DART: A STRUCTURED WEB PROGRAMMING LANGUAGE

- New programming language
- New programming tools
- New open source project

WARNING: THIS IS ONLY A TECHNOLOGY PREVIEW
SPEAKER INTRODUCTION

Lars Bak

Gilad Bracha

Beta
Self
OOVM
V8

Dart
Strongtalk
JVMs

JLS
Newspeak

25 years

DART

Monday, October 10, 2011
DART AGENDA

• The motivation
• The language
• Code samples
• Demos
• The open source project

Monday, October 10, 2011
CURRENT WEB: THE GOOD PARTS

- Developing small applications is easy
- Platform independence
- No installation of applications
- Supports incremental development
- ... and it is everywhere
CURRENT WEB: THE BAD PARTS

- Developing large scale applications is hard
- Hard to find the program structure
- Lack of static types
- No support for libraries
- Tools support is weak
- Startup performance is bad
INNOVATION IS ESSENTIAL

- We believe Dart fills a vacuum
- The competition is NOT JavaScript
- ... but fragmented mobile platforms
HISTORY BEHIND DART

• After several years working on V8 for Chrome
• Last year, Kasper Lund & I experimented with Spot
  • a new simple language for the web
  • based on the experiences from JavaScript
• Spot was the prelude for the Dart project
SO WHAT IS DART?

- A simple and unsurprising OO programming language
- Class-based single inheritance with interfaces
- Optional static types
- Real lexical scoping
- Single-threaded
- Familiar syntax
A DIFFERENT TYPE-CHECKER

• A conventional type-checker is a lobotomized theorem prover
• Tries to prove program obeys type system
• If it can't construct a proof - program is considered invalid
  “Guilty until proven innocent”

• In Dart, you are **innocent until proven guilty**
DART TYPES AT RUNTIME

- *During development one can choose to validate types*
  
  ```dart
  T x = o; assert(o == null || o is T);
  ```

- *By default, type annotations have no effect and no cost*

- *Code runs free*
OPTIONAL TYPES

- Static checker provides warnings; tuned to be unobtrusive.
- Type annotations have no effect except ...
- During development, you can check dynamic types against declarations.
ISOLATES

- Inspired by Erlang, Dart has isolates
- Lightweight units of execution
  - Each isolate is conceptually a process
  - Nothing is shared
  - All communication takes place via message passing
- Isolates support concurrent execution
DART IS NOT DONE

• Reflection support?
• Rest arguments and enums?
• Pattern matching for easy message decoding like in Erlang?
• What about Chrome?
• ... please give feedback by joining the discussions

Monday, October 10, 2011
# DART PERFORMANCE

Relative performance compared to JavaScript on V8

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>VM</th>
<th>DartC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandelbrot:</td>
<td>18.2%</td>
<td>88.7%</td>
</tr>
<tr>
<td>DeltaBlue:</td>
<td>56.6%</td>
<td>52.2%</td>
</tr>
<tr>
<td>Richards:</td>
<td>46.0%</td>
<td>70.9%</td>
</tr>
<tr>
<td>NBody:</td>
<td>35.8%</td>
<td>63.6%</td>
</tr>
<tr>
<td>BinaryTrees:</td>
<td>77.3%</td>
<td>104.3%</td>
</tr>
<tr>
<td>Fannkuch:</td>
<td>53.8%</td>
<td>22.3%</td>
</tr>
<tr>
<td>Meteor:</td>
<td>50.3%</td>
<td>42.1%</td>
</tr>
</tbody>
</table>

Details:
- V8 revision 3.5.5.
- DartC used with the -optimize flag
SNAPSHOTTING IN THE DART VM

- Process of serializing the heap after loading the application
- Loading 54173 lines of Dart code takes 640 ms
- Loading same application from a snapshot takes 60 ms
- Startup > 10x faster
WEB APPLICATION IN DART

- Newsreader completely written in Dart
- App code: 3210 LOC
- UI library code: 13200 LOC
- Animation yields 30 fps
- Code is part of the open source project
DART EDITOR

- Editor for constructing and browsing Dart applications
- Lightweight editor based on Eclipse components
- Code is part of the open source project
DART OPEN SOURCE PROJECT

- The Dart web site: http://dartlang.org
- Dart language specification
- Dart language tutorial
- The Dart project: http://dart.googlecode.com
- Libraries and code samples
- Dart virtual machine
- Dart to JavaScript compiler
DART IS A TECHNOLOGY PREVIEW

- Dart: a programming language for the web
- Two execution modes, Dart VM or JS Engine
- Compatible with the current web
- Please try it out and participate

Monday, October 10, 2011