a fast and flexible object serializer

Martín Dias
DC-FCEN-UBA, Buenos Aires
Tristan Bourgois
developer

Martín Dias
developer

Stéphane Ducasse
promotor & financer

2010

SummerTalk 2011

Martín Dias
developer

Mariano Martinez Peck
mentor & developer
what is serialization?
serialization to other technologies
serialization
to similar environments
serialization
classes & traits
creation vs. reference
environment dependency
flexibility
class changes

<table>
<thead>
<tr>
<th>Point</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>v.1</td>
<td></td>
</tr>
<tr>
<td>x</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Point</th>
<th>v.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td></td>
</tr>
<tr>
<td>distanceToZero</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Point</th>
<th>v.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>posX</td>
<td></td>
</tr>
<tr>
<td>posY</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>v.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>posX</td>
<td></td>
</tr>
<tr>
<td>posY</td>
<td></td>
</tr>
</tbody>
</table>
how to serialize?
format

<?xml version="1.0"
<header objects="1"
<body>
<object class="Pl"
    <variable name=""
</object>
<object class="Co"
    <variable name=""
</object>

vs.

01011101010010
10001010110101
01010010101111
01010010010100
01101010
algorithms
example
<table>
<thead>
<tr>
<th>hr</th>
<th>instances</th>
<th>references</th>
<th>tr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cluster #1</td>
<td>cluster #2</td>
<td>cluster #3</td>
</tr>
<tr>
<td></td>
<td>player #1…4</td>
<td>blue</td>
<td>255</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th># clusters</th>
<th>type</th>
<th>class</th>
<th># instances</th>
<th>type</th>
<th>class</th>
<th># instances</th>
<th>int #1</th>
<th>index of player #2</th>
<th>index of blue</th>
<th>index of player #3</th>
<th>index of blue</th>
<th>…</th>
<th>index of 16rFF</th>
<th>root</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>Obj Player</td>
<td>4</td>
<td>Obj Color</td>
<td>1</td>
<td>Int</td>
<td>1 255</td>
<td>2 5 3 5</td>
<td>…</td>
<td>6 1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
algorithms
special VM-level implementation
what about fuel?
Fuel

congrete

flexible

fast

design
speed
large graphs

- Fuel
- StOMP
- SRP
- ImageSegment
- SmartRefStream
- Magma

[Bar chart showing serialization vs materialization speed for different graph processing systems]
speed
different graph sizes
design

1892 lines of code
7.2 methods per class
3.7 lines per method
188 unit tests
90% test coverage
1837 lines of test code
demo

load seaside without compilation!
demo

revive your exceptions!
future

v.2 → v.3

<table>
<thead>
<tr>
<th>x</th>
<th>posX</th>
</tr>
</thead>
<tbody>
<tr>
<td>y</td>
<td>posY</td>
</tr>
</tbody>
</table>

ImageSegment
SmartRefStream
ReferenceStream
DataStream

ClassA
  └> ClassB

ClassA
  └> ClassB

ClassA
  └> ClassB

ClassC
  └> ClassC

ClassC
  └> ClassC