Twelve New Programming Languages: Is Cloud Responsible?

Do we need programming languages?

You may think that the answer is no. But, if you go by the recent trend you may need to change your mind.

Consider for example the following questions:

- Why is Google working two (GO, DART) new programming languages?
- Why has IBM (X10), Cray (Chapel) and Red Hat (Ceylon) working on creating a new programming language of its own?
- Why did the attendees of a recent conference in London select 5 (HTML5, DART, Scala, Clojure, Node.js) new languages as the most important software development trends for 2012?
- What does Neil McAllister mean when he says that these 10 (DART, Ceylon, GO, F#, OPA, Fantom, Zimbu, X10, Haxe, Chapel) programming languages that could shake up IT?
- Why did Anders Hejlsberg of Microsoft, the creator of Turbo Pascal, Delphi and C# launch TypeScript?
Are new programming languages getting created just to satisfy the creative urge of somebody or has the technology evolution created the need for new way of programming and hence these new languages?

If you look back...

There are 3 periods in history when there was a burst of new programming languages. Each of these periods is linked to a critical point in technology evolution.

<table>
<thead>
<tr>
<th>Period</th>
<th>Driver</th>
<th>Languages</th>
</tr>
</thead>
<tbody>
<tr>
<td>1957-64</td>
<td>Invention of compiler and the era of 3GL</td>
<td>(1) Fortran, (2) Algol, (3) Lisp, (4) Cobol, (5) RPG, (6) APL, (7) Simula, (8) Basic and (9) PL/1</td>
</tr>
<tr>
<td>1978-84</td>
<td>Invention of RDBMS and the era of the 4GL</td>
<td>(1) SQL, (2) dBase, (3) C++, (4) Oracle Forms and PL/SQL, (5) Informix 4GL, (6) Gupta SQL, (7) Unify Accell and (8) Ingress</td>
</tr>
</tbody>
</table>

Other important languages which was created outside these time period are: (1) Pascal – 1970, (2) C – 1972, (3) Prolog – 1972, (4) Smalltalk – 1972, (5) Erlang – 1986, (6) Perl – 1987 and (7) C# – 2001. As you can see, they are very few.

Are we entering another period which on hindsight will be classified as another period of technology transition? Only time can tell.

Does cloud computing got anything to do with it?

It most certainly does. But that is not the whole story – there is more to it.

What could the Technology Drivers be?

If you analyze the primary motivation behind these languages, you will see several common themes emerging.
(1) Distributed/Parallel computing:

Programmability of parallel computers (Chapel from Cray, X10 from IBM), Concurrent programming (Clojure, Fantom, Go) and single language for complete cloud stack (Opa)

Cloud computing is all about distributing your process across multiple CPU and running them in parallel or concurrently. Current programming languages are not very well suited for that.

Hence this attempt to create languages tailored for parallel processing.

(2) Multi-paradigm programming:

Languages which support object oriented programming as well as functional programming (Clojure, F#, Fantom, Scala)

Functional programs are relatively easy to parallelize. However, pure functional languages have not been very successful.

Hence this attempt to create multi-paradigm programming language.

(3) Multi-platform programming:

Languages that has implementation for both JVM and CLR (Clojure, Fantom, Scala) and several other platforms (Clojure & Fantom on JavaScript engine, Scala on Android, Haxe on several platforms)

Well this is a dream which we have been chasing for decades. Will it ever be a reality? We never know.

Hence this attempt to create multi-platform language!

(4) Programming at Scale:

Readability, Modularity (Ceylon), Speed of compilation (Go, Zimbu), Suitable for large team (Scala), Meta-programmability and extensibility (Ceylon, Scala), Concise (Scala, Zimbu)

Cloud provides almost unlimited computing power which encourages us to build larger and more complex application. Building such application will require large teams and the code has to be easily maintainable.

Also, the focus on frequent integration requires the process of compilation to be speeded up.

Hence this attempt to create programming language for large application.

(5) Client side programming:

Replacement for JavaScript (Dart), same language for both client and server side (Naxe, Opa, Zimbu)

Cloud computing requires client side code to seamlessly run on multiple client devices. Though JavaScript has become a standard, nobody really likes it!

Hence this attempt to create languages which is better than JavaScript and which can run on multiple client platforms.

Quick overview of the 12 new languages mentioned earlier

Here are the 12 languages listed in an alphabetical order. (I am not including HTML5 in this list as it is strictly
<table>
<thead>
<tr>
<th>Language</th>
<th>Who is behind it?</th>
<th>Primary Driver</th>
<th>Licensing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ceylon</td>
<td>Red Hat [Home page]</td>
<td>Readability, Predictability, Tool-ability, Modularity, Meta-programmability. Runs on JVM</td>
<td>GPL v2</td>
</tr>
<tr>
<td>Chapel</td>
<td>Cray [Home page]</td>
<td>Programmability of parallel computers. Run on Cray supercomputers and various high-performance clusters. Portable to most Unix-style systems, Mac OS X and Windows</td>
<td>BSD</td>
</tr>
<tr>
<td>Clojure</td>
<td>Rich Hickey</td>
<td>Concurrency using Functional programming paradigm. Runs on JVM, CLR, and JavaScript engines</td>
<td>EPL</td>
</tr>
<tr>
<td>Dart</td>
<td>Google [Home page]</td>
<td>A replacement for JavaScript on the browser. Faster, easier to maintain, and less susceptible to subtle bugs. Dart VM needs to be compiled – can run on Linux, Mac and Windows</td>
<td>New BSD</td>
</tr>
<tr>
<td>Fantom</td>
<td>Brian Frank, Andy Frank [Home page]</td>
<td>Portability, support for functional programming and concurrency. Runs on JVM and CLR. Is also compiles to javascript. Future targets might include Objective-C for the iPhone</td>
<td>Academic Free License</td>
</tr>
<tr>
<td>Go</td>
<td>Google [Home page]</td>
<td>Compiled with the ease of programming of a dynamic language, concurrency and communication, speed of compilation. Compiler available for Linux, Mac OS X, Windows</td>
<td>BSD style + patent grant</td>
</tr>
<tr>
<td>Haxe</td>
<td>Nicolas Cannasse</td>
<td>Multi-platform support. Compiler for JavaScript, Flash, NekoVM, PHP, C++. C# and Java support is expected</td>
<td>GPL v2</td>
</tr>
<tr>
<td><strong>MLstate</strong> - <a href="http://cloudcomputing.sys-con.com/node/2265359">Home page</a></td>
<td>Targeted for cloud computing. Client-side UI, server-side logic, and database I/O are all implemented in a single language. Runtime environment. Own Web server and DBMS. Runs on 64bit Linux and Mac</td>
<td>AGPL</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td><strong>IBM</strong> - <a href="http://cloudcomputing.sys-con.com/node/2265359">Home page</a></td>
<td>Designed specifically for parallel programming, performance and scale. Runs on IBM AIX, Linux, Mac OS X, Windows</td>
<td>EPL</td>
<td></td>
</tr>
<tr>
<td><strong>Bram Moolenaar</strong> - <a href="http://cloudcomputing.sys-con.com/node/2265359">Home page</a></td>
<td>Aims to be fast, concise, portable, and easy-to-read and support GUI application to an OS kernel. Compiles to ANSI C</td>
<td>Apache v2</td>
<td></td>
</tr>
</tbody>
</table>

**Finally...**

What is most interesting is that all the 12 languages are available under some form of open source license.

**Interesting articles on history of programming languages:**

- [The History of Programming Languages [Infographic]](http://cloudcomputing.sys-con.com/node/2265359)
- [Programming Language History](http://cloudcomputing.sys-con.com/node/2265359)
- [A Brief, Incomplete, and Mostly Wrong History of Programming Languages](http://cloudcomputing.sys-con.com/node/2265359)

**Statistics on Language usage**

- [TIOBE Programming Community Index](http://cloudcomputing.sys-con.com/node/2265359)
- [Programming Language Popularity](http://cloudcomputing.sys-con.com/node/2265359)
- [Most Popular Programming Languages](http://cloudcomputing.sys-con.com/node/2265359)

© 2008 SYS-CON Media Inc.