Making Programming the Fourth ‘R of Literacy

Arun Chauhan, Indiana University

Undergraduate Honors Seminar
School of Informatics and Computing
Indiana University
Dec 7, 2009
“What our community should really aim for is the development of a curriculum that turns our subject into the fourth R—as in ’rogramming—of our education systems.

... A form of mathematics can be used as a full-fledged programming language, just like Turing Machines.”

Matthias Felleisen and Shriram Krishnamurthy
Communications of the ACM, Jul 2009
“Why can’t you be like the Math Department, which only needs a blackboard and wastepaper basket? Better still, like the Department of Philosophy. That doesn’t even need a wastepaper basket …”

Arthur C. Clarke
3001: The Final Odyssey
Computers are for Computing and ...

- Computers as general-purpose tools
  - communication, navigation, data collection, entertainment, etc.

- Computers as computing tools
  - problem solving
  - data processing and analysis
TIOBE Index

TIOBE Index Chart

Normalized fraction of total hits (%)

Dates: 2002 to 2010

Languages: Java, PHP, Visual Basic, Python, Perl, C, C++, C#, JavaScript, Ruby
## TIOBE: Top 20

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>=</td>
<td>Java</td>
<td>17.061%</td>
<td>-2.31%</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>=</td>
<td>C</td>
<td>16.285%</td>
<td>+0.12%</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>↑</td>
<td>PHP</td>
<td>9.770%</td>
<td>+0.29%</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>↓</td>
<td>C++</td>
<td>9.175%</td>
<td>-1.72%</td>
<td>A</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>=</td>
<td>(Visual) Basic</td>
<td>7.778%</td>
<td>-1.70%</td>
<td>A</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>=</td>
<td>C#</td>
<td>6.258%</td>
<td>+1.61%</td>
<td>A</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>=</td>
<td>Python</td>
<td>5.185%</td>
<td>+0.62%</td>
<td>A</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>↑</td>
<td>JavaScript</td>
<td>3.515%</td>
<td>+0.45%</td>
<td>A</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>↓</td>
<td>Perl</td>
<td>2.692%</td>
<td>-0.91%</td>
<td>A</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>↑</td>
<td>Ruby</td>
<td>2.653%</td>
<td>+0.34%</td>
<td>A</td>
</tr>
<tr>
<td>11</td>
<td>10</td>
<td>↓</td>
<td>Delphi</td>
<td>2.301%</td>
<td>-0.75%</td>
<td>A</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>↑</td>
<td>PL/SQL</td>
<td>1.494%</td>
<td>+0.35%</td>
<td>A</td>
</tr>
<tr>
<td>13</td>
<td>35</td>
<td></td>
<td>Objective-C</td>
<td>1.159%</td>
<td>+1.00%</td>
<td>A</td>
</tr>
<tr>
<td>14</td>
<td>14</td>
<td>=</td>
<td>SAS</td>
<td>0.911%</td>
<td>+0.07%</td>
<td>A</td>
</tr>
<tr>
<td>15</td>
<td>19</td>
<td>↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑</td>
<td>Lisp/Scheme</td>
<td>0.881%</td>
<td>+0.37%</td>
<td>A--</td>
</tr>
<tr>
<td>16</td>
<td>17</td>
<td>↑</td>
<td>ABAP</td>
<td>0.723%</td>
<td>+0.12%</td>
<td>A-</td>
</tr>
<tr>
<td>17</td>
<td>15</td>
<td>↓</td>
<td>Pascal</td>
<td>0.698%</td>
<td>+0.01%</td>
<td>B</td>
</tr>
<tr>
<td>18</td>
<td>21</td>
<td>↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑↑</td>
<td>ActionScript</td>
<td>0.655%</td>
<td>+0.17%</td>
<td>B</td>
</tr>
<tr>
<td>19</td>
<td>12</td>
<td></td>
<td>D</td>
<td>0.587%</td>
<td>-0.60%</td>
<td>B</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
<td>=</td>
<td>Lua</td>
<td>0.585%</td>
<td>+0.09%</td>
<td>B</td>
</tr>
</tbody>
</table>
Teaching Programming

Table 1.

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>...</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>9</td>
<td>16</td>
<td>?</td>
<td>...</td>
<td>?</td>
</tr>
</tbody>
</table>
Teaching Programming

Figure 1.

\[ \text{placeImage (rocket, 25, 0, )} \]

Figure 2.

\[ \text{placeImage (rocket, 25, 0, )} = \]

\[ \text{rocket} \]
Teaching Programming

Table 2.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>10</td>
<td>20</td>
<td>30</td>
<td>...</td>
<td>height(t) = ?</td>
</tr>
</tbody>
</table>

Table 3.

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>...</th>
<th>t</th>
</tr>
</thead>
<tbody>
<tr>
<td>rocket(t) = ?</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Teaching Programming

Figure 3.

\[ \text{rocket}(t) = \text{placeImage} ( \bullet, 25, 10 \cdot t, ) \]

Figure 4.

\[ \text{rocket}(t) = \text{placeImage} ( \bullet, 25, \text{height}(t), ) \]
Scratch
http://scratch.mit.edu/
Problem

- Nice programming languages
  - domain-specific
  - often dynamically typed and interpreted
- Poor performance
  - inefficient use of computing resources
  - inefficient use of energy
“It is a capital mistake to theorize before one has data. Insensibly one begins to twist facts to suit theories, instead of theories to suit facts.”

Sir Arthur Conan Doyle
A Scandal in Bohemia
One Slide Primer on Locality

**Memory**

```
x
y
```

**Cache**

```
x
y
```

Temporal locality

Spatial locality

x = 10;
...
y = x + 2;
An Empirical Study

Implementing A Big Expression

- AMD Opteron
- PowerPC 970 (Apple G5)
- Intel Xeon
- Intel Itanium 2

Vector size (thousands of double elements) / time(parenthesized) / time(distributed)
Challenge #1: There is no mathematical model for data locality
Reuse Distances

\[ x = a + b; \]
\[ c = a + d[i] \times 100; \]
\[ y = x \times 10; \]

Reuse Distance = 6 (a, b, c, d, i, 100)
Concurrency Trends
(ExaScale Computing Study, Peter Kogge et al.)

![Concurrency Trends Graph](image-url)
Parallelism is Useful!
Other Applications

- Climate modeling
- Molecular dynamics
- N-body simulation
- Multi-grid simulation
- Financial and economic modeling
- ...
Challenge #2:
There is no easy way to write parallel programs
Types of (Parallel) Programmers

- Mainstream Parallelism-Oblivious Developers
  - Joe needs high level Programming Models designed for Domain Experts

- Parallelism–Aware Developers
  - Stephanie needs simple Parallel Programming Models with safety nets

- Concurrency Experts
  - Focus of today’s Parallel Programming Models

Courtesy: Vivek Sarkar, Rice University
One Slide Primer on Parallelism

Shared Memory

\[ x = 10; \]
\[ \ldots \]
\[ y = x + 2; \]

Distributed Memory

\[ x = 20; \]
\[ \ldots \]
\[ y = x + 2; \]
Parallelism Oblivious Users

- Programming languages-driven
  - implicit parallelism, compiler support
- Operating System-driven
  - innovative solutions to leverage extra cores
- Architecture-driven
  - Instruction-level parallelism, hyper-threading
Observations for Parallelism-Aware and Expert Users

- Completely automatic parallelization has had limited success

- Writing parallel programs is hard; optimizing and maintaining them is harder!

- Compilation technology has worked well in communication optimization
Concluding Remarks

- Educating the next generation for the fourth ‘R
  - Computing is a core technique in an increasing number of fields
  - programming is no longer restricted to scientists and engineers
- Taking care of non-expert programmers
  - an exponentially growing class
  - locality and parallelism problems
- Solving problems for expert programmers
  - tools to address computational bottlenecks
Toward Exascale ($10^{18}$)
What Should You Do?

- Educate yourself in the basics
  - computer architecture
  - programming languages
  - compilers
- Learn parallel programming!
Research Interests

- High-level Languages
  - Ruby, MATLAB, R, etc.
- Heterogeneous parallel computing
- Large memory-footprint applications
- Automatic parallelization
http://www.cs.indiana.edu/~achauhan

http://phi.cs.indiana.edu/