses = RTS.getNodes('bus');
estimates = Macrovector(busstops, length(buses) , 'uint16');
arrivals = Macrovector(busstops, length(buses) , 'uint16');
travelTime = Macrovector(busstops, length(busstops), length(buses) , 'uint16');
busSensors = SensorVector('BusSensor',busstops,'uint16');
routes = uint8({[1 2 3 4], [5 6 7 8]}); %Example routes

every(1000)
    [busID,r] =  busSensors.sense();
    busTime = RTS.getTime();
    travelTime(routes{r},routes{r},busID)[1,3] = busTime - arrivals(routes{r}, busID);
    arrivals(routes{r},busID)[1,2] = busTime;
    estimates(routes{r},busID) = travelTime(routes{r},routes{r},busID)[2,3] + busTime;
    baseDisplay(estimates(routes{r},:));
end
```
RTS = RunTimeSystem();
busstops = RTS.getNodes('stopnode');
buses = RTS.getNodes('bus');
estimates = Macrovector(busstops, length(buses) , 'uint16');
arrivals = Macrovector(busstops, length(buses) , 'uint16');
travelTime = Macrovector(busstops, length(busstops), length(buses) , 'uint16');
busSensors = SensorVector('BusSensor',busstops,'uint16');
routes = uint8({[1 2 3 4], [ 5 6 7 8]}); %Example routes

every(1000)
    [busID,r] =  busSensors.sense();
    busTime = RTS.getTime();
    travelTime(routes{r},routes{r},busID)[1,3] = busTime - arrivals(routes{r}, busID);
    arrivals(routes{r},busID)[1,2] = busTime;
    estimates(routes{r},busID) = travelTime(routes{r},routes{r},busID)[2,3] + busTime;
    baseDisplay(estimates(routes{r},:));
end
RTS = RunTimeSystem();
busstops = RTS.getNodes('stopnode');
buses = RTS.getNodes('bus');
estimates = Macrovector(busstops, length(buses), 'uint16');
arrivals = Macrovector(busstops, length(buses), 'uint16');
travelTime = Macrovector(busstops, length(busstops), length(buses), 'uint16');
busSensors = SensorVector('BusSensor', busstops, 'uint16');
routes = uint8([[[1 2 3 4], [5 6 7 8]]]; %Example routes

every(1000)
    [busID,r] = busSensors.sense();
    busTime = RTS.getTime();
    travelTime(routes{r},routes{r},busID)[1,3] = busTime - arrivals(routes{r}, busID);
    arrivals(routes{r},busID)[1,2] = busTime;
    estimates(routes{r},busID) = travelTime(routes{r},routes{r},busID)[2,3] + busTime;
    baseDisplay(estimates(routes{r},:));
end
Simulation Results

Centralized Display

Centralized Implementation

Distributed Implementation
Simulation Results

![Simulation Results Diagram]

Centralized Display

- Centralized Implementation
- Distributed Implementation
Simulation Results

Centralized Display

<table>
<thead>
<tr>
<th>Centralized Implementation</th>
<th>9,000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>8,000</td>
</tr>
<tr>
<td></td>
<td>7,000</td>
</tr>
<tr>
<td></td>
<td>6,000</td>
</tr>
<tr>
<td></td>
<td>5,000</td>
</tr>
<tr>
<td></td>
<td>4,000</td>
</tr>
<tr>
<td></td>
<td>3,000</td>
</tr>
<tr>
<td></td>
<td>2,000</td>
</tr>
<tr>
<td></td>
<td>1,000</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

48

Distributed Implementation

Centralized Implementation

Distributed Implementation
Simulation Results

- Distributed Display: 8,156
- Centralized Implementation: 4,273
Simulation Results

Distributed Display

Centralized Implementation

Distributed Implementation
Simulation Results

Centralized Implementation

Distributed Implementation
Simulation Results
Simulation Results

Base Station Range

- Centralized Implementation
- Distributed Implementation
Simulation Results

Base Station Range

- Centralized Implementation
- Distributed Implementation

Bar chart showing:
- Centralized Implementation: 1,468
- Distributed Implementation: 2,472
Simulation Results

Base Station Range

0
1,000
2,000
3,000
4,000
5,000
6,000
7,000
8,000
9,000

1,468

Centralized Implementation
Distributed Implementation
Simulation Results

Base Station Range

1,468

2,472

Centralized Implementation

Distributed Implementation
Simulation Results

- Additional Route
- Centralized Implementation: 6,210
- Distributed Implementation: 3,439
Simulation Results

Centralized Implementation

Distributed Implementation
Simulation Results

Additional Route

Centralized Implementation

Distributed Implementation
Related Work
Related Work

- Node level programming
- TinyOS, LiteOS, MantisOS, and Contiki
Related Work

• Node level programming
  • TinyOS, LiteOS, MantisOS, and Contiki

• Macroprogramming systems
  • TinyDB and Cougar: Database
  • Hood, Regions, and Proto: Areas
  • Semantic Streams, Flask, and Regiment: Streams
Related Work

- Node level programming
  - TinyOS, LiteOS, MantisOS, and Contiki
- Macroprogramming systems
  - TinyDB and Cougar: **Database**
  - Hood, Regions, and Proto: **Areas**
  - Semantic Streams, Flask, and Regiment: **Streams**
  - Marionette and Pleiades: **Imperative**
Related Work
Related Work

• User specified distribution

• High Performance Fortran and Split-C
Related Work

- User specified distribution
- High Performance Fortran and Split-C
- Automatic decomposition
- MagnetOS, Coign, and J-Orchestra
Related Work

• User specified distribution
• High Performance Fortran and Split-C
• Automatic decomposition
• MagnetOS, Coign, and J-Orchestra
• Parallel Languages
• SET Language, *Lisp, and NESL
Limitations and Future Work

• Limitations
• Mobile Networks
• Mobile Software Agents

• Future Work
• Quality of service
• Automatic adaption to changes
Conclusions
Conclusions

• A user writes simple, easy to understand macroprograms
Conclusions

- A user writes simple, easy to understand macroprograms
- Implementations are chosen automatically for the best performance
MacroLab is available at:
http://www.cs.virginia.edu/hnat/MacroLab/

hnat@cs.virginia.edu
Backup Slides
Actual Breakdown
## Actual Breakdown

<table>
<thead>
<tr>
<th>RTS (ROM/RAM)</th>
<th>MacroLab (ROM/RAM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>558/66</td>
<td>1,264/125</td>
</tr>
<tr>
<td>558/66</td>
<td>1,144/24</td>
</tr>
</tbody>
</table>
Dot-Product
Simulation Results

Figure 11. Neither decomposition is best for all deployment scenarios. Small changes in the deployment scenario changes the optimal implementation between centralized and distributed.
Power Measurement

Figure 8. Oscilloscope power measurements of MacroLab and nesC Surge implementations.
Figure 12. Changing the base station range changes the balance between the two decompositions.
Route

Figure 13. Adding a new route at various distances changes the balance between the two decompositions.
Static Cost

Figure 14. Estimated and measured messaging costs. The parameter $\alpha$ is the ratio of buses on long routes versus those on all other routes.