A DSL for Distributed Reactive Workflows

Mathijs Saey, Joeri De Koster, Wolfgang De Meuter

Context

We are producing a lot of data, we need software that reacts to this data instantaneously.

Scale of data forces us to execute on a cluster. Need to deal with partial failure, replication, consistency, ...

Problem Statement

We want a programming language which allows one to write scalable, reactive big data applications from a set of existing, reactive components.

Related Work

<table>
<thead>
<tr>
<th></th>
<th>Reactive Programming</th>
<th>Stream Processing</th>
<th>Scientific Workflows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reactive</td>
<td>✓</td>
<td>?</td>
<td>×</td>
</tr>
<tr>
<td>Scalable</td>
<td>×</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Existing Components</td>
<td>?</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Write Reactive Components

```plaintext
component Distance, in: [p1, p2], out: [distance] do
  react (x1, y1), (x2, y2) do
    distance = sqrt(square(x2 - x1) + square(y2 - y1)) ~> distance
end
```

```
component GeoFilter, in: [json], out: [inside, outside] do
  fields area
  init area_json do
    area = area_json
  end
  react json do
    loc = ... # extract location from json
    r = System.cmd "in_area" ["--area " + area, loc]
    if r == "inside", do: json ~> inside, else: json ~> outside
  end
```

Compose Reactive Workflows

```
Data Store

Count

Geo Filter

Data Store

Geo Visualisation

Skitter passes messages between connected components and activates react and other functions.

Effects

```
component Count, in: [any], out: [current] do
  effect state_change
  fields count
  init _ do
    count = 0
  end
  react _ do
    count = count + 1
    count ~> current
  end
```

Execute on a Cluster

```
Skitter automatically handles replication and partial failure handling based on the effects of a component.
```

Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Effect</th>
<th>Additional Primitives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mutuable state</td>
<td>state_change</td>
<td></td>
</tr>
<tr>
<td>Foreign process with mutable state</td>
<td>state_change hidden</td>
<td>create_checkpoint, restore_checkpoint, clean_checkpoint</td>
</tr>
<tr>
<td>I/O may occur</td>
<td>external_effect</td>
<td></td>
</tr>
</tbody>
</table>

From scratch

Interface between component and workflow called by runtime when data is present. Publish data to connected components.

As a wrapper for foreign code

Each token entering a workflow is processed concurrently. Skitter will replicate individual components as needed.