MrLazy: Lazy Runtime Label Propagation for MapReduce

Sherif Akoush, Lucian Carata, Ripduman Sohan, and Andy Hopper

HotCloud 2014
June 2014
The NHS plan to share our medical data can save lives – but must be done right

Care.data, the grand project to make the medical records of the UK population available for scientific and commercial use, is not inherently evil – far from it – but its execution has been badly bungled. Here's how the government can regain our trust
England players' passport numbers revealed in teamsheet blunder

- last updated Wed 4 Jun 2014

<table>
<thead>
<tr>
<th>NO.</th>
<th>NAME / FIRST NAME(s)</th>
<th>DATE OF BIRTH</th>
<th>PASSPORT NO.</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>Foster Ben (GK)</td>
<td>03/04/1983</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Milner James</td>
<td>04/01/1986</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Jones Phil</td>
<td>21/02/1992</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Smalling Chris</td>
<td>22/11/1989</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Lampard Frank</td>
<td>20/05/1986</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Wilshere Jack</td>
<td>01/01/1992</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Oxладe-Chamberlain Alex</td>
<td>15/08/1993</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Barkley Ross</td>
<td>05/12/1993</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Rooney Wayne</td>
<td>24/10/1985</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Lambert Rickie</td>
<td>16/02/1982</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Forster Fraser (GK)</td>
<td>17/03/1988</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Henderson Jordan</td>
<td>17/06/1990</td>
<td></td>
</tr>
</tbody>
</table>

ITV News
Information Flow Control (IFC)

• IFC*
  • Propagate Record + Sensitivity Metadata
  • Control Information Flow by Checking Metadata against Policies

• But...
  • Many In-House Computations
  • No Need for Active Checking
  • Only When Publishing Some Results

• Lazy IFC
  • Track and Use Lineage
  • Evaluate Output Labels When Needed

Labels (Metadata)

• More than one Label per Record
  • Different Country Regulations, Data Quality...

• Field-Level

• Dynamic Properties
  • Users Opting In/Out
  • Sensitivity of Data Expires in 2 Years
  • New Policies
MapReduce Paradigm

DFS

64 MB

Split 1

Split 2

Split N

Map

Reduce

Shuffle

DFS

64 MB

File 1

File 2

$((K_{IN}, V_{IN}), (K_{MED}, V_{MED}))$

$((K_{MED}, \text{List}(V_{MED})), (K_{OUT}, V_{OUT}))$
Record-Level Lineage for MapReduce
Lazy IFC for MapReduce

Split 1
Split 2
Split N

Map
Reduce
Shuffle

DFS

(K_{IN}, V_{IN})
(K_{MED}, V_{MED})
(K_{MED, List (V_{MED})})
(K_{OUT}, V_{OUT})

l1, q1
l2, q2
l3, q3
l4, q4
l5, q5

fn(x)
fn(x)
fn(x)
fn(x)
fn(x)

a, 2
b, 3
Lineage Capture in Hadoop MapReduce

• Record-Level Lineage
• No Changes to User Code
• Always-On Feature
  • Treat Lineage for Map and Reduce Tasks \textit{Separately}
  • Lineage Reconstruction
Field-Level Enforcement

• One Record Can Have Fields With Different Sensitivity
  • Player Name vs. Passport Number
• Field-Level (Conservative) Visibility By Static Analysis

```java
map(Text key, Text value)
{
  String str[] = value.toString().split(“,”)
  Text name = new Text (str[0])
  write(name, 1)
}
```
Prototype Evaluation

• Implementation in Hadoop MapReduce
• 7-node Cluster
• Dataset from BigDataBench: 120 GB
• Join and Filter Job
Overheads (Lineage Capture)

- **Storage**
  - 50% of Output
  - Delete When Not Needed
  - Trading Space for Time

![Bar chart showing runtime and lineage reconstruction overheads for Base and With Lineage scenarios.](chart.png)
Policy 1: Users Opt-out of Data Sharing

- 5% of Users
Policy 2: Sensitivity of Data Lasts 2 Years

- Dynamic Behaviour

![Graph showing dynamic behaviour between Naive (Recomputation) and MrLazy. The Naive (Recomputation) bar is significantly higher than MrLazy.](graph.png)
Other Challenges

• Dealing with State
  • In-lining Instructions to Expose State
  • TopK
• Subtle Data Leakage
  • Differential Privacy
Conclusion

• Delay Output Label (Metadata) Computation
• Fine-Grained Lineage as Audit Mechanism
• Non-Prohibitive Overheads

Future Work:
• Reducing Overheads
• Large-Scale Evaluation
• Recomputation-Based Recovery from Failures
Thanks

Sherif.Akoush@cl.cam.ac.uk
http://www.cl.cam.ac.uk/~sa497/